Fast and deep neuromorphic learning with first-spike coding

J. Göltz, A. Baumbach, S. Billaudelle, A. F. Kungl,
O. Breitwieser, K. Meier, J. Schemmel
julian.goeltz@kip.uni-heidelberg.de
Heidelberg University
Heidelberg, Germany

L. Kriener
M. A. Petrovici
petrovici@pyl.unibe.ch
University of Bern
Bern, Switzerland

ABSTRACT

For a biological agent operating under environmental pressure, energy consumption and reaction times are of critical importance. Similarly, engineered systems also strive for short time-to-solution and low energy-to-solution characteristics. At the level of neuronal implementation, this implies achieving the desired results with as few and as early spikes as possible. In the time-to-first-spike coding framework, both of these goals are inherently emerging features of learning. Here, we describe a rigorous derivation of error-backpropagation-based learning for hierarchical networks of leaky integrate-and-fire neurons. This narrows the gap between previous existing models of first-spike-time learning and biological neuronal dynamics, thereby also enabling fast and energy-efficient inference on analog neuromorphic devices that inherit these dynamics from their biological archetypes.

CCS CONCEPTS

• **Computing methodologies → Learning paradigms**; • **Hardware → Analog and mixed-signal circuits**.

KEYWORDS

deep learning, error backpropagation, spiking neural networks, time-to-first-spike coding, analog neuromorphic hardware

ACM Reference Format:
J. Göltz, A. Baumbach, S. Billaudelle, A. F. Kungl, O. Breitwieser, K. Meier, J. Schemmel, L. Kriener, and M. A. Petrovici. 2020. Fast and deep neuromorphic learning with first-spike coding. In *NICE '20: Neuro-Inspired Computational Elements Workshop, March 17–20, 2020, Heidelberg, Germany*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In many applications, the time and energy to solution represent essential commodities. For spiking networks, optimal use of these resources is often equivalent to having as few and as early spikes as possible. However, the discrete and therefore discontinuous nature of spikes makes it difficult to apply optimization algorithms based on differentiable loss functions.

In the time-to-first-spike (TTFS) coding scheme, a neuron encodes a continuous variable as the time elapsed before its first spike. For such networks, an efficient gradient-descent-based learning scheme was proposed in [4], using error backpropagation on a continuous function of output spike times. However, this approach is limited to a neuron model without leak, which is neither biologically plausible, nor compatible with most analog VLSI neuron dynamics [8].

We generalize the method to include an exact, closed-form expression for finite membrane time constants (Section 2), demonstrate the learning process using a 3-layer network in software simulations (Section 3), and apply this framework to a network emulated on neuromorphic hardware (Section 4). The latter is particularly relevant for neuromorphic systems based on analog neurosynaptic cores, as it explicitly exploits their inherent parallelism, speed and/or power efficiency, while ensuring direct compatibility with the finite time constants of their neuron and synapse dynamics.

2 DIFFERENTIABLE FUNCTIONS FOR ERROR BACKPROPAGATION

Consider a hierarchical feed-forward network as shown in Fig. 1A. In our coding scheme, information is provided by the first-spike times of neurons: input neurons spike earlier for black pixels as compared to white pixels and the inferred class is given by the first neuron to spike in the label layer (Fig. 1 B).

Error backpropagation requires a loss function that is differentiable with respect to both synaptic weights and output spike times. Here, we choose a loss that, when optimized, decreases the first-spike time of the correct label neuron, while maximizing its (temporal) distance to the spike times of all other label neurons:

$$L[\mathbf{t}, j] = -\log\left[\frac{\exp(-t_j/\xi\tau_\mathrm{s})}{\sum_i \exp(-t_i/\xi\tau_\mathrm{s})}\right] + \alpha \exp\left(\frac{t_j}{\tau_\mathrm{s}}\right) , \qquad (1)$$

with label spike times t_i, the index of the correct label j, synaptic time constant τ_s and scaling hyperparameters ξ and α.

The gradient of the loss now depends on the used neuron model. For leaky integrate-and-fire (LIF) neurons with current-based synapses, spikes occur when the voltage

$$u(t) \propto \sum_i w_i \theta(t - t_i) \left[\exp\left(-\frac{t-t_i}{\tau_\mathrm{m}}\right) - \exp\left(-\frac{t-t_i}{\tau_\mathrm{s}}\right)\right] \qquad (2)$$

crosses the threshold ϑ. This is markedly different from the much simpler (non-leaky) integrate-and-fire (IF) model [4], where the membrane time constant τ_m is infinite and single post-synaptic potentials (PSPs) are hence monotonic. Whereas a sequence of small weight reductions can push individual IF output spikes to arbitrary points in time, the latest possible spike time of an LIF neuron is related to the PSP peak, which occurs at a finite time and is independent of the synaptic weight (Fig. 1C,D). Since the PSP

NICE '20, March 17–20, 2020, Heidelberg, Germany
© 2020 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

