An Essay on Evolution of Spiking Neural Network "Animats" Implemented on an Analog Neuromorphic System.

Daniel Sander Isaksen

13th February 2020

Abstract

Will we ever build a neural substrate able to contain our human consciousness?

Contents

Chapter 1

Introduction

For the past 70 years, a myriad of researchers and developers has been on a quest for a human-made system that might replicate the human function and behavior [20]. The development of autonomous systems has gotten so far it can imitate human function efficiently, but only in a few use-cases at a time. Several scientists and philosophers have proposed theories and tools to evaluate the level of intelligence of a system. For example, Alan Turing proposed the Turing Test, inspired by the party game The Imitation Game, to assess whether a machine is intelligent [42]. Computer programs might contain autonomous functions that seem to stretch over such a broad field of capabilities that it might fool other agents for prolonged periods. However, with time, most illusions are broken. In the search for artificial intelligence, researchers are continuously developing new technologies and theories.

In 1943, the first mathematical neuron model was published by McCulloch and Pitts, which modeled the neuron as a logic-gate.[11]. Through improvements to the McCulloch-Pitts neuron, by scientists as Hebb [8], we eventually got the Perceptron. Introduced by Rosenblatt [34] and refined by Minsky [29], the Perceptron is a binary classifier which models synaptic plasticity and thus, enabled the first "self-**learning**" Artificial Neural Network (ANN). ANNs, in its many forms, runs very well on modern computers based on works by Von Neumann and Alan Turing [31], theoreticians that were highly supportive of "learning" machines. ANNs, especially with the method of Deep Learning, pioneered by LeCun et al. [23], are now the most used tool for developing modern artificial intelligence technologies [20]. The use of deep learning on modern-day computer architecture has been proven very efficient for most computational tasks but also very inefficient for intricate tasks that are easily solved by biological systems. When trying to solve these tasks on ordinary computers, the complexity and size of computational models are increasing so fast, that the energy usage of computation has become a real environmental concern [**computation_energy_environment**].

As our understanding of neurobiology has progressed, more advanced models have been developed, including details of the ion flows and proteins that affect the membrane potential [9]. These models are often called Leaky Integrate and Fire Neurons, and the neural networks are often called Spiking Neural Networks (SNN). As biology has been the source of inspiration for most successful architectures of artificial intelligence [29], SNNs might be the next step towards an agent of low-cost human-level intelligence. Even though our computational power has grown exponentially for 70 years, the efficiency of our systems does not apply to extensive simulations running SNNs. Furthermore, a more suitable technology might be required, termed Neuromorphic Systems [17][38].

If a system can solve the same intricate tasks as a human, with ease, and also have a general behavior as a human, it could likely pass The Turing Test, but would it then have a human level of intelligence? Based on the plethora of theories regarding intelligence that exist today, the answer could be both yes and no. (ADD REFERENCE TO INTELLIGENCE DISCUSSIONS) If a system was to have a human level of intelligence, one could argue that it would need to experience like a human. Some scientists might question whether the system was conscious like a human and draw parallels to the experience of the system [39]. The Integrated Information Theory (IIT) is one of many theories questioning consciousness and intelligence, and it provides mathematical tools to assess information about the cognitive abilities of a system.

After unraveling the mystery of life, some scientists went on to question consciousness. Professionals use the term consciousness with different meanings, but **some philosophers argue that being conscious requires a subjective experience, and this is how the term is to be read in this essay.** If a system was to have a human level of intelligence, it would most likely need to have subjective experiences, and thus it would most likely need to be conscious. IIT comes with tools to evaluate the phenomenology of a causal structure, that is, how well a system integrates information based on experiences, and this might be a suitable tool to evaluate parts of the intelligence of a human-made system.

Most systems today that can be considered intelligent are developed on traditional computers. The problem with traditional computers is that they are highly modular, with different parts ultimately fulfilling their function in the bigger system. Modularity implies a low level of integration per definition, and it follows that all programs implemented on this arcitecture will have a low level of integration. Using IIT to evaluate the intelligence of a computer program with autonomous functionality then seems to be useless. However, some neuromorphic systems, built to resemble neural architecture, have more integrated architecture. A program written for one of these systems might be considered conscious by IIT and might lead to new insights into the requirements for creating a system with a human level of intelligence.

Chapter 2

Evolutionary Algorithms

Nature is exceptional at inventing new complex, resilient solutions through evolution of organisastions, from the molecular level to the ecosystems. In this project, biologically inspired Evolutionary Algorithms (EAs) are used to develop algorithms to run on hardware that has been developed with EAs. See section 4.2 on page 14 for more about Evolvable Hardware. In biology, the processes and structure of the cells of living cellular organisms are governed by Deoxyribonucleic acid (DNA). DNA is usually structured in **chromosomes** where the minimal units are called **genes**, and they determine the physical characteristics and development of the organism. In EAs the chromosomes may take form, for example, as strings, arrays or graphs. Similarly, genes associate to a segment in the chromosome, which encodes a building block of the final solution. Another term is **allele**. Individuals inherit two alleles for each gene, one from each parent. Alleles represent alternative forms of a gene that arise by mutation, and can for example result in different **phenotypic** traits, like different pigmentation. The aggregation of the alleles for a specific gene is termed the **genotype** for the gene. Although the genotype is the most influcencal factor for the phenotype, other factors also play an important role, like environmental factors. [40] In EAs, there is usually only one allele per gene, which results in genotypes normally being identical to the chromosome. The phenotype is usually the final solution of the EA.

EAs are in general population-based optimization algorithms, adopting a higher-level of abstraction of an evolutionary scheme found in nature. The classical Simple Genetic Algorithm (SGA) involves individuals, phenotypes, genotypes, fitness function, selection, recombination, and mutation. Every individual is a candidate in solution space, with their genes being the parameters of the solution, and the phenotypes being the resulting characteristics of the solution, when in contact with the environment. The fitness function is those parts of an environment that are designed to assess how good a solution fits with the problem. As in nature, only the fittest individuals are stochastically selected to recombine or survive, based on their more deterministic fitness levels. If only the fittest individuals were to be selected, then the search would most likely end in the first available local optima, before the solution space has been desirably explored. After the first selection, individuals are recombined, and their offspring mutated. Now, all the exploration steps are completed, for the current generation, and the fittest individuals of the current solution-pool are then brought to the next generation. The evolution typically runs until certain criteria have been met, or until a set number of generations has been reached. The best solutions can then be saved, and the rest wiped out, for a new epoch to start. [15] The SGA has inspired mind-boggling variations of algorithms under the relatively wide EA umbrella, and among these are the Island Model, NeuroEvolution, ..., and ... .

