UNIVERSITY
OF OSLO

Master's Thesis

NEUROMORPHIC COMPUTING

With Spiking Neural Networks

Brage Wiseth

Departement of Informatics
Faculty of Mathematics and Natural Sciences

Philip Haflinger - Supervisor
Yngve Hafting - Co-Supervisor

2025-12-26

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae.

ACKNOWLEDGEMENTS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia.

CONTENTS

Wordcount: 17204

GLOSSARY

***AI* – Artificial Intenligence** 4, 5, 5, 7, 7, 34

***ANN* – Artificial Neural Network** 10, 33

***CMOS* – Complementary Metal-Oxide-Semiconductor** 5

***CNN* – Convolutional Neural Network** 11

***CPU* – Central Processing Unit** 5

***GPT* – Generative Pre-trained Transformer** 11

***GPU* – Graphics Processing Unit** 4, 5, 11

***LLM* – Large Language Model** 11

***LSTM* – Long Short-Term Memory** 11

***MLP* – Multi Layer Perceptron** 9, 10, 10, 11

***RNN* – Recurrent Neural Network** 11

***SNN* – Spiking Neural Network** 32, 32, 33

1 • INTRODUCTION

The development of intelligent machines is a significant objective in modern science and engineering. While the concept has historical roots in philosophy and early automata, the field has transitioned from speculative theory to practical application. Currently, artificial intelligence is central to technological and economic development. Understanding the mechanisms of intelligence and reproducing them in synthetic systems offers the potential for improved analysis of biological minds and the creation of tools for applications ranging from personalized medicine to automated scientific discovery.

In recent years, great strides has been made towards this goal. Deep Learning, which utilizes multilayered neural networks, has exceeded previous performance benchmarks. These systems have demonstrated high proficiency in tasks previously limited to human capability. For example, AlphaFold has addressed complex problems in protein folding [1], reinforcement learning agents have mastered the complexity of games such as Go [1], and Large Language Models have shown capabilities in text generation that approach human fluency. Consequently, Artificial Intenligence (AI) is increasingly viewed as a general-purpose technology that may influence societal infrastructure.

However, despite these advances, there are significant limitations to the current approach. The success of modern deep learning relies heavily on scaling, which involves increasing data volume and computational power. This strategy is approaching physical and economic boundaries. Training state-of-the-art models consumes substantial energy and results in a large carbon footprint [1]. Although specialized hardware allows for more efficient computations, the underlying architecture and algorithms imposes an intrinsic limit on scalability independent of the underlying hardware. Furthermore, the requirement for massive datasets presents challenges in sourcing and curation. Additionally, evidence suggests that this scaling approach yields diminishing returns. Models often function as statistical correlation engines; they lack common-sense reasoning, struggle with out-of-distribution generalization, and are prone to brittle failure modes [1].

These limitations are evident when comparing artificial systems to biological intelligence. The human brain demonstrates that high-level intelligence is possible without massive energy consumption or dataset sizes. The brain operates on approximately 20 watts [1]. With this limited energy budget, it manages biological functions, processes real-time multi-sensory data, and performs abstract reasoning. In contrast, deep learning models require Graphics Processing Unit (GPU) clusters with significantly higher power requirements to match a fraction of these capabilities. There is also a discrepancy in learning efficiency. Deep learning models are sample-inefficient, often requiring vast numbers of examples to learn a representation. Biological systems, however, are capable of “one-shot” or “few-shot” learning and can acquire new information without catastrophic forgetting. This suggests the inefficiency of current AI is a paradigmatic issue rather than just an engineering problem.

The proposed direction for addressing these issues involves biological inspiration in both hardware and algorithm design, specifically Neuromorphic Computing. This field attempts to engineer computer architectures that mimic the biological structure of the nervous system. Unlike traditional AI, which runs as software on general-purpose hardware, neuromorphic engineering aims to align the algorithm with the physical substrate. It moves away from clock-driven processing toward asynchronous, event-driven systems. In this paradigm, information is encoded as sparse, discrete events or "spikes." Similar to biological neurons, a neuromorphic processor consumes minimal energy when inactive, processing information only when triggered. This approach is being pursued by both industrial groups, such as Intel's Loihi [1], and academic projects like SpiNNaker [1]. These systems represent a shift from calculation-based machines to those capable of real-time adaptation.

Although neuromorphic systems achieve optimal performance on co-designed platforms—where the algorithm is embedded directly into the hardware—there is significant value in executing neuromorphic algorithms on traditional von Neumann architectures. In this thesis, we explore biologically inspired algorithms deployed on traditional Central Processing Unit (CPU) and GPU hardware. We examine how event-driven, biologically plausible computation can address limitations in scalability, data efficiency, and energy consumption, even when simulated on standard processors. We present approaches for efficient information coding and learning algorithms inspired by neural mechanisms. However, biological inspiration is constrained by engineering realities. We cannot strictly replicate the brain's 3D connectivity due to the planar limitations of Complementary Metal-Oxide-Semiconductor (CMOS) manufacturing. Therefore, a pragmatic approach is necessary. We must identify and abstract the functional principles of the brain, separating core computational mechanisms from biological details that may be evolutionary artifacts. Concretely, this thesis aims to:

- **Investigate Sparse Information Flow**: Explore how information can be encoded and processed using sparse, asynchronous events (spikes) within a neural network.
- **Develop Biologically Inspired Learning**: Explore and evaluate learning algorithms that are suitable for such networks, adhering to the constraints of locality and efficiency.

The succeeding chapter lays the historical and theoretical foundation, covering early neuroscience and the development of artificial neural networks based on simple models of the brain. Following this, we review relevant modern neuroscience literature, extracting key concepts that will inform the methodology. We also provide consise overview on machine learning concepts and frameworks. Finally, we detail the implementation of these principles in a neuromorphic context and evaluate their performance against standard benchmarks.

2 • BACKGROUND

This section outlines the historical and theoretical evolution of AI, reviewing key concepts in modern neuroscience that motivate the methodology used in this thesis.

We begin at a shared origin point, a time when AI research and neuroscience were intertwined. We then trace the diverging path that led to modern Deep Learning, examining why it has drifted from biological plausibility. Subsequently, we explore the "neuromorphic path". In Section 2.2, we detail the specific physical principles and neuroscientific insights upon which the neuromorphic methods in this thesis is built. In Section 2.3, we contrast the architectural mechanics of deep learning and neuromorphic systems, specifically addressing why the former is computationally powerful yet energetically inefficient. We conclude with a review of existing frameworks, identifying their strengths and weaknesses to contextualize the contributions of this work.

2.1 • HISTORY & DEVELOPMENTS

Historically, the understanding of neural tissue was dominated by the reticular theory, which claimed that the brain consisted of a continuous, fused network of nerve fibers. This paradigm was fundamentally challenged by the work of Santiago Ramón y Cajal. Through the application of novel staining techniques, Cajal established the neuron doctrine, demonstrating that the nervous system is composed of discrete, individual cells [1]. Building on these findings, Heinrich Wilhelm Gottfried Von Waldeyer-Hartz proposed the "Neuron Doctrine" and coined the term "neurons" to describe these dicrete cells. Subsequent analysis using electron microscopy has provided irrefutable validation of this discrete cellular structure.

The conceptualization of the brain as a collection of discrete units facilitated the development of mathematical models describing neural function. In 1943, Warren McCulloch and Walter Pitts published A Logical Calculus of the Ideas Immanent in Nervous Activity, introducing the first formal model of the neuron.

The McCulloch-Pitts (M-P) neuron abstracted biological complexity into a binary decision device governed by the following logic:

- **Integration**: The neuron receives multiple binary inputs, weighted as either excitatory or inhibitory.
- **Summation**: The unit calculates the linear sum of these weighted inputs.
- **Thresholding**: If the aggregate sum exceeds a fixed threshold, the neuron outputs a 1 (firing); otherwise, it outputs a 0 (silence).

McCulloch and Pitts demonstrated that networks of these units could theoretically compute any logical operation (AND, OR, NOT) [1]. This abstraction established the foundational link between biological processes and digital logic, suggesting that neural function could be replicated in electronic hardware. Consequently, the M-P neuron serves as the common ancestor for both computational neuroscience and artificial intelligence.

However, despite its theoretical utility, the original M-P model presented significant functional limitations. The connectivity was static, requiring circuits to be manually designed rather than learned. Furthermore, the restriction to binary weights precluded the modeling of graded signal intensity, preventing the system from capturing the nuance of real-world sensory input.

In 1949, Donald Hebb addressed the critical issue of plasticity in his work The Organization of Behavior. He proposed a theoretical mechanism for synaptic modification, now known as Hebbian learning, which provided a biological basis for how neural networks could adapt over time. Hebb postulated:

> "Let us assume that the persistence or repetition of a reverberatory activity (or "trace") tends to induce lasting cellular changes that add to its stability. … When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased" [1].

This principle is colloquially summarized as "neurons that fire together, wire together" [1]. Crucially, this describes a local and decentralized learning rule; a synapse does not require a global error signal or external supervision to adjust. It requires only the correlation between the pre-synaptic input and the post-synaptic output. The convergence of the M-P architectural model and the Hebbian plasticity framework established the prerequisite conditions for the development of modern neural networks.

2.1.1 • THE PERCEPTRON

In 1957, Frank Rosenblatt advanced these theoretical concepts by engineering the Perceptron. The "Mark I Perceptron" was a hardware implementation of the neural model, distinguished by a crucial innovation: a weight-adjustment mechanism based on Hebbian principles. Rosenblatt introduced the perceptron learning rule, an iterative algorithm capable of minimizing error automatically. The system processed an input pattern (e.g., a pixelated character) and produced a binary classification. When the output deviated from the target, the algorithm adjusted the weights proportional to the error: strengthening connections that should have contributed to a correct firing and weakening those that led to false positives.

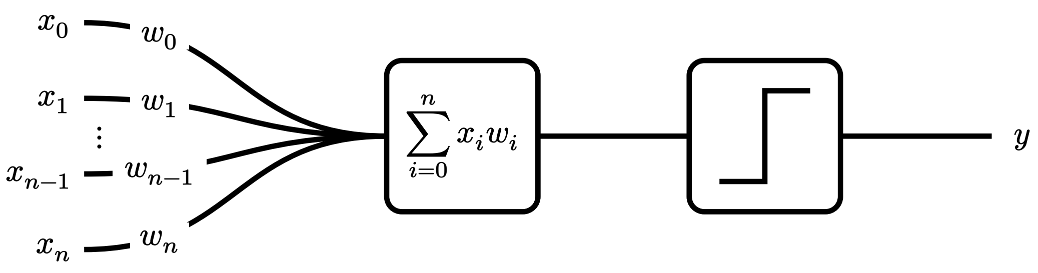

Figure 1: The perceptron model. Inputs x_i are multiplied by weights w_i and summed. If the linear combination $\sum x_i w_i$ exceeds the bias b, the neuron activates.

Consequently, the Perceptron was capable of converging on a solution for any problem where the data was linearly separable. This success generated significant enthusiasm, with contemporary reports suggesting that such machines would soon mimic human consciousness [1].

These expectations were abruptly tempered by theoretical limitations. In 1969, Marvin Minsky and Seymour Papert published Perceptrons, a rigorous mathematical analysis of the architecture. They demonstrated that a single-layer perceptron is fundamentally a linear classifier. While capable of learning operations like AND or OR, it is mathematically incapable of solving the XOR (Exclusive OR) problem. In the XOR case, the classes cannot be separated by a single hyperplane. This proof highlighted a severe boundary on the utility of single-layer networks for complex, non-linear tasks.

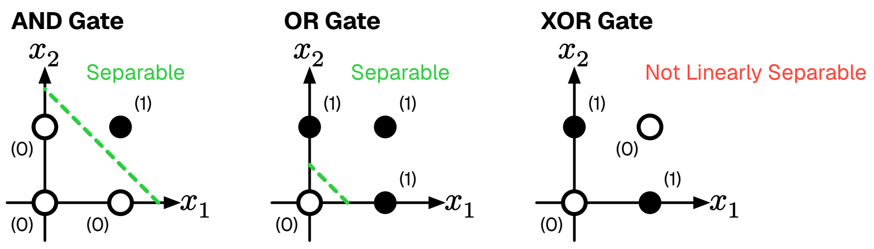

Figure 2: The XOR problem. Unlike AND/OR, the data points for XOR cannot be separated by a single linear boundary.