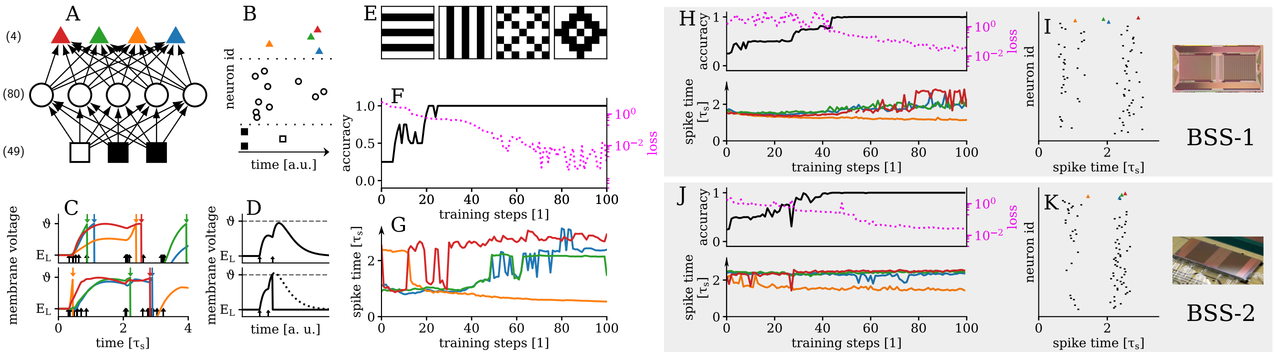

Figure 1: (A) Hierarchical network structure and neuron numbers per layer. Colors encode labels. (B) Input (□, ■), hidden (○) and label (▲) spike times. The first label neuron to spike determines the inferred class (▲). (C) Membrane voltage traces before (top) and after (bottom) training, with input (upward arrows) and output (downward arrows) spikes. Voltage traces are not used during training, only spike times, given in units of τ_s. (D) Illustration of a key challenge posed by finite membrane time constants: small variations of input spike times or synaptic weights (not shown) result in a discontinuity induced by the forgetting membrane. (E) Input pattern set consisting of four classes. (F) Accuracy increase and corresponding decrease of loss during learning. (G) Evolution of label neuron spike times during training for the same class as in C. The correct neuron's spike time decreases while all others are pushed back, producing a distinct gap. (H,I) Accuracy, loss, spike time evolution during training and raster plot after training on BrainScaleS-1 [6]. (J,K) Same as in (H,I) but on BrainScaleS-2 [1].

shape is a difference of exponentials, a general closed-form solution for $u(T) = \vartheta$ does not exist. However, for special cases, we can find

$$\frac{T}{\tau_s} = \ln\left[\frac{a_1}{a_\infty+\vartheta}\right], \qquad \tau_m = \infty \text{ (IF)}\ ; \qquad (3)$$

$$\frac{T}{\tau_s} = \frac{b}{a_1} - W\left[-\frac{g_L\vartheta}{a_1}\exp\left(\frac{b}{a_1}\right)\right], \qquad \tau_m = \tau_s \text{ (LIF)}\ ; \qquad (4)$$

$$\frac{T}{\tau_s} = 2\ln\left\{2a_2/\left[a_1+\left(a_1^2-4a_2g_L\vartheta\right)^{1/2}\right]\right\}, \qquad \tau_m = 2\tau_s \text{ (LIF)}\ ; \qquad (5)$$

where g_L is the leak conductance towards a resting potential $E_L = 0$ and W the Lambert W function, $a_n = \sum_i w_i e^{t_i/n\tau_s}, b = \sum_i w_i \frac{t_i}{\tau_s} e^{t_i/\tau_s}$ with summation over all i with $t_i < T$. Eq. (3) is the solution discussed in [4]. All these equations now allow a recursive calculation of the required $\partial L/\partial t_i$ and $\partial t_i/\partial w_{ik}$. While both new rules for finite τ_m work well in practice [3], we found that learning is more robust for $\tau_m = \tau_s$, so we focus on this scenario in the following.

3 CLASSIFYING A SIMPLE DATASET

We showcase the above framework in a pattern classification task (Fig. 1E), with the spiking network simulated in NEST [2]. To assist learning, the updates were normalized, and for layers with too few output spikes the weights were increased to have sufficient activity.

Fig. 1F shows the evolution of the loss during training, along with the associated classification accuracy. As misclassifications are given disproportionate weight, the loss continues to decrease long after perfect accuracy is achieved.

While not used for training, voltages help understand the learning process. Fig. 1C shows voltages in the label layer for one class (orange) before and after training, illustrating how the trained weights make the correct neuron spike earliest by a large margin.

4 FAST NEUROMORPHIC CLASSIFICATION

In this framework, classification speed is a function of the network depth and the time constants τ_m and τ_s. Assuming typical biological timescales, most input patterns in the above scenario are classified within several ms. By leveraging the speedup of neuromorphic systems such as BrainScaleS [1, 7], with intrinsic acceleration factors of 10^3-10^4, the same computation can be achieved within µs. The robustness of our framework for the given task is evidenced by the clear separation of first-spike times (Fig. 1G,H,J).

