SUPERVISED TRAINING OF SIAMESE SPIKING NEURAL NETWORKS WITH EARTH MOVER'S DISTANCE

Mateusz Pabian, Dominik Rzepka, Mirosław Pawlak

Department of Measurement and Electronics
AGH University of Science and Technology, Kraków

ABSTRACT

This study adapts the highly-versatile siamese neural network model to the event data domain. We introduce a supervised training framework for optimizing Earth Mover's Distance (EMD) between spike trains with spiking neural networks (SNN). We train this model on images of the MNIST dataset converted into spiking domain with novel conversion schemes. The quality of the siamese embeddings of input images was evaluated by measuring the classifier performance for different dataset coding types. The models achieved performance similar to existing SNN-based approaches (F1-score of up to 0.9386) while using only about 15% of hidden layer neurons to classify each example. Furthermore, models which did not employ a sparse neural code were about 45% slower than their sparse counterparts. These properties make the model suitable for low energy consumption and low prediction latency applications.

***Index Terms*—** spiking neural networks, siamese neural networks, event-based computing, sparse coding

1. INTRODUCTION

Siamese neural networks are a type of machine learning models which are trained to optimize a similarity measure between network outputs for different examples of data from some domain [1]. The resulting model is thus capable of querying inputs based on their similarity, i.e. given some input signal, find the most similar signal or a set of signals. Siamese networks have achieved state-of-the-art results on person re-identification [2], as well as object- [3] and person-tracking tasks [4]. Furthermore, these networks achieve competitive performance in change point detection [5] and can even be used to design ranking engines [6].

Adapting the siamese model to spiking neural networks (SNN), which process data using asynchronous spikes [7], necessitates leveraging existing spike train similarity measures developed by the neuroscientific community. Commonly used measures include van Rossum distance [8], Schreiber's similarity measure [9], Victor-Purpura distance [10] and Earth Mover's Distance (EMD) [11]. Importantly, the choice of a spike train similarity measure is heavily dependent on a chosen neural coding scheme [12]. Various schemes have been observed in biological neurons [13], such as encoding information using impulse frequency [14], relative timing of the first spike [15], or using a group activity pattern across a population of neurons [16]. In the context of SNNs the type of neural coding depends on how the network is trained and the chosen neuron model.

Training SNNs such that the output spike train matches a predetermined temporal pattern based on some similarity measure has been studied extensively. Several early studies, such as ReSuMe [17] or the Tempotron [18], focused on single-layered networks. More recent works however emphasise the ability to train multilayer SNNs. For example, Lin *et al.* [19] propose a general supervised learning rule that minimizes an L_2-distance between kernel-smoothed spike trains, Zenke & Ganguli [20] optimize the van Rossum distance between spike trains, whereas Xing *et al.* [21] train a network by minimizing spike count differences in predetermined time windows over all neurons in the output layer. To the best of our knowledge, the only work directly related to adapting the siamese model to SNNs was summarized by Luo *et al.* [22]. Their model achieved competitive performance on several visual object tracking benchmarks with low precision loss with respect to the original nonspiking network.

In this paper, we propose a novel supervised training scheme for multilayer siamese SNNs which optimizes EMD between output spike trains [11]. We build the network using neurons tuned to respond to precise timing of input events [23]. In contrast to the work of Luo *et al.* [22], our siamese SNN is optimized in the spiking domain, rather than be a product of converting an existing neural network to the spiking domain. Additionally, we opt to encode information using single events instead of bursts of spiking activity. We train the proposed models on images from the MNIST dataset [24] converted to the spiking domain using a novel input coding scheme based on the concept of implicit information [25] carried by events in the spiking domain. Finally, we assess the trained models in terms of classification F1-score, hidden layer activation sparsity and prediction latency.

This work was supported by the Polish National Center of Science under Grant DEC-2017/27/B/ST7/03082.

2. METHODS AND ALGORITHMS

2.1. Backpropagation Algorithm for IF Neural Networks

Let us briefly summarize the model first described in [23], which trains a spiking neural network that is sensitive to the timing of input spikes rather than their rate. This type of network uses Integrate-and-Fire (IF) neurons with exponentially decaying synaptic current kernels. For a given kernel-smoothed input spike train signal $x(t)$ with spike-times $\{t_i, i = 1, \dots, N\}$

$$x(t) = \sum_{i=1}^{N} \kappa(t - t_i) = \sum_{i=1}^{N} \exp\left(-\frac{t - t_i}{\tau_{syn}}\right) u\left(t - t_i\right), \quad (1)$$