The publication of Perceptrons coincided with a significant reduction in neural network research funding, a period retrospectively termed the “First AI Winter”. It is worth noting that Minsky and Papert acknowledged that a Multi Layer Perceptron (MLP), a network stacking multiple layers of neurons, could theoretically solve the XOR problem by creating complex, non-linear decision boundaries.

However, a critical algorithmic gap remained: the “credit assignment problem”. While researchers knew that hidden layers could represent complex features, there was no known method to propagate error signals back through the layers to adjust the weights of hidden neurons effectively. Rosenblatt’s rule was mathematically valid only for the output layer. The field remained stagnant until a method for training multi-layer networks could be formalized.

2.1.2 • DEEP LEARNING

The critique presented by Minsky and Papert precipitated a contraction in funding; despite this, theoretical inquiry persisted. It was widely hypothesized that the limitations of the single perceptron could be overcome by a MLP. By organizing neurons into hierarchical layers, the network could theoretically perform successive non-linear transformations on the input space, enabling the formation of complex decision boundaries. The primary impediment was not the architecture itself, but the absence of a viable learning algorithm.

In a single-layer perceptron, error attribution is immediate: if the output deviates from the target, the error is directly derived from the weights of the output layer. However, in a multi-layer architecture, quantifying the contribution of a specific neuron within the "hidden" layers to the final output error presents a significant challenge. This is formally known as the Credit Assignment Problem [1], and it remained the central theoretical obstacle for over a decade.

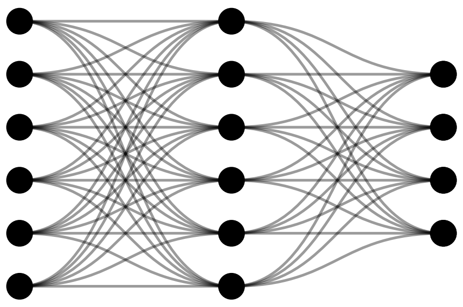

Figure 3: A MLP. By inserting "hidden layers" between input and output, the network can approximate non-linear functions such as XOR. The historical challenge lay in deriving a method to train these intermediate layers.

The solution to this theoretical impasse was popularized in 1986 by Rumelhart, Hinton, and Williams in their seminal paper *Learning representations by back-propagating errors* [1]. They demonstrated that the Chain Rule of calculus could be applied recursively to propagate the error signal from the output layer backwards through the hidden layers. This algorithm, known as Backpropagation, allowed the network to calculate the gradient of the loss function with respect to every weight in the system. Effectively, it provided a mathematical method to tell each hidden neuron exactly how much it contributed to the total error, finally solving the credit assignment problem.

Unlike Hebbian plasticity, which is local and biological, Backpropagation relies on global error signals and precise backward data flow—mechanisms effectively absent in organic tissue. Consequently, the field of Artificial Neural Network (ANN) effectively decoupled from neuroscience. It transitioned into a branch of engineering and applied mathematics, prioritizing statistical optimization over biological realism. Paradoxically, it was this abandonment of biological fidelity that enabled the rapid scaling and performance breakthroughs that followed.

ACHIEVEMENTS

With the training mechanism solved, the field exploded. The combination of Backpropagation, massive datasets, and GPU hardware led to a "Cambrian Explosion" of neural architectures, each solving domains previously thought impossible for computers.

The revolution began in earnest with computer vision. Convolutional Neural Networks (CNNs), such as AlexNet (2012) [1] and later ResNet [1], introduced the idea of learning hierarchical features—detecting edges, then shapes, then objects—much like the human visual cortex. This allowed machines to classify images with superhuman accuracy.

Soon after, the focus shifted to sequence data. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) architectures gave machines a short-term memory, enabling breakthroughs in speech recognition and machine translation. However, the true paradigm shift occurred with the introduction of the Transformer architecture in 2017. By utilizing an "attention mechanism" to parallelize the processing of language, Transformers allowed for the training of massive Large Language Models (LLMs) like the Generative Pre-trained Transformer (GPT).

These techniques have even transcended media generation. Deep Learning has solved fundamental scientific problems; notably, DeepMind's AlphaFold utilized these architectures to predict the 3D structure of proteins from their amino acid sequences, a 50-year-old grand challenge in biology [1].

SHORTCOMINGS

Deep learning has achieved substantial success across various domains. However, this performance relies heavily on computational scaling. The underlying algorithms, while originally inspired by biological principles, have diverged significantly to prioritize mathematical optimization on standard hardware. By simulating neural networks on architectures not designed for them—using algorithms developed to address the specific constraints of the MLP rather than the broader goal of efficient intelligence—this approach faces fundamental barriers that cannot be overcome merely by increasing hardware resources.

A primary limitation stems from the underlying computer architecture. Modern systems rely on the Von Neumann architecture, which physically separates the processing unit from the memory. Deep neural networks, which are defined by large matrices of synaptic weights, require constant data transfer between these components. For every token generated or inference step performed, billions of weight parameters must be fetched from memory, processed, and written back. This creates a memory bottleneck where system performance is limited not by processing speed, but by the available bandwidth to memory [1].

Concurrently, the computational cost of training state-of-the-art models is growing exponentially. Deep learning relies on dense matrix multiplications, where the number of operations scales quadratically with the network size. As models grow to encompass trillions of parameters to achieve marginal gains in performance, the hardware requirements become economically and physically unsustainable. This reliance on brute-force scaling yields diminishing returns, suggesting that the current architectural paradigm is approaching an efficiency plateau.

This architectural mismatch manifests acutely in energy consumption. In modern CMOS technology, the energy cost of moving data significantly exceeds the cost of processing it. Retrieving a single byte of data from off-chip DRAM consumes approximately three orders of magnitude more energy than performing a floating-point operation on that data [1]. As highlighted in the introduction, this discrepancy has led to the proliferation of "Megawatt Models." In stark contrast, the human brain operates on approximately 20 watts. By co-locating memory and computation, biological systems manage complex multimodal processing with an energy budget comparable to a dim lightbulb.

Furthermore, backpropagation-based learning exhibits low sample efficiency compared to biological systems. Deep learning models often require millions of examples to establish robust representations, whereas biological agents demonstrate "one-shot" or "few-shot" learning capabilities. Additionally, the resulting models function as "black boxes." While empirically effective, their distributed internal representations are often opaque, making it difficult to trace the causal logic behind specific decisions or failures.

These limitations can be traced to the deviation from biological constraints. To solve the training problem, mainstream AI adopted mechanisms that are biologically implausible. Backpropagation relies on a global error signal and requires the backward pass to use the exact same synaptic weights as the forward pass (the "weight transport problem"). In biological tissue, synapses are unidirectional, and there is no known mechanism for a neuron to access the weight of a downstream synapse to calculate a gradient.

While a detailed technical analysis of these inefficiencies is presented in Section 2.3, the broader implication is clear: we have achieved artificial intelligence at the cost of efficiency and explainability. This realization has renewed interest in alternative architectures that align more closely with biological principles. Although the majority of research has focused on deep learning, a parallel subset of the field has continued to investigate systems that mimic the physical operation of the nervous system.

2.1.3 • BIRTH OF NEUROMORPHIC

While the artificial intelligence community debated symbolic logic versus connectionism during the "AI Winter," significant developments were occurring in hardware physics. In the late 1980s at Caltech, physicist Carver Mead—a pioneer of VLSI (Very Large Scale Integration) design—began to question the trajectory of digital computing.

Mead observed that while digital computers were becoming exponentially faster, they were also becoming less efficient in terms of energy per operation. He noted that using transistors as rigid, high-power switches to perform boolean logic was energetically wasteful compared to the biological systems they aimed to emulate.

In 1990, Mead published his seminal paper, *Neuromorphic Electronic Systems* [1], coining the term “neuromorphic” to describe hardware that mimics the biological structure of the nervous system. His thesis proposed that rather than simulating neural equations via software on digital computers, engineers should construct physical hardware that exploits the same physical laws as the biological nervous system.

The foundational insight of the field was the physical analogy between silicon physics and ion-channel physics. In standard digital electronics, transistors are operated in “strong inversion,” driven by high voltages to act as binary switches. Mead realized that a single transistor, operating in its “subthreshold” region, follows the same exponential Boltzmann statistics that govern the flow of ions through biological channels.

This realization implied that a single transistor could physically compute the non-linear functions used by biological neurons, but with significantly higher speed and lower power consumption. Consequently, synaptic functions could be implemented by single transistors rather than complex arrangements of logic gates.

To demonstrate this concept, Mead and his doctoral student Misha Mahowald developed the **Silicon Retina** in 1991 [1]. Unlike a standard camera, which captures full frames at fixed intervals (generating redundant data), the Silicon Retina operated asynchronously. It utilized analog circuits to compute spatial and temporal derivatives directly on-chip, outputting discrete “events” only when local light intensity changed.

This event-driven approach solved the redundancy problem inherent in frame-based sampling. If the scene remained static, the system transmitted no data and consumed negligible energy. This demonstrated that by aligning the hardware physics with the computational task, sensory information could be processed with a fraction of the power required by conventional digital systems.

Since the inception of neuromorphic computing, neuroscience has also advanced significantly. While Mead’s early work was based on the physical intuition of the transistor, modern neuromorphic engineering now incorporates a richer understanding of neuronal dynamics, synaptic plasticity, and network architecture. To advance the field, we must combine these foundational hardware insights with the principles of modern mechanistic neuroscience.

2.2 • BIOLOGICAL PRINCIPLES

The biological brain remains the gold standard for energy-efficient, robust, and adaptive computation. Modern neuroscience has revealed that the brain's efficiency stems from its specific physical mechanisms, not just its connectivity. To engineer systems that truly rival biological performance, we cannot simply mimic the brain's output; we must emulate its internal dynamics. We must move away from the spherical cow abstractions of early cybernetics and look at the neuron as it actually functions: a complex, time-dependent, and event-driven processor.

This section provides a mechanistic overview of the nervous system, translating biological observations into the computational primitives required for neuromorphic engineering. It explores the structural hierarchy of the neuron, the physics of the action potential, and the mathematical models used to capture these dynamics in silicon.

2.2.1 • NEURON STRUCTURE & FUNCTION

The neuron is the fundamental computational unit of the brain as outlined in section 2.1 - neuron doctrine. While it shares standard cellular machinery (like mitochondria and a nucleus) with other cells, it is morphologically specialized for information transmission. A neuron consists of three functional zones:

- **The Input (Dendrites)**: A branching tree structure that collects signals from thousands of upstream neurons. This is where inputs are integrated.
- **The Integration Zone (Soma)**: The cell body where electrical potentials from the dendrites summate.
- **The Output (Axon)**: A long, cable-like structure that transmits the neuron's own signal to downstream targets.

These structures allow neurons to form complex networks connected by synapses. Unlike the weighted lines in an artificial neural network, a biological synapse is a complex chemical junction. When a pre-synaptic neuron activates, it releases neurotransmitters (packets of chemicals) that bind to the post-synaptic dendrite, causing ion channels to open and current to flow.

To successfully emulate the brain, one must first understand the physical "chassis" of biological intelligence. The neuron is the fundamental computational unit of the nervous system. While it shares standard cellular machinery with other cell types—such as a nucleus and mitochondria—it is morphologically specialized for the rapid reception, integration, and transmission of information.

The structure of a neuron is polarized, meaning information flows in a specific direction through three distinct functional zones. The process begins at the dendrites, a vast, tree-like structure branching out from the cell body. These dendrites act as the input stage, covered in thousands of receptor sites that capture chemical signals from upstream neurons. These signals are conducted toward the central cell body, or soma. The soma serves as the integration center; it collects the minute electrical currents generated by the dendrites and sums them together. Finally, the processed signal is transmitted via the axon, a long, cable-like projection that extends toward other cells. Many axons are wrapped in myelin, a fatty insulating layer that functions similarly to wire cladding, ensuring the signal travels efficiently over long distances without loss.

