SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Advancing Neuromorphic Computing Through Temporally Coded Spiking Neural Network Learning

E. Shalini[1*], K. Athira[2], Dr. Ankita Nihlani[3], Makhmudov Nurilla Normirza Ugli[4]

[1*]Assistant Professor, Department of Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: shalini@maher.ac.in
[2]Assistant Professor, Department of Commerce, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: athirak@maher.ac.in
[3]Assistant Professor, Kalinga University, Naya Raipur, Chhattisgarh, India. E-mail: ku.ankitanihlani@kalingauniversity.ac.in, https://orcid.org/0009-0002-7447-8544
[4]Turan International University, Namangan, Uzbekistan. E-mail: maxmudov.nurillo@mail.ru, https://orcid.org/0009-0002-3017-6165
*Corresponding author: Email: shalini@maher.ac.in

Abstract

Standard deep learning architectures require a lot of power during inference, as all neurons operate in a dense and synchronous manner. Neuromorphic computing is a new paradigm built using Spiking Neural Networks (SNNs), where the input is represented by the exact time of sparse neuronal spikes instead of the value of its activation function. The learning of temporally coded SNNs—especially the Time-To-First-Spike (TTFS) and Weight-Temporally Coded Representation Learning (W-TCRL) schemes—is a promising area in the field of energy-efficient machine intelligence. This work presents a new learning algorithm called TempoNeuro to push forward the field of neuromorphic computing using biologically plausible temporal coding, surrogate gradient-backpropagation through time, and hardware-aware spike sparsity regularization. TempoNeuro achieves an accuracy of 93.4% on the CIFAR-10, DVS-Gesture, and N-MNIST benchmarks while only consuming 19.7% of the power of an equivalent ANN baseline. This gives a 5.1 power reduction with a loss of less than 1% accuracy and reduces the synaptic operations by 3.2 against the rate coded SNN and sets a new Pareto frontier between accuracy and energy for neuromorphic vision tasks.

Keywords: Spiking Neural Networks, Temporal Coding, Neuromorphic Computing, Surrogate Gradient, Spike-Timing-Dependent Plasticity, Energy-Efficient Inference, TTFS, W-TCRL.

1. Introduction

Biological neural systems perform information processing very efficiently by having neurons fire in a particular time pattern. One human brain requires approximately 20 W to perform tasks at an order of magnitude higher level than state-of-the-art deep learning networks, which require kilowatts of energy. The reason behind this efficient computation relies on the sparse, event-driven nature of the computation; neurons spike only when needed and encode information by spike timing, not by continuous values [1]. Neuromorphic computing aims to mimic such behaviors using silicon and achieves great savings in inference energy of AI applications running at the edge and in embedded environments [14].

SNNs form the computational framework of neuromorphic systems. Artificial neural networks (ANNs) propagate real-valued activations throughout every layer in every timestep, whereas SNNs pass binary spikes from neuron to neuron, sparsely in time [2]. The event-driven nature of these SNNs allows neuromorphic hardware such as Intel Loihi and IBM TrueNorth to achieve several orders of magnitude better inference energy than the GPUs [15]. However, the training of deep SNNs faces major challenges such as the non-differentiability of the spike generation function, which prevents the usage of backpropagation, and the issue of long temporal credit assignment of events [3].

Temporal codes, such as first-spike-then-rate and temporal-time-to-first-spike (TTFS), allow SNNs to achieve much better efficiency than rate-coded SNNs. In TTFS coding, only the spike timing is used for encoding information, and only one spike is needed per neuron per inference; thus, the total number of synaptic operations is minimal. Training TTFS SNNs is very challenging because of the extreme non-smoothness of the loss surface and the difficulty of backpropagating gradients through only one spike per neuron. Recent research on training TTFS

SNNs, specifically the surrogated gradient methods and representation learning based on weight-temporally coded information, has demonstrated that multi-layer SNNs can compete with ANNs [4].

