PROCEEDINGS OF SCIENCE

A spiking neural network with fixed synaptic weights based on logistic maps for a classification task

Alexander G. Sboev,[a,b,c] **Dmitriy E. Kunitsyn,**[a,c,*] **Alexey V. Serenko**[a] **and Roman B. Rybka**[a,b]

[a]*NRC "Kurchatov Institute",*
Akademika Kurchatova sq., 1, Moscow, Russian Federation
[b]*Russian Technological University "MIREA",*
Vernadsky av., 78, Moscow, 119454, Russian Federation
[c]*NRNU "MEPhi",*
Kashira Hwy, 31, Moscow, Russian Federation
E-mail: sag111@mail.ru, selibrin@mail.ru, serenko@phystech.edu, rybkarb@gmail.com

Spiking neural networks are increasingly popular for machine learning applications, thanks to ongoing progress in the hardware implementation of spiking networks in low-energy-consuming neuromorphic hardware. Still, obtaining a spiking neural network model that solves a classification task with the same level of accuracy as a artificial neural network remains a challenge. Of especial relevance is the development of spiking neural network models trained on base of local synaptic plasticity rules that can be implemented either in digital neuromorphic chips or in memristive devices. However, existing spiking networks with local learning all have, to our knowledge, one-layer topology, and no multi-layer ones have been proposed so far. As an initial step towards resolving this problem, we study the possibility of using a non-trainable layer of spiking neurons as an encoder layer within a prospective multi-layer spiking neural network, implying that the prospective subsequent layers could be trained on base of local plasticity. We study a spiking neural network model with non-trainable synaptic weights preset on base of logistic maps, similarly to what was proposed recently in the literature for formal neural networks. We show that one layer of spiking neurons with such weights can transform input vectors preserving the information about the classes of the input vectors, so that this information can be extracted from the neuron's output spiking rates by a subsequent classifier, such as Gradient Boosting. The accuracy obtained on the Fisher's Iris classification task is 95%, with the deviation range of 5% over the five cross-validation folds. This is on par with other existing methods for Fisher's Iris classification with spiking neural networks, which shows the prospective possibility of using the proposed layer as an encoder within a multi-layer network.

**** The 6th International Workshop on Deep Learning in Computational Physics (DLCP2022) ****
**** 6-8 July 2022 ****
**** JINR, Dubna, Russia ****

[*]Speaker

PoS(DLCP2022)010

Introduction

Spiking neural networks [1] are biologically-inspired neural network models relevant for machine learning applications due to ultra-low power consumption of neuromorphic hardware on which such networks can be deployed [2, 3]. However, a universally robust method for training spiking neural networks still remains an open question [4].

There exist approaches for obtaining spiking neural networks that solve a classification task on the base of a trained formal neural network [5], and approaches for training spiking networks with error backpropagation [6]. These approaches allow creating arbitrarily deep spiking networks [7] but complicate the implementation of learning directly on the chip, and therefore cannot benefit from the advantages of hardware implementation at the training stage.

More desirable are training algorithms based on local plasticity rules, in which updating a synaptic weight only requires information from its adjacent neurons. However, many existing spiking neural network learning methods based on synaptic plasticity [8–10] employ non-local plasticity rules, involving e. g. aggregation of updates for weights of different neurons having the same convolution kernel.

Local learning methods based on Hebbian plasticity rules, such as Spike-Timing-Dependent plasticity (STDP), proved to be implementable either in digital neurochips such as Loihi [11] or in analog devices, such as memristive crossbars [12–14]. There, learning can be based on the rate selectivity of the Hebbian plasticity [15–18], or on the ability of STDP to stabilize the output spiking rate of a neuron [19], or on the ability of STDP to make a neuron selective to repeating input spike patterns [19, 20], or on the sensitivity of STDP to correlation in the input spike sequences [21].However, learning methods based on local plasticity rules have only been shown for networks of relatively shallow topologies.