Functionally, the neuron operates as an electrochemical battery. Unlike digital circuits that rely on the flow of electrons, the brain computes by moving ions —specifically sodium (Na+) and potassium (K+)—across a liquid membrane. By actively pumping positive ions out of the cell, the neuron maintains a stable voltage difference known as the resting potential, typically around −70 mV. In this state, the cell is polarized, effectively storing potential energy like a compressed spring, waiting for an input to trigger a release. The computational decision-making process occurs when the neuron receives input. As neurotransmitters bind to the dendrites, ion channels open, allowing positive ions to flow back into the cell. This raises the internal voltage, a process called depolarization. These voltage changes accumulate at the soma. If the sum of these inputs causes the membrane potential to cross a critical threshold (approximately −55 mV), the neuron undergoes a rapid, non-linear reaction. Voltage-gated channels snap open, causing a massive influx of ions and generating an action potential—a sharp electrical pulse that travels down the axon. Immediately after firing, the neuron pumps the ions out to reset its voltage, entering a brief refractory period where it cannot fire again. This “all-or-nothing” mechanism ensures that noise is

Figure 4: Rendering of a morphilogicaligy plausibe neuron

filtered out, transmitting only significant, integrated information to the network. Troughout the brain there are several types of nourons such as GLIA cells, which acts as supporting structure and energy delivery. The wast majority of neurons in the brain are the spiking neurons which all communicate via a uniform action potential. There are some neurons that emit graded potentials where the shape of the outputed signal can vary in shape and form.

2.2.2 • ACTION POTENTIAL & SPIKE TRAINS

The defining feature of biological computation is the Action Potential or spike. In standard Artificial Neural Networks (ANNs), neurons communicate using continuous values (e.g., 0.75) representing an average firing rate. In contrast, biological neurons communicate using discrete, binary events.

The mechanism is all-or-nothing. The neuron maintains a voltage difference across its membrane (Membrane Potential). As inputs arrive, this potential fluctuates. If the potential crosses a specific Threshold, the neuron fires a spike—a rapid, uniform pulse of voltage that travels down the axon.

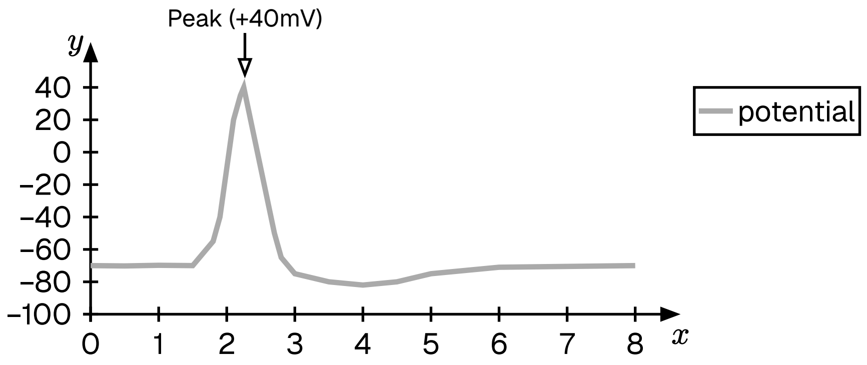

Figure 5: action potential

This "Spiking" paradigm shifts the nature of computation from Spatial (what represents the data) to Spatio-Temporal (when the data happens):

- Event-Driven: Computation only occurs when data changes (a spike arrives). This creates inherent sparsity and energy efficiency.
- Temporal Coding: Information can be encoded in the precise timing of spikes (e.g., time-to-first-spike) rather than just the frequency.

Action potentials are in general uniform and look the same for every neuron every time they spike, Graded potentials are primarily generated by sensory input

The artificial neurons used in most deep learning models (like ReLU or sigmoid units) are static. They compute a weighted sum of their inputs, apply an activation function, and output a single, continuous value (like 0.83 or 5.2). This value is assumed to represent the neuron's firing rate. Biological neurons don't work this way. They are spiking neurons, and their computation is:

- Temporal: They integrate inputs over time. A neuron's internal state (its membrane potential) rises and falls based on when inputs arrive.
- Event-Driven: They do not communicate with continuous values. They communicate using discrete, all-or-nothing electrical pulses called action potentials, or "spikes." A neuron only fires a spike when its internal potential crosses a specific threshold.
- Efficient: Because they are event-driven, they are sparse. A neuron spends most of its time silent, only consuming energy when it receives or sends a spike.

In this model, information is not just in how many spikes there are (a rate code), but when they occur (a temporal code). A spike arriving a few milliseconds earlier or later can completely change the computational outcome. While the biological action potential is a complex, continuous voltage fluctuation that lasts approximately one to two milliseconds, it is remarkably consistent in shape. For a given neuron, every spike looks nearly identical. This observation leads to a crucial simplification in neuromorphic engineering: if the shape of the spike is constant, it carries no information. The information must therefore be contained entirely in the timing of the event.

To model this mathematically, we abstract the physical voltage pulse into a dimensionless point process. We treat the spike as an event that occurs at a precise instant in time, tf, with zero duration. The mathematical tool used to represent this is the Dirac delta function, delta(t).

The Dirac delta is an idealized function defined by two properties: it is zero everywhere except at $t = 0$ (where it is infinitely high), and its integral over time is exactly equal to 1.

$$\delta(t) = \begin{cases} \infty \text{ if } t = 0 \\ 0 \text{ if } t \neq 0 \end{cases} \int_{-\infty}^{+\infty} \delta(t) \, \mathrm{d}t = 1$$

Equation 1: The defining properties of the Dirac delta function.

In the context of a neural network, we represent the output of a neuron not as a continuous voltage curve, but as a “spike train”—a sequence of these impulses summed together. If a neuron fires spikes at specific times $t_1, t_2, ..., t_n$, the resulting signal $S(t)$ is modeled as:

$$S(t) = \sum f\delta(t - t((f)))$$

Equation 2: A spike train represented as a sum of Dirac delta functions.

This abstraction is powerful because it simplifies the interaction between neurons. In the Leaky Integrate-and-Fire (LIF) model, when this delta function arrives at a synapse, it acts as an instantaneous “kick” to the current. Because the integral of the delta function is 1, integrating this input simply adds a discrete step-change to the post-synaptic neuron’s membrane potential, perfectly mimicking the rapid charge deposition of a real biological synapse without requiring us to simulate the complex curve of the voltage pulse.

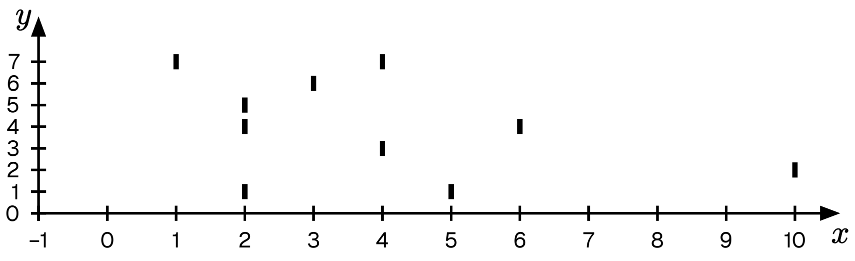

Figure 6: Spike Train

2.2.3 • NEURON MODELS

Real neurons exhibit complex non-linear dynamics, often modeled by the Hodgkin-Huxley equations which simulate individual ion channel conductances. However, these are too computationally expensive for large-scale neuromorphic systems.

Instead, we use the Generalized Leaky Integrate-and-Fire (GLIF) model. It captures the essential behavior of a neuron—integration of inputs and threshold firing—without simulating the chemical micro-physics.

The dynamics of the membrane potential u(t) are governed by a linear differential equation:

Where taum is the membrane time constant (how fast the neuron “forgets” input), u“rest” is the resting potential, and I(t) is the input current. In a spiking network, the input current I(t) is not continuous, but a series of discrete spikes arriving at specific times. We can model this input as a sum of Dirac delta functions, scaled by the synaptic weight q:

Because the system is linear below the threshold, we can solve this differential equation analytically. The membrane potential at any time t is the sum of the decaying effects of all previous incoming spikes:

This equation represents the core of most neuromorphic algorithms. It defines a system that integrates information over time, leaks it to ensure temporal relevance, and resets upon firing. While it simplifies away complex bifurcation dynamics (like Hopf bifurcations or chaotic attractors found in biology), the GLIF model provides the necessary abstraction to build scalable, event-driven computing architectures.

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -(u - u_{\text{rest}}) + R(u)I(t)$$

Equation 3: Generalized leaky integrate and fire differential equation gouvering the dynamics of a neurons membrane potentia

Considering all input currents to be uniform packets (spikes), the dirac delta function fits well into the mathematical framework

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -(u - u_{\text{rest}}) + R(u)q\delta(t)$$

Equation 4: Generalized leaky integrate and fire differential equation gouvering the dynamics of a neurons membrane potential

solving the equation

$$u(t) = u_{\text{rest}} + \sum_f (u_r - \varphi) \exp\left(-\frac{t - t(f)}{\tau_m}\right) + \frac{R}{\tau_m} \int_0^\infty \exp\left(-\frac{s}{\tau_m}\right) I(t - s)\,\mathrm{d}s$$

Equation 5: Generalized leaky integrate and fire differential equation gouvering the dynamics of a neurons membrane potential

In the quest to simulate the brain, there exists a fundamental trade-off between biological realism and computational efficiency. At the high end of the spectrum lie conductance-based models, most notably the Hodgkin-Huxley model (1952). This model describes the neuron not as a simple integrator, but as an electrical circuit with variable resistors representing the precise opening and closing dynamics of sodium and potassium ion channels.

Large-scale initiatives, such as the Blue Brain Project, utilize these highly detailed “multi-compartment” models. They simulate the neuron not as a single point, but as a complex 3D structure, breaking the dendrites and axon into hundreds of distinct segments to model how current flows through the physical geometry of the cell. While these simulations provide invaluable insights into pharmaceutical interactions and pathology, they are computationally prohibitive for real-time engineering. Simulating a single second of brain activity using these models can require supercomputers and massive energy expenditure.

To build practical neuromorphic hardware, engineers must simplify these equations into the Generalized Leaky Integrate-and-Fire (GLIF) or Izhikevich models. However, this abstraction comes at a cost in behavioral fidelity.

A standard LIF neuron is a "one-dimensional" system. Its state is defined entirely by a single variable: the membrane potential. Consequently, it supports only a limited set of firing modes. It can handle tonic spiking (regular, repeated firing under constant input) and phasic spiking (firing once at the onset of input).

However, a standard LIF neuron cannot replicate complex behaviors such as bursting (clusters of rapid spikes followed by silence) or chattering. These behaviors require a "memory" of previous activity that lasts longer than the simple membrane time constant.

To capture these complex dynamics without reverting to the heavy Hodgkin-Huxley equations, the model must be augmented with a second variable, typically called the adaptation variable (w).

$$\tau_m \frac{\mathrm{d}u}{\mathrm{d}t} = -(u - u_{\text{rest}}) + RI(t) - w\tau_w \frac{\mathrm{d}w}{\mathrm{d}t} = a(u - u_{\text{rest}}) - w$$

Equation 6: The Adaptive LIF model. The variable w provides negative feedback, allowing for bursting dynamics.

In this "Adaptive" formulation, every time the neuron spikes, the variable w increases, acting as a drag or "fatigue" on the membrane potential. This negative feedback loop allows the simple linear model to exhibit sophisticated neuro-computational properties, including spike-frequency adaptation and bursting, bridging the gap between silicon efficiency and biological complexity.

When pre-synaptic neurons fire, they release neurotransmitters (like glutamate or dopamine) across the synapse. These chemicals open ion channels on the post-synaptic neuron, causing its internal electrical potential (the "membrane potential") to increase. If this potential, which is integrated over time, reaches a critical threshold, the neuron itself fires an action potential (a spike) down its axon.

