I don’t know how to start this shit, yo.. now

Summary

Write your summary here...

Preface

Write your preface here...

Table of Contents

List of Tables

List of Figures

Abbreviations

ANN	=	*Artificial* Neural Network
RC	=	Reservoir Computing

Chapter 1

Introduction

> They're trying to understand what space is. That's tough for them. They break distances down into concentrations of chemicals. For them, space is a range of taste intensities.
>
> Greg Bear, Blood Music

Countless manhours have been spent improving the design and manufacturing process of the digital computer, creating more and more complex architectures capable of operating at ever greater speeds. The brain is not subject to this top down design philosophy, yet through a process of self organization neurons are capable of forming highly complex networks capable of solving complex tasks, with far greater energy efficiency, robustness and parallellism than any designed processor. Inspired by work done in the field of material computing systems such as the mecobo platform of nascence [citation needed], living neural networks grown from human stem cells *in vitro* on *Micro Electrode Arrays* (MEA) are interfaced with a digital computer, forming a hybrid neuro-digital system. This system utilizes the theoretical framework of *Reservoir Computing* to help translate between the digital and biological parts of the system, allowing it to solve simple tasks as a proof of concept. The ultimate goal of the cyborg is to further our understanding of the underlying principles that governing how nature computes.

1.1 Complexity

In the 50's and 60's there was much optimism in the burgeoning field of artificial intelligence. In 1965 H. A. Simon claimed "machines will be capable, within twenty years, of doing any work a man can do."[8] , while Marvin Minsky boldly claimed in 1967 that "Within a generation ... the problem of creating 'artificial intelligence' will substantially be solved." [1]. Had they chosen to predict any other field, such as logistics, information sharing or communications their statements would have been prophetic and visionary, so why

did artificial intelligence turn out so differently? The researchers sought to make machines that could use logic similar to that of high level human thinking. Therefore, it followed that the machine had to be programmed with rules governing logic in order to reach sound conclusions. To represent the prior and deduced knowledge, the researchers designed programming languages such as lisp that could accurately describe these operations. In order to actually execute these lisp programs hardware had to be created, supporting the primitive operations such as addition, subtraction and loading from memory. Regardless of the underlying platform a lisp program did not change meaning, and the binary adder in the heart of the processor never interacted with the floating point adder. In short, each piece of the puzzle was self contained, and the *Complexity* of the system was similar to that of the sum of its parts, which made it feasible to build large systems. Nature, on the other hand, applies a completely different method. Complex structures appear with no blueprint, arising from a process of *self-organization* driven by a set of growth rules. This self organizing process is capable of producing incredibly complex, robust and diverse structures whose functionality arises not from specialized components working in isolation, but from the interaction of many components. Applying the superposition principle to the processor makes sense, it allows us to study each individual component in isolation before investigating how they interact. On the other hand, applying the superposition principle to nature leaves us blind to the fact that the purpose of most components is to interact rather than performing a specific function.
It seems the reason the researchers failed was their underestimation of the role played by complexity, believing it was an incidental product of evolution rather than a necessity.

1.2 Computation

The invention of the digital computer will be remembered as one of, if not the most significant technological advances of mankind. This is fitting, because it very clearly demonstrates the differences in approach between the top down engineering approach of humans, and the self organization of nature. Since the components of a processor and a program is isolated and specialized it is completely necessary that each component behaves reliably, as even the slightest miscalculation can throw the whole system off balance. Because of this a processor has to run all its instructions in an ordered fashion, parallelism can at best be achieved by running sequential programs at the same time. In spite of these weaknesses, the digital computer is so ubiquitus that other approaches have been dubbed *Unconventional Computing*. Unconvential computing, as implied by the name, comes in many forms such as buckets of water [3], or blobs of carbon nanotubes [cite nascence]. These approaches seek to utilize the self organizing collective behaviors of naturally occuring processes in physical systems, utilizing the interactions that arise from the collective behavior of the system to perform calculations, rather than the calculated sequential activation of specialized heterogenous components of traditional digital computers. As an example, consider the effort spent modelling and simulating snow [Cite SIGGRAPH frozen paper] used in motion pictures such as Disney's frozen. Dozens of machines in large rendering farms spend weeks rendering the snow in final movie, however if you bring some dynamite and a helicopter nature will gladly provide you with an avalanche "for free".

