University of Glasgow

College of Medical, Veterinary, and Life Sciences

Artificial Neural Networks for Problems in Computational Cognition

Author:
Henry Powell (BA, MA)

Supervisor:
Prof. Emily Cross

Submitted in fulfilment of
the requirements for the
degree of Doctor of
Philosophy

October 30, 2022

Abstract

Computationally modelling human level cognitive abilities is one of the principal goals of artificial intelligence research, one that draws together work from the human neurosciences, psychology, cognitive science, computer science, and mathematics. In the past 30 years, work towards this goal has been substantially accelerated by the development of neural network approaches, at least in part due to advances in algorithms that can train these networks efficiently [Rumelhart et al., 1986b] and computer hardware that is optimised for matrix computations [Krizhevsky et al., 2012]. Parallel to this body of work, research in social robotics has developed to the extent that embodied and socially intelligent artificial agents are becoming parts of our everyday lives. Where robots were traditionally placed as tools to be used to improve the efficiency of a number of industrial tasks, now they are increasingly expected to emulate humans in complex, dynamic, and unpredictable social environments. In such cases, endowing these robotic platforms with (approaching) human–like cognitive capabilities will significantly improve the efficacy of these systems, and likely see their uptake quicken as they come to be seen as safe, effective, and flexible partners in socially oriented situations such as physical healthcare, education, mental well–being, and commerce. Taken together, it would seem that neural network approaches are well placed to allow us to bestow these agents with the kinds of cognitive abilities that they require to meet this goal. However, the nascent nature of the interaction of these two fields and the risk that comes along with integrating social robots too quickly into high risk social areas, means that there is significant work still to be done before we can convince ourselves that neural networks are the right approach to this problem.

In this thesis I contribute theoretical and empirical work that lends weight to the argument that neural network approaches are well suited to modelling human cognition for use in social robots. In Chapter 1 I provide a general introduction to human cognition and neural networks and motivate the use of these approaches to problems in social robotics and human–robot interaction. This chapter is written in such a way that readers with no technical background can get a good understanding of the concepts that are at the center of the thesis' aims. In Chapter 2, I provide a more in–depth and technical overview of the mathematical concepts that are at the heart of modern neural networks, specifically detailing the logic behind the deep learning approaches that are used in the empirical chapters of the thesis. While a full understanding of this chapter requires a stronger mathematical background than the previous chapter, the concepts are explained in such a way that a non–technical reader should come out of it with a solid high level understanding of these ideas. Chapters Chapter 3 through Chapter 5 contain the empirical work that was carried out in order to attempt to answer the above questions. Specif-

ically, Chapter 3 explores the viability of using deep learning as an approach to modelling human social–cognitive abilities by looking at the problems of subjective psychological stress and self–disclosure. I test a number of "off-the-shelf" deep learning architectures on a novel dataset and find that in all cases these models are able to score significantly above average on the task of classifying audio segments in relation to how much the person performing the contained utterance believed themselves to be stressed and performing an act of self-disclosure. In Chapter 4, I develop the work on subjective-self disclosure modelling in human–robot social interaction by collecting a much larger multi modal dataset that contains video recorded interactions between participants and a Pepper robot. I provide a novel multi-modal deep learning attention architecture, and a custom loss function, and compare the performance of our model to a number of non-neural network approach baselines. I find that all versions of our model significantly outperform the baseline approaches, and that our novel loss improves on performance when compared to other standard loss functions for regression and classification problems for subjective self–disclosure modelling. In Chapter 5, I move away from deep learning and consider how neural network models based more concretely on contemporary computational neuroscience might be used to bestow artificial agents with human like cognitive abilities. Here, I detail a novel biological neural network algorithm that is able to solve cognitive planning problems by producing short path solutions on graphs. I show how a number of such planning problems can be framed as graph traversal problem and show how our algorithm is able to form solutions to these problems in a number of experimental settings. Finally, in Chapter 6 I provide a final overview of this empirical work and explain its impact both within and without academia before outlining a number of limitations of the approaches that were used and discuss some potentially fruitful avenues for future research in these areas.

Acknowledgements

This work is dedicated first and foremost to my parents who have provided unending support and encouragement throughout my life and the many pivots in my career.

Enormous gratitude is also given to my supervisor Emily Cross who trusted me to plot my own course throughout my Ph.D. None of this would have been possible without her continuous feedback and optimism for my work, and I have no doubt that its quality would have suffered considerably without her input. The work contained in this thesis suffered innumerable set backs due to COVID-19 and other unforeseeable issues, and my motivation to continue throughout all of that would not have been possible without her compassion and understanding.

I also owe considerable thanks to my other supervisors throughout my academic career, especially Steve Butterfill, whose mentorship and continuous care over my development made my switch from philosophy to the sciences possible. I would not be where I am without him.

Thanks also to the labs that I worked in during my internships: the Robotics, Brain, and Cognitive, Sciences Lab at IIT, Genoa, and the A.I. team at Merck KGaA, Darmstadt, Germany. Special thanks goes to Alessandra Sciutti and Helmut Linde, the heads of those two labs, for their mentorship.

I am likewise hugely grateful to Chris Manser for playing support, and to Office, without whom my post graduate studies would have been completed much sooner and with considerably less trouble.

My thanks also to SoBots lab at the University of Glasgow for their consistent feedback and friendship. My development as a scientist was immeasurably improved by being part of such a diverse and talented lab.

Finally, to my wife who followed me around the world without question and accepted all the risks that have lead us both to this wonderful point in our lives. None of this would have been possible without her unending love and support.

Contents

List of Figures

List of Tables

Author's Declaration

I declare that, except where explicit reference is made to the contribution of others, that this thesis is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution.

Printed name: Henry Powell
Signature:

Author Contributions

Chapter 3:

HP: Conceptualisation, data pre-processing, data modification, model building, model training, model analysis, writing, visualisation. **GL**: Conceptualisation, dataset design, data collection, writing. **J-NL**: Data collection. **ESC**: Conceptualisation, supervision, writing.

Chapter 4:

HP: Conceptualisation, data-preprocessing, data modification, model design, model building, model training, model analysis, data analysis, writing, visualisation. **GL**: Conceptualisation, data set design, data collection. **ESC**: Conceptualisation, supervision, writing.

Chapter 5:

HP: Conceptualisation, model design, model building, model experimentation, writing. **MW**: Model building, model experimentation, writing. **AVH**: Writing. **HL**: Conceptualisation, supervision, writing.

Key:
HP: Henry Powell, **GL:** Guy Laban, **J-NL:** Jean-Nöel George, **ESC:** Emily S. Cross, **MW:** Mathias Winkel, **AVH**, Alexander V. Hopp, **HL:** Helmut Linde.

Notation

a	A scalar value.
$\mathbf{a}$	A vector.
M	A matrix.
W	A weight matrix.
W_{ji}	A weight matrix for the weights *to* the j^{th} layer *from* the i^{th} layer of the network.
W^r_{ji}	As above but this time denoting that the connection is recurrent.
w_{ji}	A weight *to* the j^{th} neuron *from* the i^{th} neuron.
w^r_{ji}	As above but denoting that the connection is recurrent.
b	The bias of a neuron.
$\mathbf{b}$	A bias vector, collecting the biases for that layer of the network.
$\mathbf{s}$	The state-layer vector collecting the output of the neurons in the state layer.
$\mathbf{h}$	The hidden-layer vector collecting the output of the neurons in a hidden layer.
$\mathbf{y}$	The output-layer vector collecting the output of the neurons in the output layer.
x	A single input to a neuron.
$\mathbf{x}$	The input vector collecting the inputs to a network.

Chapter 1

General Introduction

This chapter provides an introduction and overview of the concepts and fields of research that the empirical chapters of the thesis are embedded in. Namely, computational cognition, artificial neural networks, human–robot interaction, and social robotics. To being with, I discuss cognition, providing a general and uncontroversial definition, as well as some intuitive examples of different cognitive capabilities. I then give some insight, by way of a worked example, into how we might divide up the space of cognition, and human cognitive capabilities, so that we are equipped to differentiate aspects of cognition with respect to their character and how they appear to us in experience. I then elaborate on how we might think about *computational* cognition, what it means for something to be computational with respect to the way in which it is studied, and what the benefits of thinking about cognition in a computational way are. Next, I introduce neural networks by first giving a simple introduction to biological neurons. I discuss the structure of a typical neocortical neuron and the way in which it can be thought to process information in the form of small electrical impulses. This leads on to an example of a computational model of a human neuron: the perceptron. Since the perceptron is at the heart of many of the neural network models that are used in this thesis, I go into some depth regarding how the perceptron models the decision making process of a biological neuron, what the mathematical model for its function is, and how these neurons can be strung together into networks in order to solve a number of interesting problems that reflect human cognitive capabilities. In the following section I introduce deep learning, a type of neural network approach that organises a large number of perceptron style neurons in a specific way. I talk about why we tend to organize these computational neurons like this and why this particular structure naturally lends these networks to modelling cognition. I then outline why these networks can be hard to train and what conditions need to be met in order for them to perform well in the real world. Next, I introduce the concept of human–robot interaction and one of it's subfields, known as social

robotics. I give a brief definition of what social robots are and in what kinds of situations they may turn out to be useful in. This discussion sheds light on some of the issues that are currently ongoing in the field of research that sits at the intersection of deep learning and social robotics. I talk about why there is a gap at this academic juncture that requires serious and careful consideration, and then argue that this thesis attempts to demonstrate how this gap can be dealt with appropriately. In the penultimate section, I provide an overview of the research principals that guided the empirical work contained in the central chapters, and give an overview of the work conducted as well as its results. Finally, I review and tie together all of these discussions and give a precise motivation for the work that follows.

1.1 Computational Cognition

1.1.1 Cognition

While a thorough analysis of the meaning of cognition is well beyond the scope of this introduction, it is worth trying to encapsulate some of the key, and uncontroversial, concepts that belong to it.

Loosely defined, cognition can be understood to be the brain's manipulation, handling, representation, and storage of knowledge (defined in some appropriate way) accumulated from within and without the body. This includes, but is certainly not limited to, perception, action (including motor control and language production), reasoning, learning, and problems related to memory. Since, this list captures a vast range of complex phenomena, it is worth making an attempt to form a rough taxonomy to provide greater clarity. Daniel Kahneman, in his book "Thinking, fast and slow" [Kahneman, 2011], separates cognitive processes into two broad classes, which have come to be known as *type-1* and *type-2*. Intuitively, we can think of these two kinds of process as being distinct with respect to the time scale over which they take place and how much conscious control we have over them. Type-1 processes are taken to be fast and automatic whereas type-2 processes are thought to be slow and deliberate. Their difference can be made more distinct by considering the following example:

As I sit at my desk I realise that I feel thirsty (perhaps my mouth is dry) and decide that I want to drink something. I look to the right of my computer monitor and see that I have both a bottle of water and cup of coffee. The water seems more appealing at first; it's hot in the room, I'm thirsty, and the condensation collecting on the side of the bottle makes it look particularly appealing. I then realise that I'm feeling tiered, given that it's still somewhat early in the morning,

and I reason that drinking the coffee would provide me with the energy boost that I need to complete my morning tasks. I weigh up the two alternatives and decide that feeling more awake trumps having a satisfying drink and so I reach for the cup of coffee. Upon grasping the cup I feel that it is still extremely hot causing my hand to recoil. In the end I decide to let my coffee cool a bit and take a drink from my water bottle instead.

This example, while somewhat contrived (I could, of course, just drink both) describes a range of phenomena that come about as the result of, or are directly, different kinds of cognitive processes: the sensation of having a dry mouth and feeling thirsty, scanning the environment and identifying the bottle of water and mug of coffee, inferring the result of drinking each of these, reasoning about the pros and cons of drinking each of these, deciding on drinking from the cup of coffee, performing a reach-to-grasp action towards the cup of coffee, recoiling from the coffee mug after feeling how hot it is, and finally, reaching for, and drinking from the bottle of water. We can, given some time, loosely categorise some of these into type-1 and type-2 processes. Feeling thirsty, locating and recognising the coffee and water, reaching for the coffee, and pulling my hand away from the mug are all fast and automatic processes which feasibly do not involve representing those things in conscious thought. They happen, in some way, at a level beneath consciousness, and perhaps for the best, given that it would be incredibly difficult to function effectively in our day to day lives if we had to consciously represent, plan, and act upon all of these. On the other hand, thinking about the respective effects of drinking coffee and water, comparing and contrasting these effects with respect to different goals (drinking water would quench my thirst me and cool me down, whereas drinking coffee would quench my thirst and wake me up), formulating a plan to let the coffee cool before drinking it and drinking from the water instead are all higher level conscious process that perhaps involve explicitly representing aspects of these things in my conscious experience. The former, which we can label as type-1 processes, occur over a matter of milliseconds, and appear to us to be automatically performed, whereas the latter, which we can refer to as type-2 processes, occur in the order of seconds and appear to us as if we have far more control over them.

There are, of course, some obvious problems with this account. For instance, some of these processes do not seem to obviously fit as well into one of the two classes as others. Reaching for and grasping the cup of coffee seems as if it could occur over a matter of seconds and could thus be argued to be more of a type-2 cognitive process. What is more, surely I have more control over this motoric action than the one that is involved when my hand recoils from the coffee mug. I can easily choose to maneuver around an unforeseen obstacle were it to fall into

the path of my reach but it doesn't seem to so obviously be the case that I have the same control over the short path my hand takes as it releases and moves away from the hot coffee mug. Indeed, scholars have raised a number of well-established theoretical and empirical issues with dual process theories of cognition (for a more in-depth discussion of these see [Evans, 2011][Evans and Stanovich, 2013]). The above is not an attempt to argue that dual process theories are the correct (or only) way of thinking about cognition. Instead, the definition of cognition that was offered at the start of this section, the explication of dual processes of cognition, and the coffee vs. water drinking example are presented as a means to illustrate how we might think about what kinds of things we mean when we talk about cognition and the ways in which we might go about thinking about different kinds of cognitive processes. In computer science, and specifically in the field of machine learning, we can be relatively agnostic with respect to different strict definitions of cognition. That being said, it should be clear from an understanding of the above when a particular study is attempting to model something that is cognitive, such as facial recognition, speech understanding, gesture recognition and interpretation, and emotion recognition, and something that is not, such as atmospheres of exoplanets, migratory patterns of starlings, avalanche risk, and so on.

1.1.2 Computation

With this basic primer of cognition in place, we can now consider what exactly we mean when talk about *computational* cognition. It should come as no surprise that computational cognition involves the use of computers. More specifically, computational cognition specifies a field of study that seeks to understand or replicate one or many cognitive processes naturally occurring within the human brain by means of computational modelling. While this might appear to imply that the use of a computer is a requirement, a computational model can also be straightforwardly mathematical in nature, i.e. one that involves no implementation onto a computer system.

This raises the question as to why we might want to involve mathematics and computers into the study of cognition at all, seeing as questions related to our abilities to perform cognitive tasks are plausibly best answered by more empirical approaches. More specifically, if we wanted to know something about how the brain is able to recognise faces from a visual scene, surely the best thing to do is to collect empirical data via something like functional magnetic resonance imagining while a subject is performing some kind of task related to recognising human faces? In reality, this framing is misleading as it suggests that computational approaches

and ones that we describe as more empirical in nature are mutually exclusive. In fact, computational approaches to cognition often work in tandem with other approaches, and more often than not, will base their models on observations that have been collected via empirical means. For instance, take the now famous Haken, Kelso, Bunz equation (adapted from the original found in [Haken et al., 1985]):

$$\dot{\phi} = -a \sin \phi - 2k \sin 2\phi \tag{1.1}$$

The specifics of the equation are largely irrelevant. What's important is that the equation itself describes a particular type of behaviour that occurs during human bi-manual coordination. During a number of behavioural experiments, the authors noticed that when participants were asked to wag their index fingers at specific frequencies either in or out-of-phase with the index finger on their opposite hand, a number of interesting phenomena occurred. Firstly, they noticed that in-phase wagging was more stable and easier to maintain at higher frequencies. Second, that out-of-phase wagging was easy to maintain at lower frequencies. As the required wagging frequency increased, the experimenters noticed that the participant's synchronised out-of phase wagging eventually broke down into a chaotic asynchronous state before quickly reemerging as synchronised in-phase wagging. This result is made especially surprising by the fact that subjects claimed that they were not intentionally trying to change the phase of their fingers. Equation (1.1) is a computational model of this behaviour. It explains how the phase relation of the two fingers ϕ changes as a function of some parameters a and k i.e. for different values of a and k the equation will evaluate to either a positive (in-phase) or negative (out-of-phase) value. It is worth restating here that even if this model were to never be implemented in some kind of computer system, it is still a computational model.

But how does this provide more insight than a simple description of the behaviour? After all, a description, such as the one provided after Equation (1.1), is easier to understand and ostensibly provides the same information. The first thing to note is that while they do, in some sense, provide the same information, the equation is able to answer more specific questions about that behaviour such as: at what specific frequencies of out-of-phase behavior does the transition to in-phase behaviour occur? What phase of behaviour can I expect if I start wagging my fingers at a specific frequency? What is more, the model provides us with a means by which we can generate very specific research hypotheses. For instance, the equation suggests that the behaviour should have certain symmetries. One such symmetry would be that this phase transitioning behaviour should be invariant

under handedness i.e. it shouldn't matter, in out-of-phase behaviour, if the initial starting position of the hands is with the left finger raised and the right finger retracted or vice versa. This is a testable hypothesis that we can either confirm or deny by undertaking further empirical study. In the case that the hypothesis is confirmed to be false, we can then adapt our model to reflect this new knowledge that we have gained about the phenomena. Further, computational models provide a principled means by which we can extend research beyond the spheres in which it was originally conducted. A reasonable question that we might ask upon the conclusion of the experiments that lead to the Haken, Kelso, Bunz equation is: to what other physical or cognitive systems does this equation apply? Just fingers and hands? Or can we use it to predict how two peoples' gaits will synchronise as they walk next to each other [van Ulzen et al., 2008]? Or perhaps how oscillating neurons in connected parts of the brain will change their behaviour in relation to one another's firing patterns [Jirsa et al., 1998]? Perhaps it might also describe how different aspects of learning and memory might function [Pellecchia et al., 2005]. In each of these cases, having the behaviour formulated as a computational model allows us to capture a very broad range of behaviours in a concise and well-formulated manner and in way that makes it clear how these fields of research might be related. Many of these extensions to other fields may well yield a confirmation of the null-hypothesis, as it does in [van Ulzen et al., 2008], but these results still contribute significantly to their respective fields.

There is, of course, one more advantage to computational models (specifically in relation to the aims of this thesis) that has not yet been mentioned. Namely, that they allow us to implement the given behaviours into computational systems that can then be used to emulate those behaviours to some predetermined effect. This is the point at which two fields of research diverge. In the previous case, we were interested in deriving computational models because they told us something about the cognitive system that we were interested in studying. That is, they contributed to answering a predetermined set of empirical questions. Distinct from this, we might not be so interested in precisely how the particular cognitive system performs its function, but only in replicating that ability to some sufficient degree. For example, a researcher might be interested in facial recognition and use models from the computational neurosciences to develop an algorithm that was able to do this with the aim of implementing that algorithm in an embodied robot or computer application. Similarly, an engineer may want to use the Haken, Kelso, Bunz equation to ensure that the motor-cognitive capabilities of his humanoid robot matched those of a real human being.

For the sake of conceptual clarity, it is worth, at this point, dividing the field of computational cognitive research into two categories so that I might situate

the aims of this thesis more concretely. I have already suggested that researchers may be interested in using computational models to either understand a cognitive system more concretely, or to emulate those cognitive systems to some effect. Thus we can loosely divide computational cognition research into what I will describe as being either *empirically* or *engineering* focused. That is, research in this domain is primarily concerned with either answering empirical questions about a cognitive system, or trying to engineer a cognitive system to perform some task. Of course, much like dual systems theories, these are not strict or well-defined categories and research exists that aims at doing both. The purpose of making this distinction is to clarify where this thesis falls with respect to its aims. Stated briefly, the focus of this thesis falls in line with the latter of these categories. That is, the research that follows aims to computationally model a given set of cognitive abilities to some sufficient and well defined degree. The specifics of which cognitive systems were chosen to investigate will be introduced in depth at the end of this introduction. For now, it is more important to raise a question. Namely, what approach should be chosen to create the computational models that we will use to perform the tasks that we are interested in?

To help answer this question it is useful to reformulate the problem as one of function approximation (in the mathematical sense). That is, how can we find a function that describes a specific cognitive behaviour? In the case of the Haken, Kelso, Bunz equation, a function was found that describes how an output (a value describing the phase of two oscillating fingers) can be determined from an input (some values of a and k). Similarly, in the facial recognition case, how can we find a function that maps a set of inputs (perhaps pixels from a digital photograph) to an output (a location or area of the photograph that contains a human face)? Posed in this rather abstract way, the modelling of a whole array of cognitive abilities can be formulated as function approximation problems. And so, the question of what approach to use to create our computational models of cognition becomes: what is the best way of approximating a function that describes a particular cognitive ability? The benefit of posing the question in this way is that it makes it clear that the modelling problem can potentially be answered by any approach that is able to approximate functions in a satisfactory way.

As it turns out, at least two ways exist in which we can try to approximate these functions. First, we can choose to model them by deriving equations analytically from empirical observations and then fine tune the terms of those equations according to how well they fit our observations. While this approach is perhaps more intuitive, it becomes less tractable as the function one is trying to model becomes more and more complex. In these cases, we can instead pose the problem in a particular way (namely, as an optimisation problem) and have our compu-

tational model be learned in some well defined algorithmic way. Since cognitive abilities such as facial recognition, memory, motor planning, and emotion recognition involve a number of distinct parts of the brain [Eichenbaum and Lipton, 2008][Penhune and Steele, 2012][Schupp et al., 2006] it is likely that a model that sufficiently captures their abilities will be extremely complex. In these cases, having a system that is able to learn that model is highly desirable. One such means by which we can learn these complicated functions is via (artificial) neural networks.

1.2 Neural Networks

As the name suggests, neural networks are collections of computationally modeled biological neurons (such as those found in the human brain) that are connected to one another to form a network. Before we discuss how these neurons can be modelled and how they are connected together, it is worth reviewing some basic concepts from neurobiology that will help to illustrate the different component parts of these networks.

1.2.1 Biological Neurons

The human brain is composed of roughly 10^{11} neurons [Gurney, 2018]. Each neuron is connected to up to thousands of other neurons by means of fibrous arms called axons. Signals are sent from one neuron to another by means of electrical impulses that pass from the cell body, down the axon, and to a neighbouring cell via that subsequent cell's dendrites. The passing of this electrical signal is mediated by synapses that sit at the end of the many axon terminals. The informational currency of the brain consists of the electrical signals that are passed between the neurons. The precise measure of this information is determined by a number of factors. Each neuron will receive potentially thousands of input signals from its neighbouring neurons a given point in time in a cognitive process. The receiving neuron will need someway of combining these incoming signals and producing an output which it can then send its neighbours further down in the network. The biological specifics of how these incoming signals are collated is not too important. What is important is the idea that these incoming signals are first modified by some synaptic weight. That is, a particular synaptic junction between two neurons is able to modify the amplitude of the in coming signal in a fixed. A slightly more abstract way to think about this process is that a neuron's dendritic system is able to weight incoming information according to how important that information is to the cognitive process that is occurring. If an incoming signal from a neighbouring axon is in some way more important to that cognitive process that

another incoming signal then the more important signal will be amplified (and the less important one diminished) such that its effect on the firing of the neighbouring neuron is more pronounced. This means that each neuron will not simply be receiving the input from a neuron earlier on in the chain. More accurately, each neuron will receive a *weighted* input from all other neurons to which its dendrites are connected. The next step in determining what the output of a neuron looks like is to combine these weighted inputs in some way. One way that we can think of this happening is through simple summation. That is, the weighted inputs to a neuron are simply summed together by the neuron's cell body to form a single current of some magnitude. The final, and perhaps most important, step in the process is determine whether the neuron fires at all given the weighted sum of its inputs. In practice it is undesirable to have all of a network's neurons firing at the same time as this can overload the system and cause catastrophic epileptic behaviour that prevents any useful function from happening at all. We thus require some means of determining exactly when a neuron should fire. Crudely speaking, we might only want a neuron to fire when it receives a sufficiently large signal from its neighbours, thus ruling out the possibility that it could fire upon receiving some very small amount of unintentionally leaked output charge from a neighbouring neuron. To formalize this mechanism we can think of neurons as having an output threshold i.e. some level of charge that is required to be reached in order for that neuron to fire. Collected together, we can think of a simplified model of a neuron as consisting of a number of central components: an operation that weights and sums incoming signals and a threshold function that allows the neuron to fire on the condition that its weighted sum is above some predetermined value. The proper function of a neuron organised in this way produces the characteristic spiking behaviour (this behaviour can be seen in models constructed in [Izhikevich, 2003]) that can be observed when the membrane potential of a particular neuron is measured over the course of some cognitive process. As can be seen from Section 1.2.1 charge is slowly built up over a very short time, until it reaches some threshold, at which point the neuron fires (spikes to a particular value) and then resets. For the sake of completion it is worth noting that this resetting is performed by connections to types of neurons known as *inhibitory* neurons, whose outgoing signal inhibits the activity of a connected neuron after it fires to ensure that it is reset. In fact, neurons in the neocortex (the outer layer of the brain that is in part responsible for much of what we consider to be higher level cognitive functioning [Wiltgen et al., 2004][Adolphs, 2002][Kimppa et al., 2015]) can be roughly grouped into two types: excitatory (i.e. those that produce the spiking behaviour that allows electrical impulses to be sent down a neuronal chain) and

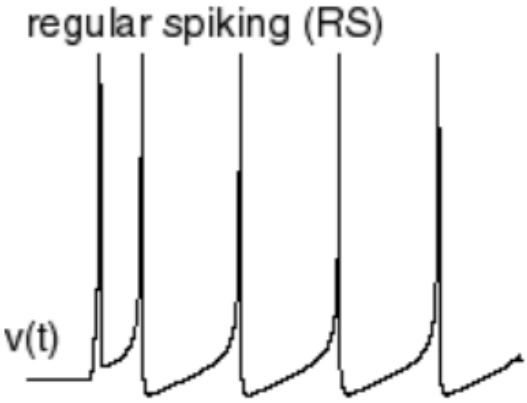


Figure 1.1: Example of spiking neuron behaviour observed from a Izhikevich neuron. Adapted from [Izhikevich, 2003]. $v(t)$ signifies the electrical potential of the neuron over time.

inhibitory (those that, amoung other things, inhibit the firing of other neurons to prevent epileptic overloading of the network).

1.2.2 The Perceptron

Neurons are the computational substrate of human cognition. That is to say, loosely speaking, that any given human cognitive ability can be broken down into tasks that are performed by particular cells and networks of cells in the brain. Facial recognition, for instance, can, at least in part, be traced to the operation of interconnected neuronal networks that exist throughout the visual cortices, patches of cortical tissue distributed on the caudal region of the neocortex. In turn, the operation of these networks can be broken down into tasks that can be thought to function by virtue of smaller collections of neurons. A small collection of neurons in V1, for instance, might fire when a particular low level pattern (such an edge or a direction of a stimulus) is detected in a region of a person's visual field [Movshon et al., 1978][DeAngelis et al., 1995]. Collections of these neuronal clusters will therefore fire together when a specific collection of patterns is detected. The detection of this set of patterns via collected electrical signals might then cause a small collection of neurons in V2 to fire, which have learned to fire when the input patterns form a human nose or some other more complex facial representation [Willmore et al., 2010]. This same logic would then apply to all component parts of a human face, getting increasingly abstract as you move through the different networks in the visual cortices (edges to lines, lines to nose shapes, nose shapes to face shapes and so on). This would eventually allow some cluster of neurons at the final level in the chain to recognise a particular face. In sum, the complex cognitive task of facial recognition has been broken down into a number of sub-

tasks arranged into a kind of conceptual hierarchy which, at its core, functions as the result of very simple operations occurring at the level of individual neurons.

Computational models of neural networks, used within the context of an engineering framework, can be seen as an attempt to emulate this computational functionality. That is, perhaps a whole range of cognitive processes can be engineered by constructing networks of artificial neurons in the correct way. At the core of these models will be the artificial neuron itself, a computational model of a biological neuron that, when connected together with other neurons modeled in this way, will be able to emulate human cognitive capabilities. Reframed from the perspective of function approximation, we can say that the goal of neural network research from a engineering perspective is to create and train a network of artificial neurons so that it is able to approximate a particular function i.e. given some input, that neural network should produce an output that is as close as possible to the output that would be produced by the function we are trying to model (recall in this case that the functions of interest here are cognitive abilities themselves).

Perhaps the simplest computational model of a biological neuron (and certainly the most famous within the field of machine learning) is the perceptron. The perceptron was first proposed as a general purpose computing unit (i.e. one that was able to perform a range of computational tasks) by Frank Rosenblatt in 1957 [Rosenblatt, 1958]. This model takes the component parts of a biological neuron that were just discussed: weighted inputs, a method to combine these weighted inputs, and a threshold function, and formalizes them within a computational frame work. A visual overview of a perceptron can be seen in Section 1.2.2. Here a number of weighted input values are summed together and then used as an input into a step function that evaluates to 1 (indicating that the neuron will "fire" by outputting a value of 1) if the weighted sum of the input exceeds 0, and 0 (indicating that neuron will not fire) otherwise. The computational model of the perceptron is as follows:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b)\right) \tag{1.2}$$

$$f(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1.3}$$

where $\sum_{i=1}^{n} w_i x_i$ is simply the weighted sum of the n inputs x, $f(x)$ is the step function that formalises the threshold at which the neuron fires and b is a parameter that determines the value of the threshold (which is known as the neuron's bias). Creating a network of these neurons is as simple as connecting a number of these

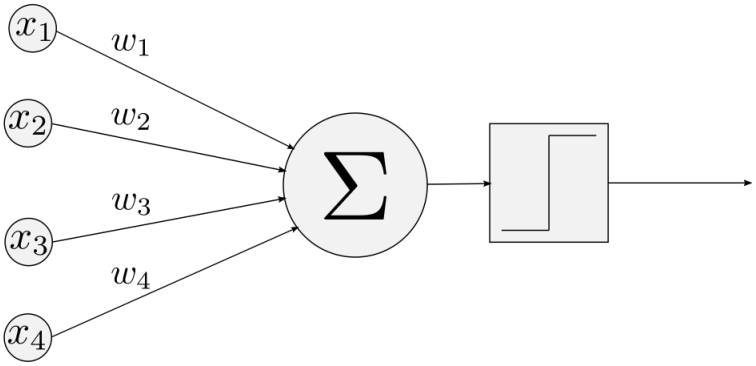


Figure 1.2: Model of a perceptron. Inputs x_1 to x_4 are multiplied by "synaptic weights" w_1 to w_4. Then summed in the "cell body" before being fed through the threshold step function.

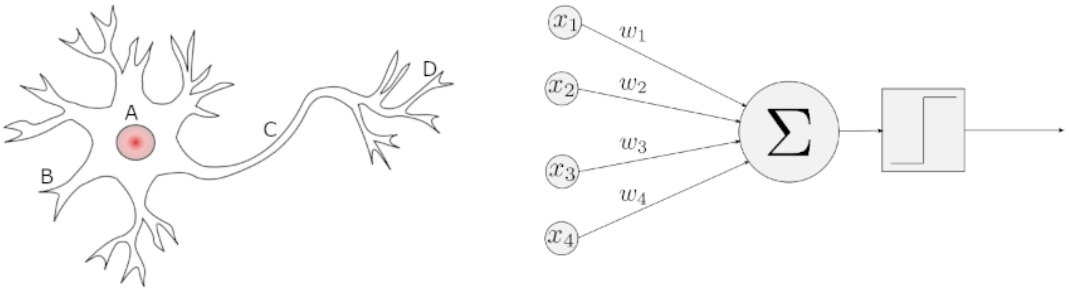


Figure 1.3: Visual comparison of a biological neuron (left) with the cell body (A), dendrites (B), axon (C), and synaptic terminals (D) shown (adapted from https://en.wikipedia.org/wiki/Neuron) and a perceptron (right).

perceptrons together in such a way that the inputs of the neurons are the outputs of set of other neurons. It is still not clear, however, how exactly this network of neurons is supposed to resemble a mathematical function. Firstly, the network itself needs to reflect the logic of mathematical functions, in that it needs to have some way of receiving input and providing output. Secondly, we need some way to allow the network to learn to approximate a given function from data i.e. it's very unlikely that a neural network will perfectly resemble a given function if we just set its weights and biases at random or even if we make educated guesses as to what these weights and biases should be. The solution to this first problem is straightforward. To model an function input we simply select one or neurons in the network that will take externally determined values as their input instead of outputs from other neurons. These inputs could be values representing the starting positions of two index fingers if we wanted the network to learn to approximate the Haken, Kelson, Bunz model or pixel values if we wanted the network to learn to recognise human faces. Similarly, function outputs can be represented by a number of neurons that do not connect to any other neurons in the network. We simply read the outputs of these neurons and take those outputs to be the outputs of our function. For example, if we were attempting to model a simplified version of the Haken, Kelso, Bunz equation where the function output tells us simply whether the input periodicity of the fingers would result in a in-phase or out-of-phase movement, we can set a single percetpron as the output neuron and have it such that an output of 0 means the fingers are out-of-phase and an output of 1 means that they are in-phase. Similarly, we can have an output of 1 represent that a face is present in an input image or 0 otherwise. Allowing the network to learn is more complicated, and I consider this issue in more depth in the methods section. For now, it is sufficient to point out that it is the parameters of our neural network (the weights and the biases) that will ultimately determine the output of each neuron, and therefore the values of the output neuron. In this way, changing the values of these parameters in the right way will ultimately lead the output of our network to reflect the output of the function that we are trying to model. In roughly this way perceptrons networks have been constructed to perform a number or tasks such as hand digit recognition as in [Kussul et al., 2001].

Of course, just as spiking neurons are not the only kinds of neurons present in the brain, perceptrons are not the only models of neurons that can be used to solve computational problems. Indeed, the final study of the thesis details how a network of computationally modelled continuous attractor neurons, in conjunction with a network of spiking neurons, can solve a range of problems in cognitive planning. However, since a majority of the methods used in the thesis use versions of perceptrons to solve their given tasks, understanding how function approximation

can be achieved by networks of neurons modelled using perceptrons is sufficient for a general introduction.

We might well ask at this point exactly how well these networks of neurons are able to model such complicated functions, especially given that all we seem to be doing is making changes to a limited set of parameters that relate to very basic computational units. Given how ostensibly simple these neural networks are, it might be surprising to discover that neural networks are able to approximate *any* function (with a few caveats). This revelation is the result of something called the Universal Approximation Theorem. This theorem states that given a sufficient number of neurons and a sufficient amount of training data, a network of artificial neurons (such as perceptrons) with a specified activation function (we used a step function previously but, suffice to say, may other choices are possible) can successfully approximate any continuous and bounded mathematical function. This result has been verified under a number of conditions, such as when a network is arbitrarily wide (when all the neurons are arranged into a single layer) [Hornik et al., 1989], when a network is arbitrarily deep (when the number of neurons in a given layer is relatively small but there are a very large number of layers) [Lu et al., 2017], when the threshold function is a continuous sigmoid [Cybenko, 1989], and when the neuron model is more complex [Zhou, 2020]. Of course, using this result to affirm that neural networks are suitable for approximating any cognitive process framed as a mathematical function assumes that these functions obey the conditions of the theorem i.e. that they are continuous and bounded. It is, however, extremely difficult to determine when a particular cognitive capability that we are trying to model is non-continuous and when a sufficiently well constructed neural network with enough training data would not be able to do a good enough job at modelling this discontinuity. In practice, the Universal Approximation Theorem serves as well defined and rigorous justification for using neural networks to try to emulate cognitive functions.

We now understand loosely how biological neurons propagate information and how networks of these neurons can break down complex tasks down into smaller and smaller subtasks. We also understand how these neurons, and networks of them, can be modelled computationally and roughly how these computational models can be used to approximate cognitive tasks when they are framed as mathematical functions. We also know, from the Universal Approximation Theorem, that neural networks are likely very well suited to the task of learning such functions. What the Universal Approximation Theorem points out, however, is that the architecture of the network, how the neurons are connected and arranged, will play a significant role in the how successful those networks are at approximating such functions. This leads us to the field of Deep Learning, which is by far the

most successful neural network method that has been applied to the problem of modelling various cognitive functions from an engineering perspective.

1.3 Deep Learning

1.3.1 Architecture

Deep learning is a form of machine learning which can be defined as the general process of designing and utilising algorithms which are able to extract patterns from data. Importantly, these algorithms should be able to extract these patterns automatically. That is, they should learn these patterns by way of a specific learning algorithm rather than because a significant amount of domain knowledge has been introduced to them from the start [Deisenroth et al., 2020]. Lastly, good machine learning algorithms should be domain general, meaning that one algorithm can applied to lots of different problems [Bengio, 2009]. Even from this very quick description, it should be clear that neural networks, as I have defined and explained them so far, can be implemented as machine learning methods: The patterns that they learn from data amount to the functions that they are aiming to approximate. They do this pattern learning automatically by way of an algorithm that is able to update their weights and bias as they are exposed to training data.

In the previous section, I briefly mentioned that the function of a neural network can be determined by its structure, often referred to as its *architecture*. Rather straightforwardly, the term "deep learning" comes from the basic kind of neural network architecture that is used in this branch of machine learning. A neural network is described as "deep" when it is constructed from many layers of neurons that are stacked on top of one another. This structure does not naturally follow from from any particular feature of artificial neurons. Every neuron could instead be connected to every other neuron in a kind of single interconnected blob (see Section 1.3.1 for a visual example of how these kinds of networks would differ visually). However, it turns out that stacking neurons in this way biases the network to form pattern representations in a particularly useful way. Recall from the previous discussion of facial recognition that different patches of the neocortex dedicated to visual processing were able to represent entities in the visual field at different levels of detail: one patch would represent the presence of lines, another the presence of collections of lines, all the way up to whole facial features and eventually faces. Organising deep neural networks into layers of neurons allows the network to form representations of their input space in this same hierarchical manner. As an example, the layers of AlexNet, a famous deep learning architecture trained on millions of input images for the task of classification, have been shown to

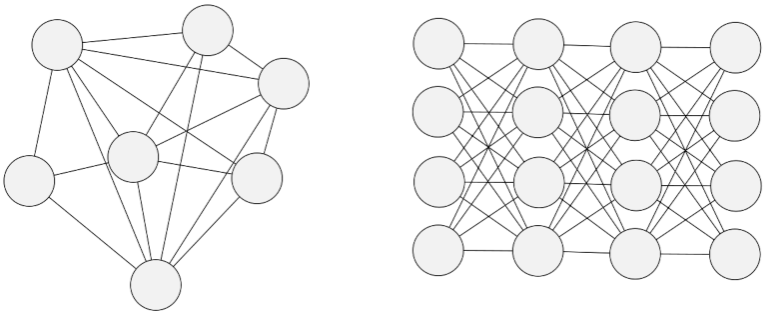


Figure 1.4: Visual comparison of two possible neural network architectures. A fully connected network with no inherent structure (left). A fully connected neural network with neurons organised into layers (right).

hierarchically represent different features of various objects. At the lowest levels, the neurons of AlexNet represent edges and very small patches of colour, while at the highest level they represent whole faces, animals, and vehicles [Krizhevsky et al., 2012][Yu et al., 2016]. This ability to learn task based representations in a hierarchy has a number of advantages. First, it naturally applies itself to the problem of learning about entities in the real world which are typically constructed (or, at least, can be interpreted as being so) in a similar hierarchical manner: faces are made of eyes and noses and mouths, which are made of pupils, and nostrils, and teeth, which are constructed from different sorts of polygons which are constructed of a number of lines and edges and so on [Bengio, 2009]. What is more, this hierarchical structural learning means that deep neural networks are able to generalize to a very large number of tasks (i.e. they have exceptionally high domain generality). This is because many such tasks can likewise be understood and represented hierarchically. As we have already pointed out, this is particularly appealing from the perspective of wanting to solve problems in computational cognition since a large amount of research from the human neurosciences suggests that the brain likewise decomposes complex problems into task-based hierarchies - not just in the visual domain but so too in areas such as motor control [Merel et al., 2019], learning [Yokoi and Diedrichsen, 2019], and language comprehension [Caucheteux et al., 2021].

1.3.2 Training

How deep learning architectures are trained is explained in detail in the next chapter. For now it is important to understand that successfully training neural networks poses a number of challenges. Most importantly, large networks that contain many thousands of neurons and hundreds of layers contain potentially billions

of trainable parameters. GPT-3 [Brown et al., 2020] a currently state-of-the-art natural language model, contains approximately 175 billion trainable parameters. Even much smaller networks, such as AlexNet which contains a meager 61 million parameters and over 600 million connections, provide a considerable challenge to optimisation algorithms. These methods require a principled way to determine the performance of a deep neural network during a round of training and a subsequent means by which to change every single parameter value by some amount such that the performance of the network will improve on the next training iteration. Fortunately, with the introduction of advanced graphics processing units that are optimised to perform the kinds of mathematical operations that are required to train large deep networks, and developments in the performance of parallel processing, training deep neural networks, even with billions of parameters, is no longer an issue just so long as you have the computing power available. What remains a large factor in the successful training of deep neural networks is the training data.

The larger the neural network, and the more complex the problem, the more training data is required to successfully train that network. For example, imageNet [Deng et al., 2009], a popular image dataset, contains over 14 million images. Similarly, NTURGB+D, a popular dataset for video based action recognition [Shahroudy et al., 2016] contains 114,480 video samples. EpicKitchens [Damen et al., 2020], a first-person perspective video dataset of people performing various tasks in their home kitchens, contains 20 million analysable frames of video, 90,000 labelled action segments, and roughly 100 hours of video recordings. Each of these datasets are regularly used to obtain state-of-the-art performance of a number of cognitive tasks including image classification [Krizhevsky et al., 2012], segmentation [Iglovikov and Shvets, 2018], action recognition [Shahroudy et al., 2016], action prediction [Wang et al., 2019], text based description of images [Venugopalan et al., 2017], and action affordance prediction [Nagarajan et al., 2020] among others. This poses a demanding challenge for deep learning researchers, especially those who aim to model cognitive problems that have had little attention paid to them by the machine learning community at large. For such problems, the only answer is to design and collect a novel dataset that is large enough to produce good results from a deep learning model or to derive research questions from publicly available dataset.

This issue brings us to the next section, which addresses the question that motivates the work on computational cognition and neural networks contained in this thesis. Namely, can we bring deep learning, and neural network research more generally, to bear on problems in the in human-robot interaction (HRI), a field in which modelling human cognition computationally is extremely desirable?

1.4 Human–Robot Interaction (HRI)

1.4.1 What

HRI is a vast research field that includes research stemming from a large number of scientific disciplines: from psychology and cognitive science to engineering and artificial intelligence. While there is no strict definition of what human–robot interaction is specifically, Sheridan provides a useful taxonomy that can help us to better locate the sphere of interest of the remaining chapters of the thesis. Sheridan splits HRI into four sub-fields, which are quoted here directly from [Sheridan, 2016]:

1. Human supervisory control of robots in performance of routine tasks.
2. Remote control of space, airborne, terrestrial, and undersea vehicles for non-routine tasks in hazardous or inaccessible environments.
3. Automated vehicles in which a human is a passenger, including automated highway and rail vehicles and commercial aircraft.
4. Human–robot social interaction, including robot devices to provide entertainment, teaching, comfort, and assistance for children and elderly, autistic, and handicapped persons.

It is fairly self evident that the computational modelling of human cognitive abilities would lend itself most naturally to the fourth of these categories. Robots that are designed to operate within such social interaction settings are most commonly referred to as social robots. Social robots have been variously defined but tend to be understood as computational agents that are designed to engage in socially oriented interactions with humans (and perhaps other social robots) [Dautenhahn, 2007] [Hegel et al., 2009] [Lee et al., 2006][Henschel et al., 2021].

One of the principle goals of the social robotics subfield of HRI is to develop robots that are able to socially engage with humans just as humans interact with other humans. A natural starting point for such research is to investigate the mechanisms by which humans are able to successfully interact with one another in social situations, and attempt to emulate those capabilities in a robotic platform. A significant part of what makes human social interactions as seamless and flexible as they are is the cognitive processes that allow us to interpret, predict, and respond to an assortment of behavioural and other cues offered by a social interaction partner. Thus, implementing functionally accurate computational models of these cognitive abilities is, uncontroversially, a good step towards the goal of creating truly effective robots that operate in this sphere.

1.4.2 Why

But why apply neural network methods to human–robot interaction at all? Firstly, the impact and presence of social robots has, and will likely continue to, increase dramatically in the near future (c.f., [Henschel et al., 2021]). Already, robots and artificial agents that are expected to be able to flexibly respond and interact with human social partners are being deployed throughout peoples' homes and on their persons in the form of voice agents like Amazon's Alexa, Microsoft's Cortana, or Google's Siri. These voice agents tend to respond to a limited set of knowledge-seeking requests involving fact checks and minor task demands such as setting a timer, playing a song, or sending a message. However, as users' perceptions of the capabilities of these agents increases, the kinds of requests or utterances that they will be expected to respond to will vary rapidly and likely increase in complexity. As an example, Amazon's Alexa research teams have recently published a study which seeks to better respond to strange user requests such as 'Alexa, do you want to build a snowman?' - a question that references a popular kids' film franchise [Shani et al., 2021]. Moreover, we are increasingly seeing the deployment of social robots into a very wide range of social situations that will likely make large demands of the cognitive capabilities of those machines. These situations include, but are not limited to, physical health care (especially in response to the COVID-19 pandemic [Aymerich-Franch and Ferrer, 2020]), mental health care [Broekens et al., 2009][Laban et al., 2021a], industry [Lenz et al., 2008], companions for the elderly [Broekens et al., 2009], hospitality [Henschel et al., 2021][Logan et al., 2019], and airport information assistance [Triebel et al., 2016]. As the scope of the roles expected of social robots increases, so to does the demand that these robots have cognitive systems that allow them to interact with humans in a safe, reliable, and effective manner. With all of these things in mind, it is clear that a great desire is emerging within the field of HRI, as in social robotics more specifically, for effective computational models of cognition that allow social robots and artificial agents to perform all of these tasks in the desired way.

Secondly, the research that sits at the intersection of HRI and neural networks remains somewhat in its infancy. Here it is important to make a distinction. Of course, neural network research that investigates cognitive-like capabilities has been around since at least the invention of the perceptron. However, research that looks into how neural networks, and specifically deep learning, can be applied to embodied agents that interact with humans in real world environments is relatively sparse (especially when compared to the volume of deep learning research dedicated to areas such as image recognition or machine translation). This factor, combined

with the increasing desire for socially intelligent agents outlined above, creates a very clear motivation for wanting to make progress in effectively engineering and emulating human cognitive abilities.

That neural network research focused on HRI is still in the early stages of its development can be explained by a number of factors. First and foremost, the collection of ecologically valid HRI data is very challenging. A vast majority of data that is collected in HRI experiments is in relatively low volume and, necessarily, collected in a tightly controlled laboratory environment [Henschel et al., 2020] [Cross and Ramsey, 2021]. Both of these factors make this data unsuitable for neural network learning. Firstly, because, as discussed in section Section 1.3.2, neural networks tend to require very large amounts of data to be able to effectively approximate the function in question. Secondly, if the goal of a computational model of cognition is that it should be implemented in a robotic platform that is to function in a particular real world environment, then it is essential that the data that is used to train that model is collected from experimental protocols that mimic that real world environment as closely as possible. This is so that the model has appropriate experience with the range of situations that could occur when it is deployed, but also so that, as experimenters, we can make meaningful conclusions about how that model will operate in that environment. This latter concern is especially important in situations where that particular model is to be implemented in environments that are especially sensitive to failure such as mental or physical care.

This might well cause someone to question why we would want to use neural network approaches in this field at all. First and foremost, developments in cost effective solutions for user robotics have lead to the development of many low cost social robots such as SoftBank's Pepper and Nao, Anki's Cosmo, or Consequential Robotics' MiRo. This has made the collection of ecologically valid HRI datasets significantly more straightforward as protocols are now able to use the same commercially-available robots that people will likely be using in their homes now and in the near future. What is more, the portability of these robots means that experimenters are able to deploy them in settings that allow far more flexibility of set up than with much larger, more expensive, robots. This opens up the possibility to develop experimental paradigms that more accurately reflect how the robots will be deployed in the real world. In terms of the capabilities of neural network approaches more specifically: modern network training techniques such as transfer learning allow neural networks to take knowledge that they have learned in one task and apply it to a new, similar, task. This means that powerful neural network architectures trained of very large datasets can be partially retrained on much smaller HRI specific datasets to be retooled to perform a specific task in the

field of HRI. Lastly, as discussed in section Section 1.3.1, a major draw to deep learning approaches is that their layered structure biases these networks towards learning tasks in a hierarchical fashion which mimics the way in which the brain plausibly also approaches such problems. This gives them an incredible amount of flexibility with respect to the kinds of cognitive tasks that they are able to perform, which in turn might suggest that an architecture that is successful at one particular cognitive task might quickly be applied to some other. This means that developments into the application of neural network approaches in HRI are likely to have significant impact all across the field.

Taken together, these considerations significantly motivate research into the application of computational models of cognition learned via neural networks and the application of these models to problems in socially oriented robots. The empirical work detailed in this thesis is an attempt to build on this burgeoning body of research and contribute to demonstrating the efficacy of neural networks in this domain.

This background raises a number of questions that can be seen to motivate the work of this thesis. Firstly, an obvious question to ask is simply whether or not neural network approaches, and specifically deep learning, are, in the first instance, appropriate tools to use in the modeling of cognition for social robotics. This may indeed not be the case since, as just discussed, HRI datasets large enough to train deep learning models effectively are difficult and time consuming to collect, not least because they necessarily involve the control and set up of complex robotic platforms that have to function in similarly complex and highly non-linear natural environments. Second, a related question is to what extent we can rely the representational power of relatively simple, small neural network architectures to get the desired performance in their respective learning tasks. Smaller networks have fewer trainable parameters and are thus possible to train on smaller amounts of data. Larger networks, however, are representationally more powerful and thus plausible better suited to the complex modeling problems that are required at the intersection of HRI and neural network-based learning. As a further consideration, using large pretrained neural networks and transfer learning can help to mediate the impact of small cardinality datasets. Thus, understanding better the trade off between simpler, smaller networks, and larger, more complex ones will likely play an important part in engineering appropriate neural network platforms for social robots. Lastly, there is a further foundational question regarding the use of deep learning approaches at all. As a tool, deep learning is rapidly becoming the principal means by which human-like cognitive capabilities are being bestowed upon computational platforms. This is not to say, however, that deep learning approaches are the only means by which this can be done. A final research ques-

tion which this thesis seeks to investigate is the extent to which neural network approaches that are distinct from deep learning can be used to emulate the same cognitive abilities. Research in the cognitive neurosciences has developed significantly since the modelling of the perceptron and it is likely that this field of work could point to other ways in which human cognitive abilities can be effectively modeled.

With this in mind, we can now concretely state the overall aims of the thesis:

1. To investigate the extent to which neural network models are appropriate methods by which social robots can be endowed with human-like cognitive abilities.

2. To develop novel datasets that are collected specifically with real world human-robot interaction scenarios and neural networks in mind.

