Nonlinear Dynamics and Neuroscience

Izvestiya Vysshikh Uchebnykh Zavedeniy. Applied Nonlinear Dynamics. 2024;32(2)

Article DOI: 10.18500/0869-6632-003094

Spiking neural network with local plasticity and sparse connectivity for audio classification*

R. B. Rybka[1,2]✉, *D. S. Vlasov*[1], *A. I. Manzhurov*[1], *A. V. Serenko*[1], *A. G. Sboev*[1,2]

[1]National Research Centre "Kurchatov Institute", Moscow, Russia
[2]National Research Nuclear University "MEPhI", Moscow, Russia
E-mail: ✉rybka_rb@nrcki.ru, vfked0d@gmail.com, manzhurov99@mail.ru, serenko@phystech.edu, sag111@mail.ru

Received 22.09.2023, accepted 5.12.2023, available online 1.03.2024, published 29.03.2024

Abstract. *Purpose.* Studying the possibility of implementing a data classification method based on a spiking neural network, which has a low number of connections and is trained based on local plasticity rules, such as Spike-Timing-Dependent Plasticity. *Methods.* As the basic architecture of a spiking neural network we use a network included an input layer and layers of excitatory and inhibitory spiking neurons (Leaky Integrate and Fire). Various options for organizing connections in the selected neural network are explored. We have proposed a method for organizing connectivity between layers of neurons, in which synaptic connections are formed with a certain probability, calculated on the basis of the spatial arrangement of neurons in the layers. In this case, a limited area of connectivity leads to a higher sparseness of connections in the overall network. We use frequency-based coding of data into spike trains, and logistic regression is used for decoding. *Results.* As a result, based on the proposed method of organizing connections, a set of spiking neural network architectures with different connectivity coefficients for different layers of the original network was implemented. A study of the resulting spiking network architectures was carried out using the Free Spoken Digits dataset, consisting of 3000 audio recordings corresponding to 10 classes of digits from 0 to 9. *Conclusion.* It is shown that the proposed method of organizing connections for the selected spiking neural network allows reducing the number of connections by up to 60% compared to a fully connected architecture. At the same time, the accuracy of solving the classification problem does not deteriorate and is 0.92...0.95 according to the F1 metric. This matches the accuracy of standard support vector machine, k-nearest neighbor, and random forest classifiers. The source code for this article is publicly available: https://github.com/sag111/Sparse-WTA-SNN.

Keywords: spiking neural network, STDP, sparse connectivity, free spoken digits dataset, audio classification.

Acknowledgements. The study was supported by a grant from the Russian Science Foundation № 21-11-00328, https://rscf.ru/project/21-11-00328/. Computational experiments were carried out using the equipment of the Center for Collective Use "Complex for Modeling and Processing Data of Mega-Class Research Installations" of the National Research Center "Kurchatov Institute", http://ckp.nrcki.ru/.

For citation: Rybka RB, Vlasov DS, Manzhurov AI, Serenko AV, Sboev AG. Spiking neural network with local plasticity and sparse connectivity for audio classification. Izvestiya VUZ. Applied Nonlinear Dynamics. 2024;32(2):239–252. DOI: 10.18500/0869-6632-003094

*The paper presents materials of a talk given at the conference "Neuroinformatics — 2023".

Introduction

One of the motivations for research on spiking neural networks (SNN) is to explore the possibility of utilizing the abilities of the brain of living organisms in computer models. This fundamental direction has recently acquired a more practical form, which is associated with progress in the creation of neuromorphic chips that allow simulating bioinspired spiking neural networks on energy-efficient computing devices [1,2]. While many existing approaches are focused on offline learning of SNN with subsequent transfer to neuromorphic chips, it is promising to create methods for this class of devices that would allow online learning. These include learning methods based on local synaptic plasticity, implemented using Spike-Timing-Dependent Plasticity (STDP), where the change in the weight of a synaptic connection is proportional to the time interval from the arriving of a presynaptic spike to emitting the postsynaptic spike. The relevance of STDP is due to the prospective possibility of hardware implementation of SNNs with STDP. Therein, a synapse with STDP could be implemented on base of a memristor [3–5], the change in conductivity of which depends on the duration of overlapping presynaptic and postsynaptic voltage pulses.

Currently, there are several methods for training spiking neural networks with STDP [6–9]. One of the efficient approaches for solving classification problems is the SNN network with STDP based on a three-layer architecture [10].

This architecture was previously used to classify images of handwritten digits [10, 11], real-valued vector data and audio information [12]. Despite its efficiency, it is quite resource-intensive, so the purpose of this paper is to study the possibility of reducing the number of connections in such an SNN. This formulation of the problem is due to the presence of restrictions on the number of connections of existing neurochips, which makes it relevant to reduce the computational complexity of SNN models.

