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.

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? To answer, consider the approaches the researchers employed:
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.

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

2.1 Complex Systems

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.

Creating a bridge between machine and neural culture is not useful in itself without a way to interpret the electrical activity measured from the network, and a way to encode information as electrical pulses to be transmitted back to the network. To harness the computational power of neural networks it is necessary to employ a simplified model of neurons which is capable of describing how they self-organize into computationally capable networks, while leaving out leaving out the implementation details. Rather than starting with modeling neurons, which are, at least from our point of view, the apex of evolution, simpler biological systems can be studied. While nature exhibits a staggering diversity in both form and function there are also many fundamental similarities, after all they have all emerged from the same evolutionary processes. Eco-systems, ant-hills and social networks all exhibit complex non-linear behavior, where the global behavior of the system cannot be traced back to a single part. Natures acceptance of non-linearity fundamentally differs from products engineered and designed by humans. Our designs carefully contain complexity: We carefully design large systems out of individual heterogonous components which can be understood in isolation, creating clockworks where the complexity is a sum of parts. In all of natures systems the behavior global behavior arises ultimately from the interplay of homogenous components. While these homogenous components may organize themselves into complex specialized units such as the heart, these complex components not only consist of homogenous cells, they are also assembled by locally interacting cells with no blueprint, knowledge, or even an appreciation of the scale that the final complex component will operate in. This process of *self organization* is by necessity highly robust. While every heart has the same functionality, pumping blood, there are no two identical hearts, even in identical twins there are variations, should you look carefully enough.

In these *Complex Systems* positive feedback can loops amplify small perturbations into cascading effects, changing the entire system, while negative feedback loops may cause other states to be relatively stable, resulting in multiple meta-stable states, so called *attractors*, that give the system some measure of order.
The first step in creating a theoretical framework for bridging the gap between neuron and machine is then to study simpler bio-inspired models exhibiting this complex behavior, such as *cellular automata*. A cellular automaton is a model of a single cell that will change its state based only on its immediate neighbors. They are capable of solving global problems such as contour-extraction [6], establishing that local interactions can produce 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 Reservoir Computing

The theory behind complex systems provides us with a useful model of biological computation with emphasis on how a *computational substrate* can be achieved. If the fundamental purpose of a 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. Clearly designing a cellular computer in a top down manner is intractable due to the intricate and unpredictable relation between cause and effect. The best we could realistically achieve with the typical top down approach is implementing a turing machine which would be both slow and incredibly brittle. Seemingly, classifying the neural culture as a complex system has not provided any useful tools for understanding how to interact with it, on the other hand it makes this task seem futile. However, in computer science the recent field of *reser-*

voir computing has emerged, embracing the complexity and unpredictability of certain complex systems. 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 how source vector machines work: The reservoir acts as a kernel, projecting input into a high-dimensional feature space. Figure ??? 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 ??? shows a typical reservoir computing setup which follows a similar method of operation as the SVM in figure ???. 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.

2.2.1 Linear and nonlinear output layers

2.3 Evolution In Materio

2.4 Neurons As Computers

Even a single neuron is an extremely complex system when compared to the transistor. Together they form vast interconnected networks where information is exchanged using electrochemical signals. A fundamental difference between neurons and conventional computers is that these networks have not been designed, they do not come with a blueprint, and they constantly modify themselves in response to the environment. Due to the complexities of this process it is completely necessary to restrict ourselves to a simple model of the neuron, leaving genetic activations and chemical pathways to the neurologists and chemists. In order to understand how neurons can assemble into complex networks capable of thought one must of course understand the underlying chemical and biological principles, but from a computer scientists perspective these are mere implementation details necessitated by existing in a physical universe. Were the physical laws altered the chemical pathways would surely differ, but the resulting behavior would probably [According to who?] still be similar to our universe. In this viewpoint, the very essence of neurons is their ability to self-organize into structures in a bottom-up fashion in a complex interplay between behavior and form. Following this rationale the neuron will be seen as a simple node in a network, communicating using electrical pulses, so-called spikes or action potentials. These spikes are short bursts of electricity which after firing causes a quick refactory period in which the firing cell will not respond to stimuli. When fired these spikes stimulates connected neurons, which may in turn release their charge causing a signal to cascade in a feedback loop. These electrical pulses can be seen as the “language” of neurons, and by measuring electrical activity and stimulating the network using electrodes, it is possible to set up two-way communication between a machine and a cluster of neurons.

