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

Daniel Sander Isaksen

19th 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 [22]. 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 [51]. 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 [40] and refined by Minsky et al. [35], 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 [37], theoreticians that were highly supportive of "learning" machines. ANNs, especially with the method of Deep Learning, pioneered by LeCun et al. [25], are now the most used tool for developing modern artificial intelligence technologies [22]. 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. The complexity of the functions approximated by computation is growing exponentially, and so are the energy- and material costs of the computing systems, and thus they are not sustainable.

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 [35], 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 [19][44].

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 [22]. 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 [48]. 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 this essay uses the term.** 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 architecture 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.

The initial motivation behind this project was the exploration of a neuromorphic system that might lead to new applications in robotics, artificial intelligence, or brain-computer interface technology. When the cost of simulation of the nervous system, which includes solving a massive amount of differential equations, can be reduced by several orders of magnitude, then a whole new world of possibilities could open: Efficient modeling of differential equations will allow even the simplest robots to have advanced control loops. Maybe inherent characteristics in the neuromorphic system could be utilized to process complex signals, like EEG signals, with ease. Ultimately, maybe something equivalent to human consciousness could be stored in such a system that can emulate millions of neurons at a fraction of the cost of an ordinary computer. Nevertheless, these dreams are all hidden away in the far future. For now, the question remains what methods are most useful when it comes to training the physical neural network that the neuromorphic system, BrainScaleS, emulates.

Chapter 2

Background theory

2.1 Evolutionary Algorithms

Nature is exceptional at inventing new complex, resilient solutions through the evolution of organizations, from the molecular level to the ecosystems. This project uses biologically inspired Evolutionary Algorithms (EAs) to develop algorithms to run on hardware. See section 2.3.2 on page 14 for more about Evolvable Hardware. In biology, Deoxyribonucleic acid (DNA) governs the processes and structure of the cells of living cellular organisms). The minimal units of DNA are called **genes**, and they are structured in **chromosomes**. They determine the physical characteristics and development of the organism. In EAs, the chromosomes may take the form of data structures like strings, arrays, or graphs where each segment of the data structure is a gene. 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 influential factor for the phenotype, other factors also play an important role, like environmental factors. [49] 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 in general are 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. A **gentotype** in EAs is each candidate solution, with its genes being the parameters of the solution. A **phenotype** is the resulting characteristic of a particular 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, 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 prematurely in the first available local optima. After the first selection, individuals are recombined, and their offspring mutated. In optimization algorithms like EAs, there needs to be a balance between how much the best solutions are **exploited**, and how much the solution space is **explored** for new solutions. The evolution typically runs until certain criteria are met or until a set number of generations has been reached. [16] The SGA has inspired mind-boggling variations of algorithms under the relatively wide EA umbrella, and examples of these are the Island Model [46] and Evolutionary Optimization [45].

2.2 Spiking Neural Networks

Spiking Neural Networks (SNNs), often called The Third Generation of Artificial Neural Networks (ANNs), were proposed by the mathematician Wolfgang Maas [28]. The design of SNNs bases itself on modern neuroscience, with advanced neuron models based on what we know of synaptic connections between neurons and what stimulates the membrane potential of the cells [20]. Biological neurons are inherently different from networks of simple logic gates like The Multilayer Perceptron, and modeling them could 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 [26] [10].

2.3 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.

Consider the history of our computer architecture: The philosopher George Boole's maybe most excellent work was his 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 and led to the vacuum tube technology used in early computers. John von Neumann draws parallels to the human neurons in his report on the EDVAC, which was used by Alan Turing as a theoretical source of understanding while designing the Automatic Computing Engine [37]. The theoretical attempt to "capture human intelligence" and mechanical analogies to neurons led to our computer architecture. Combinatorial logic eventually led to the development of modern Information Theory, which was proposed by the same Shannon in his A Mathematical Theory of Communication. Thus, the foundational basics of our computer programs are also rooted in the search for intelligent systems. The most successful artificial intelligence programs were, for a long time, only compositions of logical if-then sentences [22]. Today most artificial intelligence programs are based on arithmetic approximation of complex functions through the use of millions of logic gates in the form of transistors. Over the past 70 years, our understanding of the nervous system and human psychology has developed drastically. However, modern computing is still rooted in the same "outdated" knowledge of the human brain.