where τ_{syn} is the synaptic current time constant and $u(t)$ denotes a Heaviside step function. We define the IF neuron dynamics by the following differential equation

$$\frac{dV(t)}{dt} = \sum_{i=1}^{N} w_i \kappa(t - t_i), \quad (2)$$

where $V(t)$ is the membrane voltage of the postsynaptic neuron, i is the presynaptic neuron index and $\{w_i\}$ are synaptic weights. Importantly, each presynaptic neuron is associated with exactly one spike-time t_i. The closed-form solution of (2) is given by

$$V(t) = V_0 + \tau_{syn} \sum_{i=1}^{N} w_i \left[1 - \exp\left(-\frac{t - t_i}{\tau_{syn}}\right)\right] u\left(t - t_i\right), \quad (3)$$

where V_0 is the initial membrane voltage.

The neuron is said to fire at time t_{out} if the voltage crosses a threshold V_{thr} from below. Then, a subset of input spikes that cause the postsynaptic neuron to fire

$$C = \{i : t_i < t_{out}\} \quad (4)$$

is called a *causal set*. Assuming, without the loss of generality, that $V_0 = 0$, the solution to (3) that describes the relationship between the causal set of input spikes and the time of a postsynaptic neuron spike t_{out} is given in the implicit form

$$z_{out} = \frac{\sum_{i \in C} w_i z_i}{\sum_{i \in C} w_i - \frac{V_{thr}}{\tau_{syn}}}, \quad (5)$$

where $z_i = \exp(\frac{t_i}{\tau_{syn}})$ and $z_{out} = \exp(\frac{t_{out}}{\tau_{syn}})$. This formula is differentiable with respect to synaptic weights $\{w_i\}$ and the transformed input spike times $\{z_i\}$, therefore it can be used to train a spiking network using the backpropagation algorithm.

2.2. Training a Spiking Siamese Neural Network

We train the network to optimize EMD-based similarity between output spike trains [11]. Note that each neuron of the model described in Section 2.1 returns the time of a single event. Therefore, in order to construct the output spike train the events obtained from the last layer are concatenated and sorted in ascending order. We call the resulting point process a *spike train embedding* $f(t)$. EMD is given by

$$\text{EMD}(f, g) = \int_{-\infty}^{\infty} |F(t) - G(t)| \, dt. \quad (6)$$

where F, G are cumulative distribution functions of point processes f, g, respectively. The function F (G is defined analogously) takes the form $F(t) = \frac{1}{P} \sum_{i=1}^{P} u(t - t_i)$. The distributions F, G are piecewise constant nondecreasing functions, hence the numerical evaluation of (6) is straightforward [11]. The overall computational complexity of EMD for a pair of spike trains $f(t)$ and $g(t)$, with P and R events respectively, is $\mathcal{O}(P + R)$, assuming that events $\{t_i, i = 1, \dots, P\}$ and $\{\tau_i, i = 1, \dots, R\}$ are sorted [26].

In this work, a triplet-based loss function was used to optimize the network. Let f_a, f_p, f_n be the spike train embeddings of examples a, p, n called the *anchor*, *positive* and *negative* respectively. Then the formula

$$L_{anp} = \max\left(0, \alpha + \text{EMD}(f_a, f_p) - \text{EMD}(f_a, f_n)\right), \quad (7)$$

describes the contribution of a given triplet to the total loss. Minimizing L_{anp} ensures that the distance between f_a and f_p (or the anchor-positive pair of examples) is smaller than the distance between f_a and f_n (anchor-negative pair) by at least some margin α.

Overall, the loss function is a composite of a triplet loss averaged over the set of valid triplets in the mini-batch Q (the 'batch-all' strategy from [4]), and the spike regularization term [23] which promotes network spiking activity by ensuring a nonnegative denominator of (5)

$$L = \frac{1}{|Q|} \sum_{q \in Q} L_q + K \sum_{j=1}^{M} \max\left(0, \frac{V_{thr}}{\tau_{syn}} - \sum_{i=1}^{N} w_{ij}\right), \quad (8)$$
$$Q = \{a, n, p : \ a \neq n \neq p, \ y_a = y_p \neq y_n\}$$

where y_a, y_p, y_n are the class labels of examples a, p, n, the index j runs over all neurons in the network M, and K is a hyperparameter. Finally, let $Q_{AT} = \{a, n, p \in Q : L_{anp} \neq 0\}$ be the set of active triplets which contribute to a nonzero loss for the current batch of examples. Then the ratio of active triplets $\text{AT} = \frac{|Q_{AT}|}{|Q|}$ is used as an early stopping criterion for the training procedure.