2.1 Standard Types of Recombination, Mutation and Selection

(NEED TO BE FILLED OUT) Typical methods for recombination are ... . Here is how these work. Typical methods for mutation are ... . Here is how these work.

Typical methods for selection are the roulette wheel, ... Here is how these work.

2.2 The Island Model

As a scheme for running multiple epochs in parallel, the heuristic of islands is probably adopted because of the natural isolation that occur on islands. The scheme normally includes migrations, which only happens a few times over an epoch of many generations. The Island Model can be useful for a more distributed exploration in the search, or for processing of a higher number of solutions per time, depending on the hardware.

2.3 NeuroEvolution

NeuroEvolution includes the use of neural networks in the optimization algorithm, where the networks are trained on a specific solution, which is then explored further using an EA.

Chapter 3

Spiking Neural Networks

Spiking Neural Networks (SNNs), often called The Third Generation of Artificial Neural Networks (ANNs), are designed after modern knowledge for neurobiology, with advanced neuron models based on what we know of synaptic connections between neurons, and what stimulates the membrane potential of the cells.[18] This is inherently different than networks of simple logic gates, like The Multilayer Perceptron, and it is thought to open up new horizons of biologically inspired computing.[11] [1] As the complex neuron model of the SNNs may suggest, the SNNs are expensive to simulate with binary computers, and some may argue that the Perceptron is good enough for the solutions that we would like to find using a logic-gate based computer. If we want to create autonomous systems that can efficiently mimick the capabilities of biological creatures, others may argue that systems closely related to the architecture of the very efficient brain, is the way forward.[24] [10].

Chapter 4

Neuromorphic Electronic Systems

Today, most developers consider the von Neumann architecture as the most efficient and scalable architecture for doing arithmetic computation. However, as our expectations of the functionality of the computer grow increasingly similar to the functionality of biological systems, we will require devices that are not arithmetic machines but rather "thinking machines." It is essential to consider how computer architecture usually has been modeled after then brain since the very beginning, but always at a simplified notion compared to the current knowledge of biology and psychology. George Boole's maybe most excellent work was his philosophical and mathematical An Investigation of the Laws of Thought, where the notion that human reasoning could be summed up by symbolic logic [6]. This work later inspired Claude Shannon to build the first "switching circuits" that facilitated the design for combinatorial logic circuits, which found the basis of our modern computers. John von Neumann draws parallels to the human neurons in his report on the EDVAC, which was used by Alan Turing as a source for understanding the design of a digital computer [31]. For further development of the modern computer and its algorithms, Boolean logic as been quintessential. The most successful artificial intelligence programs were, for a long time, only compositions of if-then sentences [20]. Through this brief investigation into the history of computation, it is safe to assume that modern arithmetic computer architecture bases itself on outdated knowledge of the best "thinking machine" that we know of; the human brain. Even though the human brain can perform arithmetics, it is an activity that consumes a lot more power than a modern cellphone does for the same amount of arithmetic work. However, the human brain performs other marvels, like image recognition, without spending much energy at all, to the very contrary of modern cellphones [28]. The brain has a small, underdeveloped center for arithmetics, while the core of the computer only performs arithmetics [14]. To successfully create machines that can efficiently perform functions similar to perception or even sensing, we ought, therefore, to explore solutions based on modern biology and psychology.

Even many of the founders of computing, like Ada Lovelace, Neumann, and Turing, admitted that the digital computer could not achieve the same functionality as the human brain. A more biologically inspired machine was attempted by Rosenblatt with the Perceptron as mentioned in **??** on page ??, and this is now the foundational algorithm for the ANNs powering state-of-the-art artificial intelligence, on modern von Neumann computers. However, modern computers are becoming increasingly power-hungry, and we are reaching the end of Moore's law [26] [30]. The very same man that coined the term "Moore's Law" after his friend George Moore proposed a theoretical system that could overcome the limits of modern digital computers. This man was Carver Mead, who proposed Neuromorphic Electronic Systems, as devices implemented and organized after the same principles as the nervous system. The idea is that the nervous system performs "computational" marvels with excellent efficiency and that this is mainly due to how the nervous system utilizes the physical characteristics of its units [26].

There are several challenges related to building such a neuromorphic system. An accurate brain model is needed to design better devices. To be able to build the devices, developers first need to research proper techniques and materials. Any utilization of the systems requires a programming framework. Lastly, the developers need to prove that the system has applicability to real-world problems. [38].

4.1 Models, Algorithms, Devices, and Frameworks

There are several popular categories of spiking neuron models.

Biologically Plausible and Biologically Inspired models incorporate some aspects of biological systems, but that has been proved useful in applications. An example is the Hodkin-Huxley model, which describes the electrical characteristics of neurons, much like a conductance-based electrical circuit. Another example is the popular Izhikevich model, which is simple but reproduces many of the computational features of the neuron. [38]

Integrate-and-Fire (IAF) models are varying on the complexity spectrum. Some are advanced, while others are less biologically realistic but useful. The model Leaky IAF is the most popular, and it comes with several additions, like Exponential Leaky IAF and Adaptive Exponential Leaky IAF. Typical SNNs usually incorporate one of these variations of IAF. [38]

Synaptic models typically include ion channels or Spike-timing-dependent plasticity (STDP). STDP can be simplified to cover: pre-synaptic and post-synaptic "weights," axonal restructuring, dendritic restructuring and neuronal excitability, Long-Term Potentiation, and Long-Term Depression. These are the processes of neuronal dynamics that are most important to model when building a neuromorphic system. The BrainScaleS system implements several of these STDP process, including LTP and LTD [27][37]. [38]