The efficiency of our brains outperforms modern supercomputers in several aspects, like pattern recognition, grasping the "essence" of books, and creative thinking. In other aspects, like arithmetics and approximation, the supercomputer is far more efficient. 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 [33]. The brain has a small, underdeveloped center for arithmetics, while the core of the computer only performs arithmetics [15]. 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 chapter 1 on page 5, 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 [31] [36]. 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 [31].

The challenges related to building a neuromorphic system are many; 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. [44].

2.3.1 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. [44]

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. [44]

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 [32][43]. [44]

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. [44]

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.” [44]

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 [19] [44]. 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 [31]. 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. [24]. For digital communication and storage of information, analog neuromorphic systems often utilize some digital components. A common term for a hybrid neuromorphic system is **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 [31]. It works in subthreshold mode. Another hybrid is the Tianjic chip architecture, which performs computing with both analog and digital circuitry [39]. 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 [31] and works in a supertreshold mode because of the high rate of operation. [44]

2.3.2 Evolvable Hardware

Developers of electronic systems need to optimize for both robustness and efficiency[31]. 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 [49]. 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 [31]. Inspired by biology, Evolvable Hardware (EH) is an attempt to copy these traits mentioned above of biological systems by applying EAs to hardware design. See section 2.1 on page 9 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." [21]

The standard hardware for the development of EH is 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)[50]. [49]

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 [21]. As mentioned earlier, Mead proposed neuromorphic systems in 1990 in the EH peak [31].

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][49][31]. Today, the NoC paradigm contain many successful neuromorphic architectures [49]. 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 [32][12][14][49].

High Input Count Analog Neural Network (HICANN)

In this project, the analog NoC ASIC called HICANN is of particular interest[41] 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 [53]. [21][49] The HICANN is an essential component of the Physical Neuromorphic System of the Human Brain Project (HBP) [29], also known as BrainScaleS [32], see section 2.3.3.

2.3.3 BrainScaleS

The Human Brain Project is a European Flagship that 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 partnering 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].

The main applicaiton of BrainScaleS is the study of physical neural dynamics at a higher rate than available in biology. Thus, BrainScaleS was designed with a speed-up factor for 1000 - 10 000 compared to biological wall time. Modeling of biological neurons and synapses are realized with physical, analog components, while the interconnections are digital and programmable to increase efficiency [53]. In addition to brain research, this neuromorphic system may 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 the loss, which is inherent in the biological brain.

The power- and material costs when simulating the neuron is reduced by several orders of magnitude by using only a few analog components per neuron, compared to several million involved in the same task while solving these equations numerically in a microprocessor core. [42] As the neurons on the HICANN chip are 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 theoretical worst-case power consumption of a wafer module is 2 kW. [53]

The current BrainScaleS implementation is 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. Wafer-Scale integration was chosen to allow for the extreme bandwidth requirements of the accelerated system [53]. The Wafer-Scale integration is partly following the proposal of C. Mead, who proposed interconnecting chips with analog components by integrating the production wafer [31]. The base chips on the BrainScaleS wafer are the HICANN, section 2.3.2 on page 15. A single 200 mm wafer carries 384 HICANN chips. Equipped to the wafer is FPGAs that handle communication with other wafers and with the host computer. In addition to delivering signals to and from each wafer, the FPGAs are used to configure the chips. [53]. On one wafer module, there are 48 FPGAs, 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. One fundamental concept with the HICANN is that it allows the construction of neurons with over 10 000 pre-synaptic connections. The high connectivity 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. [53]

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 [42], a predecessor 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):

$$-C_m\frac{dV}{dt} = g_l(V - E_l) - g_l\Delta_{th}exp(\frac{V - V_{th}}{\Delta_{th}}) + g_e(t)(V - E_e) + g_I(t)(V - E_i) + w(t) \tag{2.1}$$