Toffoli argues that “Nothing makes sense in computing except in the light of evolution” [7], Perhaps the crowning achievement of evolution is the human brain, capable of performing vastly complex tasks, however only recently has understanding the brain from a computational perspective become feasible.

The vast complexity of the human brain has made it a very difficult subject to study and copy. Rather than understanding the human brain as a whole a more feasible approach is to understand the underlying processes that allow neural networks to self-organize into computationally capable networks. In [Cite DeMarse flight controller] a neural network is grown in an MEA and interfaced with a flight simulator. [Cite AHDNN] follows a similar approach, using neurons to control a simple robot. The contribution presented in this paper builds on this work, but adding RC...

Chapter 2

Background

In this body of work references are made to computations done by both artificial and real neurons. To make the distinction between these cases clear all computation done by computer simulated approximations of neurons will be prefixed as artificial.

The underlying goal of the thesis, performing computations with neurons serves as a red thread throughout the background chapter. To create a cyborg, it is necessery to create a bridge between machine and neural cultures. This bridge must provide a medium which can be understood by both neural tissue and the computer, analogous to how humans communicate with soundwaves and visual cues. Once a medium has been established, the real challenge presents itself, namely that of finding a common “language”. The topics discussed in this section serves to motivate how this goal can be achieved is structured as follows: First *Complex Systems* are introduced as a framework to discuss the computational capabilities in a wide range of systems that exhibit system dynamics similar to that of neurons. Next, *Evolution in Materio*, EiM for short, introduces computation done in unstructured matter through the process of evolution. The goals of EiM are closely aligned to the goal of this thesis, as both study massively parallel computation happening in physical matter shaped by the process of evolution. Next section, *Neurons As Computers*, gives a basic overview of neurons, and more importantly motivates narrowing the scope of how neurons will be modeled, a necessity considering the size of the field and the background of the author. Finally, *Reservoir Computing* is introduced, tying together the previous sections by introducing a framework of thought capable of tapping into the computational properties of neural networks.

2.1 Complex Systems

TODO: Introduser Adaptive networks, siter Sayama: ”Modeling complex systems with adaptive networks”

Nature, unlike humans, does not shy away from complexity. Since complexity in itself is not the goal of evolution (evolution has no goal), it is interesting to note that systems that

arise from a process of evolution display much more complexity than human-designed systems, and the study of these systems provide useful models and terminology for describing and understanding how neural networks form. The study of these *Complex Systems* is a broad, cross-disciplinary field. To see why, consider that most fields deal with systems, be they financial, chemical, social or electrical in nature. All of these fields feature systems in which the global behavior *Emerges* from local interaction of individual components through a process of *Self Organization*. These behavior of these systems exhibit a complexity that is greater than the sum of complexity of its individual components, i.e the relationship between the complexity of the system and its constituent components is *Non Linear*. Common for these systems is therefore the futility of reductionism since the interesting aspects of the system only occurs when components interact. Although complex systems are not ordered or linear, they are not chaotic either. Unlike chaotic systems, complex systems contain *Attractors*, states that are relatively stable. Fig [Strange attractor] shows the *Strange Attractor*, a pattern formed by a particle in a vector field that orbits two attractors, arbitrarily switching between them. In complex systems both dampening and amplifying feedback loops form. The dampening loops allow the system to fall into the attractors, while the amplifying loops can amplify minor perturbations, eventually causing a cascading feedback loop moving the system to a different attractor.
The immediate benefit of classifying neural cultures as a complex systems is that studying simpler models of complex systems can act as a stepping stone. A simple, biological inspired model capable of exhibiting complex behavior is the *Cellular Automaton*, a discrete model of a single cell which changes between a discrete set of states based only on its immediate neighbors. Together they are capable of solving global problems such as contour-extraction [6], providing an example of local interactions producing interesting global behavior.
Cellular automata are even sufficiently powerful to express a turing machine, but as Sipper puts it: “This is perhaps the quintessential example of a slow bullet train: embedding a sequential universal Turing machine within the highly parallel cellular-automaton model.” Embeddeding turing machines into cellular automatas is of little use, but it’s useful to know that cellular automata are sufficiently powerful if we are to apply it as a model for the processes governing neural networks. The real power of cellular automatas as a model for neural networks is how they model the *Phase Transitions* in behavior (i.e dynamics). In Langton’s pioneering paper *Computation on the Edge of Chaos* [4] the system dynamics of cellular automata are shown to follow phase transitions similar to physical matter. Langton explored the rule space of cellular automata and found that the ratio between transitions that led to cell death and life had similarities to temperature in physical systems. As expected, rules which tended to favor cell death led to static or periodic systems, while rules favoring life over death led to chaotic systems. More interstingly is what happened when the rules favored life and death equally. In these systems which exists at the border between orderly and chaotic systems Langton found a *critical* phase where the system was neither chaotic nor ordered. It is important to note that Langton did not seek to solve a specific problem with his automatons, but to explore which automatas capable of supporting universal computation, hypothesized by Wolfram [9]. Criticality applies to any dynamic system, not just cellular automata, and the study of adaptive networks [5] suggests that many systems exhibit a homeostatic regulation of system dynamics to ensure that it stays