However, the speed advantages of such analog systems compared to software simulations come at the cost of reduced control, and training needs to cope with phenomena such as spike time jitter and neuron parameter variability. In particular, this implies $\tau_m \neq \tau_s$, so the derived learning rule is only an approximation of true gradient descent in these systems. Nonetheless, we found that, on both BrainScaleS generations, the application of the ideal learning rule still leads to good results for the chosen task (Fig. 1H-K). The number of training steps can only be compared roughly because of the critical dependence on the learning rate, but when using similar hyperparameters, we observe convergence after a similar number of training steps in both software simulations and hardware emulations (Fig. 1F,H,J). Since the dynamical timescales directly affect the duration of the network emulation between synaptic updates, the hardware acceleration provides a corresponding reduction of the total training time.

5 DISCUSSION AND OUTLOOK

Building on work from [4], our model extends the backpropagation-based time-to-first-spike learning framework to include more biologically plausible and neuromorphic-hardware-compatible neuronal dynamics. As the algorithm minimizes the time before the first spike in the label layer, the trained network needs less than $2\tau_s$ to classify an input pattern, using, on average, less than one spike per neuron in the network. Importantly, synaptic updates only require access to spike times and no other variables such as voltages, which are not easily accessible on most analog substrates.

The speed and sparsity enforced by the learning paradigm is particularly beneficial for neuromorphic systems when considering the critical aspects of time-to-solution, energy-to-solution and I/O bandwidth. On the accelerated BrainScaleS systems, a single classification takes less than 10 µs of wall-clock time. Taking into consideration relaxation times between patterns, our setup is able to handle a pattern throughput of at least 20 kHz, independently of emulated network size. The complexity of the learned dataset was mostly limited by the size of the used substrate and we expect the framework to scale to significantly more challenging problems, as suggested by the FPGA-based experiments in [5].

ACKNOWLEDGMENT

The authors wish to thank Dominik Dold, Jakob Jordan and Nico Gürtler for valuable discussions and gratefully acknowledge funding from the European Union under grant agreements 604102, 720270, 785907 (HBP) and the Manfred Stärk Foundation.

REFERENCES

[1] Sebastian Billaudelle, Yannik Stradmann, Korbinian Schreiber, Benjamin Cramer, Andreas Baumbach, Domnik Dold, Julian Göltz, Akos F. Kungl, Timo C. Wunderlich, Andreas Hartel, Eric Müller, Oliver J. Breitwieser, Christian Mauch, Mitja Kleider, Andreas Grübl, David Stöckel, Christian Pehle, Arthur Heimbrecht, Philipp Spilger, Gerd Kiene, Vitali Karasenko, Walter Senn, Karlheinz Meier, Johannes Schemmel, and Mihai A. Petrovici. 2019. Versatile emulation of spiking neural networks on an accelerated neuromorphic substrate. in preparation.

[2] Marc-Oliver Gewaltig and Markus Diesmann. 2007. NEST (NEural Simulation Tool). *Scholarpedia* 2, 4 (2007), 1430.

[3] Julian Göltz. 2019. *Training Deep Networks with Time-to-First-Spike Coding on the BrainScaleS Wafer-Scale System.* Master's thesis. Universität Heidelberg. http://www.kip.uni-heidelberg.de/Veroeffentlichungen/details.php?id=3909

[4] Hesham Mostafa. 2017. Supervised learning based on temporal coding in spiking neural networks. *IEEE transactions on neural networks and learning systems* 29, 7 (2017), 3227–3235.

[5] H. Mostafa, B. U. Pedroni, S. Sheik, and G. Cauwenberghs. 2017. Fast classification using sparsely active spiking networks. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–4. https://doi.org/10.1109/ISCAS.2017.8050527

[6] Johannes Schemmel, Daniel Brüderle, Andreas Grübl, Matthias Hock, Karlheinz Meier, and Sebastian Millner. 2010. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 1947–1950.

[7] Sebastian Schmitt, Johann Klähn, Guillaume Bellec, Andreas Grübl, Maurice Güttler, Andreas Hartel, Stephan Hartmann, Dan Husmann, Kai Husmann, Sebastian Jeltsch, et al. 2017. Neuromorphic hardware in the loop: Training a deep spiking network on the brainscales wafer-scale system. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2227–2234.

[8] Chetan Singh Thakur Thakur, Jamal Molin, Gert Cauwenberghs, Giacomo Indiveri, Kundan Kumar, Ning Qiao, Johannes Schemmel, Runchun Mark Wang, Elisabetta Chicca, Jennifer Olson Hasler, et al. 2018. Large-scale neuromorphic spiking array processors: A quest to mimic the brain. *Frontiers in neuroscience* 12 (2018), 891.