The principles of sparse connectivity between neurons were studied earlier in a number of works on spiking and other artificial neural networks. For example, a $\sim 70\%$ reduction in connectivity by zeroing out weights that are below a given threshold in a network [13] consisting of two convolutional layers that process input data and then pass it on to spiking convolutional layers allows to reduce power consumption while maintaining the accuracy of image recognition on video in the IVS 3cls [14] dataset at the level of 71.5%. In [15], limiting the number of connections per neuron by about 50% is shown to reduce network power consumption and, at the same time, by training a multilayer convolutional network using the backpropagation method, achieve good performance on MNIST problems (99.51%), CIFAR-10 (94.10%), N-MNIST (99.53%), DVS-Gesture (98.20%). An SNN [16] in which connections between neurons are set in a probabilistic way based on the spatial coordinates of neurons, shows the accuracy of 97.8% on the handwritten digit and letter classification task from the EMNIST dataset.

Due to the possibility of spatial localization of neurons in layers in the chosen SNN architecture, in this work we have chosen an approach to establish sparse connectivity based on the probabilistic formation of connections in a given area. For this method, the effectiveness of using sparse connections between different layers of the original SNN is investigated. According to experts [17], the greatest effect from the use of SNNs implemented on neurochips is achieved when analyzing streaming data, an example of which is audio data. Therefore, in this work, we used the Free Spoken Digit Dataset (FSDD), an open benchmark for audio classification algorithms, as data for the study.

The main contribution of this article is:

- the effectiveness of the learning method based on local plasticity for the SNN with sparse connections was evaluated using a set of audio data,

- the impact of the SNN connectivity level on the accuracy of the audio data classification problem was assessed,
- the importance of using sparse connections between different layers of a three-layer SNN was determined.

The article is organized as follows: Section “Data and preprocessing” describes the dataset, methods for extracting significant features of audio and transforming at spike moments; Section “Spiking Neural Network” describes the models of neurons and synapses, the architecture of the spiking network, and the learning algorithm; Section “Experiments” presents the results of experimental studies of the selected sparse connectivity method for various configurations of SNN architectures; analysis of the results and comparison with other approaches is presented in Section “Analysis of results”.

1. Data and preprocessing

1.1. Dataset. The free-spoken-digit-dataset (FSDD) consisting of 3000 audio recordings of the pronunciation of numbers from 0 to 9 in English is considered as a test classification task for testing the proposed method. The FSDD dataset contains 10 classes of 300 WAV-format audio records, up to 1 second long. Examples of audio waveforms for some classes are shown in Fig. 1.

The dataset was formed as follows: 6 people pronounced the numbers from zero to nine 50 times with different intonations and speed. In order to be able to consistently compare the accuracy when classifying the dataset by various machine learning methods, the following splitting of data into test and training samples is recommended by default: the first 5 out of 50 (10%) audio pronunciations by each person in all classes are assigned to the test sample, the remaining 45 audio (90%) — to the training sample.

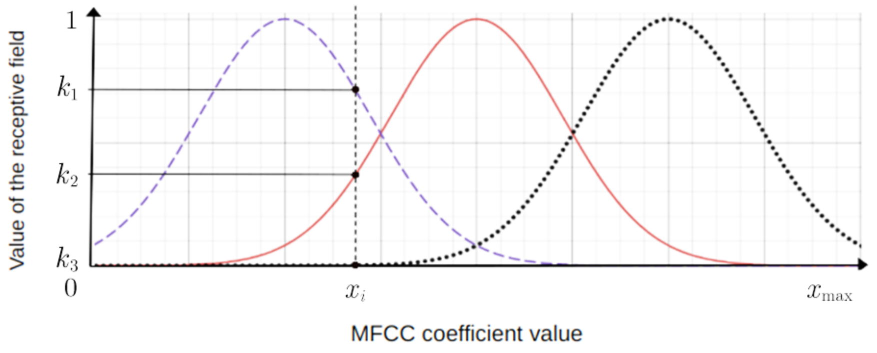

Fig. 1. An example of using three Gaussian receptive fields with different μ_j. $\mathbf{k}$ is the new vector of the x_i component of the input vector after GRF processing (color online)

1.2. Feature extraction. To feed data to the spiking neural network, a set of Mel-frequency cepstral coefficients (MFCC) was obtained for the audio records using the fast Fourier transform (FFT) and the discrete cosine transform. This is done using the open-source python package (ISC License) for music and audio analysis — librosa [18]. Further, the average value of each MFCC coefficients is calculated over audio length. The averaged values are normalized from 0 to 1. The result is a new vector of $K = 30$ averaged and normalized MFCC coefficients (other features were set by default from librosa package: the window size for Fourier transform is 250 ms, the stride is 64 ms).

1.3. Encoding into spike sequences. To improve classification accuracy when using SNN, at the preprocessing stage, the resulting feature vector is processed using M Gaussian receptive fields (GRF) (see Eq. (1)). To do this, the interval of values of the averaged and normalized MFCC coefficients is divided into M equal intervals. In each interval $j = 1, \ldots, M$, a Gaussian peak with center μ_j in the middle of the j-th interval. The value of the component x_i of the input vector is replaced by a set of values $G_j(x_i)$ characterizing the proximity of x_i to the center of the j-th receptive field:

$$G_j(x_i) = \exp\left(\frac{(x_i - \mu_j)^2}{\sigma^2}\right). \tag{1}$$

The value of the component x_i of the input vector is replaced by a set of values $G_j(x_i)$ characterizing the proximity of x_i to the center of the j-th receptive field. Thus, the dimension of the input increases by the factor of M. Fig. 1 shows an example for three receptive fields. In this research we used $M = 7$ as found to be optimal in an earlier work with the same dataset [12], $\sigma = \frac{2(\max_x - \min_x)}{3(M-2)}$, $\mu_j = \min_x + (\max_x - \min_x) \cdot \frac{j}{M-1}$, where $\max_x$ and $\min_x$ are maximum and minimum values of x_i among all training set vectors.

