MACHINE LEARNING WITH SPIKES

Brage Wiseth
Universitiy of Oslo
bragewi@uio.no

2025-07-02

1 ~ INTRODUCTION AND THEORY

The concept of intelligence, how it arises … is probably been some of the longest standing questions in human history. How and if it can be reproduced artificially is a particuarly hot topic today. Getting answers to these questions will not only help us understand our own minds but also brings the promise of unlocking new technology discovering new drugs or materials, it may be the last invention humans ever need to make. In recent years we have crept ever closer to answer some of these questions. New state of the art artificial inteligence systems have achieved remarkable success like the sophisticated language capabilities of GPT models and the protein-folding predictions of AlphaFold. `TODO: needs citation`

Despite these triumphs, a significant gap persists between artificial systems and their biological counterparts. Evidently these AI systems might posses superhuman capabilities in one or a few domain none of them surpass humans in all, what we call Artificial General Inteligence (AGI). Also more relevant to this thesis is that current state-of-the-art ANNs, while functionally powerful, require vast amount of data, computatuon and energy resources. This demand stands in stark contrast to the biological brain—an extraordinarily complex and efficient organ estimated to operate on merely 20-30 Watts while also beeing in the AGI category. This profound difference in efficiency and capability suggests that contemporary ANN paradigms, might be missing or oversimplifying fundamental principles crucial for truly intelligent and scalable computation.

This thesis wants to explore new approaches that first and foremost might solve the critical limitations of scalability and energy efficiency in artificial intelligence. But also hopefully lay the foundation for systems that might eventually fall into the AGI category. This likely requires moving beyond current mainstream ANN architectures. It will explore the potential of incorporating more sophisticated biological principles into AI design. This involves investigating alternative computational paradigms, potentially inspired by mechanisms such as sparse, event-driven processing observed in Spiking Neural Networks (SNNs), the role of temporal dynamics in neural coding, or the potential computational advantages of systems operating near critical states. The central challenge lies in identifying and abstracting the truly essential biological mechanisms for intelligence and efficiency, distinguishing core principles from intricate biological details that may not be necessary for artificial implementation. Concretly this thesis wants to

- Explore how information flow based on sparse event might be implemented in a network, encoding
- Explore learning algorithms suitable for such a network, and challange SOTA

`TODO: add or remove research questions`

The different sections of the thesis

- A
- B
- C

1.1 ~ THE ESTASBLISHED APPROACH

[TODO: add MLP explanation and ilustration] The term Aritifical Inteligence forms an umbrella over many different techniques that make use of machines to do some intelligent task. The most promising way to acheive AI to day is trough deep neural networks. The neural networks of today almost exscluvely use some variation of the Multi Layer Perceptron (MLP) concept. It is a fairly old idea based on a simple model on how the brain processes information. The MLP evolved from early attempts to create computational models inspired by biological neurons. The model of the neuron that the MLP is based on has synapses just like the biological one, the synapses functions as inputs which when firing will exite the reciving neuron more or less depending on the strenth of the connection. If the reciving neuron get exited above a threshold it will fire and pass the signal downstream to another reciving neuron. Which is conceptually similar to how real neurons operate. Its roots lie in the foundational work of who proposed a simplified binary threshold model of a neuron, and perceptron, which introduced a learning rule for a single computational neuron capable of classifying linearly separable patterns. However, to the MLP was the understanding that stacking multiple layers of these perceptron-like units could overcome these limitations by creating more complex decision boundaries. The critical breakthrough enabling the practical use of MLPs was the independent development and subsequent popularization of the backpropagation algorithm in the 1970s and 1980s (with key work by Werbos, Parker, LeCun, and notably Rumelhart, Hinton, and Williams in 1986). Backpropagation provided an efficient method to calculate the gradient of the error function with respect to the network's weights, allowing for effective training of these deeper, multi-layered architectures. This combination—multiple layers of interconnected units, typically using non-linear activation functions, trained via backpropagation—defines the MLP, which became a foundational architecture for neural networks and paved the way for the deep learning revolution.

1.2 ~ PROBLEMS WITH THE ESTABLISHED APPROACH

backprop it requires freezing the entire network and separates computation and learning into two separate stages, local connectetions that should be independent of eachother have to wait

1.3 ~ NEUROSCIENCE 101

The perceptron, and its evolution into Multi-Layer Perceptrons (MLPs), represent foundational models in artificial intelligence inspired by early concepts of neural computation.

Indeed, certain core principles resonate with biological observations: the brain comprises interconnected neurons, often organized in broadly hierarchical structures or layers[1] that process information sequentially from sensory input to higher cognitive areas. Furthermore, individual neurons integrate incoming signals—analogous to a weighted sum in MLPs and generate an output spike or 'fire' only when a certain threshold is exceeded, a mechanism abstracted by the activation functions used in artificial neurons.