3. To develop deep learning approaches, based on a solid proof-of-concept foundation, that are engineered with specific domain knowledge in mind.

4. To investigate alternative, novel neural network approaches to the problem of modeling human cognition.

1.5 The Current Research Approach

1.5.1 Research Principals

The three empirical works that are presented in chapters Chapter 3, Chapter 4, and Chapter 5 all follow a similar research approach which aims to fulfill the aims and address the concerns detailed in the previous section. This approach can be split into five components:

- **Strong appeal to contemporary research in the human brain sciences:** All three studies began with a thorough literature review on contemporary research in the area of interest, focusing on the most up-to-date and well established work in psychology and neuroscience. This helped to identify opportunities for cognitive modelling that had yet to be significantly addressed and to ensure that these models had the possibility of having a large impact outside of a purely academic context.

- **High quality, well labelled data:** In the previous section I outlined a number of demands on data collection that are required by neural network approaches. In order to manage this demand, all experimental procedures that were used to collect the data that the models were trained on followed

a strict protocol. First, we ensured that each experiment was designed to specifically generate data that were relevant to the cognitive ability that we were attempting to model as opposed to ones that have been derived from datasets that don't specifically address those concerns. While this may seem obvious, it ensures that the trained models are indeed emulations of specific cognitive abilities and are more likely to generalise well into real-world scenarios. Second, we ensured that the data used to train the models were of appropriate quality. This involved checking, cropping, cleaning (i.e. filtering), and labelling every data point by hand to ensure that all of the training data provided the best opportunities for model learning. Third, for the studies detailed in chapters Chapter 3 and Chapter 4, each data collection experiment was designed such that the participants labelled the audio recordings that were used to train the models. Since participants were labelling their own interactions with respect to their subjective experiences of them (their perceived stress and self-disclosure) all labels were, by definition, accurately assigned - since it is very unlikely that someone would be wrong about how they perceived an interaction. This avoids problems that can occur in very large machine learning data sets where a number of training or testing samples are incorrectly labelled which can lead to poor model performance or, worse, results that are not reflective of the models true capabilities. Finally, we aimed, to the best of our abilities and resources, to collect as much data as possible in each case in order to avoid the aforementioned issue that many HRI datasets contain sample sizes inappropriate for neural network approaches.

- **Ecologically valid experimental design:** To address the problem of ecological validity, we ensured that all our lab-based HRI experiments involved the use of embodied social robots. This was to try to emulate as closely as possible the feeling of the interaction that people are likely to have with social-robots in the real world. Next, both lab-based studies (Chapter 3 and Chapter 4 use a "Wizard of Oz" whereby an experimenter controls the responses of the robots out of view of the participants. While this approach will arguably limit the ecological validity of the interaction (see [Henschel et al., 2020] [Cross and Ramsey, 2021]), we wanted to ensure that the interactions themselves were free-flowing and natural so as to create a sense of how advanced social robots might (someday soon) operate in real world settings. To limit the pitfalls of this approach, we made sure that the responses were limited to a set of utterances in such a way that mimicked the capabilities of the robot as it would operate autonomously.

- **Extensive Model Testing:** For each of the models that are discussed across all three empirical studies, we prioritised a robust testing procedure that ensured we had a good understanding of the best approach to each problem. Our priority was not to produce the best possible model in each case but to understand which factors contributed most significantly to good model performance. As such, in Chapter 3 we begin by testing the hypothesis that neural network approaches are effective *at all* at modelling the problem at hand and choose to experiment with a number of different basic architectures on the problem before evolving to more complex approaches to the problem. In Chapter 4 we perform extensive ablation experiments on our novel architecture to test out every possible combination of network parameters that we were interested in. Finally, in Chapter 5 we design a number of complex tasks to test the performance on our model to ensure that the results that we outline are not simply confined to basic cases of the problem. This ensure that our model can handle a number of challenging edge case tasks, thus increasing the likelihood that it would be effective in a large range of scenarios if deployed in a real robot.

- **Deductive modesty:** In [Ramsey, 2021] Ramsey points out a need for greater modesty in the cognitive neuroscience literature due to, among other factors, "an incentive structure that requires newsworthy results". I believe that such a requirement is likewise relevant to the machine learning research field and as such make efforts to draw conclusions from our studies which are proportionate to the results that we provide. In all cases we ensure that drawbacks to our methods and results are clearly detailed and, in the case of Chapter 3 and Chapter 4, detail specific concerns that need to be addressed before any such models can be applied in actual social robots.

1.5.2 Overview of Studies Conducted

Is Deep Learning a Valid Approach for Inferring Subjective Self-Perceptions in Human-Robot Interactions?

A significant challenge in creating socially intelligent artificial agents is constructing cognitive models that imbue those agents with the abilities to pick up on a person's subjective perceptions of themselves. These include, but are not limited to, how stressed a person considers themselves to be (subjective psychological stress), and how much personal or sensitive information that person considers themselves to be sharing during an interaction (subjective self-disclosure). These measures differ from those already studied to some degree in the HRI and machine learning

literatures [Soleymani et al., 2019] [Bara et al., 2020] in that what is measured is how much stress or self disclosure that person perceives themselves to be under or sharing rather than how these factors are perceived by an interaction partner. Since the aims of this thesis concern the use of neural network approaches to such problems, in this study we aimed to test the hypothesis that neural network approaches were well-suited to modelling this challenging task. To address this aim, we collected a dataset of interactions between participants and three different kinds of agents (a human, an embodied humanoid robot, and a voice agent) and asked participants to rate their interactions with respect to their subjective levels of psychological stress and self-disclosure. We then constructed a machine learning problem wherein a model was to learn to predict the level of psychological stress and self-disclosure from an audio snippet in a given interaction. We trained six deep learning models on these data using a number of different input features and framed the problem as both a regression and classification version of this task. We then compared the results to chance baselines. In all cases we found that the models performed well above chance. Despite these promising results, we argue that well above chance accuracy is not sufficient for the goal of deploying these models in real social robots. This prompts the conclusion that further research was warranted in this area in order to improve upon these results.

Multimodal Deep Learning of Subjective Self-Disclosure in Human-Robot Interactions

Following from the previous study we aimed to develop a more effective deep learning model on the problem of subjective self-disclosure scoring. This problem was chosen over subjective psychological stress due its perceived impact and its relative novelty within the field of HRI. This study builds upon our previous work in a number of ways. First, we collected a much larger subjective self-disclosure dataset from interactions between participants and a SoftBank Pepper robot that took place over Zoom. Visual and audio dimensions of these interactions were recorded (as opposed to just audio in the previous study) so that we might leverage facial features related to subjective self-disclosure. We then performed a larger number of transformations on the input data to gather a range of representations including: facial action units, gaze, facial feature embeddings from an large pretrained ResNet, mel-filter cepstral coefficients, and audio embeddings from a large pretrained transformer model. We then performed base line experiments by training support vector machine models on each feature set individually. This allowed us to more accurately determine how well our neural network models performed. We then construct a novel multi-modal attention network leveraging recent results from the emotion recognition literature which, in turn, uses two pretrained neural

network backbones to take advantage of transfer learning. We also detail a novel loss function that attempts to strike a balance between the regression and classification versions of the problem that were outlined in the previous chapter. We perform an extensive ablation experiment on our attention network which explores the effects of the input data representations, loss function, and experimental framing on our model results. We found that all versions of the model significantly outperform the baseline models. Further, we find that the model trained on a version of the input data formed by principal components analysis combined with our novel loss function performed the best on this challenging task.

A Hybrid Biological Neural Network Model for Solving Problems in Cognitive Planning

In this study we argue that a large number of important cognitive planning behaviours that humans use to operate in both individual and social situations can be formulated as graph traversal problems on cognitive maps. This involves representing such problems as a graph, i.e. a set of nodes that represent different states of the planning problem and edges between those nodes as possible transitions between these states. This graph-based representation is commonly referred to as a "cognitive map" and we argue that solving a planning problem on these maps amounts to moving a node of activation from a starting node (representing an as-is state of the system) to and ending node (representing the required end state of the system). While there is a large amount of work in the neurosciences on cognitive maps [Tolman, 1948] and their relation to graph problems from mathematics [George et al., 2021], little work has been done to attempt to show how the brain might make use of neural network structures and dynamics to allow it solve these kinds of planning problems. In this chapter we present a novel hybrid neural network based on computational models of biological neurons from the neocortex (spiking neurons), the entorhinal cortex, and the hippocampal formation (continuous attractor neurons) which solves these cognitive planning problems using short path solutions. We test this model on a number of complex cognitive map formations and show that in each case the model is able to traverse the map successfully.

1.6 Summary

In this introductory chapter, I have aimed to provide a general practical and theoretical introduction to the empirical work that is contained in this thesis. I have aimed to provided an introductory discussion of the field of research from

which the main topics of this thesis are taken, and attempted to illustrate clearly the niche within that field that the work falls into as well as the main questions that it seeks to answer.

I began by considering what researchers might mean when they talk about cognition and defined it as: the brain's manipulation, handling, representation, and storage of knowledge (defined in some appropriate way) accumulated from within and without the body. I then discussed, by way of an example, what kinds of behaviours might be thought to come about as the result of, or themselves be, a cognitive process. These processes where then looked upon through the lens of dual-systems theory so that we might understand how different kinds of cognitive processes can be categorised. In the next section, I addressed the notion of computation and what it means for a particular cognitive process to be computationally modelled. An argument was given in support of the use of computational models with respect to how they allow us to understand cognitive processes more precisely and how they provide opportunities for the generation of testable research hypotheses. I then claimed that computational cognitive research can be split up into two streams which differ with respect to their goals. On the one hand, researchers might be interested in discovering truths about human cognition by way of computational models, in which case we defined those studies as being *empirical* in nature. On the other hand, researchers might want to emulate those cognitive functions to perform some task to a given standard, in which case we define those problems as being related to *engineering*. I then stated that the majority of the work done in this thesis was situation within this second category. Next, we reframed the problem of modelling cognition to one of function approximation so that we might understand how neural networks are able to go about modelling different human cognitive capabilities. Biological neurons were then reviewed as a necessary prerequisite to understanding how computational neural networks are constructed. I then explained the perceptron and how it can be used to construct neural networks that are able to model a mathematical function (and therefore an aspect of cognition). Next, deep learning was discussed. I explained how it was a form of machine learning and how the name *deep* learning comes from how a neural networks neurons can be stacked into layers. I then showed that this hierarchical layered structure biased the networks to model tasks likewise hierarchically and how this can be argued to mirror the way in which cognitive tasks are performed by the brain. We then saw that training deep network was challenging because of demands made by the very large number of parameters that deep neural networks tend to have. From these problems we identified a particular challenge facing researchers who wanted to develop neural network models based on not so well recognised cognitive functions i.e. that they would need to either find a way to

gear an existing dataset towards that particular topic or to collect one themselves. This lead us to the area that concerns a majority of the thesis: HRI and social robotics. In section Section 1.4 we looked at a definition of social robotics in HRI and claimed that this was the area in which effectively functioning computational models of cognition would be most useful. I then provided a number of arguments for why we might want to develop neural network models of computational cognition for problems in HRI despite all the challenges which were previously outlined. First, I explained how the rapidly expanding use and need for social robots in a number of contexts has created a need for more research to be conducted in this area. Second, I claimed that experimental resources, such a research robots and data collection equipment, were developing in such a way that made the collection of large ecologically valid HRI datasets more possible. Finally, that there was still a large amount of work to be done in this area largely due to the challenge posed by the need for large ecologically valid datasets that employed embodied artificial agents rather than virtual agents or on-line chat bots. In the final section of the chapter I reviewed the four research principals that were used to guide each of the three empirical projects of the thesis. Namely, strong appeal to contemporary research in human brain sciences, use of high-quality, well labelled data, utilisation of ecologically valid experimental design, extensive model testing, and deductive modesty. Finally, the three empirical studies contained in the thesis, including their background, aims, methods, and results were outlined.

Throughout the following chapters I have used the above background, considerations, constraints, and motivations to conduct three empirical studies that all look to develop models of particular human cognitive capabilities that are aimed to be deployed in socially intelligent robots. My hope is that by way of the methods and results of those studies, a good case can be made in favour of continuing to conduct neural network research in the field of human–robot interaction despite the considerable challenges involved.

Chapter 2

Methods: Modeling Time with Neural Networks

2.1 Introduction

The problem of modeling time in neural networks arises in areas like natural language processing, machine translation, text generation, and gesture recognition among others where being able to make sense of a given data point depends on being able to remember information about the data points that came before it. Consider the example of being able to predict the next word in a sentence. The piece of writing might start off by saying: "When I was young I decided that it was my dream to become a chef". After a few paragraphs describing the rest of the protagonists life we might return to this idea: "It was at that point, after quitting my desk job in the city, that I decided to realise my childhood dream of...". In order to predict the next word in the sentence it is necessary that we recall from earlier on that the protagonist's childhood dream was that she wanted to become a chef. As it turns out, modelling the kinds of decision making processes that require this kind of memory is not easy.

Many human cognitive abilities unfold over time, particularly in social situations, and require a representation of past, and a prediction of future, states of the world. Consider coordinating with a social partner while performing a physical task like cooking a meal. First, at any point, I need to understand what my partner is doing: is she reaching for the spatula, or the knife? Does she intended to use the knife to chop some vegetables or does she want to wash it? Predicting these things will allow me to plan and organise my own actions in such a way that our task is completed smoothly i.e. without getting in each other's way or attempting to perform the same task as one another.

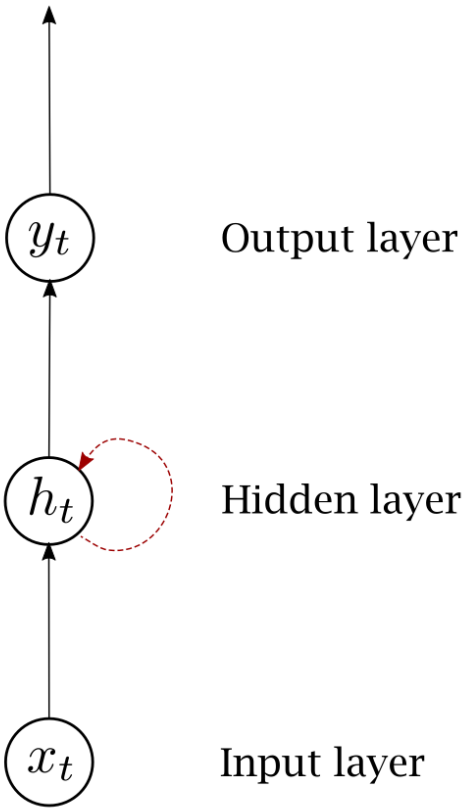
These kinds of predictions, guessing someone's intentions, predicting their future actions and so on, would be very hard if we were only able to represent what that person was doing at a single moment in time (imagine the difficulty of trying to predict what a person was about to do from a photograph as opposed to from a section of video). Good predictions of these sorts will, rather, come about as a result of how I see my social partner's actions, movements, and verbal utterances evolve over different time scales. Thus, if I want a model to be able to represent these kinds of social cognitive predictive abilities, that model will need a good way to represent time, and form predictions based on what has happened in the recent past and present. Recurrent neural networks are the natural choice for molding these kinds of abilities, and are likewise the architectures of choice in the following empirical chapters. As such, I choose to focus solely on these kinds of architectures in what follows.

Recurrent neural networks share a close resemblance with feedforward neural networks (i.e. the kinds of fully connected, layered networks we saw in 1). The principle difference being that recurrent networks have what's called a recurrent edge. This is an edge between two neurons (or the same neuron) that feeds information back through the network - in the opposite direction to the networks forward edges - essentially storing that neuron's output as a sort of memory. The most basic example of this can be seen in Figure 2.1a. A common means of understanding how this recurrent edge works is by "unrolling" this simple network so that it is more clear how the recurrent edge passes information from one time step to the next (as in Figure 2.1b).

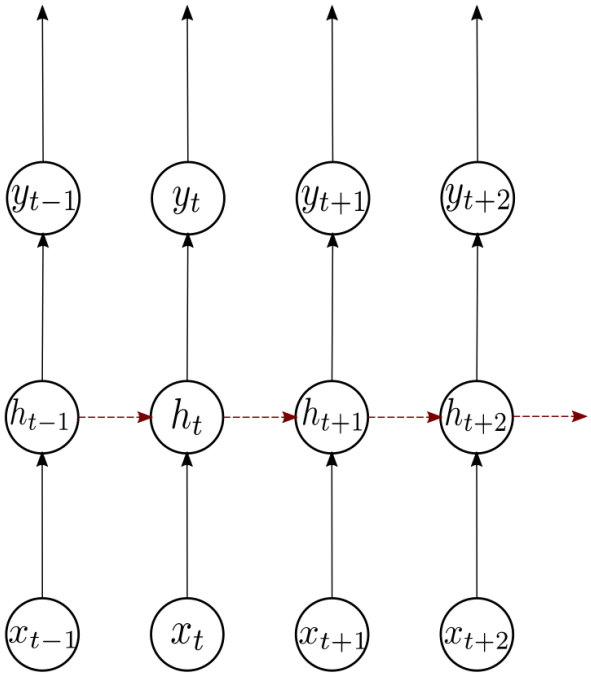
In this chapter the aim is to introduce the fundamental concepts and mathematical ideas behind the operation of neural networks that attempt to model time with a particular focus on the methods that are utilised throughout the thesis. Section 1 introduces the most popular kinds of activation functions that are used in contemporary neural network architectures that model time. Section 2 reviews the early work on recurrent networks and provides mathematical descriptions of their operation and architecture. Finally, Section 3 discusses how recurrent networks are trained, how this training differs from training feedforward networks, and some of the main issues that confront researchers wanting to utilize these kinds of neural networks on their data.

2.2 Activation Functions

Before describing recurrent neural networks it is worth covering activation functions as they are a key component in understanding how these networks - and neural networks in general - compute their output. The activation function deter-



(a) A simple recurrent network with the recurrent edge show in red.



(b) An "unrolled" simple recurrent network with the recurrent edge passing information across time steps.

mines the magnitude of the output of a given neuron and the kind of activation function chosen for a layer's neurons will effect the output of those neurons and the behaviour of the network more generally. This is the same way in which neuron activations are determined in non-recurrent networks. As many recurrent networks use a modified version of back propagation for learning weights and biases, it is required that the activation functions are both continuous and differentiable at all points (the reason for this is spelled out in more detail in section Section 2.4.1). The most commonly used activation functions in the literature are the sigmoid, hyperbolic tangent (tanh), and the rectified linear unit (ReLU).

2.2.1 Sigmoid Function

The sigmoid function takes a real valued input and outputs a value in the range [0,1] where the output is given by:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

This function is among the most widely used in the literature [Hochreiter and Schmidhuber, 1997b] [Jaeger, 2001][Gregor et al., 2015] but has its drawbacks. Most pressingly is the fact that the derivative of the sigmoid function at all points is close to zero which leads to problems when wanting to learn long-term dependencies between data points (again, more detail on this problem is given in section 4).

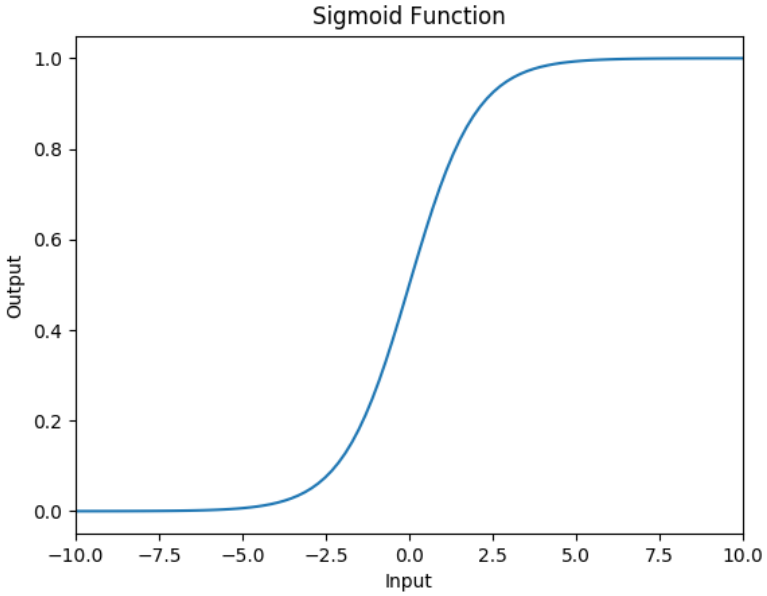


Figure 2.2: Illustration of the sigmoid activation function for input values between -10 and 10.

2.2.2 Hyperbolic Tangent Function (TanH)

The tanh activation function is a modified sigmoid function that changes the range of its output from the range [0,1] to the range [-1,1] the value of which of given by:

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{2.2}$$

It's relation to the sigmoid function can be seen more clearly in the equation:

$$\sigma(x) = \frac{\tanh(x/2) + 1}{2} \tag{2.3}$$

The principle benefit of using the tanh function is that it is able to map negative inputs to negative outputs, where the sigmoid function would be unable to map negative input to any output < 1 and strongly negative inputs converge on 0.

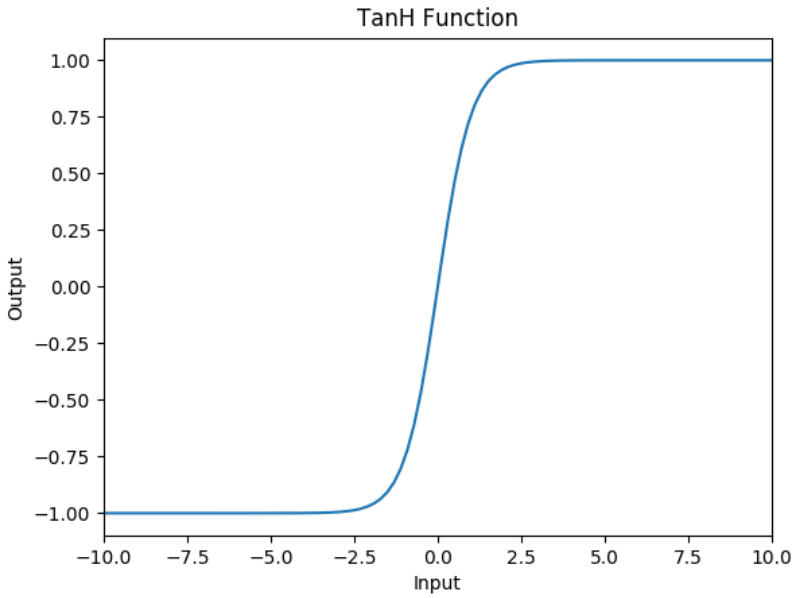


Figure 2.3: Illustration of the tanh activation function for input values between -10 and 10.

2.2.3 Rectified Linear Unit Function (ReLU)

Finally, the ReLU activation function is given by:

$$R(x) = \max(x, 0) \tag{2.4}$$

What chiefly distinguishes the ReLU from sigmoid and tanh functions is that it is open ended with respect to its mapping of positive inputs to positive outputs. Whilst sigmoid and tanh functions have an upper-bound of 1 for positive inputs, the range of ReLU is $[0, \infty)$. The second notable feature is that inputs below 0 all

map to 0 which means that inputs < 0 will be equated and indistinguishable to later layers in the network. This is a problem if the data being modelled results in many neuron inputs below zero or if having negative values as the outputs to certain neurons makes sense. This can happen, for instance, if the network outputs are expected to model something like positions of body parts in 3D space where the origin of your coordinate system is the center hip joint. In this case knee, ankle, foot joint locations would naturally be represented as negative values. The main advantage of this activation function is that it leads to faster convergence when using stochastic gradient descent for learning [Krizhevsky et al., 2012][Dahl et al., 2013]. This particular feature has resulted in the use of this activation function in many state-of-the-art architectures [Bell et al., 2016][Jing et al., 2017].

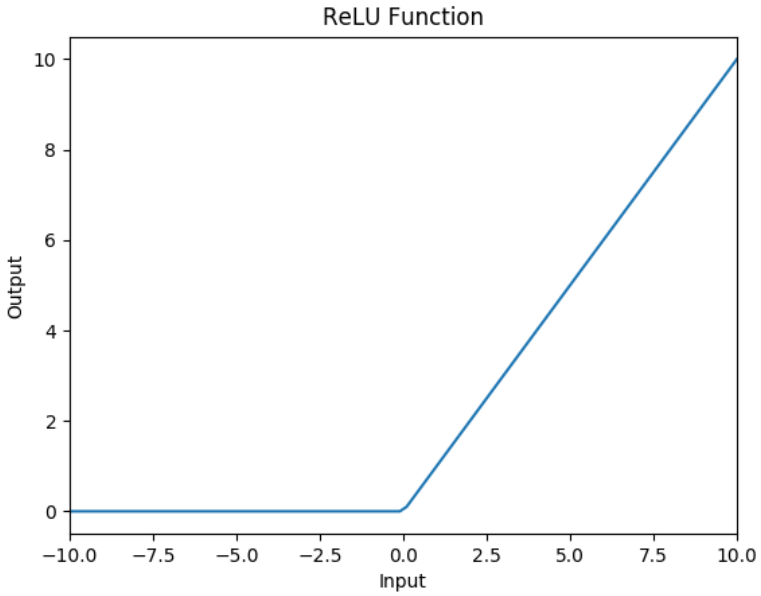


Figure 2.4: Illustration of the ReLU activation function for input values between -10 and 10.

2.3 Simple Recurrent Neural Networks

2.3.1 Hopfield Networks

One of the earliest recurrent architectures was introduced by Hopfield [Hopfield, 1982] and networks of this type are thus referred to as Hopfield networks. Hopfield networks were designed to function as generalized content addressable memory systems, meaning that they could retrieve an item stored in memory from some partial or corrupted piece of information provided as input. To elucidate this idea, imagine if you were shown a photograph of a friend, but that part of this photograph had been damaged such that some portion of the person's face was obscured. Despite not having all of the visual information you would none-the-less be able to identify the person in the photo as your friend. Similarly, a content addressable memory system would be able to take as input an image that had been obscured by some noise and return from memory the identity of the object in that image. That is, given some input $x_1, x_2, ..., x_n$ assigned to the n nodes in the network, Hopfield networks will return an output vector that most closely resembles that input from its memory. Most importantly, Hopfield networks were one of the first architectures to introduce recurrent connections. These connections meant that information could be passed back through the network rather that just fed in a forward direction as in feedforward neural networks (for an overview of these kinds of networks see [Svozil et al., 1997]). Weights between the neurons in Hopfield networks obey the following rules:

$$w_{ji} = w_{ij}, \forall_{i,j} \tag{2.5}$$

$$w_{ii} = 0, \forall_i \tag{2.6}$$

where w_{ji} denotes the weighted connection to the j^{th} neuron from the i^{th} neuron. Equation (5) picks out that the weights between the neurons are symmetric. This means that the weight of a connection to the j^{th} neuron from the i^{th} neuron is the same as the weight of the connection to the i^{th} neuron from the j^{th} neuron. Equation (6) denotes the fact that that no neuron has a connection directly to itself. The state s of the j^{th} neuron is updated by:

$$s_j = \begin{cases} +1 & \text{if } \sum w_{ji} s_i \geq b \\ -1 & \text{if } \sum w_{ji} s_i < b \end{cases} \tag{2.7}$$

where $\sum w_{ji} s_i$ is a weighted sum of the inputs to the j^{th} neuron and b denotes the bias or firing threshold of that neuron (with $b = 0$ given in [Hopfield, 1982]).

A noise-effected input is put into the network by setting the values of the neurons according to the values of the input vector. The network will then run until the the neurons have updated to match some set of values that is stored in memory at which point the values are read out to provide a recovered version of the input data. The general idea is that the system will be drawn towards some proximal steady state by beginning in that state's attractor basin by virtue of the input's proximity to that steady state (given that it is an altered version of the vector belonging to that steady state). In this case the steady state of the system is equal to the output vector corresponding to the recovered version of the input. Other than providing a basis for later work on recurrent networks, Hopfield networks also provided theoretical grounding for auto-encoder networks that are typically used for reducing the dimensionality of data [Hinton and Salakhutdinov, 2006].

2.3.2 Jordan Networks

Jordan networks [Jordan, 1997] introduced a recurrently connected layer of state-neurons [1] to the feed-forward architecture. These state neurons take as input the output from the previous time step. In this way state neurons are able to update the neurons in the network's hidden layer with information about the state of the network at the previous time step. Network states at time $t+1$ are thus able to be effected by past system states thus providing the network with a kind of memory. The vectors at time t for the state (s), hidden (h), and output (y) layers respectively are given by:

$$\mathbf{s}_t = \phi_s(W^r_{sy}\mathbf{y}_{(t-1)} + W^r_{ss}\mathbf{s}_{(t-1)} + \mathbf{b}_s) \tag{2.8}$$

$$\mathbf{h}_t = \phi_h(W_{hx}\mathbf{x}_t + W_{hs}\mathbf{s}_t + \mathbf{b}_h) \tag{2.9}$$

$$\mathbf{y}_t = \phi_y(W_{yh}\mathbf{h}_t + \mathbf{b}_y) \tag{2.10}$$

where W^r_{sy} is the weight matrix to the state layer from the output layer (superscript r denotes that the connection is recurrent with recurrent weights set at $w^r_{ji} = 1$), W_{hx} is the weight matrix to the hidden layer from the input layer, W_{yh} is the weight matrix to the output layer from the hidden layer, $\mathbf{x}_t$ is the input vector at time t, $\mathbf{y}_{(t-1)}$ is the output vector at time $t-1$ and $\mathbf{b}_s$, $\mathbf{b}_h$ and $\mathbf{b}_y$ are the bias vectors for the state layer, hidden layer, and the output layer respectively. Finally, ϕ denotes an activation function (in the simulations carried out in [Jordan, 1997], the activation function is the identity function: $\phi(x) = x$. The final important feature of Jordan networks is that the state neurons have interconnected recurrent edges meaning that the output of each state neuron is fed both into itself and into

[1] These are sometimes referred to as state units but I use the term neurons here for consistency.

the other neurons in the state layer. This means that at each time step the output at $t-1$ is able to be combined with a running accumulation of the state vectors from earlier time steps thus in principle being able to store network states from further back in time. This feature marks a principle difference between recurrent networks and Markov Models as the latter rely on the operational principle that a state at t depends only upon the previous state at $t-1$.

2.3.3 Elman Networks

Elman networks [Elman, 1990] are variations on Jordan networks. The principal difference being that the recurrent connections exist between the neurons in the hidden layer and the neurons in the state layer[2]. This can be seen more clearly by comparing the layer-vector equations of Jordan networks (above) to the equivalent equations in Elman networks where the vectors at time t for the state, hidden, and output layers are given by:

$$\mathbf{s}_t = \sigma_s(W^r_{sh}\mathbf{h}_{(t-1)} + \mathbf{b}_s) \tag{2.11}$$

$$\mathbf{h}_t = \sigma_h(W_{hx}\mathbf{x}_t + W_{hs}\mathbf{s}_{(t-1)} + \mathbf{b}_h) \tag{2.12}$$

$$\mathbf{y}_t = \sigma_y(W_{yh}\mathbf{h}_t + \mathbf{b}_y) \tag{2.13}$$

Notice that the computation for the state layer vector $\mathbf{s}_t$ contains the state vector for the hidden layer $\mathbf{h}_{(t-1)}$ at the previous time step and no recurrent connection to itself whereas the equivalent computation in a Jordan network comprised of the *output* vector $\mathbf{y}_{(t-1)}$ from the previous time step *as well as* a recurrent input from the state layer to itself. Another way of putting this is to say that for a Jordan network the state of the hidden layer at a given time is a function of the network *output* of the previous time step whereas for an Elman network the activations of the hidden layer is a function of the activation of the same hidden layer from the previous time step. This particular component of these simple recurrent networks is one of the key features of more state-of-the-art recurrent networks called Long Short-Term Memory networks [Hochreiter and Schmidhuber, 1997b]. The activation of the hidden layer at the previous time step is stored in a copy layer the outputs of which are then used to computed activations at subsequent time steps. As in Jordan networks the weights of the recurrent edges are at fixed values across all time steps.

Both Jordan and Elman networks are useful for any machine learning system that is designed to make predictions about currently unknown states of affairs

[2][Elman, 1990] calls this layer the context layer but again I will use the term state layer for notational convenience and as the function of these neurons is sufficiently similar.

based on past knowledge. For instance both networks could, in principal, be used to make weather forecasts where tomorrow's weather depended on the current weather, i.e. a representation in the state of the network at time t, and the weather from the day before, a state representation of the network at time $t-1$.

2.4 Training Recurrent Neural Networks

2.4.1 Back Propagation Through Time

In feedforward networks the model parameters (the weights and biases) are learned by back propagating an error gradient through the network. This works by first defining some loss function $\mathcal{L}(\hat{y}, y)$ which compares the output y of the network to some target $\hat{y}$. The rate of change of this loss as a function of changes in the weights and biases is used as means to determine how much the weights and biases should be changed at each stage of the training. Learning algorithms ensure that the weights are changed so as to minimize the loss function meaning that when the network converges the outputs are as close to the desired output targets as possible. For recurrent networks the problem of learning becomes more complicated because we need to ensure that errors are back propagated through time steps. Thus, to compute the necessary total loss for the network across time it is necessary to define a cost function $\mathcal{E}$ that sums over the differences between $\hat{y}$ and y for all time steps t:

$$\mathcal{E}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{t=1}^{T} \mathcal{E}_t(\hat{\mathbf{y}}_t, \mathbf{y}_t) \tag{2.14}$$

Furthermore, for recurrent networks we need to compute the derivatives of the cost function with respect to the recurrent weights, not just for the weights between the layers as in feedforward networks. Calculating these derivatives using the chain rule we get:

$$\frac{\partial \mathcal{E}}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathcal{E}_t}{\partial W} \tag{2.15}$$

$$\frac{\partial \mathcal{E}_t}{\partial W} = \sum_{1 \leq k \leq t} \left(\frac{\partial \mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} \frac{\partial \mathbf{x}_k}{\partial W} \right) \tag{2.16}$$

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \geq i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{(i-1)}} \tag{2.17}$$

For notational convenience in these equations I have collected the parameters of the network into the single term W. These differ from the traditional back

propagation through time equations as laid out in [Werbos, 1990] and are adapted from [Pascanu et al., 2012] as they demonstrate more succinctly the computations necessary for calculating the gradient through time. Similar to feedforward networks gradient computations can be used to tweak the network parameters until the cost function reaches a (ideally) global minimum.

2.4.2 Vanishing and Exploding Gradients

Training and optimizing even simple feedforward networks has previously been shown to be a NP-complete problem [Judd, 1987] [Blum and Rivest, 1992], meaning that a finding a solution to such a problem is very hard and the difficult of find such a solution grows rapidly with the complexity of the problem. Despite this even deep feedforward neural networks have been successfully trained using back propagation algorithms [Rumelhart et al., 1986b] [Rumelhart et al., 1986a]. However, the training of recurrent networks has proved to be more difficult. This is largely due to problems known as the vanishing and exploding gradient problems [Bengio et al., 1994][Pascanu et al., 2012]. These occur due in combination of the facts that recurrent neurons tend to have fixed edge weights (as in both Jordan and Elman networks) and that the goal of recurrent networks is to model long term dependencies in sequential data. To illustrate this problem consider the following two cases. In both cases we take a simplified recurrent network with a single hidden layer and single output layer with a recurrent edge connecting the hidden layer to itself. In the first case however we set the hidden layer's recurrent weight to some value > 1 and in the second case we set the layer's recurrent weight to some value < 1. In the first case at each time step the output of the hidden neuron is being fed back into itself and combined with the weight $w_{hh} > 1$ and because the weight is fixed this does not change as time continues. This means that at every time step the output of the hidden neuron along the recurrent edge is being multiplied by a value greater than one and then fed back into itself for computing the output of the hidden neurons at the next time step, and so to for every time step after that. It should be clear from this that as the run time of the network increases, or as the difference between the time t and some future time τ increases, the contribution of that input at time (t) will grow exponentially large. In the second case, where the recurrent weight $w_{hh} < 1$, the opposite will happen. Because the output of each hidden neuron along its recurrent edge at some time step t is being multiplied by a value < 1 the contribution will diminish exponentially as the difference between t and τ grows large. As a result of this the rate of change of the error ε of the hidden layer with respect to the input $\frac{\partial \varepsilon_t}{\partial x_t}$ (the gradient) will either grow exponentially (explode) or shrink exponentially (vanish) creating an

unstable learning rate. This means that, in the case of a vanishing gradient, the network will have difficulty learning long range dependencies between points in a sequence because the contribution of an input in the past will diminish to zero. As [Pascanu et al., 2012] show, this problem is inherent to networks that utilize the computation of partial derivatives over consecutive time steps to compute error in order to update the network. This is a problem as a majority of algorithms for recurrent networks utilize these kinds of gradient methods to optimize learning [Rumelhart et al., 1986b, Williams and Zipser, 1989, Werbos, 1990].

2.5 Modern Recurrent Neural Networks

2.5.1 Long Short-Term Memory Networks

On the back of this problem a large area of research within the field of recurrent networks is the design of learning algorithms that avoid the problems of vanishing and exploding gradients. With respect to the exploding gradient problem, [Pascanu et al., 2012] adapt an algorithm from [Mikolov, 2012] that clips each value of the gradient if it equals or exceeds a given threshold. This essentially scales down values to prevent the gradient from exploding. This does however add a further element of complication to the procedure of training the network, because the threshold becomes an additional hyper-parameter that needs to be determined and set by the experimenters. As it turns out however the training process is relatively insensitive to changes in this hyper-parameter and the algorithm works well even for small threshold values. [Williams and Zipser, 1995] proposes a version of back propagation through time known as truncated back propagation through time. The concept behind the algorithm is relatively simple. Recall that the vanishing and exploding gradient problems occur as a function of the accumulation of weighted sums of the outputs of the hidden neurons. It is possible therefore that by limiting the amount of time steps over which these weighted sums can accumulate we can prevent the gradient from either vanishing or exploding. The draw back with this method is that by excluding information from previous time steps you weaken the networks ability to accurately model relationships between data points that are far away from each other in time.

Another, and perhaps the most successful approach to avoid the problem of vanishing and exploding gradients, is the development of long short-term memory networks. The operation of these networks is relatively complicated compared to the networks that have been examined so far but since they are responsible for so many of the major results in modern sequence modelling it is worth exploring them in detail. Long short-term memory networks were first described by [Hochreiter

and Schmidhuber, 1997b]. These networks fundamentally change the architecture of simple recurrent networks by virtue of computational units known as cells. As with simple recurrent networks the state of a memory cell is a combination of the input to that cell at some time t and the output of a cell at time $t-1$ which in turn is influenced by a cell at $t-2$ and so on. The key principle to understand the flow of information in a memory cell is the cell state C. This is, in a sense, a stream of input that is added to and subtracted from before it becomes the output of the cell. When the cell state enters the memory cell it is a copy of the output of the cell at $t-1$ but without having gone through the activation function from that previous cell. This distinguishes it from that previous cell's output $\mathbf{y}_{(t-1)}$. The second difference between simple recurrent networks and long short-term memory networks is that, where simple recurrent networks usually have one activation function, long short-term memory networks use a combination of sigmoid and tanh functions within each cell to influence each cell's output. These additional functions act as gates that allow the information in the cell state to be influenced to different degrees by the input at that time step and the input from the cell at the previous time step. In principle these gates determine what information the cell should forget and what information the cell should hold onto. Thus, by the end of all the processing in the memory cell the cell state is updated so that it includes only the important information from the last time step and the important information from the input at the current time step. Since their conception the long short term-memory architecture has been used to make great advances in sequence modelling [Kalchbrenner et al., 2015] performing extremely well on tasks like phoneme classification [Graves and Schmidhuber, 2005] and speech recognition [Graves et al., 2013].

The first gate in the memory cell is referred to as the forget gate f, and its output at time t is determined by:

$$f_t = \sigma(W^r_{fy_{(t-1)}}[\mathbf{y}_{t-1}, \mathbf{x}_t] + b_f) \tag{2.18}$$

Where $[y_{t-1}, x_t]$ denotes the concatenation operation on the output vector from the previous time step and the input from the current time step. For each element in the concatenated vector the sigmoid function will compress the input into the range [0,1] where 0 represents forgetting that input and 1 represents remembering it. This will then inform the information in the cell state which of the corresponding elements in its vector should be kept and which should be forgotten. The next two gates are the input gate i and a tanh gate $\tilde{C}$. The input gate decides which parts of the cell state will be updated while $\tilde{C}$ creates a vector of new state values

based on the input at the current time step and the output of the previous time step. The state vectors of these gates are given by:

$$i_t = \sigma(W_{ih_{(t-1)}}[\mathbf{h}_{(t-1)}, \mathbf{x}_t] + b_i) \tag{2.19}$$

$$\tilde{C}_t = \tanh(W_{\tilde{C}h_{(t-1)}}[\mathbf{h}_{(t-1)}, \mathbf{x}_t] + b_{\tilde{C}}) \tag{2.20}$$

The next step is to determine the cell state based on the information that has been processed so far. The current cell state C is given by:

$$C_t = f_t C_{(t-1)} + i_t \tilde{C}_t \tag{2.21}$$

From this equation we can see that the information from the forget gate, the previous cell state, the input and the tanh gate are combined together to give a new cell state which then becomes the cell state for the memory cell at time $(t + 1)$. Finally the output y of the memory cell is produced by filtering this cell state through a final tanh layer and combining it with the output of another sigmoid layer that determines which elements of the cell state will be output. These are given by:

$$o_t = \sigma(W_{oh_{(t-1)}}[\mathbf{h}_{(t-1)}, \mathbf{x}_t] + b_o) \tag{2.22}$$

$$h_t = o_t \tanh(C_t) \tag{2.23}$$

From this point h_t is delivered both as the output of the memory cell and as input to the next memory cell at $t + 1$. Using this relatively complicated series of operations long short-term memory networks are able to avoid the problems of forgetting long term dependencies that arise with the vanishing and exploding gradient problem.

2.5.2 Attention

In the field of neural machine translation - using neural networks for the task of translating written sentences - RNNs arranged into an encoder-decoder architecture [Cho et al., 2014, Sutskever et al., 2014] have been used to a great degree of success in a number of tasks [Lu et al., 2014]. These architectures are commonly assigned the task of predicting the next word in a sentence by encoding an input sentence into a fixed length vector and then decoding this fixed length vector to form a prediction over possible next words. This method is then used to translate an input sentence in one language into another language by outputting a probable

translation of a word in the input given the previously translated words. Intuitively, encoder-decoder architectures can be thought to be learning an abstract representation of their inputs that equally correspond to their outputs. Think of how "dog" in English and "chien" in French are very different words: they are different lengths and contain none of the same letters. However, people who speak both languages are able to relate these two words by way of a common representation i.e. a medium sized mammal that barks and has a long tail and a snout. In the same way, encoders will learn an abstract numerical representation that effectively links together the source and the target words. For the encoder the task is to model a fixed length vector c from an input sentence $x = (x_1, ..., x_{T_x})$ which is a sequence of T vectors of the same size learned by some embedding process. The context vector can be learned with an RNN (or some other recurrent architecture) such that c is some function of the hidden states of the RNN for each word in the input sentence:

$$c = f(\{h_1, .., h_T\}) \tag{2.24}$$

where each hidden state is some non-linear function of the t^{th} word in the sentence and the previous hidden state:

$$h_t = q(x_t, h_{t-1}) \tag{2.25}$$

This is identical in form to the RNNs discussed above with the only difference being that each fixed length input vector is indexed not by time but by word order - which makes sense given that we are dealing with text data.

Formally, the decoder finds:

$$p(\mathbf{y}) = \prod_{t=1}^{T_x} p(y|\{y_1, ..., y_{t-1}\}, c) \tag{2.26}$$

where $p(\mathbf{y})$ is a probability over possible translations of the input sentence, y_t is the t^{th} word in the translated sentence and c is the vector that is encoded by the encoder layer. Finally, the product is over the ordered conditionals of the translated individual words, and $\mathbf{y} = (y_1, ..., y_{t-1})$. We can then model the individual probabilities on the right hand side of Equation (2.26) using an RNN:

$$p(y|\{y_1, ..., y_{t-1}\}, c) = g(y_{t-1}, h_t, c) \tag{2.27}$$

where f is a non-linear function learned by the RNN (or some other recurrent architecture) which in turn is a function of the previous translated word y_{t-1} in the translated sentence and the hidden state h_t of the RNN for word t.

A major drawback of these architectures is in the construction of c. In principal the encoder is required to compress all the information about the input sentence $\mathbf{x}$ into a fixed length vector c. Since the size of c is fixed we suppose that a vector of this length is appropriately sized for representing information from sentences of all length i.e. it should capture the import components of very short sentences to very long sentences. This obviously becomes problematic as the size of the input sentence grows very large which is clearly a problem for the generalizability of the models themselves.

To address this problem [Bahdanau et al., 2014] proposed a mechanism that has come to be known as "attention". In their model, the fixed length vector c of the encoder-decoder model described above is replaced by i so-called context vectors where i indexes each word in the sentence. Each context vector c_i in turn is a function of a sequence of annotations $S = (s_1, ..., s_{T_x})$. Intuitively the annotation for the i^{th} word in the input sequence contains information about the entire sequence in relation to that word, paying particular focus to the words on either side of it. This makes sense outside of the context of neural networks since, in a given sentence, the interpretation of a word at a given position in a sentence is likely to be strongly influenced by the words on either side of it. Thus the hidden state for the encoder RNN at the i_{th} word in the input sequence is computed as:

$$h_i = f(s_{i-1}, y_{i-1}, c_i) \tag{2.28}$$

As a consequence we adjust the way in which we compute the probabilities over possible output sentences. Now, each ordered conditional from Equation (2.26) becomes:

$$p(y_i|y_1, ..., y_{i-1}, \mathbf{x}) = g(y_{i-1}, h_i, c_i) \tag{2.29}$$

such that each probability is computed as a non-linear function of the previous outputted word, the hidden state for word i, and the context vector for the i^{th} word. We can see from Equation (2.29) that the major change from the traditional encoder-decoder model is that each input word is assigned its own context vector meaning that the size of the latently represented information about the input sentence will grow in proportion to the size of the input thus, in principle, solving the problem of representing arbitrarily lengthed input sequences with a fixed length latent vector.

The context vector for each word is computed as a weighted sum of the annotations:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} s_j \tag{2.30}$$

where

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{2.31}$$

and

$$e_{ij} = a(h_{i-1}, s_j) \tag{2.32}$$

Equation (2.32) produces what is called an alignment model. Intuitively the alignment model computes and scores how well the i^{th} output word matches the words around the j^{th} input. This score is then used to weight the i^{th} annotation as in Equation (2.31). Since the alignment model is a function of the previous hidden state and the j^{th} annotation it can be learned using a neural network (a linear fully connected network is used in [Bahdanau et al., 2014]).

At a high level we can understand α_{ij} to represent the importance of a particular word in relation to outputting the next word in the translation. This value tells the model how much attention should be placed on a given word in the input sentence when translating the next word in the sentence. For example, when interpreting the word "dog" from the sentence "a dog walked into the woods" we want the model to attend more strongly to the words "a" and "walked" since these words relate directly to the dog (where "a" picks out the particular dog and "walked" is an action performed by the dog itself).

Since [Bahdanau et al., 2014] RNN based attention mechanisms have been implemented to great affect in a number of different challenging time series modelling tasks from text classification [Liu and Guo, 2019], sentiment analysis [Wang et al., 2016], action recognition [Liu et al., 2017], and video captioning [Gao et al., 2017], in each case improving upon non-attention based RNN methods.

2.5.3 Transformers

Since the success of RNN based attention mechanisms increasing focus has been placed on the attention mechanism itself in relation to the ability that these kinds of methods have to model complex time series tasks. This culminated in the work of [Vaswani et al., 2017] which showed that models constructed *only* using stacked attention mechanisms were able to outperform RNN based attention methods on a number of natural language processing tasks as well as being significantly faster and more efficient to train. Surprisingly these models, known as Transformers, do

away entirely with recurrent connections while still being able to model long term dependencies in time series related problems.

Transformers retain the overall encoder-decoder architecture outlined in Section 2.5.2 in the sense that a stack of attention layers encodes an input sentence into a continuous internal representation that is then decoded to produce the output sentence one element at a time. Transformers introduce three key computational components that allow them to model time series data in such an efficient way: the multi-head attention mechanism, a position-wise feed-forward network, and a positional encoding for the source data. Each of these will now be explained in turn.

[Vaswani et al., 2017] generalize the attention model in [Bahdanau et al., 2014] in the following way. Let the context vector c_i be the output of an attention function, call the annotations h_i values v, the inputs around position j queries q, and the output at position i a key k. In this framework an attention function maps a set of key-value pairs and a query to an output. Just as above, the output is a weighted sum of the values, and the weights are computed as a function of the queries q and the keys k. In the transformer architecture the queries and keys have dimension d_k while the value vector has dimension d_v. The architecture then uses a version of attention called dot-product attention since we perform the dot product operation on the query and key vectors (before running the result through a softmax layer to normalise the resulting weights to between 0 and 1) and then on the resulting vector and the value vector respectively. The transformer architecture modifies this logic somewhat by scaling the dot product of the queries and keys by a factor of $\sqrt{d_k}$ which prevents the output of the softmax function from growing too large as the size of d_k gets large, making the model more compatible with larger inputs. In reality these operations are performed on matrices that collect together a set of the queries, keys, and values giving the resulting formula for computing the attention function:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.33}$$

The principal algorithmic development with respect to the attention mechanism itself comes in the stacking of these dot-product attention layers. First each value, key, and query vector is copied h times and fed through separate linear neural network layers to create h distinct representations of each vector. Each one of the h sets of queries, keys, and values is then fed through its own scaled dot-product attention mechanism where each attention mechanism is referred to as a *head*, hence we refer to the mechanism as a whole as a *multi-head attention mechanism*. Finally, the outputs of the attention heads are concatenated and projected through

another linear neural network layer. We can then modify Equation (2.33) to reflect this stacked-head architecture:

$$\text{MultiHeadAttention}(Q, K, V) = \text{concat}(\text{head}_1, .., \text{head}_h)W^O \tag{2.34}$$

where

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.35}$$

Where W^O, W_i^Q, W_i^K, and W_i^V introduce learnable weight matrices of the i^{th} attention head.

The output of each encoder and decoder layer is computed in part by a fully connected linear neural network layer that acts upon each individual input x to the network (in the case of translation this would be the equivalent to each word in the input sentence). Since the network acts on each position of the input, we refer to this network as a *position-wise feed forward* network. We compute the output of this network using a ReLU activation function:

$$\text{PositionWiseFeedforward}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2. \tag{2.36}$$

On obvious question arises from this set up. Namely, given that there are no recurrent connections present, how does a transformer model time? To do this the authors use a positional encoding: a matrix that can be used to impart temporal information on a matrix of the same dimension that contains time series data. In this case the positional embedding is computed as alternating sine and cosine functions on odd and even numbered input positions respectively:

$$\text{PE}_{(pos,2i)} = \sin\left(\frac{pos}{1000^{2i/d_{model}}}\right) \tag{2.37}$$

$$\text{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{1000^{2i/d_{model}}}\right) \tag{2.38}$$

This creates a geometric progression where the encoding has a larger and larger effect on the input matrix as you move down the positions, i.e. the rows of the input (a visualisation of this position encoding matrix is show in Figure 2.5).This positional encoding matrix is then added piece-wise to the input matrix. In an abstract sense, this imparts temporal information on the input data since the effect of the positional encoding function will increase as the position of the input gets further away from the beginning. For example, in the sentence "the cat sat on the mat", since the word "cat" if close to the start of the sentence that the word

"mat", the numerical representation of that word will be more significantly altered by the positional encoding. Thus the model is able to represent things as being closer or further away from a particular point, an ideal representational format for modelling time.