2.5

Chapter 3

Making Of A Cyborg

This chapter describes the physical components, biological and mechanical, used to create a cyborg.

3.1 Concept

A biological neural network is grown in a *MEA*, short for *micro electrode array* which contains electrodes which interface with neurons using electrical signals. These MEAs are then *embodied* in a physical robot which is controlled by the electrical signals as shown in [ref cyborg concept] This system is a *closed loop*, it does not require any outside intervention, such as a human controller using a joystick to operate, it's simply driven by the neurons response to what the sensors of the machine senses.

3.2 Platform

Providing an interface between neurons and a computer allows using neural network for computation, however it is impractical to move the neural cultures outside of the laboratory. Rather than moving the cell cultures outside the safe confines of the incubator, the robot and cell cultures are linked over a network connection. Thanks to this decoupling the cell cultures do not have to be embedded in a physical robot, instead they can be interfaced with any robot connected to the internet, and even to virtual robots that only exist as a simulation.

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. In contrast with previous work the robot used in the NTNU cyborg project is a sophisticated robot which is programmed to move by itself, follow a person, take a selfie with someone and upload it to facebook, and even perform a secret handshake. By utilizing an already functioning robot as a host the cyborg project can focus on extending the capabilities of the robot, making it a true hybrid between digital and cellular computing.

3.3 Growing Neurons In Vitro

The neuron cultures used in the cyborg 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 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 Neuron Interfacing Hardware

The hardware used to interface with neuron cultures for the cyborg is an *MEA2100* system purchased from multichannel systems. The MEA2100 system is built to conduct in-vitro experiments electrically active cell cultures such as neurons. The principal components of the MEA2100 systems are:

3.5 Micro Electrode Array

Introduced in the previous chapter, the *MEA* is equipped with an array of microscopic electrodes capable of sensing and delivering voltages to and from nearby neurons. **??** shows an empty MEA, **??** shows an MEA used by the department of neuroscience with a live neuron culture.

3.6 Headstage

The electrodes of the MEAs are measured and stimulated by the headstage which contains the necessary high precision electronics needed for microvolt range readings. **??** shows the same type of headstage used in this paper along with an MEA.

3.7 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

An RC Cyborg Platform

This chapter provides an overview of the components comprising the final system. fig??? shows an early idealized version of the cyborg which remains the core focus of the system. In this ideal cyborg an interface connects the biological nerual network to an ANN readout layer, which in turn controls a robot whose input is processed and relayed back to the neural network. With this guiding principle the final system has been designed with the following components.

Core Reservoir Interface
: Responsible for providing a bdirectional bridge between the reservoir and the outside world.

Data Processing
: Processes the signals from the biological neural network, as well as the sensory input from the robot to a neuro-compatible format.

Agent Control
: Embodies the biological neural network in a robot, either artificial or real.

These components are enough to satisfy the ideal version of the cyborg, but for a cyborg to function in a practical setting additional components are necessary:

Communication
: While the Core Reservoir Interface provides a bidirectional bridge, it is the Communications module that extends that bridge to any machine connected to a network.

Recording
: Data from reservoirs is stored in a database, making experiment data accessible to any computer connected to the network.

Online Reconfiguration
: The Online Reconfiguration module is responsible for providing the cyborg with a filter that interprets the data from the biological neural network which is achieved by reconfiguring the filter when the system is running to adapt to the current reservoir.

Together these components form a system, whose architecture shown in fig. ??? looks quite different from the ideal cyborg.

4.1 A Closed Loop Example

The final architecture is quite extensive compared to the ideal version, but it still exists to provide a closed loop embodiment of the neural network. To give an idea of how the system is used it is necessary with an example run.

Setup The user accesses a web interface provided by the main server (SHODAN) and configures the experiment. This entails setting up a database recording, selecting filter and parameters and selecting what sort of agent should be running.

Launch Once the experiment is configured the main server contacts the lab computer (MEAME) and requests data acquisition to start. MEAME creates a TCP socket which holds the data read from the reservoir.

Execution After setup is done the

4.2 SHODAN

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

Write your appendix here...