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 introduced addition to the standard IAF model is 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* $w(t)$:

$$-\tau_w\frac{dw}{dt} = w(t) - a(V - E_l) \tag{2.2}$$

Every time a spike is emitted by the neuron, w changes its value: $w \to w + b$. 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 2.1 and the adaption function of equation 2.2 can be deactivated to reduce the AdExp model to the standard IAF model. [42][7]

fig. 2.3.3 shows the individual circuit components and fig. 2.3.3 illustrates the firing modes of this neuron circuit. In the currently effective implementation of BrainScaleS, BSS-1, the neurons are implemented in a 180 mm CMOS technology. By design, the system is scalable with newer generations of CMOS technology and HICANN chips [53]. Millner et al. [34] describes the hardware implementation of the neuron to great detail. The paper also report that the emulation of an IAF neuron with the implemented hardware is 3000 times more power-efficient and accelerated by a factor of 10 000 when compared to an Izhikevich neuron simulated on a supercomputer [34]. The Izhikevich neuron is a popular model because it gives an accurate representation of a neuron's functionality while being simple to model [44].

Communication

on a wafer happens on several levels; There is the communication happening between the local neurons and synapses, then there are Layer 1 (L1) and Layer 2 (L2) channels depending on how far the communication reaches. This subsection discusses the inter-neuron communication of the system in terms of how closely it models nature, which is

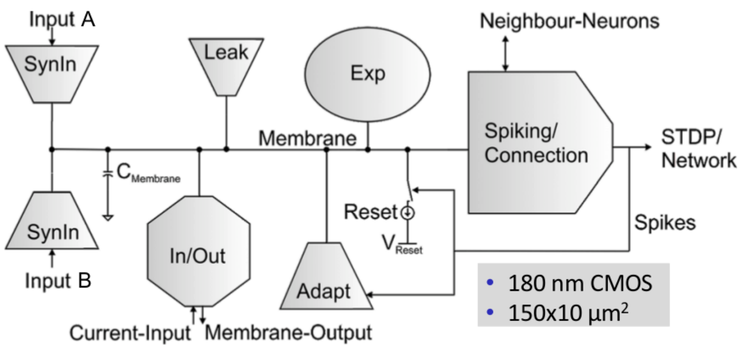

Figure 2.1: Schematic diagram of the AdExp neuron circuit, taken from [42].

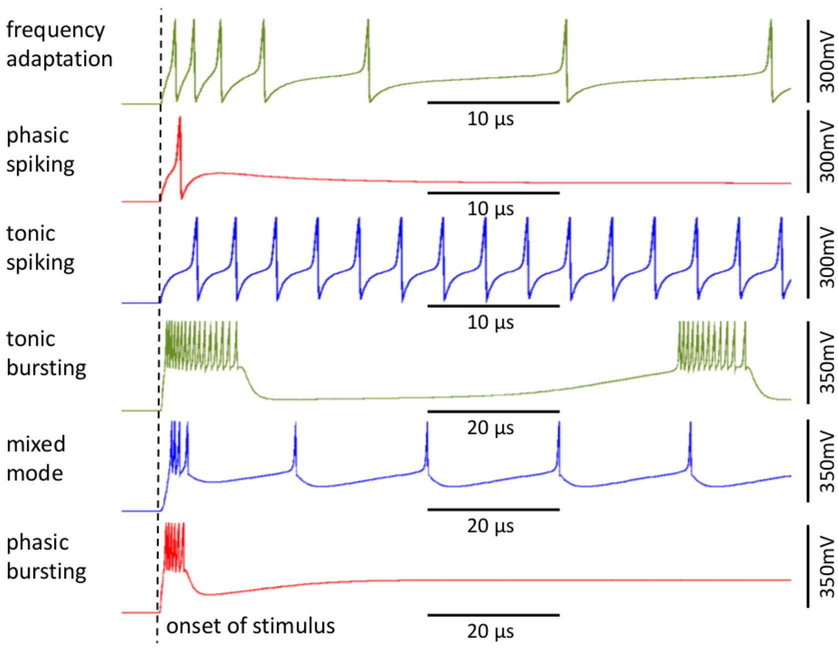

Figure 2.2: Example of firing modes of the AdExp neuron circuit, taken from [42].