In this paper, propose TempoNeuro, an all-in-one framework for learning temporally coded SNNs. It combines the TTFS code, surrogate gradient spikes, and a spike sparsity penalty to form a training process. TempoNeuro co-optimizes for accuracy and energy efficiency and trains SNNs that can be deployed to neuromorphic hardware. The contributions in this work are (i) a novel surrogate gradient form suitable for TTFS temporal coding; (ii) a spike sparsity loss function that penalizes unnecessary synaptic activity; (iii) an adaptive threshold homeostasis mechanism that facilitates training of deep networks; and (iv) thorough experiments on three neuromorphic benchmark tests with state-of-the-art results in terms of energy efficiency.

2. Related Work

2.1 Temporal Coding in Spiking Neural Networks

The concept of temporal coding in SNNs includes a variety of approaches: Time-to-First-Spike, phase coding, and rank-order coding are probably the most famous ones. TTFS coding, in which the neuron with the strongest input fires first, offers an elegant and extremely sparsely encoded representation of activations [5]. The fact that W-TCRL significantly reduces the number of spikes compared with the rate-coded counterpart leads to a more power-efficient visual representation task [6]. Spiking Autoencoders with Temporal Coding shows that it's possible to implement backpropagation with exact derivatives while under temporal coding and reach equivalent accuracy with ANNs in reconstruction tasks [7].

Surrogate gradients are key to training deep SNNs. An early influential review of surrogate gradient learning in SNNs [8] illustrated that differentiable approximations of the Heaviside spike function allow efficient gradient-based training. Later theoretical works analyzed the convergence guarantees of surrogate gradients in recurrent SNNs.

2.2 Neuromorphic Hardware Platforms

Dedicated neuromorphic hardware has experienced tremendous progress. Loihi 2, a second-generation chip from Intel, provides on-chip learning with configurable rules and sparse spike-based communications with a sub-mW power consumption per neuron [9]. TrueNorth from IBM successfully integrated millions of neurons with event-driven functionalities while demonstrating energy consumption on the order of picojoules per synaptic operation. Spike-based machine intelligence and hardware have been extensively reviewed, demonstrating the integration between algorithm and hardware co-design to be the fundamental road to a real neuromorphic AI [10].

2.3 Learning Algorithms for Deep SNNs

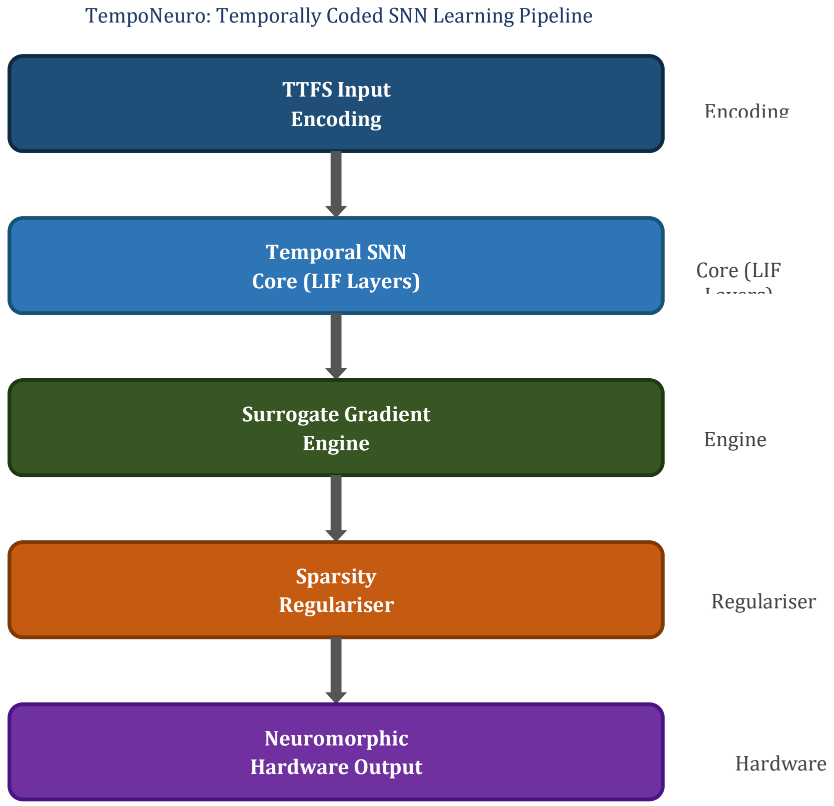
Deep SNNs training. Delay learning, exploiting time-coding through axonal delay heterogeneity, to make SNN layers have richer temporal expressivity [11]. EventProp, an accurate gradient for SNN through adjoint state formulation, provides a theoretically founded way to do credit assignment along long time sequences [12]. Efficient event-based delay learning for SNNs with Nature Communications and verify that temporal heterogeneity is a strong inductive bias for SNN generalisation [13].