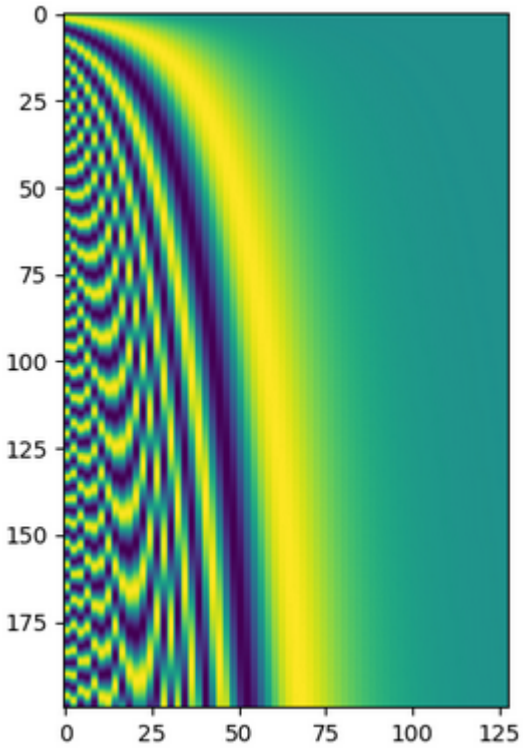


Figure 2.5: A visualisation of the positional encoding matrix on a input time series with 200 time steps and 127 features.

With this formulation in place, we can now describe the transformer architecture as a whole. First, our input is added to the positional encoding matrix before being sent to the encoder layer. The resulting matrix is then copied. The first copy is split into query, key, and value vectors and sent to the first h-head attention stack. The output of the first attention stack is then added to the copied input (via what is known as a residual connection [He et al., 2016]) and then layer normalized. This output is then copied, with the first copy getting sent to a position-wise feed-forward network. Again, the output of this network is added via a residual connection to the copied attention-stack output and layer normalized.

In the decoder, we take the output of the network at the previous time step and positionally encode it. During network training this amounts to feeding the network a ground truth matrix of targets for the input data. For a neural translation model this would be the embeddings of the translated words for the input sentence. In its current iteration the network has access to all the information in the ground truth matrix since we provide the entire target i.e. the whole of the translated sentence. This is a problem because we want the model to predict the next word

in the sentence based only on the preceding words (as in Equation (2.29)) as in the real world this is the only information that the decoder layer will be able access. To prevent the decoder layer from accessing information at subsequent positions, a mask is applied to the input. This masked and positionally encoded input is copied and then fed to a multi-head attention mechanism (referred to in [Vaswani et al., 2017] as a masked multi head attention mechanism for this reason). As in the encoder layer, the output of this attention sublayer is added to the copied input and layer normalised. This output is copied and projected as a value vector into a second multi-head attention sublayer (as a marked difference from the encoder layer which contains only a single attention sub layer). The key and value vectors for this second multi-head attention sub-layer are constructed from the output of the encoder layer. Once again the output from this attention mechanism is added to the copied output from the masked multi-head attention sublayer and layer normalised. The final step in the decoder, just as in the encoder layer, is to copy this output, pass one copy to a position-wise feedforward network, add the residual copy to the output of this network and then layer normalize for a last time. Finally, to obtain the output probabilities $p(\mathbf{y}) = \prod_{t=1}^{T_x} p(y|\{y_1, ..., y_{t-1}\})$ we pass the decoder output to a linear feedforward layer and then to a softmax layer to retrieve the normalized probabilities. A visualisation of the whole set up is shown in Figure 2.6.

Since their conceptualisation transformer models have formed a significant part of the state-of-the-art for natural language processing tasks culminating in a number of landmark papers in the field [Devlin et al., 2019, Brown et al., 2020]. Outside of the natural language setting transformers have been successfully adapted to become state-of-the-art in a number of other time series modeling tasks including video understanding [Fan et al., 2021, Feichtenhofer et al., 2018], action recognition [Mazzia et al., 2021], and stock market prediction [Ding et al., 2020].

2.5.4 Modern Hopfield Networks

Since the development of early binary Hopfield networks [Hopfield, 1982] - ones where the input stored patterns are composed of strings of either 1s or 0s - much work has been undertaken to improve their performance in a number of areas, most notably with respect to increasing their storage capacity. Recall that, at a high level, Hopfield networks function by recalling stored patterns given a partial input. One way in which we can query the effectiveness of Hopfield networks is thus by asking how many patterns a network of a particular size can store? Intuitively the more patterns a network can store, the more flexible that network will be with respect to different inputs. Straightforwardly, the more patterns a

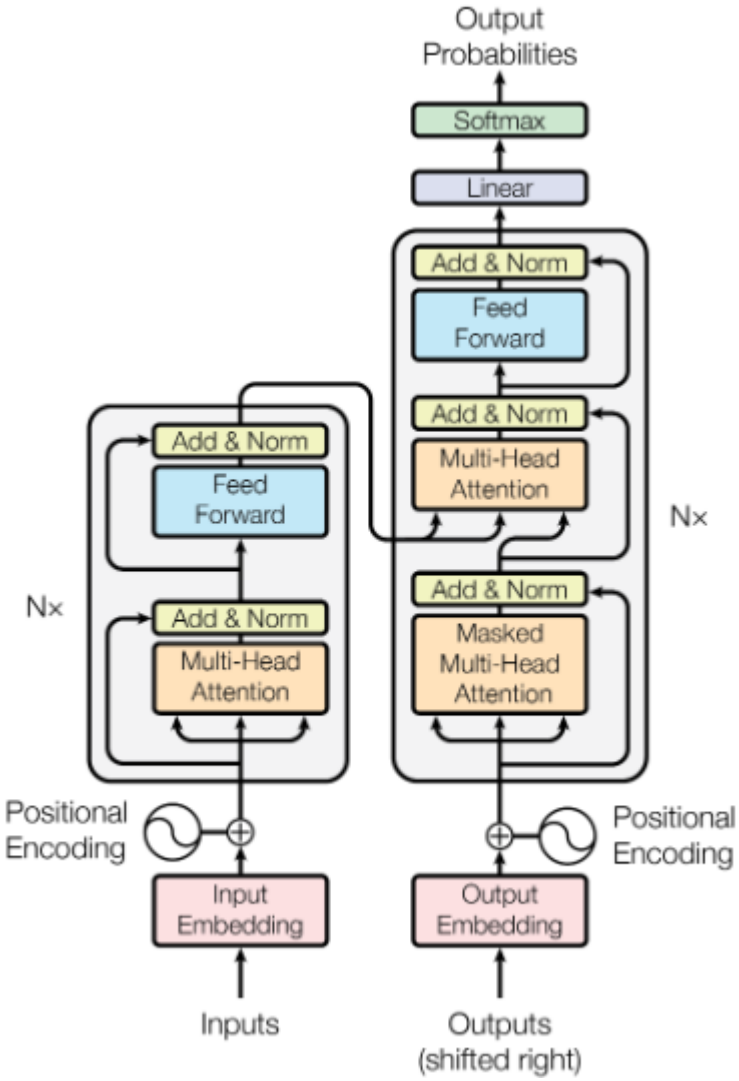


Figure 2.6: A visualisation of the entire transformer architecture reproduced from [Vaswani et al., 2017]

network can store, the higher number of partial inputs it will be able to recreate accurately. For these classical Hopfield networks, the number of possible stored patterns depends on the number of neurons and the specifics of the network's update rules (e.g. Equation (2.7)). Classical Hopfield networks with different update rules and implementations are able to store a number of patterns ranging from $\frac{Cd}{\log(d)}$ to $Cd\log(d)$ [Abu-Mostafa and St. Jacques, 1985, McEliece et al., 1987, Folli et al., 2017] where d is the dimension of the problem space, i.e. the size of the patterns that are to be retrieved or the number of cells in the network.

The ability of Hopfield networks to store and retrieve patterns of a particular size is of clear importance to a number of machine learning tasks. In their classical iteration however they do not generalize well to more complex problem spaces. This is due to the fact that classical Hopfield networks represent binary and hence discrete task spaces i.e. where the input data and stored patterns are strings of

1s and 0s. For problems like image recognition, video classification, or natural language processing it is necessary that a neural network model be able to deal with continuous task spaces since the inputs to the networks in these cases are likely to be vectors or matrices over the set of real numbers.

Having a Hopfield network that operates over continuous task spaces is advantageous for a number of reasons. Firstly, it allows Hopfield networks to be integrated easily into layered neural network architectures such as those in deep learning. In principal this allows a particular layer of a network to be bestowed with an associative memory by virtue of Hopfield networks' ability to store and retrieve patterns. Secondly, continuous states allow the network to be differentiable which is an essential factor in permitting the back propagation algorithm to work. This means that Hopfield networks can be stacked into layers to form deep networks, greatly increasing their ability to perform well on machine learning tasks by representing increasingly abstract features of the input space as the number of layers grows.

To address this issue [Ramsauer et al., 2021] generalised the classical Hopfield network into one that could deal with continuous input spaces. To grasp how this generalisation is possible we need to first understand two concepts related to how Hopfield networks learn patterns: energy functions and update rules.

Energy functions work in much the same way as a loss function for standard neural networks in the sense that learning in a Hopfield network amounts to minimizing that network's energy function. Here, a network's energy is a function of an input and the networks hidden state. For classical Hopfield networks the input amounts to the partial version of some internally stored pattern that you query the network with. The hidden state in this case is a matrix of stored patterns in the form of binary vectors. For these classical Hopfield networks the energy of the network can be expressed as:

$$\mathrm{E} = -\exp(\mathrm{lse}(1, \mathbf{x}^T H)) \tag{2.39}$$

[Demircigil et al., 2017] where $\mathbf{x}$ is the network state which is initialised as a query vector i.e. the partial input, H is the matrix of stored patterns and where lse() is the LogSumExp function:

$$\mathrm{lse}(\beta, \mathbf{x}) = \beta^{-1} \log \left(\sum_{i+1}^{N} \exp(\beta x_i) \right) \tag{2.40}$$

where N is the number of stored patterns. From Equation (2.39) we can see that for binary networks we set $\beta = 1$. At an abstract level the idea is that when this energy function is minimized you are able to retrieve the stored pattern that

corresponds to the partial input (the query). This raises the question of how to minimize the energy function. In Hopfield networks this is done by an update rule which tells the network how to adjust its parameters (the weights between the neurons and their respective firing thresholds) in such a way that minimizes the Network's energy. As we saw in section Section 2.3 a Hopfield network is updated i.e. the state of each neuron is changed, using the following formula:

$$s_j = \begin{cases} +1 & \text{if } \sum w_{ji} s_i \geq b \\ -1 & \text{if } \sum w_{ji} s_i < b. \end{cases} \tag{2.41}$$

in practice each iterative update of the network will drive the network state towards a position of low energy. Pictured as a point on the energy surface, this position of low energy amounts to the minimum of some valley on the surface. These minimum positions, within which the network is no longer able to change its state, equate to the state of the network now representing the retrieved pattern. Thus minimizing the network energy amounts to pushing the network state into one of the valleys so that a stored pattern can be retrieved. This is similar in principal to optimizing a deep network with respect to some cost function where we update the weights of the network in order to drive the cost function into some global or local minimum.

In their current iteration, these update and energy rules still only deal with binary patterns where each element of a pattern vector in H is a string of 1s and 0s. [Ramsauer et al., 2021] generalise these binary energy functions and update rules to continuous states in the following way. First the energy E of the network is computed as:

$$\mathrm{E} = -\mathrm{lse}(\beta, \mathbf{x}^T H) + \frac{1}{2}\mathbf{x}^T\mathbf{x} + c \tag{2.42}$$

where $c = \beta^{-1} \log \mathrm{N} + \frac{1}{2} \max_i \|\mathbf{x}_i\|^2$ which bounds the norm of the state vector x to prevent the energy from exploding. To generalize to continuous states we allow β to be some value greater than 1 i.e. some positive real valued number (rather than fixing it as 1 in classical Hopfield networks as in Equation (2.39)).

To update the network state we use the following rule:

$$\mathbf{x}^{new} = H\mathrm{softmax}(\beta H^T \mathbf{x}) \tag{2.43}$$

Abstractly, this can be related to the functioning of recurrent architectures outlined above in the sense that the state of the network at some time is a function of the state of the network at the previous time step. Again, this update function pushes the energy into a low-energy valley to retrieve the most likely pat-

tern. In fact, if we interpret softmax($\beta H^T \mathbf{x}$) in Equation (2.43) as outputting a vector of probabilities over the stored patterns H, we can see that as x is updated softmax($\beta H^T \mathbf{x}$) will approach a vector with zeroes everywhere other that the position of some vector in H which will causes x to equal that stored pattern and hence retrieve it.

The authors of [Ramsauer et al., 2021] show analytically that Equation (2.43) can be expressed as:

$$\text{softmax}(\beta \mathbf{x} H) H^T \tag{2.44}$$

and that this is equivalent to Equation (2.33) i.e. the attention mechanism in a transformer architecture. Not only this, but continuous state Hopfield networks can be shown to function as pooling mechanisms such as those commonly found in convolutional neural network architectures. While an explanation of how this works is beyond the scope of this chapter, it demonstrates the remarkable flexibility of Hopfield networks with respect to helping to solving machine learning problems. Indeed, architectures containing continuous state Hopfield networks have been shown to produce state of the art results on drug-design [Ramsauer et al., 2021] and immune repertoire classification [Widrich et al., 2020] problems, two complex machine learning tasks.

2.6 Conclusion

In this chapter, I have summarised the architectural and mathematical concepts underpinning neural networks that are commonly used to model time series data. These particular types of networks were chosen due to their ability to model memory, which was argued to be of central importance to a large number of socially oriented human cognitive abilities. While there have been a number of advances in the field beyond these, many, if not all, of these are based in part on the ideas covered above. First, I reviewed the most commonly used activation functions, which are the operations that determine the output of each neuron in a recurrent network. Next, I reviewed the three earliest recurrent architectures: the Hopfield network, the Jordan network, and the Elman network. These were seen to be the first neural network architectures that introduced recurrent edges between the hidden neurons. These were seen to act as a sort of memory that allowed the network to "remember" the output of the network at previous time steps. I then went into some detail about the most common processes by which recurrent neural networks are trained - namely, via calculating error gradients and updating the parameters of the network accordingly. We then saw how, because of the recurrent nature

of these networks, this training process could lead to bad training results due to problems known as the vanishing and exploding gradient. Finally, I reviewed the solutions to these problems that have been provided in the literature with a focus on long short-term memory networks, attention mechanisms, transformers, and continuous state Hopfield networks which, in some format, account for a large number of important results in sequence learning over the past decade.

Chapter 3

Is Deep Learning a Valid Approach for Inferring Subjective Self-Perceptions in Human–Robot Interactions

Henry Powell

Guy Laban

Jean-Nöel George

Emily S. Cross

[1]A version of this chapter was accepted for publication and presentation at *ACM International Conference on Human–Robot Interaction 2022* under the title: "Is Deep Learning a Valid Approach for Inferring Subjective Self-Disclosure in Human-Robot Interactions?"

Abstract

Advances in artificial agents and social robots are already beginning to demonstrate how these machines can provide care to individuals, be integrated into psychosocial interventions, and support people's mental health. One limitation of these platforms has been the ability of the models they operate on to infer meaningful social information about people's subjective perceptions, specifically from non-invasive behavioral cues. Accordingly, our paper aims to demonstrate how different deep learning architectures trained on data from human-human, human-agent, and human–robot interactions can help artificial agents to extract meaning, in terms of people's subjective perceptions, in speech-based interactions. Here we focus on identifying people's perceptions of their subjective self-disclosure (i.e., to what extent one perceives to be sharing information with an agent), and identifying the degree of one's psychological stress (i.e., the extent to which one is experiencing their life to be stressful). We approached this problem in a data-first manner, prioritizing high quality data over complex model architectures. In this context, we aimed to examine the extent to which relatively simple deep neural networks could extract non-lexical features related to these two kinds of subjective self-perception. We show that five standard neural network architectures and one novel architecture, which we call a Hopfield Convolutional Neural Network, are all able to extract meaningful features from speech data relating to subjective self-disclosure and psychological stress.

3.1 Introduction

Artificial agents are autonomous machines or computer software that interact and communicate with humans or other agents by following social behaviours and rules attached to their role [Breazeal, 2003]. Both social robots [Robinson et al., 2019, Scoglio et al., 2019], embodied [Lucas et al., 2017][Scherer et al., 2016], and disembodied [Lee et al., 2020, Tudor Car et al., 2020] agents are gradually being introduced as viable means to deliver psychological and emotional health-care interventions to support or improve mental and physical health. These cognitive agents can function autonomously in physical (in the case of social robots) and virtual spaces and within social settings, be programmed to support clinical management, and hold great promise for supporting people in need [Henschel et al., 2021]. Cognitive agents' embodied cognition and human-compatible designs can elicit socially meaningful information and behaviours from humans [Hortensius et al., 2018][Hortensius and Cross, 2018] (e.g., [Laban et al., 2021b][Lucas et al., 2014]), encourage human users to establish meaningful social relationships with them ([Cross and Ramsey, 2021][Henschel et al., 2021]; e.g., [Croes and Antheunis, 2020][Cross et al., 2019b][Riddoch and Cross, 2021]), and provide innovative, nuanced and potentially cost-effective eHealth solutions for supporting psychological health.

Nevertheless, it is uncontroversial that artificial agents do not yet offer the same opportunities as humans for social interactions (see [Cross et al., 2019a]). Cognitive agents are still limited in terms of inferring meaningful social information from speech disclosures. Most humans, on the other hand, effortlessly engage in theory of mind (i.e., taking the perspective of another person) and are generally capable of using these abilities when communicating and generally navigating a complex social world [Catmur et al., 2016][Premack and Woodruff, 1978]. While humans are not always successful in understanding others' subjective perceptions [Keysar et al., 2003], their ability to infer information and make presumptions about others is a natural learning process honed across psychological development [Baron-Cohen, 1991]. Further, these abilities support a substantial part of human social cognition [Catmur et al., 2016]. As such, humans can effortlessly infer how a conversation partner subjectively perceives themselves, a situation, and others, based on verbal and non-verbal social cues transmitted during speech [Byom and Mutlu, 2013].

The ability to understand others based on such cues is crucial in the context of delivering psychosocial therapy, interventions, or when diagnosing someone's psychological health [Fernyhough, 2008][Mattingly, 1991]. Without the ability to infer the internal states of others, via direct explicit disclosure or indirect implicit

behaviours, it becomes impossible to maintain the intervention flow, and in turn, to provide meaningful psychological support [Fernyhough, 2008]. While humans can intuitively infer complex social information regarding a conversation partner, artificial agents need to synthesize and analyze multiple kinds (or channels) of data from a human interaction partner in order to appropriately and accurately"read" complex social meanings [Kappas et al., 2020].

Hence, the aim of the current study is to determine how different deep learning architectures trained on data from human–human, human–(voice) agent, and human–robot speech interactions can help those cognitive agents to synthesize and extract meaning about a human interlocutor's subjective perceptions of themselves, the situation, and others. We specifically focus on validating the viability of using deep learning approaches for predicting subjective perceptions of an interlocutor's self-disclosure, and the degree of periodic psychological stress from vocal disclosures. Subjective self-disclosure and perceptions of periodic stress are both important elements in health intervention communication [Colquhoun et al., 2017][Wight et al., 2016]. Due to the rapid nature of ongoing speech interactions, and the availability of voice data in speech interactions, we decided to focus on voice parameters (i.e. acoustic as opposed to lexical parameters) in the presented architectures. In addition, vocal prosody features and voice signals provide implicit indicators to behaviour and emotions [Frick, 1985][Roach et al., 1998, Yang et al., 2013][R. et al., 2003][Giddens et al., 2013], and the psycho-physiological underpinnings associated with these cues and can aid in more fully understanding a person's mental state (e.g., [Giddens et al., 2013][Ruiz et al., 1990][Slavich et al., 2019]). Moreover, voice-based social signals are thought to be a primary means by which a person communicates subjective self-disclosure and psychological stress [Soleymani et al., 2019][Cozby, 1973][Omarzu, 2000][Van Puyvelde et al., 2018].

While we treat subjective self-disclosure and psychological stress as separate factors in this study, the two are often thought to be correlated. Intuitively, we can understand that a person's being significantly stressed may well influence the degree to which they are willing to self-disclose information to a social partner. After all, one's being stressed, and the situations that have lead to that state affairs, may well make up the content of a self-disclosure. Empirically, studies have shown that, in certain situations, these two factors are indeed correlated with one another [Han and Yu, 2012][Hamid, 2000], although the nature of this correlation is not consistent across groups. In light of this, our motivation for studying subjective self-disclosure and psychological stress along side one another was not only due the importance of these two factors in the social situations described above, but also because an understanding of one may well influence how well we are able to model the other and vice versa. While a full treatment of this correlation with respect to

our aim of modelling subjective self-disclosure and psychological stress is beyond the scope of this initial study, this idea, and its potential for further work in this area, is discussed in Section 3.8.

3.1.1 Subjective Self-Disclosure

Self-disclosure is a communication behaviour aimed at revealing oneself to others. It is a key factor for building relationships between two individuals [Jourard and Lasakow, 1958][Pearce and Sharp, 1973] where people share thoughts and feelings with others, especially when experiencing unique and challenging life events [Gable et al., 2004]. Disclosure thus serves an evolutionary function of strengthening interpersonal relationships, but also produces a wide variety of health benefits, including coping with stress and traumatic events and eliciting help and support [Frattaroli, 2006][Frisina et al., 2004][Kennedy-Moore and Watson, 2001]. Self-disclosure is a complex social dynamic that consist of multiple dimensions. One dimension of self-disclosure includes how one's disclosure is objectively perceived by others from the shared content of the disclosure, or one's behaviour when communicating a disclosure. Another dimension of disclosure refers to subjective self-disclosure, being the extent of personal information one perceives to be sharing during an interaction [Antaki et al., 2005][Kreiner and Levi-Belz, 2019][Levi-Belz and Kreiner, 2016][Omarzu, 2000]. Self-reported measurements (e.g., [Jourard, 1971][Jourard and Lasakow, 1958]) convey subjective dimensions of self-disclosure evaluating people's retrospective perceptions [Kahn et al., 2012][Kreiner and Levi-Belz, 2019]. One's self-perceptions of self-disclosure are meaningful in understanding how one perceives certain settings, situations, and oneself [Schlosser, 2020]. Furthermore, health interventions, therapy, and clinical communication are dependent on open channels of communications, relying on one's belief to be sharing and disclosing relevant and personal information from which a listener can identify stressors and respond accordingly [Colquhoun et al., 2017][Wight et al., 2016]. Self-disclosure appears to play a critical role in successful treatment outcomes [Sloan, 2010] and has a positive impact on mental and physical health [Derlega et al., 1993]. This is particularly important for self-help eHealth platforms, autonomous assistant systems, and for personalizing interventions, as these should be able to use the rich input provided by users to extract salient information, identify patterns and emotional states, and respond accordingly [Riva et al., 2012].

Previous studies demonstrated that people's perceptions of self-disclosure are relatively aligned with their actual observed behaviour of disclosure [Laban et al., 2021b]. Accordingly, since single dimensions cannot capture the complex nature of self-disclosure, as it is a multidimensional behaviour [Kreiner and Levi-Belz,

2019], we are interested in determining to what extent subjective dimension of self-disclosure can be objectively observed and predicted from vocal behaviour. Therefore, we test the viability of a number of standard deep learning architectures and one novel one with the aim of predicting individuals' perceptions of subjective self-disclosure from their actual vocal output during a speech-based social interaction.

3.1.2 Psychological Stress

Psychological stress is the extent to which an individual perceives that their demands exceed their ability to cope, and therefore they subjectively appraise a situation or period of time as stressful [Cohen et al., 1983][Lazarus, 1974][Lazarus, 1966][Lazarus and Folkman, 1984] [Phillips, 2013]. Previous studies demonstrate that using subjective, self-report instruments for measuring psychological stress (i.e., the perceived stress scale (PSS) [Cohen et al., 1983]) are valid and reliable (see [Roberti et al., 2006]), and can explain how people perceive specific events or life periods as stressful [Feizi et al., 2012][Rossi et al., 2021]. Subjective measurements of psychological stress are often used to determine the effectiveness and validity of stress-reducing interventions (e.g., [Stillwell et al., 2017][Zadok-Gurman et al., 2021, Stächele et al., 2020]). Such measurements are also used to assess relationships between psychological stress (as a subjective perception) and psychiatric conditions (e.g., [Li and Lyu, 2021][Wiegner et al., 2015][Zandifar et al., 2020][Aslan et al., 2020], physical health issues (e.g., [Wiegner et al., 2015][Rueggeberg et al., 2012][Vancampfort et al., 2017][Yang et al., 2020b]), and to help predict objective biological markers of stress, like cortisol levels (e.g., [Walvekar et al., 2015][Wu et al., 2018][Linz et al., 2018]).

Here we are interested in determining to what extent one's subjective perceptions of psychological stress can be objectively observed and predicted from vocal behaviour. To accomplish this, we test the viability of a number of standard and one novel deep learning architectures with the aim of predicting individuals' perceptions of their periodic psychological stress from their actual vocal output during speech.

3.1.3 The Current Paper

The remainder of the paper takes the following form. In Section 2, we describe the experimental paradigm and data collection from [Laban et al., 2021b][Laban et al., 2020] in detail. In Section 3, we explain the data augmentation techniques we used to balance the data set. Next, in Section 4 we outline our deep learning experiments and the neural network architectures we used to conduct them. In Section 5,

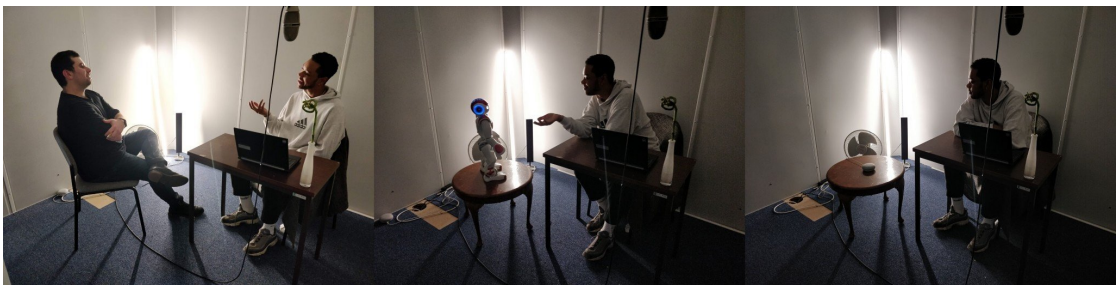


Figure 3.1: Illustration of the experimental design. From left to right: human talking to a human agent, human talking to the social robot NAO (SoftBank Robotics), and human talking to the disembodied agent (voice assistant Google Nest Mini).

we detail the performance of the model and the results of our experiments on other popular architectures in comparison to our own. In Section 6 we discuss limitations of the model in relation to the aim of implementing it in real-world scenarios, and provide some avenues for further improvements. Finally, in Section 7, we summarize our work and discuss the broader contribution of the models and our findings.

3.2 Data Set and Data Collection

In order to generate data for the models, three laboratory experiments were conducted, as reported in detail previously [Laban et al., 2021b][Laban et al., 2020]. Exploratory empirical results of all three experiments are reported in [Laban et al., 2021b]. The three laboratory experiments (N1 = 26; N2 = 27; N3 = 61) consisted of within-subjects experimental designs with three treatments. In a randomized order, participants were asked one (in the first experiment) or two (in the second and third experiments) pre-defined questions about their everyday life experiences by each of the three agents: (1) a humanoid social robot (NAO by Softbank Robotics), (2) a human, or (3) a disembodied agent (a "Google mini" voice assistant) (See Figure Figure 3.1). The three agents communicated the same pre-scripted questions via the Wizard of Oz (WoZ) technique controlled by the experimenter (except for the human agent), demonstrating different visual and verbal cues that corresponded appropriately to their form and capabilities.

The questions' topics were randomly allocated to the agents, and the questions within each topic were randomly ordered. All three experiments took place in a sound-isolated recording laboratory. The recording room was completely soundproof to ensure the highest possible sound quality for the recordings to facilitate offline analyses. After the three interactions, participants answered a questionnaire.

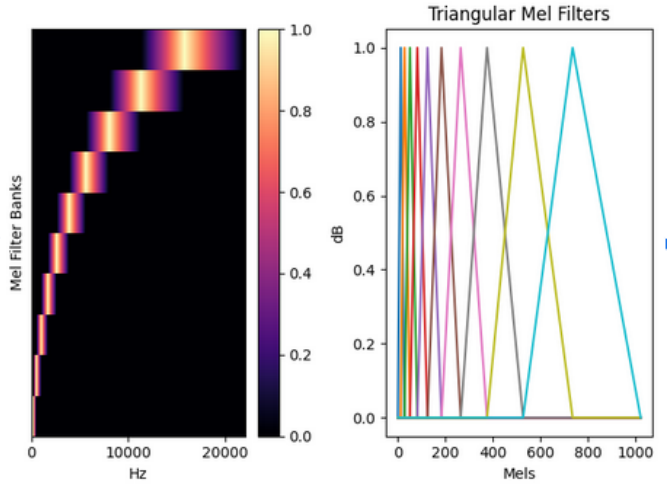


Figure 3.2: Example of 9 computed mel-filter banks.

For a full and detailed report of the data collection methodology, the sample, and dataset, see [Laban et al., 2021b].

For our deep learning experiments we chose to collapse the data into a singular dataset as opposed to training individual models along a voice agent, embodied robot, human agent split. An investigation into the behavioural differences between these three conditions was carried out in [Laban et al., 2021b]. For the deep learning arm of our experiment we were interested in capturing representational components of speech data that were common to all three of these conditions. This was largely due to the consideration that, in a real world environment, these social robots will plausible need to be flexible with respect to their ability to detect perceived psychological stress and self-disclosure in a variety of dyadic interaction scenarios. We believed that the most efficient way to capture this flexibility was by training models on collated data from all scenarios so that task-specific, dyad-agnostic based representations could be learned by our models. A further consideration that we took in this regard was that splitting the data into three groups would significantly reduce the amount of data that we had available for model training in each case. Since the success of our models was very likely to be tied to the quantity of data we had relating to our problem, we decided that collating the data was the most sensible approach.

We acknowledge the limitation of using WoZ paradigms to collect ecologically valid data for interactions involving artificial agents. When such agents are eventually introduced into real world contexts, the aim will be for them to function autonomously in natural, free-flowing interactions, which will possibly unfold quite differently from the staged conversational set up that Wizard of Oz paradigms offer. Keeping this limitation in mind, in order to minimise the impact of the WoZ approach, the interactions were restrained to a limited vocabulary that corresponded

to the current state of the technology. Additionally, considering the complexity of speech interactions with artificial agents in general, the experimental settings of WoZ offered reliable parameters for collecting speech data at present [Brutti et al., 2010][Niebuhr and Michaud, 2015].

3.2.1 Measurements

Subjective Self-Disclosure

Participants were requested to report their level of perceived self-disclosure via the sub-scale of work and studies disclosure in Jourard's Self-Disclosure Questionnaire [Jourard, 1971]. This questionnaire was adapted and adjusted for the context of the study, addressing the statements to general life experiences. The measurement included ten self-reported items for which participants reported the extent to which they disclosed information to each of the agents on a scale of one (not at all) to seven (to a great extent). The scale was found to be reliable in Experiments 1, 2 and 3 when applied to all of the agents. In the second experiment, the reliability score of the scale when applied to the human agent was only moderate (see [Laban et al., 2021b] for reliability and mean scores of the scales).

Perceived Psychological Stress

This scale was added to the second and third experiments. Participants were requested to report their periodic stress in the past month on ten statement items of the perceived stress scale [Cohen et al., 1983], evaluating these on a scale of 1 (never) to five (very often). The scale was found to be reliable in both experiments (see [Laban et al., 2021b] for reliability and mean scores of the scales).

3.3 Feature Sets and Data Augmentation

We were interested in examining the effects that two different kinds of feature sets would have on the deep learning models that we used in our experiments. The first feature set chosen was log mel spetrograms and their cepstral coefficients. This data representation was chosen because representing inputs in log mel space has been shown to be an effective data representation for complex speech recognition tasks [Meng et al., 2019][Etienne et al., 2018]. Log mel spectra are two-dimensional representations of one-dimensional amplitude signals. These are produced by first applying a fast-Fourier transform to the signal using a sliding window (see Figure 3.3). The Fourier transformed windows, which are now in 2D, are then concatenated to produce a time-series of amplitude spectra in the Hz

domain. To produce mel-spectra, these time series are then transformed from the Hz domain into the mel-frequency domain, a log scale domain which matches the way in which humans perceive the distances between two pitches. We used the following standard equation to convert a frequency f to a mel-frequency m:

$$m = 2595\log_{10}\left(1 + \frac{f}{700}\right)$$

The cepstral coefficients are produced by taking a cosine-transform of the logs of the powers of the individual mel frequencies. To produce a singular feature set we then concatenated the log mel spectra with their associated cepstral coefficients. For our experiments we computed 128 mel-filter banks and applied them to the Fourier windows and then computed 20 cepstral coefficients resulting in a 148 dimensional feature space for our input data.

The second feature set we chose to investigate were so called "hand crafted" features from the Geneva Minimalistic Acoustic Parameter Set (eGeMAPS)[Eyben et al., 2016]. This is an acoustic feature set designed to avoid over fitting in machine learning models by not overwhelming the models with thousands of brute-forced features. eGeMAPS contains 88 statically computed low-level descriptors of an audio signal including frequency, amplitude, and spectral parameters. To create a time series of each WAV data point we took eGeMAP features of sliding windows of the amplitude data in 10ms segments. An example of these features can be seen in Table 3.1.

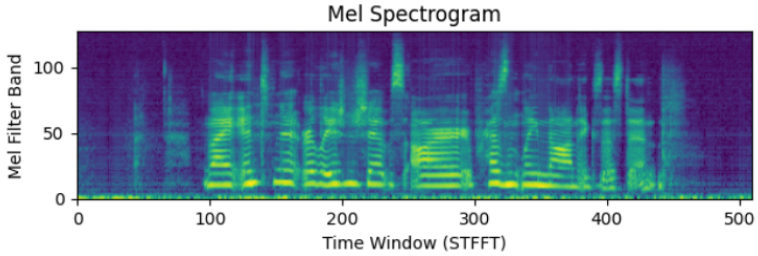


Figure 3.3: Example of a log mel spectrogram transformed from a one-dimensional amplitude signal.

The raw data from [Laban et al., 2021b][Laban et al., 2020] consisted of 625 interactions as waveform audio files. In the case of self-disclosure scores, the authors found that no participants rated their interactions as 7 on the self-disclosure scale so this class was removed, shifting the self-disclosure scale to 1-6. There was also a large degree of bias toward the central scores in the scale, meaning that a majority of participants scored their interactions in the range $[2, 5]$, creating a large degree of class imbalance. This is particularly problematic as the most underrepresented classes were the subjective self-disclosure scores of 1 and 6 i.e. interactions in which

participants were sharing very minimal personal information, or a great deal respectively. Since it is perhaps most important for an interactor to recognize when a person is maximally self-disclosing, our model should be proficient in detecting when this is the case and obviously this task becomes difficult if the the number of samples for the self-disclosure score of 6 is very small. In the stress detection case the authors found that participants very rarely rated their psychological stress at the lowest level i.e. deserving of a score of 1 and never found them so stressful that they warranted the maximum score of 5. Because the score-5 class was empty we decided to remove this as a possible classification option for our model thus reducing the size of the class space from 5 to 4. This did however mean that the score-1 class was very underrepresented.

To combat this class imbalance problem and produce a more balanced dataset in both the self-disclosure and stress detection problems we augmented the raw data in a number of different ways. Data augmentation is the technique of performing a specific set of transformations on a dataset in order to create new examples.The ways in which the datasets were balanced depended on the two feature sets that we considered for our experiments: log mel features and eGeMAPS. We now detail the augmentation techniques we used to balance the data in each case.

3.3.1 Log Mel Features

The first augmentation technique applied to the log mel version of the data was vocal-tract length perturbation [Jaitly and Hinton, 2013]. The length of a person's vocal tract is one of the key factors in determining the qualities of that person's voice. The intuition behind vocal tract length perturbation is that, if we can computationally mimic a shift in the length of the performer's vocal tract by transforming the data, then we will have a new example of that data point because it simulates the speech segment being uttered by a different person. This changes the quality of the voice in the data point without changing the underlying features of the data that we are aiming to capture in the model. Visually this has the effect of stretching the mel spectrogram slightly in the frequency domain and is similar to image warping techniques used in image classification tasks. We computed vocal tract length perturbation by shifting the central frequency of the mel filter banks (a visualisation of the filter banks can be seen in Figure 3.2) used to transform the data from the Hz domain to the log mel domain using a fixed warping coefficient and the following formula:

$$f' = \begin{cases} f\alpha & f \leq F_{hi}\frac{min(\alpha,1)}{\alpha} \\ S/2 - \frac{S/2 - F_{hi}min(\alpha,1)}{S/2 - F_{hi}\frac{min(\alpha,1)}{\alpha}}(S/2 - f) & otherwise \end{cases}$$

Where f refers to the starting frequency, f' is the transformed frequency, α is the fixed warping coefficient, and F_{hi} is a boundary frequency chosen in order to cover the significant formants in the signal. As in [Jaitly and Hinton, 2013] we set $F_{hi} = 4800$. While drawing the warping coefficient from a uniform distribution in a certain range is a common technique [Jaitly and Hinton, 2013][Kim et al., 2019] we found, in line with [Rebai et al., 2017], that choosing fixed warping coefficients of 0.9 and 1.1 produced the best results. This also allowed us to apply two separate perturbations to the data in the underrepresented classes.

3.3.2 eGeMAP features

Since the eGeMAP time series we produced from the original raw audio files don't naturally lend themselves to the same techniques for augmentation detailed above, we instead used weighted random sampling to ensure that the network was being trained on an even number of examples from each of the classes. Weighted random sampling, a development of sequential or uniform random sampling [Ahrens and Dieter, 1985], assigns a weight to each example in a training dataset where the weight is the reciprocal of the probability that that example would be chosen at random. This means that examples from underrepresented classes are more likely to be chosen in a batch of input data that is used to train a deep learning model. As a result, during one epoch of training, the model is shown proportionally fewer examples from the better represented classes, meaning that for each epoch, the model is trained on roughly similar amounts of examples from each class. Over the entirety of training then, since in normal circumstances the whole dataset is shown to the model every epoch, the model will see examples from the underrepresented class as much as it would without weighted random sampling, but will see fewer individual examples from the over represented classes. However, since we repeat the sampling of the dataset each epoch, random sampling algorithms will assure that every sample of every class is seen by the model at least once during training such that, in sum, the whole of the dataset is still being used over the course of model training. In general this has the effect of regularising network training such that a models performance is not biased towards one particular class. Not following these kinds of regularisation procedures can lead to over fitting, and indeed, in our initial experiments we found that networks overfit without the weighted random sampling. Additionally, we found that weighted random sampling in this way increased the stability of the learning procedure of our models when trained on eGeMAP features, despite recent work that has shown that, under specific assumptions about network architecture and learning algorithms, importance related sampling can have a limited positive effect on network training

[Byrd and Lipton, 2018]. eGeMAP features were extracted from the raw WAV files using the opensmile toolkit in python [Eyben et al., 2010].

3.3.3 N-class Classification Vs. Regression

In [Laban et al., 2021b, Laban et al., 2020] participants were asked to rate their degree of self-disclosure and psychological stress on a discrete scale. This raises a question about how best to frame the associated machine learning problem that we were interested in solving. On the one hand, separation of answers into discrete categories like this might suggest that the best way to frame the problem is as an n-class classification problem. In these problems each model is tasked with classifying a given input audio sample into one of the n classes associated with either the self-disclosure of psychological stress scores. In this case it would make sense to use something like a negative log likelihood loss function which would, in principal, force the model to learn a probability distribution over the n-classes. One potential issue with this approach is that it fails to capture the scaled nature of the class structure. Losses designed for n-class classification problems, in general, treat incorrect guesses in the same way. For instance, a model designed to classify different animals from photographs treats an incorrect guess of an elephant the same as an incorrect guess of a monkey (if the correct class was a tiger) because both are not tigers and neither elephants or monkeys are close to tigers. Similarly, framing our problem as a classification problem would mean that, if the correct score for a given input example was 4, a guess of a 3 and a 7 would be treated as equally incorrect and the same loss would be applied to each guess. This however misses something about the data. Namely, that both self disclosure and perceived psychological stress are scaled perceptions, meaning that an interaction that you rate as a 6 on a self disclosure scale will feel closer to an interaction that you rate a 7 than one that you rate a 2. As such we should want our loss function to represent that a guess of a 7, when the score of the given input is 6, is better than a guess of 3.

One way to capture this sense of scale in the input data is to frame the problem not as a n-class classification problem with discrete classes, but as a regression problem where the output of the model is some value $\in \mathbb{R}$. In this case it is typical to use a mean squared error loss function which will penalize a model's guess more severely the further away it is from the ground truth value. There are also draw backs with this approach, however, as least-squares approaches to inference assume that there is a correct value to predict, and that empirical data will distribute in a Gaussian fashion around this point. Because participants were asked to score

Loudness Mean	Loudness stdNorm	Loudness Percentile 20.0	Loudness Percentile 50.0	Loudness Percentile 80.0
0.2909	0.1079	0.2652	0.280	0.3098

Table 3.1: Example of 5 eGeMAP features for the first participant tested related to the loudness of the first 10ms window of the amplitude signal.

their interactions in a discrete way, we can be sure that the empirical data will not be Gaussianly distributed around some real value.

Both approaches have their respective merits and drawbacks which should be taken into account. In some sense, however, a discussion about which way to frame the problem will ultimately come down to which approach produces the best results. In light of this we decided to test both approaches and train on models on both regression and n-class classification problem sets.

3.4 Deep Learning Experiments

Our aim was to explore the efficacy of deep neural networks on the challenging task of learning non-linguistic features of an interactor's subjective self-disclosure and stress levels from their speech. In our experiments, we used five standard deep neural network architectures and one novel architecture that we designed to leverage the spatio-temporal nature of the input data space as well as make use of some key advances in time series modelling in the field of artificial neural networks over the past couple of years.

3.4.1 Neural Network Architectures

In this section we outline the architectures of the neural networks that we used in our experiments.

Linear Neural Network

Our linear network consisted of five fully connected layers where each hidden layer consisted of 1024 neurons. We applied drop-out and batch normalization to each layer to prevent over-fitting. Each layer was then passed through an ReLU non-linear activation function before its output was passed to the next layer. For the regression version of the problem, the output layer consisted of a single neuron. For the n-class classification problem the output layer consisted of a number of neurons equal to the number of classes relevant to that problem. The architecture of this stack of linear layers was also used as the classification stack in each one of the other networks that we used in our experiments.

Convolutional Neural Network

Convolutional neural networks [LeCun et al., 1989] have been used successfully in a number of tasks related to time series modelling [Kalchbrenner et al., 2014]. To test the efficacy of these architectures, we constructed a network with two one-dimensional convolutional layers and a linear stack for classification. The first convolutional layer passes a nx5 convolutional kernel with a stride of 1 over each data sample along the time dimension, where n is the number of features for each problem. The number of feature maps produced by this first layer was $\frac{t}{5}$ where t is the number of time steps in each sample fed to the network. This produced 35 feature maps for the log mel feature set and 15 for the eGeMAP feature set. Each of these feature maps was then fed through an ReLU non-linearity before being summarised by a 1D max pooling layer with a 3x3 kernel. The second convolutional layer contained 15 nx5 kernels with a stride of 1 and a max pooling layer with the same parameters as in the previous layer. Both layers also contained 1D batch normalisation to prevent overfitting. Finally the output of the second convolutional layer was fed to a linear classification stack that mirrors the structure of the linear neural network above.

Long Short-Term Memory Network

Long Short-Term Memory (LSTM) networks have been shown to produce state-of-the-art results on a number of time series problems including a number of audio classification tasks from emotion recognition [Schmitt and Schuller, 2018] to music genre classification[Dai et al., 2016]. For our experiments we used a simple single layer LSTM network with 296 LSTM cells. The output of this layer was then fed to a linear classification stack as above.

Convolutional Long Short-Term Memory Network

Convolutional Long Short-Term Memory Networks (CNNLSTM)[Sainath et al., 2015] utilize a hybrid-architecture whereby an input data point, usually a multi-variate time series, is fed through m either one-dimensional or two-dimensional convolutional layers supplemented with max pooling for averaging the features learned by the convolutional kernels and dropout for regularization. These feature maps are then fed through n long short-term memory layers to extract temporal features. Finally the data is fed through p fully connect linear layers and a softmax layer for classification. Our version of a CNNLSTM simply combines the three architectures above: The input is fed into a two-layer 1D convolutional stack then into a single LSTM layer before being fed into a linear classifier.

Hopfield Network

The limitation of CNNLSTM models is that their capacity to store temporally extended relations between points in data are limited by the LSTM layers. While LSTMs are partial solutions to the exploding/vanishing gradient [Pascanu et al., 2012] problems they still suffer from poor performance when faced with longer sequences. More recently attention based models were introduced for natural language processing tasks [Bahdanau et al., 2014] that improved on the base performance of LSTMs (and recurrent architectures more generally) by allowing the model to learn a vector embedding (a so called context vector) that teaches the model which parts of a sentence are relevant to which other parts. Since this time, attention based LSTM models have proven successful in a number of natural language processing tasks, such as sentiment classification [Wang et al., 2016][Yang et al., 2017] and emotion recognition [Xie et al., 2019]. A major breakthrough for sequence modelling tasks came with the introduction of the transformer architecture which showed that above benchmark performance on a number of natural language modelling tasks could be achieved using only stacked attention layers and a positional encoding that imparted latent temporal information in the input data [Vaswani et al., 2017, Brown et al., 2020]. The huge success of attention models for language based tasks has also lead to augmentations in the CNNLSTM architecture which introduce attention layers in conjunction with the LSTM layers in the original model [Miao et al., 2019][Zhang et al., 2019]. More recently, [Ramsauer et al., 2021] showed that the multi-head attention mechanism in transformer models is equivalent to an update rule in a modern continuous-state Hopfield network, a version of traditional Hopfield networks [Hopfield, 1982]. The authors also showed that Hopfield layers could be used as straightforward replacements to LSTM layers and included the added benefits of greatly increased convergence time as well as the ability to store patterns of much greater length. To test this we created a Hopfield network that simply replaced the LSTM layer in our LSTM model with a Hopfield layer. Since the Hopfield layer cannot encode temporal information from the data natively (as is done via the existence of recurrent connections between neurons in the hidden layers of recurrent neural networks such as LSTMs) we use positional encoding as in [Vaswani et al., 2017]. The positional encoding is a static matrix generated using the following formula:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/dmodel})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/dmodel})$$

where pos is the position, i is the dimension of the input (in our case this refers to the 128 filter bands from the mel-spectrograms or the 88 eGeMAP features), and d is the dimension of the model. The reason for using a static encoding over a learned encoding via embedding was two-fold. Firstly, testing showed that using a learned embedding had negative effects on the model's performance, and secondly as is noted in [Vaswani et al., 2017] the static positional encoding has the benefit that it generalizes to input lengths greater than that which the model was trained on. I.e. the same positional encoding could be applied to a longer sequence length after the model had been trained. This facilitates generalization to a wider set of data which is clearly of benefit to the wider aims of the project.

Hopfield Convolutional Neural Network

For the final model we designed a network architecture that combined the spatio-temporal representational power of hybrid networks like CNNLSTMs while aiming to improve on their performance by taking inspiration from the developments in the attention literature just outlined. Our model, which we call a Hopfield Convolutional Neural Network, replaces the LSTM layers in traditional CNNLSTMs with a Hopfield layer. In this model, we again simply replace the LSTM layer in our CNNLSTM architecture with a Hopfield layer and use a static positional encoding (as in [Vaswani et al., 2017]) to inform the Hopfield layer about the temporal position of each observation in each data point.

3.4.2 Experiments

We split the data for each problem into training and test datasets. The test set in each case contained participants that had not been seen by the model in the training phase so as to reflect the kinds of examples that it might see in a real world scenario. Test participants were selected such that the test set contained as even a balance of examples from the classes in each problem as possible and that the number of test to training samples that the model experienced during training was between 10% and 20%. The reason the train-test split was inconsistent was due to the fact that we split the training and test sets by participant. Each participant represented two or three interactions, regularly with different stress and self-disclosure scores and lengths of time. Therefore one participant might represent significantly more samples per class than another participant when the interactions were split into windows of a fixed length. Finally to ensure consistency in our comparison between models and between feature sets, the same training and test participants were used in each case.

Since each model needed to be fed with samples of consistent size we split the input data up into windows of constant length: 150 frames of data for log mel features and 75 frames for eGeMAP features as these were found to produce the best results for each problem. Each network was trained on mini batches of 200 samples (i.e. 200 windows of a given length) from the training data set over a period of 300 epochs for the log mel feature set and 100 epochs for the eGeMAP feature set. The differences in the epoch hyper-parameters were due to the speeds at which the networks tended to converge in each case. For each network we used the ADAM optimiser [Kingma and Ba, 2014]. Finally, we used a mean-squared error loss function for the regression problem and a negative log likelihood loss for the classification problem. [2]

We trained the architectures from section 4.1 on the log mel and eGeMAP feature sets separately. This was in order to explore how effective each of these literature-standard feature types were at capturing informative features from the data, for both the subjective self-disclosure and stress classification problems. Interestingly, we found that including the mel-frequency cepstral coefficients had a negative effect on the classification accuracy in the stress detection task. As a result, the models in this task were trained on the mel-spectrograms only. Each model was validated according to an accuracy metric defined as the percentage of correctly classified samples from a test set i.e. what percentage of examples from the test set the model correctly identified as belonging to a ground-truth stress or self-disclosure score. For the regression problem set we computed the classification accuracy by rounding the regression score for each input to the nearest integer. We then compared this result to the ground truth integer score when computing the accuracy of the input batch. For the n-class classification problem set we computed the accuracy in the standard way.

3.5 Results

The results of our experiments are displayed in tables Table 3.2, Table 3.3, Table 3.4, and Table 3.5. We found that all networks for both the self-disclosure and stress detection problems, for both the mel-spectrogram and eGeMAP feature sets, and in both the regression and n-class classification problem sets learned meaningful features from the data such that they were able to achieve accuracy scores significantly above chance. For each problem, we found that a different architecture achieved the highest results.

As a regression problem: In the self-disclosure task on the mel-spectrogram features, we found that all models performed effectively identically scoring around

[2] A table containing all network hyperparameters for both feature sets is displayed in Table 3.6

48% in each case, while for the eGeMAP features the linear net was the most accurate with a score of 43.52%.For the stress detection problem, we found that the LSTM model scored highest on the mel-spectrogram feature set (59.35%), while in the eGeMAP the CNNLSTM model performed best at 53.86% accuracy. We further found that, for both the self-disclosure and stress detection problems, log-mel features were the most informative, leading to significantly better accuracy scores than the eGeMAP features.