Although the perceptron captures the basic idea of "integrating inputs to make a decision," a lot is left on the table. A lot of progress and new ideas have surfaced since its invention. The neuron, once thought to be a simple switch, turns out to be a complex computational device.

This forces us to rethink everything:

- Information Encoding: How is information represented? Is it in the rate of spikes, the precise timing of the first spike, or in correlations between patterns of spikes? This is a key research topic not explored by older models.
- Learning Rules: How does the brain learn? It is entirely different from what deep learning uses. The brain's learning is local (like STDP) and continuous.
- Network Architecture: The brain isn't a simple feed-forward stack of layers. It is a deeply recurrent, complex, and fully asynchronous graph where signals propagate at their own speed.

A lot of work has been put into modeling the neuron. The most biologically realistic models, like the Hodgkin-Huxley model, are incredibly complex and computationally expensive, simulating the precise dynamics of multiple ion channels.

For most neuromorphic computing, a balance is struck with simpler, more efficient models. The most popular are the Leaky Integrate-and-Fire (LIF) models.
- Integrate: The model's potential rises as it receives input spikes (summing their weights).
- Leaky: The potential "leaks" away over time, modeling the neuron's tendency to return to its resting state.
- Fire: If the potential crosses the threshold, it fires a spike and its potential is reset.

The Generalized Leaky Integrate-and-Fire (GLIF) and other variants (like the Izhikevich model) are the best models we have today for large-scale, efficient simulation. They add more biological realism (like adaptation, where a neuron's firing rate slows over time) without the extreme computational cost of full biophysical models.

We will use GLIF neuron for their simpliity and computational effeciency … Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

2.2.4 • NEURAL CODING

In digital representaton, a piece of information can be represented in various different ways, all using bits as the lowest information quanta. combining bits we get a richer representation. float is a popular choice of combining bits. For instance representing a luminance value of a pixel in an image is straight forward, each pixel gets its own float. Representing information using analog systems offers far more presicion, in electronic systems, values can be represented with currents or voltages which can be any value. The brain uses action potentials to communicate, action potentials are analog voltages so we might excpect that information gets packaged into the shape of the action potential, however as we have discussed the vast majority of neurons in the brain are spiking neurons, and their action potentials are uniform in size, looking more like individual bits in time. The information stored in brain signals are much more similar to the discrete digital representation than analog. The information must be encoded in the relative timing between spikes.

It is observed that neurons fire in short bursts called spikes. Experiments show that neurons fire repetably. A sequence of spikes is called a spike train, and exactly how information is encoded in a spike train is a topic of hot debate in neuroscience. A popular idea is that information is encoded in the average value of spikes per time called rate encoding. Temporal encoding the brain most likely uses a combination of all. The time to first spike encoding could be understood like this it is not about the absolute timing of the neurons rather a race of which spikes come first. the first connections would exite the post-synaptic neurons first and they should inhibit the others (lateral inhibition)

Understanding how the brain processes information requires deciphering the "neural code"—the set of rules by which sensory stimuli and internal states are translated into sequences of action potentials. Unlike a digital bus where voltage levels map directly to binary values, the brain utilizes a more complex, multi-faceted signaling scheme.

The most traditional interpretation of neural activity is Rate Coding. In this paradigm, information is conveyed by the average frequency of action potentials over a specific window of time. A strong stimulus, such as a bright light or heavy pressure, results in a high firing rate, while a weak stimulus results in few or no spikes. This model treats the neuron essentially as an analog-to-digital converter where the precise timing of individual spikes is considered noise, and only the mean activity carries the signal. While rate coding is robust and easily observed in motor neurons, it suffers from significant latency; the system must wait to count spikes over a period of time before a value can be determined.

However, the rapid reaction times observed in biological systems suggest that rate coding is insufficient for time-critical processing. This has led to the prominence of Temporal Coding theories in neuromorphic engineering. In temporal coding, the precise timing of a spike carries significant information. A primary example is "time-to-first-spike" coding, where the intensity of a stimulus is mapped to the latency of the response; a stronger stimulus causes the neuron to fire earlier relative to a stimulus onset.

This shift from rate to timing fundamentally changes the computational model. By utilizing the exact arrival time of an action potential, neural circuits can process information asynchronously and sparsely. A decision can be made as soon as the first few salient spikes arrive, rather than waiting to average activity over a long duration. This temporal precision allows the brain to maximize information transmission per spike, achieving the extreme energy efficiency that neuromorphic hardware aims to replicate.

However, relying solely on single neurons to represent complex data is risky due to biological noise and component failure. To ensure robustness, the nervous system employs Population Coding. In this scheme, variables—such as the direction of arm movement or the orientation of a visual edge—are not represented by a single neuron, but by the joint activity of a large ensemble. Each neuron in the population has a "tuning curve," responding maximally to a specific value but also firing weakly for similar values. By averaging the vectors of activity across the entire population, the brain can extract a precise signal from noisy individual components. This distributed representation ensures that the failure of a few neurons does not corrupt the message, a principle of fault tolerance that is highly desirable in neuromorphic hardware.

Finally, the brain optimizes for metabolic efficiency through Sparse Coding. This theory posits that neural systems minimize the number of active neurons required to represent a stimulus. At any given moment, out of billions of neurons, only a tiny fraction are firing. This stands in contrast to "dense" coding (where many units participate) or "local" coding (the hypothetical "grandmother cell" where one unique unit represents one unique object). Sparse coding strikes a mathematical balance: it creates a representation that has a high capacity for information but consumes minimal energy. For neuromorphic engineers, this is the holy grail: mimicking the brain's ability to process massive sensory streams while leaving the vast majority of the chip in a dormant, zero-power state.

RATE CODE

A rate code uses the firing rate of a single neuron as carrier of information

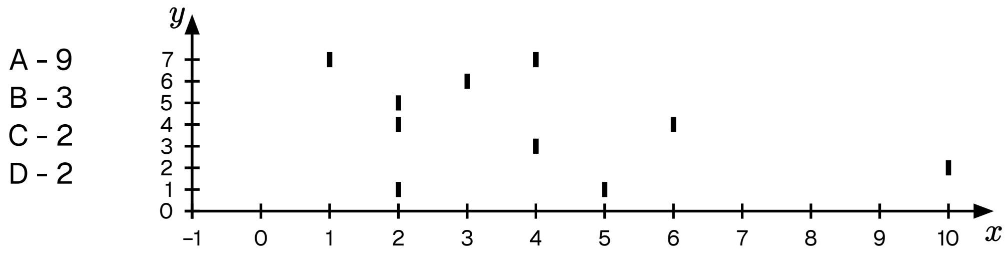

Figure 7: The perceptron—a simple model of how a neuron operates. Inputs x_i get multiplied by weights w_i and summed. If the sum $\sum w_i x_i$ surpasses a threshold (or "bias" b), the neuron fires.

TEMPORAL CODE

In an temporal code information is stored in time time to first spike, each neuron/input sends out one spike and the imformation is stored in the delay, shorter delay means stronger input, in such a sceme the computation could work with a balance of excitatiry and inhibitory neurons, the first to arrive (the stronges ones) win a "race" and inhibits others.

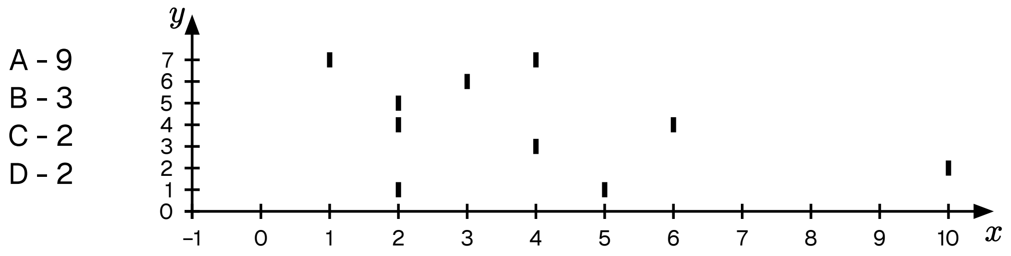

Figure 8: The perceptron—a simple model of how a neuron operates. Inputs x_i get multiplied by weights w_i and summed. If the sum $\sum w_i x_i$ surpasses a threshold (or "bias" b), the neuron fires.

POPULATION CODE

Both rate codes and temportal codes is most likely also combined with population codes in the brain

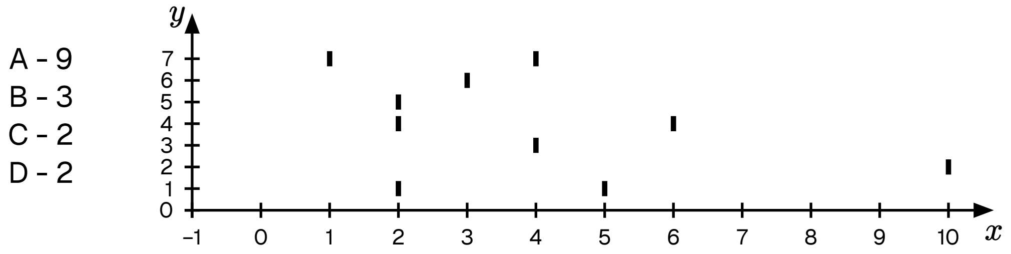

Figure 9: The perceptron—a simple model of how a neuron operates. Inputs x_i get multiplied by weights w_i and summed. If the sum $\sum w_i x_i$ surpasses a threshold (or "bias" b), the neuron fires.

SPARSE CODE

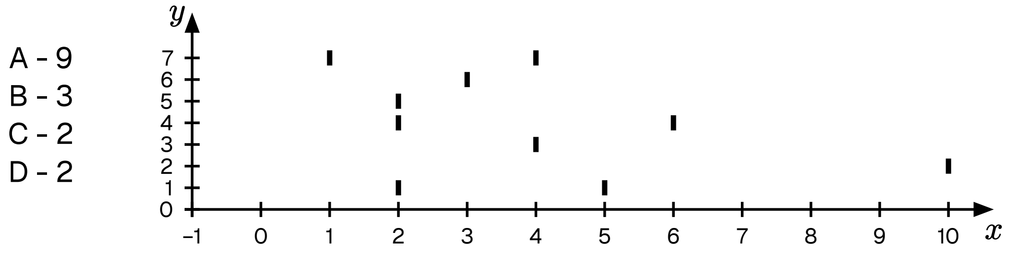

Figure 10: The perceptron—a simple model of how a neuron operates. Inputs x_i get multiplied by weights w_i and summed. If the sum $\sum w_i x_i$ surpasses a threshold (or "bias" b), the neuron fires.

PHASE AMBIGUITY FOR TEMPORAL ENCODINGS

For rate coding phase is a non issue as we can find the instantanious firing rate at any phase, for time to first spike encoding we need a reference signal. If the reference signal starts at time t0 we have started the phase and if the pattern does not match up with the reference signal we could miss it. Evidence suggests that brain waves could play the role of a global reference signal. This is the fundamental trade off between the two. A learning scheme where inputs have to occur in the same episode repeatedly two inputs that happen at the same time yields a stronger response. If the pattern is random then they would sometimes occur in the same episode and sometimes not. Two strong responses should occur in the same time frame.

NEURON CODES CAN COEXIST It was once theorized that the brain uses one or the other, but evidence shows that the same neural circuit can dynamically support both codes. Neurons can encode different features of a stimulus simultaneously using both the average firing rate and the precise timing of spikes (e.g., intensity by rate and frequency by timing in the auditory system). Context Determines Dominance: The brain may switch its primary reliance on one code or the other based on the situation, such as the duration of a stimulus or the level of background noise. For example, fast, synchronized inputs favor temporal coding, while slow, constant inputs favor rate coding. Population Coding is a "Scale" of Representation: Population coding refers to the idea that information is distributed across a large group of neurons rather than just one (specificity coding). The activity within this population can be measured in terms of:

- Population Rate Coding: The meaning is derived from the average firing rate across all the neurons in the population.
- Population Temporal Coding: The meaning is derived from the synchronization or precise relative timing of spikes among the neurons in the population.