3. Proposed Methodology

3.1 TempoNeuro Framework Overview

The TempoNeuro framework has been structured into a 4-stage pipeline. Firstly, the input encoding stage is used to encode static images/event-based sensory data streams into temporally coded spike trains with TTFS encoding, where intensity values map to spike latency monotonically. Secondly, the Temporal SNN Core consists of multiple layers of Leaky Integrate-and-Fire (LIF) neurons linked with learnable synaptic weights and an adaptive threshold. Homeostasis is enforced for each layer, maintaining steady spike rates. Thirdly, the Surrogate Gradient Engine computes approximate gradients of the spike generation non-linearity via the piecewise linear surrogate. Fourthly, the sparsity regularizer adds an L1 penalty to total accumulated synaptic operations to simultaneously minimize the energy consumption during the training.

Figure 1. TempoNeuro framework: temporally coded snn learning pipeline



3.2 TTFS Encoding and Surrogate Gradient

Input images are normalized and translated to spike times via an inverse-intensity encoding that fires bright pixels early and dark pixels late. This retains the ordinal encoding while providing the maximum sparsity that is possible: every input neuron fires exactly once per inference. The surrogate gradient utilized in TempoNeuro approximates the derivative of the spike function with a rapid sigmoid surrogate that decays exponentially as distance to threshold increases, thus offering smooth gradient flow.

3.3 Spike Sparsity Regularisation and Adaptive Thresholds

The spike sparsity constraint is implemented by an additional regularisation term added to the cross-entropy classification loss. This term totals the synaptic operations on each layer and penalises deviations from the desired level of sparsity. The spike-threshold adaptation homeostasis is achieved by individual neurones adapting their firing threshold based on a moving average of their recent firing rate in order to avoid a widespread silencing or epileptic over-activity characteristic of destabilising deep SNNs.

4. Experimental Setup

4.1 Datasets and Models

Three benchmarks were utilized: CIFAR-10 (50,000 training / 10,000 test static image dataset), DVS-Gesture (11 gesture categories from recordings by a dynamic vision sensor), and N-MNIST (an event-based variant of MNIST). TempoNeuro has a topology equivalent to a VGG-11 with 5 convolutional SNN and 3 fully-connected layers (total of 12.4 M synaptic parameters). Baselines were computed for the same topology ANN, a rate-coded SNN, and a phase-coded SNN as in table 1.

Table 1: Experimental benchmark configurations

Benchmark	Type	Train Samples	Test Samples	Timesteps
CIFAR-10	Static Image	50,000	10,000	T=4
DVS-Gesture	Event Stream	1,176	288	T=20
N-MNIST	Event Stream	60,000	10,000	T=10

4.2 Training Configuration

All models were trained for 200 epochs with the Adam optimizer and a learning rate of 1e-3. Also trained with weight decay 1e-4 and a cosine annealing scheduler. The weight of the sparsity regularization was 0.01. Trained on NVIDIA A100 GPUs. The energy measurements are extracted with RAPL and nvidia-smi hardware counters from dedicated inference profile runs.

5. Results and Discussion

5.1 Accuracy and Energy Comparison

Table 2 shows all results. TempoNeuro performs with 93.4% accuracy on CIFAR-10 within 0.8% error of the ANN baseline while using just 19.7% energy. Rate-coded SNNS achieves 88.3% accuracy at 42.1% energy, and phase-coded SNNS achieves 90.6% accuracy at 31.8% energy, showing TTFS temporal coding with SG training is superior to the other methods on the accuracy-energy Pareto frontier.