Network models are usually SNNs, and popular variations are Spiking Feed-Forward, Spiking Deep Belief, Spiking Hebbian, Spiking Hopfield, Associative Memories, Spiking Winner-Take-All, Spiking Probabilistic, Spiking Random. Common for all these is that the typical training is similar to traditional ANNs, which will not utilize the full potential of either the model or the neuromorphic system. [38]

Algorithms and Learning are specific to network characteristics, and can be divided into two types, on-chip training or transferred learning from off-chip training. Typically, online training is unsupervised, while offline training is unsupervised. Many researchers advocate that neuromorphic systems may have big self-learning capabilities that have yet to be proven. Catherine Schuman wrote about this: "In particular, we need to understand how to best utilize the hardware itself in training and learning, as neuromorphic hardware systems will likely allow us to explore larger-scale spiking neural networks in a more computationally and resource-efficient way than is possible on traditional von Neumann architectures." [38]

Devices and Hardware divides into three categories; digital, analog, and mixed analog/digital. The digital hardware consists of Boolean logic-gates and is usually synchronous a clock-based, but not always. FPGA is often used to either prototype or to give the feature of radically different network topologies, models, and algorithms. IBM TrueNorth is a digital system that has fully custom ASIC design, is partially asynchronous but uses a clock for necessary time steps. It has deterministic behavior but can generate stochasticity similar to what is possible with software. SpiNNaker is a fully custom digital, massively parallel system. It uses ARM integer chips, with custom interconnect to handle spikes. The interconnect is very advanced and highly flexible but makes the system less efficient than some other systems [17] [38]. Analog neuromorphic systems benefit from the physical characteristics of analog circuits and are often asynchronous, fuzzy, with the conservation of charge, amplification thresholding, and integration. As mentioned earlier, the first neuromorphic system proposed was a wafer-scale analog system [26]. The idea was to utilize the unique characteristics of transistors to make computing more energy efficient. Researches have used different types of Field-Programmable Transistor Arrays (FPTA), and these are often used for analog circuit design or to model neurons, synapses, or other components of the nervous system. Traditionally, computing with analog circuits is complicated because of global asynchrony and noisy, unreliable components. Researchers theorize that neuromorphic design might overcome these traditional issues and use evolutionary algorithms to help design circuits that better utilize the characteristics of the analog components. [22]. For digital communication and storage of information, analog neuromorphic systems often utilize some digital components. These systems can be classified as mixed analog/digital neuromorphic systems. An example of values to be stored digitally is synapse weight values, intrinsic information of the system, or time series of activity. The systems which use some digital components are also easier to program, which is essential for development. A mixed analog/digital system is Neurogrid, an analog chip that has an architecture very close to Mead's definition [26]. It works in subthreshold mode. Another hybrid is the Tianjic chip architecture, which performs computing with both analog and digital circuitry [33]. A third mixed analog/digital system is BrainScaleS, which is the system of focus for this project. It has a wafer-scale analog architecture, which was also proposed by Mead [26] and works in a supertreshold mode because of the high rate of operation. [38]

4.2 Evolutionary Hardware

Developers of electronic systems need to optimize for both robustness and efficiency[26]. Evolutionary processes give rise to diversity at every level of biology and lead to systems with high functional redundancy, which enables elements that are structurally different to perform the same function under certain conditions. At the same time, they can have distinct functions in other conditions [40]. The high utilization of the characteristics of the elements in biological systems leads to high efficiency in terms of both energy and material, and this inspires researchers in the design of artificial systems [26]. Inspired by biology, Evolvable Hardware (EH) is an attempt to copy these aforementioned traits of biological systems by applying EAs to hardware design. See chapter 2 on page 7 for an explanation of EAs.

There are two main classes of EH, Extrinsic and Intrinsic EH. Extrinsic EH is the approach of evaluating the evolved electronic circuit through simulation rather than through actual building and testing. This approach might be an advantage in terms of costs and hardware design, but is limited by the simulation and will not fully utilize the specific device characteristics. Intrinsic EH, on the other hand, is characterized by the evaluation of configurations on programmable hardware. This approach might lead to high utilization of the actual device but is limited by a pre-built design, which might be costly to change. Therefore, Intrinsic EH is often configurable, which again might lead to results that are very complicated to understand from a human perspective. Common challenges with Extrinsic EH are that the solution can only be as good as the model of the simulation. The model of the EH can "overfit" with the simulation, resulting in a "reality gap." [19]

As mentioned, development of EH is commonly conducted on Field-Programmable devices, either digital or analog. Digital devices are typically Field-Programmable Gate Arrays (FPGAs). Analog devices are typically Field-Programmable Analog/Transistor Arrays (FPAAs/FPTAs).

While there are excellent synthesis tools for digital circuitry, analog electronics development is lacking these same kinds of tools. EH has therefore proven useful for designing analog circuits, which are required to create and process analog signals. The Heidelberg FPTA has been one out of several successful architectures in this field, and has been used to realize a wide range of applications; including analog filters, comparators, Digital to Analog Converters (DACs), Analog to Digital Converters (ADC), and Operational Amplifiers (Op-Amps)[41]. [40]

Since the peak of EH, over 20 years ago, the size and complexity of the problems solved by EH have not increased much, and the solutions seldom compete with traditional designs [19]. As mentioned earlier, Mead proposed neuromorphic systems in 1990 in the EH peak [26]. Furthermore, a natural way to realize such systems are through configurable circuits. A specialized branch of EH that spawned in the early 2000s was the Networks-on-Chip (NoC) paradigm, which was, similarly to neuromorphic systems, a promising solution to the high-throughput and high-interconnect requirements of large-scale multi-processor systems, also called the "von Neumann bottleneck" [5][40][26]. Today, the NoC paradigm contain many successful neuromorphic architectures [40]. Most are developed by and for researchers, but some are oriented towards commercial applications; Qualcomm, Intel and IBM are all developing their own neuromorphic systems [27][12][13][40].