Thus, a multi-layer spiking neural network, in which the first layers would transform the input data and the subsequent layers would train under local plasticity rules to perform the classification, remains a relevant goal. As an initial step towards that goal, the current paper studies whether a layer of spiking neurons with non-trainable weights could serve as an encoder.

The layer of spiking neurons have their synaptic weights set on base of logistic functions as proposed recently in the literature for artificial neural networks [22]. The benchmark input data for classification is Fisher's Iris, encoded with mean spiking rates. Then, the Gradient Boosting classifier is trained to predict the classes by the output spiking rates of the neurons, achieving accuracy on par with existing spiking neural network-based approaches. This proves that the output spiking rates of the proposed layer retain the information on the classes of its input data, thus suggesting the possible usage of such layer as an encoder in a multi-layer network.

1. Input data

For a benchmark classification task we use Fisher's Iris [23]: 150 samples of three species of Iris flowers – Iris setosa, Iris virginica, and Iris versicolor – 50 of each species. A sample is a vector of four rational values describing different measurements of a flower: sepal length, sepal width, petal length, and petal width. Iris setosa is linearly separable from the other two classes, while the latter are not linearly separable.

Before encoding the vectors $\vec{x}$ of the dataset into the input of the spiking neurons, they are minmaxscale-normalized, so that each of the four components x_i, $i = 1 \ldots 4$, will range from 0 to 1 in the training set X:

$$x_i \leftarrow \frac{x_i - \min\limits_{\vec{u} \in X} u_i}{\max\limits_{\vec{u} \in X} u_i - \min\limits_{\vec{u} \in X} u_i},$$

and then preprocessed with Gaussian receptive fields, each component of the normalized vector being replaced by M values based on the distance of the component value x_i to the center of the corresponding receptive field μ_j, $j = 1 \ldots M$:

$$\vec{x} \leftarrow \left\{ e^{\left(\frac{x_i - \mu_j}{\sigma}\right)^2}, \quad i - 1 \ldots 4, \quad j = 1 \ldots M \right\},$$

where $\mu_j = \frac{j}{M-1}$, and $\sigma = \frac{1}{2} \cdot \frac{1}{M-2}$ [24, 25].

The input vectors are thus transformed from their original dimensionality of 4 to a higher dimensionality of $K = 4 \cdot M$, where the number of the receptive fields M is an adjustable parameter, and its optimal value for this task has been found to be 4.

This choice of preprocessing is motivated by our prior work [19], where minmaxscale normalization and Gaussian receptive fields proved to be efficient for a spiking neural network on the Fisher's Iris classification task. Whether the proposed layer can work as an encoder without Gaussian receptive fields is a direction of further work.

2. Model

2.1 Spiking neuron layer setup

The proposed layer consists of N spiking neurons, the optimal value found for N being 33. Each of the $K = 4 \cdot M = 16$ input vector components is assigned a Poisson generator, and during presenting an input vector $\vec{x}$, the i-th generator emits spikes with the mean rate $r \cdot x_i$, where $r = 25\,642$ Hz, for $T = 5$ s.

All generators are connected to all neurons via synapses with fixed weights. The efficacy of a synapse connecting i-th input generator to j-th neuron the strength of which is characterized by synaptic weights fixed according to the following rule, borrowed from the literature on artificial neural networks [22]:

$$w_{i\,1} = A \cdot \sin\left(\frac{i \cdot \pi}{K \cdot B}\right), \quad w_{i\,j+1} = 1 - R \cdot w_{i\,j}^2.$$

Here, $A = 0.3$ and $B = 5.9$ as in the original paper [22], and the optimal value of r has been found at 1.68.

The output of the proposed layer is an N-dimensional vector of spike counts of each of the neurons during the time T of presenting an input vector.