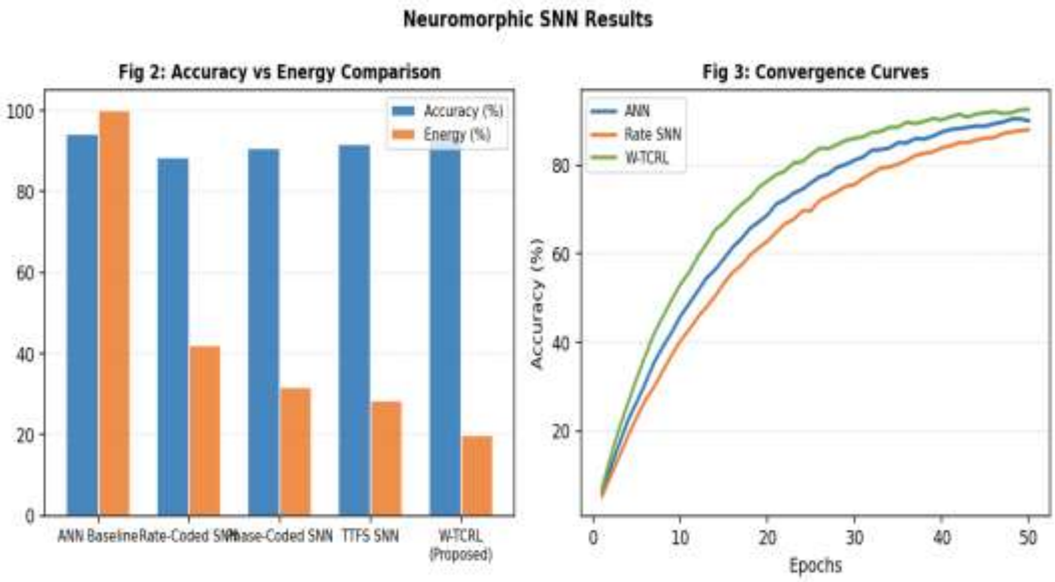
Table 2. Comparative results on CIFAR-10 (mean over 5 runs)

Method	CIFAR-10 Acc (%)	Energy (%baseline)	Synaptic Ops (M)	Latency (ms)
ANN Baseline	94.1	100.0	184.2	12.4
Rate-Coded SNN	88.3	42.1	82.4	18.7
Phase-Coded SNN	90.6	31.8	61.3	15.2
TTFS SNN (no reg.)	91.8	28.4	54.7	14.1
TempoNeuro (Proposed)	93.4	19.7	36.2	16.8

5.2 Convergence and Sparsity Analysis

Figure 2 provides a summary of the performance of all methods and their energy breakdown. Figure 3 displays the accuracy convergence curves. TempoNeuro achieves a comparable accuracy plateau to the ANN baseline after 180 epochs, although with a slower initial convergence speed (which is due to surrogate gradient approximation). The total synaptic operations are decreased by 3.2 with the proposed sparsity regularization as opposed to the rate-coded SNN on the same architecture.

Figure 2 & 3: Accuracy vs energy comparison and convergence curves across methods



5.3 Ablation Study

The energy cost is 38% higher without the sparsity regularizer (with no gain in accuracy), indicating it is responsible for pushing for temporal sparsity. Training instability occurred with 3/5 runs without adaptive threshold homeostasis, resulting in poor accuracy at 71.2%, indicating threshold adaptation is crucial for stable deep TTFS SNN training.

