Exploring training spiking neural network training methods for temporal data

Richard klotiņš

Supervisor: Dorian Florescu

March 31, 2025

Abstract

Filler abstract - Spiking Neural Networks (SNNs), the third generation of neural networks, offer a promising alternative to traditional Artificial Neural Networks (ANNs) due to their energy efficiency, temporal processing capabilities, and potential for neuromorphic hardware implementation. Unlike ANNs, which rely on continuous-valued activations, SNNs process information through discrete spike events, closely mimicking biological neural systems. This characteristic enables them to efficiently handle sequential data, making them suitable for applications such as speech recognition, event-based vision, and edge computing. Training SNNs, however, remains a significant challenge due to the non-differentiability of spike events. While ANN-to-SNN conversion provides a workaround, it imposes computational costs and limits architectural flexibility. Direct training methods, such as Backpropagation-Through-Time (BPTT) with surrogate gradients, local learning rules, and biologically inspired approaches like e-prop and EventProp, have emerged as viable alternatives. Each method presents trade-offs in terms of computational complexity, biological plausibility, and performance. Notably, EventProp has demonstrated state-of-the-art results while reducing memory and computational overhead compared to BPTT. This work explores the theoretical advantages of SNNs, their energy-efficient processing, and recent advancements in training methodologies. It also highlights their potential impact on low-power computing applications, particularly in audio processing, where temporal encoding is crucial. By addressing current limitations and leveraging novel training strategies, SNNs could play a pivotal role in next-generation AI systems.

1 Introduction

Artificial Neural Networks (ANN) are computational models inspired by the brain; made up of layers of interconnected "neurons" which learn patterns when

given large amounts of data. After learning patterns from the data, ANNs can make inference to predict an output based on some given input. ANNs have long been used for application ranging from computer vision to financial forecasting. However the use of ANNs has surged recently, increasing fourfold since 2017, driven by the development of generative AI models like ChatGPT. In addition to this, models are growing in size and becoming more complex, consequently the global AI power consumption is increasing at an exponential rate [1]. Not only does this make models expensive to operate, but it also raises sustainability concerns.

A graphics processing unit (GPU) running a large language model (LLM) can consume hundreds of watts of power, whereas the human brain processes sensory information, controls motor functions, runs its own language model, and enables conscious thought — all while using only about 20 watts. Such efficient information processing has motivated research into neuromorphic computing - a field aiming to emulate the brain's neural architecture. Spiking Neural Networks (SNNs) are considered the third generation of neural network models [2], they more closely model how biological neurons behave in the brain than ANNs. For a second generation neuron, the output is a weighted sum of the inputs passed through an activation function (Figure 1 (a). For a third generation neuron, the output is a train of spikes which varies in timing and frequency (Figure 1 (b). Since information is encoded in the timing of outputs, SNNs can inherently capture temporal patterns and dynamic behaviours in sequential data, making them theoretically better at processing tasks involving time-dependent signals, such as speech recognition, sensory processing, and event-based data streams. It has been shown that SNNs theoretically posses higher computational power than the previous generation of ANNs [3]. Unlike ANNs, spiking neurons only activate when necessary and don't require a clock signal, enabling massive power savings. For instance, the leading neuromorphic platform TrueNorth is capable of simulating a million spiking neurons in real-time while consuming 63 mW. The equivalent network executed on a high-performance computing platform was 100–200× slower than real-time and consumed 100,000 to 300,000× more energy per synaptic event [4]. The power savings of SNNs could be game-changing in edge computing devices that run on battery, like smartphones or remote environmental sensors. On top of this, their low-latency could benefit autonomous vehicles and robotics. The scope of SNN applications is great. In addition to the advantage in theoretical computational power, the power efficiency, and low-latency, SNNs tend to be more explainable (heard this on a video from MIT training SNNs to drive a car).