in the critical phase, including neurons [2].

2.2 Evolution In Materio

Classifying neurons as a complex system has not gotten us closer to a solution for interfacing with them in a meaningful way it seems. However, one useful conclusion is that neurons work nothing like human designed conventional computers, and that to understand how neurons compute we should look at more *unconventional computing*, specifically work done on physical matter. Two pioneers in this field were the british duo Pask and Beer which studied how unstructured matter could be used to perform computational tasks. In the introduction Toffolis statement that computation does not make sense except in the light of evolution seems to contradict this notion of computation, but evolution is not reserved only for nature.

In one experiment [cite ???] the duo used silver in an acidic solution which would form short-lived silver filaments when subjected to electric currents. From our perspective of computation, it was not before they tuned the parameters of the system in order to evolve a tone discriminator that the system could truly be classified as a computing one. Evolution in materio represents a very different approach than conventional processors. While it is impossible to program unstructured matter with imperative instructions like in a conventional computer, we can nonetheless “instruct” the material computer in a declarative matter by specifying what sort of result we are after and letting the evolutionary process handle the rest. Material computing gives us a much better model of how neurons can compute: Material computation happens on a massively parallel scale, it occurs in a substrate where all structure is self-organized rather than imposed by a designer, and both are products of evolution.

TODO: Write about Odd Rune’s stuff, clean up and add citations.

2.3 Neurons As Computers

It might seem that the previous section has provided the key to unlocking the computational power of neural cultures. This line of thinking neglects the fact that neurons have already been shaped by the process of evolution over billions of years, indeed neural cultures can be viewed as the result of an EiM experiment a billion years in the making. Our goal is not to apply the principle of EiM to neural cultures, but it does provide an interesting angle: When Pask and Beer tweaked their silver solution to discriminate tones, they did viewed the solution as a black box, and only its performance on the task at hand was evaluated. While evolution does not optimize for a specific functionality the black box approach is similar. Just like Pask and Beer did not consider the exact inner workings of silver filaments, it is necessary to apply the black box liberally when approaching neurons.

If the neuron was the result of an EiM experiment, the ideal model of the neuron would be the criteria used to evaluate fitness, eschewing all implementation details. Of course, no such criterias exist, evolution does not have a plan, but it illustrates why as much as possible of the cells inner working will be considered to be inside a black box.

2.3.1 Neurons

The neuron, or nerve cell, is the basic building block of both the human brain and the nerve system. There are many types of neurons in the human body, but to reduce the scope the focus of this thesis is a simplified model of the neurons that make up the brain. The model neuron, shown in fig [a figure of a neuron] consists of three parts: The body, *Soma*, the *Dendritic network* and an *Axon*. The dendritic network acts as a receptor sensing electrical activity around the neuron, while the axon transmits electric pulses to neighboring cells. The connection between two neurons is called a *Synapse*. Axons and dendritic networks are themselves vastly complex, and viewing them as simple electrical transmitters neglects the importance of neurotransmitters. However, given the strong correlation between neurotransmitter concentrations and electrical activity is considered part of the black box. When the neuron is sufficiently excited by electrical activity it will fire an electrical pulse that travels along the axon, which in turn stimulate other neurons. Together neurons form networks in a complex interplay between topology and behavior: The behavior of the network decides its behavior, and the behavior in turn causes some synapses to wither, and others to form in a process that is not well understood. The model used for the neuron for the rest of this paper is a node in an *Adaptive Network* which communicates through electrical pulses. The missing part in our model is the underlying rules that dictates the growth of the network.