However, this abstraction, while powerful, significantly simplifies the underlying neurobiology. Decades of rigorous neuroscience research reveal that brain function emerges from complex electro-chemical and molecular dynamics far richer than the simple weighted sum and static activation. While it's crucial to discern which biological details are fundamental to computation versus those that are merely implementation specifics[2], moving beyond the standard MLP model is necessary to capture more sophisticated aspects of neural processing.

A primary departure lies in the nature of neural communication. Unlike the continuous-valued activations typically passed between layers in an MLP (often interpreted as representing average firing rates), biological neurons communicate primarily through discrete, stereotyped, all-or-none electrical events known as action potentials, or 'spikes'. Information in the brain is encoded not just in the rate of these spikes (rate coding), but critically also in their precise timing, relative delays, and synchronous firing across populations (temporal coding) (Gerstner et al., 2014). For instance, the relative timing of spikes arriving at a neuron can determine its response, allowing the brain to process temporal patterns with high fidelity – a capability less naturally captured by standard MLPs. Spikes can thus be seen as event-based signals carrying rich temporal information.

Furthermore, neural systems exhibit complex dynamics beyond simple feedforward processing. Evidence suggests that cortical networks may operate near a critical state, balanced at the 'edge of chaos,' a regime potentially optimal for information transmission, storage capacity, and computational power (Beggs & Plenz, 2003; Chialvo, 2010). Systems like the visual cortex demonstrate this complexity, where intricate patterns of spatio-temporal spiking activity underlie feature detection, object recognition, and dynamic processing (Hubel & Wiesel, 1962; Thorpe et al., 1996). These biologically observed principles—event-based communication, temporal coding, and complex network dynamics—motivate the exploration of Spiking Neural Networks (SNNs), which explicitly model individual spike events and their timing, offering a potentially more powerful and biologically plausible framework for computation than traditional MLPs.

1.4 ~ THE PROBLEM WITH SPIKES: DISCONTINUITY

While models like Spiking Neural Networks (SNNs) offer greater biological plausibility and potential advantages in processing temporal information and energy efficiency, their adop-

[1] While often conceptualized in layers (e.g., layers of the neocortex), the brain's connectivity is vastly more complex than typical feedforward ANNs, featuring extensive recurrent connections, feedback loops, and long-range projections that make a simple 'unrolling' into discrete layers an oversimplification

[2] 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.

tion 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 (Rumelhart et al., 1986). 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.

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.

1.5 ~ NEW APPROACHES

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) (Neftci et al., 2019; Zenke & Ganguli, 2018). This allows backpropagation-like algorithms (often termed "spatio-temporal backpropagation" or similar) to estimate gradients and train deep SNNs, albeit with approximations.

Bio-Inspired Local Learning Rules: Drawing inspiration from neuroscience, researchers explore learning rules based on local activity, such as Spike-Timing-Dependent Plasticity (STDP). STDP adjusts synaptic weights based on the relative timing of pre- and post-synaptic spikes (Gerstner et al., 1996; Bi & Poo, 1998). While biologically plausible and inherently suited to spike timing, purely local rules like STDP often struggle to match the performance of gradient-based methods on complex supervised learning tasks and can be harder to scale or direct towards a specific global objective. Hybrid approaches combining STDP with other mechanisms are also being investigated.

Conversion Methods: Another strategy involves training a conventional ANN (like an MLP or CNN) using standard backpropagation and then converting the trained network into an SNN (Cao et al., 2015; Diehl et al., 2015). This leverages the power of gradient-based training but may not fully exploit the unique temporal dynamics SNNs offer, and often requires careful parameter tuning during conversion.

Gradient-Free Optimization: Techniques like evolutionary algorithms or reinforcement learning can optimize SNN parameters without requiring explicit gradients, but they often suffer from lower sample efficiency and scalability issues compared to gradient descent, particularly for very large networks.

Therefore, while moving towards more biologically realistic, event-driven models like SNNs is conceptually appealing, overcoming the fundamental incompatibility with standard gradient-based optimization remains a critical area of active research and development. The success of SNNs in practice hinges significantly on the effectiveness and scalability of these alternative or adapted training techniques.

2 ~ *METHODOLOGY*

Step: explore learning algorithms
experiment 1 set up random pattern across 1 dimension, inject a repeating pattern for the network to learn

Step: explore edge detection
can be achived by duplicating and inverting the signal

Step: explore different neuronal models

Step: prove mathematical equivalence to other ANNs

Step: expand to 2d

Step: expand network to more layers figure out how to connect (AER)

Step: explore how learning algos work with multi-layer

Step: try to learn MNIST

NOTETOSELF: Steps either build on eachother or are independent, this way we can do bite size research and stop at any point if we run out of time, we dont have to make an entire system in order to have something to write about

```
do something
do something else
while still something to do
    do even more
    if not done yet then
        wait a bit
        resume working
    else
        go home
```

3 ~ *Results*

4 ~ Discussion

4Bibliography