For the n-class classification problem: in the case of self-disclosure, both the HopfieldCNN and the CNNLSTM scored best on the mel-spectrogram feature set (with 48.04% accuracy each) while the HopfieldCNN scored best on the eGeMAP features with (35.48%). For the psychological stress problem the HopfieldCNN scored highest on the mel-spectrogram features with 58.61%. On the eGeMAP features, the Hopfield network scored highest with 51.85%. As with the regression version of the problem, we found that mel-spectrograms was the most useful way to represent the input data.

Taken together, these results suggest that for the self-disclosure problem both a regression framing and a n-class classification framing with mel-spectrogram features are equally useful and that many models are plausibly well placed to tackle this problem given that a ceiling score of around 48% was achieved by all but four of the different models. Similarly for the psychological stress problem, both regression framing and n-class classification framing with mel-spectrogram features produced similar best results with the regression framed LSTM and the n-class classification HopfieldCNN model achieving around 59% accuracy.

For both the mel-spectrogram and the eGeMAP features in both the self disclosure and stress detection tasks, and in both the regression and n-class classification problem sets, we found that the networks tended to overfit the training data. To combat this we set the network dropout values to 10% for the mel-spectrogram features and 90% for the eGeMAP features to account for the degree of over fitting that we experienced in both cases. None-the-less we found that learning in all networks was difficult, as is clear from the results. We hypothesize that the failure to achieve much higher accuracy scores may be able to be put down to the difficulty of the task. It's intuitive that an important way in which we ascertain whether someone is disclosing personal information or is stressed is informed in no small part by the lexical properties of their speech i.e. what it is they are saying as opposed to how they are saying it. Since lexical features were absent from the feature sets in both cases (since we were specifically interested in investigating whether networks could learn non-lexical properties of speech for both problems) it makes sense that both of the tasks would be significantly harder than if we had included lexical based features. However it is clear from the results that the data

were informative enough to allow the networks to lean non-lexical features despite the intuitively challenging nature of the task.

Regression Problem		
Model Type	Mel-Spec Accuracy(%)	eGeMAPS Accuracy(%)
Chance	16.67	16.67
LNN	48.2	43.52
CNN	48.28	42.42
LSTM	48.34	41.05
CNNLSTM	48.13	40.08
Hopfield	47.8	42.74
HopfieldCNN	48.28	42.85

Table 3.2: Self-Disclosure Model Accuracy for Mel-Spectrogram and eGeMAPS feature sets framed as a regression problem.

Regression Problem		
Model Type	Mel-Spec Accuracy (%)	eGeMAPS Accuracy (%)
Chance	25	25
LNN	38.8	44.44
CNN	48.48	48.15
LSTM	59.35	42.59
CNNLSTM	53.86	48.15
Hopfield	41.45	46.44
HopfieldCNN	47.28	44.44

Table 3.3: Stress Detection Model Accuracy for Mel-Spectrogram and eGeMAPS feature sets framed as a regression problem.

Classification Problem		
Model Type	Mel-Spec Accuracy(%)	eGeMAPS Accuracy(%)
Chance	16.67	16.67
LNN	29.34	22.53
CNN	36.63	33.38
LSTM	36.82	26.77
CNNLSTM	48.04	34.21
Hopfield	32.02	33.74
HopfieldCNN	48.04	35.48

Table 3.4: Self-Disclosure Model Accuracy for Mel-Spectrogram and eGeMAPS feature sets framed as a classification problem.

Classification Problem		
Model Type	Mel-Spec Accuracy (%)	eGeMAPS Accuracy (%)
Chance	25	25
LNN	42.88	40.74
CNN	46.93	46.3
LSTM	35.5	35.19
CNNLSTM	38.03	40.74
Hopfield	37.84	51.85
HopfieldCNN	58.61	46.63

Table 3.5: Stress Detection Model Accuracy for Mel-Spectrogram and eGeMAPS feature sets framed as a classification problem.

Hyperparameter	eGeMAP Models (%)	MelSpec Models (%)
Learning Rate	0.1	0.1
Epochs	100	300
Input Size (frames)	75	150
Batch Size	200	200
Dropout	0.9	0.1

Table 3.6: Network Hyperparameters

3.6 General Discussion

An overview of these results show that networks that are able to model temporal relations between the data generally do the best on both tasks. The LSTM, CNNLSTM, Hopfield, and Hopfield CNNLSTM models all contain methods by which temporal relations can be modeled and, with the exception of one experimental condition (self-disclosure task using eGeMAPS features as a regression problem) these networks performed best. LSTM subnetworks contain recurrent connections between neurons (see section 4 of chapter 2 for more detail) which incorporate a representation of a data point further back in time with a representation of a data point at the current moment in time. This bestows the network with a kind of memory that allows it to relate representations of datapoints over a given interval of time. Hopfield networks use a positional encoding matrix (see section 6.3 chapter 2 for more details) to represent the passage of time in a given datapoint and the networks are able to memorize temporal patterns, in part, due this encoding. Given this representational capacity, it is not surprising that these networks performed best overall given that the nature of the problem in each case was the classification of time series data. Interestingly however there was little consistency with regard to which network was best. For the self-disclosure problem, scores of around 48% were achieved by most models with no model having a clear advantage over any of the other top scorers. In the psychological stress task however, the LSTM and HopfieldCNN models had a clear advantage, both scoring around 59% accuracy, a 10 percentage point increase from the second most successful models. These results could in part be explained by the relative complexity of each task. It may be the case that signals of psychological stress are more obvious than those of subjective self-disclosure and therefore less data is required for models to achieve relatively high accuracy rates. An extended hypothesis from this observation would be that, in order to properly model the signal of subjective self disclosure more data and more sophisticated modeling approaches may be required. This variation in model performance leaves the question of which model approach is best relatively unanswered. One conclusion that can be taken away however, is that it is likely that effective models for these problems should involve

an ability to effectively model temporal data (i.e. using some kind of recurrent architecture or positional encoding).

There is no doubt that the results that we present here are insufficient to achieve the goal of implementing a self-disclosure recognition capability in a social robot. If this goal is to truly be achieved it is sensible to claim that classification accuracy of much larger than 59% will be necessary. The reasons for these limited results are likely to be two-fold. First, and most significantly, while the amount of data that was collected was sufficient to show that neural networks have the potential to perform well on this task, it was unlikely to be enough to truly reach state-of-the art performance. As a comparison, [Soleymani et al., 2019] showed that self-disclosure can be very effectively modelled between two human subjects but with collated datasets that comprised over 200 participants. This result would suggest that much better performance in our HRI version of the task might be possible with further data collection. Secondly, it is likely that our networks did not house significant enough representational capacity for such hard problems. Evidence of this can be seen when comparing our models to state-of-the-art speech and language recognition systems (with numbers of parameters in the order of millions rather than hundreds of millions or billions) that utilise large pretrained networks on top of smaller customer trained ones. Going forward, it would make sense to both collect more data and also to use model training techniques that allow our models to represent complex problems using other data sources (such as with transfer learning). Finally, we looked only at the non-lexical speech component of both self-disclosure and perceived psychological stress. Both of these behaviours are likely to realise themselves through a number of behavioural modalities from speech, to facial changes, and less course-grained body language. Indeed, [Soleymani et al., 2019] found that the lexical aspects of speech and visual components where important parts of being able to successful classify instances of self-disclosure. As such, it is likely that our focus on only non-lexical aspects of speech may have limited our models' ability to perform at top level on these tasks.

3.7 Future Work and Improvements

Since the long-term goal of this field of research is to implement autonomous computational social agents in real-world environments, it is absolutely essential that the quality of the models used in such agents is optimised. This is especially true in the context of care and mental health, where ineffective communication with people can be especially damaging and ethically complex [Murphy et al., 2021]. In light of this, we emphasize that the current results provide only a modest step

toward realizing this goal. With this in mind, there are a number of areas in which the current findings can be improved upon.

Firstly, we did no significant hyper-parameter tuning. Network tuning techniques like Bayesian optimisation [Snoek et al., 2012] and random search [Bergstra and Bengio, 2012] have regularly been shown to improve the performance of deep neural networks, and are both plausible candidates for improving the performances of the networks we used in our experiments.

Second, as mentioned above, we believe that adding lexical features to the models' input would likely improve the results by some degree. Natural language processing is one of the most active fields of research in deep learning and affective computing, and the number of significant developments in recent years that have lead to the production of networks like BERT[Devlin et al., 2019] and GPT3[Brown et al., 2020] might well hold significant benefits for the interests of this project. Indeed, efforts to model non-subjective self-disclosure in the human–human context have shown that the addition of lexical features that are extracted using BERT has a beneficial effect on the performance of these models [Soleymani et al., 2019].

Third, the task of learning non-lexical features of speech is not new. Significant results using deep learning have been shown in learning acoustic features of people's voices for problems like emotion recognition [Hossain and Muhammad, 2019], deception detection[Mendels et al., 2017], and speech intention [Gu et al., 2017]. As such, a number of networks trained on large acoustic datasets could be used to explore transfer learning. This involves taking networks that have learned features from similar datasets, fixing the weights of the lower layers and then learning new weights for higher layers by training the model on the new dataset. In principle, this would allow acoustic features perhaps not present in any of the DAVISS models to be used to improve the performance of those models. While we have no empirical evidence to say for sure that this technique would improve the accuracy of the networks we used, it is certainly one potential fruitful avenue for development.

Fourth, stress and self-disclosure are likely to be multi-faceted phenomena, in that their expression could well be realised not just in speech, but in other modalities such as hand and body gestures, pose, facial expression, heart rate, galvanic skin response, and gaze cues, plausibly among many others. Thus, "human" or "above human" levels of recognition of both psychological stress and self disclosure might well only be possible with a model that has been trained on a much broader set of features that takes into account many (or all) of these complex modalities. However, considering the necessity for non-intrusive sensing technologies for artificial agents to understand and read people's subjective perceptions, we believe

that features like voice should be prioritized by virtue of the ability to collect these data in a minimally invasive manner.

Fifth, further research into this area might look into an analysis of the networks to provide, for instance, a more detailed understanding of which features contribute most significantly to the success of each model. Feature ranking algorithms can be applied to deep learning models in order to make the results more interpretable, and could thus elucidate, for instance, which eGeMAP features contribute the most to successful classification of self-disclosure or perceived psychological stress [Wojtas and Chen, 2020]. Further, algorithms to interpret layer outputs could be applied to determine which regions of the input mel-spectrograms were being passed to successive layers in CNN networks. Techniques such as those used in [Yu et al., 2016] and outlined in [Samek et al., 2017] can be useful in helping researchers understand how deep neural networks are interpreting input data and which parts of the input are seen as important to the task at hand. A further analysis could also compare how well these networks perform in comparison to average, or specialist, human subjects. A follow up experiment could ask a group of participants to rank the same interactions according to their perceived levels of stress and self-disclosure and then compare those results to trained neural network models. This would have the advantage of providing a more accurate insight into how good a particular model is at these tasks. It would be interesting to see how good a particular model performance was in comparison to these results. In that case we would be able to say with more assurance whether a score of 59.35% was good for the task or not. Such an analysis however is outside of the scope of this chapter as our principal aim was to determine whether these kinds of neural networks were simply able to perform these tasks in a way to warranted further investigation rather than how it was that they were performing said tasks.

Sixth, a comparison might be made between the tasks that we have investigated here, namely, classifying instances of subjective-self disclosure and perceived psychological stress, and the task of emotion recognition. Both of these fields involve the analysis of a number of behavioural modalities including lexical and non-lexical components of speech, and stress and self-disclosure can intuitively be seen as related to human emotions like happiness, sadness, or anger. What is more, deep learning approaches have been widely successful in the task of classifying different emotions [Kahou et al., 2016], even under the constraints of having relatively small datasets [Ng et al., 2015]. This may raise the question of how our approach differs from those used in the emotion recognition literature and how similar the tasks are. Since, in this paper, we investigate a large number of neural network architectures there is a large degree of overlap between some of the networks that we used and those used in the emotion recognition literature.

For example, [Wang et al., 2020] use variations of LSTM models for recognising emotion in speech segments, [Zhao et al., 2019] use 1D and 2D CNNLSTMs for a similar set of tasks, while [Huang et al., 2014] use CNNs and 2D spectrograms as model input for the same kind of problem. While there are similarities in these techniques, all of these papers present more sophisticated adaptations of these basic architectures in order to achieve their state-of-the-art results. In [Huang et al., 2014], for instance, the authors use a two-step training procedure on two linked CNN models and a custom loss function that was tailored to the specific task of emotion recognition. In our experiments we used standard implementations of these architectures with no custom functionality or loss functions. One potential hypothesis, if we take it as granted that all of these tasks are indeed similar in some important respects, is that our results were limited by the simplicity of our methodology. Follow on experiments to push our results further should look at implementing network training techniques and architectures that have proven successful in the field of emotion recognition. However, one important difference between our problem and that of emotion recognition is in the nature of the classification task. Many deep learning related emotion recognition problems relate to classifying different emotions such as happiness, sadness, contentment, boredom, and so on. On the other hand, our task involves recognising different degrees of the same emotion or behavioural affect. Succinctly, emotion recognition tasks can be seen as attempting to classify different *kinds* of things, whereas the tasks that we are interested in can be seen as classifying different *degrees* of the same thing. One hypothesis that might explain the difference between our results of those of the emotion recognition literature may then be that differentiating between emotions is easier due to the fact that each class has quite different behavioural features - smiling, in the case of happiness rather than frowning in the case of anger or sadness, for example. On the other hand, the differences between the classes in our case may be far more subtle. It is not hard to imagine that some important feature of one's speech that is present in a level 3 instance of self-disclosure or perceived stress is not hugely different from how that feature manifests itself in a level 2 or 4 interaction. In light of this, a further study might look into training the same model using the same data on the tasks of emotion recognition and those of subjective self-disclosure and perceived psychological stress. These models might then be analysed to determine how similar the tasks are with respects to how well the models perform, and also what aspects of their input data they deem to be most important to each task.

Finally, as we discussed in the introduction, this study does not consider ways in which subjective self-disclosure and psychological stress may be related and, indeed, empirical work has shown that these two factors may well be correlated in

set of social contexts [Han and Yu, 2012][Hamid, 2000]. To test this hypothesis, we ran a Pearson's correlation analysis between the self-disclosure and psychological stress scores that each participant assigned to their interactions. Further, we tested to see whether a linear regression model trained on these scores would corroborate the existence of such a correlation. The results of the regression model can be seen in Figure 3.4.

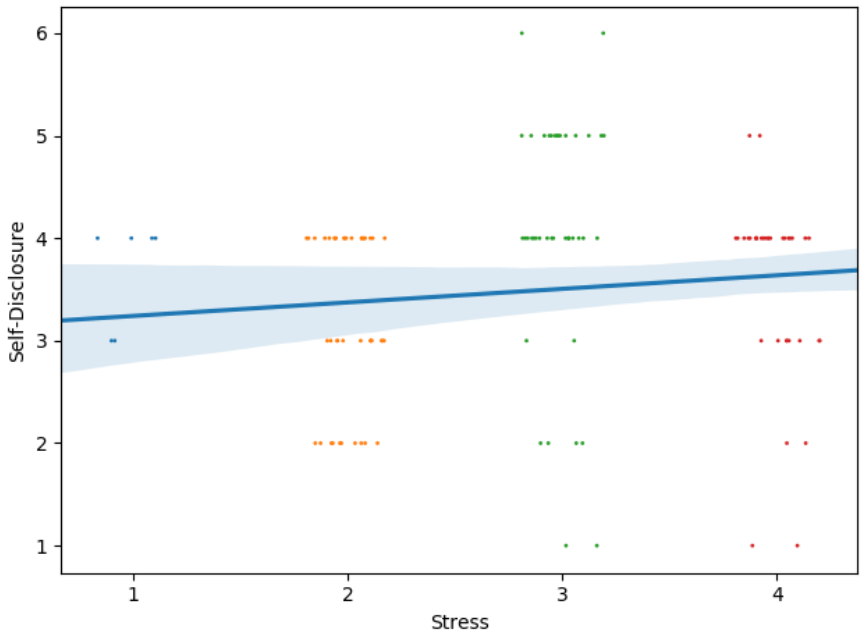


Figure 3.4: Linear regression plot exploring the interaction between self-disclosure and psychological stress factors. Data points are jitterd to show density of score assignments.

Our correlation analysis showed a minor positive Pearson's correlation of $r = .12$. Likewise, our regression model produced a line of fit that was marginally positively inclined. These results, taken with the empirical work in this area, suggest these two factors may well effect one another, albeit perhaps not in a significant way. Nonetheless, we take it that this suggests that a potentially fruitful avenue for this work would be to investigate how the interaction of these two factors affects the performance of our models. For instance, audio features learned by the psychological stress model could be combined with the input features in the self-disclosure model to see if these features helped the model in the self-disclosure task. Likewise, a learned embedding of the self-disclosure features could be used to try to improve the performance of our models in the psychological stress task. We plan to investigate both of these in future studies.

3.8 Conclusion

The aim of the current study was to experimentally validate the efficacy of deep learning models for classifying a person's subjective self-perceptions: specifically, their self-reported levels of periodic psychological stress and self-disclosure. We approached the problem in a data-centric manner, i.e., by applying robust experimental design methodology and prioritising the collection of high quality data over complex computational models. The data collection procedures for the raw data used in the present study allowed for maximal control over the robustness and quality of the data collected. Further, allowing participants to rate their own interactions carried the benefit that each sample was accurately labelled. That is, it is highly unlikely (if not impossible) that a person can be wrong about how stressed they felt or how much personal information they felt they were sharing.

To probe this problem, we investigated the effectiveness of two different standard feature sets from the speech recognition literature: log mel and eGeMAP features. We also tested how the performance of the models varied when we framed the problem as one of n-class classification and of regression. Our experiments were conducted using five standard neural network architectures (LNN, CNN, LSTM, CNNLSTM, and a Hopfield network) as well as a novel architecture that replaced the LSTM layer in a CNNLSTM with a Hopfield layer (what we have called a HopfieldCNN). Our findings suggest that even relatively non-complex deep learning models such as these were able to learn informative features from both the log mel and eGeMAP input data spaces and in both the regression and the n-class classification versions of the task.

This study provides novel scientific and technical contributions to the affective computing and human-computer interaction research communities in a number of ways. To our knowledge, this is the first attempt to investigate deep learning's ability to extract features related to a person's subjective experience from their speech in human–robot interactions. We also contribute a large dataset consisting of 625 interactions recorded with high quality data capture devices in both the eGeMAP and mel-spectrogram versions of the raw audio data. Finally, we detail a novel deep learning architecture that is competitive with other highly popular and successful models archetypes. By studying the connection between subjective perceptions and non-intrusive behavioural mechanisms that imply physiological reactions (i.e., the human voice for the purposes of this study), we can further learn about relationships between cognition and biological markers, physiology, and behavioral outcomes. While the presented architectures are relatively straight-forward, these models (and other similar models that use the same approach) conceptually mark small steps towards creating machines and agents that understand people from

their subjective point of view by synthesizing available non-intrusive behavioral cues. Adapting the architectures presented here could help equip artificial agents to understand humans better, and accordingly, enable the creation of more effective tools and agents for delivering psychosocial and health interventions.

Our results, however, do show that much improvement can and should be made before deep learning platforms are seriously considered as tools to assist carers/ people (see [Petrovic and Gaggioli, 2020]) in the context of care-giving, and in terms of being introduced or implemented into autonomous cognitive agents (such as robots or voice assistants). We do, however, believe that our results show that such progress is possible and that there are promising avenues for future research in this space.

Chapter 4

Multimodal Deep Learning of Subjective Self-Disclosure in Human-Robot Interactions

Henry Powell

Guy Laban

Emily S. Cross

[1]A version of this chapter is currently in preparation for submission to *IEEE Transactions on Affective Computing.*

Abstract

Subjective self-disclosure is an important and relatively well understood feature of human social interaction. While much has been done in the psychological and neuroscientific literature to characterise the features and consequences of subjective self-disclosure, little work has been done thus far to develop computational systems that are able to accurately model it. Indeed, even less work has been done that attempts to model specifically how human interactors self-disclose with embodied robotic partners. A need to do just this will become more pressing as we require social robots and other socially oriented computer systems to work in conjunction with, and in some instances take the place of, humans in various social roles. In this paper our aim is to improve and build upon previous work in this area by developing a custom multi modal attention network based on models from the emotion recognition literature, training this model on a large self-collected self-disclosure video corpus, and constructing a new loss function, the scale preserving cross entropy loss, that improves upon both classification and regression versions of this problem. Our results show that the best performing model, trained with our novel loss function, achieves an F1 score of 0.83, an improvement of 0.48 from the best baseline model. Building on our previous work, this result makes significant headway in the aim of allowing social robots to pick up on an interaction partner's self-disclosures, an ability that will be essential in social robots with truly human-like socially cognitive abilities.

4.1 Introduction

4.1.1 Subjective Self-Disclosure

self-disclosure is usually thought to be the sharing of one's thoughts, feelings, or sensitive personal information during a social interaction. It is an import facet of human sociality and can contribute to many aspects of our lives. As discussed in [Kreiner and Levi-Belz, 2019], self-disclosure can contribute to the extent to which we form bonds with one another - i.e. how intimate and important we consider our relationship with others to be - as well as contributing significantly to our mental and physical health [Jourard, 1971][Jourard and Lasakow, 1958]. In what follows, we focus specifically on subjective self-disclosure, which picks out the degree to which one *believes* themselves to be sharing personal information. For example, it may be the case that someone shares some information which may, in general, not be perceived to be particularly personal or sensitive. However, this information might well be sensitive or important to the person disclosing it. Here the term subjective is supposed to clarify that what is important in an act of self-disclosure is that the person performing it believes themselves to be sharing sensitive thoughts, feelings, or personal information.

Given how important subjective self-disclosure can be to the development of meaningful personal relationships, it seems uncontroversial to say the following: If we aim to develop social robots and socially oriented computing systems that function alongside, and in place of, real human interactors, then being sensitive to such self-disclosures would be an important feature of such systems. Despite this, there is very little, if any, work in the field of human–robot interaction that seeks to model the ability to detect and measure self-disclosure with the aim bestowing a socially oriented computer system with this ability.

In this study, we aim to address this problem and to improve on our results in [Powell et al., 2022] where we found that a number of standard deep learning architectures were able to perform well above average on the task of ranking the degree of self-disclosure in recorded interactions. We did this in a number of ways. Firstly, by developing a significantly larger data set that included an visual as well as an audio modality in order to capture markers of subjective self-disclosure that may be present in how facial features evolve over time. Secondly, by developing a more sophisticated deep learning model that was better suited to the task i.e. one that was developed using domain knowledge of the problem and the data representations we used as input to our model. Thirdly, to address the problem of experimental framing that we experienced in that study, i.e. how to model data that was both categorical and scaled.

4.1.2 The Current Paper

The remainder of the paper takes the following form: in section Section 4.2 we detail the design, data collection and data pre-processing for the experiment that we conducted in order to form the dataset used to perform our deep learning experiments. Next, in section Section 4.3 we outline which features we extracted from the processed dataset and the means by which we extracted them. In section Section 4.4, we describe the architecture of our multi modal attention network in detail. We then describe the experiments we conducted to produce baselines to which we could compare the performance of this model. Further, we detail the parameters of our ablation experiment to test the effects of the loss functions, feature sets, and experimental framings that we used, and finally, the specific details of the training implementation. Then, in section Section 4.5 we present the results of the ablation study before finally, in section Section 4.6, discussing some areas for further improvement to our approach and some issues with it.

4.1.3 Our Contribution

Our contributions to the field of human–robot interaction (HRI) research are as follows:

1. We present the most extensive attempt to model subjective self-disclosure in human–robot interaction so far,
2. We provide, to date, the largest dataset specifically designed for the problem of self-disclosure modelling in HRI,
3. A multi modal attention based architecture designed specifically for self-disclosure modelling from audio and video data,
4. A novel loss function, the scale preserving cross entropy loss, that effectively deals with problems that fall between regression and classification and outperforms both squared error and cross entropy approaches to self-disclosure modelling.

4.2 Data Set and Data Collection

In order to generate data for the models, a long-term mediated online experiment was conducted, as reported in [Laban et al., 2021c]. We repeat that protocol here verbatim for consistency: A 2 (Discussion Theme: COVID-19 related vs. general) by 10 (chat sessions across time) between-groups repeated measures experimental

design was followed. Participants were randomly assigned to one of the two discussion topic groups, according to which they conversed with the robot Pepper (SoftBank Robotics) via Zoom video chats about general everyday topics (e.g., social relationships, work-life balance, health and well-being). One group's conversation topics were framed within the context of the COVID-19 pandemic (e.g., social relationships during the pandemic, sustaining mental health during the pandemic, etc.), whereas the other group's conversation topics were similar, except that no explicit mention of the COVID-19 pandemic was ever made. Participants were scheduled to interact with the robot twice a week during prearranged times for five weeks of participation, resulting in 10 interactions in total. Each interaction consisted of the robot asking the participant 3 questions (x3 repetitions), starting with a generic question to build rapport (e.g., how was your week/weekend), followed by two additional questions that corresponded to one of the 10 randomly ordered topics (for the topics, questions, and examples see [Laban et al., 2021c]). The topic of each interaction was assigned randomly before the experimental procedure started, as was the order of the questions. After conversing with Pepper via the zoom chat, participants filled a questionnaire reporting for their perceptions of their subjective disclosure via an adapted version of Jourad self-disclosure questionnaire [Jourard, 1971]. The zoom chats were recorded for analysis purposes. Each interaction with the robot lasted between 5 to 10 minutes, and another 10-20 minutes were taken up completing questionnaires. This lead to $40 \times 10 = 400$ interactions each comprising of at least 3 conversational segments that we were able to use to train our models. Due to participant drop out and some issues with poor recording conditions (obscured faces, videos being too dark, poor audio quality, or issues related to bad internet connections) this actual figure was 391.

Once the dataset was collected the videos were segmented by hand to isolate the sections that contained only the participants' speech. Most videos contained three speech segments comprised of the participants' answers to each of Pepper's questions. However, some participants followed up on Pepper's responses to their answers resulting in a number of additional speech segments that we were able to add to the corpus. Each of the segments was then labelled by an experimenter in accordance to the self-disclosure score that each participant had assigned to their respective interaction instances. This lead to a total of 1,248 speech and video segments that were used in our deep learning experiments.

4.3 Feature Extraction

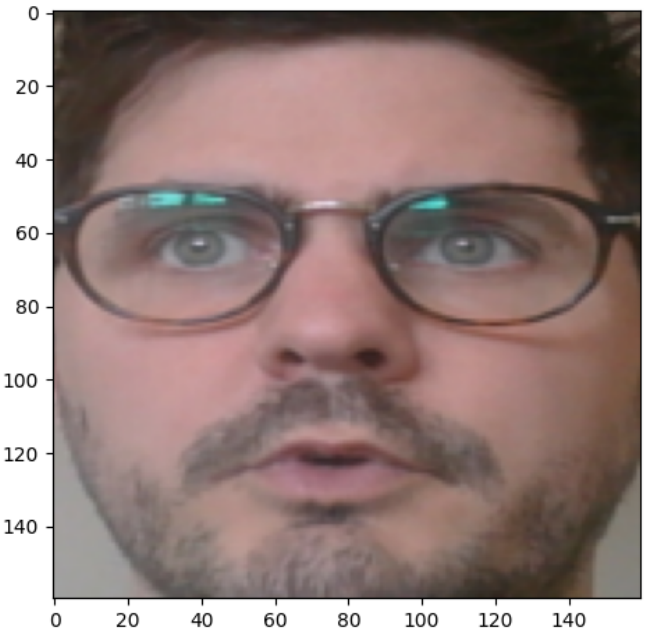
4.3.1 Visual Features

We extracted a number of visual feature types using a combination of state-of-the art feature extraction models. First, we extracted frame-by-frame gaze and action unit features using the OpenFace 2.2 library [Baltrusaitis et al., 2018] (see Figure 4.1b for visual example). To account for missing frames in each time series that came about as a result of the OpenFace models not registering the presence of a human face, we interpolated the missing frames with the recorded data using spline interpolation. We then filtered and smoothed the resulting multivariate time series with a Savitsky-Golay filter (using a sliding window of 11 frames and a polynomial order of 3). To test the the affects of smoothing and filtering on the results we treated smoothed/filtered and non-smoothed/filtered feature sets as separate in our initial experiments.

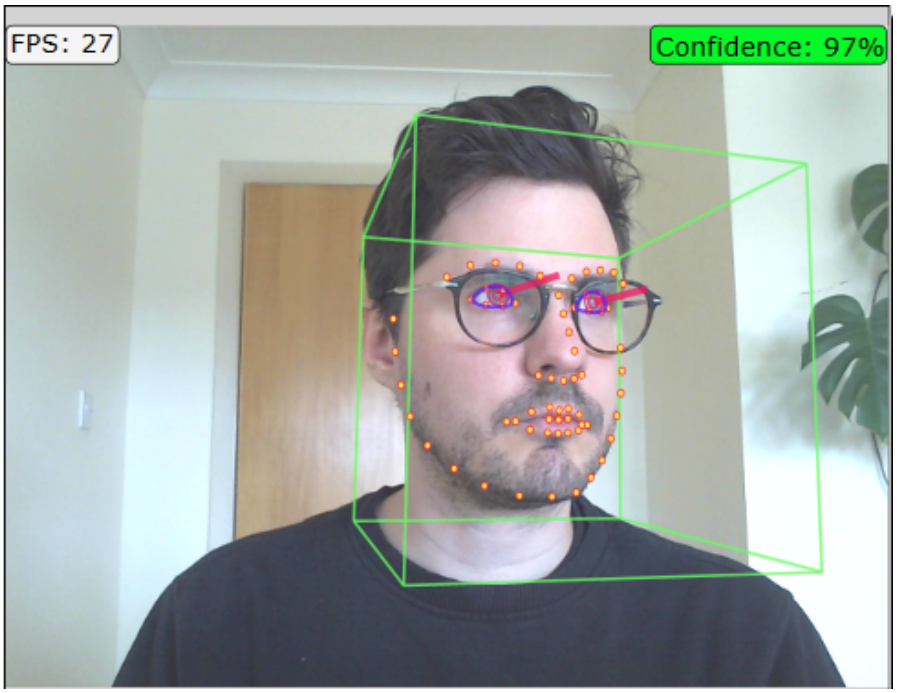
Next, we extracted facial features using an InceptionV1 ResNet [Szegedy et al., 2015][He et al., 2016] architecture pretrained on the VGGFace2 dataset [Cao et al., 2017]. VGGFace2 consists of 3.31 million images of celebrity faces organized into 9131 subject categories with large variances in pose, age, illumination, and ethnicity. The InceptionV1 ResNet that we used scored an accuracy of 99.6% on this dataset. Pre-processing of the video frames in this case consisted of extracting a 160x160 pixel sub-region of each frame that contained pixel and feature-wise normalization of the subject's face (an example of the MTCNN output can be seen in Figure 4.1a. This was done using a pretrained multi-task cascaded convolutional neural network [Zhang et al., 2016] on each video frame. The pretrained ResNet produces 512 facial features for each video frame. Similar to our approach with the the OpenFace features we interpolated and filtered the resulting time series to experiment with the effects that this would have the models' scores.

4.3.2 Audio Features

For audio features we first produced a mel-filter cepstral coefficient (MFCC) matrix for each video's audio modality. This was done using PyTorch's audio feature extraction library using 256 mel-filter banks. This feature set was chosen due MFCCs well established ability to capture significant audio features for human speech recognition tasks [Yang et al., 2020a][Pawar and Kokate, 2021][Kumaran et al., 2021]. This was a departure from our previous work on using log-mel spectrograms to recognize subjective self-disclosures [Powell et al., 2022], where we found that spectrogram features were more effective at capturing significant self-disclosure related features from subjects' speech. In the case of the current



(a) Example output of the MTCNN used for facial feature extraction: a 160x160 pixel normalised face image.



(b) OpenFace 2.2 processing facial action units, and gaze from an input video.

study we found that MFCC features produced better results at initial testing and it is for this reason that we went with MFCC features over the log mel spectrogram alternative. We also experimented with the effects of cepstral mean and variance normalization of MFCC features on our baseline models' performance (detailed in Section 4.4.1 as this was a factor that would also have to be taken into account when training our deep learning models.

Second, we extracted audio features directly from each sound file's amplitude array using Facebook AIs wav2vec2.0 architecure [Baevski et al., 2020]. Wav2vec2.0 uses a stack of convolutional neural network based feature encoders and generates contextualised audio representations using a transformer model [Vaswani et al., 2017]. We used a wav2vec2.0 model pretrained on 960 hours of unlabelled audio data from the LibriSpeech dataset [Panayotov et al., 2015]. To get the feature sets for each wav file we took the outputs from the models 12 transformer layers which resulted in 12 t x 768 feature matrices where the value t was determined by the number of frames in the audio file.

4.4 Deep Learning Experiments

4.4.1 Support Vector Machine Baselines

Since we were working with a novel dataset designed specifically for our deep learning experiments we needed some way of establishing a baseline that we were able to compare our results to. Following [Lin et al., 2021] we used Gaussian kernel support vector machines (SVM) trained on our extracted audio and visual features separately to establish such a baseline. For each feature type, a vector representing the mean over all frames in each example was computed and the SVMs were tasked with classifying the self-disclosure score for each interaction. Each model was trained using 3 fold cross validation and the average f1 score was used as a means to measure the overall performance of each model.

The results of these baseline experiments (illustrated in Section 4.4.1) indicate that the facial features extracted using InceptionV1 pretrained on VGGFace2 were significantly the most informative for the task while for the audio features, the MFCC representation was the most informative. Overall video features were the most useful feature sets in discrmiitinating the self-disclosure score classes. The results also show that the problem is a difficult one given that the best f1 score measured was only 0.36. One surprising result was that the word2vec2.0 features performed so poorly. We hypothesised that, given the strong relative performance of the InvceptionV1 features, that word2vec2.0 would also perform relatively well given that both models are pretrained on large amounts of task relevant data.

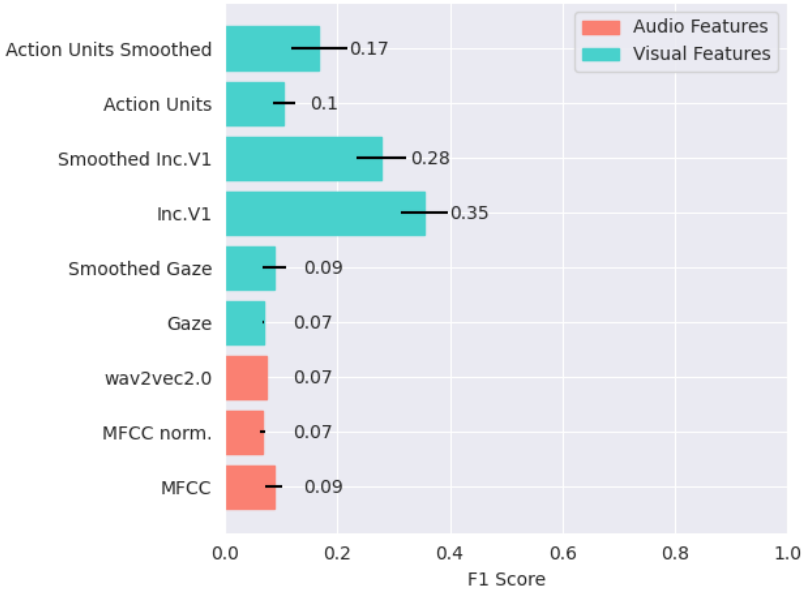


Figure 4.2: Gaussian SVM baseline F1 scores for individual smoothed/filtered and unsmoothed/unfiltered audio and visual feature sets. Standard deviation is represented by black error bars.

One possible explanation of why word2vec2.0 features performed so poorly in the baseline test is with respect to how the mean vector for each frame was computed. The word2vec2.0 features for each frame were of far higher dimension than both the MFCC features ($12 \times t^W \times 178$ vs. $256 \times t^M$) and the visual feature set of the highest dimensionality (InceptionV1 features at $t^I \times 512$) [2]. Thus condensing the word2vec2.0 features across both the time and attention-head dimensions into a single 178 dimensional vector could have meant that too much information was lost leading to the feature dramatically losing its discriminative ability with respect to the task.

4.4.2 Multimodal Attention Network

[H]

Extending the work we conducted in [Powell et al., 2022] we designed a multimodal attention network that processes the audio and visual features of each video in separate streams and then combines these representations in a late fusion fashion before being classified by a linear neural network layer. This approach was motivated by our observations in [Powell et al., 2022] that concluded that 'off the shelf' neural network architectures, i.e. ones that were not designed specifically for the task at hand and used no pretraining, produced less than desirable results

[2] Here t^W, t^M, and t^I refer to the time dimension of the word2vec2.0 features, the MFCC features, and the InceptionV1 features respectively.

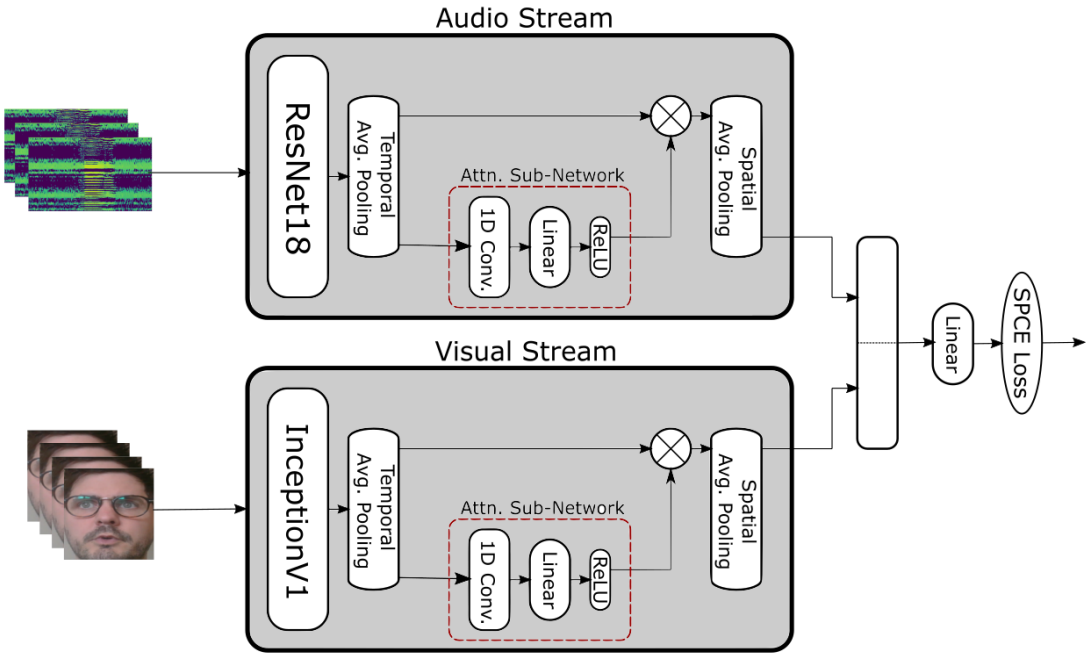


Figure 4.3: Illustration of our multi-modal attention network. Segments of MFCC matrices (top) and face-cropped video frames (bot) are fed into two similar streams. MFCC segments are fed through an ImageNet pretrained 2DResNet backbone before being average pooled, and cloned. One copy is then sent through the attention subnetwork before being multiplied to the other ResNet output copy. This representation is then average pooled once again producing the final audio embedding. The same process occurs with the frame input except that the backbone is a InceptionV1 ResNet architecture pretrained on VGGFace2. The resulting audio and visual embeddings are then concatenated and fed through a linear classification layer. The network probabilities are then used to compute the scale-preserving cross entropy loss by which the parameters of the network are optimised.

on the audio-only version of this task. We aimed to improve on our previous work by: firstly, taking into account video recordings of the interactions. Secondly, designing a custom neural network architecture that deals with audio and visual features separately before combining them into one latent representation. Thirdly, using pretrained neural network backbones in each feature processing stream and finally, experimenting with feature fusion using principle components analysis to prevent our results from being limited by being only able to use a single feature representation.

The design of this architecture is inspired by other deep learning approaches that utilize attention mechanisms leveraged from deep convolutional neural networks for recognition tasks involving visual and audio data captured from human subjects - specifically in emotion recognition and related tasks [Zhao et al., 2021] [Zhao et al., 2020]. Our approach is similar to that in [Zhao et al., 2020] in that we use their convolutional architecture for each of the attention mechanisms, although in our case we use only frame-wise attention in both the audio and visual streams. We also use an InceptionV1 ResNet trained on VGGFace2 instead of the 3DResnet used in that study as it was more suited to our problem and our baseline SVM experiment showed convincingly that this feature representation was the most informative for the task. As in [Zhao et al., 2020] we compute the frame-wise attention (i.e. along the time dimension in each case) for the audio and visual streams in the following way. We adapt their formulation here for the sake of completeness and clarity with respect to how we have modified their approach. The full architecture is displayed in Section 4.4.2

Audio Temporal Attention Subnetwork

Let $\mathbf{x}_i^A$ be the i^{th} audio feature matrix input. We first center crop $\mathbf{x}_i^A$ to a fixed length l such that $\frac{l}{s} \in \mathbb{N}$ for some positive integer s giving $\mathbf{x}_i^{A'}$. If the time dimension of $\mathbf{x}_i^A$ is less than l then we pad the input on ether side with zeros such that it's length is now equal to l. We then split $\mathbf{x}_i^{A'}$ into s segments and stack them on top of one another such that $\mathbf{x}_i^{A'} \in \mathbb{R}^{s \times \frac{l}{s} \times n}$. The model then receives a batch of size b of these tensors which is then fed through the model's audio stream.

The first step of the audio stream is to process each of the $b \times s$ feature segments through a ResNet18 model [He et al., 2016] pretrained on the ImageNet dataset. This may sound surprising given that we are using using an ImageNet trained model on MFCC audio representations (since ImageNet contains no MFCC examples) but research has shown that using such ResNets on MFCC features matrices dependably improves model scores [Palanisamy et al., 2020] and indeed we also found this to be the case in our experiments. We then take the output F_j^A of the fifth convolutional stack of the pretrained ResNet18 model and perform spa-

tial average pooling over the feature maps producing $F_j^{A'}$ (where j indexes over the feature matrix segments). This downsamples the output of the ResNet from $F_j^A \in \mathbb{R}^{s \times h \times w \times c}$ to $F_j^{A'} \in \mathbb{R}^{s \times c}$ where s is the number of segments, h and w are the height and width of the feature maps respectively, and c is the number of channels, creating a $1 \times c$ length descriptor for each of the segments. The goal is now to learn an $s \times 1$ length descriptor for the audio feature matrix segments where the k^{th} element of the descriptor weights the k^{th} segment according to its importance in classifying the input sample. This descriptor is learned using a convolutional stack that consists of a 1D convolutional layer, a fully connected linear layer, and a ReLU non-linearity such that:

$$H^A = W_1^A(W_2^A(F_j^{A'})^T)^T \tag{4.1}$$

Where W_1^A and W_2^A are $s \times s$ and $1 \times c$ learnable parameter matrices for the linear and convolutional layers respectively. We then compute the activation of the audio attention subnetwork A^A i.e. the $s \times 1$ length segment descriptor as:

$$A^A = \text{ReLU}(H^A) \tag{4.2}$$

The output embedding for the audio stream, i.e. the representation of which audio segments are most relevant to the classification of the input example to a particular self-disclosure class, is computed via:

$$E^A = \sum_{j=1}^{S} F_j^{A'} A^A \tag{4.3}$$

Visual Temporal Attention Subnetwork

The approach to achieve the audio embedding E^V for the visual features extracted from the videos follows precisely the same steps as the audio temporal attention algorithm. The principal differences in practice are that we use the InceptionV1 ResNet archietcture trained on VGGFace2 that we used in our SVM baseline experiments instead of the ResNet18 model.

Given the output embeddings E^A and E^V for the audio and visual processing streams we then summarize the features using average pooling by computing the mean of each embedding vector along the time domain (i.e. across segments) giving $E^{A'}$ and $E^{V'}$. These are then concatenated before being fed to a linear

layer containing 7 neurons representing each of the self-disclosure score classes. This produces output $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = \text{Softmax}(W^{AV}\text{concat}(E^A, E^V)) \tag{4.4}$$

where W^{AV} is a learnable parameter matrix related to the linear output layer, concat() is the concatenation operation, and Softmax() is the softmax function that return normalized probabilities over the seven self-disclosure score classes.

4.4.3 Ablation Experiment Parameters

In our experiments we tested the influence of two different visual feature sets, two experimental framings, and four different loss functions to determine the best configuration for the problem.

Visual Feature Sets

First, we wanted to test the efficacy of just the facial features output by the InceptionV1 ResNet architecture pretrained on VGGFace2 as our SVM experiments showed that these were likely to be the most informative visual features for the task. Next we wanted to test a combination of all visual features that we extracted as detailed in Section 4.4.1. To reduce the dimensionality of this feature space we concatenated all of the visual features together after the visual input has been passed through the ResNetV1 backbone in the visual stream and performed principal components analysis with parameters set such that 99% of the variance in the data was explained by the resulting feature matrix. This resulted in a dimensionality change in this feature space from a 555 dimensional feature vector to a 67 dimensional feature vector for each video frame.

Classification Vs. Regression

In [Powell et al., 2022] the authors found that there was a nuance in the approach to classifying self-disclosure scores. As we state in that study, participants rated the degree of self-disclosure in their interactions on a likert scale between 1 and 7. This means that each score falls into a discrete class meaning that one plausible way to frame the problem is as an n-class classification problem. However, loss functions related to n-class classification problems often treat incorrect guesses in the same manner i.e. there is no sense in which one guess can be numerically represented as being closer to a correct guess than any of the other possible guesses. The self-disclosure score data, however, is scaled in the sense that a model guess

of 2 for ground truth self-disclosure score of 1 should be treated as a better guess than 6 or 7. In this light an argument could be made that the problem is better represented as a regression problem. In [Powell et al., 2022] the authors found that framing the problem in both ways produced similar results and as such no clear empirically informed decision could be made about what approach worked best. In light of this we decided to test the effects of both approaches on our results.

Loss Function

We wanted to study the effect of loss function on the problem. Standardly, regression based methods minimize a mean-squared error loss in order to optimize the parameters of a given model. Since we had no good reason to suspect that this particular problem required an alternative regression-based loss function we chose only to base our regression results on the mean squared error loss. For the classification version of the problem we chose a categorical cross-entropy loss function for our experiments. For this loss, research has shown that label smoothing, a technique whereby standard 'hard' labels are modified by a smoothing parameter α via $y_k^{LS} = y_k(1-\alpha) + \frac{\alpha}{K}$ where k indexes over the total number of classes (seven in the case of this study), can drastically improve results [Yuan et al., 2020]. As such we chose to include a cross entropy loss with label smoothing as part of ablation study. Last, we wanted to explore the possibility of designing a custom loss function that was able to strike a balance between the classification and regression versions of the task i.e. one that leveraged the fact that the data was categorical while also preserving the notion that certain guesses were better with respect to a ground truth label than others. Taking inspiration from [Zhao et al., 2020] we designed a custom cross entropy loss function that penalises guesses with greater severity the further they are from the ground truth label. For example, for an input sample with labelled self-disclosure score of 7 a guess of 1 will result in a higher loss than a guess of 2, a guess of 2 will result in a higher loss than a guess of 3, and so on. To do this we amended the standard cross-entropy loss function which can be expressed as:

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} \mathbb{1}_{[c=y_i]} \log p_{i,c} \tag{4.5}$$

where N is the number of input samples, C is the number of classes, $\mathbb{1}_{[c=y_i]}$ is an indicator variable that equals 1 when the predicted class is the same as the ground truth class and $p_{i,c}$ is the probability that the i^{th} sample belongs to the c^{th} class. We added a penalty term to Equation (4.5) that formalizes the idea that guesses

at a greater distance from the ground truth should be penalised more severely. This gives what we term the scale preserving cross-entropy loss:

$$\mathcal{L}_{SPCE} = -\frac{1}{N}\sum_{i=1}^{N}(1+\lambda(|y-\hat{y}|)^{\mu})\sum_{c=1}^{C}\mathbb{1}_{[c=y_i]}\log p_{i,c} \tag{4.6}$$

where λ and μ are hyperparamters that change the degree to which a incorrect guess is penalised with respect to how far away it is from the ground truth self-disclosure label.

Lastly, we wanted to test the contribution of the attention mechanisms. To do this, after the first ablation study was completed, we took the best performing model and removed the attention mechanisms to see how this would affect the training.

Taken together, the parameters of the ablation study lead to nine different training configurations for the multimodal attention network, the specifications of which are presented in Section 4.5.

4.4.4 Model Training

Regression and classification models were trained over 100 epochs, while the SPCE models where trained on 150 since we found that they took longer to converge. All network version we trained using the Adam optimizer [Kingma and Ba, 2014], an initial learning rate of 0.01, and mini-batch size of 35. Audio feature inputs were cropped to length $l = 128$ and divided in to $s = 4$ segments. Visual input features were cropped to $l = 210$ frames and divided into $s = 7$ segments. We prepared the training data as in [Powell et al., 2022] splitting the training and testing datasets into an 80/20 split and used weighted random sampling to account for imbalanced classes. As in [Powell et al., 2022] the training and test dataset were split by participant such that the model would be tested on participants that it had not seen during the training phase. We chose not to train the networks using cross validation due to the fact that these kinds of approaches tend to overestimate the performance of models during training time and give a less reliable measure of how well a model will generalize. Traditional train-test splits, such as the one we used in this experiment, will give a more realistic estimation of how a model will perform since, in the real world, the model will be used overwhelmingly on data that it has not seen. Since the long-term goal of this project is to see these models used in real world environments we decided that this train-test schema was the most appropriate.

Each model was trained five times and the average F1 score and standard deviation over all five training instances were computed to give a balanced assessment of the model's performance. We chose to validate the models using f1 scores so that our results were directly comparable to those produced by our SVM experiments.

4.5 Results

The results of our ablation study are displayed in Section 4.5. We found that all versions of the multimodal attention network scored significantly above the best SVM baseline. Interestingly, departing from [Powell et al., 2022], where regression and classification models performed about as well as each other, we found that a classification framing (treating self-disclosure scores as discrete classes) was significantly more effective at modelling the problem than a regression framing (treating the scores as being derived from the continuous number line). In all cases we found that, within each experimental framing, the features derived from principle components analysis outperformed models trained on just InvceptionV1 facial features. This is perhaps unsurprising for two reasons. Firstly, because this feature set was comprised of three times the number of features than the pure InceptionV1 feature set before it was condensed to its principal components. Secondly, because significantly reducing the number of features (from 512 in the pure InceptionV1 case to 67 in the principal components case) would mean that our model was less susceptible to the curse of dimensionality i.e. that it would require much less data to effectively model that smaller set of features. Further, we found that label smoothing produced improvements in results when compared to the non-label smoothing variant of the cross entropy loss. Finally, we found that our scale preserving cross-entropy loss outperformed all but one version of the model (principal component features with label smoothing cross-entropy loss) to which it equalled in performance.