3. RESULTS AND DISCUSSION

3.1. Experimental Setup

We test our approach by training siamese SNNs on a MNIST dataset [24] in three seperate settings which differ primarily in the number of events passed down to network input

neurons. This allows us to explore how the resulting model properties are influenced by different input data pixel-to-spike conversion methods. Each 28x28 image is first flattened into a 784-element vector. The resulting vectors are processed differently, depending on the experimental setting:

Black&white (adapted from [23]): the vector representation of each image is binarized with a threshold of 50% of global maximum pixel intensity, and one of two time instants t_0 or t_1 is assigned to the value of each bit. We set $t_0 = 0$ ms and $t_1 = 1.79$ ms, as in [23].

Binary: each vector representation is binarized as in the *black&white* setting, however the 'late' event-time t_1 is set to infinity (which corresponds to a lack of event for black pixels).

Grayscale: the original grayscale images are converted to the spiking domain by modeling each pixel as an artificial neuron responding to a driving signal (synaptic current) of constant intensity I proportional to image pixel intensity (in range 0-1). For $V_0 = 0$, the formula for the membrane voltage of IF-based converter neurons is $V(t) = \frac{t}{\tau_{syn}} I$, and the spike time corresponding to a pixel of a given intensity is

$$t_{out} = \frac{V_{thr}\tau_{syn}}{I}. \tag{9}$$

This model retains the desirable property that $t_{out} \to \infty$ as $I \to 0$.

Note that in *binary* and *grayscale* settings some channels might not have any events associated with them. In this context, a lack of event occurence carries implicit information [25] that can be exploited by the network.

We use the same 784-400-400-10 network (denoting the number of neurons in input, hidden and output layers, respectively) in all of our experiments. Each model was optimized using RMSprop [27] with a learning rate of 10^{-3}, synapse regularization parameter $K = 400$, synaptic time constant $\tau_{syn} = 1$ ms, voltage threshold $V_{thr} = 1$ mV, and the triplet loss margin $\alpha = 0.1$. Additionally, we apply L_2 regularization with $\lambda = 10^{-3}$ to the loss function (8). Finally, a batch size of $n = 256$ was used across all experiments. No data-augmentation methods were used.

3.2. MNIST Digit Classification

In order to evaluate our approach, we measure k-Nearest Neighbour (k-NN) classifier performance as a proxy for embedding space example proximity. For a trained model, a spike train embeddings was computed for each example, then k training-set embeddings closest to a given test-set embedding were selected, which then could be used to determine the test example label prediction using majority voting. We find that the classifier performance is not influenced by changing k for $k \geq 7$, therefore we set $k = 7$ for all experiments. Compared to other SNN-based approaches (Table 1), our best-performing model achieved a similar level of performance using a novel approach to dataset encoding and signal transformation defined by the trained spiking neural network. Importantly, the focus of this work was to show that the proposed methodology can be used to train multilayer spiking siamese neural networks with timing-sensitive neural coding. Obtaining high accuracy was a secondary objective, mainly as a proxy for determining whether the training procedure was successful or not.

Table 1: Classifier performance of different SNNs on MNIST (best reported results from each paper).

input encoding	model type	performance
-	k-NN Euclidean baseline [24]	0.950
firing rate	Spiking RBM [28]	0.926
	STPD-trained network [29]	0.950
	Spiking NN [30]	0.986
	Spiking CNN [30]	**0.991**
spike-time	Spiking NN [23]	0.975
	Siamese Spiking NN (our)	0.946

3.3. Hidden Layer Activation Sparsity

Interestingly, we have observed that models trained in the *binary* and *grayscale* settings exhibit sparse internal representation of the input signal. We call a group of neurons *quiescent* when they do not fire for a given input signal $x(t)$, or equivalently $H_{q,x} = \{h \in H, t > 0 : V_{h,x}(t) < V_{thr}\}$. Let the *network sparsity index* QN_x be the fraction of quiescent neurons $H_{q,x}$ to all neurons in hidden layers H

$$QN_x = \frac{|H_{q,x}|}{|H|}. \tag{10}$$

We compute the ratio QN_x for each example in the test-set, which we denote QN for brevity. The resulting hidden layer activation sparsity empirical distribution is presented in Fig. 1. The results suggest that this neural activity sparsity is context-based, meaning that a different subset of neurons will respond to each input signal. Only a negligible number of neurons never fire in response to any image (corresponding to $QN_x = 1$ for those images), which implies that the observed sparsity is a result of the causal set neuron selection and is fundamentally different from permanently inactive neurons which can be pruned from the network. Lastly, while there seems to be a slight trade-off between classifier performance and network sparsity (as evidenced by Table 2), it can be considered small given that only about 15% of neurons are used to process each example.