High Input Count Analog Neural Network (HICANN) In this project, the analog NoC ASIC called HICANN is of particular interest[35] because of its analog implementation of neurons and synapses. The HICANN is a full custom featuring configurable neural network arrays. The neuron model implemented is based on a spiking neural network model and is realized using Op-Amps and capacitors. While using analog circuitry to model neurons and synapses, the HICANN uses a digital, asynchronous bus interconnect, both on-chip and for external connections, featuring DACs and decoders. Manufactured in a 180 nm CMOS technology, the HICANN features 114,688 programmable dynamic synapses and up to 512 neurons [44]. [19][40] The HICANN is an essential component of the Physical Neuromorphic System of the Human Brain Project (HBP) [25], also known as BrainScaleS [27], see section 4.3.

4.3 BrainScaleS

The Human Brain Project, a European Flagship, is developing a European research infrastructure advancing brain research, medicine, and brain-inspired information technology for both industry and science. The project is now entering its third phase, with over 100 partnering institutions from over 20 countries in Europe, as well as over 100 partering projects. There are 12 subprojects in HBP, that span the development of six ICT-based Platforms. One of these six platforms is the Neuromorphic Computing (NMC) Platform, with two systems; the mixed-signal VLSI BrainScaleS

(Brain-inspired multiScale computation in neuromorphic hybrid systemS) and the massively parallel digital SpiNNaker (Spiking Neural Network architecture).[4]. As mentioned in section 4.2 on page 14 the BrainScaleS system is based on physical, analog, models of neurons and synapses, while the programmable interconnections are digital [44] The system is has an accelerated speed-up factor of 1000-10000 compared to biology, as relevant time constants are scaled down, with the goal of emulating various evolutionary mechanisms of the brain in one single set-up. The emphasis on the system is to use our knowledge of biology to model physical electronic neurons that neuroscientists can access and study at a much higher rate, due to our limited access to biological neurons. In addition to brain research, it is thought that this neuromorphic system will enable new applications in robotics, artificial intelligence and human-machine interfaces. Using a physical model keeps a one-to-one relationship between the neurons and synapses of the biological example and the model, preserving the fault tolerance concerning loss which is inherent in the biological brain. By using only a few transistors to emulate the neuron's differential equations compared to several millions involved in the same task while solving these equations numerically in a microprocessor core, the power consumption is reduced by several orders of magnitude. [36] As the neurons on the HICANN chip is emulated with analog electronics rather than with a high number of arithmetic operations, the circuitry is power efficient, and a complete HICANN consumes only $1.3W/cm^2$. The wafer module as a whole is designed for a worst-case power consumption of 2 kW. [44]

The accelerated network has extreme requirements in terms of communication bandwith, and this is met by employing wafer scale integration for the inter-chip connectivity[44]. The wafer scale integration is in accordance with the proposal of C. Mead, which proposed interconnecting chips with analog components by integrating the production wafer [26]. The base chips on the BrainScaleS wafer is the HICANN, see section 4.2 on page 15. A single 200 mm wafer carries 384 HICANN chips, and is mounted into a module which delivers power and handles data traffic using FPGAs. In addition to delivering signals to and from each wafer, the FPGAs are used to configure the chips. [44]. On one wafer module there is 48 FPGA module, each equipped with Gigabit-Ethernet, to handle the high amount of event data per time that will occur on the accelerated system. 12 Gigabit connections are routed to each edge of the module, respectively, to communicate with other wafer modules and the host computer. On the wafer, one FPGA controls 8 HICANN chips that together account for one reticle. Every HICANN has two full-duplex serial LVDS links with separate clock and data lines to the FPGA module, and each link is capable of transmitting two GBit/s. The input count per neuron can be increased to 14336 synapses per neuron, by combining up to 64 adjacent neurons. This leads to a transmission requirement of 1.4 GEvents/s per neuron, which is why the silicon wafer is kept as a whole, to produce shorter transmission lines and a lower capacitive load. The current BrainScaleS implementation,c at the Kirchhoff-Institute for Physics at Heidelberg University, enables up to

20 wafer modules, with up to 200 000 neurons and 40 000 000 synapses, per wafer. [44]

The Neuron Model at the basis of the HICANN is called the Adaptive Exponential Integrate-and-Fire model (AdExp) [7]. The AdExp model was co-developed by the FACETS project [36], a pre-decessor to the BranScaleS project, which is now part of the NMC platform of HBP. The model contains several additions compared to the standard Integrate-and-Fire model (IAF):

$$-C_m\frac{dV}{dt} = g_l(V-E_l) - g_l\Delta_{th}exp(\frac{V-V_{th}}{\Delta_{th}}) + g_e(t)(V-E_e) + g_l(t)(V-E_i) + w(t) \tag{4.1}$$

The variables $C_m, g_l E_l, E_e$ and E_i are the membrane capacity, the leakage conducatance and the leakage, exciatory and inhibitory reversal potentials. The variables $g_e(t)$ and $g_i(t)$ represent the total excitatory and inhibitory synaptic conductances. The addition to the standard IAF model is introduced as the *exponential* term on the right hand side of the equation, which models the near-asymptotic growth of the membrane potential under certain conditions. The *treshold potential* V_{th} represents the critical value above which this rapid growth can occur, and the *slope factor* Δ_{th} determines the rapidness of the triggered growth. Such a situation is interpreted as a spike, and each time a spike is detected, a separately generated output event signal is transmitted to possible connected target neurons or recording devices, and the membrane potential is forced to a reset potential V_{reset} by an adjustable reset conductance. A second equation describes the temporal evolution of the socalled *adaption current* $w(t)$:

$$-\tau_w\frac{dw}{dt} = w(t) - a(V-E_l) \tag{4.2}$$

Every time a spike is emitted by the neuron, w changes its value: $w \rightarrow w + b$. The time constant and the efficacy of the so-called *sub-treshold* adaption mechanism are given by τ_w and a, while b defines the amount of so-called *spike-triggered* adaption. The exponential term of equation 4.1 and the adaption function of equation 4.2 can be deactivated to reduce the AdExp model to the standard IAF model. [36][7] In the currently operative implementation of BrainScaleS, BSS-1, the neurons are impemented in a 180 mm CMOS technology.

fig. 4.3 shows the individual circuit components and fig. 4.3 illustrates the firing modes of this neuron circuit.