Finally, we found that the attention mechanisms played a significant role in the model's performance, increasing the F1 score of the best performing model from 0.73 to 0.83. This result is perhaps not surprising since the literature on attention mechanism shows convincingly that attention helps neural network models develop more efficient representations on time-series based classification tasks. Intuitively speaking, the attention mechanism in our model will have weighted the features of the input embeddings that were important with respect to classifying each of the attention classes. Without this mechanism the model will have treated each of the features from the input embeddings equally which would mean that non-important features would have been carried over with greater presence into subsequent layers of the network. More technically, the temporal average pooling layer will condense

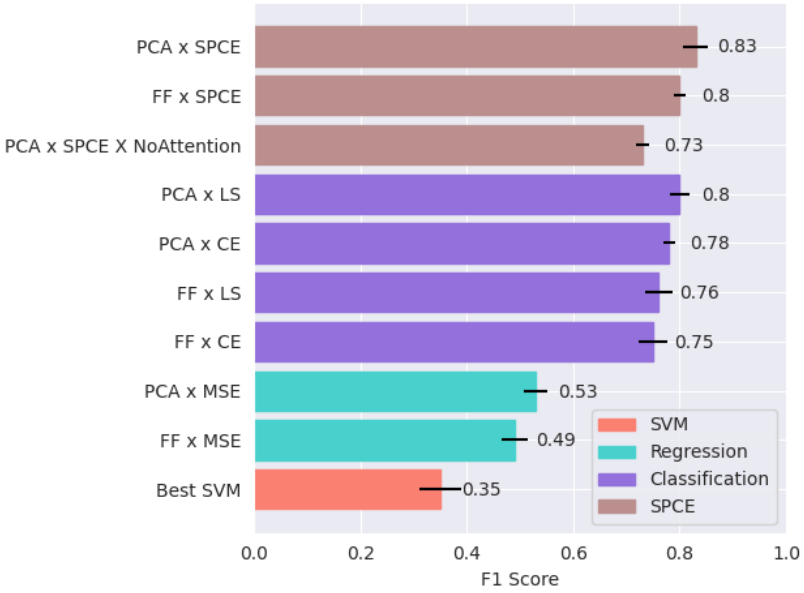


Figure 4.4: F1 scores for our multimodal attention network trained on a different combination of data input representations (principal components analysis data (PCA), face features only (FF)) and loss functions (categorical cross entropy (CE), cross entropy with label smoothing (SE), mean squared error (MSE), and our scale preserving cross entropy loss (SPCE)). We have also colour coded the different experimental framings we used for the deep learning experiments.

the matrix embeddings output from the pretrained backbones into a single vector. Each element of this vector can be seen as an abstract feature in the embedding space. Naturally, not all of these abstract features will be equally informative for the network with respect to the task of classifying a participant's degree of self-disclosure. The attention subnetwork will then learn an attention vector that will discriminate the importance of each of these features. As such, each element in the attention vector can be understood as a weight that will be applied to each of the condensed embedding features, with higher values indicating that that feature is of more importance to the classification of the given input. As such, the attention vector is required to be of the same dimension as the condensed embedding, so that they can be multiplied together to produce a weighted version of the output of the average pooling layer.

4.6 Discussion and Conclusion

Overall we report significant increases on performance in this task from [Powell et al., 2022]. We hypothesise that this is due to a number of significant developments from that work. Firstly, we collected a much larger dataset meaning that the models had more examples to learn from. Second, in this case all interactions

were recorded between the same interaction dyad i.e. between a human and a Pepper robot. In [Powell et al., 2022], the authors collected interaction data between three different interaction dyads: human–human, human–embodied robot, and human–voice agent. One reason that results may have been worse in that case is due to the possibility that vocal features particular to each self-disclosure class may have been modulated by the kind of agent the participant was interacting with. In our study, since the interaction partner was always the same, there would not have been this variability and thus the learning task could have been easier. Third, we used significantly more sophisticated models, leveraging representational power provided by large deep neural networks trained on extremely large datasets. Further, our use of frame-wise attention mechanisms make use of deep learning techniques that have shown to be state-of-the art on video and language modelling tasks perhaps providing a straightforward upgrade of the 'off-the-shelf' models that were used in the previous study. Lastly, and perhaps most obviously, in this study we modelled two different sensory modalities (audio and visual) as opposed to the single sensory modality that was considered in [Powell et al., 2022]. It may well be the case that the auditory domain holds less discriminative information than the visual domain for self-disclosure modelling and thus, the previous study was automatically at a disadvantage in only considering the former.

One surprising observation from our baseline experiments was that the visual features were most effective at allowing the SVM models to predict a particular subjective self-disclosure score. While much of the literature on self-disclosure is varied with respects to its definitions one thing that is generally agreed upon is that self-disclosure is primarily a verbally communicated social phenomenon [Cozby, 1973][Omarzu, 2000]. In light of this it might be expected that the audio modality would produce the best results. It may well be the case that the way in which the audio features were averaged caused some of the information to be lost. Unfortunately, a thorough investigation of why the visual features were the most informative is outside of the scope of this paper.

While this study shows significant improvement on previous work done on modelling self-disclosure with neural networks, it remains to be seen whether the advances that we detail here are significant enough for these models to be implemented in social robots. As discussed in [Powell et al., 2022], there is a significant risk associated with an incorrect self-disclosure scoring in a real world setting. Assuming that a person is sharing very little self-disclosure when in fact they believe themselves to be sharing a significant amount could lead to that person feeling as if they are being ignored or that the sensitive information that they are sharing is not worthy of the listener's consideration. Conversely, assigning a very high self-disclosure score in a situation where an interaction partner does not believe

themselves to be sharing a significant amount of personal information could cause undue levels of attention to be paid to a situation which is not important. The issue described in both of these cases would be significantly confounded within the context of mental health interventions, where the risks associated with not picking up on a patient's self-disclosure related signals could be very damaging. As such, considerably more work needs to be done before models like ours are considered for real world application. There are at least two ways that steps could be taken in this direction. Firstly, significantly more data should be collected to improve the performance of the models. Secondly, a study should be carried out to asses the differences between model performance and the performance on the same task by a trained professional. It is often the case that the quality of a machine learning model and it's viability as a real world application is measure with respect to its ability to achieve 'human-like' performance. It makes sense that a model that is effective at recognizing the degree to which a person is disclosing personal information should be able to do so at least as well as a trained professional (particularly if that model is to be implemented within the context of health care interventions).

Further, there are ways in which improvements on our approach might be made in the short term. Firstly, since we found that performance on the task was improved when visual features were combined using principal components analysis, it's likely to also be the case that performance improvements could be achieved by combining audio features. In particular, we did not experiment with ways to combine outputs from the transformer layers of wave2vec2.0 with the MFCC features. Additionally, more empirical work could be done to ascertain the best way to combine feature sets in both the audio and visual cases. For one such example, [Lin et al., 2021] used a denoising autoencoder to learn a compressed latent representation of the concatenated input features. A future study should empirically test the hypothesis that such a latent representation exists in a more effective input feature space than the one produced by principal components analysis. Further studies could also look into experimenting with other kinds of attention. In [Zhao et al., 2020] the authors use channel-wise attention and spatial attention in the visual stream on top of the frame-wise attention that both of our methods share. One development along these lines could be to implement an attention mechanism that produces a descriptor over the features i.e. the columns of the input matrices. In this way the model would hold a representation of not only which frames of the input are important to its classification but also which features are important. Lastly, the model could be altered to leverage 3D ResNets to produce higher dimensional features over the input video frames. This approach however would require a 3DResNet trained on a very large video dataset focused on the modelling of human faces and, to our knowledge, no such pretrained model is publicly

available. Taking from the modelling literature on self-disclosure [Soleymani et al., 2019], show that very good results on the task of (non-subjective) self-disclosure modelling between two human interactors can be achieved multi-modally with the addition of lexical features. In that study, the authors use a pretrained BERT language model [Devlin et al., 2019] to extract features related to the words used in each utterance. A significant part of self-disclosure (at least in the human–human case) is thought to be communicated verbally [Cozby, 1973][Omarzu, 2000]. This a future study could look at including this modality in the human–robot interaction version of the task.

Once it is found that increasing the dataset size and complexity of the models has no impact on model performance, a useful follow on study should investigate how the model comes to its decision with respect to the self-disclosure classifications that it makes. One fruitful avenue in this regard would be to look at which parts of the input are being attended to by the model. For example, for their emotion recognition model, [Zhao et al., 2020] utilise the Grad-cam [Selvaraju et al., 2017] algorithm to produce a heat map over the input frames of video data that shows which individual frames and which parts of each frame the model is attending to to make its decisions. In our case, this algorithm could be applied to determine which local areas of a person's face contain the most telling features for each of the self-disclosure classes. This same approach could be applied to the spectrograms that we used for the input to the audio recognition arm of the model. This investigation would help considerably in understanding how and why the model makes the decisions that it does. Since, in this case, we were only interested in pushing model performance along the dimension of its accuracy, we chose not to pursue this line of investigation. Of course, there are considerable advantages to understanding how neural network models behave and how they come to the conclusions that they do. In particular, such an understanding is likely to contribute significantly to the goal of getting people to trust robots and artificial agents that make use of such networks. Moreover, understanding what kinds of representations are useful for different kinds of tasks will significantly streamline training processes. In these cases, experimenters and neural network engineers will have to spend significantly less time and computational resources engineering massive sets of features in hopes of catching ones that the model finds useful. These are issues that are currently at the forefront of a considerable amount of machine learning research and are dealt with in more detail in Chapter 6 of this thesis.

It is clear that our approach draws significantly on work in the emotion recognition literature. As discussed in the previous chapter, there are some clear differences between both the behavioural phenomena of self-disclosure and emotion and the machine learning tasks that are used to model them. Primarily, emotion

can be seen as a state of mind that gives rise to observable behaviours. Being angry, for instance, is a phenomenal experience that people undergo in response to a number of life events. This phenomenal experience is likely to be generated by a complex time-series of neural events that will, in turn, reliably give rise to changes in behaviour that are detectable by human interaction partners. We have a good idea that someone is angry, for instance, when they clench their fists, furrow their brow, tense their jaw, and speak loudly and aggressively, among a range of other behaviours. Emotion detection, then, can be understood to be the task of detecting a state-of-mind of an individual through the recognition of behaviours that reliably indicate that a person is currently in that state-of-mind. Self-disclosure, on the other hand, is an action that realises the intention to self-disclose. This action is usually verbal but could feasibly be written or communicated in some other way. It is unlikely, and unintuitive, that self-disclosure is a state-of-mind in the same way that an emotion is. However, the two are likely to be causally related. It's not hard to imagine that, for certain people, being in a certain emotional state might make it more or less likely that that person will self-disclose the reason for them being in that emotional state. If I am angry, I am likely to want to express to a person why I am angry, what situation or what person has made me angry, for example.

The two problems do however overlap with respect to how they can be approached from a machine learning standpoint. Self-disclosure and emotion are, and give rise to, behaviours that can be detected by audio-visual sensors. Thus, we can use the data produced by these sensors to train neural network models to model and classify such behaviours. Further, research suggests that the kinds of behaviours that are indicators of emotion and self-disclosure overlap. Both emotion and self-disclosure are communicated through facial feature changes and lexical and non-lexical changes in speech, for instance. Thus, as we do in this study, we can use similar techniques to model both emotion and self-disclosure. Since the field of emotion recognition is significantly more developed than that of self-disclosure classification, it makes sense to draw inspiration from this field of work. However, as discussed in [Powell et al., 2022], the problem of self-disclosure classification can be seen as subtly distinct from that of emotion classification. This is in the sense that the former is interested in distinguishing between degrees of the same behaviour rather than different kinds of behaviours that have been grouped together. This distinction in part gives rise to the challenge that we faced in regards to loss function. Distinguishing between different kinds of emotion could sensibly be argued to be an n-class classification problem, where each emotion that you are interested in is a different class. Ostensibly, there is no need to think about degrees of behaviour, since all we are interested in is whether a particular video,

for example, shows a person who is angry, sad, happy, upset and so on. Because, as we have discussed above, the task of self-disclosure classification can be seen as somewhere between a classification and regression task (where we want to predict where some input will land on a scale) there is a need to develop models that are able to strike a balance between these two kinds of problems.

Finally, as in [Powell et al., 2022], we made the choice to collate the data across the experimental sub-groups. That is to say, participants from the COVID-related-question group and the non-COVID-related-question subgroup were grouped together for the purposes of model training. Likewise we collated all interactions from all different points in time (i.e. the model was trained on interactions from all weeks). As we were only interested in how well we could train a neural network model to predict a person's degree of subjective-self disclosure, it was of greater importance that we train the models on as much data as possible. A model in production in a real world environment would need to have learned features across and common to a variety of sensitive topics and as such we felt that combining the two groups into one would be better suited to meet this constraint. A follow on experiment may wish to look into the differences in model training between the two groups. However, using only the current dataset, this would drastically impact dataset size (effectively dividing it in two in the case of question-type subgroups or ten in the case of successive interactions over the five weeks) for each model training condition, and we do not believe that an effectively trained model could be produced on such small datasets. A follow on study would, therefore, need to collect significantly more data to correct this issue. A further interesting analysis might look at the behavioural differences between the two subgroups. It could, for example, look into how the features of a subjects voice and face change depending on the kind of question and how long that participant has been interacting with the robot partner. Again, since our aims at this stage where only to try to push the accuracy of the model beyond that which we achieved in the previous paper, such a behavioural experiment is outwith the scope of the current project. However, a behavioural analysis of these groups with respect to the difference in their self-disclosure is currently planned by a different team in our group lead by the second author of this paper.

We believe that this study makes significant strides into the new field of subjective self-disclosure modelling. Not only do we show considerable improvements over results of any previous studies on the topic but we provide an extensive and high quality multimodal dataset that can be used and expanded on by researchers in the field.

Preface to Chapter 5

From September 2020 to December of 2021 I carried out an internship with Merck KGaA's artificial intelligence team based in Darmstadt, Germany. The goal of the research team was to investigate cutting edge machine learning and artificial intelligence techniques that strongly leveraged research from contemporary neuroscience. I undertook this internship in order to develop the scope of my thesis to look at neural network research on computational cognition outside of the field of deep learning. While the approach and methods used in this chapter differ from those discussed in the previous chapters, the goal of the research remains the same. Namely, to investigate how neural network methods can used to model human–like cognitive capabilities that could, in principal, be deployed in robots that were designed to function in complex social environments.

The mathematical notation in this chapter differs slightly from that used in the thesis so far (e.g. the change from $\mathbf{x}$ to $\vec{x}$ to indicate a vector). This was done intentionally as we found these notational differences to clearer in this case. In each case the notation is explained, usually immediately after an equation or other mathematical expression.

Chapter 5

A Hybrid Biological Neural Network Model for Solving Problems in Cognitive Planning

Henry Powell

Mathias Winkel

Alexander V. Hopp

Helmut Linde

[1]A version of this chapter was accepted for publication in *Nature Scientific Reports* on 4/12/2022.

Abstract

A variety of behaviors, like spatial navigation or bodily motion, can be formulated as graph traversal problems through cognitive maps. We present a neural network model which can solve such tasks and is compatible with a broad range of empirical findings about the mammalian neocortex and hippocampus. The neurons and synaptic connections in the model represent structures that can result from self-organization into a cognitive map via Hebbian learning, i.e. into a graph in which each neuron represents a point of some abstract task-relevant manifold and the recurrent connections encode a distance metric on the manifold. Graph traversal problems are solved by wave-like activation patterns which travel through the recurrent network and guide a localized peak of activity onto a path from some starting position to a target state.

5.1 Introduction

Understanding the computational principles of the human brain is one of the most ambitious goals of neuroscience, and one which promises vast intellectual and practical benefits. Yet in the face of its enormous complexity, even after decades of intense research, the understanding of the brain's algorithms remains very vague, at best.

Some level of insight stems from the thorough analysis of neural feed-forward architectures. There, a neuron is usually considered to be an electrical component which computes an output by applying a non-linear function on some weighted sum of its synaptic inputs and transmits the result to a next higher layer of neurons.

This simplistic but effective model has been exploited in many technical applications in the form of (deep) artificial neural networks. Their neurons are typically organized in layers, each of which sends its output signals only to the next higher layer. Such feed-forward processing has shaped our intuition of neurons as "feature detectors" which fire when a certain approximate configuration of input signals is present, and which aggregate simple features to more and more complex ones layer by layer.

In the brain, though, the overwhelming majority of connections between neurons are recurrent, i. e. they connect neurons within the same cortical area or transmit information from higher areas back to lower ones. For example, in the visual cortex, synapses from the lateral geniculate nucleus of the thalamus, i. e. the feed-forward connections, make up only 5 %–10 % of the excitatory synapses in their target layer 4 of V1 in cats and monkeys [Douglas and Martin, 2007]. The understanding of neurons as "feature detectors" can therefore only represent a small fragment of the over-all picture.

Several possible explanations of the function of these recurrent connections have been proposed. For example, it has been suggested that neural activity follows almost chaotic trajectories in an extremely high-dimensional state space while the dynamics are still sensitive enough to be influenced by the relatively small share of feed-forward connections [Singer and Lazar, 2016]. In this conceptual framework, attention signals, past memories, and sensory input are merged in order to guide the system towards well-separated, lower-dimensional subspaces which represent certain states of perception. It is also hypothesized that top-down projections from higher cortical areas transmit predictions or expectations to influence how the lower areas interpret the incoming sensory data [Miller and Buschman, 2013, Kveraga et al., 2007]. Such predictions are thought to play a role in noise-reduction and signal-restoration or to direct attention bottom-up to features which deviate from the prediction and thus require some executive

reaction. Nevertheless, the full computational purpose of the recurrent connections is still little understood [Douglas and Martin, 2007].

In the present paper, we propose a new algorithmic role which recurrent neural connections might play, namely as a computational substrate to solve graph traversal problems. We argue that many cognitive tasks like navigation or motion planning can be framed as finding a path from a starting position to some target position in a space of possible states. The possible states may be encoded by neurons via their "feature-detector property". Allowed transitions between nearby states would then be encoded in recurrent connections, which can form naturally via Hebbian learning since the feature detectors' receptive fields overlap. They may eventually form a "map" of some external system. Activation propagating through the network can then be used to find a short path through this map. In effect, the neural dynamics then implement an algorithm similar to Breadth-First Search on a graph.

The remainder of the paper is organized as follows: In Section 5.2, we give a conceptual overview, describe the technical details of the proposed model and show some simulation results for an exemplary numerical implementation of the model. We then review empiric support for some components of the model in Section 5.4. Limitations, implications and ideas for further development are discussed in Section 5.5. The more technical details related to general graph theory and to the numerical implementation can be found in Section 5.3.

5.2 Proposed Model

5.2.1 A Network of Neurons that Represents a Manifold of Stimuli

We consider a neural network which is exposed to some external stimuli-generating process under the assumption that the possible stimuli can be organized in some continuous manifold[2] in the sense that similar stimuli are located close to each other on this manifold. For example, in the case of a mouse running through a maze all possible perceptions can be associated with a particular position in a two-dimensional map, and neighboring positions will generate similar perceptions, see Figure 5.1a.

Proprioception, i. e. the sense of location of body parts, can also be a source of stimuli. For example, for a simplified arm with two degrees of freedom every

[2]In mathematics, a manifold is a topological space which has the structure of a Euclidean space locally at each point. In contrast to a (globally) Euclidean space, manifolds can be topologically diverse and – when endowed with a Riemannian metric – curved. For example, a saddle-shaped hyperbolic plane, a sphere or a torus are manifolds.

possible position of the arm corresponds to one specific stimulus, cf. Figure 5.1b. All possible stimuli combined give rise to a two-dimensional manifold. The example also shows that the manifold will usually be restricted since not every conceivable combination of two joint angles might be a physically viable position for the arm.

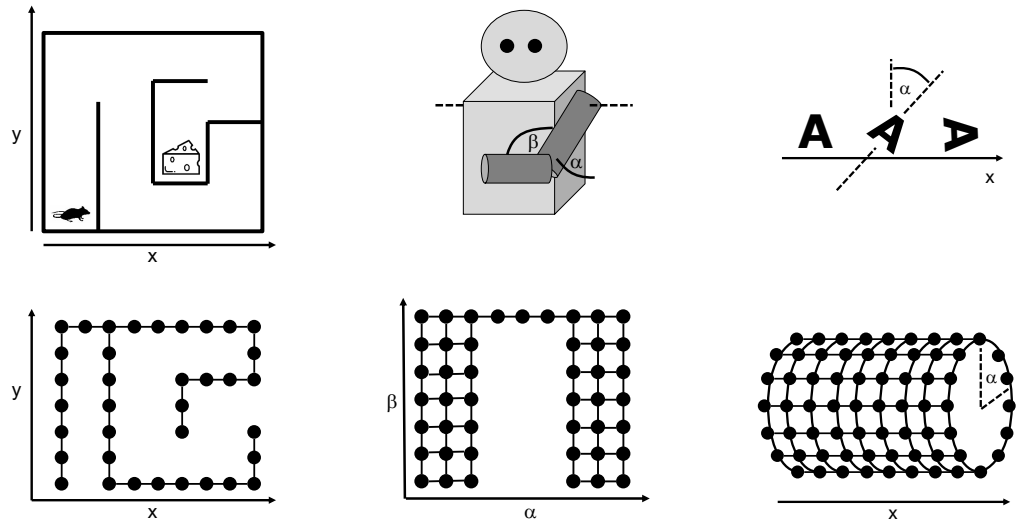
The manifold of potential stimuli needs not necessarily be embedded in a flat Euclidean space as in the case of the maze. For example, if the stimuli are two-dimensional figures which can be shifted horizontally or rotated on a screen, the corresponding manifold is two-dimensional (one translational parameter plus one for the rotation angle) but it is not isomorphic to a flat plane since a change of the rotation angle by 2π maps the figure onto itself again, see Figure 5.1c.

We assume that such manifolds of stimuli are approximated by the connectivity structure of a neural network which forms via a learning process. The result is a neural structure which we call a *cognitive map*. The defining property of a cognitive map is that is has a neural encoding for every possible stimulus and that two similar stimuli, i. e. stimuli which are close to each other in the manifold of stimuli, are represented by similar encodings, i. e. encodings which are close to each other in the cognitive map (of course, we do not imply that two neurons which are close to each other in the connectivity structure are also close to each other with respect to their physical location in the neural tissue). There is considerable evidence, which we review in Section 5.4.1, that such cognitive maps are implemented by the brain, but the details of the encoding of stimuli remain mostly unclear.

For the model, we make a very simplistic choice and assume a single-neuron encoding, i. e. the manifold of stimuli is covered by the receptive fields of individual neurons. Each such receptive field is a small localized area in the manifold and two neighboring receptive fields may overlap, see Figure 5.2. Such an encoding is a typical outcome for a single layer of neurons which are trained in a competitive Hebbian learning process [Rumelhart and Zipser, 1985]. Examples for such competitive learning algorithms are Kohonen Maps [Kohonen, 1982], (Growing) Neural Gas [Martinetz and Schulten, 1994] or variants of sparse coding dictionary learning [Elad, 2010].

The key idea of the model is that solving a problem that can be formulated as a planning problem in the manifold of stimuli, can be solved as a planning problem in a corresponding cognitive map. To this end, it is not enough to consider the cognitive map as a set of individual points, but its topology must be known as well. This topological information will be encoded in the recurrent connections of the neural network.

It seems natural that a neural network could learn this topology via Hebbian learning: Two neurons with close-by receptive fields in the manifold will be excited simultaneously relatively often because their receptive fields overlap. Con-



(a) Approximate positions in the maze are encoded in single neurons. Overlapping receptive fields lead to recurrent connections which resemble the structure of the maze. The planning problem is to find a way through the maze given the current position of the cheese and the mouse.

(b) Approximate positions of the "arm" are encoded in single neurons. Physically impossible positions where the "arm" intersects with the "body" are not encoded at all (because they have never been observed by the neural network) giving rise to the gap in the center of the cognitive map. An example planning problem is to move the "hand" from behind the body to a position in front of the body without collision.

(c) The visual stimulus is always the letter "A", but at different x-positions and tilted at different angles α. Due to the periodicity of the stimulus under change of α, the resulting cognitive map has the topology of a cylinder. An example planning problem in this case is the decision whether the "A" has to be moved/tilted to the left or to the right to convert it from some given position to another one.

Figure 5.1: Three examples of stimuli-generating processes and recurrent neural networks representing the corresponding manifold of stimuli.

sequently, recurrent connections within the cognitive map will be strengthened between such neurons and the topology of the neural network will approximate the topology of the manifold, see Figure 5.2. This idea has been explored in more detail by Curto and Itskov in [Curto and Itskov, 2008]. Indeed, previous work on the formation of neocortical maps that code for ocular dominance and stimulus orientation suggest that the formation of cognitive maps could well occur in this fashion [Miller, 1992]. For a review and comparison of these kinds of cognitive maps see [Erwin et al., 1995]. Recent studies also show that recurrent neural networks might serve even more purposes, for example for working memory [Kim and Sejnowski, 2021, Xie et al., 2022] or image recognition [Wang et al., 2022].

One of the key assumptions of the model is that the agent has formulated the cognitive map in advance, for example by exploring the environment or investi-

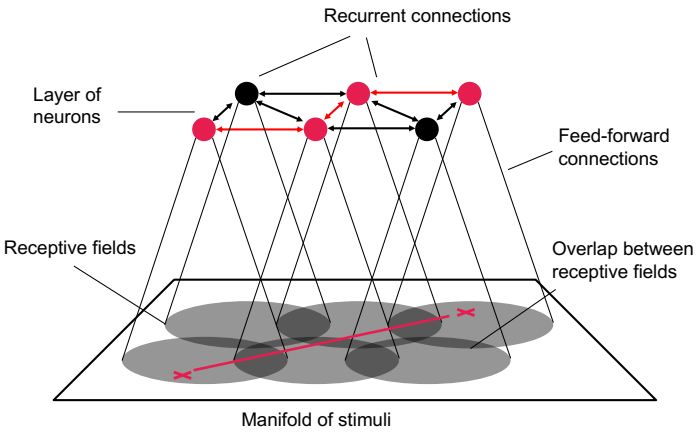


Figure 5.2: In the model, the recurrent connections within a single layer of neurons approximate the topology of the manifold of stimuli. During the learning process, the strongest recurrent connections are formed between neurons with overlapping receptive fields. The problem of finding a route through the manifold (red line) is thus approximated by the problem of finding a path through the graph of recurrent neural connections (red path).

gating the object which is to be manipulated. Thus, the scope of our planning problems are such that the goal state is known and can be represented readily by the planning agent. In some scenarios, however, such as in the case of a mouse in a maze looking for a reward, it may be the case that the agent does not have a clear idea of where the goal state is. In these cases, it would be difficult for the "to-be" representation to be activated since the agent cannot necessarily represent them directly. This is not necessarily a problem for our model, however, as we do not require that the "to-be" state is directly perceived, only that it is represented by the planning agent such as its being recalled in memory. In the case where the agent does not know exactly where the reward is, or what the best final configuration of the system is to be, we would assume that there would be some assumption on the part of the agent about what that end-state might be. This assumption, again, would likely be the result of a learned probability distribution over possible/ desired end states that was learned when the cognitive map was being formulated. Thus, in the case of the rat in the maze, the mouse need only represent where it thinks the goal is likely to be in order for the start of the planning process to take place.

To avoid confusion with related concepts in machine learning, note that the present definition of recurrence is not exactly the same as the one used, for example, in Long Short-Term Memory networks [Hochreiter and Schmidhuber, 1997a]. Those algorithms employ recurrent connections as a loop to mix some input signal of a neural network with the output signal from a previous time step. The present model, however, separates between the primary excitation by some ex-

ternal stimulus via feed-forward connections and the resulting dynamics of the network mediated by the recurrent connections as described in the following.

5.2.2 Dynamics Required for Solving Planning Problems

Having set up a network that represents a manifold of stimuli, we need to endow this network of feed-forward and recurrent connections with dynamics. We do so by imposing two interacting mechanisms.

First, the neurons in the network should exhibit continuous attractor dynamics [Rolls, 2010]: If a "clique" of a few tightly connected neurons are activated by a stimulus via the corresponding feed-forward pass, they keep activating each other while inhibiting their wider neighborhood. The result is a self-sustained, localized neural activity surrounded by a "trench of inhibition". In the model, this encodes the as-is situation or the starting position for the planning problem. Such a state is called an "attractor" since it is stable under small perturbations of the dynamics, and it is part of a continuous landscape of attractors with different locations across the network. For a recent review of attractor networks, the reader is referred to [Rolls, 2010]. The dynamics of these kinds of bumps of activity in neural sheets of different kinds has been studied in depth in [Amari, 1977] and applied to more general problems in neurosciene [Taylor, 1999] but have not, as of yet, been used as means to solve planning problems in the way proposed here.

Second, the neural network should allow for wave-like expansion of activity. If a small number of close-by neurons are activated by some hypothetical executive brain function (i. e. not via the feed-forward pass), they activate their neighbors, which in turn activate theirs, and so on. The result is a wave-like front of activity propagating through the recurrent network. The neurons which have been activated first encode the to-be state or the end position of the planning problem.

The key to solving a planning problem is in the interaction between the two types of dynamics, namely in what happens when the expanding wave front hits the stationary peak of activity. On the side where the wave is approaching it, the "trench of inhibition" surrounding the peak is in part neutralized by the additional excitatory activation from the wave. Consequently, the containment of the activity peak is somewhat "softer" on the side where the wave hit it and it may move a step towards the direction of the incoming wave. This process repeats, leading to a small change of position with every incoming wave front. The localized peak of excitation will follow the wave fronts back to their source, thus moving along a route through the manifold from start to end position, see Figure 5.3.

The two types of dynamics described above are seemingly contradictory, since the first one restricts the system to localized activity, while the second one per-

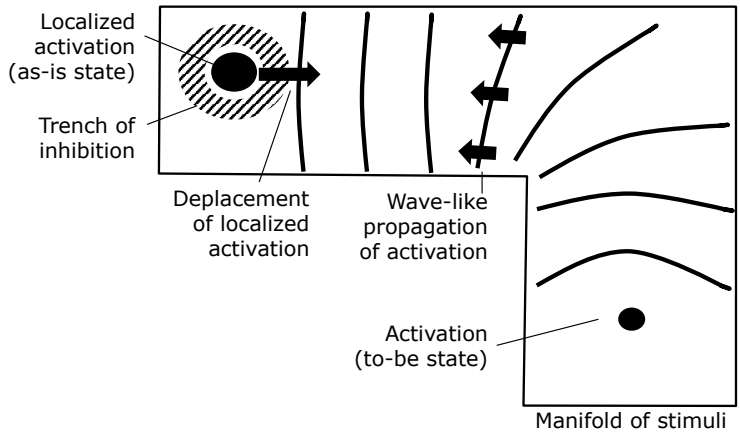


Figure 5.3: The as-is state of the system is encoded in a stable, localized, and self-sustained peak of activity surrounded by a "trench" of inhibition (top left corner). A planning process is started by stimulating the neurons which encode the to-be position (bottom right corner). The resulting waves of activity travel through the network and interact with the localized peak. Each incoming wave front shifts the peak slightly towards its direction of origin. Note that, for reasons of simplicity, we did not draw the neural network in this figure but only the manifold which it approximates.

mits a wave-like propagation of activity throughout the system. To resolve the conflict in numerical simulations, we have split the dynamics into a *continuous attractor layer* and a *wave propagation layer*, which are responsible for different aspects of the system's dynamical behaviour. We discuss the concepts of a numerical implementation in Section 5.2.4 and ideas for a biologically more plausible implementation in Section 5.5.

5.2.3 Connection to Real-Life Cognitive Processes

To make the proposed concept more tangible we present a rough sketch of how it could be embedded in a real-life cognitive process along with a speculative proposal for its anatomical implementation in the special case of motor control.

As an example, we consider a human grabbing a cup of coffee and we explain how the presented model complements and details the processes described in [Kolb et al., 2019] for that particular case. According to our hypothesis, the as-is position of the subject's arm is encoded as a localized peak of activity in the cognitive map encoding the complex manifold of arm positions. Anatomically, this cognitive map is certainly of a more complicated structure than the one in our simple model and it is possibly shared between primary motor cortex and primary somatosensory cortex.

We assume that the encoding of the arm's state works in a bi-directional way, somewhat like the string of a puppet: When the arm is moved by external forces,

the neural representation of its position mediated by afferent somatosensory signals moves along with it. On the other hand, if the representation in the cortical map is changed slightly by some cognitive process, then some hypothetical control mechanism of the primary motor cortex sends efferent signals to the muscles in an attempt to make the arm follow its neural representation and bring the limb and its representation back into congruence.

If now the human subject decides to grab the cup of coffee, some executive brain function with heavy involvement from prefrontal cortex constructs a to-be state of holding the cup: The final position of the hand with the fingers around the cup handle is what the person consciously thinks of. The high-level instructions generated by prefrontal cortex are possibly translated by the premotor cortex into a specific target state in the cognitive map that represents the manifold of possible arm positions. The neurons of the primary motor cortex and/or the primary somatosensory cortex representing this target state are thus activated.

The activation creates waves of activity propagating through the network, reaching the representation of the as-is state and shifting it slightly towards the to-be state. The hypothetical muscle control mechanism reacts on this disturbance and performs a motor action to keep the physical position of the arm and its representation in the cognitive map in line. As long as the person implicitly represents the to-be state, the arm "automatically" performs the complicated sequence of many individual joint movements which is necessary to grab the cup.

This concept can be extended to flexibly consider restrictions that have not been hard-coded in the cognitive map by learning. For example, in order to grab the cup of coffee, the arm may need to avoid obstacles on the way. To this end, the hypothetical executive brain function which defines the target state of the hand could also temporarily "block" certain regions of the cognitive map (e. g. via inhibition) which it associates with the discomfort of a collision. Those parts of the network which are blocked cannot conduct the "planning waves" anymore and thus a path around those regions will be found.

5.2.4 Implementation in a Numerical Proof-of-Concept

To substantiate the presented conceptual ideas, we performed numerical experiments using multiple different setups. In each case, the implementation of the model employs two neural networks that both represent the same manifold of stimuli.

The continuous attractor layer is a sheet of neurons that models the functionality of a network of place cells in the human hippocampus [O'Keefe and Dostrovsky, 1971, O'Keefe, 1976]. Each neuron is implemented as a rate-coded cell embedded in

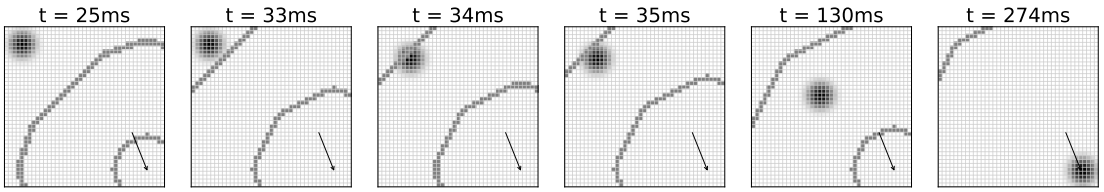


Figure 5.4: Activity in the wave propagation layer (greyish lines) and the continuous attractor layer (circular blob-like structure) overlaid on top of each other at different time points during the simulation. The grid signifies the neural network structure, i. e. every grid cell in the visualization corresponds to one neuron in each, the wave propagation layer and the continuous attractor layer. The position of the external wave propagation layer stimulation (to-be state) is shown with an arrow. Starting from an initial position in the top left of the sheet, the activation bump traces back the incoming waves to their source in the bottom right.

its neighborhood via short-range excitatory and long-range inhibitory connections as in [Guanella et al., 2007]. This structure allows the formation of a self-sustaining "bump" of activity, which can be shifted through the network by external perturbations. The bump represents the as-is state of the planning problem, which is to be solved by moving the bump to its target state.

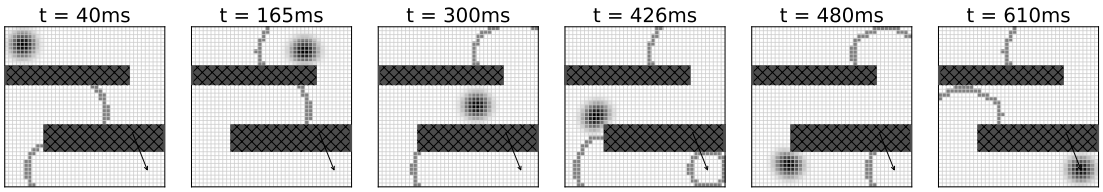
The wave propagation layer is constructed with an identical number of excitatory and inhibitory Izhikevich neurons [Izhikevich, 2003, Izhikevich, 2004], properly connected to allow for stable signal propagation across the manifold of stimuli. The target node is permanently stimulated, causing it to emit waves of activation which travel through the network.

The interaction between the two layers is modeled in a rather simplistic way. As in [Guanella et al., 2007], a time-dependent direction vector was introduced in the synaptic weight matrix of the continuous attractor layer. It has the effect of shifting the synaptic weights in a particular direction which in turn causes the location of the activation bump in the attractor layer to shift to a neighbouring neuron. The direction vector is updated whenever a wave of activity in the wave propagation layer newly enters the region which corresponds to the bump in the continuous attractor layer. Its direction is set to point from the center of the bump to the center of the overlap area between bump and wave, thus causing a shift of the bump towards the incoming wave fronts.

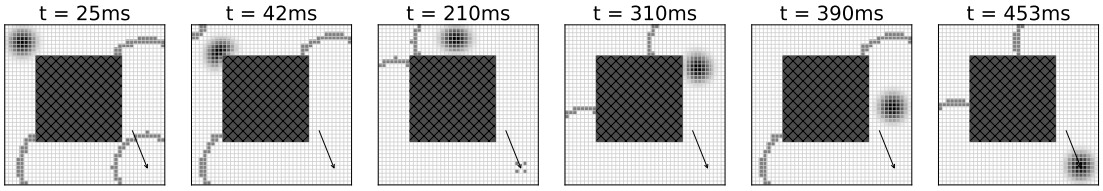
For more details on the implementation, see Section 5.3 below.

5.2.5 Results of the Numerical Experiments

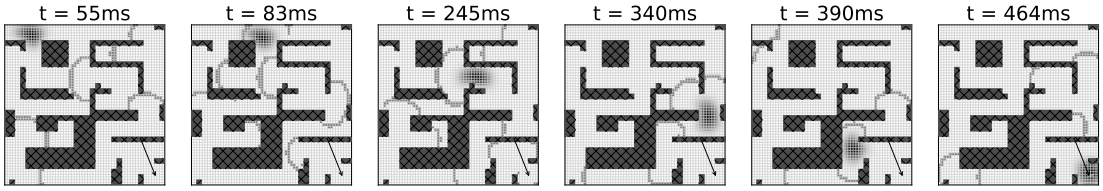
In a very simple initial configuration, the path finding algorithm was tested on a fully populated quadratic grid of neurons as described before. Figure 5.4 shows snapshots of wave activity and continuous attractor position at some represen-



(a) S Maze



(b) Block Maze



(c) Complex Maze

Figure 5.5: Simulations where specific portions of the neural layers were blocked for traversal (dark hatched regions) show the model's capability of solving complex planning problems. Note, that especially in the very fine structure of Figure 5.5c leftover excitation can trigger waves apparently spontaneously in the simulation region, such as at the right center at $t = 83\,\mathrm{ms}$. As the corresponding neurons are not constantly stimulated, these are usually singular events that do not disturb the overall process.

tative time points during the simulation. As expected, stimulation of the wave propagation layer in the lower right of the cognitive map causes the emission of waves, which in turn shift the bump in the continuous attractor layer from its starting position in the upper left towards its target state.

As described in Section 5.2.3, the manifold of stimuli represented by the neural network can be curved, branched, or of different topology, either permanently or temporarily. The purpose of the model is to allow for a reliable solution to the underlying graph traversal problems independent of potential obstacles in the networks. For this reason we investigated whether the bump of activation in the continuous attractor layer was able to successfully navigate through the graph from the starting node to the end node in the presence of nodes that could not be traversed. To test this idea we constructed different "mazes", blocking off sections of the graph by zeroing the synaptic connections of the respective neurons in the wave propagation layer and by clamping activation functions of the corresponding neurons in the continuous attractor layer to zero, see Figure 5.5. We found that in

all these setups, the algorithm was able to successfully navigate the bump in the continuous attractor layer through the mazes.

5.2.6 Relation to Existing Graph Traversal Algorithms

To conclude this section, we highlight a few parallels between the presented approach and the classical Breadth-First Search (*BFS*) algorithm.

BFS begins at some start node s of the graph and marks this node as "visited". In each step, it then chooses one node which is "visited" but not "finished" and checks whether there are still unvisited nodes that have an edge to this node. If so, the corresponding nodes are also marked as "visited", the current node is marked as "finished" and another iteration of the algorithm is started. For a more formal treatment of *BFS*, we refer to Section 5.3.

The approach presented here is a *parallelized* variant of this algorithm. Assuming that all neurons always obtain sufficient current to become activated, the propagating wave corresponds to the step of the algorithm in which the neighbors of the currently considered node are investigated. In contrast to *BFS*, the algorithm performs this step for all candidate nodes in a single step. That is, it considers *all* nodes currently marked as visited, checks the neighbors of all these nodes *at once* and marks them as visited if necessary. This close connection also allows to derive theoretical performance properties for the algorithm based on the behavior of *BFS*. As a more in-depth analysis of this connection is not within the scope of this paper, we refrain from going into detail here and refer again to Section 5.3.

Having all ingredients of the proposed conceptual framework in place, the following section reviews some experimental evidence indicating that it could in principle be employed by biological brains.

5.3 Methods and Experiments

Connection to Mathematical Graph Traversal Problems

As the model described in Section 5.2 uses a neural network of neurons to solve planning problems in the cognitive map, it is natural to interpret this network as a graph consisting of nodes representing the neurons and edges representing their synaptic connections. Thus, the planning problem in the network translates into a graph traversal problem in the corresponding graph. In the following, we hence introduce some basic terminology used in the field of graph theory.

We refrain from giving too many details and references, as most of the standard formalism can be found in classical books on mathematical optimization. In par-

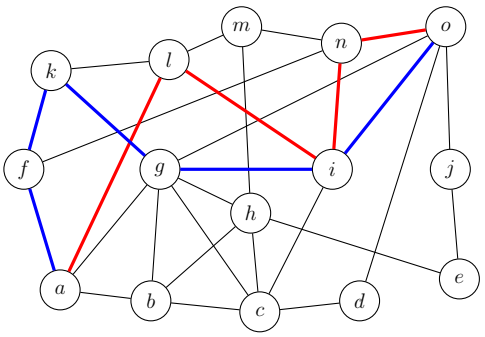


Figure 5.6: An example of a graph G with 15 nodes. The red resp. blue edges show two a-o-paths in the graph.

ticular, we refer to [Korte and Vygen, 2018, Schrijver, 2003] for references, details, proofs and further discussions.

A *graph* G is a pair $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of *nodes* and $\mathcal{E}$ is the set of *edges*, where each edge is set of two nodes. For two nodes $s, t \in \mathcal{V}$, an *s-t-path* is a path of nodes starting at s and ending at the target node t such that any two consecutive nodes along the path are connected by an edge. A node $t \in \mathcal{V}$ is *reachable* from another node $s \in V$ if there exists an s-t-path in G. An example of a graph with 15 nodes and different paths is given in Figure 5.6.

In the following, we let $G = (\mathcal{V}, \mathcal{E})$ be a fixed graph. For simplicity, we assume that every node is reachable from every node.

We are interested in finding a path between two given nodes $s, t \in \mathcal{V}$ in G. The idea is that the node t represents the neuron encoding the to-be state and s represents the neuron encoding the as-is state of the underlying planning problem. To formalize this problem, we denote by $\texttt{Path}(s, t)$ the problem of finding a path from s to t for given nodes s, t.

Even though this problem technically only asks for finding *some* path from s to t, shorter paths that use as few connections as possible are superior to longer paths using more connections. The reason is that the fewer connections a path has, the fewer intermediate states are traversed in the planning problem. When considering the previous example of grabbing a cup of coffee, a possible solution could be to move the arm around the head before performing actually reaching towards the coffee cup. This is not the movement that would be performed in actual behavior. However, we are similarly not obliged to find the shortest possible path. Considering the previous example again, a shortest path would reflect a movement with as few intermediate positions as possible. This might correspond to stretching the arm in such a way that the cup can barely be reached and might yield an unrealistic behavior. Thus, in summary, our goal is to find reasonably *short* paths that do not necessarily need to be *shortest* paths.

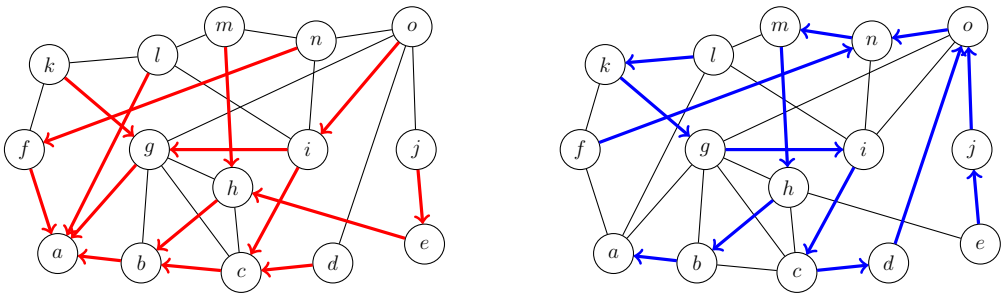


Figure 5.7: Exemplary result of $BFS(a)$ (left) and $DFS(a)$ (right). Each node but a points to its parent node.

The $\texttt{Path}(s,t)$ problem is a well-investigated problem in computer science and mathematics. With *BFS*and *DFS*, two standard path finding algorithms from computer science are described in Section 5.3. There, we also argue why these approaches cannot directly be applied to our scenario due to the fact that the graphs we consider represent neural networks in which algorithms have to be performed in a biological plausible way.

Mathematical Background and Solving the Path Problem in Typical Graphs

In the following, we consider how the $\texttt{Path}(s,t)$ problem can be solved in general graphs that do not represent neural networks. We later discuss what problems occur when trying to adapt these algorithms to such graphs when using the neurons as computational substrate. In all of the following, we omit technical details and proofs and instead refer to [Schrijver, 2003, Korte and Vygen, 2018] again.

Consider some fixed graph $G = (\mathcal{V}, \mathcal{E})$ and two nodes $s, t \in \mathcal{V}$. For simplicity, we assume that there is at least one path between any pair of nodes. The most basic class of algorithms that can be used to solve the $\texttt{Path}(s,t)$ problem is the class of *graph search algorithms*. Two of the most prominent examples of graph search algorithms are *Breadth-First-Search* (*BFS*) and *Depth-First-Search* (*DFS*).

Both algorithms start at the starting node s and traverse the graph iteratively by following its edges. Intuitively, $DFS(s)$ tries to follow a single path starting in s for as long as possible, only returning to a previously considered node and starting a "new" path if it is strictly necessary. In contrast to this, $BFS(s)$ tries to always visit a node as close as possible to the starting node s next. A visualization of the results of these two algorithms applied to the same graph starting at node a is given in Figure 5.7. By remembering which nodes are already completely explored, both of these algorithms find all nodes reachable from the starting node s.

In particular, the algorithms are typically implemented in a way that the traversed paths can easily be recovered from the data produced by the algorithms.

As both algorithms are very similar, they can be implemented as a realization of a general scheme for finding paths in a graph. This scheme is given in Algorithm 1. It uses a generic data structure D that only has to allow for the two basic operations of inserting in and removing nodes from it. In each step, the algorithm extracts a node u from D and checks for unvisited nodes among all nodes which have an edge towards u. For each such node w, the algorithm inserts w into the data structure D and remembers that the node w was reached from u by marking u as the parent of w. To avoid visiting vertices more than once, the node w is then also marked as visited. After performing this step for each such node, the node u is completely explored and it is not necessary to consider it again.

Depending on the specific data structure that is chosen for D, this then yields either the *BFS*or the *DFS* algorithm. More precisely, if D is chosen as a queue that inserts and removes nodes *first-in-first-out*, then Algorithm 1 yields the *BFS*algorithm. If D is chosen as a stack that inserts and removes nodes *last-in-first-out*, then one obtains the *DFS* algorithm.

```
   Data: graph G = (V, E). node s ∈ V
   Result: calculated parent p(w) for each w ∈ V
 1 D := (s)
 2 p(s) := s
 3 Mark s as visited
 4 while D ≠ ∅ do
 5 |   u := ExtractElement(D)
 6 |   foreach w that has an edge to u do
 7 |   |   if w is not marked as visited then
 8 |   |   |   Insert w into D
 9 |   |   |   p(w) := u
10 |   |   |   Mark w as visited
11 return list of parents p
```

Algorithm 1: The generic graph traversal algorithm. Choosing a queue for D yields the *BFS*algorithm, choosing a stack yields the *DFS* algorithm.

Both variants of this algorithmic scheme can solve the $\texttt{Path}(s,t)$ problem. However, as mentioned in Section 5.2, we want to find a *short* path from s to t. This is guaranteed if we use the $BFS(s)$ algorithm as this algorithm always finds shortest paths with respect to the number of edges. We later argue why this result implies that we are able to find short paths in the neural network representing the manifold of stimuli, even if we cannot guarantee that they are shortest paths.

We now discuss why it is not biologically plausible that graph traversal problems in the brain are solved by exactly one of these algorithms. The main obstacle is that Algorithm 1 requires the data structure D to organize the nodes that still have to be considered, as well as a mechanism to remember which nodes have

already been visited. Especially the data structure D which might have to store a large number of nodes and is in some sense "global" cannot be implemented in the brain in a way it can be implemented in a computer. The reason is that individual neurons in a neural network can only access local information or information that was just sent to them by a pre-synaptic neuron. In a neural network, however, neurons are only able to communicate with their synaptic neighbors via sending and receiving electric current.

As discussed in Section 5.2.6, our network configuration yields a wave propagation that behaves like a "parallelized" version of *BFS*where a set of nodes can be visited simultaneously. This also explains how using a wave propagation algorithm can find short paths, but not necessarily shortest paths: A neuron potentially receives current from more than one neuron, hence it is not possible to uniquely retrace the path to the starting node. However, as wave propagation behaves like a parallelized *BFS*algorithm, the paths that can be obtained via backtracking will never be too long. Although this behavior has some similarities with other well-understood graph problems like virus propagation [Bonnet et al., 2017, Kephart and White, 1991, Van Mieghem et al., 2009] or diffusion processes [Ibe, 2013] in networks, the respective theories are not directly applicable to our specific scenario.

Neuronal Network Setup – Exemplary Implementation of the Model

Splitting Dynamics to Two Network Layers As described in Section 5.2.2, for our numerical implementation of the model, we separated the two different types of dynamics into distinct layers of neurons, the *continuous attractor layer* and the *wave propagation layer*. The split into two layers makes the model more transparent and ensures that parameter changes have limited and traceable effects on the over-all dynamics. As an additional simplification, we do not explicitly model the feed-forward connections which drive the wave propagation layer, but we rather directly activate certain neurons in this layer.