2.2.5 • NEURAL NETWORKS & DYNAMICS

By now we have coverd the neuron behaviour on its own. The neuron on its own is not very usefull and needs to connect to other neurons to form functional circuits. In the brain we find distinct regions with specific network configurations that form circuits that do specific tasks

While the individual neuron acts as the computational unit, complex behavior emerges only when these units are organized into functional circuits. The brain is not a random web of connections; it is structured into specific architectural "motifs" that appear repeatedly across different cortical areas. Understanding these motifs is essential for designing neuromorphic architectures that go beyond simple feed-forward processing.

The most fundamental distinction in neural circuitry is between Feed-Forward and Recurrent connections.

Feed-Forward: In sensory areas (like the early visual cortex), information flows unidirectionally from input to output. This allows for rapid, reflex-like feature extraction.

Recurrent: In higher cognitive areas, neurons form feedback loops, connecting back to themselves or their neighbors. This recurrence introduces a time component to the computation, allowing the network to maintain a "state" or short-term memory even after the input stimulus has vanished.

- Lateral Inhibition and Winner-Take-All

A ubiquitous circuit motif in the brain is Lateral Inhibition. In this configuration, an active neuron excites itself while inhibiting its neighbors via interneurons. This competition results in a "Winner-Take-All" (WTA) dynamic, where the neuron with the strongest input suppresses all others.

In neuromorphic engineering, WTA circuits are crucial. They provide a mechanism for noise reduction and categorical decision-making (e.g., deciding which specific pixel is the "edge") without requiring a central processor to compare values. The decision emerges physically from the circuit's competition.

ATTRACTOR DYNAMICS

Longer patterns require a latched state such as neurons entering repeated firing like a state machine Some neurons have other properties like bursting modes or continuous firing once the threshold has been reached. Inhibition should make a neuron not fire. When recurrent circuits interact with inhibition, stable patterns of activity called Attractors emerge. These are specific states toward which the system naturally evolves, acting as the neural basis for memory and spatial orientation.

Point Attractors: The network settles into a specific static firing pattern. This is analogous to an energy minimum in physics and is believed to be the mechanism behind associative memory (e.g., recognizing a face from a partial clue).

Continuous Attractors (Line/Ring): Instead of settling to a single point, the network maintains a bump of activity that can move continuously along a manifold.

Ring Attractors: Found in the Drosophila (fruit fly) heading-direction system. A "bump" of activity moves around a ring of neurons to represent the fly's compass heading.

Line Attractors: Used in the oculomotor system to maintain eye position.

These dynamics highlight a major shift from standard digital logic. In a digital memory, data is static and must be read; in an attractor network, the memory is an active, self-sustaining process.

EXCITATORY-INHIBITORY BALANCE

Finally, for any of these dynamics to function, the network must maintain a precise Excitation-Inhibition (E/I) Balance. The brain operates at a critical point of instability.

- Too much excitation leads to runaway feedback (similar to an epileptic seizure).
- Too much inhibition leads to signal death (quiescence).

Modern neuromorphic chips increasingly implement "homeostatic plasticity"—algorithms that automatically adjust synaptic weights to keep the network in this critical regime, ensuring that signals can propagate through deep layers without fading out or exploding.

MACRO ARCHITECTURES

Moving beyond local circuits, the neocortex—the seat of higher intelligence—displays a remarkable uniformity in its structure. In 1957, Vernon Mountcastle discovered that the cortex is organized into vertical structures called Cortical Columns.

A cortical column is a cylinder of tissue approximately 0.5 mm in diameter containing roughly 6 distinct layers of neurons. It acts as the fundamental "functional unit" or "microchip" of the brain. Whether the cortex is processing visual input, auditory signals, or motor commands, the internal circuitry of the column remains largely the same.

This suggests a profound principle for neuromorphic engineering: the brain uses a single, universal algorithm to solve many different problems. If we can reverse-engineer the operation of a single column, we can potentially scale that architecture to solve any modality of data.

THE THOUSAND BRAINS THEORY

Building on the concept of the cortical column, Jeff Hawkins (founder of Numenta) proposed the Thousand Brains Theory of Intelligence. This theory challenges the traditional view that the brain builds a single, global model of the world.

Instead, Hawkins argues that every single cortical column learns a complete, independent model of the object it is sensing (a "reference frame").

Distributed Modeling: If you touch a coffee cup with five fingers, your brain creates five separate models of the cup simultaneously.

Voting: These columns communicate laterally to vote on the object's identity. If one column thinks it feels a "curved handle" and another feels "warm ceramic," they reach a consensus that the object is a "Coffee Cup."

This theory is highly influential in modern neuromorphic design because it implies that intelligence is massively parallel and decentralized. It moves away from the fragile, deep hierarchies of Deep Learning toward robust, distributed consensus networks.

THE VISUAL HIERARCHY

While columns provide the local processing power, they are arranged globally into a hierarchy. The most studied example—and the blueprint for almost all modern computer vision—is the Primate Visual Cortex.

Information from the retina travels to the Primary Visual Cortex (V1) at the back of the brain. From there, it splits into two distinct pathways:

The Ventral Stream ("What" Pathway): Travels to the temporal lobe. It processes object identity. V1 detects edges; V2 detects textures; V4 detects shapes; and the Inferior Temporal (IT) cortex detects full objects (like faces).

The Dorsal Stream ("Where" Pathway): Travels to the parietal lobe. It processes spatial location and movement, guiding motor actions (e.g., reaching for a glass).

Neuromorphic cameras (like the Silicon Retina mentioned earlier) are often paired with spiking neural networks designed to mimic this specific split, separating the "flow" of motion from the "recognition" of objects to process dynamic scenes efficiently.

ASSOCIATIVE MEMORY

In this biological architecture, memory is not stored in a separate "hard drive" or register. Instead, memory is Associative and Content-Addressable.

Memory is stored physically in the synaptic weights between neurons. To retrieve a memory, the brain does not look up an address; it re-activates the specific pattern of neurons associated with that memory. Because of the attractor dynamics mentioned earlier, the brain allows for Pattern Completion: if a network receives a noisy or partial input (e.g., seeing half of a familiar face), the energy landscape of the network naturally "slides" the activity back into the learned state of the full face. This robustness to missing data is a key advantage of neuromorphic memory architectures over traditional RAM.

Write about dynamical models and hoph bifucations, write about modes of firing depending on bifurcations ...

2.2.6 • BIOLOGICAL LEARNING

This different model of computation requires a different model of learning. Deep learning's backpropagation is a "global" algorithm. To update a synapse in the first layer, you need an error signal calculated at the very last layer and propagated all the way back. This is highly effective but biologically implausible; there is no known mechanism for such a precise, "backward" error signal in the brain.

Instead, the brain appears to use local learning rules. The most famous is Hebb's rule: "Neurons that fire together, wire together."

A modern, measurable version of this principle is Spike-Timing-Dependent Plasticity (STDP). STDP is a learning rule that adjusts the strength (the "weight") of a synapse based purely on the relative timing of spikes between the pre-synaptic neuron (the sender) and the post-synaptic neuron (the receiver).

The rule is simple and local:

- LTP (Long-Term Potentiation): If the pre-synaptic neuron fires just before the post-synaptic neuron (meaning it likely contributed to the firing), the connection between them is strengthened.
- LTD (Long-Term Depression): If the pre-synaptic neuron fires just after the post-synaptic neuron (meaning it fired too late and did not contribute), the connection is weakened.

This mechanism allows the network to learn correlations, causal relationships, and temporal patterns directly from the stream of incoming spikes, without any "supervisor" or global error signal. It is the biological alternative to backpropagation and a cornerstone of modern neuromorphic learning.

- no evidence that brain does not use global error signal like backpropagation uses
- each neuron weight gets updated by a local rule
- spikes are discrete and does not have a gradient

While spiking neural networks offer greater biological plausibility and potential advantages in processing temporal information and energy efficiency, their adoption faces significant challenges, primarily stemming from the nature of their core computational element: the discrete spike.

A cornerstone of the success of modern deep learning, particularly with Multi-Layer Perceptrons (MLPs) and related architectures, is the backpropagation algorithm. Backpropagation relies fundamentally on the network's components being differentiable; specifically, the activation functions mapping a neuron's weighted input sum to its output must have a well-defined gradient. This allows the chain rule of calculus to efficiently compute how small changes in network weights affect the final output error, enabling effective gradient-based optimization (like Stochastic Gradient Descent and its variants). These techniques have proven exceptionally powerful for training deep networks on large datasets.

However, when we transition from the continuous-valued, rate-coded signals typical of MLPs to the binary, event-based spikes used in SNNs, this differentiability is lost. The spiking mechanism itself—where a neuron fires an all-or-none spike only when its internal state (e.g., membrane potential) crosses a threshold—is inherently discontinuous. Mathematically, this firing decision is often represented by a step function (like the Heaviside step function), whose derivative is zero almost everywhere and undefined (or infinite) at the threshold.

2.3 • MACHINE LEARNING TECHNICALITIES

In this section we go trough the techincal details behind deep learing and neuromorphic engineering

An important point is to declare whether a mechanism is bio plausible. An engineer might not care wether or not a mechanism is used by the brain and is crucial to the brains function. An engineer might only care about if the mechanics works and is effective for the system that is created in the engineers vision. An engineer might just use the brain as an inspiration. Evolution althoug achieved remarkable feats is not guaranteed to foster up the most optimal solution, only good enough to survive to the next generation. However discussing wether or not a mechanism is bio plausible is still useful for understanding our own brain. And creating bio plausible artificial systems can contribute to more than one field.

2.3.1 • FORWARD AND BACKWARD PROPAGATION

To understand the mechanism of modern neural networks, we must first formalize the objective. The goal of a neural network is to approximate a function $f * (\boldsymbol{x})$ by defining a parameterized mapping $y = f(\boldsymbol{x}\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the learnable parameters (weights and biases).

The Forward Pass The process begins with the forward pass, where information flows from the input layer through the hidden layers to the output. Mathematically, a deep network is a composition of non-linear functions. For a network with L layers, the activation of the l-th layer, denoted as $\boldsymbol{a}(l)$, is computed as:

$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{a}_{l-1} + \boldsymbol{b}_l$$

$$\boldsymbol{a}^{(l)} = \sigma\left(\boldsymbol{z}^{(l)}\right)$$

Here, $\boldsymbol{W}$ is the weight matrix, $\boldsymbol{b}$ is the bias vector, and $\sigma(\cdot)$ is the non-linear activation function. This recursion continues until the final output $\hat{\boldsymbol{y}}$ is produced.

The Loss Function To evaluate the quality of this prediction, we introduce a Loss Function (or Cost Function), denoted as $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$. This scalar function quantifies the divergence between the network's prediction $\hat{\boldsymbol{y}}$ and the ground truth $\boldsymbol{y}$. Common examples include Mean Squared Error (MSE) for regression or Cross-Entropy for classification. The training objective is to minimize this loss with respect to the parameters:

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}(f(\boldsymbol{x}\boldsymbol{\theta})\boldsymbol{y})$$

Backpropagation and Automatic Differentiation Since the parameters bold(theta) determine the output, they are responsible for the error. To minimize the loss, we must calculate the contribution of every single parameter to that error. This requires computing the gradient of the loss function with respect to every weight: $\nabla W \mathcal{L}$.