The system is designed to be scalable with newer generations of CMOS technology and HICANN chips [44].

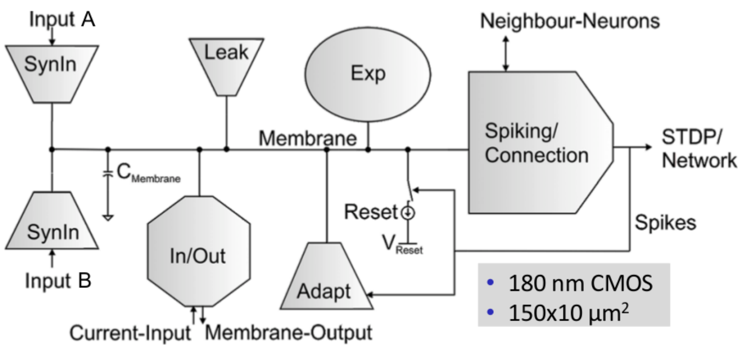

Figure 4.1: Schematic diagram of the AdExp neuron circuit, taken from [36].

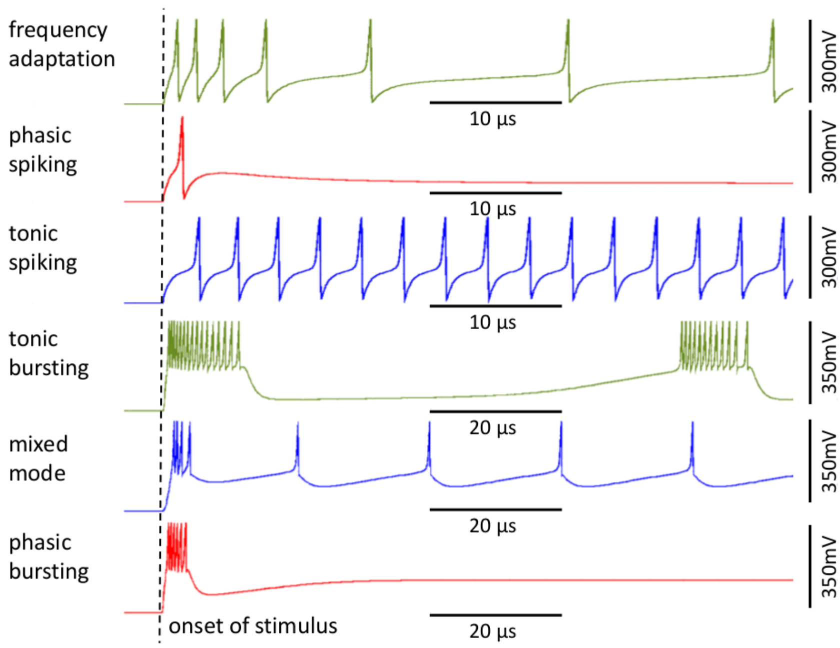

Figure 4.2: Example of firing modes of the AdExp neuron circuit, taken from [36].

Chapter 5

Autonomous Agents, Animats

The humanmade, intelligent autonomous agent would be different from a program in terms of it being able to operate like an independent, thinking entity. In contrast to a program that follows specific logical rules, the only constraints of an autonomous agent would be the mechanisms imposed on it by its architecture and environment, which is analogous to our biological constraints. If the agent had real intelligence, it should be able to adapt its perception, function, and logic to match its experiences of a changing environment and expand and explore its cognitive abilities [16]. As Alan Turing points out, and as we have discussed in **??** on page ??, this would not be possible with discrete-state logic-gate systems. However, Turing predicts that during the beginning of the current millennia, a discrete-state machine would be able to fool a human for a limited amount of time, which is indeed the case today [42]. The progress made in computational cognitive science is genuinely astonishing, where competence oriented agents are modeling advanced human abilities. The agents exhibit impressive performance, but they often lack in terms of general intelligence [43]. The question remains whether the performance of artificially intelligent agents will continue to increase if we only continue to develop competence oriented agents. Further exploration of alternative methods may be appropriate. One such method is the artificial animal, the "animat," explored in detail by Stewart W. Wilson in 1985, and first proposed as the "child machine" by Alan Turing in 1950 [43] [42].

5.1 Animat

The basics of the animat approach are to work upwards towards higher levels of intelligence. Essentially, there is a focus on the whole of the system, contrary to competence oriented modeling, which focuses on specific functions. The requirements are an environment that modeled after the problem side and an animat modeled after the solution side. The animat needs to have a sufficient sensor/motor system to satisfy its own needs. It is put through an increasingly challenging environment, and the goal is to the minimum increase in animat complexity necessary to satisfy its own needs in the given environment.

Wilson claims that the animat approach might be vital in developing machine perception. That is, developing an agent with abilities to develop its understanding of the environment [43]. For an autonomous agent to have a subjective experience, one could argue that it is necessary to have perception. Combined with hardware and neural networks that are non-discrete, the animat approach might be the technique required to develop such machine capabilities as can be coined truly intelligent or conscious.

Chapter 6

Integrated Information Theory

The Integrated Information Theory (IIT) has been proposed as a mathematical way to understand consciousness. Essentially, it revolves around analysing the phenomenology of a system. That is, how a causal structure experiences an environment. The theory tries to assume properties of consciousness (axioms), and from there it posulates the physical properties that must be necessary in a neural substrate hosting the system. The causal structure is based on measures of how a system is able to integrate information, and the tools provided is the specification of a cause-effect structure (CES). The amount of information is larger the higher the amount of possible states in the system. Integrated information is when there is no way to cut without loosing information.The CES is an unfolding of a model of the neural substrate, in the sense that the model is cut in every possible way (imagine cutting a network of nodes). The CES is mapped to the properties of experience, and the CES can be quantified by Φ. [32]

Although IIT might seem overly abstract, the first tangible result was published as a study on the question "Why does space feel the way it does?". Here a model of the Visual Cortex 1 and 2 (V1 and V2) was cut into a CES, using the knowledge of the grid cells that have been proven to be important for localization [21].