Frequency coding is used to transform the components of the input vector into sequences of spikes: each element of the input layer emits spikes with a certain frequency ν during the entire time $t_e = 350$ ms, of processing the current audio:

$$\nu = \nu_{\max} \cdot k, \tag{2}$$

where $\nu_{\max}$ is the maximum frequency constant, k is the value of the input vector component after preprocessing. After presenting an audio recording, the input are silent for $t_p = 50$ ms so as to allow the neuron potentials relax. The remaining parameters set during the experiments are presented in the Table 1.

Table 1. Results of the first stage of the research

Type of connection between layers of neurons				$\nu_{\max}$, Hz	**Layer grid**	**F1**
From excitatory to inhibitory	**From inhibitory to excitatory**	**From input to excitatory**	**From input to inhibitory**			
Counter-partnership	All-to-all*	All-to-all	Fixed Amount(10%)	550	—	0.93
Sparse ($P_{exc\_inh} = 0.6$; $R_{exc\_inh} = 0.25$)	All-to-all*	All-to-all	Fixed Amount(10%)	550	Regular	0.93
Sparse ($P_{exc\_inh} = 0.7$; $R_{exc\_inh} = 0.55$)				550	Irregular	0.93
Counter-partnership	Sparse ($P_{inh\_exc} = 0.4$; $R_{inh\_exc} = 0.7$)	All-to-all	Fixed Amount(10%)	550	Regular	0.94
	Sparse ($P_{inh\_exc} = 0.7$; $R_{inh\_exc} = 0.8$)			850	Irregular	0.93
Counter-partnership	All-to-all*	Probabilistic ($P_{gen\_exc} = 0.4$)	Fixed Amount(10%)	950	Not applicable	0.94
Counter-partnership	All-to-all*	All-to-all	Probabilistic ($P_{gen\_inh} = 0.2$)	550	Not applicable	0.62

* means excluding connections between counter-partners

2. Spiking Neural Network

2.1. Neuron model. The spiking neuron models in this work are Leaky Integrate-and-Fire (LIF), in which the state variable, membrane potential $V(t)$, changes in accordance with Eq. (3), as if the neuron's membrane was an electric capacitor with capacitance C_m and with a leakage that would drive the potential to its resting level V_rest in the characteristic time τ_m if the synaptic input $I_\mathrm{syn}(t)$ was absent.

$$\frac{dV}{dt} = -\frac{V(t) - V_\mathrm{rest}}{\tau_\mathrm{m}} + \frac{I_\mathrm{syn}(t)}{C_\mathrm{m}}. \tag{3}$$

The postsynaptic current $I_\mathrm{syn}(t)$ is described by the synaptic conductance model (see Eq. (4)): each incoming synapse i has the conductance $g_i(t) \cdot w_i(t)$, through which the neuron's membrane is charged by a voltage source with the potential $E_\mathrm{rev\_exc/inh}$.

$$I_\mathrm{syn}(t) = \sum_i w_i(t) g_i(t) \cdot \left(E_\mathrm{rev\_exc/inh} - V(t)\right). \tag{4}$$

The synaptic conductance increases by $\sigma_\mathrm{syn\_exc/inh}$ whenever a spike arrives (let the input spike times be denoted t_pre), and then relaxes to zero (see Eq. (5)). The trainable strength of the synapse is modelled by modulating the synaptic conductance with the non-dimensional weight $w_i(t)$ in the range $[0;1]$.

$$\frac{dg_i(t)}{dt} = -\frac{g_i(t)}{\tau_\mathrm{syn\_exc/inh}} + \sigma_\mathrm{syn\_exc/inh} \cdot \delta\left(t - t_\mathrm{pre}\right). \tag{5}$$

As soon as $V(t)$ reaches the threshold V_th, the neuron fires an output spike, and $V(t)$ is instantaneously reset to V_reset. After a spike, a refractory period t_ref begins during which the input spikes have no effect on the neuron. The operation of a LIF neuron is schematically depicted in Figure 2.

In order to prevent the neurons from firing too many or too few spikes, the threshold potential is adaptive, increasing by Θ^+ whenever a spike is fired and gradually relaxing to Θ_rest:

$$\frac{dV_\mathrm{th}}{dt} = -\frac{V_\mathrm{th}(t) - \Theta_\mathrm{rest}}{dt} + \Theta^+ \cdot H(V(t) - V_\mathrm{th}(t)), \tag{6}$$

where H is the Heaviside step function.