2.4 Reservoir Computing

So far we have arrived at a model of the neuron as a complex adaptive network which can be interfaced with using electrical signals, but the fundamental issue of actually utilizing neurons for computing remains. The final piece of the puzzle comes in the form of *Reservoir Computing*, a technique developed to exploit the dynamics of complex systems. If the fundamental rule governing the neuron is to create networks exhibiting the same complexity behavior as Langton's automatas then harnessing the computational capabilities of these simple models is a first step towards interfacing and understanding neurons. In reservoir computing, a complex systems is used as a *reservoir* [**?**] which "acts as a complex nonlinear dynamic filter that transforms the input signals using a high-dimensional temporal map, not unlike the operation of an explicit, temporal kernel function."

In order to explain, schrauwen makes a comparison to the the machine learning technique of source vector machines work, as shown in fig [rm 1]: The reservoir acts as a kernel, projecting input into a high-dimensional feature space. Figure [rm 1] shows this technique, note that the regression performed upon the feauture space is a simple linear regression, an important point both in SVMs and reservoir computing. Figure [rm 2] shows a typical reservoir computing setup which follows a similar method of operation as the SVM in figure [rm 1] The reservoir serves as the high dimensional feature space, while the output layer is only capable of linearly separating the resulting dynamics. Schrauwen points out two major differences between SVMs and RCs. First, SVMs only implicitly expands the input to high dimensional space in order to make the problem tractable, while reservoirs do not. Secondly, kernels are not capable of handling temporal signals. The second difference is very important, it is what allows reservoirs to implicitly encode temporal signals

in their dynamics, making reservoirs a natural fit for tasks such as speech recognition. In other terms, the properties that make complex systems so hard to work with such as sensitivity to initial conditions also allow them to discern very subtle nuances in input, and their complex behavioral patterns causes the systems to change their behavior to new input based on previous input. In light of this, asking how to build a computer using Langton's automatons is the wrong question, instead the focus should be on how exploit the computation that is already occuring.
There are many examples of reservoirs which have been successfully exploited: In [**?**] an *echo state network* is utilized to solve classification problems. More esoteric reservoirs have been used, for instance in [**?**] the idea of reservoir computing is taken quite literally using a bucket of water as a reservoir.

2.4.1 Linear and nonlinear output layers

TODO: Elaborate on the use of linear vs nonlin classifiers.

Chapter 3

Making Of A Cyborg

> What is real? How do you define 'real'? If you're talking about what you can feel, what you can smell, what you can taste and see, then 'real' is simply electrical signals interpreted by your brain.
>
> Morpheus to Neo - The Matrix

3.1 Concept

Shown in figure [rm 3] a conceptual cyborg is shown. This conceptual cyborg is comprised of three main components: An *MEA*, short for *Micro Electrode Array* in which a biological neural network is grown. A *neural interface*, allowing two-way communication between the neural network and the outside world. A robotic body, responding to movement commands and equipped with a sensor allowing it to perceive its environment. In this concept neural readouts are transformed into a Left and Right signal by a feed forward neural network, controlling the direction of the robot. Simultaneously data retrieved from the sensors of the robot are processed in a feedback processor and fed back to the neural network. The conceptual cyborg is a closed loop system: The only input to the system is what the sensors perceive which the cyborg must act upon.
The major challenges involved in realizing the cyborg is divided into two major areas. Firstly, the infrastructure, comprising of everything from moving data between computers, to the neurons themselves must be constructed and implemented. This challenge can in turn be divided into software and hardware, the latter being the focus of this chapter. The second challenge is actually configuring the artificial feed forward neural network such that the cyborg does something that we deem useful. This task is done in software, and will be covered together with the software part of the first challenge in the next chapter.