The IIT is a central part of the related works of this essay, and thus, some parts need to be explained:

6.1 Causal Structures

6.2 Conceptual Structures

In this project, I hope to utilize IIT as a way to analyze the perception of automated agents, evolved as SNN animats on BrainScaleS. (See chapter 5 on page 19, chapter 3 on page 9, chapter 4 on page 11). As the complexity and states of the animats is known, the animats can be analyzed. This is a challenging task, which will be attempted near the end of the project.

Chapter 7

Related Works

7.1 Albantakis et al., Evolution of Integrated Causal Structures in Animats Exposed to Environments of Increasing Complexity, 2014

The study uses the animat approach, see sect. chapter 5 on page 19, to evolve autonomous agents on a traditional, discrete-state, computer. The animats are evolved using a modification of the Simple Genetic Algorithm, see sect. chapter 2 on page 7. As is required in the animat approach, the study varies the complexity of the task environments. Through analyzis by use of the IIT, the researchers show that "in complex environments with a premium on context-sensitivity and memory, integrated brain architectures have an evolutionary advantage over modular ones."[2]

The study finds that during evolution, some animats may develop a brain that constist of one main complex, while other animats have brains that are constituted of segregated modules with feed-forward architecture. The study assumes that the evolution of animats will follow the same patterns as natural evolution, where characteristics of economy, functions, adaptiblity and robustness are favorable to simplicity of design. (Cite some studies in biology here...)

Each animats brain consist of 8 deterministic logic-gate elements. These are 2 sensors, 4 hidden elements and 2 motors (left, right). Each animat has a genotype which specifies the architecture of the neural network, and the logic functions of each element in the network. The animats are evolved over 60 000 generations, stating with an initial population of 100 animats with no connections between the hidden elements. For mutation, 100 animats are selected using roulette wheel selection, see sect. chapter 2 on page 7. The reproduction of animats happens without crossover, and up to three different mutation mechanics can happen, at a probabilistic rate.[3]

The three different mechanics are: Point mutations, by a probability of $p = 0.5\%$ per loci, with uniform integer replacement. Deletion, by $p = 2\%$ per genotype, a sequence between 16 and 512 adjacent loci is deleted. Duplication, by $p = 5\%$ per genotype, a sequence between 16 and 512 adjacent loci is duplicated and inserted at a random location within the genotype. The genotype is always between 1000 and 20 000 loci. Sequences of 10 loci encode one set of connections.[3]

The environment is a "falling blocks game", with a varying shape and size of the blocks. Additionaly, the task for each type of block varies between catching and avoiding of the block.

Most importantly, the work of Albantakis et al. applies the IIT and makes three predictions:

- The number of concepts, and their $\phi^m ax$ (see sect. **??** on page ??), should incrase during adaption, proportional to the amount of internal computation necessary to solve a task.
- Given a limited number of hidden elements, integration should also increase during adaption, particularly in tasks that require more memory.
- When the animat has a limited sensory and motor capacity, it needs to rely more on memory when the complexity of the environment increases. Therefore the integration of the animat should also increase during evolution, when under sensory or motor limiations.[3]

The study does find that during evolution, complex environments tend to lead to an increased number of concepts (functions) in the causal structure, and also more integrated conceptual structures (see sect. **??** on page ??). Animats with more integrated structures also seem to be more robust, which make sense because when there is a limited amount of elements in the brain, more integrated structures can contain more functions. The study also found that animats with a higher fitness usually had a higher degeneracy, that is, a function could be performed by several structures in the brain. This leads to flexibility and robustness, and could also lead to development of higher-order functions.[3]

What does the study say of the IIT, and how can it be used? According to IIt, integrated conceptual structures underlie consciousness [39]. This study found that animats with a higher number of integrated conceptrual structures found an advantage in complex environments, which may indicate that the IIT can be a useful tool in the quest for intelligent machines.

Chapter 8

Conclusion

Bibliography

[1] Syed Ahmed Aamir et al. "An Accelerated LIF Neuronal Network Array for a Large-Scale Mixed-Signal Neuromorphic Architecture". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.12 (Dec. 2018), pp. 4299–4312. ISSN: 1549-8328, 1558-0806. DOI: 10.1109/TCSI.2018.2840718. URL: https://ieeexplore.ieee.org/document/8398542/ (visited on 08/29/2019).

[2] Larissa Albantakis et al. "Evolution of Integrated Causal Structures in Animats Exposed to Environments of Increasing Complexity". In: *PLoS Computational Biology* 10.12 (Dec. 18, 2014). Ed. by Daniel Polani, e1003966. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003966. URL: https://dx.plos.org/10.1371/journal.pcbi.1003966 (visited on 01/09/2020).

[3] Larissa Albantakis et al. "Evolution of Integrated Causal Structures in Animats Exposed to Environments of Increasing Complexity". In: *PLoS Computational Biology* 10.12 (Dec. 18, 2014). Ed. by Daniel Polani, e1003966. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003966. URL: https://dx.plos.org/10.1371/journal.pcbi.1003966 (visited on 04/04/2019).

[4] Katrin Amunts et al. "The Human Brain Project: Creating a European Research Infrastructure to Decode the Human Brain". In: *Neuron* 92.3 (Nov. 2, 2016), pp. 574–581. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2016.10.046. URL: http://www.sciencedirect.com/science/article/pii/S0896627316307966 (visited on 02/12/2020).

[5] L. Benini and G. De Micheli. "Networks on chips: a new SoC paradigm". In: *Computer* 35.1 (Jan. 2002), pp. 70–78. ISSN: 1558-0814. DOI: 10.1109/2.976921.

[6] George Boole. *An Investigation of the Laws of Thought: On which are Founded the Mathematical Theories of Logic and Probabilities*. Google-Books-ID: pn_HORLo1uIC. Dover Publications, 1854. 476 pp.

[7] Romain Brette and Wulfram Gerstner. "Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity". In: *Journal of Neurophysiology* 94.5 (Nov. 1, 2005), pp. 3637–3642. ISSN: 0022-3077. DOI: 10.1152/jn.00686.2005. URL: https://journals.physiology.org/doi/full/10.1152/jn.00686.2005 (visited on 02/13/2020).