2   Spiking neuron model

The leaky integrate-and-fire (LIF) neuron model is the most widely used for SNNs. simple, fast, and scalable for large networks, though less biologically detailed. Equations (1), (2), and (3) define how LIF neurons behave.
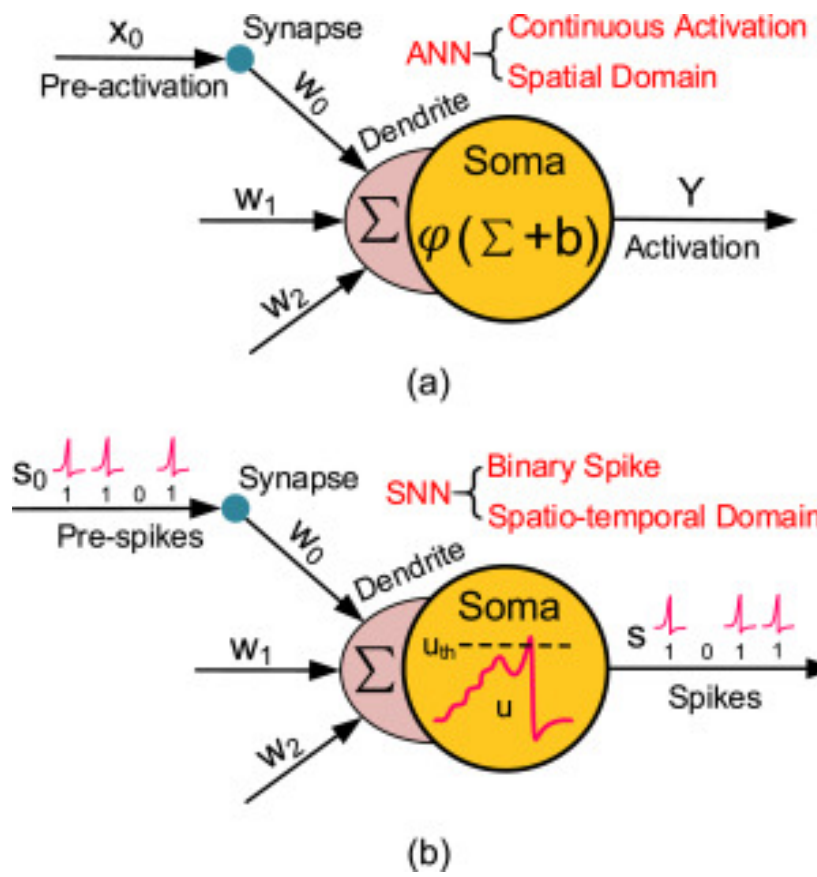
Figure 1: (a) ANN neuron (b) SNN neuron

$$H[t] = V[t-1] + \frac{1}{\tau}(X[t] - (V[t-1] - V_{reset})), \tag{1}$$

$$S[t] = \Theta(H[t] - V_{th}), \tag{2}$$

$$V[t] = H[t](1 - S[t]) + V_{reset}S[t] \tag{3}$$

Literature discussing different parameters to adjust: firing threshold [5], membrane time constant [6], weights [7], activation function between charging and firing [8].

Other models of neurons include: **Parallel Spiking neuron model**: [9]
Izhikevich – efficiently reproduces diverse spiking patterns with low computational cost;
Hodgkin–Huxley – offers high biophysical accuracy but is computationally intensive.

Todo: Information coding

Todo: Why I will focus on audio data.

3   Training Neural Networks

Backpropagation is the most widely used and effective method for training second generation artificial neural networks. It operates by iteratively adjusting the network's weights to minimize error. The process follows these key steps:

1. **Forward Pass**: The model receives an input and processes it through its layers, generating an output.
2. **Error Calculation**: The output is compared to the desired target, and a **loss function** quantifies the error.
3. **Gradient Computation**: The derivative of the loss function is computed with respect to each weight in the network using the **chain rule of calculus**.
4. **Weight Update**: The weights are adjusted by subtracting a fraction of their corresponding gradient, typically scaled by a learning rate. This step moves the model toward a configuration that reduces error.

This iterative optimization nudges the network toward improved accuracy, ensuring better alignment between predictions and target values over multiple training cycles. The problem with applying this algorithm to SNNs is that spike events are non-differentiable, therefore the gradient network cannot be calculated and the weights cannot be adjusted. So how are SNNs trained?

3.1 ANN-to-SNN conversion

Due to the popularity of ANNs, literature on training them is quite advanced, so a natural and popular choice for training SNNs has been by converting a trained ANN model into an SNN model. Such ways of training have had good results for some tests, however such a method incurs large computational costs during conversion and is limited by the architecture of ANNs which are less adaptable to dynamic data [10]. Thus, to fully harness the benefits of SNNs — from energy efficiency to novel architectures — effective direct training methods are essential.

3.2 Direct training

There are several approaches to directly training SNNS. ### Local learning rules

Local learning rules - Biologically feasible - Asynchronous unsupervised learning - More efficient learning using spiking behaviours

Memristor based STDP is an algorithm which is tailored to neuromorphic hardware - taking advantage of memristor behaviours. It uses local learning rules and is deployable on neuromorphic devices. [11]

3.2.1 BPTT + SG

- A very prominent method for directly training SNNs
- SG used to approximate the gradient
- Does well on static datasets
- Needs a ton of memory [12] The most prominent approach is backpropagation-through-time (BPTT), often paired with the surrogate gradient (SG) method. Since spikes are inherently non-differentiable, SG provides an approximation of their gradients, enabling the use of BPTT to optimize the model's parameters. This combination has become a prominent approach for training SNNs, allowing models to learn from temporal data and allowing further exploration of novel SNN architectures. However, BPTT is not considered a biologically plausible learning method, as neurons in the brain are believed to learn using local information [12] — that is, signals received directly from neighbouring neurons. As a result, BPTT may not fully leverage the efficiencies observed in biological systems. For example, implementing local learning rules could reduce data movement across the chip, potentially enhancing energy efficiency.