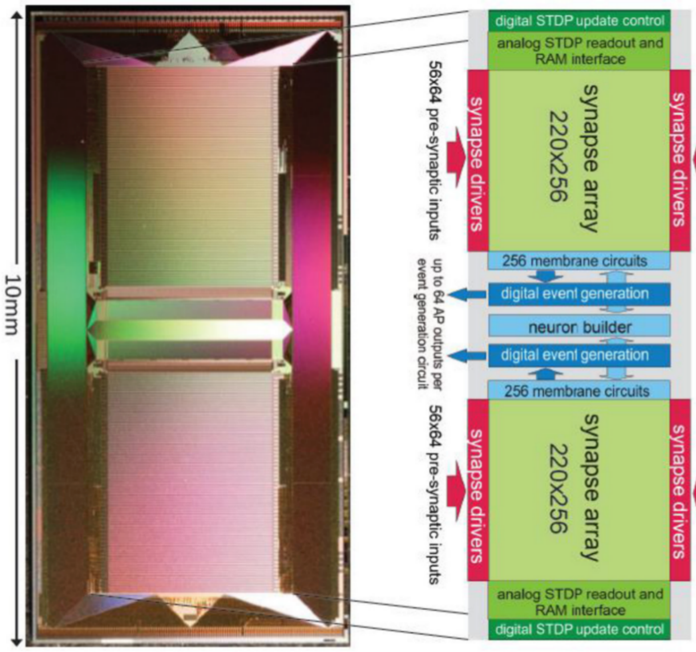

Figure 2.3: To the left is a single HICANN chip, and to the right is diagram of the ANC which is located at the center of the chip, taken from [47].

vital information for any comparison analysis - for example, involving Integrated Information Theory, section 2.5 on page 21.

The composing structure of the neurons, with their respective synapses, is called the Analog Network Core (ANC), shown in fig. 2.3.3. The neurons are constructed by Dendrite Membrane (DenMem) circuits that allow neurons with up to 14336 synaptic inputs per neuron. The synaptic weights, stored in a four-bit SRAM, are represented by a current generated by a DAC. Short-Term Depression (STD) and Spike-Timing Dependent Plasticity (STDP) are implemented by the use of capacitors modulating the signals and by an algorithm that manipulates the digitally stored weights. fig. 2.3.3 shows a schematic diagram of the synaptic circuit, where g_max is a programmable analog parameter controlling the scale of the DAC. In each ANC, communication is asynchronous and mixed-signal. The cores are highly integrated systems working in a continuous-time mode. [42]

The communication protocol between ANCs is L1, a real-time serial event protocol operating at up to 2 Gb/s. The protocol uses two time-frame bits and six address data bits and uses continuous-time transmission [41]. This digital transmission protocol limits power consumption while retaining continuous communication. [42]. Wafer-Scale integration was selected to support the channel density requirements of an accelerated system, where each neuron can receive over 10k pre-synaptic inputs [42]. The solution of the Wafer-Scale integration is explained in detail in [53] and

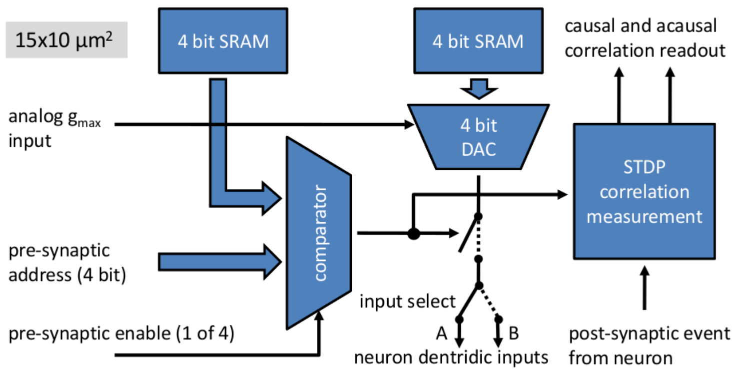

Figure 2.4: Schematic diagram of a synapse circuit, taken from [42].