[8] Richard E. Brown and Peter M. Milner. "The legacy of Donald O. Hebb: more than the Hebb Synapse". In: *Nature Reviews Neuroscience* 4.12 (Dec. 2003), pp. 1013–1019. ISSN: 1471-0048. DOI: 10.1038/nrn1257. URL: https://www.nature.com/articles/nrn1257 (visited on 01/14/2020).

[9] A. N. Burkitt. "A Review of the Integrate-and-fire Neuron Model: I. Homogeneous Synaptic Input". In: *Biological Cybernetics* 95.1 (July 1, 2006), pp. 1–19. ISSN: 1432-0770. DOI: 10.1007/s00422-006-0068-6. URL: https://doi.org/10.1007/s00422-006-0068-6 (visited on 01/13/2020).

[10] Yongqiang Cao, Yang Chen, and Deepak Khosla. "Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition". In: *International Journal of Computer Vision* 113.1 (May 1, 2015), pp. 54–66. ISSN: 1573-1405. DOI: 10.1007/s11263-014-0788-3. URL: https://doi.org/10.1007/s11263-014-0788-3 (visited on 01/13/2020).

[11] Jack D. Cowan. "Discussion: McCulloch-Pitts and related neural nets from 1943 to 1989". In: *Bulletin of Mathematical Biology* 52.1 (Jan. 1, 1990), pp. 73–97. ISSN: 1522-9602. DOI: 10.1007/BF02459569. URL: https://doi.org/10.1007/BF02459569 (visited on 01/13/2020).

[12] Mike Davies et al. "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning". In: *IEEE Micro* 38.1 (Jan. 2018), pp. 82–99. ISSN: 1937-4143. DOI: 10.1109/MM.2018.112130359.

[13] Michael V. DeBole et al. "TrueNorth: Accelerating From Zero to 64 Million Neurons in 10 Years". In: *Computer* 52.5 (May 2019), pp. 20–29. ISSN: 1558-0814. DOI: 10.1109/MC.2019.2903009.

[14] Stanislas Dehaene et al. "Arithmetic and the brain". In: *Current Opinion in Neurobiology* 14.2 (Apr. 1, 2004), pp. 218–224. ISSN: 0959-4388. DOI: 10.1016/j.conb.2004.03.008. URL: http://www.sciencedirect.com/science/article/pii/S0959438804000406 (visited on 01/29/2020).

[15] Agoston E. Eiben and James E. Smith. *Introduction to evolutionary computing*. 2. ed. Natural computing series. OCLC: 934627991. Heidelberg: Springer, 2015. 287 pp. ISBN: 978-3-662-44873-1 978-3-662-44874-8.

[16] Stan Franklin and Art Graesser. "Is It an agent, or just a program?: A taxonomy for autonomous agents". In: *Intelligent Agents III Agent Theories, Architectures, and Languages*. Ed. by Jörg P. Müller, Michael J. Wooldridge, and Nicholas R. Jennings. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1997, pp. 21–35. ISBN: 978-3-540-68057-4. DOI: 10.1007/BFb0013570.

[17] Steve Furber. "Large-scale neuromorphic computing systems". In: *Journal of Neural Engineering* 13.5 (Oct. 1, 2016), p. 051001. ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2560/13/5/051001. URL: http://stacks.iop.org/1741-2552/13/i=5/a=051001?key=crossref.c170009b6af4be582bc1fbfbf3ccad04 (visited on 08/21/2019).

[18] Samanwoy Ghosh-Dastidar and Hojjat Adeli. "Improved spiking neural networks for EEG classification and epilepsy and seizure detection". In: *Integrated Computer-Aided Engineering* 14.3 (May 13, 2007), pp. 187–212. ISSN: 18758835, 10692509. DOI: 10.3233/ICA-2007-14301. URL: https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/ICA-2007-14301 (visited on 01/09/2020).

[19] Pauline C. Haddow and Andy M. Tyrrell. "Challenges of evolvable hardware: past, present and the path to a promising future". In: *Genetic Programming and Evolvable Machines* 12.3 (Sept. 1, 2011), pp. 183–215. ISSN: 1573-7632. DOI: 10.1007/s10710-011-9141-6. URL: https://doi.org/10.1007/s10710-011-9141-6 (visited on 01/30/2020).

[20] Michael Haenlein and Andreas Kaplan. "A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence". In: *California Management Review* 61.4 (Aug. 1, 2019), pp. 5–14. ISSN: 0008-1256. DOI: 10.1177/0008125619864925. URL: https://doi.org/10.1177/0008125619864925 (visited on 01/13/2020).

[21] Andrew Haun and Giulio Tononi. "Why Does Space Feel the Way it Does? Towards a Principled Account of Spatial Experience". In: *Entropy* 21.12 (Dec. 2019), p. 1160. DOI: 10.3390/e21121160. URL: https://www.mdpi.com/1099-4300/21/12/1160 (visited on 01/12/2020).

[22] J. Langeheine et al. "A CMOS FPTA chip for intrinsic hardware evolution of analog electronic circuits". In: *Proceedings Third NASA/DoD Workshop on Evolvable Hardware. EH-2001*. Proceedings Third NASA/DoD Workshop on Evolvable Hardware. EH-2001. ISSN: null. July 2001, pp. 172–175. DOI: 10.1109/EH.2001.937959.

[23] Yann LeCun et al. *Gradient-Based Learning Applied to Document Recognition*. Nov. 1998. URL: http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf (visited on 11/22/2019).

[24] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. "Training Deep Spiking Neural Networks Using Backpropagation". In: *Frontiers in Neuroscience* 10 (2016). ISSN: 1662-453X. DOI: 10.3389/fnins.2016.00508. URL: https://www.frontiersin.org/articles/10.3389/fnins.2016.00508/full (visited on 01/13/2020).