2.2 Neuron model

The dynamics of a spiking neuron j in the proposed layer is governed by the Leaky Integrate-and-Fire model [26], in which the state of a neuron is described by its membrane potential $V_j(t)$:

$$C_\mathrm{m}\frac{dV_j}{dt} = \frac{V_\mathrm{rest} - V_j(t)}{\tau_\mathrm{m}} + I^j_\mathrm{syn}(t),$$

where $C_\mathrm{m} = 262.55$ pF, $V_\mathrm{rest} = -70$ mV, $\tau_\mathrm{m} = 5$ ms. As soon as $V(t) \geqslant V_\mathrm{th} = -66.98$ mV, the neuron fires a spike, after which $V(t)$ is instantaneously reset to V_rest and is clamped to it during the refractory period $\tau_\mathrm{ref} = 2$ ms.

I^j_syn is the incoming current from the synapses, to which an input spike emitted by i-th input generator at time t^i_sp adds an exponential pulse:

$$I^j_\mathrm{syn} = \sum_i w_{ij} \sum_{t^i_\mathrm{sp}} \frac{q_\mathrm{syn}}{\tau_\mathrm{syn}} e^{-\frac{t-t^i_\mathrm{sp}}{\tau_\mathrm{syn}}}\Theta(t - t^i_\mathrm{sp}),$$

where $q_\mathrm{syn} = 5$ fC, $\tau_\mathrm{syn} = 5$ ms, $\Theta(t)$ is the Heaviside step function, and w_{ij} is the weight of the synapse connecting i-th input generator to the current neuron.

This neuron model has been chosen over more complex and biologically plausible models, such as Hodgkin-Huxley [27] or FitzHugh-Nagumo [28], following our prior work [29] on training spiking neural networks to solve classification tasks. The constants have been set in accordance with the latter, except the membrane capacity C_m and the threshold V_th which have been adjusted for the current classification task.

3. Experiments

In order to assess the feasibility of the proposed spiking layer as an encoder, the Gradient Boosting classifier (from the scikit-learn library [30], with the number of estimators being 1000, the learning rate of 0.01, and the maximum depth of 15) is used in this work as a temporary, test-bench replacement for the prospective subsequent decoder layers. The performance of the encoder layer is therefore characterized by the accuracy with which Gradient Boosting is able to decode the classes of Fisher's Iris, receiving as its input the vectors of spike counts from the encoder layer.

Classification performance is assessed using 5-fold cross-validation: the 150 vectors of the dataset are split into 5 non-overlapping subsets, one subset containing 10 vectors from each class. In different folds, a different subset is considered the testing set, and the remaining subsets form the training set. For each fold, the classification performance assessment – calculating the normalization coefficients for the training set, training the Gradient Boosting on the outputs of the spiking layer in response to the training set, and testing the accuracy of Gradient Boosting on the spiking layer outputs in response to the testing set – are performed independently.

F1-macro is used as the metric for classification accuracy. The optimal values of the model parameters mentioned in the preceding sections as adjustable have been found using hyperopt [31] so that to maximize the F1-macro score on the training set averaged over all folds.

PoS(DLCP2022)010

4. Results

The classification performance of the proposed approach, measured by the F1-macro score on the testing set averaged over the testing sets of all folds, is presented in Table 1. For comparison, the accuracy of an artificial neural network is cited, and a few existing classification methods based on spiking neural networks. Though the performance of the proposed spiking layer is inferior to an artificial neural network [32], it is on par with a spiking neural network trained by error backpropagation [33], and with a spiking neural network [19] trained by the local learning mechanism of Spike-Timing-Dependent Plasticity.

Table 1: The F1-macro of Fisher's Iris classification by the proposed spiking layer with Gradient Boosting decoding (mean and deviation range over 5 cross-validation folds) and by several other existing approaches

Classification approach	F1-macro, %
Proposed approach	95 ± 5
Spiking neural network trained by a local learning rule [19]	97 ± 3
Spiking neural network trained by backpropagation [33]	96
4-layer artificial neural network [32]	100

.