[47]. The inter-ANC communication on a wafer is real-time, serial, and only dependent on the states of the local circuits involved.

Each HICANN chip has an L2 channel to allow for inter-wafer communication. Due to latency issues involved with real-time long-range communication, L2 is based on packets-switching and digitized time stamps [42] [47]. This digital communication in-between wafers, or with the host computer, is handled by FPGAs [53]. Therefore, inter-neuron communication between wafers has a lower level of integration.

2.4 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 [18]. As Alan Turing points out, and as we have discussed in chapter 1 on page 5, 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 [51]. 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 [52]. 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 [52] [51].

2.4.1 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. The environment of the animats is supposed to be 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 [52]. 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.

2.5 Integrated Information Theory

The Integrated Information Theory (IIT) has is proposed as a mathematical way to understand consciousness. It revolves around analyzing the phenomenology of a system, that is, how a system experiences the environment. IIT 3.0 mostly does this by finding the maximally irreducible conceptual structures formed by a causal structure interacting with the environment. The theory tries to assume properties of consciousness (axioms), and from there, it postulates the properties that must be necessary for a physical substrate hosting the system. The causal structure, also called a cause-effect structure (CES), is analyzed to find conceptual structures that measure how well a system integrates information. A more substantial amount of integrated information results in a higher amount of possible states in the system. Integrated information is when there is no way to cut the CES without losing 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 maps to the properties of experience, and the CES can be quantified by Φ [38].

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 are proven to be essential for localization [23].

A summary of corollaries of IIT, based on the article by Oizumi et. al [38]: An intelligent system will usually consist of a main **complex**, and smaller supporting complexes. This central complex will be the most conscious part of the system, like the part of our brain that we could not function without, while the supporting complexes could be compared to smaller parts of the brain, like, for example, the visual cortex. Intelligent systems will not be modular, that is, if the system produces less functionality than their components, and they have to produce functionality that is more than a higher order of the combined components. Inactive systems can be conscious in the way that a system may have significant parts that are ready to be activated or that are passively affecting the state of the system. A system can perform very complex functions but still not be conscious. In particular, feed-forward networks would not be conscious. An example if this is a microprocessor implementing a neural network that, in some cases, can recognize thousands of different objects faster than a human can recognize one object. IIT states that it is not only the functionality of a system that determines how conscious it is but also how the function performs internally. A network that can recognize objects based on internal states and previous experiences is more conscious compared to a feed-forward neural network, which only recognizes the object based on the external input it gets. Networks that are not necessarily feed-forward only, but simulated on a physical substrate that implements functionality based on numerical approximation, would not be termed conscious in IIT 3.0 [30]. A final corollary that summarises the ones above says that an intelligent system can develop concepts based on other concepts within itself without the need for external stimuli. These internal concepts would more often be connected to a large number of other concepts and not contribute to specific details.

In this project I hope to utilize IIT as a way to analyze the perception of automated agents, evolved as SNN section 2.2 on page 10 animats section 2.4 on page 20 on BrainScaleS section 2.3.3 on page 15. As the complexity and states of the animats are known, the animats can be analyzed. Analyzing the animats in regards to IIT will only be attempted if time permits.

Chapter 3

Related Works

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

The study uses the animat approach, section 2.4 on page 20, to evolve autonomous agents on a traditional, discrete-state computer. The authors use a Simple Genetic Algorithm to evolve the animats, section 2.1 on page 9. As is required in the animat approach, the study varies the complexity of the task environments. Through analysis 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 system that consists of one main conceptual complex. In contrast, other animats develop networks of segregated modules with feed-forward architecture. The study assumes that the evolution of animats will follow the same patterns as a natural evolution, where characteristics of economy, functions, adaptability, and robustness are favorable to the simplicity of design.