[25] Henry Markram et al. "Introducing the Human Brain Project". In: *Procedia Computer Science*. Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11) 7 (Jan. 1, 2011), pp. 39–42. ISSN: 1877-0509. DOI: 10.1016/j.procs.2011.12.015. URL: http://www.sciencedirect.com/science/article/pii/S1877050911006806 (visited on 01/26/2020).

[26] C. Mead. "Neuromorphic electronic systems". In: *Proceedings of the IEEE* 78.10 (Oct. 1990), pp. 1629–1636. ISSN: 1558-2256. DOI: 10.1109/5.58356.

[27] Karlheinz Meier. “A mixed-signal universal neuromorphic computing system”. In: *2015 IEEE International Electron Devices Meeting (IEDM)*. 2015 IEEE International Electron Devices Meeting (IEDM). ISSN: 2156-017X. Dec. 2015, pp. 4.6.1–4.6.4. DOI: 10.1109/IEDM.2015.7409627.

[28] Karlheinz Meier. “Special report : Can we copy the brain? - The brain as computer”. In: *IEEE Spectrum* 54.6 (June 2017), pp. 28–33. ISSN: 1939-9340. DOI: 10.1109/MSPEC.2017.7934228.

[29] Marvin Minsky and Seymour A. Papert. *Perceptrons: An Introduction to Computational Geometry*. Google-Books-ID: PLQ5DwAAQBAJ. MIT Press, Sept. 22, 2017. 317 pp. ISBN: 978-0-262-53477-2.

[30] G.E. Moore. “Cramming More Components Onto Integrated Circuits”. In: *Proceedings of the IEEE* 86.1 (Jan. 1998), pp. 82–85. ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.1998.658762. URL: http://ieeexplore.ieee.org/document/658762/ (visited on 01/30/2020).

[31] J. von Neumann. “First draft of a report on the EDVAC”. In: *IEEE Annals of the History of Computing* 15.4 (1993), pp. 27–75. ISSN: 1934-1547. DOI: 10.1109/85.238389.

[32] Masafumi Oizumi, Larissa Albantakis, and Giulio Tononi. “From the Phenomenology to the Mechanisms of Consciousness: Integrated Information Theory 3.0”. In: *PLoS Computational Biology* 10.5 (May 8, 2014). Ed. by Olaf Sporns, e1003588. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003588. URL: http://dx.plos.org/10.1371/journal.pcbi.1003588 (visited on 01/09/2020).

[33] Jing Pei et al. “Towards artificial general intelligence with hybrid Tianjic chip architecture”. In: *Nature* 572.7767 (Aug. 2019), pp. 106–111. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1424-8. URL: https://www.nature.com/articles/s41586-019-1424-8 (visited on 12/05/2019).

[34] F. Rosenblatt. “The perceptron: A probabilistic model for information storage and organization in the brain”. In: *Psychological Review* 65.6 (1958), pp. 386–408. ISSN: 1939-1471(Electronic),0033-295X(Print). DOI: 10.1037/h0042519.

[35] Johannes Schemmel, Johannes Fieres, and Karlheinz Meier. “Wafer-scale integration of analog neural networks”. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). ISSN: 2161-4407. June 2008, pp. 431–438. DOI: 10.1109/IJCNN.2008.4633828.

[36] Johannes Schemmel et al. “A wafer-scale neuromorphic hardware system for large-scale neural modeling”. In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. 2010 IEEE International Symposium on Circuits and Systems - ISCAS 2010. Paris, France: IEEE, May 2010, pp. 1947–1950. ISBN: 978-1-4244-5308-

5. DOI: 10.1109/ISCAS.2010.5536970. URL: http://ieeexplore.ieee.org/document/5536970/ (visited on 08/29/2019).

[37] J. Schemmel et al. "Implementing Synaptic Plasticity in a VLSI Spiking Neural Network Model". In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. The 2006 IEEE International Joint Conference on Neural Network Proceedings. ISSN: 2161-4407. July 2006, pp. 1–6. DOI: 10.1109/IJCNN.2006.246651.

[38] Catherine D. Schuman et al. "A Survey of Neuromorphic Computing and Neural Networks in Hardware". In: *arXiv:1705.06963 [cs]* (May 19, 2017). arXiv: 1705.06963. URL: http://arxiv.org/abs/1705.06963 (visited on 01/13/2020).

[39] Giulio Tononi et al. "Integrated information theory: from consciousness to its physical substrate". In: *Nature Reviews Neuroscience* 17.7 (July 2016), pp. 450–461. ISSN: 1471-003X, 1471-0048. DOI: 10.1038/nrn.2016.44. URL: http://www.nature.com/articles/nrn.2016.44 (visited on 01/09/2020).

[40] Martin Albrecht Trefzer and Andrew M. Tyrrell. *Evolvable hardware: from practice to application*. Natural computing series. Berlin Heidelberg: Springer, 2015. 411 pp. ISBN: 978-3-662-44615-7 978-3-662-44616-4.

[41] Martin Trefzer et al. "Operational Amplifiers: An Example for Multi-objective Optimization on an Analog Evolvable Hardware Platform". In: *Evolvable Systems: From Biology to Hardware*. Ed. by J. Manuel Moreno, Jordi Madrenas, and Jordi Cosp. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 86–97. ISBN: 978-3-540-28737-7. DOI: 10.1007/11549703_9.

[42] Alan M. Turing. "Computing Machinery and Intelligence". In: *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Ed. by Robert Epstein, Gary Roberts, and Grace Beber. Dordrecht: Springer Netherlands, 2009, pp. 23–65. ISBN: 978-1-4020-6710-5. DOI: 10.1007/978-1-4020-6710-5_3. URL: https://doi.org/10.1007/978-1-4020-6710-5_3 (visited on 01/12/2020).

[43] S. W. Wilson. *The Animat Path to AI*. 1991.

[44] K. Zoschke et al. "Full wafer redistribution and wafer embedding as key technologies for a multi-scale neuromorphic hardware cluster". In: *2017 IEEE 19th Electronics Packaging Technology Conference (EPTC)*. 2017 IEEE 19th Electronics Packaging Technology Conference (EPTC). Dec. 2017, pp. 1–8. DOI: 10.1109/EPTC.2017.8277579.