The constants of the neuron and synapse models are chosen following prior work on similar SNN architectures [10,12]. Their values, different for different layers of the network, are in Table 2. There, the values of the postsynaptic current constants $E_\mathrm{rev\_exc/inh}$, $\sigma_\mathrm{syn\_exc/inh}$, and $\tau_\mathrm{syn\_exc/inh}$ depend also on whether their synapse i is excitatory or inhibitory, and are denoted with subscripts syn_exc or syn_inh respectively.

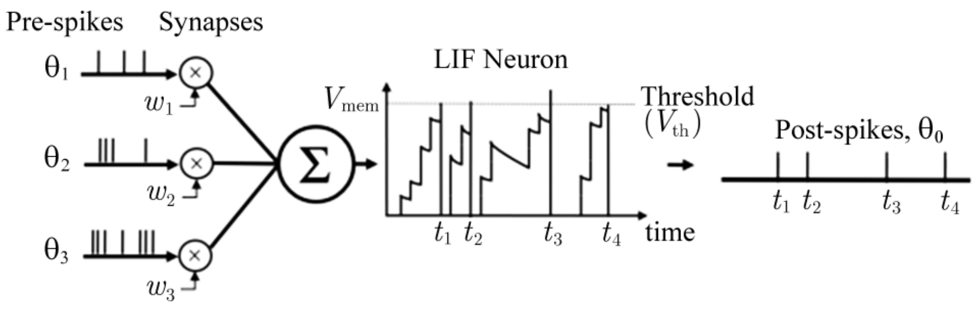

Fig. 2. An example of the Leaky Integrate-and-Fire neuron dynamics

Table 2. Neuron model constants for neurons of excitatory and inhibitory layers

Parameter	**Exc. neurons**	**Inh. neurons**
Refractory period t_{ref}, ms	4	3
Membrane leakage τ_{m}, ms	130	30
Membrane capacitance C_{m}, pF	100	10
Synaptic conductance increment q_{syn}, S	1	1
Conductance decay constant for exc. synapses $\tau_{\text{syn\_exc}}$, ms	1	1
Conductance decay constant for inh. synapses $\tau_{\text{syn\_inh}}$, ms	2	2
Dynamic threshold resting value Θ_{rest}, mV	-72	-40
Dynamic threshold increment Θ_{+}, mV	0.05	0
Membrane potential resting value V_{rest}, mV	-65	-45
Initial threshold $V_{\text{th}}(t=0)$, mV	-52	-40
Reversal potential for excitatory synapses $E_{\text{rev\_exc}}$, mV	0	0
Reversal potential for inhibitory synapses $E_{\text{rev\_exc}}$, mV	-160	-160

2.2. Connections between neurons. Synaptic connections in SNNs ensure the transfer of information between neurons via spikes. Connections between neurons can be with static or dynamic weights. The weight of the connection, multiplied by the spike passing through it, determines the contribution of the spike to the change in the membrane potential of the postsynaptic neuron.

Synapses with constant weights do not change their value when simulating an SNN. Dynamic weights change according to the law of plasticity. Spike Timing Dependent Plasticity (STDP) [19] is used as a plasticity model in this paper. In this approach, the weight of the synapse is adjusted depending on the relative arrival time of the pre- and post- synaptic spikes within a short time interval (tens of milliseconds). If the presynaptic neuron emits a spike just before the postsynaptic neuron sends its own, then the weight of the connection will increase. In the opposite case, if the presynaptic neuron emitted a spike already after the postsynaptic neuron, then the weight of this synapse will decrease. Thus, the change in weight is described by the following formulas: [19] according to Eq. (7) each time an input spike arrives at t_{pre} or a postsynaptic spike occurs at t_{post}:

$$\Delta w = \begin{cases} -A^{-} \cdot \exp\left(\frac{t_{\text{post}} - t_{\text{pre}}}{\tau^{-}}\right) & \text{if } t_{\text{post}} > t_{\text{pre}}; \\ A^{+} \cdot \exp\left(-\frac{t_{\text{pre}} - t_{\text{post}}}{\tau^{+}}\right) & \text{if } t_{\text{post}} < t_{\text{pre}}. \end{cases} \tag{7}$$

If $t_{\text{post}} = t_{\text{pre}}$, such pair of spikes is by convention excluded from consideration and does not cause any weight change.

2.3. SNN architecture. The SNN considered in this study (see Fig. 3) is a modification of an SNN [10] consisting of three layers, input, excitatory and inhibitory. The input layer consists of $K \cdot M$ spike emitters, one for each component of the preprocessed input vector, that emit spikes with the mean rate depending on the values of the corresponding input vector components, as described in Section “Encoding into spike sequences”. Spikes emitted by the input layer arrive at synapses that connect the input layer to the excitatory and inhibitory layers. These layers contain The excitatory layer processes incoming spike sequences from the input layer. The connections from the input spike emitters to the excitatory neurons (the topology of which is described in more detail in Section “Sparse connectivity”) are excitatory, and have their weights changed by Spike-Timing-Dependent Plasticity (STDP). The inhibitory layer is used to create competition between neurons in the excitatory layer. For each excitatory neuron, there exists one associated inhibitory neuron, which receives spikes from it via an excitatory synapse with a fixed weight