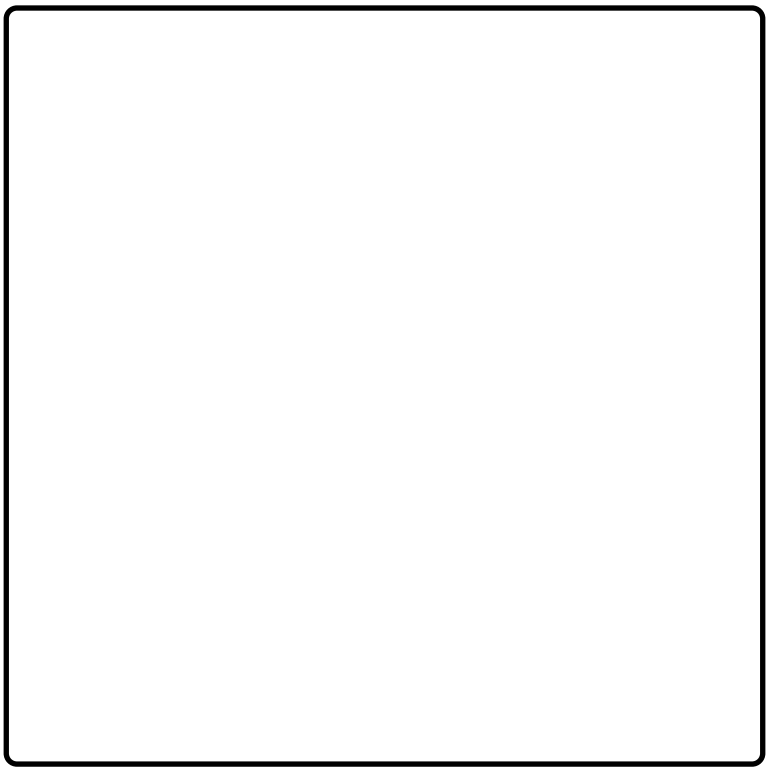
Figure 11: Gradient decent

Because the network is a nested composite function, we utilize the Chain Rule of calculus. This process is known as Backpropagation. We compute the gradients starting from the output layer and propagate them backward toward the input. For a specific weight w connecting neuron j to i in layer l, the gradient is decomposed as:

$$\frac{\partial \mathcal{L}}{\partial w_{ij}} = \frac{\partial \mathcal{L}}{\partial a_i} \cdot \frac{\partial a_i}{\partial z_i} \cdot \frac{\partial z_i}{\partial w_{ij}}$$

In modern deep learning frameworks, this is implemented via Automatic Differentiation (AutoDiff). Unlike symbolic differentiation (which can lead to expression swell) or numerical differentiation (which is computationally expensive and prone to approximation errors), AutoDiff constructs a computational graph of elementary operations during the forward pass. It then traverses this graph in reverse to compute exact gradients efficiently.

Gradient Dynamics and Activation Functions The choice of the activation function $\sigma(\cdot)$ is critical not only for the network's expressivity but for its trainability. The "Universal Approximation Theorem" states that a feedforward network with a single hidden layer containing a finite number of neurons can approximate any continuous function, provided the activation function is non-linear [1].

However, the specific choice of non-linearity dictates the stability of the gradients. During backpropagation, gradients are multiplied primarily by the derivative of the activation function $\sigma(z)$.

Figure 12: Common activation functions. The Sigmoid function (left) saturates at the extremes, causing gradients to vanish. The ReLU function (right) maintains a constant gradient for positive inputs.

If we utilize the traditional Sigmoid or Tanh functions, the derivative is strictly less than 1 (max 0.25 for Sigmoid). As we multiply these small numbers across many layers, the gradient exponentially decays toward zero. This is known as the Vanishing Gradient Problem, effectively preventing deep networks from learning. Conversely, if weights are initialized poorly, gradients can accumulate to become infinitely large, known as Gradient Explosion.

To mitigate this, modern networks often employ the Rectified Linear Unit (ReLU), defined as f(x)=max(0,x). Since the derivative of ReLU is either 0 or 1, it preserves the gradient magnitude for active neurons, allowing for the successful training of significantly deeper architectures.

$$\Delta w = \begin{cases} A^{+} \exp\left(-\Delta \frac{t}{\tau^{+}}\right) \text{ if } \Delta t > 0 \\ -A^{-} \exp\left(\Delta \frac{t}{\tau^{-}}\right) \text{ if } \Delta t < 0 \end{cases}$$

Equation 7: Standard weight update rule for STDP.

Where $\Delta t = t_{\text{post}} - t_{\text{pre}}$. The nwtwork architecthure should be paralizeable allowing for each neuron to operate independenly with only a local influence from other neurons, some neurons can act as hubs allowing clusters with specialzed tasks to communicate.

Forward and backwards propagation works in similar ways for a Spiking Neural Network (SNN) depending on the implementation the main difference and what gives SNN an edge is that the propagation is often asynchronos saving energy, the propagation of information is also sparse in comparison to ANN only events gets propagated when they happen zero values are not propageted as they are treated as an absens of an event

In memory computing Have to be parallell Add von Neumann and in memory computing diagrams

In contrast to the brain which has a "compute-in-memory" architecture; synapses store memory and perform processing in the exact same physical location. Modern computers, are built on the Von Neumann architecture, which physically separates the Processing Unit (CPU/GPU) from the Memory (RAM). For Deep Learning, this is catastrophic. A neural network is essentially a massive collection of weights (memory) that must be multiplied by inputs (compute). To run a modern LLM, the hardware must constantly shuttle billions of weights back and forth between the memory chips and the processor cores for every single token generated.

2.3.2 • KEY CONCEPTS IN NEUROMORPHIC ENGINEERING

From this foundation, the field of Neuromorphic Engineering was established on three core principles that directly oppose the Von Neumann architecture:

- Analog Computation:

Summing two numbers is verry effecint to do with analog computers, you simply add two currents. If you can

- Asynchronous Communication:

Removing the global clock. Parts of the chip only operate when there is data to process, eliminating the massive power drain of "clock distribution" seen in modern CPUs.

- Co-location of Memory and Compute:

Eliminating the separation between the processor and the RAM. In a neuromorphic chip, the "weight" of a neural connection is stored physically within the circuit that does the processing, effectively destroying the Von Neumann bottleneck. Why did it not take off immediately? Despite the brilliance of Mead's insight, neuromorphic computing remained a niche academic curiosity for decades. The reason was economic: Moore's Law. For thirty years, it was simply cheaper and easier to simulate neural networks on rapidly improving, general-purpose digital CPUs than it was to design difficult, noisy, custom analog hardware. However, as discussed in the previous chapter, Moore's Law is now slowing, and the energy demands of Deep Learning are exploding. The "free lunch" of digital scaling is over. This has forced the AI community to look back at Mead's original vision. We are now seeing a renaissance of these ideas, evolving from purely analog circuits to modern Spiking Neural Networks (SNN) implemented on digital-neuromorphic hybrids like Intel's Loihi and IBM's TrueNorth.

2.3.3 • BRIDGE

Consequently, standard backpropagation cannot be directly applied to SNNs. Gradients calculated using the chain rule become zero or undefined at the spiking neurons, preventing error signals from flowing backward through the network to update the weights effectively. This incompatibility represents a substantial obstacle, as it seemingly precludes the use of the highly successful and well-understood gradient-based optimization toolkit that underpins much of modern AI.

Surrogate Gradients: A popular approach involves using a "surrogate" function during the backward pass of training. While the forward pass uses the discontinuous spike generation, the backward pass replaces the step function's derivative with a smooth, differentiable approximation (e.g., a fast sigmoid or a clipped linear function). This allows backpropagation-like algorithms (often termed "spatio-temporal backpropagation" or similar) to estimate gradients and train deep SNNs, albeit with approximations.

2.3.4 • HARDWARE

Figure 13: Von Neuman

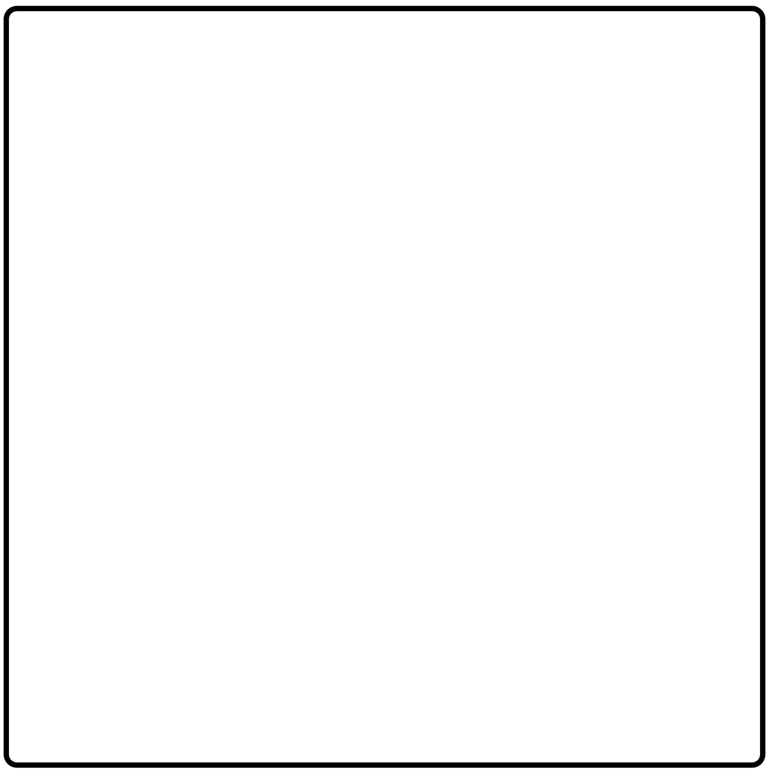
Figure 14: In-memory

2.4 • NEUROMORPHIC STATE OF THE ART

Disentangling core computational mechanisms from biological implementation details is a major ongoing challenge in neuroscience and neuromorphic engineering. Some complex molecular processes might be essential for learning or adaptation, while others might primarily serve metabolic or structural roles not directly involved in the instantaneous computation being modeled. The principles of neuromorphic computing, born from Carver Mead's vision and informed by modern neuroscience, have matured from theoretical concepts into a vibrant field of applied research. This progress is best seen in two key areas: the development of specialized, brain-inspired hardware and the creation of sophisticated software frameworks for simulating and deploying spiking neural networks (SNNs).

2.4.1 • APPLIED NEURAL CODES

A central challenge in neuromorphic engineering is encoding information into spikes. The most "standard" method is Rate Coding (frequency of spikes = intensity), but this is slow and energy-inefficient. To solve this, Simon Thorpe proposed Rank Order Coding. Thorpe observed that the human visual system processes images far too quickly for neurons to average spikes over time. Instead, he proposed that information is encoded in the order of firing. The most strongly activated neurons fire first.

The "N-of-M" Strategy: To implement this efficiently in hardware, engineers often use an "N-of-M" coding scheme.

- M (Population): A large pool of potential neurons (e.g., 1000).
- N (Active): Only the first N neurons (e.g., 50) to spike are transmitted.
- Mechanism: Once N spikes are received, the system inhibits the rest. This guarantees extreme sparsity (low power) and filters out "noise" (late spikes).

This approach transforms time into a priority queue. A downstream neuron does not wait for a "frame" to finish; it begins computing as soon as the earliest, most salient spikes arrive.

2.4.2 • NEUROMORPHIC LEARNING

The most biologically plausible learning algorithm is Spike-Timing-Dependent Plasticity (STDP). Unlike Deep Learning, which updates weights based on a global error calculated at the output, STDP updates weights based on local causality between two connected neurons.

Causal (LTP): If the input spike (pre) arrives before the output spike (post), the synapse is strengthened (Long-Term Potentiation). The input "caused" the firing.

Acausal (LTD): If the input spike arrives after the output spike, the synapse is weakened (Long-Term Depression). The input was irrelevant to the decision.

STDP allows networks to self-organize and detect repeating patterns in data without labeled supervision. However, on its own, it struggles to reach the high accuracy of modern supervised classifiers.

SURROGATE GRADIENTS

To achieve “State-of-the-Art” (SOTA) performance on complex tasks (like ImageNet or Speech Recognition), modern neuromorphic engineers often hybridize biology with Deep Learning.

The core problem is that spikes are non-differentiable (a step function has zero derivative everywhere), which breaks standard Backpropagation. The solution is Surrogate Gradient Learning.

Forward Pass: The hardware uses the true, crisp spiking physics (non-differentiable).

Backward Pass: The learning algorithm substitutes the spike with a smooth “surrogate” function (like a sigmoid) to calculate gradients.

This allows us to train SNNs using powerful optimizers (like Adam) and frameworks (like PyTorch), transferring the trained weights to the neuromorphic chip for efficient inference.