Each neural network "animat" consists of eight deterministic logic-gate elements. These are two sensors, four hidden elements, and two motors (left, right). For every animat, a genotype defines the network architecture and logic-functions for each node.

The animats are evolved over 60 000 generations, starting 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. section 2.1 on page 9. 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 varying shape and size of the blocks. Additionally, the task for each type of block varies between catching and avoiding 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. section 2.5 on page 21), should increase during adaption, proportional to the amount of internal computation necessary to solve a task.
- Given the 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 limitations.[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. section 2.5 on page 21). Animats with more integrated structures also seem to be more robust, which makes 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 higher fitness usually had a higher degeneracy. In biology, degeneracy means that structurally different elements can perform the same function and correlates with flexibility and robustness. The author states that this could also lead to the 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 [48]. This study found that animats with a higher number of integrated conceptual 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 4

Conclusion

This essay taps into a diverse set of research fields. The ambition is to scope the project down to a combination of this diverse knowledge that might bring a contribution to science. The most prominent topics are artificial intelligence, evolutionary computing, neuromorphic electronic systems, neurobiology, and consciousness theory. These topics are, of course, all highly relevant in a philosophical discussion of artificial intelligence, which has become the main tag for this essay.

One experiment , examined through the lenses of a couple of thinkers, would be the ideal fit for this project. Which perspectives to analyze the experiment through is yet to be decided, but this chapter will propose an initial experiment: To evolve SNN section 2.2 on page 10 animats section 2.4 on page 20 implemented on BrainScaleS, section 2.3.3 on page 15 to play a falling blocks game. The experiment would be an iterative replication of [3] on BrainScaleS. The experiment should provide a basis for comparison of systems, from implementation with logic gates to implementation with continuous variables to implementation on BrainScaleS. The comparisons can provide various answers based on the perspectives chosen.

4.0.1 Methods

in these experiments are chosen to provide a comparison of systems; Digital discrete, digital continuous, and mixed-signal continuous. The discrete digital implementation includes both the game and the animat in Python, with the evolutionary algorithm replicating the one used in [3]. The continuous digital step has two parts. The first would be to implement the animat as an SNN in a simulator that has a fitting AdExp model section 2.3.3 on page 16. This simulator would most likely be NEST [17], using PyNN [13]. The second part would be to implement the game in PyNN Finally, the accelerated timescale of BrainScaleS requires the mixed-signal continuous implementation of the experiment to have both the animat and the game in PyNN. Additionally, the neural network animats would need a design that allows evolution, which requires certain compliances to hardware constraints. The three different implementations of the same experiment will not have much value unless one tries to answer specific questions by the comparison. The idea is that specific questions can be answered when needed if the results and configuration are ready to be replicated.

After each of the first, second, and third steps, there should be data of results and configurations that can be analyzed and re-run to produce metrics according to the perspectives chosen for the theoretical contribution.

4.0.2 Perspectives

From the perspective of efficiency in neuromorphic systems , an animat approach could be interesting in terms of finding circuit designs that better utilize the characteristics of the architecture. The BrainScaleS architecture carries several of the elements of the design that Carver Mead proposed, which are analog components and wafer-scale integration. In theory, such a system should be efficient solving complex differential equations with only a few transistors, given the correct hardware configuration. The apparent problem of neuromorphic systems is that they are hard to train. Ordinary methods are inefficient and may not utilize the strengths of the neuromorphic system. As was proposed in [45], methods from evolutionary computing allow for topological and qualitative changes to the networks that might utilize the characteristics of the transistors in a way that could not have been designed by humans. Using noise as a resource [27], by Wolfgang Maass, is an excellent example of unique transistor characteristics that could not have been done by design. Comparing the various implementations that are in a simulation and on neuromorphic hardware, it might be possible to make a measure of efficiency.