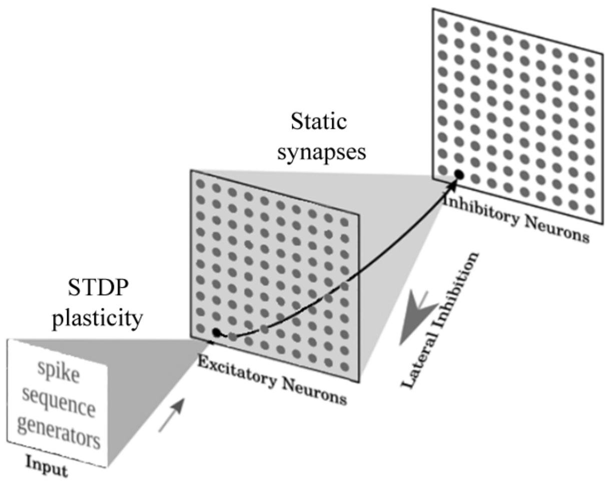

Fig. 3. Three-layer SNN architecture

$w_{\mathrm{exc-inh}} > 0$ (marked with a black arrow in Fig. 3). This inhibitory neuron is connected to all other neurons of the excitatory layer, except the one it gets spikes from, by static inhibitory connections with a fixed weight $w_{\mathrm{inh-exc}} < 0$ (in Fig. 3 indicated by the area between the inhibitory and excitatory layers). Additionally, for greater activity of neurons in the inhibitory layer, static connections are introduced from the input layer with weights $w_{\mathrm{input-inh}} > 0$.

2.4. Sparse connectivity. In the spiking neural network used, the layers of excitatory and inhibitory neurons are two-dimensional square areas, located mirror-symmetrically relative to each other. Neurons in layers can be located inside a square layer either randomly (irregular grid) or structured at the nodes of a regular square grid. The formation of sparse connections between the layers of excitatory and inhibitory neurons occurs as follows:

1. The presynaptic neuron (from which connections will begin) is projected onto the square area of the postsynaptic neurons layer (with which the connection is established).
2. The projection of the presynaptic neuron will be the center of a circle of a certain radius on the postsynaptic layer. Only those neurons of the postsynaptic layer that fell into the region of the circle can, with some probability, establish a connection with the presynaptic neuron.

This process is shown in Figure 4 and is carried out for all neurons of the presynaptic layer.

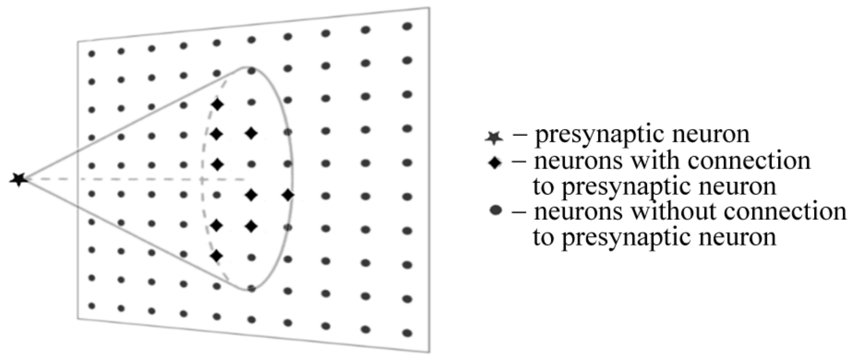
✶ – presynaptic neuron
◆ – neurons with connection to presynaptic neuron
● – neurons without connection to presynaptic neuron

Fig. 4. An example of the sparse connectivity method

Thus, the sparse connectivity method is characterized by the following parameters:

- probability P of the formation of a connection between neurons of different layers falling into the specified areas;
- the radius R of the area for connections;
- the regularity or irregularity of the network of neurons in the layer to which the connection is established.

In this work, this method is used to organize connections between the following layers of neurons:

- From the layer of excitatory neurons to the layer of inhibitory ones. For synapses with static positive weights ($W_{exc\_inh} > 0$), the following configuration parameters are selected: connection formation probability (P_{exc_inh}), connection region radius (R_{exc_inh}) and the grid regularity of the layer of inhibitory neurons.
- From the layer of inhibitory neurons to excitatory ones. For synapses with static negative weights ($W_{inh\_exc} < 0$), the following configuration parameters are selected: connection formation probability (P_{inh_exc}), connection region radius (R_{inh_exc}) and regularity of the excitatory neurons layer grid.

The size of the 2D areas for layers of excitatory and inhibitory neurons was initially set to select the R coefficient in the conducted experiments (1 mm $\times$ 1 mm). The size can be set to any size; what is important is the ratio of size and radius, which characterizes the sparse connectivity.

For the input layer, consisting of spike emitters, a clear position in space is not specified since when establishing connections from emittors, only the probability of its formation is used, and the projection of the neuron on the postsynaptic layer is not built. In this case, P_{gen_inh} and P_{gen_exc} are adjustable hyperparameters, and values found for them will be presented below, in Table 1.