Activation in the *continuous attractor layer* C represents the start node s, that in the course of the simulation will move towards the target node t, which is permanently stimulated in the *wave propagation layer* P. Waves of activation are travelling from t across P. As soon as the wave front reaches a node in P that is connected to a node in proximity to the current activation in C, the activation in C is moved towards it. Thus, every arriving wave front will pull the activation in C closer to t, forcing the activation to trace back the wave propagation to its origin t.

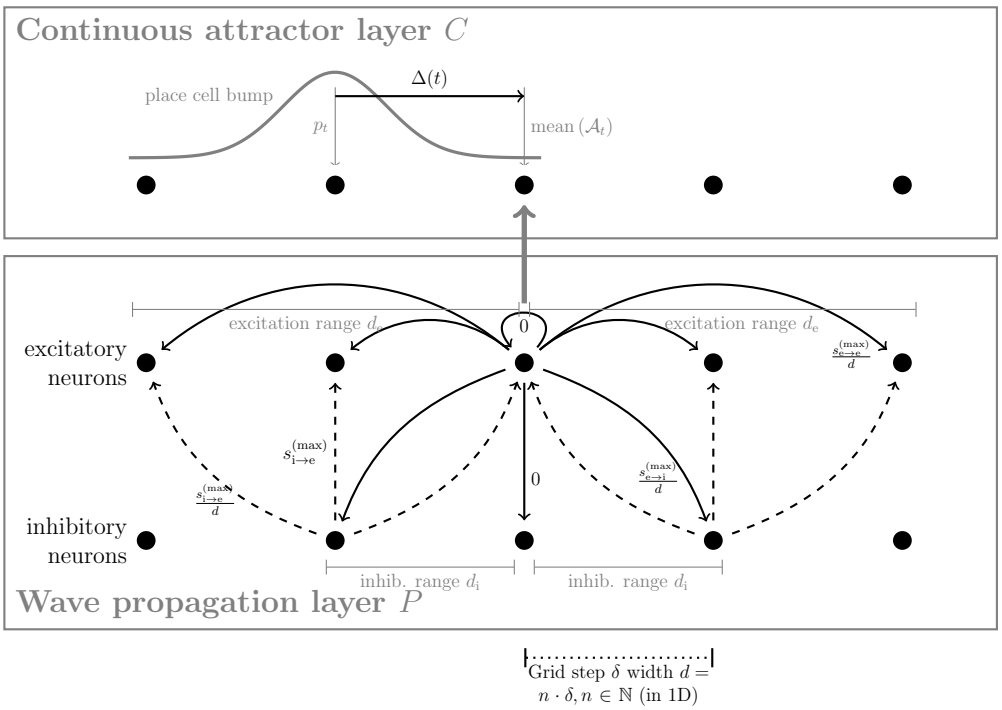


Figure 5.8: Connectivity of the neurons. For simplicity, this visualization only contains a 1D representation. In the wave propagation layer, excitatory synapses are drawn as solid arrows, dashed arrows indicate inhibitory synapses. Upon its activation, the central excitatory neuron stimulates a ring of inhibitory neurons that in turn suppress circles of excitatory neurons to prevent an avalanche of activation and support a circular wave-like expansion of the activation across the sheet of excitatory neurons. Furthermore, overlap between the active neurons in C and P is used to compute the direction vector $\Delta(t)$ used for biasing synapses in C and thus shifting activity there.

In detail, these dynamics require a very specific network configuration which is described in the following. Figure 5.8 contains a general overview of the intra- and inter-layer connectivity used in the model and our simulations.

Spiking Neuron Model in the Wave Propagation Layer In the performed experiments, the wave propagation layer P is constructed with an identical number of excitatory and inhibitory Izhikevich neurons [Izhikevich, 2003, Izhikevich, 2004], that cover a regular quadratic grid of 41×41 points on the manifold of stimuli.

The spiking behavior of each artificial neuron is modeled as a function of its membrane potential dynamics $v(t)$ using the two coupled ordinary differential equations $\frac{\mathrm{d}}{\mathrm{d}t}v = 0.04v^2 + 5v + 140 - u + I$ and $\frac{\mathrm{d}}{\mathrm{d}t}u = a \cdot (bv - u)$. Here, v is the membrane potential in mV, u an internal recovery variable, and I represents synaptic or DC input current. The internal parameters a (scale of u / recovery speed) and b (sensitivity of u to fluctuations in v) are dimensionless. Time t is measured in ms. If the membrane potential grows beyond the threshold param-

	excitatory RS	inhibitory FS
a	0.02	0.1
b	0.2	0.2
c	-65	-65
d	8	2

(a) Neuron model parameters (homogeneous setup).

	excitatory RS . . . CH	inhibitory LTS . . . FS
a	0.02	$0.02 + 0.08 r_i$
b	0.2	$0.25 - 0.05 r_i$
c	$-65 + 15 r_e^2$	-65
d	$8 - 6 r_e^2$	2

(b) Neuron model parameters (heterogeneous setup).

$s_{\mathrm{e}\to\mathrm{e}}^{(\max)}$	50
$s_{\mathrm{e}\to\mathrm{i}}^{(\max)}$	$0.5\, s_{\mathrm{e}\to\mathrm{e}}^{(\max)}$
$s_{\mathrm{i}\to\mathrm{e}}^{(\max)}$	$-9\, s_{\mathrm{e}\to\mathrm{e}}^{(\max)}$
d_{e}	2

(c) Synaptic strength parameters.

Table 5.1: Parameters used in our simulations of the wave propagation layer P.

eter $v \geq 30\,\mathrm{mV}$, the neuron is spiking and the variables are reset via $v \leftarrow c$ and $u \leftarrow u + d$. Again, c (after-spike reset value of v) and d (after-spike offset value of u) are dimensionless internal parameters.

If not stated otherwise in the following, the parameters listed in Table 5.1a were used for the spiking neuron model in P. They correspond to regular spiking (RS) excitatory and fast spiking (FS) inhibitory neurons. In contrast to [Izhikevich, 2003], neuron properties were not randomized to allow for reproducible analyses. The effect of a more biologically plausible heterogeneous neuron property and synaptic strength distribution is analyzed under *Numerical Experiments* below. Compared to [Izhikevich, 2003], the coupling strength in P is large to account for the extremely sparse adjacency matrix as every neuron is only connected to its few proximal neighbours in our configuration. Whenever a neuron in P is to be stimulated externally, a DC current of $I = 25$ is applied to it. As in [Izhikevich, 2003], the simulation time step was fixed to 1 ms with one sub-step in P for numerical stability.

Synaptic Connections in the Wave Propagation Layer As described before, neurons in P correspond to reachable locations in the manifold of stimuli. Thus, it is plausible to assume that neurons representing near-by locations in a suitable metric on the respective manifold will also be closely connected. Assuming that neurons will not have a very strictly defined region of responsibility, but there will also be some overlap, this is consistent with a Hebbian learning approach: Neurons that are sensitive to nearby regions will often fire at the same instant in time, strengthening their mutual connectivity.

As depicted in Figure 5.8, the excitatory neurons are driving nearby excitatory and inhibitory neurons with a synaptic strength of

$$s_{\mathrm{e}\to\mathrm{e}}(d) \coloneqq \begin{cases} \dfrac{s_{\mathrm{e}\to\mathrm{e}}^{(\max)}}{d}, & \text{for } 0 < d \leq d_{\mathrm{e}} \\ 0, & \text{else} \end{cases}, \tag{5.1}$$

where $s_{\mathrm{e}\to\mathrm{i}}(d)$ is defined analogously. Here, d is the distance between nodes in the manifold of stimuli. For simplicity, we model this manifold as a two-dimensional quadratic mesh with grid spacing $\delta = 1$ where some connections might be missing. The choice $s \propto 1/d$ was made to represent the assumption that recurrent coupling will be strongest to nearest neighbours and will decay with distance. Note that (5.1) in particular implies that we have $s_{\mathrm{e}\to\mathrm{e}}(0), s_{\mathrm{e}\to\mathrm{i}}(0) = 0$, which prevents self-excitation. To restrict to only localized interaction, we exclude interaction beyond a predefined excitation range d_{e} and inhibition range d_{i}, respectively. Values of the parameters in the expressions for the synaptic strengths used in the simulations are given in Table 5.1c.

The inhibitory neurons suppress activation of the excitatory neurons by reducing their input current via synaptic strength

$$s_{\mathrm{i}\to\mathrm{e}}(d) \coloneqq \begin{cases} s_{\mathrm{i}\to\mathrm{e}}^{(\max)}, & \text{for } d = 0 \\ \dfrac{s_{\mathrm{i}\to\mathrm{e}}^{(\max)}}{d}, & \text{for } 0 < d \leq d_{\mathrm{s}} \\ 0, & \text{else} \end{cases}. \tag{5.2}$$

Wave Propagation Dynamics The described setup allows for wave-like expansion of neuronal activity from an externally driven excitatory neuron as shown in Figure 5.9.

If the activity of the excitatory neurons grows too much in a region, the respective inhibitory neurons will start spiking to eventually suppress activity locally. This suppression happens with a delay of two time steps due to the causal signal travelling time through $s_{\mathrm{e}\to\mathrm{e}}$ and $s_{\mathrm{e}\to\mathrm{i}}$, but could also be implemented via different synaptic time constants, i. e. AMPA (excitatory) vs. GABA A (inhibitory). Thus, the inhibitory neurons prevent an avalanche-like activity by turning off active excitatory neurons.

As can be seen in Figure 5.9, this effectively means that propagating signals in the excitatory sub-network are followed by similarly shaped propagating signals in the inhibitory sub-network. In this respect, signal propagation does not behave like physical waves, such as ripples on water: They do not interfere in constructive and

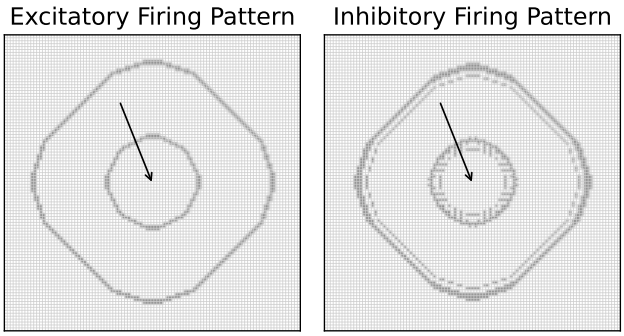


Figure 5.9: Activity patterns of the excitatory and inhibitory neurons on a 101×101 quadratic neuron grid. Spiking neurons are shown as gray areas. One excitatory neuron at the grid center (arrow) is driven by an external DC current to regular spiking activity. Due to the nearest-neighbour connections, this activity is propagating in patterns that resemble a circular wave structure. The inhibitory neurons prevent catastrophic avalanche-like dynamics by suppressing highly active regions. The specific pattern shape is an artifact of the underlying regular grid structure and thus not perfectly circular. This could be alleviated using, e. g. a hexagonal instead of a quadratic mesh of neurons.

destructive manner to form interference patterns. Instead, activity stops where to propagating signals touch as shown in Figure 5.10. This is an important property in our setup as it ensures that signals do not run through each other in the wave propagation layer but do mutually annihilate. Thus, the wave fronts tend to form stable and continuous patterns and activation of the continuous attractor layer from different directions is vastly reduced.

With the capability of propagating signals as circular waves from the target neuron t across the manifold of stimuli in P, it is now necessary to set up a representation of the start neuron s in C. This will be done in the following subsection before the coupling between P and C will be described.

Neuron Model for Place Cell Dynamics The *continuous attractor layer* C, implements a sheet of neurons that models the functionality of a network of place cells in the human hippocampus using rate-coding neurons [O'Keefe and Dostrovsky, 1971, O'Keefe, 1976] and thus the manifold of stimuli. As for the wave propagation layer, we also use a quadratic 41×41 grid of neurons for this layer. Activation in the continuous attractor layer will appear as bump, the center of which represents the most likely current location on the manifold of stimuli.

This bump of activation is used to represent the current position in the graph of synaptic connections representing the cognitive map. Planning in the manifold of stimuli thus amounts to moving the bump through the sheet of neurons where each neuron can be thought of as one node in this graph. With respect e. g. to

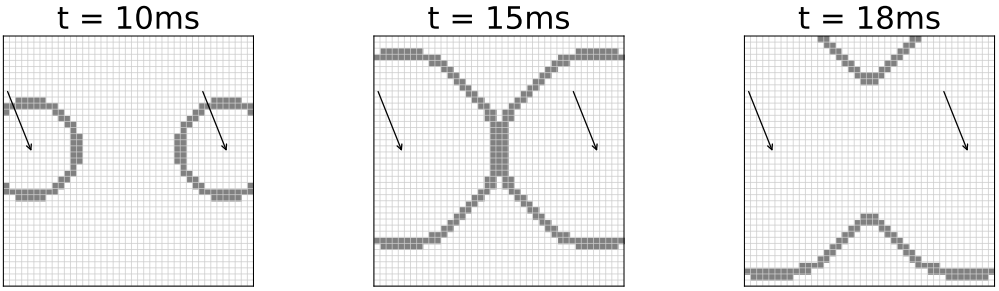


Figure 5.10: Activity patterns of the excitatory neuron grid where two neurons are driven to periodic spiking activity (arrows) at different instants in time. Again, spiking neurons are shown as gray areas and neuronal connections are set up as described in Section 5.3. As soon as the signal propagation fronts touch, they annihilate each other due to the inhibitory activity that accompanies them. Instead of forming interference patterns or travelling through each other, the remaining wave fronts merge and continue propagating as a well-defined line of activity.

the robot arm example in Figure 5.1b, the place cell bump represents the current state of the system i. e. the current angles of the arm's two degrees-of-freedom. As the bump moves through the continuous attractor layer, and thus through the graph, the robot arm will alter its configuration creating a movement trajectory through the 2D space.

Synaptic Connectivity to Realize Continuous Attractor Dynamics Our methodology for modelling the continuous attractor place cell dynamics adapts the computational approach used in [Guanella et al., 2007] by including a computational consideration for synaptic connections between continuous attractor neurons and an associated update rule that depends on information from the wave propagation layer P.

The synaptic weight function connecting each neuron in the continuous attractor sheet to each other neuron is given by a weighted Gaussian. This allows for the degrading activation of cells in the immediate neighbourhood of a given neuron and the simultaneous inhibition of neurons that are further away, thus giving rise to the bump-shaped activity in the sheet itself. The mathematical implementation of these synaptic connections also allows for the locus of activation in the sheet to be shifted in a given direction which is, in turn, how the graph implemented by this neuron sheet is able to be traversed.

The synaptic weight $w_{\vec{i},\vec{j}} \in \mathbb{R}^{(N_x \times N_y)\times(N_x \times N_y)}$ connecting a neuron at position $\vec{i} = (i_x, i_y)$ to a neuron at position $\vec{j} = (j_x, j_y)$ is given by

$$w_{\vec{i},\vec{j}} := J \cdot \exp\left(-\frac{1}{\sigma^2}\left\|\left(\frac{i_x - j_x}{N_x}, \frac{i_y - j_y}{N_y}\right) + \vec{\Delta}(t)\right\|^2\right) - T\,. \tag{5.3}$$

σ	Gaussian width	0.03
T	Gaussian shift	0.05
J	Synaptic connection strength	12
τ	Stabilization strength	0.8

Table 5.2: Parameters for the continuous attractor layer C.

Here, J determines the strength of the synaptic connections, $\|\cdot\|$ is the Euclidean norm, σ modulates the width of the Gaussian, T shifts the Gaussian by a fixed amount, $\vec{\Delta}(t)$ is a direction vector which we discuss in detail later, and N_x and N_y give the size of the two dimensions of the sheet.

In order to update the activation of the continuous attractor neurons and to subsequently move the bump of activation across the neuron sheet, we compute the activation $A_{\vec{j}}$ of the continuous attractor neuron $\vec{j}$ at time $t+1$ using

$$B_{\vec{j}}(t+1) = \sum_{\vec{i}} A_{\vec{i}}(t) w_{\vec{i},\vec{j}}(t)\,, \tag{5.4}$$

$$A_{\vec{j}}(t+1) = (1-\tau)B_{\vec{j}}(t+1) + \tau \frac{B_{\vec{j}}(t+1)}{\sum_{\vec{i}} A_{\vec{i}}(t)}\,, \tag{5.5}$$

where $B_{\vec{j}}(t+1)$ is a transfer function that accumulates the incoming current from all neurons to neuron $\vec{j}$ and τ is a fixed parameter that determines stabilization towards a floating average activity.

Simulation parameters for the continuous attractor layer C are given in Table 5.2. They have been manually tuned to ensure development of stable, Gaussian shaped activity with an effective diameter of approximately twelve neurons in C.

As in [Guanella et al., 2007], a direction vector $\vec{\Delta}(t) \in \mathbb{R}^2$ has been introduced in Equation (5.3). It has the effect of shifting the synaptic weights in a particular direction which in turn causes the location of the activation bump in the attractor layer to shift to a neighbouring neuron. In other words, it is this direction vector that allows the graph to be traversed by informing the place cell sheet from which direction the wave front is coming in P. Thus all that remains for the completion of the necessary computations is to compute $\vec{\Delta}(t)$ as a function of the propagating wave and the continuous attractor position.

Layer Interaction - Direction Vector The interaction between the wave propagation layer P and the continuous attractor layer C is mediated via the direction vector $\vec{\Delta}(t)$. The direction vector is computed such that it points from the center of the bump of activity towards the center of the overlap between bump and incoming wave as follows. Let $\mathcal{C}_t$ and $\mathcal{P}_t$ denote the sets of positions of active neurons at time t in layer C and P, respectively. Note that each possible position

corresponds to exactly one neuron in the wave propagation layer and exactly one neuron in the continuous attractor layer as they have the same spatial resolution in the implementation. Now let $\mathcal{A}_t \coloneqq \mathcal{C}_t \cap \mathcal{P}_t$. Then,

$$\text{mean}\left(\mathcal{A}_t\right) = \frac{1}{|\mathcal{A}_t|} \sum_{\vec{i} \in \mathcal{A}_t} \vec{i} \tag{5.6}$$

is the average position of overlap. We compute the direction vector from the current position p_t of the central neuron in the continuous attractor layer activation bump to $\text{mean}\left(\mathcal{A}_t\right)$ via

$$\vec{\Delta}(t) = \text{mean}\left(\mathcal{A}_t\right) - p_t\,. \tag{5.7}$$

Layer Interaction - Recovery Period In order to prevent the wave from interacting with the back side of the bump in C and thus pulling it back again, we introduce a recovery period R of a few time steps after moving the bump. During R, which is selected as the ratio of bump size to wave propagation speed, $\mathcal{A}_t$ is assumed to be empty, which prevents any further movement. In our experiments, we used $R = 12\,\text{ms}$. As the bump had a diameter of eleven cells and the maximum wave propagation speed was one cell per ms, this allowed every wave front to interact with the bump at most once.

It is worth acknowledging at this point that this approach of connecting the two layers, which we have chosen for reasons of simplicity, is somewhat artificial. We discuss this and other limitations of our implementation in Section 5.5.2.

Numerical Experiments

In order to test the complex neuronal network configuration described in the previous sections and to study its properties and dynamics, we performed numerical experiments using multiple different setups. [3] Results of our simulations are presented in Section 5.2.5. In the following, we will add some more in-depth analyses on specific properties of the model as observed in the simulations.

Transmission Velocity In our setup, no synaptic transmission delay, as e. g. in [Izhikevich, 2006], is implemented. As, due to the strong nearest-neighbour connectivity, only few pre-synaptic spiking neurons are sufficient to raise the membrane potential above threshold, the waves are travelling across P with a velocity of approximately one neuronal "ring" per time step, cf. Figure 5.4. In contrast, the continuous attractor can only move a distance of at most half its width per

[3] Source code used for our studies is published at `https://github.com/emdgroup/brain_waves_for_planning_problems`.

incoming wave. Accordingly, its velocity is tightly coupled to the spike frequency of the stimulated neuron while still being bound due to the recovery period R. In the specific case of the simulation in Figure 5.4, in total nine wave fronts were observed to be required traversing the Gaussian continuous attractor activity zone to finally pull it on a straight line to its destination over a distance of $d = 45.25$.

Obstacles and Complex Setups In the S-shaped maze Figure 5.5a, the continuous attractor activity moves towards the target node t on a direct path around the obstacles. Due to the optimal path being more than two times longer than in Figure 5.4, the time to reach the target is accordingly longer as well. This is also in line with the required travel times from s to t in Figures 5.5b and 5.5c, where – despite its complexity – a path through the maze is found fastest due to it being shorter than in the other cases of Figure 5.5. This observation is also evidenced by the fact that our model is a parallelized version of *BFS*, cf. Sections 5.2.6 and 5.3, which is guaranteed to find the shortest path in an unweighted and undirected graph.

Symmetric Paths An additional interesting observation can be made in the central block setup, Figure 5.5b: The setup is perfectly symmetric with respect to the two possible paths. Thus, in principle it can not be solved with our model. However, after interaction with several wave fronts, a minor shift of the continuous attractor position occurs due to numerical instability. This is further emphasized by subsequent incoming waves, finally pulling the continuous attractor onto a path to the target node t. While such numerical instabilities are clearly resulting from the specific implementation of our model on a computer system, also organically grown biologic networks will never be perfectly symmetric. Here, natural variations in synaptic connectivity and neuron properties will break potential symmetries, favoring one of the possible paths. In the following, we inspect the influence of these variations on the overall performance of the model.

Heterogeneous Neuron Properties and Synaptic Strengths In the simulation experiments described up to now, a homogeneous wave propagation layer P is employed. There, all neurons are subject to the same internal parameters, being either regular spiking excitatory neurons or fast spiking inhibitory neurons. Also, synaptic strengths are strictly set as described previously with parameters from Table 5.1c. This setup is rather artificial. Natural neuronal networks will exhibit a broad variability in neuron properties and in the strength of synaptic connectivity.

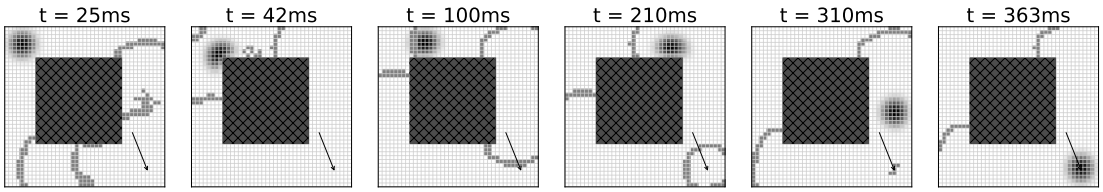


Figure 5.11: Block setup as in Figure 5.5 but with a heterogeneous neuron configuration in P.

To account for this natural variability, we randomized the individual neuron's internal properties as suggested in [Izhikevich, 2003], see Table 5.1b. As in [Izhikevich, 2003], heterogeneity is achieved by randomizing neuron model parameters using random variables r_e and r_i for each excitatory and inhibitory neuron. These are equally distributed in the interval $[0; 1]$ and vary neuron models between regular spiking ($r_e = 0$) and *chattering (CH, $r_e = 1$)* or fast spiking ($r_i = 1$) for excitatory neurons and *low-threshold spiking (LTS, $r_i = 0$)* for inhibitory neurons. By squaring r_e, excitatory neuron distribution is biased towards RS. In addition, after initializing synaptic strengths in P, we randomly varied them individually by up to $\pm 10\,\%$.

Despite this strong modification to the original numerically ideal setup, a structured wave propagation is still possible in P as can be seen in Figure 5.11. While the stereotypical circular form of the wave fronts dissolves in the simulation, they continue to traverse P completely. As before, they reach the continuous attractor bump and are able to guide it to their origin. Apparently, the overall connection scheme in P is more important for stable wave propagation than homogeneity in the individual synaptic strengths and neuron properties.

An interesting aspect of this simulation when compared to Figure 5.5b is the apparent capability of solving the graph traversal problem quicker than with the homogeneous neuronal network. As already indicated, this is an artifact of the explicitly broken symmetry in the heterogeneous configuration: The wave fronts from different directions differ in shape when arriving at the initial position of the continuous attractor layer activity. Thus, one of them is immediately preferred and target-oriented movement of the bump starts earlier than before. This capability of breaking symmetries and thus quickly resolving ambiguous situations is an explicit advantage of the more biologically realistic heterogeneous configuration.

5.4 Empirical Evidence

In this section we review empirical findings which are relevant for the model. We dedicate one subsection to each of several key model assumptions on the neural connectivity and dynamics.

5.4.1 Cognitive Maps

The concept of "cognitive maps" was first proposed by Edward Tolman [Tolman and Honzik, 1930, Tolman, 1948] who conducted experiments to understand how rats were able to navigate mazes to seek rewards. He noticed that these animals showed remarkably flexible behaviour when confronted with novel versions of their maze environments, such as finding previously unexplored shortcuts or finding new routes when obstructions made old ones untraversable. Tolman theorized that this behaviour was made possible by the rats having an internal model (or map) of the mazes which they used to navigate and which they updated when new information about the maze was presented.

A body of evidence suggests that neural structures in the hippocampus and enthorinal cortex potentially support cognitive maps used for spatial navigation [O'Keefe and Dostrovsky, 1971, O'Keefe and Nadel, 1978, Bush et al., 2015]. Within these networks, specific kinds of neurons are thought to be responsible for the representation of particular aspects of cognitive maps. Some examples are place cells [O'Keefe and Dostrovsky, 1971, O'Keefe and Nadel, 1978] which code for the current location of a subject in space, grid cells which contribute to the problem of locating the subject in that space [Hafting et al., 2005] as well as supporting the stabilisation of the attractor dynamics of the place cell network [Guanella et al., 2007], head-direction cells [Taube et al., 1990] which code for the direction in which the subject's head is currently facing, and reward cells [Gauthier and Tank, 2018] which code for the location of a reward in the same environment.

The brain regions supporting spatially aligned cognitive maps might also be utilized in the representation of cognitive maps in non-spatial domains: In [Constantinescu et al., 2016], fMRI recordings taken from participants while they performed a navigation task in a non-spatial domain showed that similar regions of the brain were active for this task as for the task outlined in [Doeller et al., 2010] where participants navigated a virtual space using a VR apparatus. Not only were the same spatial-task aligned regions active for this non-spatial-domain navigation task but the firing patterns of the neurons recorded in the former displayed the same hexagonal firing patterns that are characteristic of enthorinal grid-cells. Further, according to [Cameron et al., 2001], activation of neurons in the hippocampus (one of the principal sites for place cells) is indicative of how well participants were

able to perform in a task related to pairing words. Supporting this observation with respect to the role played by these brain regions in the operation of abstract cognitive maps, [Alvarez et al., 2002] found that lesions to the hippocampus significantly impaired performance on a task of associating pairs of odors by how similar they smelled. Finally, complementing these findings, rat studies have shown that hippocampal cells can code for components in navigation tasks in auditory [Aronov et al., 2017, Sakurai, 2002], olfactory [Eichenbaum et al., 1987], and visual [Fried et al., 1997] task spaces.

Taken as a whole, the above body of research provides good evidence for the following ideas: Firstly, that cognitive maps exist in humans. Secondly, that these maps can and are used for solving problems in a general class of task spaces. Thirdly, that hippocampal and enthorinal cells likely play a key role in the construction and operation of these maps.

5.4.2 Feed-Forward and Recurrent Connections

As described in Section 5.2.1, the proposed model is built around a particular *theme of connectivity*: Each neuron represents a certain pattern in sensory perception mediated via feed-forward connections. Such a pattern could be, for example, all the percepts associated with a particular position in a maze, a certain body posture, or some letter in the field of vision, cf. Figure 5.1. In addition, recurrent connections between two neurons strengthen whenever they are activated simultaneously. In the following, we give an overview of some relevant experimental observations which are consistent with this mode of connectivity.

The most prominent example of neurons which are often interpreted as pattern detectors are the cells in primary visual cortex. These neurons fire when a certain pattern (like a small edge of bright/dark contrast) is perceived at a particular position and orientation in the visual field. On the one hand, these neurons receive their feed-forward input from the lateral geniculate nucleus. On the other hand, they are connected to each other through a tight network of recurrent connections. Several studies (see e. g. [Ko et al., 2013, Iacaruso et al., 2017, Ko et al., 2011]) have shown that two such cells are preferentially connected when their receptive fields are co-oriented and co-axially aligned. Due to the statistical properties of natural images, where elongated edges appear frequently, such two cells can also be expected to be positively correlated in their firing due to feed-forward activation.

Similar statements are valid for auditory cortex: Neurons in primary auditory cortex receive feed-forward input from thalamocortical connections as well as intracortical signals via recurrent connections. The feed-forward input is tonotopically organized and A1 neurons typically respond to one or several characteristic

frequencies. There is evidence for a cross-frequency integration via intracortical input: For example, neurons in A1 show subthreshold responses to frequency ranges broader than can be accounted for by their thalamic inputs [Kaur et al., 2004] while the latency of their response is shortest at their characteristic frequency [Kaur et al., 2005]. Additional supporting facts are reviewed in the introduction of [Kratz and Manis, 2015]. By analogy from the visual cortex, one might expect that intra-cortical connections are strongest between neurons if their characteristic frequencies differ by a harmonic interval, e. g. by a full octave, since such intervals are most highly correlated in the frequency spectra of natural sounds [Abdallah and Plumbley, 2006a, Abdallah and Plumbley, 2006b]. Indeed, some support for this hypothesis is reviewed in [Wang, 2013]: Tracing the diffusion of a marker substance after local injection into cat auditory cortex shows that "the intrinsic connections of A1 arising from nearby cylinders of neurons are not homogenous and clusters of cells can be identified by their unique pattern of connections within A1" [Wallace et al., 1991]. In particular, horizontal connections displayed a periodic pattern along the tonotopic axis. In similar tracing experiments on cat A1 it was found that injections into a specific cortical location caused labeling at other A1 locations that were harmonically related to the injection site [Kadia et al., 1999].

The somatosensory cortex is another brain region where several empirical findings are in line with the postulated theme of connectivity. Area 3b in the somatosensory cortex contains neurons which respond to tactile stimuli. Their receptive fields are not dissimilar to those of cells in V1. Experiments on non-human primates suggest that "3b neurons act as local spatiotemporal filters that are maximally excited by the presence of particular stimulus features" [DiCarlo et al., 1998].

Regarding the recurrent connections in somatosensory cortex, some empirical support stems from the well-studied rodent barrel cortex. Here, the animal's facial whiskers are represented somatotopically by the columns of primary somatosensory cortex. Neighboring columns of the barrel cortex are connected via a dense network of recurrent connections. Sensory deprivation studies indicate that the formation of these connections depends on the feed-forward activation of the respective columns: If the whiskers corresponding to one of the columns are trimmed during early post-natal development, the density of recurrent connections with this column is reduced [Wallace and Sakmann, 2008, Broser et al., 2008]. Conversely, synchronous co-activation over the course of a few hours can lead to increased functional connectivity in the primary somatosensory cortex [Vidyasagar et al., 2014].

The primary somatosensory cortex also receives proprioceptive signals from the body which represent individual joint angles. Taken as a whole, these signals

characterize the current posture of the animal and there is an obvious analogy to the arm example, cf. Figure 5.1b. We are not aware of any experimental results regarding the recurrent connections between proprioception detectors, but it seems reasonable to expect that the results about processing of tactile input in the somatosensory cortex can be extrapolated to the case of proprioception. This would imply that a recurrent network structure roughly similar to Figure 5.1b should emerge and thus support the model for controlling the arm.

Area 3a of the somatosensory cortex, whose neurons exhibit primarily proprioceptive responses, is also densely connected to the primary motor cortex. It contains many corticomotoneuronal cells which drive motoneurons of the hand in the spinal cord [Delhaye et al., 2018]. This tight integration between sensory processing and motor control might be a hint that the hypothetical string-of-a-puppet muscle control mechanism from Section 5.2.3 is not too far from reality.

In summary, evidence from primary sensory cortical areas seems to suggest a common cortical theme of connectivity in which neurons are tuned to specific patterns in their feed-forward input from other brain regions, while being connected intracortically based on statistical correlations between these patterns.

5.4.3 Wave Phenomena in Neural Tissue

In the model we present, the target state of a cognitive planning task is encoded by localized activation within the cognitive map. Starting from there, neural activation travels through the recurrent network in what resembles expanding wave fronts.

There is a large amount of empirical evidence for different types of wave-like phenomena in neural tissue. We summarize some of the experimental findings, focusing on fast waves (a few tens of $\mathrm{cm\,s^{-1}}$). These waves are suspected to have some unknown computational purpose in the brain [Muller et al., 2018] and they seem to bear the most resemblance with the waves postulated in the model.

Using multielectrode local field potential recordings, voltage-sensitive dye, and multiunit measurements, traveling cortical waves have been observed in several brain areas, including motor cortex, visual cortex, and non-visual sensory cortices of different species. There is evidence for wave-like propagation of activity both in sub-threshold potentials and in the spatiotemporal firing patterns of spiking neurons [Sato et al., 2012].

In the motor cortex of wake, behaving monkeys, Rubino et al. [Rubino et al., 2006] observed wave-like propagation of local field potentials. They found correlations between some properties of these wave patterns and the location of the visual target to be reached in the motor task. On the level of individual neurons, Takahasi

et al. found a "spatiotemporal spike patterning that closely matches propagating wave activity as measured by LFPs in terms of both its spatial anisotropy and its transmission velocity" [Takahashi et al., 2015].

In the visual cortex, a localized visual stimulus elicits traveling waves which traverse the field of vision. For example, Muller et al. have observed such waves rather directly in single-trial voltage-sensitive dye imaging data measured from awake, behaving monkeys [Muller et al., 2014].

The detailed propagation mechanisms which lead to fast travelling waves in cortical tissue are still under discussion. The prevalent view seems to be that waves are actually propagated through the circuitry of the respective cortical area rather than, being the result of spatiotemporally organized activation that stems from some other brain region. Two competing mechanisms for waves [Sato et al., 2012] are: (1) strictly localized generation of activity followed by monosynaptic propagation through long-range horizontal connections of the superficial cortical layers or (2) a "chain reaction" of firing neurons leading to a self-sustaining spread of activity through the deeper cortical layers. While possibly both mechanisms play a role in the brain, only the second one is incorporated in the model.

5.4.4 Spatial Navigation Using Place Cells

Finding a short path through a maze-like environment, cf. Figure 5.1a, is one of the planning problems the model is capable of solving. In this case, each neuron of the continuous attractor layer represents a "place cell" which encodes a particular location in the maze.

Place cells were discovered by John O'Keefe and Jonathan Dostrovsky in 1971 in the hippocampus of rats [O'Keefe and Dostrovsky, 1971]. They are pyramidal cells that are active when an animal is located in a certain area ("place field"), of the environment. Place cells are thought to use a mixture of external sensory information and stabilizing internal dynamics to organize their activity: On the one hand, they integrate external environmental cues from different sensory modalities to anchor their activity to the real world. This is evidenced by the fact that their activity is affected by changes in the environment and that it is stable under a removal of a subset of cues [Barry et al., 2006, Jeffery, 2011]. On the other hand, firing patterns are then stabilized and maintained by internal network dynamics as cells remain active under conditions of total sensory deprivation [Quirk et al., 1990]. Collectively, the place cells are thought to form a cognitive map of the animal's environment.

In theoretical or computational studies, continuous attractor models are often used to describe place cell dynamics. Just as we do in the present article, it is typ-

ically assumed that each place cell responds, on the one hand, to location-specific patterns of sensory cues and, on the other hand, to stimulation via recurrent connections from cells with overlapping place fields.

5.4.5 Targeted Motion Caused by Localized Neuron Stimulation

In our model, the process of motion planning is triggered by stimulating the neurons which represent the body's to-be position, cf. Figure 5.1b. In the present section, we review some experimental results that support the biological plausibility of this assumption.

In 2002, Graziano et al. reported results from electrical microstimulation experiments in the primary motor and premotor cortex of monkeys [Michael S.A. Graziano and Moore, 2002]. Stimulation of different sites in the cortical tissue for a duration of 500 ms resulted in complex body motions involving many individual muscle commands. The stimulation of one particular site typically led to smooth movements with a certain end state, independent of the initial posture of the monkey, while stimulating a different location in the cortical tissue led to a different end state. In particular, Graziano et al. noted that stimulation at a fixed site can have the different effects in terms of low-level muscle commands: For example, a monkey's arm might either stretch or flex to reach a partially flexed position, depending on its initial condition. In terms of the model presented here, this would be explained by two wave fronts propagating in opposite directions away from the to-be location, only one of which hits the localized peak of activity encoding the as-is location and pulls it closer to the to-be state. Graziano et al. also reported that the motions stopped as soon as the electrical stimulus was turned off. This is fully consistent with our model, where stopping the to-be activation means that no more wave fronts are created and thus the as-is peak of activity remains where it is.

After this original discovery by Graziano et al. in 2002, several additional studies have confirmed and extended their results, see [Graziano, 2016] for an overview. Similar effects of motor cortex stimulation have been observed in a variety of different primate and rodent species. The results also hold true for different types of neural stimulation: electrical, chemical and optogenetic. The neural structures which cause the bodily motions towards a specific target state have been named *ethological maps* or *action maps* [Graziano, 2016].

Furthermore, several studies suggest that such action maps are shaped by experience: Restricting limb movements for thirty days in a rat can cause the action map to deteriorate. A recovery of the map is observed during the weeks after

freeing the restrained limb [Budri et al., 2014]. Conversely, a reversible local deactivation of neural activity in the action map can temporarily disable a grasping action in rats [Brown and Teskey, 2014]. A permanent lesion in the cortical tissue can disable an action permanently. The animal can re-learn the action, though, and the cortical tissue reorganizes to represent the newly re-learnt action at a different site [Ramanathan et al., 2006]. These observed plasticity phenomena are fully in line with our model which emphasises a self-organized formation of the cognitive map via Hebbian processes both for the feature learning and for the construction of the recurrent connections.

5.4.6 Participation of the Primary Sensory Cortex in Non-Sensory Tasks

For the first two examples in Figure 5.1, the association with a planning task is obvious. Our third example, the geometric transformations of the letter "A", may appear a bit more surprising, though: After all, the neural structures in visual sensory cortex would then be involved in "planning tasks". The tissue of at least V1 fits the previously explained theme of connectivity, but it is often thought of as a pure perception mechanism which aggregates optical features in the field of vision and thus performs some kind of preprocessing for the higher cortical areas.

However, there is evidence that the visual sensory cortex plays a much more active role in cognition than pure feature detection on the incoming stream of visual sensory information. In particular, the visual cortex is active in visual imagery, that is, when a subject with closed eyes mentally imagines a visual stimulus [Pearson, 2019]. Experiments suggest that mental imagery leads to activation patterns in the early visual cortex which are composed of the same visual features as during actual sensory perception: Using multi-voxel pattern classification on fMRI measurements of the visual cortex, it is possible to train machine learning models which can accurately decode cortical activation and determine which image in the field of vision has caused the neural response. The same models, trained only on perceptual images, have been used successfully to decode cortical activation caused by purely mental images [Naselaris et al., 2015].

Based on such findings, it has been suggested that "the visual cortex is something akin to a 'representational blackboard' that can form representations from either the bottom-up or top-down inputs" [Pearson, 2019]. In our model, we take this line of thinking one step further and speculate that the early visual cortex does not only represent visual features, but that it also encodes possible transformations like rotation, scaling or translation via its recurrent connections. In this view, the "blackboard" becomes more of a "magnetic board" on which mental im-

ages can be placed and shifted around according to rules which have been learned by experience.

Of course, despite the over-simplifying Figure 5.1c, we do not intend to imply that there were any neurons in the visual cortex with a complex pattern like the whole letter "A" as a receptive field. In reality, we would expect the letter to be represented in early visual cortex as a spatio-temporal multi-neuron activity pattern. The current version of our model, on the other hand, allows for single-neuron encoding only and thus reserves one neuron for each possible position of the letter. We will discuss this and other limitations of the proposed model in Section 5.5.

5.4.7 Temporal Dynamics

The concept presented in this article implies predictions about the temporal dynamics of cognitive planning processes which can be compared to experiments: The bump of activity only starts moving when the first wave front arrives. Assuming that every wave front has a similar effect on the bump, its speed of movement should be proportional to the frequency with which waves are emitted. Thus both the time until movement onset and the duration of the whole planning process should be proportional to the length of the traversed path in the cortical map. Increased frequency of wave emission should accelerate the process.

One supporting piece of evidence is provided by mental imagery: Experiments in the 1970s [Shepard and Metzler, 1971, Cooper and Shepard, 1973] have triggered a series of studies on mental rotation tasks, where the time to compare a rotated object with a template has often been found to increase proportionally with the angle of rotation required to align the two objects.

In the case of bodily motions, the total time to complete the cognitive task is not a well suited measure since it strongly depends on mechanical properties of the limbs. Yet for electrical stimulation of the motor cortex (cf. Section 5.4.5) Graziano et al. report that the speed of evoked arm movements increases with stimulation frequency [Graziano et al., 2005]. Assuming that this frequency determines the rate at which the hypothetical waves of activation are emitted, this is consistent with our model.

In addition, our model makes the specific prediction that the latency between stimulation and the onset of muscle activation should increase with the distance between initial and target posture. The reason is that the very first wave front needs to travel through the cognitive map before the bump of activation starts being shifted, and only then muscular activation can be triggered by the bump's deflection. The travel time of this wave front thus becomes an additive compo-

nent of the total latency and it can be expected to be roughly proportional to the distance between initial and target posture as measured in the metric of the cognitive map. We are not aware of any studies having examined this particular relationship yet.

5.5 Discussion

The model proposed here is, to the best of our knowledge, the first model that allows for solving graph problems in a biological plausible way such that the solution (i. e. the specific path) can be calculated directly on the neural network as the only computational substrate.

Similar approaches and models have been investigated earlier, especially in the field of neuromorphic computing. For example, in [Muller et al., 1996, Aimone et al., 2019, Aimone et al., 2021, Hamilton et al., 2019, Kay et al., 2020] graphs are modeled using neurons and synapses, and computations are performed by exciting specific neurons which induces propagation of current in the graph and observing the spiking behavior. Also, models using two or more cell layers and spiking neural neurons have been used for unsupervised learning of orientation, disparity, and motion representations [Barbier et al., 2021] or modeling the tactile processing pathway [Parvizi-Fard et al., 2021]. In addition, recurrent neural networks were recently also used to model and analyze working memory [Kim and Sejnowski, 2021, Xie et al., 2022] or image recognition tasks [Wang et al., 2022]. These models are however either designed for very specific tasks [Parvizi-Fard et al., 2021], do not guarantee a stable performance [Wang et al., 2022] or lack biological plausibility [Kim and Sejnowski, 2021, Barbier et al., 2021, Xie et al., 2022] . Furthermore, [Chen and Gong, 2019] describes another neural computation mechanism which "might be a general computational mechanism of cortical circuits" [Chen and Gong, 2019] using circuit models of spiking neurons. This mechanism is developed for understanding how spontaneous activity is involved in visual processing and is not investigated in terms of its applicability for solving planning problems.

Although some models are more general than the one presented here and allow for solving more complex problems like dynamic programs [Aimone et al., 2019], enumeration problems [Hamilton et al., 2019] or the longest shortest path problem [Kay et al., 2020], we are not aware of any model explicitly discussing the biological plausibility despite the need for more neurobiologically realistic models [Pulvermüller et al., 2021]. In fact, most of these approaches are far from being biologically plausible as they e. g. require additional artificial memory [Aimone et al., 2019] or a preprocessing step that changes the graph depending on the input data [Kay et al., 2020]. Also, the model of Muller et al. [Muller et al., 1996] as

well as the very recent model of Aimone et al. [Aimone et al., 2021] which are biologically more plausible do not discuss how a specific path can then be computed in the graph, even if the length of a path can be calculated [Aimone et al., 2021]. In addition, some models try to describe actually observed wave propagation in the brain [Galinsky and Frank, 2020b, Galinsky and Frank, 2020a].

Our model has not been created with the intent to explain empirical findings from one particular brain region, mental task or experimental technique in full detail. Rather, we sought to explore ways in which a generic algorithmic framework might solve seemingly very different problems based on more or less the same neural substrate. Working on a relatively high level of abstraction and ignoring most of the domain-specific aspects may not only help our understanding of computational principles employed by the brain but also pave the road to the development of new useful algorithms in artificial intelligence. Nevertheless, it is important to note that many features of our model are in line with experimental results from various areas of brain science and we review those findings in Section 5.4.

In the following we discuss limitations of the presented model and potential avenues for further research.

5.5.1 Single-Neuron vs. Multi-Neuron Encoding

In our model, each point on a cortical map is represented by a single neuron and a distance on the map is directly encoded in a synaptic strength between two neurons. The graph of synaptic connections an therefore be considered as a coarse-grained version of the underlying manifold of stimuli. Yet such a single-neuron representation is possible only for manifolds of a very low dimension, since the number of points necessary to represent the manifold grows exponentially with each additional dimension. For tasks like bodily movement, where dozens of joints need to be coordinated, the number of neurons required to represent every possible posture in a single-neuron encoding is prohibitive. Therefore, it is desirable to encode manifolds of stimuli in a more economical way – for example, by representing each point of the manifold by a certain set of neurons. It is an open question how distance relationships between such groups of neurons could be encoded and whether the dynamics from our model could be replicated in such a scenario.

5.5.2 Wave Propagation and Continuous Attractor Layers

Certain design choices made in the numerical implementation should be discussed regarding their biological plausibility and possible alternative mechanisms.

If the wave propagation layer and the continuous attractor layer were to form organically as two separate sub-circuits in a real biological system, each of their neurons would need to act as a feature detector, since otherwise it is not clear how the right structure of recurrent connections could develop. Then each feature will be represented by two detectors – one in each layer – and there must be some unknown mechanism which establishes a link between every pair of corresponding feature detectors.

Moreover, the split of the model dynamics into two layers leads to a somewhat artificial implementation of the interaction between them: As described in Section 5.2.4, we compute the direction into which the activation bump in the continuous attractor layer is shifted whenever a wave front arrives at the corresponding location in the wave propagation layer. The details of this mechanism do not appear to be biologically plausible and we would rather expect that the bump is moved only by the aggregated effects of local interactions between synaptically connected neurons. This interaction could be mediated by the connections between corresponding feature detectors which we postulated above.

Alternatively, an elegant and biologically plausible model could be obtained by merging the wave propagation and continuous attractor dynamics into a single layer of neurons. In such a single-layer model, the whole network can form in a self-organized way: First, the feed-forward connections are generated via a process of competitive Hebbian learning, leading to a network of individual feature detectors. In a second step, these detectors establish recurrent connections among each other, again driven by Hebbian learning, to create the graph structure required in our model.

In the single-layer scenario, the model must allow for continuous attractor dynamics and wave-like expansion of activity simultaneously. We speculate that this is possible in principle, for example by imposing a time delay on the effect of inhibition – which appears biologically plausible considering that it is mediated in an extra step via inhibitory interneurons. The time delay of inhibition has only minimal effect on the quasi-static peak of activity and thus conserves the landscape of continuous attractors from the two-layer scenario. On the other hand, the time delay allows activation patterns with strong temporal fluctuations to emit waves of activity before inhibition has any effect.

The interaction between the waves and the localized peak of activity could potentially shift the peak in the direction of the incoming waves without the need to impose any artificial assumptions to the model: The incoming waves are annihilated by the peak's "trench of inhibition", but they also increase the level of activation of the neurons on the side of the bump which is hit by the wave. Due to the attractor dynamics of the network the bump recovers from this deforma-

tion, but in the process it changes its location slightly towards the direction of the incoming wave.

Realizing the effects described above will require a very careful numerical set-up of the model and tuning of its parameters. We consider this an interesting direction for future research since the potential outcome is a rather elegant model with a high degree of biological plausibility.

5.5.3 Embedding into a Bigger Picture

While the model focuses on the solution of graph traversal problems, it appears desirable to embed it into a broader context of sensory perception, decision making, and motion control in the brain. One particular question is how the hypothetical "puppet string mechanism" – which we postulated to connect proprioception and motion control – could be implemented in a neural substrate. Similarly, if our model provides an appropriate description of place cells and their role in navigation, the question arises how a shift in place cell activity is translated into appropriate muscle commands to propel the animal in the corresponding direction.

It is intriguing to speculate about a deeper connection between our model and object recognition: On the same neural substrate, our hypothetical waves might travel through a space of possible transformations, starting from a perceived stimulus and "searching" for a previously learned representative of the same class of objects. This could explain why recognition of rotated objects is much faster than the corresponding mental rotation task [Corballis et al., 1978]: The former would require only one wave to travel through the cognitive map, while the later would require many waves to move the bump of activity.

And finally, an open question is the connection between the model and the hypothetical executive brain functions which are assumed to define the target state for the graph traversal problem and activate the corresponding neurons.

5.6 Conclusion

We have shown that a wide range of cognitive tasks, especially those that involve planning, can be represented as graph problems. To this end, we have detailed one possible role for the recurrent connections that exist throughout the brain as computational substrate for solving graph traversal problems. We showed in which way such problems can be modeled as finding a short path from a start node to some target node in a graph that maps to a manifold representing a relevant task space. Our review of empirical evidence indicates that a theme of connectivity can

be observed in the neural structure throughout (at least) the neocortex which is well suited to realize the proposed model.

We constructed a two-layer neural network consisting of a layer of neurons that implemented a continuous attractor sheet modelled after neurons found in the human hippocampus and enthorinal cortex. This sheet of neurons enacts a "bump" of activity centered on the neuron representing the start node s in the graph. As a second step we implemented a sheet of spiking neurons that generated a wave of activation across the same sheet starting from a individual neuron that represented the target node t in the graph. Finally, we implemented an interaction algorithm which caused the bump of activation in the continuous attractor layer to move in the direction of the wave front as it reached the region of activation in the continuous attractor layer. We found that this model was successfully able to move the activation bump in the continuous attractor layer through the sheet of neurons to the location that mapped to the target node t in the graph, thus solving the graph traversal problem. We found further that the model was robust to large changes in the graph structure. Specifically we showed that if large portions of the graph are made inaccessible and the relevant neurons in the model were zeroed out that the model is still able to guide the activation bump from the start node s to the target node t successfully.

Despite its relatively small scale we believe that models such as ours may provide a starting point in understanding how brains are able to exhibit flexible behaviour with respect to different kinds of cognitive tasks. Apparently, the stereotypical theme of connectivity encountered across the neocortex allows the brain to create a model of its environment based on sensory perception. Once established, the neural structure can be used as a "planning board" to support different cognitive tasks.

Next to a deeper understanding of the brain, we believe that models like ours can be an inspiration for new algorithms of artificial intelligence (AI). Artificial neural networks used in technical applications today are typically input-driven, they rely on feed-forward processing of information through several layers of neurons, and they are trained via supervised learning. The brain, however, continuously integrates sensory input into its own dynamics, its connectivity structure is mostly recurrent, and learning happens to a large extent in an unsupervised way. In all three of these fundamental differences, our model is aligned more closely to the properties of the brain than those of most other AI algorithms. At the same time, it shows how relevant computational problems can be solved with a very generic approach that relies heavily on self-organization. Potential applications include motion control for robots, especially in scenarios which require a high degree of flexibility and continuous adaptation to changing circumstances. As an

additional interesting feature, since the model is based on artificial spiking neurons, it can potentially be implemented very efficiently in neuromorphic computer hardware.