3.2 Platform Architecture

The centerpiece of the cyborg is the neural culture, but for all their robustness they cannot survive long outside the laboratory, and should therefore not be physically connected to the robot. While this decoupling the neural cultures from the physical robot is a practical necessity, it also has far reaching consequences on the design space, and even philosophical ramifications. The decoupling between culture and machine means that the primary focus is to expose the neural neural interface to the rest of the world, which is done over network in this project. With this architecture the robot itself becomes less interesting, it does not even have to be a physical robot at all, in fact all work presented in this thesis has been done with a fully virtual robot.
Similar network architectures have been implemented, in [Cite Application] a neuron culture is used to control a simple wall-avoiding robot as a proof of concept.

3.3 Wetware

As opposed to hardware and software, the term wetware describes system components of biological origin, i.e "wet" components. The wetware of the cyborg is thus the neural networks which are being grown in MEAs at the department of neuroscience. The MEAs are seeded with neural stem cells of either human or rat origin which then spontaneously form networks. At seeding there is no network at all, only a "soup" of dissociated neurons which over the course of several weeks start forming networks.

As the networks starts "maturing" a common phenomenon is neurons firing monotonic spikes automatically. The activity from these so-called pacemaker neurons can be seen in **??**. In the figure each cell in the grid corresponds to one of the electrodes as seen in **??**, however at this stage the monotonic spiking activity tends to be transient, starting and stopping randomly.

3.4 Neural Interface

An *MEA2100* system has been purchased from the lab equipment vendor multichannel systems. This system system is built to conduct in-vitro experiments electrically active cell cultures such as neurons contained in micro electrode arrays. Essentially, the MEA2100 provides very precise electrodes for inducing voltage and measuring. In addition to voltage and current the equipment comes with temperature controls allowing neural cultures to survive for prolonged experiments, allowing experiments on live neurons to last up to 30 minutes. The essential components of the MEA2100 neural interface as shown in fig ??? are as follows:

3.4.1 Micro Electrode Array

Shown in fig ???, the *MEA*s used in the cyborg project has been procured from multichannel systems. These MEAs feature 60 electrodes (with one of these serving as reference voltage). Each MEA is seeded with a neural culture, meaning that once seeded an MEA will be the host of a single culture, each capable of living for over a year. Fig ??? shows a seeded MEA, nicknamed “Frank” as a play on frankensteins monster due to it’s tendency to develop *Organoids*, small proto-organs with structures similar to eyes and other sensor organs.

3.4.2 Headstage

The headstage is responsible for performing the actual readings and stimuli on the MEA. The electrodes of the MEAs are measured and stimulated by the headstage which contains the necessary high precision electronics needed for microvolt range readings. Fig ??? shows the headstage used by the cyborg project.

3.4.3 Interface board

The interface board connects to up to two head-stages and is responsible for interfacing with the data acquisition computer, as well as auxiliary equipment such as temperature controls. The interface board has two modes of operation. In the first mode the interface board processes and filters data from up to two headstages as shown in **??** which can then be acquired on a normal computer connected via USB. In the second mode of operation a Texas instruments TMS320C6454 digital signal processor is activated which can then be interfaced with using the secondary USB port as shown in **??**

Chapter 4

30000 Lines Of Code

> Look at you, hacker. A pathetic creature of meat and bone. Panting and sweating as you run through my corridors. How can you challenge a perfect immortal machine?
>
> SHODAN - System Shock

Creating a cyborg is a massive undertaking, thus a quite extensive software suite has been created to make research feasible. This chapter will focus on the more mundane engineering aspects of the software developed for this thesis while the next chapter will focus on the reconfigurable reservoir computing module at the heart of the project. The following section contains an overview of the design philosophy, and an overview of the individual components which are discussed in the remainder of the chapter.

4.1 Design Philosophy

As alluded to in the discussion of physical vs virtual robot in the previous chapter, modularizing the cyborg is important not only as a practical matter. While the chief purpose of the software system is to interface with neurons, the design does not require this. In fact, the guiding principle of the software system is to serve as a general reservoir computing system, where neurons serve as one of many possible reservoirs. This deviates from the conceptual cyborg presented in the previous chapter as the current design does not require a feed forward neural net to be used to process reservoir output.

4.1.1 Implementation

The system today is comprised of two separate projects, *MEAME* and *SHODAN*. The first project, MEAME, a “clever” pun on the similarity on the words MEA and “meme”, is responsible for interfacing with the lab software, thus in the framework of the design philosophy it serves as an implementation of a single reservoir.