3.4. Classifier Time-Performance

In order to investigate how the classifier performance changes as output events are observed over time, we simulate the use-case where classifier is asked to update its prediction when-

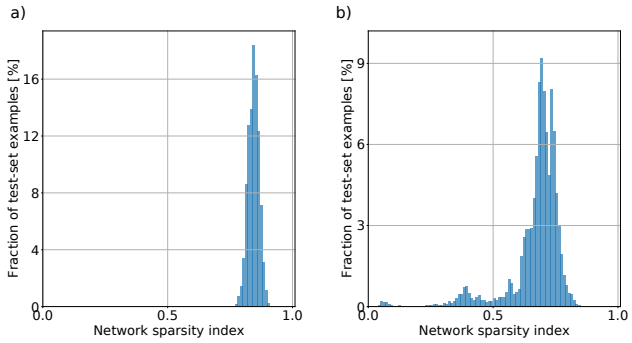

Fig. 1: Network sparsity index empirical distributions for the a) *binary* and b) *grayscale* models.

Table 2: Summary of test-set-averaged classifier performance and observed network sparsity indices QN.

	F1-score	QN
black&white	**0.9466**	0.0013
binary	0.9386	**0.8423**
grayscale	0.9238	0.6698

ever a new output event occurs by comparing with the embeddings of the training-set. This procedure provides valuable insight into prediction latency. Fig. 2 shows the class-averaged classifier accuracy for models trained with different encoding schemes. Interestingly, while the classifier performance of *binary* and *grayscale* models roughly correlates with the overall number of observed output spikes, the performance for the *black&white* model stops improving quite early and reaches its maximum only after observing a small number of late events. If we consider the maximum-accuracy performance of the model as its steady-state, then we find that the *black&white* model achieves steady-state about 45% later than the other models.

Overall, the model time-performance curves show that the quality of class label prediction increases as more output events are observed. A similar classifier accuracy vs. time study was conducted by Diehl *et al.* [30] by describing a rate-coding artificial-to-spiking neural network conversion scheme. They report that the steepness of the time-accuracy curve depends on the network structure and input signal properties. Our results seem to give further proof to their conclusions, although we did not vary the network structure.

4. CONCLUSION

In this paper we presented a novel approach to supervised training of multilayer spiking siamese neural networks applied to image domain. The proposed model of a timing-sensitive spiking neural network can be trained even when some data inputs are represented by a lack of event (through

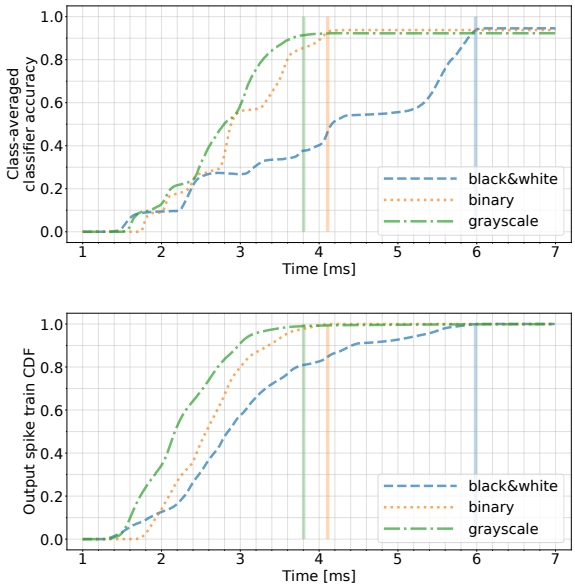

Fig. 2: Top row: the change in class-averaged classifier accuracy over time. Bottom row: cumulative distribution functions of output spike-times for all test-set examples, regardless of class labels. Thick vertical lines denote the time when each model reaches its steady-state performance.

implicit information). This training procedure results in models which take less time to make high-accuracy predictions and process signals using only a small subset of hidden layer neurons firing in response to the input event stream, compared to models trained with explicit event encoding. Concretely, the *black&white* model reached an F1-score of 0.9466 using almost all neurons to make predictions, whereas the *binary* model achieved a slightly lower F1-score of 0.9386, however it encodes information using only 15% of all hidden layer neurons, and is significantly faster to reach its steady-state performance. Further investigation is warranted in order to determine whether the observed accuracy-sparsity trade-off is an artifact of our training procedure, or whether it is an inherent property of spiking neural networks. Additionally, we aim to study the effect of output layer dimensionality on these properties in subsequent research.