Conclusion

A layer of spiking neurons with synaptic weights fixed based on logistic functions can transform real-valued vectors of Fisher's Iris so that information about their classes is preserved in the neurons' output spiking rates, and can be extracted using Gradient Boosting. The classification performance thus obtained is on par with other spiking neural network learning methods.

This result constitutes evidence that spiking neurons with non-trainable weights can be used to transform input vectors not impeding, and hopefully facilitating, their further classification. A layer of such neurons could prospectively be employed as an encoder within a multi-layer network. In the future work, we are planning to assess the necessity of Gaussian receptive fields preprocessing, and the possibility of employing a trainable spiking layer as the decoder.

Acknowledgments

This work has been carried out using computing resources of the federal collective usage center Complex for Simulation and Data Processing for Mega-science Facilities at NRC "Kurchatov Institute", http://ckp.nrcki.ru/.

References

[1] H. Paugam-Moisy and S.M. Bohte, *Computing with spiking neuron networks*, in *Handbook of Natural Computing*, G. Rozenberg, T. Back and J. Kok, eds., pp. 335–376, Springer Berlin Heidelberg (2012), DOI.

[2] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa and E. Eleftheriou, *Low-power neuromorphic hardware for signal processing applications: A review of architectural and system-level design approaches*, *IEEE Signal Processing Magazine* **36** (2019) 97.

[3] S. Furber, *Large-scale neuromorphic computing systems*, *Journal of neural engineering* **13** (2016) 051001.

[4] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L.P. Maguire and T. McGinnity, *A review of learning in biologically plausible spiking neural networks*, *Neural Networks* **122** (2020) 253.

[5] P.U. Diehl, G. Zarrella, A. Cassidy, B.U. Pedroni and E. Neftci, *Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware*, in *IEEE International Conference on Rebooting Computing (ICRC)*, pp. 1–8, IEEE, 2016.

[6] J.H. Lee, T. Delbruck and M. Pfeiffer, *Training deep spiking neural networks using backpropagation*, .

[7] A. Tavanaei, M. Ghodrati, S.R. Kheradpisheh, T. Masquelier and A. Maida, *Deep learning in spiking neural networks*, *Neural Networks* **111** (2019) .

[8] S.R. Kheradpisheh, M. Ganjtabesh, S.J. Thorpe and T. Masquelier, *STDP-based spiking deep convolutional neural networks for object recognition*, *Neural Networks* **99** (2018) 56 .

[9] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S.J. Thorpe and T. Masquelier, *Combining STDP and reward-modulated STDP in deep convolutional spiking neural networks for digit recognition*, *arXiv preprint arXiv:1804.00227* (2018) .

[10] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S.J. Thorpe and T. Masquelier, *Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks*, *Pattern Recognition* **94** (2019) 87 .

[11] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S.H. Choday et al., *Loihi: A neuromorphic manycore processor with on-chip learning*, *IEEE Micro* **38** (2018) 82.

[12] S. Saïghi, C.G. Mayr, T. Serrano-Gotarredona, H. Schmidt, G. Lecerf, J. Tomas et al., *Plasticity in memristive devices for spiking neural networks*, *Frontiers in Neuroscience* **9** (2015) 51.

[13] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri and B. Linares-Barranco, *STDP and STDP variations with memristors for spiking neuromorphic learning systems*, *Frontiers in neuroscience* **7** (2013) 2.

[14] B.S. Shvetsov, A.A. Minnekhanov, A.V. Emelyanov, A.I. Ilyasov, Y.V. Grishchenko, M.L. Zanaveskin et al., *Parylene-based memristive crossbar structures with multilevel resistive switching for neuromorphic computing*, *Nanotechnology* **33** (2022) 255201.

[15] P.U. Diehl and M. Cook, *Unsupervised learning of digit recognition using Spike-Timing-Dependent Plasticity*, *Frontiers in Computational Neuroscience* (2015) .