2.5. SNN learning. The process of adjusting the weights during training in the layer with STDP (from input to excitatory neurons) is performed using Algorithm.

Algorithm. SNN learning process

```
Input: training data matrix $X^{train}$ of preprocessed input vectors $\mathbf{x}_i$ of each audio in dataset,
  neuron parameters, plasticity parameters, synapse parameters, initial weight distribution
Optimized parameters: $N_{epoch}$, $\nu_{max}$, $P_{exc\_inh}$, $P_{inh\_exc}$, $P_{gen\_exc}$, $P_{gen\_inh}$, $R_{exc\_inh}$,
  $R_{inh\_exc}$
Constant network parameters: Table 3
Output data: SNN model, vector of neuron activity frequencies in the excitatory layer for
  each example of the training set $\mathbf{v}_i$.
 1: Neural network initialization: neurons, synapses and initial weights.
 2: for $k$ in $N_{epoch}$ do
 3:     for each $\mathbf{x}_i$ in $X^{train}$ do
 4:         for each $k_{i\_j}$ in $\mathbf{x}_i$ do
 5:             Generating spikes sequences $\mathbf{x}_{i\_j}^{seq}$ with length $t_e$ and frequency $\nu_{i\_j}$.
 6:         end for
 7:         Simulating SNN during $t_e$ time steps using spikes sequences array $\mathbf{x}_i^{seq}$.
 8:         Simulating SNN without inpus signal during $t_p$ time steps for membrane potential resting
            to initial value.
 9:     end for
10: end for
11: Stop changing weights.
12: Collecting and saving frequencies vector of excitatory neuron layer activities $\mathbf{v}_i$ during presenting
    samples of input data.
13: Return SNN model, vector of neuronal activities frequencies $\mathbf{v}_i$.
```

Table 3. Network and synapse parameters

Parameter	Value	Description
$w_{\text{exc}-\text{inh}}$	13	Static synaptic weights from exc to inh neurons
$w_{\text{inh}-\text{exc}}$	-12	Static synaptic weights from inh to exc neurons
t_p	50 ms	Intervector pause
t_e	350 ms	Spike train length
N	400	Number of neurons in every layer
A^-	0.55	STDP weight depression amplitude
A^+	1.0	STDP weight potentiation amplitude
τ_-	20 ms	STDP depression time window constant
τ_+	20 ms	STDP depression time window constant

After obtaining the trained SNN, the general process of classifying an audio recording consists of the following steps:

1) preprocessing of audio test samples using MFCC and GRF methods,
2) conversion of the received values into spike sequences,
3) simulation of the SNN and calculation of the excitatory neuron activity frequency vectors,
4) defining audio sample class from frequency vector.

3. Experiments

The experiments were carried out in two stages. At the first stage, the feasibility of using sparse connectivity in different layers of the chosen SNN architecture was investigated.

To do this, sparse connectivity was applied in turn to the links between different layers. Based on the results of the experiments, layers were selected the use of sparse connectivity in which led to the best results. At the second stage of the experiments, we studied the application of sparse connectivity to several layers of the network at once. All experiments were carried out in the mode with automatic selection of hyperparameters based on the open-source HyperOpt library [20].

Measuring accuracy was performed using the micro-averaged F1-score metric which is calculated from the precision and recall of predicting that class (Eq. 8), where the precision is, for each class L, the number True Positive (TP_L) of samples from the class L predicted correctly as belonging to L, divided by the total number of samples predicted as belonging to L, including the False Positive (FP_L) samples that do not actually belong to L but were misattributed to it by the network. Recall is the number of true positive samples divided by the number of all samples belonging to L, including the False Negative (FP_L) samples that belong to L but were not identified as such by the network.

$$\begin{aligned} \text{Precision} &= \frac{\sum_L TP_L}{\sum_L TP_L + \sum_L FP_L}, \\ \text{Recall} &= \frac{\sum_L TP_L}{\sum_L TP_L + \sum_L FN_L}, \\ \text{F1} &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \end{aligned} \tag{8}$$

The results of the first stage of experiments are presented in the Table 1.

Baseline for comparison is the accuracy of the original SNN architecture, which was 0.93 F1 with the network parameters specified in the Table 3 and the maximum spike frequency $\nu_{\max} = 550$ Hz.

As can be seen from the Table 1, when replacing the initial connections with a fixed number (10% of all-to-all connections) between the input and inhibitory layers with sparse ones there is a significant decrease in the accuracy of audio classification (the best result on HyperOpt is 0.62 F1 at $P_{gen\_inh} = 0.2$. In this regard, in the experiments at the second stage, only the initial connection was used between the input and inhibitory layers, which is also discharged with the probability of connection formation $P_{gen\_inh} = 0.1$. It is also worth noting that the regularity or irregularity of the network of excitatory and inhibitory neurons does not significantly affect the classification accuracy, so the variation of this parameter can be neglected.

Thus, after the first stage of the study, the following intermediate conclusions were made:

- The sparse connectivity method works successfully in all connectivity areas of neuron layers, except for the area of connections between the input and inhibitory layers.
- The regularity or irregularity of the network of neurons in the layer has no significant differences.