Looking further afield, simple, interpretable systems will become increasingly important as human-inspired socially intelligent systems become more prevalent in our day-to-day lives, and become more important to the functioning of certain public infrastructures. As mobile robots begin to more readily pilot automobiles, deliver medication to patients in hospitals, and provide assistance to specialists in uncertain environments, it will be important that the ways in which these systems make their decisions are transparent to their users. As discussed in previous chapters, deep learning provides something of a black box with regards to a model's decision making. This may prove to be problematic in high impact areas such as the ones just mentioned, where a user's trust in an artificially intelligent system will be pivotal in the successful adoption of that technology. Intuitively speaking, a system will be much harder to trust when its decision making process is lost in a neural network black box. Our model, on the other hand, provides a straightforwardly interpretable decision making process. The next step in the model's activity is based on the activation of a neuron, or set of neurons, with which the currently activated 'as-is' location has the highest synaptic weight, and where the degree of synaptic weight is learned through statistical exposure to that problem.

Further, the benefit of this interpretability will, in principal, be compounded by the extent to which the algorithm is useable by such artificial agents. Because of the generality of the model, it can, in principal, be used to successfully solve any problem that requires planning. In the case of social robotics, this will be particularly useful, since planning problems become much more complex at the social scale. However, this does the raise the question as to what extent the model is able to scale to much larger networks. In our case, the dimensionality of the model is tied to the dimensionality of the state-space - significantly larger mazes will require a larger number of to-be representations, for example. Planning problems with very large state-spaces, such as those that are likely to occur with multi-agent social scenarios, will require significantly larger networks than those which we experimented on in this chapter.

All this points to the fact that further empirical work is required to truly understand how successful our model would be in a real-world, socially intelligent agent. Future studies should, first of all, experiment with significantly larger neural sheets to see if the behaviour scales reliably with much larger models. While we see no reason that it would not, the behaviour of neural networks such as this is often unpredictable, and stable behaviour at one level may not translate to stable behaviour at another level without significant tuning of network parameters. An

additional further study might look to implement our model in a simple virtual agent, in a maze setup that mirrors one from a well validated empirical rat study. It would be useful to see how our model went about solving such a maze, and how this was similar, or different, to how such studies observe rats solving the same maze. We have no specific hypotheses about how our model would perform in comparison to a rat in this scenario since, in a real animal there will be many more complex neural systems interacting with its planning system to produce its final trajectory and so the comparison is not comparing like for like. A comparison would none-the-less provide useful insights into ways in which our model could be adapted to match natural behaviour more accurately. Our aim in this experiment was simply to propose a flexible planning mechanism that could, in principle, be implemented straightforwardly using the kinds of neurons and connections that we know to be active during such tasks without a concern for exactly how this planning system would interact with other neural activity. As such, a comparative analysis of the sort just described was outside of the scope of this proof-of-concept study.

Chapter 6

General Discussion

6.1 Overview and Contribution

In this thesis, I have sought to contribute to an understanding of how neural network methods can be used to model human cognitive abilities. In sight of this goal, I have investigated a number of these methods that look to imbue socially-oriented robots with human-like cognitive capabilities, such that those robots might operate in complex and highly dynamic real world environments in an effective way. My motivation for situating my investigation of neural network approaches in social robotics was for the following reasons. First, social robots are increasingly becoming a ubiquitous parts of our everyday lives: from voice agents in our phones and televisions to embodied robots that are deployed across a number of complex social settings ranging from hospitals and schools to airports and hotels [Aymerich-Franch and Ferrer, 2020][Broekens et al., 2009][Lenz et al., 2008][Logan et al., 2019][Triebel et al., 2016]. As the scope of what these robots are expected to do increases, so to will the demands on their cognitive abilities [Cross and Ramsey, 2021]. Thus, social robotics is a field in which effective computational models of human cognition are highly desirable. Second, research which looks at modelling human-inspired cognitive abilities for application in embodied social robots using neural network methods is still in its early development. As such, significant opportunities exist to meaningfully contribute to this field in a way that will have a large impact beyond purely academic endeavour. Next, we are currently at a time when many of the limitations that may well have stood in the way of progress in this field are being alleviated. Difficulties with respect to collecting large datasets are being overcome by the availability of cost-effective and widely accessible data capturing devices from lightweight motion capture systems, portable scanning devices such a fNIRS [Holtzer et al., 2011][Henschel et al., 2020], and teleconferencing software (such as Zoom or Skype). All of these methods mean it is now possible to

capture very large amounts of data without tying participants and experimenters to traditional lab-bound protocols. Further, the development of relatively inexpensive and research-appropriate social robots has meant that datasets are more reflective of the kinds of every day situations in which the resulting trained neural network models will be used, thus drastically increasing the ecological validity of those datasets and increasing the likelihood that those models will generalise well into real world settings. Lastly, neural networks methods, and especially deep learning, have shown themselves in recent years to be ideally suited to modelling such cognitive abilities [Krizhevsky et al., 2012][Iglovikov and Shvets, 2018][Nagarajan et al., 2020], largely due to the increasing availability of very large amounts of data. This is perhaps not surprising since, as we saw in the introduction, the structure of deep neural networks (and neural networks such as those discussed in Chapter 5), are biased towards representing cognitive tasks in a way that reflects how those tasks are likely dealt with by the brain [Bengio, 2009][Tolman, 1948][Whittington et al., 2020].

With these motivations in mind, I conducted three empirical experiments to address these aims. In Chapter 3, we identified two interesting human cognitive abilities that we believed would have a significant impact on social robotics if they were effectively modelled: subjective self-disclosure, and subjective psychological stress. These differed with respect to much of the work on modelling stress and self-disclosure from the literature as we were expressly interested in *subjective* instances of these factors, i.e. when participants believed themselves to be self-disclosing or feeling stressed rather than when an observer judged them as such [Soleymani et al., 2019][Bara et al., 2020]. Our aim in this study was to investigate, in the first instance, whether deep learning was an appropriate methodology for modelling these cognitive abilities. That is, we sought to answer the degree to which even basic "off-the-shelf" architectures were able to grade a particular audio snippet in accordance to how stressed the person performing the utterance felt themselves to be and how much self-disclosure they considered themselves to be sharing in that instance. We determined that, if deep learning was to be considered as an appropriate approach, then a majority of these models should score well above chance accuracy on this task. Our first goal was to attempt to deal with the constraints on data that neural network approaches impose. To do this we prioritised collecting as much data as we were able to given the financial and time constraints of the project. We then processed each data point by hand to ensure that all audio segments were maximally poised to assist with model learning. Finally, by asking participants to label their own interactions, we ensured that all data points were accurately labelled. We tested six basic architectures: a fully connected linear network, a convolutional network, a long short-term mem-

ory network, a convolutional long short-term memory network, a continuous state Hopfield associative memory network, and a novel Hopfield convolutional network on both the stress and self-disclosure tasks. We also tested two different representations of the audio input (log mel spectrograms, and eGeMAP features), and two experimental framings (regression and classification) as our data seemed to fall between categorical and scaled data. In all cases, we found that the trained models performed significantly above chance levels (48.34% vs. a chance of %16.67), thus motivating us to continue to develop these methods in future studies.

In Chapter 4, we looked to develop this work specifically in the case of modelling subjective self-disclosure. Our aim was to push the capabilities of a subjective self-disclosure model in a number of ways: first, by collecting a much larger dataset that included a visual modality (video recordings); second, by leveraging a number of techniques that have been shown to increase model performance (in particular, by using transfer learning via two large pretrained ResNets as model backbones [Bengio, 2012][Ng et al., 2015]); third, by constructing a novel architecture that used domain knowledge of the problem (i.e. using attention based modelling based on the idea that only particular parts of the video and audio inputs would contain features that were particularly informative with respect to self-disclosure); fourth, by rigorously testing the model using a large number of different parameters, including different input data representations, different loss functions, and different experimental framings; and fifth an finally, by designing a novel loss function, the scale preserving cross-entropy loss, that looked to strike a balance between the scaled and categorical nature of our data. In order that we could be assured of the efficacy of our models, we trained baseline Gaussian-Kernel support vector machines (SVM) on each of the different input representations (this also allowed us some insight into which input representations were most informative with respect to the task). We found that all model versions significantly outperformed the SVM baselines and that input features that were condensed using principal components analysis combined with our novel loss function performed best on the task with respect to F1 scores.

Taken together, the first two empirical chapters of this thesis make a significant contribution to the field. Our findings suggest that neural networks, specifically deep learning methods, show great promise with respect to being able to model complex human cognitive abilities geared towards high impact applications in real-world social robotics. We show that paying careful attention to high quality, well labelled data, can effectively deal with the problem of dataset size. We also make significant headway into the field of self-disclosure modelling, specifically by providing one of the first investigations into how deep learning can be used to model subjective self-disclosure and confirm our results with a rigorous testing

methodology. Further, we provide a number of novel techniques for approaching this problem: a multi model attention network and a novel loss function. We also aim to make versions of both of our datasets, and our models and trained parameters, publicly available such that the work that we have conducted can be improved upon (this point is covered in more detail in the next section).

Finally, in Chapter 5 we looked at the problem of cognitive planning; an ability that is at the core of the successful operation of robots that will be used in social situations. In this study we first argue, in line with a large portion of the neuroscientific literature, that many cognitive planning problems can be couched as graph traversal problems where the graphs in question represent cognitive maps [Tolman, 1948]. We then state that successful planning on these maps/ graphs would amount to solving traversal problems, where the goal is to move the activation of a neural representation of an "as-is" state to the representation of a desired "to-be" state. We propose a novel computational model that utilises two different neuronal networks based on models of spiking and continuous attractor neurons. We show that our model is able to perform graph traversal through a cognitive map that is represented by the continuous attractor layer and how our algorithm is able to provide short path solutions to these problems in a wide number of complex cognitive map formations.

In this study we show that neural network methods are capable of solving a number of significant cognitive tasks even when there is ostensibly no learning involved (of course, the maps themselves are likely to have been established via Hebbian learning but the traversal of the graph occurs without any such learning). Here, we contribute to results that show that the structures of neural networks are able to naturally reflect structures of cognitive problems, and that this structural symmetry is able to be leveraged to solve these problems in efficient ways. In this way, we contribute to a growing field of research that looks to establish how more complex biologically inspired neural network models are able to imbue artificial agents with human-like intelligence. We also provide an insight into one way in which the very large number of recurrent connections between neurons in the neocortex might function i.e. to provide a computational substrate upon which cognitive planning tasks might be solved.

6.2 Limitations, and Future Work

While the approaches taken in the three empirical chapters of this thesis show that neural network methods are well positioned to model human cognitive capabilities, a number of drawbacks to these methods remain, particularly in relation to deep learning. First, as discussed in this section and in the introduction, deep

learning models are extremely data hungry. State of the art language models such as BERT [Devlin et al., 2019] and GPT3 [Brown et al., 2020] are trained on billions of input examples, and even much smaller models that deal with problems like image recognition and action prediction are trained on datasets that contain tens of thousands or millions of data points [Krizhevsky et al., 2012][Shahroudy et al., 2016]. While, as I have shown in Chapter 3 and Chapter 4, these demands can be mitigated and decent results are possible even with much smaller datasets, truly generalizeable neural network models that safely function in real world environments will require much more training data than we were able to collect. This kind of very large dataset and the huge models that they are used to train (recall that GPT3 contained roughly 75 billion parameters) thus require an enormous amount of computing power. The direct result of this is the use of massive amounts of energy which will have a significant impact on the environment via the carbon footprint required to train and maintain these models [Strubell et al., 2019][Bender et al., 2021].

Secondly, deep learning models are often described as forms of black box learning. While the design of deep networks lends them naturally to the problem of approximating cognitive abilities framed as mathematical functions, it also prevents those models from being easy to interpret. Beyond the algorithms that visualise the outputs of particular layers of the network [Yu et al., 2016] it is very hard to decipher how deep networks come to the decisions that they do. This is particularly problematic with respect to the use of such networks in social robots. One of the main hurdles in integrating these robots in society at large is user-based trust. That is, users must trust that the agent works in the appropriate way so that they feel comfortable using it for extended periods of time [Salem et al., 2015][Kellmeyer et al., 2018]. One way that trust in a decision is established in human-human social contexts is for that decision to be explained. If, when an individual asks a colleague where they should take a vacation, the colleague answers "San Diego", it would be natural to ask why that person thought San Diego was a good place to visit. If the person planning the trip is to spend a lot of money on flights and hotels, it seems sensible to ask why a particular recommendation is provided, given that a large amount of resources could be wasted if the recommendation is bad. This same logic will naturally apply to social robots, particularly in situations where those robots have to make high impact decisions or are advising on high-stakes matters. Though explainable AI is an increasingly expanding field of research [Samek et al., 2017] (for review see [Xu et al., 2019]), it is reasonable to ask the question of whether more work should be done to look for alternative methodologies that lend themselves more naturally to explainability.

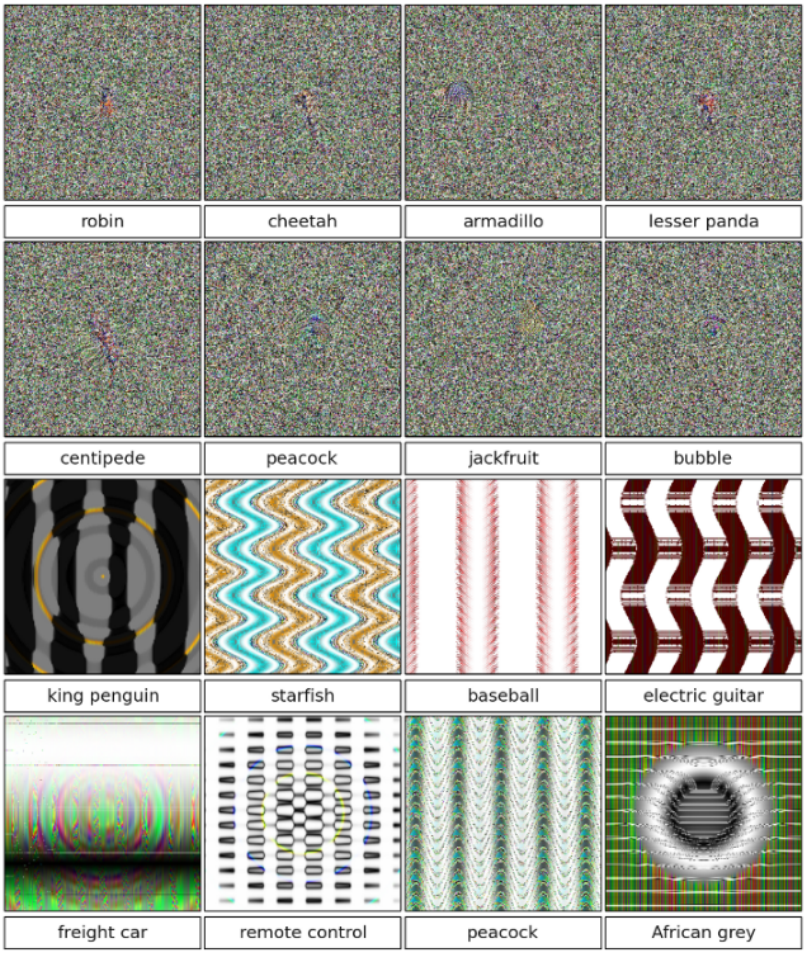


Figure 6.1: Input images to a neural network trained on ImageNet and that network's associated output labels. Each label was assigned with $\geq 99.6\%$ confidence. Adapted from [Nguyen et al., 2015].

The issue of explainability leads naturally to the problem of adversarial learning. Briefly, adversarial learning is the process by which trained neural networks are attempted to be fooled by deceptive input [Szegedy et al., 2013]. The task of learning in this context involves both the problem of formulating such inputs, detecting such inputs, and designing deep networks that are resistant to them [Biggio et al., 2013]. What adversarial attacks show is that even extensively trained neural networks are able to make incorrect decisions that they are extremely confident about. For instance, [Nguyen et al., 2015] show that noisy or nonsensical visual inputs into a network trained on ImageNet can lead that network to output high-confidence guesses as to the identity of the input. This can be seen in greater detail in figure Section 6.2 where inputs that make no sense to a human observer are none-the-less categorised as objects which we would instantly recognise.

Adversarial attacks also involve the process of applying small perturbations to model inputs that cause the model to make very different guesses about the identity

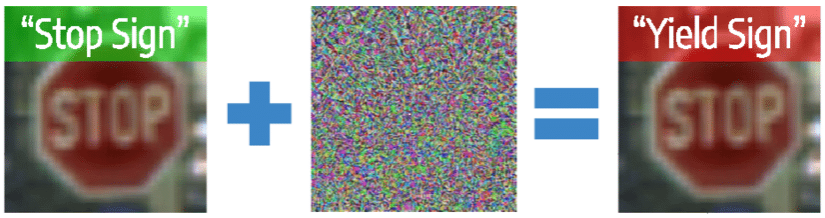


Figure 6.2: An image of a "stop" sign altered to produce a model guess of a "yield" sign. Adapted from [Carrara et al., 2018].

of those inputs. [Carrara et al., 2018] for instance show how a image of a stop sign can be imperceptibly altered to cause the recognition model to recognise the sign as a yield sign instead. This is clearly a huge issue for visual systems that are designed to function in high-risk, high-impact areas such as autonomous vehicles, where mistaking the meaning of a road sign could come at the expense of human life. The risk of such attacks in the field of social robots cannot likewise be overstated. If robots that function using neural networks are to operate in high-risk areas such as construction and mental health services (as we posit self-disclosure models to be in Chapter 3 and Chapter 4), then it is essential that these models be robust to adversarial attacks. This issue is compounded by the fact that neural networks are notoriously difficult to interpret (as discussed above). Clearly, understanding how a deep learning model is solving a problem will help significantly in understanding why it is making certain mistakes. In light of this, a clear need exists for research in this area to pay due diligence to these considerations. One way for the field to address this problem might be to normalize the requirement for papers that are submitted to journals or conferences to contain a section that explicitly deals with adversarial tests on those models in the same way in which it is standardised to perform ablation experiments on models that deal with a number of different hyperparameters.

Both studies in Chapter 3 and Chapter 4 are limited with respect to this issue. Neither study considers how adversarial attacks may effect the networks' performances and indeed, no attention has been paid to what an adversarial attack within the context of subjective self-disclosure classification may look like. Unfortunately, such an investigation was beyond the scope of both studies. It is clear, however, that this is an essential avenue for future research into this area if we are to take seriously the claim that these kinds of models might be implemented into a truly effective social robot.

Outside of the methods applied to the data, there are also obvious ways in which the data collection could be improved. First and foremost, more data could be collected. While we prioritised, in the first two empirical chapters, collecting as much data as possible, the data-hungry nature of deep networks means that the

most straightforward and effective way to improve model performance would be to collect more data. Not only would this straightforwardly give the models more examples to learn from, but including data points from social groups beyond of our subject pool would make the models vastly more effective in the real world. Conservatively speaking, what our models learned was not necessarily features relating to self-disclosure or stress from a general population, but those of a specifically British one, with specific demographic features, during a specific period of time. It may well be the case that self-disclosure signals differ from culture to culture, across years or decades, and a truly usable model of this capability will not be possible until these kinds of variations are more effectively modelled.

Further, it is clear that the ecological validity of the experiments could be improved upon. In both Chapter 3 and Chapter 4, we used a Wizard of Oz paradigm. While we maintain that this was a necessary and effective choice with respect to our goals, it is clear that a fully autonomous social robot would more accurately reflect how the situations we were attempting to model would unfurl in the real world. However, until this kind of highly adaptable (and yet experimentally controllable) conversational behaviour is possible in more readily available commercial robots, a Wizard of Oz paradigm strikes an acceptable balance between ecological validity and experimental control. In Chapter 4, unlike in the first study, participants were interacting with the Pepper robot via a Zoom video call. While this experimental choice was necessitated by restrictions on data collection caused by the COVID-19 pandemic, it is clear that an in-person protocol would be more ecologically valid (c.f., [Henschel et al., 2020]). While it is still perfectly plausible that human–robot interactions may occasionally occur via teleconferencing software, it's uncontroversial to suggest that a majority of human–robot social interactions will take place in-person in the future. In light of this, future studies should aim to collect as much in-person data as possible. Indeed, such data, when combined with the data we collected here, would contribute to more well-rounded models that can learn characteristics of self-disclosure common to both in-person and video-based interactions.

Finally, Chapter 3 and Chapter 4 provided a step-by-step build up of complexity with respect to the kinds of inputs that the models are trained on. Chapter 3, for example, considers only the audio modality, whereas Chapter 4 investigates the effects on model performance of both audio and visual inputs. There remains one obvious modality which should be investigated in future studies namely, text-based input. In [Soleymani et al., 2019], for instance, the experimenters note that verbal behaviours including those associated with word choice, were more effective than physical behavioural queues with respect to classifying a person's (non subjective) self-disclosure. This may well indicate that a treatment of the lexical features of

participants' speech will have a beneficial effect on our models, and future studies should investigate this. As in that same study, such a treatment opens the models up to the inclusion of large language models such as BERT [Devlin et al., 2019] that can be leveraged for transfer learning, which would likely increase the performance of the model even further.

The work in Chapter 5 likewise poses a number of potentially fruitful future avenues for further research. First and foremost, future work might look to close the gap between the two layers of the network. As it stands, our model uses a somewhat artificial way to construct the direction vector that informs the bump of activation in which way it should travel. One way to address this issue may be to construct the network in such a way that both travelling wave and continuous attractor dynamics are possible in the same neuronal sheet [Amari, 1977]. Lastly, future work on these kinds of models should look to investigate how they might be implemented in robot systems. At this point, the work is constrained to a computational model and rather abstract use cases in the form of the cognitive map formations that we designed in the experiments section. If we are right in our assumption that such a method can indeed be used to solve such planning problems, then an obvious next step would be to prove this claim via implementation.

6.3 Conclusion

This thesis develops and presents a number of novel, robust, and reproducible neural network techniques and datasets that look to demonstrate the validity and efficacy of these approaches in the field of human–robot interaction and social robotics. In two deep learning studies, we presented an investigation into how this approach might be used to tackle the challenging problem of subjective self-perception classification. In both cases, we designed and collected novel datasets using embodied humanoid robots in ecologically valid scenarios. In both studies, we show that deep learning approaches are well-suited to this problem and that promising model performance can be achieved even from relatively modest datasets and simplistic models. To expand upon this, we also show that more complex modelling techniques, such as transfer learning and the leveraging of attention mechanisms, can significantly increase the performance of these models. Finally, via a purely computational study, we show that neural networks also provide promise in solving problems related to how artificial agents and social robots are able to complete cognitive planning tasks.

While this general discussion has identified a number of limitations to the empirical work contained in this thesis and a number of ways in which the studies can be extended and improved upon, the impact of the results of those studies

should not be understated. This thesis contributes to a growing body of work that demonstrates the potential of using neural network methods to try to solve complex problems in social robotics. It is clear that, if the prediction that social robots will become ubiquitous parts of our everyday lives turns out to be correct, having a range of methods available to us that allow these machines to be bestowed with human-like cognitive abilities will be hugely important. Likewise, it is essential that we have a good understanding of what kinds of approaches to data collection and processing, model architecture, and hyperparameter selection tend to work best on problems presented in that field. As part of this goal, it will also be essential to appeal to neural network research that deals specifically with data and questions that exist within the sphere of social robotics. The work in this thesis attempts to meet all these needs. Firstly, via the collection and use of data that specifically involves socially oriented robots and reflects how those robots will be used in the real world. Second, by rigorously testing a large number of data representations, model architectures, training techniques, experimental framings, and loss functions. This, as well as other, research that fits these criteria (for example [Rodríguez-Moreno et al., 2020], [Atzeni and Reforgiato Recupero, 2018], [Le et al., 2018]) will continue to play an important role in the development of artificially intelligent systems that have a wide reaching impact on society, and well conducted, thorough, and modestly stated research of this kind will be essential in ensuring that these systems are safe, effective, and provide unquestionable benefit to their users.

Appendix A

Rebuttal for *ACM International Conference on Multimodal Interaction 2022* for paper "Is Deep Learning a Valid Approach for Inferring Subjective Self-Perceptions in Human–Robot Interactions?"

We thank the reviewers for their extremely helpful comments and questions. Below, we address those we consider to be the most pressing, and which we are confident we can resolve with small amendments to the paper in a short amount of time.

Reviewers 1, 3, and 4 commented on the lack of details of training and model hyperparameters as well as some information regarding the dataset. This was an oversight on our part and we agree that these are important for differentiating the models and ensuring reproducibility. Tables detailing all of this information will be included in the final paper.

Reviewer 1 argued that our accuracy scores were not high enough for the models to be implemented in a real-world scenario. While we agree that real-world application would undeniably benefit from even higher accuracy scores, we do address issues related to this directly in lines 878-885. However, as we also mention in lines 41-43 and 829-832, the primary aim of our paper was to explore how effective deep learning architectures are at modeling the problems of subjective stress and

self-disclosure. That is, we never intended to present the models as implementable in the real world in their current iteration. Rather, this work represents an important first step to explore the degree to which deep learning might be suitable for tackling this problem in the future, given further development. Reviewer 1 also commented that self-disclosure scores are continuous and therefore we should have framed the problem as a regression problem. As we discuss in lines 308-309, the participants were asked to classify their degree of self-disclosure on a 1-7 discrete scale. This was the reason that we chose to frame the problem as a discrete classification problem. We understand that these issues may have been unclear and we can make minor wording adjustments to further clarify these points.

Reviewer 2 asked about the human-human interaction in the WoZ study. To ensure consistency across interaction types and between participants we had the human-agent follow a script - which was simply a series of predetermined questions. We have no reason for thinking that this caused the participants to act unnaturally and none reported as such. A more thorough description of the paradigm is described in our paper which will be referenced in the final version.

Reviewers 2 and 3 mentioned ways in which the dataset could have been explored differently, such as by training separate models on the human-human, human-robot, and human-voice agent data subsets individually. Reviewer 2 mentioned that participants tend to portray different signifiers of self-disclosure depending on the agent they are interacting with, thus motivating the need to train separate models on each subset of data. This is an important point pertaining to the study of self-disclosure more generally and worthy of further investigation. However, our intention was to investigate whether deep learning models could extract features related to self-disclosure that were common to all interaction types and we can certainly make efforts to express this more clearly in the paper. We do intend to look into these questions in future studies and as we collect more data. For the purposes of the current study however we did not believe that we had enough data for each of these individual cases to justify the use of a deep learning approach which was one of our principal aims.

Reviewer 4 recommended that more time be taken for "intonation of parameters" which we took to mean hyperparameter tuning. As we mention in lines 41-43 and 878-885, our main goal was to investigate the baseline efficacy of a set of deep learning architectures rather than trying to achieve the best score possible in each case. We agree that hyperparameter tuning would increase the performance of these models but we see that as a next step in a larger project rather than a necessary step for the aims of this paper. The same reviewer rightfully commented that another metric such as F1 score could have been used to differentiate the model performance. Typically, F1 scores, precision, and recall tend to be used

for problems where there is significant class imbalance. While we did have large class imbalance initially, we believe that the lengths we went to to balance the classes meant that the accuracy scores we provided gave a good reflection of the performance of the models.

Finally, we greatly appreciate Reviewer 3's comments on restructuring. We intend to implement these changes and use the extra space provided to ensure the contributions that the paper makes to the field are more explicit. We also will include a more thorough review of the literature on stress detection, based on Reviewer 3's helpful suggestions.

Appendix B

Rebuttal for *ACM Conference on Human–Robot Interaction 2022* for paper "Is Deep Learning a Valid Approach for Inferring Subjective Self-Perceptions in Human–Robot Interactions?"

We thank the reviewers for their extremely useful comments which we address below.

First, we should point out that a conceptual mistake was made on our part in terms of the terminology used for the training procedure. Throughout the paper, we refer to the testing set as the validation set, suggesting that we had no unseen set of examples to train the models on. We did in fact train the models using a train and test split rather than a train and validation split. We apologize for this confusion and we will change this terminology in the final version.

Reviewer 3 suggested that our use of eGeMAP features could have been better justified. We chose these features first because para-linguistic studies have shown that they play an important part in the non-lexical recognition of emotion and, relatedly, because there is empirical evidence to suggest that variations in the emotional aspects of one's voice play an important role in picking up on stress and self-disclosure. Secondly, meta-analytic studies have shown that these features perform better than much larger feature sets on the same emotion recognition tasks, thus reducing the complexity of the input space and the dimensionality of the model without sacrificing performance. Further, we agree that we could

have investigated a larger array of feature sets but we felt that, since we aimed to probe these questions at a more fundamental level to indicate fruitful avenues for development, we believed that choosing these well-established features as a contrast to mel-spectrograms was sufficient.

Reviewer 2 commented that the sample size was too small for a deep learning study. First, the way in which the size of the dataset was reported may have been slightly misleading. When accounting for data augmentation and windowing of the input data, each network was trained on 2415 input samples and tested on 446 (rather than just 625 as we reported). This is still a relatively small dataset for a deep learning approach, however, behavioural experiments are often limited in sample size due to their complexity. Hence, here we aimed at demonstrating how, despite the sample size, high-quality data and appropriate augmentation techniques can support us in building architectures with above-chance accuracy scores that could provide solid foundations for future work. We do highlight in the paper that our work is preliminary, as many technical developments in our field are. HRIs are still extremely novel and researchers are still exploring the field to provide solid scientific foundations for future development.

Reviewer 2 addressed that the state-of-the-art should be improved by drawing from the literature on non-lexical feature learning and that we should present these related studies. A large amount of initial research was done into this body of work and it is what allowed us to settle on the input data representations and the neural network architectures that we chose. Due to restrictions in space, however, we felt it better to reference a few key papers in these fields rather than dedicating a whole section to it. However, we can see how this would be useful to the reader and we can add a section by taking out a number of the figures which reviewers pointed out weren't informative. We also note that the architectures themselves could have been made more sophisticated by drawing on this research. Again, since we aimed to explore the efficacy of standard neural network architectures we did not see this as a necessary step at this stage. However, we do intend to explore these in later studies.

Reviewer 3 pointed out that the analysis of our results lacks sufficient detail. On the basis of their comments we took this to mean firstly, that a more appropriate way to test the models would have been with leave one out cross validation (LOOCV) rather than our use of a train/test split and secondly, that we should spend more time explaining how the study relates to the field of HRI. The reason we chose a train test/split over LOOCV was firstly, while perhaps being methodologically correct, LOOCV tends to overestimate the performance (it trains on almost all data at disposition) and, most importantly, it does not really show whether an approach generalizes. The use of a traditional train/ test

split provides a more realistic account of how an approach can perform on unseen data and, correspondingly, it provides more reliable information on how the approach generalizes. Secondly, LOOCV poses particular challenges with respect to computational resources. Using LOOCV would have meant training each of the 6 networks 114 times with each training iteration occurring over the assigned number of epochs thus massively increasing both the computational cost of the procedure and the amount of time taken to explore all of the models. To address the second point, while the paper does mention the implications of our study to the field of HRI in a few places throughout the paper, we recognize that a more focused treatment of these points would be beneficial for the aims of the conference. As such we can easily collate these and put them into a dedicated section.

Reviewer 2 pointed out that an MSE loss and a regression problem were not the best choices and that a classification problem might have been better. We trained models on both of these variations and found that a regression/ MSE framework produced the best results. We are happy to hear that our approach sparks a methodological discussion about approaching behavioural data in HRI with modern machine learning techniques, and we welcome a continuation of this discussion at the conference.

Finally, reviewers pointed out that there were a number of typos and inconsistencies with the tables. We will of course correct these in the final submission.

We would like to thank the reviewers and the chair again for their valuable feedback and their invested efforts.

Appendix C

Rebuttal to *Nature Scientific Reports* for paper "A Hybrid Biological Neural Network Model for Solving Problems in Cognitive Planning"

Reviewer Comments and Detailed Answers

We thank the editor and all reviewers for their insightful comments. In the below we have responded in detail to each of the points raised by the reviewers and pointed to places within the main text of the paper where changes have been made.
Specifically, we have spent a significant amount of time including additional up-to-date references and taking care to explain parts of the problem, and our approach to it, in more detail as we found that both reviewers agreed that this was what was required most in the manuscript. Inline with both reviewers' comments, we have taken care to establish the position of our ideas within the wider literature and made efforts to make it clear how our approach differs and coincides with those that have come before it.

Reviewer 1

R1: In recent years several papers on spiking and neuromorphic graph algorithms have been published. This is certainly an important topic. In this paper authors propose a model that is using recurrent neural connections as a computational substrate to solve graph traversal problems with the help of travelling waves of cortical excitation. Attractor networks create localized activity that can move in

a cognitive map. The idea seems to be attractive and simple models have their value.
I have some critical remarks.
First, model structures in principle may develop in a self-organized manner into a cognitive map via Hebbian learning. There are many models of this sort, various self-organized maps have been used to create models of orientation and ocular dominance columns in the visual cortex (see Erwin, E., Obermayer, K., & Schulten, K. Neural Computation 1995 paper for a critical comparison).

Answer: *We agree with the reviewer that more references should be made to neural models that bear similarities to ours with respect to how cognitive maps might be formed via Hebbian learning. We have included a sentence that makes this comparison clear on page 4 and included some references that the reviewer pointed us towards.*

R1: Second, bubbles of activity have been investigated by S-I. Amari (1977) in his continuous models of cortex, arising in local cortical regions due to the recurrence neural activity. John Taylor in his book on consciousness have written about this model in context of consciousness. These bubbles amplify weak inputs and move. Although Amari model has better mathematical grounding it has not been used to solve graph traversal problems. The appearance of illusory visual shapes in drug-induced states was explained by a global wave of neural activity on hyperexcitable visual cortex (Ermentrout and Cowan 1978).

Answer: *As above, we agree with the reviewer that comparisons should be made to previous work on attractor dynamics, specifically related to bumps of neural activity. We have made this comparison clear on page 5 as well as mentioned how our own model develops these ideas, as the reviewer suggested.*

R1: Third, the algorithm that is used to create pictures in Fig. 5 and others is not clear at all. Why should the activity bump travel through this specific path in the maze? Premotor cortex and basal ganglia seem to store sequences of complex movements, and transitions between well-trained linked attractor states are natural explanation of such movements. It would be helpful to describe step by step why at each time the images show bump and wave positions. Waves seem to come in Fig. 5 from different directions.

Answer: *The actual path used by the bump is in no way pre-defined or pre-trained. Instead it is determined by the direction of first and subsequent incoming waves using the (learned) connections that represent all possible transitions in the cognitive map.*

We require more detail on what we can do to help the reader in understanding our concepts better. From our point of view, the more informal description given in Section 2.2. should be sufficient for an intuitive understanding, while the formal discussion in The "Methods" appendix gives the formal background.

If considered useful, we can however provide some videos showing the full temporal evolution in addition to the static images shown in the paper.

In the caption of Figure 5, we added an explanation on the waves that travel into unexpected directions.

R1: Fourth, sec. 2.3 is quite vague; we know that reaching, grasping, and putting things into the mouth, is based on specific connections between patches of premotor and parietal (BA5/7) cortex ... see ex. Kolb et al. An Introduction to Brain and Behavior (2019).

Answer: *We adapted Section 2.3 such that it now connects our abstract model with a possible implementation in the brain's anatomy. While this connection remains speculative, we hope that it helps the reader to see the presented model more clearly in the wider context of experimentally observed high-level brain functions.*

R1: Fifth, several competing models are worth mentioning. Computing by Modulating Spontaneous Activity (CMSA) generates activity patterns modulated by external stimuli to give rise to neural responses (Chen, G., & Gong, P. Nature Communications, 2019), or Galinsky & Frank, on brain waves. (Journal of Cognitive Neuroscience, 2020).

Answer: *Thank you pointing out these additional models. We mentioned them together with another very recent result of Galinsky & Frank in the beginning of Section 4.*

R1: Remarks like "... our model makes the specific prediction that the latency between stimulation and the onset of muscle activation should increase with the distance between initial and target posture" should be substantiated.

Answer: *We added a brief explanation of this prediction.*

R1: Figures look like they were printed on a dot printer.

Answer: *On our side, figures have been created as vector graphics and thus should not suffer from any quality issues. That said, we improved clarity of the figures, by adding a dark background to the blocked (hatched) region in Figure 5. Also, we added a grid to the figures to clearer emphasize the neural network structure and explained it in the caption of Figure 4. This should additionally support the explanations we added to address your previous remark on the algorithm for creating simulations in Figure 5.*

Reviewer 2

R2: The paper addresses a very interesting research field, it is well written and presents the context and the designed model in a sufficiently clear way. Empirical evidence from literature, which motivated the model setting, are also clearly discussed. In the discussion section, authors clearly state the points of novelty with respect to neural models in literature. My main concern about the paper is that very recent related references seem to be missing. Authors should consider to add references to recent studies and, as done for the already cited papers, highlight the differences and relations to the present work.

Answer: *Thank you for pointing out these references. We added all of them and additional references in Section 4 and other parts of the paper and briefly highlighted the differences and relations to our approach.*

R2: As in related literature, the term recurrent connections is used to indicate a coupling attraction between cells in the same cortex, which is strengthened when two cells activate at the same time. The same term is used in RNN architectures (e.g. LSTM) to indicate a feedforward activation that is directed to the very same neuron from time Ti to Ti+1. I my opinion, readers with a deep learning background would be facilitated if understanding the model if such a distinction (and/or similarities) would be (shortly) discussed before describing the model.

Answer: *We added the suggested paragraph at the end of Section 2.1*

R2: In my opinion, authors should spend some additional text in providing a high level description/intuition of how Izhikevich neurons are modeled. This would help

in understanding the parameters sued in Table 1. Do such a parameters correspond to the initialisation of the neurons activation? How?

Answer: *We agree with the reviewer that some high-level explanation of how the Izhikevich neurons are modelled is required and helpful. For the sake of clarity we have included a general description of how the state of the neuron is computed, when it is considered to be spiking, and finally, how the after-spike reset values are assigned. These have been included on page 18.*

R2: The terms neural network and neuronal network seems to be used interchangeably. However there is a distinction. Could author clarify and check consistency over the paper?

Answer: *Since our model takes a system view, we opted for the term "neural". We have removed the remaining instances of the term "neuronal".*

R2: A detail: Was the value of R (12ms) chosen for some specific reason?

Answer: *The recovery period of length R has been introduced to prevent the waves from interacting with the bump multiple times (pulling it back and forth in the worst case). In order to clarify the choice of R, we have added a explanation to this effect on page 22:.*

R2: Section "Obstacles and Complex Setups" made me to wonder if there is some kind of relation that can be observed between time to the solution and configuration of the network, placements of obstacles, or other parameters. And indeed in the last section authors report that heterogeneous neurons setup lead to a faster solution. Is there some quantitative estimation of the relation between variance and speed? Or other considerations that can be added regarding this aspect?

Answer: *As we mentioned in the Section* Obstacles and Complex Setups*, the method is able to find the shortest path from start to target node in the underlying graph. The more convoluted the shortest path is, the longer the bump will need to reach the target node. This is approximately visible when comparing Figure 5(a) to Figure 4: The path in the former is roughly a factor* $\frac{3}{\sqrt{2}} = 2.12$ *longer than in the latter as the bump travels a distance of around 3 times the edge length of the square region instead of running straight across its diagonal. Also, the time-to-solution is longer by a factor of approximately* $2.22 = \frac{610\,\mathrm{ms}}{274\,\mathrm{ms}}$.

However,a proper quantification beyond this rough estimate is still open to investigation. In particular, the situation becomes much more difficult when waves are hitting the bump from different sides as in Figure 5(b), pulling it symmetrically into different directions. (Here, the advantage of the heterogeneous network comes into play which makes such symmetries much more unlikely and is more biologically plausible. It is even more difficult to study systematically, though.) Further, the recovery period R *will play a role as it influences synchronization between waves and bump movement and thus effectiveness of the overall process. For this reason we believe that such an investigation goes beyond the scope of this publication and will have to be done in subsequent work.*

Bibliography

[Abdallah and Plumbley, 2006a] Abdallah, S. and Plumbley, M. (2006a). Geometric dependency analysis. Technical report.

[Abdallah and Plumbley, 2006b] Abdallah, S. and Plumbley, M. (2006b). Unsupervised analysis of polyphonic music by sparse coding. *IEEE Transactions on Neural Networks*, 17(1):179–196.

[Abu-Mostafa and St. Jacques, 1985] Abu-Mostafa, Y. and St. Jacques, J. (1985). Information capacity of the hopfield model. *IEEE Transactions on Information Theory*, 31(4):461–464.

[Adolphs, 2002] Adolphs, R. (2002). Neural systems for recognizing emotion. *Current opinion in neurobiology*, 12(2):169–177.

[Ahrens and Dieter, 1985] Ahrens, J. H. and Dieter, U. (1985). Sequential random sampling. *ACM Trans. Math. Softw.*, 11(2):157–169.

[Aimone et al., 2021] Aimone, J. B., Ho, Y., Parekh, O., Phillips, C. A., Pinar, A., Severa, W., and Wang, Y. (2021). Provable Advantages for Graph Algorithms in Spiking Neural Networks. In *Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures*, pages 35–47. Acm.

[Aimone et al., 2019] Aimone, J. B., Parekh, O., Phillips, C. A., Pinar, A., Severa, W., and Xu, H. (2019). Dynamic Programming with Spiking Neural Computing. In *Proceedings of the International Conference on Neuromorphic Systems*, pages 1–9. Acm.

[Alvarez et al., 2002] Alvarez, P., Wendelken, L., and Eichenbaum, H. (2002). Hippocampal formation lesions impair performance in an odor-odor association task independently of spatial context. *Neurobiology of Learning and Memory*, 78:470–476.

[Amari, 1977] Amari, S.-i. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological cybernetics*, 27(2):77–87.

[Antaki et al., 2005] Antaki, C., Barnes, R., and Leudar, I. (2005). Diagnostic formulations in psychotherapy. *Discourse Studies*, 7(6):627–647.

[Aronov et al., 2017] Aronov, D., Nevers, R., and Tank, D. (2017). Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature*, 543:719–722.

[Aslan et al., 2020] Aslan, I., Ochnik, D., and Çınar, O. (2020). Exploring perceived stress among students in turkey during the covid-19 pandemic. *International Journal of Environmental Research and Public Health*, 17(23).

[Atzeni and Reforgiato Recupero, 2018] Atzeni, M. and Reforgiato Recupero, D. (2018). Deep learning and sentiment analysis for human-robot interaction. In *European Semantic Web Conference*, pages 14–18. Springer.

[Aymerich-Franch and Ferrer, 2020] Aymerich-Franch, L. and Ferrer, I. (2020). The implementation of social robots during the covid-19 pandemic. *arXiv preprint arXiv:2007.03941*.

[Baevski et al., 2020] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

[Baltrusaitis et al., 2018] Baltrusaitis, T., Zadeh, A., Lim, Y. C., and Morency, L.-P. (2018). Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 59–66. Ieee.

[Bara et al., 2020] Bara, C.-P., Papakostas, M., and Mihalcea, R. (2020). A deep learning approach towards multimodal stress detection. In *AffCon AAAI*, pages 67–81.

[Barbier et al., 2021] Barbier, T., Teuliere, C., and Triesch, J. (2021). Spike timing-based unsupervised learning of orientation, disparity, and motion representations in a spiking neural network. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1377–1386. Ieee.

[Baron-Cohen, 1991] Baron-Cohen, S. (1991). Precursors to a theory of mind: Understanding attention in others. In *Natural theories of mind: Evolution, development and simulation of everyday mindreading.*, pages 233–251. Basil Blackwell, Cambridge, MA, US.

[Barry et al., 2006] Barry, C., Lever, C., Hayman, R., Hartley, T., Burton, S., O'Keefe, J., Jeffery, K., and Burgess, N. (2006). The boundary vector cell model of place cell firing and spatial memory. *Reviews in the Neurosciences*, 17(1-2):71–98.

[Bell et al., 2016] Bell, S., Lawrence Zitnick, C., Bala, K., and Girshick, R. (2016). Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Bender et al., 2021] Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

[Bengio, 2009] Bengio, Y. (2009). *Learning deep architectures for AI*. Now Publishers Inc.

[Bengio, 2012] Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings.

[Bengio et al., 1994] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long term dependcies with gradient-decent is difficult. *IEEE transactions on neural networks*, 5:157–166.

[Bergstra and Bengio, 2012] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).

[Biggio et al., 2013] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer.

[Blum and Rivest, 1992] Blum, A. L. and Rivest, R. L. (1992). Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127.

[Bonnet et al., 2017] Bonnet, F., Quentin, B., Défago, X., and Nguyen, T. (2017). *Killing Nodes as a Countermeasure to Virus Expansion*, pages 227–243.

[Breazeal, 2003] Breazeal, C. (2003). Toward sociable robots. *Robotics and Autonomous Systems*, 42(3):167–175.

[Broekens et al., 2009] Broekens, J., Heerink, M., Rosendal, H., et al. (2009). Assistive social robots in elderly care: a review. *Gerontechnology*, 8(2):94–103.

[Broser et al., 2008] Broser, P., Grinevich, V., Osten, P., Sakmann, B., and Wallace, D. J. (2008). Critical period plasticity of axonal arbors of layer 2/3 pyramidal neurons in rat somatosensory cortex: Layer-specific reduction of projections into deprived cortical columns. *Cerebral Cortex*, 18(7):1588–1603.

[Brown and Teskey, 2014] Brown, A. R. and Teskey, G. C. (2014). Motor cortex is functionally organized as a set of spatially distinct representations for complex movements. *The Journal of Neuroscience*, 34(41):13574–13585.

[Brown et al., 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

[Brutti et al., 2010] Brutti, A., Cristoforetti, L., Kellermann, W., Marquardt, L., and Omologo, M. (2010). Woz acoustic data collection for interactive tv. *Language Resources and Evaluation*, 44(3):205–219.

[Budri et al., 2014] Budri, M., Lodi, E., and Franchi, G. (2014). Sensorimotor restriction affects complex movement topography and reachable space in the rat motor cortex. *Frontiers in Systems Neuroscience*, 8:231.

[Bush et al., 2015] Bush, D., Barry, C., Manson, D., and Burgess, N. (2015). Using grid cells for navigation. *Neuron*, 87(3):507–520.

[Byom and Mutlu, 2013] Byom, L. J. and Mutlu, B. (2013). Theory of mind: mechanisms, methods, and new directions. *Frontiers in human neuroscience*, 7:413.

[Byrd and Lipton, 2018] Byrd, J. and Lipton, Z. C. (2018). Weighted risk minimization & deep learning. *CoRR*, abs/1812.03372.

[Cameron et al., 2001] Cameron, K., Yashar, S., Wilson, C., and Fried, I. (2001). Human hippocampal neurons predict how well word pairs will be remembered. *Neuron*, 30:289–98.

[Cao et al., 2017] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., and Zisserman, A. (2017). Vggface2: A dataset for recognising faces across pose and age. *CoRR*, abs/1710.08092.

[Carrara et al., 2018] Carrara, F., Falchi, F., Amato, G., Becarelli, R., and Caldelli, R. (2018). Detecting adversarial inputs by looking in the black box. *arXiv preprint arXiv:1803.02111*.

[Catmur et al., 2016] Catmur, C., Cross, E. S., and Over, H. (2016). Understanding self and others: from origins to disorders. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371(1686):20150066.

[Caucheteux et al., 2021] Caucheteux, C., Gramfort, A., and King, J.-R. (2021). Long-range and hierarchical language predictions in brains and algorithms. *arXiv preprint arXiv:2111.14232*.

[Chen and Gong, 2019] Chen, G. and Gong, P. (2019). Computing by modulating spontaneous cortical activity patterns as a mechanism of active visual processing. *Nature Communications*, 10(1):4915.

[Cho et al., 2014] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

[Cohen et al., 1983] Cohen, S., Kamarck, T., and Mermelstein, R. (1983). A global measure of perceived stress. *Journal of health and social behavior*, 24(4):385–396.

[Colquhoun et al., 2017] Colquhoun, H. L., Squires, J. E., Kolehmainen, N., Fraser, C., and Grimshaw, J. M. (2017). Methods for designing interventions to change healthcare professionals' behaviour: a systematic review. *Implementation Science*, 12(1):30.

[Constantinescu et al., 2016] Constantinescu, A. O., O'Reilly, J. X., and Behrens, T. E. J. (2016). Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468.

[Cooper and Shepard, 1973] Cooper, L. A. and Shepard, R. N. (1973). Chronometric studies of the rotation of mental images. In *Visual information processing.*, pages xiv, 555–xiv, 555. Academic.

[Corballis et al., 1978] Corballis, M. C., Zbrodoff, N. J., Shetzer, L. I., and Butler, P. B. (1978). Decisions about identity and orientation of rotated letters and digits. *Memory & Cognition*, 6(2):98–107.

[Cozby, 1973] Cozby, P. C. (1973). Self-disclosure: a literature review. *Psychological bulletin*, 79(2):73.

[Croes and Antheunis, 2020] Croes, E. A. J. and Antheunis, M. L. (2020). Can we be friends with mitsuku? a longitudinal study on the process of relationship formation between humans and a social chatbot. *Journal of Social and Personal Relationships*, 38(1):279–300.

[Cross et al., 2019a] Cross, E. S., Hortensius, R., and Wykowska, A. (2019a). From social brains to social robots: applying neurocognitive insights to human-robot interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 374(1771):20180024.

[Cross and Ramsey, 2021] Cross, E. S. and Ramsey, R. (2021). Mind meets machine: Towards a cognitive science of human–machine interactions. *Trends in Cognitive Sciences*, 25(3):200–212.

[Cross et al., 2019b] Cross, E. S., Riddoch, K. A., Pratts, J., Titone, S., Chaudhury, B., and Hortensius, R. (2019b). A neurocognitive investigation of the impact of socializing with a robot on empathy for pain. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 374(1771):20180034.

[Curto and Itskov, 2008] Curto, C. and Itskov, V. (2008). Cell groups reveal structure of stimulus space. *PLoS Computational Biology*, 4(10):e1000205.

[Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

[Dahl et al., 2013] Dahl, G. E., Sainath, T. N., and Hinton, G. E. (2013). Improving deep neural networks for lvcsr using rectified linear units and dropout. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8609–8613.

[Dai et al., 2016] Dai, J., Liang, S., Xue, W., Ni, C., and Liu, W. (2016). Long short-term memory recurrent neural network based segment features for music genre classification. In *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5.

[Damen et al., 2020] Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al. (2020). The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141.

[Dautenhahn, 2007] Dautenhahn, K. (2007). Socially intelligent robots: dimensions of human–robot interaction. *Philosophical transactions of the royal society B: Biological sciences*, 362(1480):679–704.

[DeAngelis et al., 1995] DeAngelis, G. C., Ohzawa, I., and Freeman, R. D. (1995). Receptive-field dynamics in the central visual pathways. *Trends in neurosciences*, 18(10):451–458.

[Deisenroth et al., 2020] Deisenroth, M. P., Faisal, A. A., and Ong, C. S. (2020). *Mathematics for machine learning*. Cambridge University Press.

[Delhaye et al., 2018] Delhaye, B. P., Long, K. H., and Bensmaia, S. J. (2018). Neural basis of touch and proprioception in primate cortex. In *Comprehensive Physiology*, pages 1575–1602. American Cancer Society.