[16] D. Querlioz, O. Bichler, P. Dollfus and C. Gamrat, *Immunity to device variations in a spiking neural network with memristive nanodevices*, *IEEE Transactions on Nanotechnology* **12** (2013) 288.

[17] V. Demin and D. Nekhaev, *Recurrent spiking neural network learning based on a competitive maximization of neuronal activity*, *Frontiers in Neuroinformatics* **12** (2018) 79.

[18] V. Demin, D. Nekhaev, I. Surazhevsky, K. Nikiruy, A. Emelyanov, S. Nikolaev et al., *Necessary conditions for STDP-based pattern recognition learning in a memristive spiking neural network*, *Neural Networks* **134** (2021) 64.

[19] A. Sboev, A. Serenko, R. Rybka and D. Vlasov, *Solving a classification task by spiking neural network with STDP based on rate and temporal input encoding*, *Mathematical Methods in the Applied Sciences* **43** (2020) 7802.

[20] T. Masquelier, R. Guyonneau and S.J. Thorpe, *Spike Timing Dependent Plasticity finds the start of repeating patterns in continuous spike trains*, *PLoS One* **3** (2008) e1377.

[21] A. Sboev, A. Serenko and R. Rybka, *Correlation encoding of input data for solving a classification task by a spiking neural network with spike-timing-dependent plasticity*, in *Biologically Inspired Cognitive Architectures*, no. 1032 in Studies in Computational Intelligence, pp. 457–462, Springer International Publishing, 2022, DOI.

[22] A. Velichko, *Neural network for low-memory iot devices and mnist image recognition using kernels based on logistic map*, *Electronics* **9** (2020) .

[23] R.A. Fisher, *The use of multiple measurements in taxonomic problems*, *Annual Eugenics* **7** (1936) 179 .

[24] Q. Yu, H. Tang, K.C. Tan and H. Yu, *A brain-inspired spiking neural network model with temporal encoding and learning*, *Neurocomputing* **138** (2014) 3.

[25] X. Wang, Z.-G. Hou, F. Lv, M. Tan and Y. Wang, *Mobile robots' modular navigation controller using spiking neural networks*, *Neurocomputing* **134** (2014) 230.

[26] A.N. Burkitt, *A review of the integrate-and-fire neuron model: II. inhomogeneous synaptic input and network properties*, *Biological cybernetics* **95** (2006) 97.

[27] A.L. Hodgkin and A.F. Huxley, *A quantitative description of membrane current and its application to conduction and excitation in nerve*, *Journal of Physiology* **117** (1952) 500.

[28] N.A. Kudryashov, R.B. Rybka and A.G. Sboev, *Analytical properties of the perturbed FitzHugh-Nagumo model*, *Applied Mathematics Letters* **76** (2018) 142.

[29] A. Sboev, R. Rybka, A. Serenko, D. Vlasov, N. Kudryashov and V. Demin, *To the role of the choice of the neuron model in spiking network learning on base of Spike-Timing-Dependent Plasticity*, in *8th Annual International Conference on Biologically Inspired Cognitive Architectures (BICA)*, V.V. Klimov and A.V. Samsonovich, eds., vol. 123 of *Procedia Computer Science*, pp. 432–439, Elsevier BV, 2018, DOI.

PoS(DLCP2022)010

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al., *Scikit-learn: Machine learning in Python*, *Journal of Machine Learning Research* **12** (2011) 2825.

[31] J. Bergstra, D. Yamins and D. Cox, *Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures*, in *Proceedings of the 30th International Conference on Machine Learning*, S. Dasgupta and D. McAllester, eds., vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 115–123, PMLR, 17–19 Jun, 2013, https://proceedings.mlr.press/v28/bergstra13.html.

[32] P. Patil and T. Sontakke, *Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network*, *Pattern Recognition* **40** (2007) 2110 .

[33] A. Tavanaei and A. Maida, *BP-STDP: Approximating backpropagation using spike timing dependent plasticity*, *Neurocomputing* **330** (2019) .