The results of the second stage are presented in the Table 4. The best classification accuracy was achieved by architectures 3 and 4, in which sparse connectivity was simultaneously used to connect the following layers: a) input and excitatory; b) inhibitory and excitatory. In both areas, in the original architecture, the layers are connected in an all-to-all manner. Thus, the use of the sparse connectivity method can significantly reduce the number of connections between layers without sacrificing accuracy. Also, according to the results of the study at both stages, it is possible to obtain the overall average value of the probability of the formation of a connection between the input and excitatory layers at the level $P_{gen\_exc} = 0.4$.

Table 4. Results of the second stage of the research

Type of connection between layers of neurons				$\nu_{\max}$, **Hz**	**Fraction of remaining connections**	**F1**
From excitatory to inhibitory	**From inhibitory to excitatory**	**From input to excitatory**	**From input to inhibitory**			
Sparse ($P_{exc\_inh} = 0.9$; $R_{exc\_inh} = 0.8$)	Sparse ($P_{inh\_exc} = 0.2$; $R_{inh\_exc} = 0.8$)	All-to-all	Fixed Amount(10%)	950	0.95	0.94
Sparse ($P_{exc\_inh} = 0.9$; $R_{exc\_inh} = 0.8$)	All-to-all*	Probabilistic ($P_{gen\_exc} = 0.4$)	Fixed Amount(10%)	950	128.21	0.94
Counter-Partnership	Sparse ($P_{inh\_exc} = 0.4$; $R_{inh\_exc} = 0.9$)	Probabilistic ($P_{gen\_exc} = 0.4$)	Fixed Amount(10%)	950	40.31	0.96
Sparse ($P_{exc\_inh} = 0.85$; $R_{exc\_inh} = 0.3$)	Sparse ($P_{inh\_exc} = 0.25$; $R_{inh\_exc} = 0.95$)	Probabilistic ($P_{gen\_exc} = 0.45$)	Fixed Amount(10%)	550	45.09	0.95

* - excluding connections between counter-partners

4. Analysis of results

According to the results of the study, carried out in two stages, it was found that the sparse connectivity method can be successfully applied to the connections between layers, initially organized according to the all-to-all principle, significantly reducing the number of connections in the SNN. The best result in reducing the number of connections was achieved on an architecture in which sparse connectivity was implemented between a) input and excitatory, b) inhibitory and excitatory layers, and amounted to 40.3% of the original number (reduction from 252400 connections to 101746) without losing accuracy on the audio classification problem compared to the original SNN architecture. Table 5 presents the results of comparing the accuracy of various machine learning methods on the problem of classifying audio recordings from the FSDD set converted to 30 MFCC. It is shown that the accuracy (by the F1-score metric according to Eq. (8)) of classification of audio data by means of SNN is comparable to the results of classification by classical methods of machine learning.

Table 5. Comparison of various machine learning methods on the FSDD dataset using MFCC

Machine learning method	**F1**
SNN with all-to-all connectivity	0.93
SNN with sparse connectivity	0.90
Random Forest	0.96
k-Nearest Neighbors	0.97
Support Vector Machine	0.95
Multilayer Perceptron	0.90

Conclusion

In the present work, we evaluated the efficiency of using the sparse connectivity method in a three-layer spiking neural network consisting of input, excitatory, and inhibitory layers for a little-studied problem of classifying an audio recording with SNN based on local plasticity. The connections between layers of neurons are established within a limited area of neurons using a given probability. Testing this method on an audio classification problem showed that the number of connections in an SNN can be reduced by 60%. In this case, the accuracy of solving the classification problem is also achieved at the level of conventional machine learning classification methods. Thus, the prospects for using this method to reduce the computational complexity of spiking neural networks can be explored in relation to various classification problems.

References

1. Davies M, Srinivasa N, Lin T-H, Chinya G, Cao Y, Choday SH, Dimou G, Joshi P, Imam N, Jain S, Liao Y, Lin C-K, Lines A, Liu R, Mathaikutty D, McCoy S, Paul A, Tse J, Venkataramanan G, Weng Y-H, Wild A, Yang Y, Wang H. Loihi: A neuromorphic manycore processor with on-chip learning. IEEE Micro. 2018;38(1):82–99. DOI: 10.1109/MM.2018.112130359.
2. Merolla PA, Arthur JV, Alvarez-Icaza R, Cassidy AS, Sawada J, Akopyan F, Jackson BL, Imam N, Guo C, Nakamura Y, Brezzo B, Vo I, Esser SK, Appuswamy R, Taba B, Amir A, Flickner MD, Risk WP, Manohar R, Modha DS. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science. 2014;345(6197):668–673. DOI: 10.1126/science.1254642.
3. Matsukatova AN, Vdovichenko AY, Patsaev TD, Forsh PA, Kashkarov PK, Demin VA, Emelyanov AV. Scalable nanocomposite parylene-based memristors: Multifilamentary resistive switching and neuromorphic applications. Nano Research. 2023;16(2):3207–3214. DOI: 10.1007/s12274-022-5027-6.
4. Matsukatova AN, Iliasov AI, Nikiruy KE, Kukueva EV, Vasiliev AL, Goncharov BV, Sitnikov AV, Zanaveskin ML, Bugaev AS, Demin VA, Rylkov VV, Emelyanov AV. Convolutional neural network based on crossbar arrays of (Co-Fe-B)×(LiNbO3)100-x nanocomposite memristors. Nanomaterials. 2022;12(19):3455. DOI: 10.3390/nano12193455.