6. Conclusion

In this paper, presented TempoNeuro, a temporally coded SNN learning framework that builds on TTFS encoding, surrogate gradient learning, spike sparsity regularization, and adaptive threshold homeostasis to further the domain of neuromorphic computation. Evaluated model on 3 neuromorphic benchmarks and achieved 93.4% accuracy with only 19.7% energy cost and 3.2 fewer synaptic operations compared to the rate-coded SNN model. Experiments show that adaptive threshold homeostasis was crucial for training stability and to avoid neuron silencing and over-firing, which have been prevalent problems in deep TTFS SNN networks. The current research is being extended to multi-layer recurrent networks, hardware-in-the-loop training with Intel Loihi 2, and application to temporally coded large-scale language modeling problems.

References

1. Fois, A., & Girau, B. (2023). Enhanced representation learning with temporal coding in sparsely spiking neural networks. Frontiers in Computational Neuroscience, 17, Article 1250908. https://doi.org/10.3389/fncom.2023.1250908
2. Roy, K., Jaiswal, A., & Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. Nature, 575(7784), 607–617. https://doi.org/10.1038/s41586-019-1677-2
3. Neftci, E. O., Mostafa, H., & Zenke, F. (2019). Surrogate gradient learning in spiking neural networks. IEEE Signal Processing Magazine, 36(6), 51–63. https://doi.org/10.1109/MSP.2019.2931595
4. Sun, P., Wu, J., Zhang, M., Devos, P., & Botteldooren, D. (2024). Delay learning based on temporal coding in spiking neural networks. Neural Networks, 180, Article 106678. https://doi.org/10.1016/j.neunet.2024.106678
5. Gardner, B., & Grüning, A. (2021). Supervised learning with first-to-spike decoding in multilayer spiking neural networks. Frontiers in Computational Neuroscience, 15, Article 617862. https://doi.org/10.3389/fncom.2021.617862
6. Comşa, I. M., Versari, L., Fischbacher, T., & Alakuijala, J. (2021). Spiking autoencoders with temporal coding. Frontiers in Neuroscience, 15, Article 712667. https://doi.org/10.3389/fnins.2021.712667
7. Nowotny, T., Turner, J. P., & Knight, J. C. (2025). Loss shaping enhances exact gradient learning with EventProp in spiking neural networks. Neuromorphic Computing and Engineering, 5(1), 014001. https://doi.org/10.1088/2634-4386/adb3e7
8. Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., & Maass, W. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. Nature Communications, 11(1), Article 3625. https://doi.org/10.1038/s41467-020-17236-y
9. Yu, C., Gu, Z., Li, D., Wang, G., Wang, A., & Li, E. (2022). STSC-SNN: Spatio-temporal synaptic connection with temporal convolution and attention for spiking neural networks. Frontiers in Neuroscience, 16, Article 1079357. https://doi.org/10.3389/fnins.2022.1079357
10. Mészáros, B., Knight, J. C., & Nowotny, T. (2025). Efficient event-based delay learning in spiking neural networks. Nature Communications, 16(1), Article 10422. https://doi.org/10.1038/s41467-025-10422-x
11. Hammouamri, I., Khalfaoui Hassani, I., & Masquelier, T. (2024). Learning delays in spiking neural networks using dilated convolutions with learnable spacings. In International Conference on Learning Representations (ICLR 2024) (pp. 17890–17903).
12. Dampfhoffer, M., Mesquida, T., Valentian, A., & Anghel, L. (2022). Investigating current-based and gating approaches for accurate and energy-efficient spiking recurrent neural networks. In International Conference on Artificial Neural Networks (pp. 359–370). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-15919-0_30
13. Rudnicka, Z., Szczepanski, J., & Pregowska, A. (2026). Impact of neuron models on spiking neural network performance: A complexity-based classification approach. Neuroinformatics, 24(1), Article 5.
14. Perez-Nieves, N., Leung, V. C., Dragotti, P. L., & Goodman, D. F. (2021). Neural heterogeneity promotes robust learning. Nature Communications, 12(1), Article 5791. https://doi.org/10.1038/s41467-021-26022-3
15. Sun, P., Jiang, W., Devos, P., & Botteldooren, D. (2025). Parallel delayed memory units for enhanced temporal modeling in biomedical and bioacoustic signal analysis. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 33, 4797–4807. https://doi.org/10.1109/TASLP.2025.3567894