2.4.3 • COMPLEX DYNAMICAL MODELS

Neuromorphic is not just about making processors but also about building machines and models to better understand the biology

Beyond these standard algorithms, there is a rich landscape of more complex theoretical models. While we will not utilize them in this specific implementation, they represent the frontier of the field:

Reservoir Computing (Liquid State Machines): Using a chaotic, randomly connected “tank” of neurons to project inputs into high-dimensional space before a simple readout layer.

Equilibrium Propagation: A physics-based learning rule that relaxes a network to an energy minimum, avoiding the need for a separate backward pass.

Attractor Networks: Recurrent networks (Ring or Line attractors) that maintain stable states (memories) even in the absence of input, crucial for spatial navigation and working memory.

2.4.4 • NEUROMORPHIC SENSORS

2.1. The Paradigm Shift: From Frame-Based to Event-Based Sensing

Traditional sensory acquisition systems, particularly in computer vision and audio processing, have historically relied on the von Neumann architecture's separation of memory and processing, coupled with a clock-driven sampling approach. In this conventional paradigm, sensors capture the state of the entire environment at fixed temporal intervals—generating frames or samples regardless of the scene's activity. While effective for static analysis, this method generates redundant data streams that impose significant latency, bandwidth, and power consumption penalties, particularly in high-speed or sparse-signal environments.

In contrast, neuromorphic engineering, a field pioneered by Carver Mead in the late 1980s, seeks to emulate the biological principles of the mammalian nervous system. Neuromorphic sensors abandon the global shutter and fixed clock in favor of asynchronous, event-driven acquisition. Information is encoded not as absolute values at a fixed rate, but as a sparse stream of "events" or "spikes" triggered only when a significant change in the physical stimulus occurs. This bio-inspired approach offers theoretical and practical advantages in terms of temporal resolution, dynamic range, and energy efficiency (pJ/event), effectively shifting the computational burden from raw data processing to sparse event management.

2.2. Neuromorphic Vision Sensors (DVS)

The most mature realization of this paradigm is the Dynamic Vision Sensor (DVS), often referred to as the event camera. Unlike standard Active Pixel Sensors (APS) that integrate photon counts over a fixed exposure time, each pixel in a DVS operates independently and asynchronously.

The fundamental operation of a DVS pixel involves continuous monitoring of the log-intensity of the incident light. An event e is generated at time t when the change in logarithmic intensity exceeds a preset threshold θ:

$\Delta \ln(I)=\ln(I(t))-\ln(I(t-\Delta t))\geq\pm\theta$

where I(t) is the photocurrent at time t. This mechanism yields a stream of events characterized by a tuple (x,y,t,p), representing the pixel coordinates, the microsecond-resolution timestamp, and the polarity of the brightness change (ON or OFF).

Recent literature highlights three key advantages of this architecture:

Temporal Resolution: Event cameras achieve effective frame rates equivalent to several kilohertz, with latencies in the microsecond range, making them ideal for high-speed robotics and ballistics tracking.

Dynamic Range (DR): Because individual pixels do not share a global exposure time, DVS pixels do not saturate easily. Modern neuromorphic sensors boast dynamic ranges exceeding 120 dB, compared to the ~60 dB typical of standard industrial cameras, allowing robust operation in scenes with simultaneous extreme brightness and darkness.

Data Sparsity: In static scenes, a DVS generates near-zero output, drastically reducing power consumption and downstream processing requirements compared to frame-based cameras that output constant data regardless of content.

2.3. Auditory and Tactile Modalities

While vision sensors dominate the field, the principles of neuromorphic sensing extend to other modalities, mirroring the specialized transduction mechanisms of the biological cochlea and skin.

2.3.1. Silicon Cochlea

Neuromorphic auditory sensors, or "silicon cochleas," emulate the basilar membrane's hydrodynamics. Instead of performing a global Fourier Transform (FFT) on a sampled audio waveform, these sensors utilize a cascade of analog band-pass filters. Each filter channel operates independently, generating spikes when the energy in its specific frequency band exceeds a threshold. This architecture provides high-fidelity temporal information crucial for tasks such as sound source localization and separation, often achieving lower latency and power consumption than DSP-based solutions. 2.3.2. Neuromorphic Tactile Sensing

Electronic skins (e-skins) and neuromorphic tactile sensors are an emerging frontier, designed to provide robots with high-frequency feedback for manipulation tasks. Recent advances utilize triboelectric, piezoelectric, or piezoresistive materials coupled with event-based readout circuits. These sensors encode pressure, vibration, and shear force changes as asynchronous spikes, allowing for the rapid detection of slip events—a critical capability for stable grasping that mimics the fast-adapting mechanoreceptors (e.g., Meissner corpuscles) in human skin.

2.4. Processing Events: Spiking Neural Networks (SNNs)

The asynchronous nature of neuromorphic sensors necessitates a departure from standard Convolutional Neural Networks (CNNs), which are optimized for dense, synchronous matrix multiplications. The natural processing counterpart for event streams is the Spiking Neural Network (SNN).

In an SNN, artificial neurons maintain an internal membrane potential that integrates incoming spikes over time; the neuron "fires" an output spike only when this potential crosses a threshold. This compatibility has driven the development of specialized neuromorphic hardware accelerators, such as Intel's Loihi and SynSense's Dynap-CNN, which support massive parallelism and local synaptic plasticity. These chips enable edge-native learning and inference with power budgets in the milliwatt range, significantly lower than standard GPU-accelerated embedded systems

2.4.5 • NEUROMORPHIC COMPUTERS

The primary goal of neuromorphic hardware is to escape the von Neumann bottleneck and emulate the power efficiency and massive parallelism of the brain. Two landmark systems define the state of the art:

IBM TrueNorth: A prominent early example, TrueNorth is a fully digital, real-time, event-driven chip. It consists of 4,096 "neurosynaptic cores," collectively housing one million digital neurons and 256 million synapses. Its architecture is explicitly non-von Neumann; processing and memory are tightly integrated within each core. TrueNorth's key achievement is its staggering power efficiency: it can perform complex SNN inference tasks (like real-time video object detection) while consuming only tens of milliwatts—orders of magnitude less than a CPU or GPU performing a similar task. However, its architecture is largely fixed, making it a powerful "inference engine" but less flexible for researching novel, on-chip learning rules.

Intel Loihi (and Loihi 2): Intel's line of neuromorphic research chips, starting with Loihi in 2017, represents a significant step towards flexible, on-chip learning. Like TrueNorth, Loihi is an asynchronous, event-driven digital chip, but with a key difference: it features programmable "learning engines" within each of its 128 neuromorphic cores. This allows researchers to implement and test dynamic learning rules, such as STDP and its variants, directly on the hardware in real-time. The second generation, Loihi 2, further refines this with greater scalability, improved performance, and more advanced, programmable neuron models, positioning it as a leading platform for cutting-edge neuromorphic algorithm research.

Neuromorphic sensors ...

2.4.6 • SIMULATION AND SOFTWARE FRAMEWORKS

Before algorithms can be deployed on specialized hardware, they must be designed, tested, and validated. This is the role of SNN simulators, which function as the "TensorFlow" or "PyTorch" of the neuromorphic world.

Brian: A highly flexible and popular SNN simulator used extensively in the computational neuroscience community. Its strength lies in its intuitive syntax, which allows researchers to define neuron models and network rules directly as a set of mathematical equations (e.g., the differential equations of a Leaky Integrate-and-Fire neuron). This makes it an ideal tool for exploring the detailed dynamics of biologically realistic models.

Nengo: A powerful, high-level framework that functions as a "neural compiler." Nengo is built on a strong theoretical foundation (the Neural Engineering Framework) that allows users to define complex computations and dynamical systems in high-level Python code. Nengo then "compiles" this functional description into an equivalent SNN. Its key advantage is its backend-agnostic nature; the same Nengo-defined network can be run on a standard CPU, a GPU, or deployed directly to neuromorphic hardware like Intel's Loihi.

2.5 • RESEARCH GAPS

Despite this immense progress in hardware and software, a fundamental challenge remains, creating a critical research gap: the training problem.

Mainstream deep learning has a powerful, universal tool: backpropagation. Neuromorphic computing does not yet have a clear equivalent. While these systems exist, they still struggle with finding an efficient, scalable, and powerful learning algorithm that is both biologically plausible and computationally effective. This gap manifests in several ways:

- Limited Supervised Learning

"Local" rules like STDP are fundamentally unsupervised. They are excellent for finding patterns and correlations but struggle with complex, task-driven "supervised" problems (e.g., "classify this audio signal into one of ten specific words").

- The Conversion Compromise

A popular workaround is to first train a conventional, non-spiking ANN using backpropagation, and then "convert" its weights to an SNN for efficient inference. This method, while practical, is a compromise. It discards the rich temporal dynamics SNNs are capable of and does not represent true "neuromorphic learning."

- The Surrogate Gradient Challenge

The firing of a spiking neuron is a non-differentiable event, which makes it incompatible with standard backpropagation. New methods, like "surrogate gradient" learning, attempt to approximate this spike event with a smooth function to enable gradient-based learning, but this is an area of intense and ongoing research.

This thesis confronts this central challenge: How to effectively and efficiently train spiking neural networks for complex, real-world temporal tasks. While hardware like Loihi provides the platform for on-chip learning, it still requires a robust and scalable algorithm. The existing approaches of ANN-to-SNN conversion or simple Hebbian rules are insufficient.

This thesis proposes a method to bridge this gap by... (...for example: "...developing a novel, event-driven surrogate gradient algorithm capable of training deep SNNs directly in the temporal domain," or "...introducing a hybrid learning rule that combines the efficiency of STDP with the task-driven power of error-based feedback," or "...proposing a new architecture for temporal credit assignment that is both hardware-friendly and scalable.")

3 • METHOD

This chapter details the computational framework developed to address the fundamental limitations of standard deep learning described in Section 2. The approach presented here aligns with the constraints of biological substrates: sparsity, asynchrony, and locality. We propose a neuromorphic architecture designed to maximize energy efficiency and computational robustness through three core mechanisms: Temporal Information Representation: A coding scheme that translates static data into asynchronous spike latencies, replacing energy-intensive rate coding with sparse time-to-first-spike (TTFS) interactions.

Simple well established highly parallazable integrate and fire neurons.

Local Plasticity Algorithms: A two-stage learning process combining structural synaptogenesis for dynamic connectivity and a localized, prediction-based weight update rule for pattern recognition. Finally, we describe the custom event-driven simulation environment constructed to validate these principles, bridging the gap between theoretical biological models and practical execution on standard digital hardware.

The methods in this thesis will be tested on visual number reckognition

3.1 • INFORMATION REPRESENTATION

The choise of a coding shceme puts key constraints on the design any processing system as it lays the founation for the flow of information. In Section 2.2.4 and Section 2.4.1 we dicussed several candidate neural codes that are both biologically plausible and have been used in previous neuromorphic systems. The neural code plays a large role in determening the effecieny of the system. Many neuromorphic systems use rate code as it is easy to translate values it is straight forward. any float value can be encoded as a rate code and integrate and fire neurons work well with this encoding. An IF neuron using rate code can reduce the multiply and add operations with just add operations. each spike arriving at a synapse adds its weight to the total. It is easy to see that for a rate code a higher rate input means more of that input contributing to the total sum. The main reason for why this thesis will not use a rate code is that it is relativly ineficient and slow compared to a temporal code. In a temporal code only one spike is required to establish its value in relation to other synapses, making a single event much more information dense. This is especially important for systems that do not have the connection density of a biological brain and must use shared buses that can be congested if too many spikes are present on the bus. Secondly to determine the value of a rate code you have to take an average of multiple spikes, imposing a delay. Using a TTFS encoding requires more sophisticated neuron models than a simple integrate and fire, it need more bookkeeping to keep track of the relative arival of incoming spikes.

That beeing said and as discussed in Section 2.2.4 combining neural codes and using the right one for the job is also an important aspect to consider, there is not a clear consenus on which code is better it is a matter of the task at hand. As we also have talked about, Combining differnt codes toghether can make for a powerful representation of information. This thesis focuses on visaul recognition and a TTFS combined with population code is the representation of choise. The popultion code will represent a distinct pattern in an image. This can be set up to detect lines and primitve shapes, deeper layers will use populations of primitve shapes to detect more complex features like circles and boxes and so on.