From the perspective of applications of neuromorphic systems , an animat approach seems like an attractive solution because of how the approach evolves systems that work well in an environment even though none of the underlying conceptual structures solves tasks that would seem necessary to succeed. The opposite of the animat approach would be the competence oriented approach of training neural networks for particular tasks. The nodes of ordinary neural networks used in machine learning today will produce one specific output per specific input, which makes competence oriented training a natural solution of choice. The analog AdExp neurons section 2.3.3 on page 16 of BrainScaleS will produce different outputs based on the internal state of the specific node, which could be problematic noise in the competence oriented approach. However, this is a valuable resource for robustness and functional redundancy, which is the ability to perform the same function with different parts of the system. An example of functional abundance is that a network can recognize cats, but when one of the parts in the network dies, there is another part that also recognizes cats. Another example of possible achievements is when the nodes can produce different outputs is that the network could use fewer nodes to produce the same functionality or more. An example would be that developers could train an ordinary ANN to perform well at the recognition of cats. However, the developers would then be quite sure that the same ANN would not perform well at finding punctuation errors in a text. It would seem that this example is not as valid for networks on BrainScaleS, although it is likely that the various tasks that one network would be efficient at would not be as specific as the two tasks in the example above. Following is a thought experiment of how animats would develop an essential blend of competencies that is hard to design. Imagine a world-simulator that included sound, forces like gravity and friction, and objects of various shapes, functions, and colors. Imagine then populating the world with animats that could manipulate the world, but that needed to survive. The condition for survival could be that they needed to touch a certain number of red, round objects per day. The color of the objects changes when touched by an animat, so being an efficient hunter of red orbs would undoubtedly be a matter of importance. The specific competencies necessary for survival in this world is, of course, object recognition, color recognition, and efficient movement. However, there might be other skills and finesses that play a difference, but that is not as obvious to a designer. Following the reasoning above, the animat approach seems a viable method for finding applications for BrainScaleS. The animat approach is a method for letting the agent find out which skills are necessary to complete the main task. If the comparisons show that the animats have higher fitness on BrainScaleS, then this would probably help make suggestions as to how to make real-world applications with the BrainScaleS technology.

From the perspective of Integrated Information Theory , an experiment involving the evolution of networks on analog components can be useful for validating parts of the theory. There is an interesting corollary from IIT 3.0, saying that even though a neural network performs very complex functions, it could not be conscious if built in a simulation, section 2.5 on page 21. On an ordinary computer, functions that simulate neuronal activity are what represent the neurons of a network. Neural networks on BrainScaleS are physical, where only a few analog components provide the functionality of each neuron and synapse, and the synapses also implement adaptive learning through plasticity section 2.3.3 on page 15. As discussed in the same section, the on-wafer communication between neurons is continuous, highly connected, and dependent on the varying internal states of the components that carry the signals. IIT supporters would generally deny that artificial intelligence based on the technology used today could be conscious but do not outright the same claims for the BrainScaleS system. Additionally, BrainScaleS closely emulates biological neural networks, including the relationship between synapses and neurons when it comes to weakness or cell death. Biological evolution proves that systems that individuals that are more robust to failure and adaptive to changing environments will make it more often than their peers. Robustness and adaptiveness are traits that also follow systems considered highly integrated by IIT. Given that BrainScaleS has a speedup factor of 10 000 compared to biological wall time, there is a possibility that evolution also is a component that can make a theoretical contribution to IIT.

From the perspective of Artificial Intelligence discussion , the exploration of neuromorphic systems is interesting because the researchers intend their architectures to simulate the brain better. Based on the history recapitalization of chapter 1 on page 5 and section 2.3 on page 10, it may seem like many computing pioneers aimed at building a "thinking machine." Modern information theory and computer architecture partly came from trying to replicate the form and function of human intelligence and the brain. George Boole, the father of logic algebra, the mathematics that led to combinatorics and the transistor, was concerned with capturing human reasoning. John von Neumann loosely based his architecture, the most popular of today, on analogies to neurons and the brain. Alan Turing, who helped bring von Neumann architecture to the world, published works on artificial intelligence that still has a direct influence on modern researchers. Viewing history from this perspective, one may wonder if further development of computing itself requires that we continue to pursue the brain in terms of hardware. Carver Mead, a professor on transistors, and a close friend of Gordon Moore, whom the famous "Moore's law" stems from, was the one who coined neuromorphic electronic systems. Turing also proposed ways to develop and test artificial intelligence. One of those was the "Child-Machine," which later turned into the "Animat." The animat has not turned out to be an effective way of developing artificial intelligence on arithmetic computers, which is nothing else than approximations of very complex functions. There is no reason to believe that Turing, or anyone else from the past, could have found better answers to questions of today than the people asking the questions today. However, there is reason to believe that a system could develop intelligence like a child does develop intelligence, given the right physical and mechanical conditions, and maybe BrainScaleS has got just that.