[Demircigil et al., 2017] Demircigil, M., Heusel, J., Löwe, M., Upgang, S., and Vermet, F. (2017). On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[Derlega et al., 1993] Derlega, V. J., Winstead, B. A., Lewis, R. J., and Maddux, J. (1993). Clients' responses to dissatisfaction in psychotherapy: A test of rusbult's exit-voice-loyalty-neglect model. *Journal of Social and Clinical Psychology*, 12(3):307–318.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[DiCarlo et al., 1998] DiCarlo, J. J., Johnson, K. O., and Hsiao, S. S. (1998). Structure of receptive fields in area 3b of primary somatosensory cortex in the alert monkey. *The Journal of Neuroscience*, 18(7):2626–2645.

[Ding et al., 2020] Ding, Q., Wu, S., Sun, H., Guo, J., and Guo, J. (2020). Hierarchical multi-scale gaussian transformer for stock movement prediction. In *Ijcai*, pages 4640–4646.

[Doeller et al., 2010] Doeller, C., Barry, C., and Burgess, N. (2010). Evidence for grid cells in a human memory network. *Nature*, 463:657–61.

[Douglas and Martin, 2007] Douglas, R. J. and Martin, K. A. (2007). Recurrent neuronal circuits in the neocortex. *Current Biology*, 17(13):R496–r500.

[Eichenbaum et al., 1987] Eichenbaum, H., Kuperstein, M., Fagan, A., and Nagode, J. (1987). Cue-sampling and goal-approach correlates of hippocampal unit activity in rats performing an odor-discrimination task. *Journal of Neuroscience*, 7(3):716–732.

[Eichenbaum and Lipton, 2008] Eichenbaum, H. and Lipton, P. A. (2008). Towards a functional organization of the medial temporal lobe memory system: role of the parahippocampal and medial entorhinal cortical areas. *Hippocampus*, 18(12):1314–1324.

[Elad, 2010] Elad, M. (2010). *Sparse and Redundant Representations*. Springer.

[Elman, 1990] Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

[Erwin et al., 1995] Erwin, E., Obermayer, K., and Schulten, K. (1995). Models of orientation and ocular dominance columns in the visual cortex: A critical comparison. *Neural computation*, 7(3):425–468.

[Etienne et al., 2018] Etienne, C., Fidanza, G., Petrovskii, A., Devillers, L., and Schmauch, B. (2018). Speech emotion recognition with data augmentation and layer-wise learning rate adjustment. *CoRR*, abs/1802.05630.

[Evans, 2011] Evans, J. S. B. (2011). Dual-process theories of reasoning: Contemporary issues and developmental applications. *Developmental Review*, 31(2-3):86–102.

[Evans and Stanovich, 2013] Evans, J. S. B. and Stanovich, K. E. (2013). Dual-process theories of higher cognition: Advancing the debate. *Perspectives on psychological science*, 8(3):223–241.

[Eyben et al., 2016] Eyben, F., Scherer, K. R., Schuller, B. W., Sundberg, J., André, E., Busso, C., Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S., and Truong, K. P. (2016). The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7(2):190–202.

[Eyben et al., 2010] Eyben, F., Wöllmer, M., and Schuller, B. (2010). Opensmile: The munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM International Conference on Multimedia*, Mm '10, page 1459–1462, New York, NY, USA. Association for Computing Machinery.

[Fan et al., 2021] Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., and Feichtenhofer, C. (2021). Multiscale vision transformers. *ArXiv*, abs/2104.11227.

[Feichtenhofer et al., 2018] Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2018). Slowfast networks for video recognition. *CoRR*, abs/1812.03982.

[Feizi et al., 2012] Feizi, A., Aliyari, R., and Roohafza, H. (2012). Association of perceived stress with stressful life events, lifestyle and sociodemographic factors: a large-scale community-based study using logistic quantile regression. *Computational and mathematical methods in medicine*, 2012:151865.

[Fernyhough, 2008] Fernyhough, C. (2008). Getting vygotskian about theory of mind: Mediation, dialogue, and the development of social understanding. *Developmental Review*, 28(2):225–262.

[Folli et al., 2017] Folli, V., Leonetti, M., and Ruocco, G. (2017). On the maximum storage capacity of the hopfield model. *Frontiers in Computational Neuroscience*, 10:144.

[Frattaroli, 2006] Frattaroli, J. (2006). Experimental disclosure and its moderators: A meta-analysis. *Psychological bulletin*, 132(6):823–865.

[Frick, 1985] Frick, R. W. (1985). Communicating emotion: The role of prosodic features. *Psychological bulletin*, 97(3):412–429.

[Fried et al., 1997] Fried, I., MacDonald, K. A., and Wilson, C. L. (1997). Single neuron activity in human hippocampus and amygdala during recognition of faces and objects. *Neuron*, 18(5):753–765.

[Frisina et al., 2004] Frisina, P. G., Borod, J. C., and Lepore, S. J. (2004). A meta-analysis of the effects of written emotional disclosure on the health outcomes of clinical populations. *The Journal of nervous and mental disease*, 192(9).

[Gable et al., 2004] Gable, S. L., Reis, H. T., Impett, E. A., and Asher, E. R. (2004). What do you do when things go right? the intrapersonal and interpersonal benefits of sharing positive events. *Journal of personality and social psychology*, 87(2):228–245.

[Galinsky and Frank, 2020a] Galinsky, V. L. and Frank, L. R. (2020a). Brain Waves: Emergence of Localized, Persistent, Weakly Evanescent Cortical Loops. *Journal of Cognitive Neuroscience*, 32(11):2178–2202.

[Galinsky and Frank, 2020b] Galinsky, V. L. and Frank, L. R. (2020b). Universal theory of brain waves: From linear loops to nonlinear synchronized spiking and collective brain rhythms. *Physical Review Research*, 2(2):023061.

[Gao et al., 2017] Gao, L., Guo, Z., Zhang, H., Xu, X., and Shen, H. T. (2017). Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055.

[Gauthier and Tank, 2018] Gauthier, J. L. and Tank, D. W. (2018). A dedicated population for reward coding in the hippocampus. *Neuron*, 99(1):179–193.e7.

[George et al., 2021] George, D., Rikhye, R. V., Gothoskar, N., Guntupalli, J. S., Dedieu, A., and Lázaro-Gredilla, M. (2021). Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nature communications*, 12(1):1–17.

[Giddens et al., 2013] Giddens, C. L., Barron, K. W., Byrd-Craven, J., Clark, K. F., and Winter, A. S. (2013). Vocal indices of stress: A review. *Journal of Voice*, 27(3):21–390.

[Graves et al., 2013] Graves, A., Jaitly, N., and Mohamed, A.-r. (2013). Hybrid speech recognition with Deep Bidirectional LSTM. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 273–278, Olomouc, Czech Republic. Ieee.

[Graves and Schmidhuber, 2005] Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

[Graziano, 2016] Graziano, M. S. (2016). Ethological action maps: A paradigm shift for the motor cortex. *Trends in Cognitive Sciences*, 20(2):121–132.

[Graziano et al., 2005] Graziano, M. S. A., Aflalo, T. N. S., and Cooke, D. F. (2005). Arm movements evoked by electrical stimulation in the motor cortex of monkeys. *Journal of Neurophysiology*, 94(6):4209–4223.

[Gregor et al., 2015] Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. (2015). DRAW: A Recurrent Neural Network For Image Generation. *arXiv:1502.04623 [cs]*. arXiv: 1502.04623.

[Gu et al., 2017] Gu, Y., Li, X., Chen, S., Zhang, J., and Marsic, I. (2017). Speech intention classification with multimodal deep learning. In *Canadian conference on artificial intelligence*, pages 260–271. Springer.

[Guanella et al., 2007] Guanella, A., Kiper, D., and Verschure, P. (2007). A model of grid cells based on a twisted torus topology. *International journal of neural systems*, 17:231–40.

[Gurney, 2018] Gurney, K. (2018). *An introduction to neural networks*. CRC press.

[Hafting et al., 2005] Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436:801–6.

[Haken et al., 1985] Haken, H., Kelso, J. S., and Bunz, H. (1985). A theoretical model of phase transitions in human hand movements. *Biological cybernetics*, 51(5):347–356.

[Hamid, 2000] Hamid, P. N. (2000). Self-disclosure and occupational stress in chinese professionals. *Psychological Reports*, 87(3_suppl):1075–1082.

[Hamilton et al., 2019] Hamilton, K. E., Mintz, T. M., and Schuman, C. D. (2019). Spike-based primitives for graph algorithms. *arXiv preprint arXiv:1903.10574*.

[Han and Yu, 2012] Han, S.-H. and Yu, H.-S. (2012). College women's self-leadership, stress of clinical practice and self disclosure in an area. *The Journal of Korean academic society of nursing education*, 18(1):131–140.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Hegel et al., 2009] Hegel, F., Muhl, C., Wrede, B., Hielscher-Fastabend, M., and Sagerer, G. (2009). Understanding social robots. In *2009 Second International Conferences on Advances in Computer-Human Interactions*, pages 169–174. Ieee.

[Henschel et al., 2020] Henschel, A., Hortensius, R., and Cross, E. S. (2020). Social cognition in the age of human–robot interaction. *Trends in Neurosciences*, 43(6):373–384.

[Henschel et al., 2021] Henschel, A., Laban, G., and Cross, E. S. (2021). What makes a robot social? a review of social robots from science fiction to a home or hospital near you. *Current Robotics Reports*, 2(1):9–19.

[Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

[Hochreiter and Schmidhuber, 1997a] Hochreiter, S. and Schmidhuber, J. (1997a). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[Hochreiter and Schmidhuber, 1997b] Hochreiter, S. and Schmidhuber, J. (1997b). Lstms can solve hard log time problems. *Neural computation*, page 8.

[Holtzer et al., 2011] Holtzer, R., Mahoney, J. R., Izzetoglu, M., Izzetoglu, K., Onaral, B., and Verghese, J. (2011). fnirs study of walking and walking while talking in young and old individuals. *Journals of Gerontology Series A: Biomedical Sciences and Medical Sciences*, 66(8):879–887.

[Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

[Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

[Hortensius and Cross, 2018] Hortensius, R. and Cross, E. S. (2018). From automata to animate beings: the scope and limits of attributing socialness to artificial agents. *Annals of the New York Academy of Sciences*, 1426(1):93–110.

[Hortensius et al., 2018] Hortensius, R., Hekele, F., and Cross, E. S. (2018). The perception of emotion in artificial agents. *IEEE Transactions on Cognitive and Developmental Systems*, 10(4):852–864.

[Hossain and Muhammad, 2019] Hossain, M. S. and Muhammad, G. (2019). Emotion recognition using deep learning approach from audio–visual emotional big data. *Information Fusion*, 49:69–78.

[Huang et al., 2014] Huang, Z., Dong, M., Mao, Q., and Zhan, Y. (2014). Speech emotion recognition using cnn. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 801–804.

[Iacaruso et al., 2017] Iacaruso, M. F., Gasler, I. T., and Hofer, S. B. (2017). Synaptic organization of visual space in primary visual cortex. *Nature*, 547(7664):449–452.

[Ibe, 2013] Ibe, O. C. (2013). *Elements of Random Walk and Diffusion Processes*. John Wiley & Sons, Inc, Hoboken, NJ.

[Iglovikov and Shvets, 2018] Iglovikov, V. and Shvets, A. (2018). Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*.

[Izhikevich, 2003] Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572.

[Izhikevich, 2004] Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15(5):1063–1070.

[Izhikevich, 2006] Izhikevich, E. M. (2006). Polychronization: Computation with Spikes. *Neural Computation*, 18(2):245–282.

[Jaeger, 2001] Jaeger, H. (2001). The" echo state" approach to analysing and training recurrent neural networks-with an erratum note'. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148.

[Jaitly and Hinton, 2013] Jaitly, N. and Hinton, G. E. (2013). Vocal tract length perturbation (vtlp) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, volume 117, page 21.

[Jeffery, 2011] Jeffery, K. J. (2011). Place cells, grid cells, attractors, and remapping. *Neural Plasticity*, 2011:1–11.

[Jing et al., 2017] Jing, L., Gulcehre, C., Peurifoy, J., Shen, Y., Tegmark, M., and SoljaÄ, M. (2017). Gated Orthogonal Recurrent Units: On Learning to Forget. *Neural computation*, page 7.

[Jirsa et al., 1998] Jirsa, V. K., Fuchs, A., and Kelso, J. A. S. (1998). Connecting cortical and behavioral dynamics: bimanual coordination. *Neural Computation*, 10(8):2019–2045.

[Jordan, 1997] Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. In *Advances in psychology*, volume 121, pages 471–495. Elsevier.

[Jourard, 1971] Jourard, S. M. (1971). *Self-disclosure: An experimental analysis of the transparent self*. John Wiley, Oxford, England.

[Jourard and Lasakow, 1958] Jourard, S. M. and Lasakow, P. (1958). Some factors in self-disclosure. *The Journal of Abnormal and Social Psychology*, 56(1):91–98.

[Judd, 1987] Judd, J. S. (1987). Learning in networks is hard. *Proceedings of the first international conference on neural networks*, 323:685–692.

[Kadia et al., 1999] Kadia, S., Liang, L., Wang, X., Doucet, J., and Ryugo, D. (1999). Horizontal connections within the primary auditory cortex of cat. In *Assoc. Res. Otolaryngol. Abstr*, volume 22, page 34.

[Kahn et al., 2012] Kahn, J. H., Hucke, B. E., Bradley, A. M., Glinski, A. J., and Malak, B. L. (2012). The distress disclosure index: A research review and multitrait–multimethod examination. *Journal of Counseling Psychology*, 59(1):134–149.

[Kahneman, 2011] Kahneman, D. (2011). *Thinking, fast and slow*. Macmillan.

[Kahou et al., 2016] Kahou, S. E., Bouthillier, X., Lamblin, P., Gulcehre, C., Michalski, V., Konda, K., Jean, S., Froumenty, P., Dauphin, Y., Boulanger-Lewandowski, N., et al. (2016). Emonets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 10(2):99–111.

[Kalchbrenner et al., 2015] Kalchbrenner, N., Danihelka, I., and Graves, A. (2015). Grid Long Short-Term Memory. *arXiv:1507.01526 [cs]*. arXiv: 1507.01526.

[Kalchbrenner et al., 2014] Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *CoRR*, abs/1404.2188.

[Kappas et al., 2020] Kappas, A., Stower, R., and Vanman, E. J. (2020). *Communicating with Robots: What We Do Wrong and What We Do Right in Artificial Social Intelligence, and What We Need to Do Better*, pages 233–254. Springer International Publishing, Cham.

[Kaur et al., 2004] Kaur, S., Lazar, R., and Metherate, R. (2004). Intracortical pathways determine breadth of subthreshold frequency receptive fields in primary auditory cortex. *Journal of Neurophysiology*, 91(6):2551–2567.

[Kaur et al., 2005] Kaur, S., Rose, H., Lazar, R., Liang, K., and Metherate, R. (2005). Spectral integration in primary auditory cortex: Laminar processing of afferent input, in vivo and in vitro. *Neuroscience*, 134(3):1033–1045.

[Kay et al., 2020] Kay, B., Date, P., and Schuman, C. (2020). Neuromorphic Graph Algorithms: Extracting Longest Shortest Paths and Minimum Spanning Trees. In *Proceedings of the Neuro-Inspired Computational Elements Workshop*, pages 1–6. Acm.

[Kellmeyer et al., 2018] Kellmeyer, P., Mueller, O., Feingold-Polak, R., and Levy-Tzedek, S. (2018). Social robots in rehabilitation: A question of trust. *Science Robotics*, 3(21):eaat1587.

[Kennedy-Moore and Watson, 2001] Kennedy-Moore, E. and Watson, J. C. (2001). How and when does emotional expression help? *Review of General Psychology*, 5(3):187–212.

[Kephart and White, 1991] Kephart, J. O. and White, S. R. (1991). Directed-graph epidemiological models of computer viruses. In *Proceedings. 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 343–359.

[Keysar et al., 2003] Keysar, B., Lin, S., and Barr, D. J. (2003). Limits on theory of mind use in adults. *Cognition*, 89(1):25–41.

[Kim et al., 2019] Kim, C., Shin, M., Garg, A., and Gowda, D. (2019). Improved vocal tract length perturbation for a state-of-the-art end-to-end speech recognition system. In *Interspeech*, pages 739–743.

[Kim and Sejnowski, 2021] Kim, R. and Sejnowski, T. J. (2021). Strong inhibitory signaling underlies stable temporal dynamics and working memory in spiking neural networks. *Nature Neuroscience*, 24(1):129–139.

[Kimppa et al., 2015] Kimppa, L., Kujala, T., Leminen, A., Vainio, M., and Shtyrov, Y. (2015). Rapid and automatic speech-specific learning mechanism in human neocortex. *Neuroimage*, 118:282–291.

[Kingma and Ba, 2014] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

[Ko et al., 2013] Ko, H., Cossell, L., Baragli, C., Antolik, J., Clopath, C., Hofer, S. B., and Mrsic-Flogel, T. D. (2013). The emergence of functional microcircuits in visual cortex. *Nature*, 496(7443):96–100.

[Ko et al., 2011] Ko, H., Hofer, S. B., Pichler, B., Buchanan, K. A., Sjöström, P. J., and Mrsic-Flogel, T. D. (2011). Functional specificity of local synaptic connections in neocortical networks. *Nature*, 473(7345):87–91.

[Kohonen, 1982] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

[Kolb et al., 2019] Kolb, B., Whishaw, I., and Teskey, G. C. (2019). *An Introduction to Brain and Behavior*. Macmillan Learning, New York, UK, 6th edition.

[Korte and Vygen, 2018] Korte, B. and Vygen, J. (2018). *Combinatorial Optimization: Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer-Verlag Berlin Heidelberg, 6th edition.

[Kratz and Manis, 2015] Kratz, M. B. and Manis, P. B. (2015). Spatial organization of excitatory synaptic inputs to layer 4 neurons in mouse primary auditory cortex. *Frontiers in Neural Circuits*, 9.

[Kreiner and Levi-Belz, 2019] Kreiner, H. and Levi-Belz, Y. (2019). Self-disclosure here and now: Combining retrospective perceived assessment with dynamic behavioral measures. *Frontiers in Psychology*, 10:558.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

[Kumaran et al., 2021] Kumaran, U., Radha Rammohan, S., Nagarajan, S. M., and Prathik, A. (2021). Fusion of mel and gammatone frequency cepstral coefficients for speech emotion recognition using deep c-rnn. *International Journal of Speech Technology*, 24(2):303–314.

[Kussul et al., 2001] Kussul, E., Baidyk, T., Kasatkina, L., and Lukovich, V. (2001). Rosenblatt perceptrons for handwritten digit recognition. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, volume 2, pages 1516–1520. Ieee.

[Kveraga et al., 2007] Kveraga, K., Ghuman, A. S., and Bar, M. (2007). Top-down predictions in the cognitive brain. *Brain and Cognition*, 65(2):145–168.

[Laban et al., 2021a] Laban, G., Ben-Zion, Z., and Cross, E. S. (2021a). Social robots for supporting post-traumatic stress disorder diagnosis and treatment. *Frontiers in Psychiatry*, 12.

[Laban et al., 2021b] Laban, G., George, J.-N., Morrison, V., and Cross, E. S. (2021b). Tell me more! assessing interactions with social robots from speech. *Paladyn, Journal of Behavioral Robotics*, 12(1):136–159.

[Laban et al., 2021c] Laban, G., Kappas, A., Morrison, V., and Cross, E. S. (2021c). Protocol for a mediated long-term experiment with a social robot. *PsyArXiv*.

[Laban et al., 2020] Laban, G., Morrison, V., and Cross, E. S. (2020). Let's talk about it! subjective and objective disclosures to social robots. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 328–330, Cambridge, United Kingdom. Association for Computing Machinery.

[Lazarus, 1966] Lazarus, R. S. (1966). *Psychological stress and the coping process.* McGraw-Hill, New York, NY, US.

[Lazarus, 1974] Lazarus, R. S. (1974). Psychological stress and coping in adaptation and illness. *The International Journal of Psychiatry in Medicine*, 5(4):321–333. Pmid: 4618837.

[Lazarus and Folkman, 1984] Lazarus, R. S. and Folkman, S. (1984). *Stress, appraisal, and coping.* Springer publishing company.

[Le et al., 2018] Le, T. D., Huynh, D. T., and Pham, H. V. (2018). Efficient human-robot interaction using deep learning with mask r-cnn: detection, recognition, tracking and segmentation. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 162–167. Ieee.

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

[Lee et al., 2006] Lee, K. M., Jung, Y., Kim, J., and Kim, S. R. (2006). Are physically embodied social agents better than disembodied social agents?: The effects of physical embodiment, tactile interaction, and people's loneliness in human–robot interaction. *International journal of human-computer studies*, 64(10):962–973.

[Lee et al., 2020] Lee, Y. C., Yamashita, N., and Huang, Y. (2020). Designing a chatbot as a mediator for promoting deep self-disclosure to a real mental health professional. *Proceedings of the ACM on Human-Computer Interaction*, 4(Cscw1):1–27.

[Lenz et al., 2008] Lenz, C., Nair, S., Rickert, M., Knoll, A., Rosel, W., Gast, J., Bannat, A., and Wallhoff, F. (2008). Joint-action for humans and industrial robots for assembly tasks. In *RO-MAN 2008-The 17th IEEE International*

Symposium on Robot and Human Interactive Communication, pages 130–135. Ieee.

[Levi-Belz and Kreiner, 2016] Levi-Belz, Y. and Kreiner, H. (2016). What you say and how you say it: Analysis of speech content and speech fluency as predictors of judged self-disclosure. *Social Psychological and Personality Science*, 7(3):232–239.

[Li and Lyu, 2021] Li, X. and Lyu, H. (2021). Epidemic risk perception, perceived stress, and mental health during covid-19 pandemic: A moderated mediating model. *Frontiers in Psychology*, 11:4100.

[Lin et al., 2021] Lin, W., Orton, I., Li, Q., Pavarini, G., and Mahmoud, M. (2021). Looking at the body: Automatic analysis of body gestures and self-adaptors in psychological distress. *IEEE Transactions on Affective Computing*.

[Linz et al., 2018] Linz, R., Singer, T., and Engert, V. (2018). Interactions of momentary thought content and subjective stress predict cortisol fluctuations in a daily life experience sampling study. *Scientific Reports*, 8(1):1–11.

[Liu and Guo, 2019] Liu, G. and Guo, J. (2019). Bidirectional lstm with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337:325–338.

[Liu et al., 2017] Liu, J., Wang, G., Hu, P., Duan, L.-Y., and Kot, A. C. (2017). Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Logan et al., 2019] Logan, D. E., Breazeal, C., Goodwin, M. S., Jeong, S., O'Connell, B., Smith-Freedman, D., Heathers, J., and Weinstock, P. (2019). Social robots for hospitalized children. *Pediatrics*, 144(1).

[Lu et al., 2014] Lu, S., Chen, Z., and Xu, B. (2014). Learning new semi-supervised deep auto-encoder features for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 122–132.

[Lu et al., 2017] Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30.

[Lucas et al., 2014] Lucas, G. M., Gratch, J., King, A., and Morency, L.-P. (2014). It's only a computer: Virtual humans increase willingness to disclose. *Computers in Human Behavior*, 37:94–100.

[Lucas et al., 2017] Lucas, G. M., Rizzo, A., Gratch, J., Scherer, S., Stratou, G., Boberg, J., and Morency, L.-P. (2017). Reporting mental health symptoms: Breaking down barriers to care with virtual human interviewers.

[Martinetz and Schulten, 1994] Martinetz, T. and Schulten, K. (1994). Topology representing networks. *Neural Networks*, 7(3):507–522.

[Mattingly, 1991] Mattingly, C. (1991). The narrative nature of clinical reasoning. *American Journal of Occupational Therapy*, 45(11):998–1005.

[Mazzia et al., 2021] Mazzia, V., Angarano, S., Salvetti, F., Angelini, F., and Chiaberge, M. (2021). Action transformer: A self-attention model for short-time human action recognition. *arXiv preprint arXiv:2107.00606*.

[McEliece et al., 1987] McEliece, R. J., Posner, E. C., Rodemich, E. R., and Venkatesh, S. S. (1987). The capacity of the hopfield associative memory. *IEEE transactions on Information Theory*, 33(4):461–482.

[Mendels et al., 2017] Mendels, G., Levitan, S. I., Lee, K.-Z., and Hirschberg, J. (2017). Hybrid acoustic-lexical deep learning approach for deception detection. In *Interspeech*, pages 1472–1476.

[Meng et al., 2019] Meng, H., Yan, T., Yuan, F., and Wei, H. (2019). Speech emotion recognition from 3d log-mel spectrograms with deep learning network. *IEEE Access*, 7:125868–125881.

[Merel et al., 2019] Merel, J., Botvinick, M., and Wayne, G. (2019). Hierarchical motor control in mammals and machines. *Nature communications*, 10(1):1–12.

[Miao et al., 2019] Miao, X., McLoughlin, I., and Yan, Y. (2019). A new time-frequency attention mechanism for tdnn and cnn-lstm-tdnn, with application to language identification. In *Interspeech*, pages 4080–4084.

[Michael S.A. Graziano and Moore, 2002] Michael S.A. Graziano, C. S. T. and Moore, T. (2002). Complex movements evoked by microstimulation of precentral cortex. *Neuron*, 34:841–851.

[Mikolov, 2012] Mikolov, T. (2012). *Statistical Language Models Based on Neural Networks*. PhD thesis, Brno University.

[Miller and Buschman, 2013] Miller, E. K. and Buschman, T. J. (2013). Cortical circuits for the control of attention. *Current Opinion in Neurobiology*, 23(2):216–222.

[Miller, 1992] Miller, K. D. (1992). Development of orientation columns via competition between on and off center inputs. *NeuroReport*, 3:73–76.

[Movshon et al., 1978] Movshon, J. A., Thompson, I. D., and Tolhurst, D. J. (1978). Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *The Journal of physiology*, 283(1):53–77.

[Muller et al., 2018] Muller, L., Chavane, F., Reynolds, J., and Sejnowski, T. J. (2018). Cortical travelling waves: mechanisms and computational principles. *Nature Reviews Neuroscience*, 19(5):255–268.

[Muller et al., 2014] Muller, L., Reynaud, A., Chavane, F., and Destexhe, A. (2014). The stimulus-evoked population response in visual cortex of awake monkey is a propagating wave. *Nature Communications*, 5(1):3675.

[Muller et al., 1996] Muller, R. U., Stead, M., and Pach, J. (1996). The hippocampus as a cognitive graph. *Journal of General Physiology*, 107(6):663–694.

[Murphy et al., 2021] Murphy, K., Di Ruggiero, E., Upshur, R., Willison, D. J., Malhotra, N., Cai, J. C., Malhotra, N., Lui, V., and Gibson, J. (2021). Artificial intelligence for good health: a scoping review of the ethics literature. *BMC Medical Ethics*, 22(1):14.

[Nagarajan et al., 2020] Nagarajan, T., Li, Y., Feichtenhofer, C., and Grauman, K. (2020). Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 163–172.

[Naselaris et al., 2015] Naselaris, T., Olman, C. A., Stansbury, D. E., Ugurbil, K., and Gallant, J. L. (2015). A voxel-wise encoding model for early visual areas decodes mental images of remembered scenes. *NeuroImage*, 105:215–228.

[Ng et al., 2015] Ng, H.-W., Nguyen, V. D., Vonikakis, V., and Winkler, S. (2015). Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pages 443–449.

[Nguyen et al., 2015] Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable

images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436.

[Niebuhr and Michaud, 2015] Niebuhr, O. and Michaud, A. (2015). Speech data acquisition -: The underestimated challenge. *Kieler Arbeiten in Linguistik und Phonetik (KALIPHO)*, 3:1–42.

[O'Keefe, 1976] O'Keefe, J. (1976). Place units in the hippocampus of the freely moving rat. *Experimental Neurology*, 51(1):78–109.

[O'Keefe and Dostrovsky, 1971] O'Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175.

[O'Keefe and Nadel, 1978] O'Keefe, J. and Nadel, L. (1978). *The Hippocampus as a Cognitive Map*. Oxford: Clarendon Press.

[Omarzu, 2000] Omarzu, J. (2000). A disclosure decision model: Determining how and when individuals will self-disclose. *Pers Soc Psychol Rev*, 4(2):174–185.

[Palanisamy et al., 2020] Palanisamy, K., Singhania, D., and Yao, A. (2020). Rethinking cnn models for audio classification. *arXiv preprint arXiv:2007.11154*.

[Panayotov et al., 2015] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

[Parvizi-Fard et al., 2021] Parvizi-Fard, A., Amiri, M., Kumar, D., Iskarous, M. M., and Thakor, N. V. (2021). A functional spiking neuronal network for tactile sensing pathway to process edge orientation. *Scientific Reports*, 11(1):1320.

[Pascanu et al., 2012] Pascanu, R., Mikolov, T., and Bengio, Y. (2012). Understanding the exploding gradient problem. *CoRR*, abs/1211.5063. arXiv: 1211.5063.

[Pawar and Kokate, 2021] Pawar, M. D. and Kokate, R. D. (2021). Convolution neural network based automatic speech emotion recognition using mel-frequency cepstrum coefficients. *Multimedia Tools and Applications*, 80(10):15563–15587.

[Pearce and Sharp, 1973] Pearce, W. B. and Sharp, S. M. (1973). Self-disclosing communication. *Journal of Communication*, 23(4):409–425.

[Pearson, 2019] Pearson, J. (2019). The human imagination: the cognitive neuroscience of visual mental imagery. *Nature Reviews Neuroscience*, 20(10):624–634.

[Pellecchia et al., 2005] Pellecchia, G. L., Shockley, K., and Turvey, M. T. (2005). Concurrent cognitive task modulates coordination dynamics. *Cognitive science*, 29(4):531–557.

[Penhune and Steele, 2012] Penhune, V. B. and Steele, C. J. (2012). Parallel contributions of cerebellar, striatal and m1 mechanisms to motor sequence learning. *Behavioural brain research*, 226(2):579–591.

[Petrovic and Gaggioli, 2020] Petrovic, M. and Gaggioli, A. (2020). Digital mental health tools for caregivers of older adults–a scoping review. *Frontiers in Public Health*, 8:128.

[Phillips, 2013] Phillips, A. C. (2013). *Perceived Stress*, pages 1453–1454. Springer New York, New York, NY.

[Powell et al., 2022] Powell, H., Laban, G., George, J.-N., and Cross, E. S. (2022). Is deep learning a valid approach for inferring subjective self-disclosure in human-robot interactions? In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, Hri '22, page 991–996. IEEE Press.

[Premack and Woodruff, 1978] Premack, D. and Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526.

[Pulvermüller et al., 2021] Pulvermüller, F., Tomasello, R., Henningsen-Schomers, M. R., and Wennekers, T. (2021). Biological constraints on neural network models of cognitive function. *Nature Reviews Neuroscience*, 22(8):488–502.

[Quirk et al., 1990] Quirk, G., Muller, R., and Kubie, J. (1990). The firing of hippocampal place cells in the dark depends on the rat's recent experience. *The Journal of Neuroscience*, 10(6):2008–2017.

[R. et al., 2003] R., K. S., Johnstone, T., and Klasmeyer, G. (2003). *Vocal expression of emotion*. Series in affective science. Handbook of affective sciences. Oxford University Press, New York, NY, US.

[Ramanathan et al., 2006] Ramanathan, D., Conner, J. M., and H. Tuszynski, M. (2006). A form of motor cortical plasticity that correlates with recovery of function after brain injury. *Proceedings of the National Academy of Sciences*, 103(30):11370–11375.

[Ramsauer et al., 2021] Ramsauer, H., Schafl, B., Lehner, J., Seidl, P., Widrich, M., Gruber, L., Holzleitner, M., Pavlovi'c, M., Sandve, G. K., Greiff, V., Kreil, D. P., Kopp, M., Klambauer, G., Brandstetter, J., and Hochreiter, S. (2021). Hopfield networks is all you need. *ArXiv*, abs/2008.02217.

[Ramsey, 2021] Ramsey, R. (2021). A call for greater modesty in psychology and cognitive neuroscience. *Collabra: Psychology*, 7(1):24091.

[Rebai et al., 2017] Rebai, I., BenAyed, Y., Mahdi, W., and Lorré, J.-P. (2017). Improving speech recognition using data augmentation and acoustic model fusion. *Procedia Computer Science*, 112:316–322. Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France.

[Riddoch and Cross, 2021] Riddoch, K. A. and Cross, E. S. (2021). "hit the robot on the head with this mallet" – making a case for including more open questions in hri research. *Frontiers in Robotics and AI*, 8:2.

[Riva et al., 2012] Riva, G., Baños, R. M., Botella, C., Wiederhold, B. K., and Gaggioli, A. (2012). Positive technology: Using interactive technologies to promote positive functioning. *Cyberpsychology, Behavior, and Social Networking*, 15(2):69–77.

[Roach et al., 1998] Roach, P., Stibbard, R., Osborne, J., Arnfield, S., and Setter, J. (1998). Transcription of prosodic and paralinguistic features of emotional speech. *Journal of the International Phonetic Association*, 28(1-2):83–94.

[Roberti et al., 2006] Roberti, J. W., Harrington, L. N., and Storch, E. A. (2006). Further psychometric support for the 10-item version of the perceived stress scale. *Journal of College Counseling*, 9(2):135–147.

[Robinson et al., 2019] Robinson, N. L., Cottier, T. V., and Kavanagh, D. J. (2019). Psychosocial health interventions by social robots: Systematic review of randomized controlled trials. *J Med Internet Res*, 21(5):1–20.

[Rodríguez-Moreno et al., 2020] Rodríguez-Moreno, I., Martínez-Otzeta, J. M., Goienetxea, I., Rodriguez-Rodriguez, I., and Sierra, B. (2020). Shedding light on people action recognition in social robotics by means of common spatial patterns. *Sensors*, 20(8):2436.

[Rolls, 2010] Rolls, E. T. (2010). Attractor networks. *WIREs Cognitive Science*, 1(1):119–134.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[Rossi et al., 2021] Rossi, R., Jannini, T. B., Socci, V., Pacitti, F., and Lorenzo, G. D. (2021). Stressful life events and resilience during the covid-19 lockdown measures in italy: Association with mental health outcomes and age. *Frontiers in Psychiatry*, 12:236.

[Rubino et al., 2006] Rubino, D., Robbins, K. A., and Hatsopoulos, N. G. (2006). Propagating waves mediate information transfer in the motor cortex. *Nature Neuroscience*, 9(12):1549–1557.

[Rueggeberg et al., 2012] Rueggeberg, R., Wrosch, C., and Miller, G. E. (2012). The different roles of perceived stress in the association between older adults' physical activity and physical health. *Health Psychology*, 31(2):164–171.

[Ruiz et al., 1990] Ruiz, R., Legros, C., and Guell, A. (1990). Voice analysis to predict the psychological or physical state of a speaker. *Aviation, Space, and Environmental Medicine*, 61(3):266–271.

[Rumelhart et al., 1986a] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986a). Learning representations by back-propagating errors. *Nature*, 323:533.

[Rumelhart et al., 1986b] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986b). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. *Science*, pages 318–362.

[Rumelhart and Zipser, 1985] Rumelhart, D. E. and Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science*, 9(1):75–112.

[Sainath et al., 2015] Sainath, T. N., Vinyals, O., Senior, A., and Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584.

[Sakurai, 2002] Sakurai, Y. (2002). Coding of auditory temporal and pitch information by hippocampal individual cells and cell assemblies in the rat. *Neuroscience*, 115(4):1153–1163.

[Salem et al., 2015] Salem, M., Lakatos, G., Amirabdollahian, F., and Dautenhahn, K. (2015). Towards safe and trustworthy social robots: ethical challenges and practical issues. In *International conference on social robotics*, pages 584–593. Springer.

[Samek et al., 2017] Samek, W., Wiegand, T., and Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*.

[Sato et al., 2012] Sato, T. K., Nauhaus, I., and Carandini, M. (2012). Traveling waves in visual cortex. *Neuron*, 75(2):218–229.

[Scherer et al., 2016] Scherer, S., Lucas, G. M., Gratch, J., Skip Rizzo, A., and Morency, L. (2016). Self-reported symptoms of depression and ptsd are associated with reduced vowel space in screening interviews. *IEEE Transactions on Affective Computing*, 7(1):59–73.

[Schlosser, 2020] Schlosser, A. E. (2020). Self-disclosure versus self-presentation on social media. *Current Opinion in Psychology*, 31:1–6. Privacy and Disclosure, Online and in Social Interactions.

[Schmitt and Schuller, 2018] Schmitt, M. and Schuller, B. (2018). Deep recurrent neural networks for emotion recognition in speech. In Seeber, B., editor, *Fortschritte der Akustik - DAGA 2018: Proceedings der 44. Jahrestagung für Akustik, München, Deutschland, 19-22 März 2018*.

[Schrijver, 2003] Schrijver, A. (2003). *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag Berlin Heidelberg.

[Schupp et al., 2006] Schupp, H. T., Flaisch, T., Stockburger, J., and Junghöfer, M. (2006). Emotion and attention: event-related brain potential studies. *Progress in brain research*, 156:31–51.

[Scoglio et al., 2019] Scoglio, A. A. J., Reilly, E. D., Gorman, J. A., and Drebing, C. E. (2019). Use of social robots in mental health and well-being research: Systematic review. *J Med Internet Res*, 21(7):e13322.

[Selvaraju et al., 2017] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

[Shahroudy et al., 2016] Shahroudy, A., Liu, J., Ng, T., and Wang, G. (2016). NTU RGB+D: A large scale dataset for 3d human activity analysis. *CoRR*, abs/1604.02808.

[Shani et al., 2021] Shani, C., Libov, A., Tolmach, S., Lewin-Eytan, L., Maarek, Y., and Shahaf, D. (2021). " alexa, what do you do for fun?" characterizing playful requests with virtual assistants. *arXiv preprint arXiv:2105.05571*.

[Shepard and Metzler, 1971] Shepard, R. N. and Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science (New York, N.Y.)*, 171(3972):701–703.

[Sheridan, 2016] Sheridan, T. B. (2016). Human–robot interaction: status and challenges. *Human factors*, 58(4):525–532.

[Singer and Lazar, 2016] Singer, W. and Lazar, A. (2016). Does the cerebral cortex exploit high-dimensional, non-linear dynamics for information processing? *Frontiers in Computational Neuroscience*, 10.

[Slavich et al., 2019] Slavich, G. M., Taylor, S., and Picard, R. W. (2019). Stress measurement using speech: Recent advancements, validation issues, and ethical and privacy considerations. *Stress*, 22(4):408–413.

[Sloan, 2010] Sloan, D. M. (2010). Self-disclosure and psychological well-being. Social psychological foundations of clinical psychology, pages 212–225. The Guilford Press, New York, NY, US.

[Snoek et al., 2012] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms.

[Soleymani et al., 2019] Soleymani, M., Stefanov, K., Kang, S.-H., Ondras, J., and Gratch, J. (2019). Multimodal analysis and estimation of intimate self-disclosure. In *2019 International Conference on Multimodal Interaction*, pages 59–68.

[Stächele et al., 2020] Stächele, T., Domes, G., Wekenborg, M., Penz, M., Kirschbaum, C., and Heinrichs, M. (2020). Effects of a 6-week internet-based stress management program on perceived stress, subjective coping skills, and sleep quality. *Frontiers in Psychiatry*, 11:463.

[Stillwell et al., 2017] Stillwell, S. B., Vermeesch, A. L., and Scott, J. G. (2017). Interventions to reduce perceived stress among graduate students: A systematic review with implications for evidence-based practice. *Worldviews on Evidence-Based Nursing*, 14(6):507–513.

[Strubell et al., 2019] Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Nips'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

[Svozil et al., 1997] Svozil, D., Kvasnicka, V., and Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, 39(1):43–62.

[Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[Szegedy et al., 2013] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

[Takahashi et al., 2015] Takahashi, K., Kim, S., Coleman, T. P., Brown, K. A., Suminski, A. J., Best, M. D., and Hatsopoulos, N. G. (2015). Large-scale spatiotemporal spike patterning consistent with wave propagation in motor cortex. *Nature Communications*, 6(1):7169.

[Taube et al., 1990] Taube, J., Muller, R., and Ranck, J. (1990). Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*, 10(2):420–435.

[Taylor, 1999] Taylor, J. G. (1999). *The race for consciousness*. MIT Press.

[Tolman, 1948] Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological review*, 55(4):189.

[Tolman and Honzik, 1930] Tolman, E. C. and Honzik, C. H. (1930). *Introduction and removal of reward and maze performance in rats*. University of California publications in psychology v. 4, no. 17. University of California Press.

[Triebel et al., 2016] Triebel, R., Arras, K., Alami, R., Beyer, L., Breuers, S., Chatila, R., Chetouani, M., Cremers, D., Evers, V., Fiore, M., et al. (2016). Spencer: A socially aware service robot for passenger guidance and help in busy airports. In *Field and service robotics*, pages 607–622. Springer.

[Tudor Car et al., 2020] Tudor Car, L., Dhinagaran, D. A., Kyaw, B. M., Kowatsch, T., Joty, S., Theng, Y.-L., and Atun, R. (2020). Conversational

agents in health care: Scoping review and conceptual analysis. *Journal of medical Internet research*, 22(8):e17158–e17158.

[Van Mieghem et al., 2009] Van Mieghem, P., Omic, J., and Kooij, R. (2009). Virus spread in networks. *IEEE/ACM Transactions on Networking*, 17(1):1–14.

[Van Puyvelde et al., 2018] Van Puyvelde, M., Neyt, X., McGlone, F., and Pattyn, N. (2018). Voice stress analysis: a new framework for voice and effort in human performance. *Frontiers in psychology*, 9:1994.

[van Ulzen et al., 2008] van Ulzen, N. R., Lamoth, C. J., Daffertshofer, A., Semin, G. R., and Beek, P. J. (2008). Characteristics of instructed and uninstructed interpersonal coordination while walking side-by-side. *Neuroscience letters*, 432(2):88–93.

[Vancampfort et al., 2017] Vancampfort, D., Koyanagi, A., Ward, P. B., Veronese, N., Carvalho, A. F., Solmi, M., Mugisha, J., Rosenbaum, S., De Hert, M., and Stubbs, B. (2017). Perceived stress and its relationship with chronic medical conditions and multimorbidity among 229,293 community-dwelling adults in 44 low- and middle-income countries. *American Journal of Epidemiology*, 186(8):979–989.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Nips'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

[Venugopalan et al., 2017] Venugopalan, S., Anne Hendricks, L., Rohrbach, M., Mooney, R., Darrell, T., and Saenko, K. (2017). Captioning images with diverse objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5753–5761.

[Vidyasagar et al., 2014] Vidyasagar, R., Folger, S. E., and Parkes, L. M. (2014). Re-wiring the brain: Increased functional connectivity within primary somatosensory cortex following synchronous co-activation. *NeuroImage*, 92:19–26.

[Wallace and Sakmann, 2008] Wallace, D. J. and Sakmann, B. (2008). Plasticity of representational maps in somatosensory cortex observed by in vivo voltage-sensitive dye imaging. *Cerebral Cortex*, 18(6):1361–1373.

[Wallace et al., 1991] Wallace, M., Kitzes, L., and Jones, E. (1991). Intrinsic inter- and intralaminar connections and their relationship to the tonotopic map in cat primary auditory cortex. *Experimental Brain Research*, 86(3).

[Walvekar et al., 2015] Walvekar, S. S., Ambekar, J. G., and Devaranavadagi, B. B. (2015). Study on serum cortisol and perceived stress scale in the police constables. *Journal of clinical and diagnostic research : JCDR*, 9(2):Bc10–bc14.

[Wang et al., 2020] Wang, J., Xue, M., Culhane, R., Diao, E., Ding, J., and Tarokh, V. (2020). Speech emotion recognition with dual-sequence lstm architecture. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6474–6478.

[Wang, 2013] Wang, X. (2013). The harmonic organization of auditory cortex. *Frontiers in Systems Neuroscience*, 7.

[Wang et al., 2019] Wang, X., Hu, J.-F., Lai, J.-H., Zhang, J., and Zheng, W.-S. (2019). Progressive teacher-student learning for early action prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3556–3565.

[Wang et al., 2016] Wang, Y., Huang, M., Zhu, X., and Zhao, L. (2016). Attention-based lstm for aspect-level sentiment classification. In Schmahl, C., editor, *Proceedings of the 2016 conference on empirical methods in natural language processing*, volume 11, pages 606–615. Oxford University Press.

[Wang et al., 2022] Wang, Z., Zhang, Y., Shi, H., Cao, L., Yan, C., and Xu, G. (2022). Recurrent spiking neural network with dynamic presynaptic currents based on backpropagation. *International Journal of Intelligent Systems*, 37(3):2242–2265.

[Werbos, 1990] Werbos, P. J. (1990). Back propogation through time: what it does and how to do it. *Proceedings of the IEEE*, 78:1550–1560.

[Whittington et al., 2020] Whittington, J. C., Muller, T. H., Mark, S., Chen, G., Barry, C., Burgess, N., and Behrens, T. E. (2020). The tolman-eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263.

[Widrich et al., 2020] Widrich, M., Schäfl, B., Ramsauer, H., Pavlovic, M., Gruber, L., Holzleitner, M., Brandstetter, J., Sandve, G. K., Greiff, V., Hochreiter, S., and Klambauer, G. (2020). Modern hopfield networks and attention for immune repertoire classification. *CoRR*, abs/2007.13505.

[Wiegner et al., 2015] Wiegner, L., Hange, D., Björkelund, C., and Ahlborg, G. (2015). Prevalence of perceived stress and associations to symptoms of exhaustion, depression and anxiety in a working age population seeking primary care - an observational study. *BMC Family Practice*, 16(1):38.

[Wight et al., 2016] Wight, D., Wimbush, E., Jepson, R., and Doi, L. (2016). Six steps in quality intervention development (6squid). *J Epidemiol Community Health*, 70(5):520.

[Williams and Zipser, 1989] Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280.

[Williams and Zipser, 1995] Williams, R. J. and Zipser, D. (1995). Gradient-based learning algorithms for recurrent networks and their computational complexity. In Chauvin, Y. and Rumelhart, D. E., editors, *Backpropagation: Theory, architectures, and applications*, pages 433–486. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

[Willmore et al., 2010] Willmore, B. D., Prenger, R. J., and Gallant, J. L. (2010). Neural representation of natural images in visual area v2. *Journal of Neuroscience*, 30(6):2102–2114.

[Wiltgen et al., 2004] Wiltgen, B. J., Brown, R. A., Talton, L. E., and Silva, A. J. (2004). New circuits for old memories: the role of the neocortex in consolidation. *Neuron*, 44(1):101–108.

[Wojtas and Chen, 2020] Wojtas, M. and Chen, K. (2020). Feature importance ranking for deep learning. *Advances in Neural Information Processing Systems*, 33:5105–5114.

[Wu et al., 2018] Wu, H., Zhou, K., Xu, P., Xue, J., Xu, X., and Liu, L. (2018). Associations of perceived stress with the present and subsequent cortisol levels in fingernails among medical students: a prospective pilot study. *Psychology research and behavior management*, 11:439–445.

[Xie et al., 2019] Xie, Y., Liang, R., Liang, Z., Huang, C., Zou, C., and Schuller, B. (2019). Speech emotion classification using attention-based lstm. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(11):1675–1685.

[Xie et al., 2022] Xie, Y., Liu, Y. H., Constantinidis, C., and Zhou, X. (2022). Neural Mechanisms of Working Memory Accuracy Revealed by Recurrent Neural Networks. *Frontiers in Systems Neuroscience*, 16:760864.

[Xu et al., 2019] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., and Zhu, J. (2019). Explainable ai: A brief survey on history, research areas, approaches and challenges. In *CCF international conference on natural language processing and Chinese computing*, pages 563–574. Springer.

[Yang et al., 2017] Yang, M., Tu, W., Wang, J., Xu, F., and Chen, X. (2017). Attention-based lstm for target-dependent sentiment classification. In *Proceedings of the thirty-first AAAI conference on artificial intelligence*, pages 5013–5014.

[Yang et al., 2020a] Yang, N., Dey, N., Sherratt, R. S., and Shi, F. (2020a). Recognize basic emotional statesin speech by machine learning techniques using mel-frequency cepstral coefficient features. *Journal of Intelligent & Fuzzy Systems*, 39(2):1925–1936.

[Yang et al., 2020b] Yang, X., Xiong, Z., Li, Z., Li, X., Xiang, W., Yuan, Y., and Li, Z. (2020b). Perceived psychological stress and associated factors in the early stages of the coronavirus disease 2019 (covid-19) epidemic: Evidence from the general chinese population. *Plos One*, 15(12):e0243605.

[Yang et al., 2013] Yang, Y., Fairbairn, C., and Cohn, J. F. (2013). Detecting depression severity from vocal prosody. *IEEE transactions on affective computing*, 4(2):142–150.

[Yokoi and Diedrichsen, 2019] Yokoi, A. and Diedrichsen, J. (2019). Neural organization of hierarchical motor sequence representations in the human neocortex. *Neuron*, 103(6):1178–1190.

[Yu et al., 2016] Yu, W., Yang, K., Bai, Y., Xiao, T., Yao, H., and Rui, Y. (2016). Visualizing and comparing alexnet and vgg using deconvolutional layers. In *Proceedings of the 33 rd International Conference on Machine Learning*.

[Yuan et al., 2020] Yuan, L., Tay, F. E., Li, G., Wang, T., and Feng, J. (2020). Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911.

[Zadok-Gurman et al., 2021] Zadok-Gurman, T., Jakobovich, R., Dvash, E., Zafrani, K., Rolnik, B., Ganz, A. B., and Lev-Ari, S. (2021). Effect of inquiry-based stress reduction (ibsr) intervention on well-being, resilience and burnout of teachers during the covid-19 pandemic. *International Journal of Environmental Research and Public Health*, 18(7).

[Zandifar et al., 2020] Zandifar, A., Badrfam, R., Yazdani, S., Arzaghi, S. M., Rahimi, F., Ghasemi, S., Khamisabadi, S., Mohammadian Khonsari, N., and Qorbani, M. (2020). Prevalence and severity of depression, anxiety, stress and perceived stress in hospitalized patients with covid-19. *Journal of Diabetes and Metabolic Disorders*, 19(2):1431–1438.

[Zhang et al., 2016] Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.

[Zhang et al., 2019] Zhang, Y., Zheng, J., Jiang, Y., Huang, G., and Chen, R. (2019). A text sentiment classification modeling method based on coordinated cnn-lstm-attention model. *Chinese Journal of Electronics*, 28(1):120–126.

[Zhao et al., 2019] Zhao, J., Mao, X., and Chen, L. (2019). Speech emotion recognition using deep 1d & 2d cnn lstm networks. *Biomedical signal processing and control*, 47:312–323.

[Zhao et al., 2020] Zhao, S., Ma, Y., Gu, Y., Yang, J., Xing, T., Xu, P., Hu, R., Chai, H., and Keutzer, K. (2020). An end-to-end visual-audio attention network for emotion recognition in user-generated videos. *CoRR*, abs/2003.00832.

[Zhao et al., 2021] Zhao, Z., Li, Q., Zhang, Z., Cummins, N., Wang, H., Tao, J., and Schuller, B. W. (2021). Combining a parallel 2d cnn with a self-attention dilated residual network for ctc-based discrete speech emotion recognition. *Neural Networks*, 141:52–60.

[Zhou, 2020] Zhou, D.-X. (2020). Universality of deep convolutional neural networks. *Applied and computational harmonic analysis*, 48(2):787–794.