3.2.2 e-prop

An interesting approach. BPTT is biologically implausible as it requires storage of past neuron states and propagating errors backwards. Moreover, e-prop facilitates online real-time learning. Also e-prop requires far less energy than BPTT. Though it has it's advantages, models trained using e-prop aren't as accurate as BPTT. [10]

3.2.3 Parallelizable LIF

Neuron spike events are sequential

3.2.4 Eventprop

A novel training algorithm which utilises precise spike gradients to train the network. Eventprop computes gradients in a more efficient manner than BPTT. It reaches SOTA performance while using 4x less memory and running 3x faster. [13]

4   NOTES:

Backpropagation based methods: - Alternatives to BPTT include OTTT, SKTT, e-prop. - Some biological feasibility to BP since the brain does send error signals. Top down reward system? - BP not fully understood, even random feedback captures some of the effectiveness. -

Algorithms for training using gradient: SpikeProp, Chronotron, tempetron, Widrow-Hoff, OSD, SOAN, Slayer, relating to STDP, OTTT.

Other ways to improve accuracy: - Using trainable axonal delay

References

[1] B. Kindig. AI Power Consumption: Rapidly Becoming Mission-Critical. Forbes. [Online]. Available: https://www.forbes.com/sites/bethkindig/2024/06/20/ai-power-consumption-rapidly-becoming-mission-critical/

[2] W. Maass, "Networks of spiking neurons: The third generation of neural network models," vol. 10, no. 9, pp. 1659–1671. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608097000117

[3] W. Maass and H. Markram, "On the computational power of circuits of spiking neurons," vol. 69, no. 4, pp. 593–616. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022000004000406

[4] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "A million

spiking-neuron integrated circuit with a scalable communication network and interface," vol. 345, no. 6197, pp. 668–673. [Online]. Available: https://www.jstor.org/stable/24745271

[5] S. Wang, T. H. Cheng, and M.-H. Lim, "LTMD: Learning Improvement of Spiking Neural Networks with Learnable Thresholding Neurons and Moderate Dropout," vol. 35, pp. 28 350–28 362. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/b5fd95d6b16d3172e307103a97f19e1b-Abstract-Conference.html

[6] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, "Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks." IEEE Computer Society, pp. 2641–2651. [Online]. Available: https://www.computer.org/csdl/proceedings-article/iccv/2021/281200c641/1BmEpXS0r1m

[7] N. Rathi and K. Roy, "DIET-SNN: A Low-Latency Spiking Neural Network With Direct Input Encoding and Leakage and Threshold Optimization," vol. 34, no. 6, pp. 3174–3182. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9556508

[8] C. Jiang and Y. Zhang. KLIF: An optimized spiking neuron unit for tuning surrogate gradient slope and membrane potential. [Online]. Available: http://arxiv.org/abs/2302.09238

[9] S. Y. Arnaud Yarga and S. U. N. Wood, "Accelerating spiking neural networks with parallelizable leaky integrate-and-fire neurons*," vol. 5, no. 1, p. 014012. [Online]. Available: https://dx.doi.org/10.1088/2634-4386/adb7fe

[10] G. Bellec, F. Scherr, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets." [Online]. Available: https://www.proquest.com/publiccontent/docview/2172488501?pq-origsite=primo&sourcetype=Working%20Papers

[11] Spoken Digits Classification Based on Spiking Neural Networks with Memristor-Based STDP. [Online]. Available: https://ieeexplore.ieee.org/document/10216628/citations?tabFilter=papers#citations

[12] C. Zhou, H. Zhang, L. Yu, Y. Ye, Z. Zhou, L. Huang, Z. Ma, X. Fan, H. Zhou, and Y. Tian, "Direct training high-performance deep spiking neural networks: A review of theories and methods," vol. 18. [Online]. Available: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2024.1383844/full

[13] T. Nowotny, J. P. Turner, and J. C. Knight, "Loss shaping enhances exact gradient learning with Eventprop in spiking neural networks," vol. 5, no. 1, p. 014001. [Online]. Available: https://dx.doi.org/10.1088/2634-4386/ada852