5. Bordanov I, Antonov A, Korolev L. Simulation of calculation errors in memristive crossbars for artificial neural networks. In: 2023 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). 15-19 May 2023, Sochi, Russian Federation. IEEE; 2023. P. 1008–1012. DOI: 10.1109/ICIEAM57311.2023.10139308.
6. Vlasov D, Minnekhanov A, Rybka R, Davydov Y, Sboev A, Serenko A, Ilyasov A, Demin V. Memristor-based spiking neural network with online reinforcement learning. Neural Networks. 2023;166:512–523. DOI: 10.1016/j.neunet.2023.07.031.
7. Tao T, Li D, Ma H, Li Y, Tan S, Liu E-X, Schutt-Aine J, Li E-P. A new pre-conditioned STDP rule and its hardware implementation in neuromorphic crossbar array. Neurocomputing. 2023;557:126682. DOI: 10.1016/j.neucom.2023.126682.
8. Sboev A, Serenko A, Rybka R, Vlasov D. Solving a classification task by spiking neural network with STDP based on rate and temporal input encoding. Mathematical Methods in the Applied Sciences. 2020;43(13):7802–7814. DOI: 10.1002/mma.6241.
9. Sboev A, Vlasov D, Rybka R, Davydov Y, Serenko A, Demin V. Modeling the dynamics of spiking networks with memristor-based STDP to solve classification tasks. Mathematics. 2021;9(24):3237. DOI: 10.3390/math9243237.
10. Diehl PU, Cook M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in Computational Neuroscience. 2015;9:99. DOI: 10.3389/fncom.2015.00099.
11. Sboev A, Davydov Y, Rybka R, Vlasov D, Serenko A. A comparison of two variants of memristive plasticity for solving the classification problem of handwritten digits recognition. In: Klimov VV, Kelley DJ, editors. Biologically Inspired Cognitive Architectures 2021. BICA 2021. Vol. 1032 of Studies in Computational Intelligence. Cham: Springer; 2022. P. 438–446. DOI: 10.1007/978-3-030-96993-6_48.
12. Vlasov D, Davydov Y, Serenko A, Rybka R, Sboev A. Spoken digits classification based on spiking neural networks with memristor-based STDP. In: 2022 International Conference on Computational Science and Computational Intelligence (CSCI). 14-16 December 2022, Las Vegas, NV, USA. IEEE; 2022. P. 330–335. DOI: 10.1109/CSCI58124.2022.00066.
13. Lien H-H, Chang T-S. Sparse compressed spiking neural network accelerator for object detection. IEEE Transactions on Circuits and Systems I: Regular Papers. 2022;69(5):2060–2069. DOI: 10.1109/TCSI.2022.3149006.
14. Tsai C-C, Yang Y-H, Lin H-W, Wu B-X, Chang EC, Liu HY, Lai J-S, Chen PY, Lin J-J, Chang JS, Wang L-J, Kuo TT, Hwang J-N, Guo J-I. The 2020 embedded deep learning object detection model compression competition for traffic in Asian countries. In: 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). 06-10 July 2020, London, UK. IEEE; 2020. P. 1–6. DOI: 10.1109/ICMEW46912.2020.9106010.
15. Han B, Zhao F, Zeng Y, Pan W. Adaptive sparse structure development with pruning and regeneration for spiking neural networks. arXiv:2211.12219. arXiv Preprint; 2022. 9 p. DOI: 10.48550/arXiv.2211.12219.
16. Amiri M, Jafari AH, Makkiabadi B, Nazari S. Recognizing intertwined patterns using a network of spiking pattern recognition platforms. Scientific Reports. 2022;12(1):19436. DOI: 10.1038/s41598-022-23320-8.
17. Timcheck J, Shrestha SB, Rubin DBD, Kupryjanow A, Orchard G, Pindor L, Shea T, Davies M. The intel neuromorphic DNS challenge. arXiv:2303.09503. arXiv Preprint; 2023. 13 p. DOI: 10.48550/arXiv.2303.09503.
18. McFee B, Raffel C, Liang D, Ellis DPW, McVicar M, Battenberg E, Nieto O. librosa: Audio and music signal analysis in python. In: Proceedings of the 14th Python in Science Conference. SciPy; 2015. P. 18–24. DOI: 10.25080/Majora-7b98e3ed-003.
19. Morrison A, Diesmann M, Gerstner W. Phenomenological models of synaptic plasticity

based on spike timing. Biological Cybernetics. 2008;98(6):459–478. DOI: 10.1007/s00422-008-0233-1.

20. Bergstra J, Yamins D, Cox DD. Hyperopt: a Python library for model selection and hyperparameter optimization. In: Proceedings of the 12th Python in Science Conference. SciPy; 2013. P. 13–19.