This thesis will use TTFS combined with population coding for visual tasks, This combination is biologically plausibe as it is fast and can be set up as filters similar to how the human visual cortex is structured.

A neuron should have the following charecteristcs:

- Acuumulate spikes in some form (integrate)
- Fire when some treshold has been reached
- Leak the potential over time

A neural code should

- be fast
- be effecient in terms of neurons used
- able to encode a wide range of stimuli
- robust to noise

Explain in detail how we encode information using neural codes using pseudo code

Code Rate code is simple will not use it Temporal codes

3.1.1 • CONVERTING SENSORY INPUT

```
procedure intensity_to_delay_encoding(image, T_max=100, T_min=0):
    normalized_image = normalize(image)
    spike_times = T_max - (T_max - T_min) * normalized_image
    return spike_times
```

Algorithm 1: Linear intensity to dely encoding

```
procedure intensity_to_delay_encoding(image, T_max=100, T_min=0):
    normalized_image = normalize(image)
    spike_times = T_max - (T_max - T_min) * normalized_image
    return spike_times
```

Algorithm 2: Logarithmic intensity to dely encoding

3.1.2 • REPRESENTING CONTRAST

easy with rate code

need global information to normalize

using ttf is difficult

using negative image - still problem with absolute contrast

can be done locally with speial sensors and using logarithmic intensity to delay

3.1.3 • DECODING

Decoding is done in the neurons and thus the neurons need to be designed to work with the chosen encoding either at individual neuron level or population level

```python
def integrate_and_fire(excitatory, inhibitory, threshold=1.0):
    excitatory_spikes = [(t, 1) for t in excitatory.flatten() if not
np.isnan(t)]
    if inhibitory is None:
        inhibitory_spikes = []
    else:
        inhibitory_spikes = [(t, -1) for t in inhibitory.flatten() if not
np.isnan(t)]
    all_spikes = excitatory_spikes + inhibitory_spikes
    if not all_spikes:
        return None
    all_spikes.sort(key=lambda x: x[0])
    integrated_potential = 0.0
    firing_time = None
    for time, spike_type in all_spikes:
        integrated_potential += spike_type
        integrated_potential = max(0, integrated_potential)
        if integrated_potential >= threshold:
            firing_time = time
            break
    return firing_time
```

How neurons use coding

We have seen that glif neuron is bio plausible and such a neuron is simple and can be realized easily on hardware the way. Using a temporal code the neuron has to differentiate between inputs in the time domain otherwise it cannot decode the information and cannot do useful work. Earlier inputs are encoded with larger values and should have a greater portion of summation going on inside the neuron.

-If order matters (temporal code) the neuron must handle it Evidence for bio-plauibility can be found from eq 1 where the R(u) is dependent on the membrane potential

For visual tasks which is what the application this thesis focuses on
1. a population code is used in conjunction with a temporal code
2. Population code is used alone (eg individual order within a population does not matter)

A counter starts when the first spike arrives

Needs a global reset or local

3.1.4 • PERCEPTRON EQUIVALENCE

The perceptron equation can be obtained with a ttfs using the inverse of firing times

$$T = \sum \frac{w}{t}$$

Equation 8: Weigthed sum

In a time to first spike scheme of we care about the order (the relative values since information is stored in time and order) we have to use weights and a neuron model that distinguish between inputs arriving earlier than others. I present a scheme where the first neuron that arrives starts a linear count where the slope of the counter is the weight additional inputs will increase or decrease the slope according to their weight. We can see that neurons arriving earlier will get more time to increase the counter and thus will carry a higher value. If the counter reaches a threshold the neuron will fire. The astute will notice that in this scheme the neuron will fire even for the smallest stimulus since the counter will count up a non zero value and eventually reach the threshold, to mitigate this we can simply say that if the counter is too slow the neuron will not fire we will see later that this scheme satisfies the criteria above.

The problem with this decoding is for strong stimuli we would ideally make the neuron respond immediately and fire, but it has to wait until the counter has reached the threshold to fix this we can also add the weight of the input directly to the potential while also starting a counter. Now if early strong inputs arrive they will fill up the potential and make the neuron fire almost immediately. Small inputs wil take some time

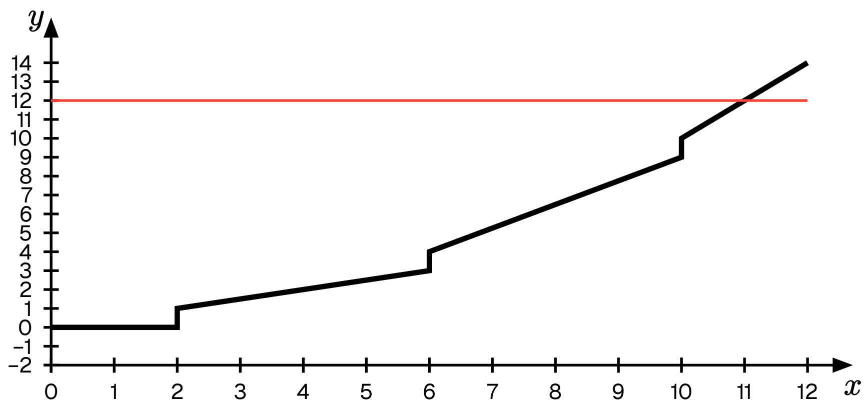

Figure 15: Proposed simplifed layout of a SNN. The neurons are connected with hirearcical busses that allow for the network to be configured as a *small world network*

Leaky integrate and fire models seem the best bet, however complex dynamics like exponential decay and analog weights and potentials seem excessive, we might do without. Binary weights 1 for excitatory and and 0 for inhibitory. Stronger weights can be modeled with multiple parallel synapses

Another way which is also based on relative firing order of single spikes could be a passcode encoding. Such an encoding could work by having neurons only react to a sequence. It has an internal state machine of sorts and will only advance to the next state if recives the correct input in the correct order. This encoding does only care about relative order not relative timings.

3.2 • PROPOSED LEARNING ALGORITHMS

If the post neuron fires then we should strengthen the weights. Synapses gets grown at random this might be inneficient but it is highly parallelizable an idea could be to grow synapses with a kind of gravity where post synapses has a pulling effect if they are already strong

The encoding fit for images are a kind of population code where a pattern comes at the same time. The relative timing between patterns is used for strength earlier patterns are stronger. In a inhibitory network this allows for winner take all

learning with binary weights

learning with more steps of quantized or continous weights

Say we want to detect the pattern ABC and the pattern ABD. First of all if the order does not matter set all the weights equal. If the order does matter the weights determine the order. Now if a neuron learns pattern ABC so well that it learns to fire on only AB then it can fire faster. However if a second neuron wants to learn ABD then inhibition from the AB neuron prohibits it. A solution can be that if a neuron originally learned ABC but now fires on AB but stil has a strong weight on C it should remember this and if it fires on AB but then C does not arrive it should be like “oh, C did not show maybe I am wrong to fire early” eg. Decrease weights for A and B It predicts!

A second way is to have a hierarchy with bypass. So one layer detects only AB then the next layer has bypass of the first layer and the second combining AB and C or D

A second problem is how to decode order. When do we start the decreasing timer, how fast, should it be in time or in amount of spikes, what to do with phase? The phase should correct itself. The weights need to be as presise as the timing of the spikes? Or we could make the neuron sensitivity proportional to its inverse potential and add leaking

```
start with a collection of neurons with arbitrary connections
if a pre-synaptic neuron fires then
    it has a chance to grow a synapse to a random post-synaptic neuron
if a post-synaptic neuron fires then
    strengthen all connections to pre-synaptic neruons that fired before
    remove connections to pre-synaptic neurons that did not fire or fired after
ⓘ a neuron can be both pre-synaptic and post-synaptic
```

Algorithm 3: Unsupervised local learning rule for induvidual neurons. Based on STDP

```
1 probability of growing a synapse is inversely
  proportional to the amount it already has
2 earlier firings should get a better chance to grow synapses,
  although this is regulated by inhibitory action
```

Algorithm 4: Growing rules for synapses

3.3 • SIMULATION

We have discussed the benefits of co algorithm design and designing new specialed computer hardware that run neuromorphic algorithms directly on the substrate via gates or analaog elements and exotic new materials. However developing biologically inspired computers and algorithms on cheap and avaliable CPU and GPU hardware is a great way to quicly iterate and test

Altough many simulation and software packages exists as outlines in Section 2.4.6. The methods in this thesis has been tested using custom simulaton software for full control

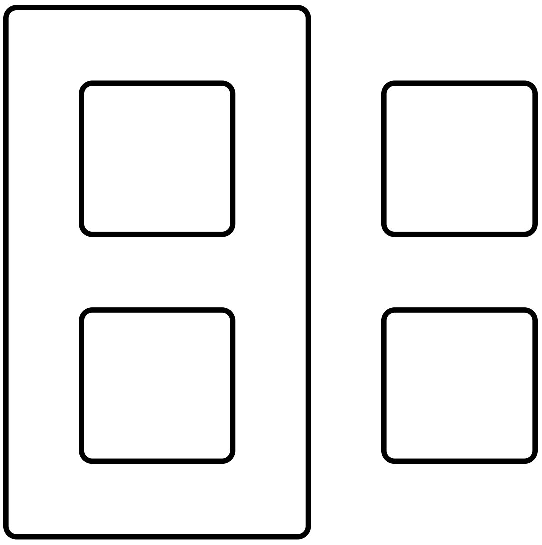

Figure 16: Simulator architechture block diagram

The simulator runs an event loop

Spikes are pushed to a heap

The simulator can run both on CPU and GPU,

When running on CPU the spikes are pushed to a heap, when running on GPU spikes are pushed to a adress event bus

the algorithms presented in this thesis are highly paralizable

It is built from the ground up using the Vulkan API

3.4 • METRICS

score - classification

power

FLOPS

4 • RESULTS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

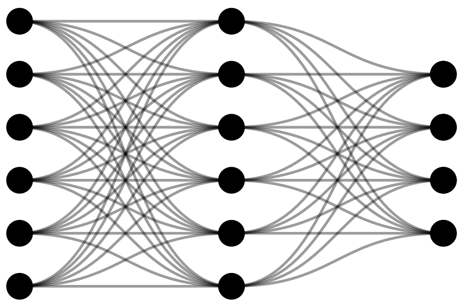
Figure 17: Neural network before learing

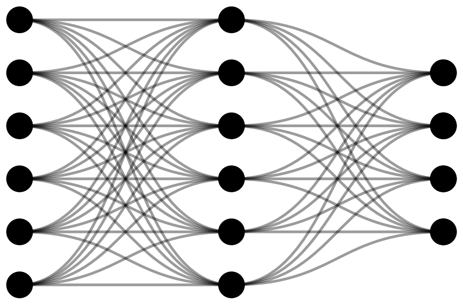
Figure 18: Neural network during learning

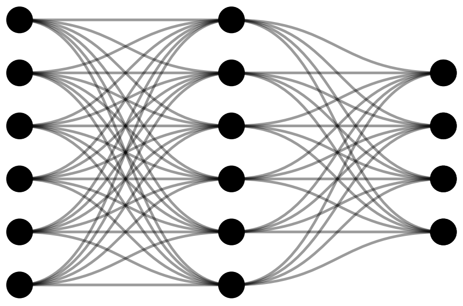
Figure 19: Neural network after learning

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

0	0	1	1
1	1	1	1

Table 1: Number of operations

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

5 • DISCUSSION

In this section we discuss the

We see a trade-off between the ability to learn and speed of learing and forgetting. Synaptic plasticitiy must be tuned in order for the right learing enviroment to form

representing contrast is difficult with a ttfs scheme, better to use special sensors and pass the encoded contrast in ttfs to a neuromophic processor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

6 • CONCLUSION

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

BIBLIOGRAPHY

[1] ?, “Placeholder,” 2025.