4.2 MEAME

The MEAME project is responsible for handling the tasks closest to the neural reservoir, and is therefore installed on a computer directly connected to the MEA2100 system. MEAME is split into two subsystems, both exposed by a unified *REST interface*, enabling access to the lab equipment via a protocol built on top of HTTP.

Both MEAME and MEAME-DSP are available online, and have been implemented by the author of this thesis.

The data acquisition part of MEAME is responsible for configuring recording parameters such as samplerate and starting or stopping recordings. It is built on top of a very thin windows only API provided by the equipment vendor, making it abundantly clear that the cyborg project is pushing the equipment far beyond its intended use, which at the same time is a point of pride and pain. Once the data acquisition equipment has been configured MEAME exposes the recorded data as a single continous stream which can be accessed via TCP. Figure ??? shows the format of this data stream, which must be demultiplexed by the receiver in order to separate data into the 60 individual channels. In order to stimulate the neurons, the HTTP interface can also handle stimuli requests, which are executed on a realtime capable digital signal processor which is embedded on the lab equipment itself.

4.2.1 MEAME-DSP

When MEAME receives a stimuli request this is communicated to the DSP through a simple protocol implemented by directly writing to registers. The DSP maintains a list of *Stim groups* which contain a list of electrode numbers and a desired frequency. Additionally the DSP is responsible for uploading which pattern to use when stimulating an electrode, however the DSP itself does not expose an API for this purpose since uploading a stimulus pattern simply consists of writing repeatedly to a single register, meaning this task is better handled at a higher level.

4.3 SHODAN

SHODAN is a reservoir computing system, developed to be able to handle any reservoir as long as the necessary transformations between reservoir output and perturbing are supplied. Although SHODAN is reservoir agnostic, a significant portion of the code is specific to neural cultures and communicating with MEAME, thus SHODAN is not a framework in the traditional sense. While all the subcomponents are reservoir agnostic, the main focus will be on the specific implementations for interfacing with neural cultures. SHODAN is comprised of the following subsystems: (Figure here???)

4.3.1 Data Acquisition

Bibliography

[1] , ???? Marvin Minsky - Wikiquote.
URL https://en.wikiquote.org/wiki/Marvin_Minsky

[2] Bornholdt, S., Rohlf, T., Jun. 2000. Topological Evolution of Dynamical Networks: Global Criticality from Local Dynamics. Physical Review Letters 84 (26), 6114–6117, arXiv: cond-mat/0003215.
URL http://arxiv.org/abs/cond-mat/0003215

[3] Fernando, C., Sojakka, S., Sep. 2003. Pattern Recognition in a Bucket. In: Advances in Artificial Life. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 588–597.
URL https://link.springer.com/chapter/10.1007/978-3-540-39432-7_63

[4] Langton, C. G., Jun. 1990. Computation at the edge of chaos: Phase transitions and emergent computation. Physica D: Nonlinear Phenomena 42 (1), 12–37.
URL http://www.sciencedirect.com/science/article/pii/016727899090064V

[5] Sayama, H., Pestov, I., Schmidt, J., Bush, B. J., Wong, C., Yamanoi, J., Gross, T., May 2013. Modeling complex systems with adaptive networks. Computers & Mathematics with Applications 65 (10), 1645–1664, arXiv: 1301.2561.
URL http://arxiv.org/abs/1301.2561

[6] Sipper, M., Jul. 1999. The emergence of cellular computing. Computer 32 (7), 18–26.

[7] Tommaso Toffoli, ???? Nothing Makes Sense in Computing Except in the Light of Evolution - Semantic Scholar.
URL /paper/Nothing-Makes-Sense-in-Computing-Except-in-the-Lig-Toffoli
5e1e174e2a470e2e4a0356ea21444ae6f5a284ee

[8] Vardi, M. Y., ???? Artificial Intelligence: Past and Future.
URL https://cacm.acm.org/magazines/2012/1/144824-artificial-intelligence-past-and-future/fulltext

[9] Wolfram, S., Jan. 1984. Universality and complexity in cellular automata. Physica D: Nonlinear Phenomena 10 (1), 1–35.
URL http://www.sciencedirect.com/science/article/pii/0167278984902458

Appendix

My appendix was surgically removed. Take that, evolution.