5. REFERENCES

[1] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, 2015.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[3] M. Dunnhofer, M. Antico, F. Sasazawa, Y. Takeda, S. Camps, N. Martinel, C. Micheloni, G. Carneiro, and D. Fontanarosa, "Siam-U-Net: Encoder-decoder siamese network for knee cartilage tracking in ultrasound images," *Medical Image Analysis*, vol. 60, pp. 101631, 2020.

[4] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[5] H. Bredin, "TristouNet: Triplet loss for speaker turn embedding," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5430–5434.

[6] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "RankIQA: Learning from rankings for no-reference image quality assessment," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1040–1049.

[7] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," *Frontiers in Neuroscience*, vol. 12, pp. 774, 2018.

[8] M. C. W. van Rossum, "A novel spike distance," *Neural Computation*, vol. 13, no. 4, pp. 751–763, 2001.

[9] S. Schreiber, J.-M. Fellous, D. Whitmer, P. Tiesinga, and T. J. Sejnowski, "A new correlation-based measure of spike timing reliability," *Neurocomputing*, vol. 52, pp. 925–931, 2003.

[10] J. D. Victor and K. P. Purpura, "Metric-space analysis of spike trains: theory, algorithms and application," *Network: Computation in Neural Systems*, vol. 8, no. 2, pp. 127–164, 1997.

[11] D. Sihn and S.-P. Kim, "A spike train distance robust to firing rate changes based on the Earth Mover's Distance," *Frontiers in Computational Neuroscience*, vol. 13, pp. 82, 2019.

[12] E. Satuvuori and T. Kreuz, "Which spike train distance is most suitable for distinguishing rate and temporal coding?," *Journal of Neuroscience Methods*, vol. 299, pp. 22–33, 2018.

[13] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, Computational Neuroscience Series, 2001.

[14] E. D. Adrian and Y. Zotterman, "The impulses produced by sensory nerve-endings: Part II. The response of a single end-organ," *The Journal of Physiology*, vol. 61, no. 2, pp. 151–171, 1926.

[15] T. Gollisch and M. Meister, "Rapid neural coding in the retina with relative spike latencies," *Science*, vol. 319, no. 5866, pp. 1108–1111, 2008.

[16] N. S. Harper and D. McAlpine, "Optimal neural population coding of an auditory spatial cue," *Nature*, vol. 430, no. 7000, pp. 682–686, 2004.

[17] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," *Neural Computation*, vol. 22, no. 2, pp. 467–510, 2010.

[18] R. Gütig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing–based decisions," *Nature Neuroscience*, vol. 9, no. 3, pp. 420–428, 2006.

[19] X. Lin, X. Wang, and Z. Hao, "Supervised learning in multilayer spiking neural networks with inner products of spike trains," *Neurocomputing*, vol. 237, pp. 59–70, 2017.

[20] F. Zenke and S. Ganguli, "SuperSpike: Supervised learning in multilayer spiking neural networks," *Neural Computation*, vol. 30, no. 6, pp. 1514–1541, 2018.

[21] Y. Xing, G. Di Caterina, and J. Soraghan, "A new spiking convolutional recurrent neural network (SCRNN) with applications to event-based hand gesture recognition," *Frontiers in Neuroscience*, vol. 14, pp. 1143, 2020.

[22] Y. Luo, M. Xu, C. Yuan, X. Cao, Y. Xu, T. Wang, and Q. Feng, "SiamSNN: Siamese spiking neural networks for energy-efficient object tracking," in *Proceedings of the 30th International Conference on Artificial Neural Networks (ICANN)*, 2021, pp. 182–194.

[23] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3227–3235, 2017.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] D. Rzepka, M. Miśkowicz, D. Kościelnik, and N. T. Thao, "Reconstruction of signals from level-crossing samples using implicit information," *IEEE Access*, vol. 6, pp. 35001–35011, 2018.

[26] S. Cohen, *Finding Color And Shape Patterns In Images*, Ph.D. thesis, Stanford University, 1999.

[27] T. Tieleman and G. Hinton, "Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[28] E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs, "Event-driven contrastive divergence for spiking neuromorphic systems," *Frontiers in Neuroscience*, vol. 7, pp. 272, 2014.

[29] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, pp. 99, 2015.

[30] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–8.