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] Andrew P. Davison et al. "PyNN: a common interface for neuronal network simulators". In: *Frontiers in Neuroinformatics* 2 (2009). ISSN: 1662-5196. DOI: 10.3389/neuro.11.011.2008. URL: https://www.frontiersin.org/articles/10.3389/neuro.11.011.2008/full (visited on 02/16/2020).

[14] 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.

[15] 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).

[16] 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.

[17] Tanguy Fardet et al. *NEST 2.20.0*. Jan. 31, 2020. DOI: 10.5281/zenodo.3605514. URL: https://zenodo.org/record/3605514#.XkiiEOEo-kA (visited on 02/16/2020).

[18] 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.

[19] 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).

[20] 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).

[21] 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).

[22] 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).

[23] 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).

[24] 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.

[25] 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).

[26] 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).

[27] W. Maass. "Noise as a Resource for Computation and Learning in Networks of Spiking Neurons". In: *Proceedings of the IEEE* 102.5 (May 2014), pp. 860–880. ISSN: 0018-9219. DOI: 10.1109/JPROC.2014.2310593.

[28] Wolfgang Maass. "Networks of spiking neurons: The third generation of neural network models". In: *Neural Networks* 10.9 (Dec. 1, 1997), pp. 1659–1671. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(97)00011-7. URL: http://www.sciencedirect.com/science/article/pii/S0893608097000117 (visited on 02/11/2020).

[29] 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).

[30] William Marshall, Jaime Gomez-Ramirez, and Giulio Tononi. "Integrated Information and State Differentiation". In: *Frontiers in Psychology* 7 (2016). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2016.00926. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2016.00926/full (visited on 02/17/2020).

[31] 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.

[32] 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.

[33] 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.

[34] Sebastian Millner et al. "A VLSI Implementation of the Adaptive Exponential Integrate-and-Fire Neuron Model". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty et al. Curran Associates, Inc., 2010, pp. 1642–1650. URL: http://papers.nips.cc/paper/3995-a-vlsi-implementation-of-the-adaptive-exponential-integrate-and-fire-neuron-model.pdf (visited on 01/26/2020).

[35] 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.

[36] 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).

[37] 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.

[38] 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).

[39] 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).

[40] 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.

[41] 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.

[42] 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).

[43] 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.

[44] 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).

[45] Catherine D. Schuman et al. "An evolutionary optimization framework for neural networks and neuromorphic architectures". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. 2016 International Joint Conference on Neural Networks (IJCNN). ISSN: 2161-4407. July 2016, pp. 145–154. DOI: 10.1109/IJCNN.2016.7727192.

[46] Catherine D. Schuman et al. "Parallel Evolutionary Optimization for Neuromorphic Network Training". In: *2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC)*. 2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC). Salt Lake City, UT, USA: IEEE, Nov. 2016, pp. 36–46. ISBN: 978-1-5090-3882-4. DOI: 10.1109/MLHPC.2016.008. URL: http://ieeexplore.ieee.org/document/7835813/ (visited on 01/09/2020).

[47] Chetan Singh Thakur et al. "Large-Scale Neuromorphic Spiking Array Processors: A Quest to Mimic the Brain". In: *Frontiers in Neuroscience* 12 (2018). ISSN: 1662-453X. DOI: 10.3389/fnins.2018.00891. URL: https://www.frontiersin.org/articles/10.3389/fnins.2018.00891/full (visited on 02/17/2020).

[48] 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).

[49] 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.

[50] 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.

[51] 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).

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

[53] 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.