MATHEMATICAL PROBLEMS OF NONLINEARITY

MSC 2010: 68T05, 92B20

Modeling the Learning of a Spiking Neural Network with Synaptic Delays

A. S. Migalev, P. M. Gotovtsev

This paper addresses the spiking (or pulsed) neural network model with synaptic time delays at dendrites. This model allows one to change the action potential generation time more precisely with the same input activity pattern. The action potential time control principle proposed previously by several researchers has been implemented in the model considered. In the neuron model the required excitatory and inhibitory presynaptic potentials are formed by weight coefficients with synaptic delays. Various neural network architectures with a long-term plasticity model are investigated. The applicability of the spike-timing-dependent plasticity based learning rule (STDP) to a neuron model with synaptic delays is considered for a more accurate positioning of action potential time. Several learning protocols with a reinforcement signal and induced activity using varieties of functions of weight change (bipolar STDP and Ricker wavelet) are used. Modeling of a single-layer neural network with the reinforcement signal modulating the weight change function amplitude has shown a limited range of available output activity. This limitation can be bypassed using the induced activity of the output neuron layer during learning. This modification of the learning protocol allows reproducing more complex output activity, including for multiple layered networks. The ability to construct desired activity on the network output on the basis of a multichannel input activity pattern was tested on single and multiple layered networks. Induced activity during learning for networks with feedback connections allows one to synchronize multichannel input spike trains with required network output. The application of the weight change function leads to association of input and output activity by the network. When the induced activity is turned off, this association, configuration on the required output, remains. Increasing the number of layers and reducing feedback connection leads to weakening of this effect, so that additional mechanisms are required to synchronize the whole network.

Keywords: pulsed neural network model, spiking neural network model, synaptic plasticity, synchronization, induced activity, time delayed synapses

Received January 16, 2019

Accepted July 06, 2019

Alexander S. Migalev

alexander.migalyov@mail.ru

Pavel M. Gotovtsev

gotovtsevpm@gmail.com

National Research Center “Kurchatov Institute”

pl. Akademika Kurchatova 1, Moscow, 123182 Russia

1. Introduction

Models of spiking neural networks with synaptic plasticity based learning make it possible to implement sufficiently detailed models of learning and adaptation on a real time basis, thus allowing physical modeling on robots of different designs. Robotics in its turn is becoming increasingly a tool of research [1]. For example, R. Brooks [2, 3] uses anthropomorphic robots as a tool for investigating the behavior and social interaction, and the research team led by H. Ishiguro [4] employs them to explore the formation and development of complex behavior during social interaction. In [5, 6], anthropomorphic robots are applied to study models of the motor nervous system. For modeling of the motor control, development of the neural network basis of construction and regulation of the robot's motions E. D'Angelo uses biophysical models of neurons and synaptic plasticity [7, 8]. Based on the developed lizard robot, A. J. Ijspeert investigates neural network mechanisms of switching motor commands and control of rhythmical motions of muscles during walking and swimming using the spinal cord model [9].

The application of robotic complexes and mobile robots as research tools requires modeling of real-time neural networks, so as to make it possible to influence the environment of the robot and to observe the consequences or the result of one's actions. The possibility of optimizing calculations is an important criterion in choosing a model of the neural network to be realized. Since the computational resources of a mobile robot are limited, encoding of information by a neuron population, with changes in the probability of spike (pulse) generation for such a modeling, is redundant. One of the possibilities of reducing the number of neurons without loss of the ability to reproduce the necessary activity is to enhance the accuracy of generation of an action potential by a neuron. The accuracy with which a spike arrives in a time interval of about $10\ \mu\text{s}$ will make it possible to reproduce and analyze, for example, audio signals with a sample rate of 44.1 kHz [10]. A continuous audio signal can be transformed into a spike train using well-known algorithms [11], and audio signals can be encoded with a rate that exceeds considerably the rate of generation of an action potential by a neuron. Pulses can be reproduced or decoded into a continuous signal, for example, by calculating the pulse response of the filter and its convolution with the generated spike train neural network [12]. Details of this method and biological prerequisites for its application are described in [13, 14]. The accuracy of the frequency and periodicity of activity is important, for example, for flight control in a neural network that controls the wing muscles of a dragonfly when pursuing a prey [15].

Consider the possibilities of exact positioning of the action potential pulse by a neuron in a short time interval between input presynaptic spikes. If the model of an excitatory postsynaptic potential (EPSP) and an inhibitory postsynaptic potential (IPSP) is represented as an α -function ($\alpha(t) = a t \exp(1 - t/b)$) or, in a more simplified form, as a saw-tooth function, then the generation of a postsynaptic spike at a given instant of time, for example, by a LIF (leaky integrate-and-fire) type model, requires the arrival of a presynaptic spike. When the α -function is used, the range of change of time of the action potential depends on the inclination of increase of the function. W. Maas [16] suggested using this property, namely, introducing, in addition to weight coefficients separately for each synapse, a variable that defines the inclination and the speed of increase of the α -function. A different method is proposed in [17], where the authors suggest creating an additional feedback of each neuron, which would adjust the time of generation of the action potential. It is well known from the results of biological experiments that, for example, the active dendrites of pyramidal cells are equivalent in function to the two-layer model of artificial neural network [18]. This makes it possible to achieve a correlated activity of the neuron with inputs. Similar results were obtained for models of passive dendrites [19]. Remote

synapses of hippocampal neurons lead to a longer, time-stretched EPSP [20]. Having compared these facts, we attempted to modify the method proposed by W. Maas [16] for representing the α -function itself as subjected to modification, based on the requirements of the neuron, its response to reinforcement signals and other modulating actions which can exist during formation of complex behavior, as described in the work of O. Yu. Sinyavskii using an agent's model [21]. In our work we have attempted to combine these results within the framework of the model of a neuron with synaptic time delays.

2. Description of the model and of the means of modeling

In this paper, the spike response (SRM) model [22] was used for neuron modeling, since this model offers the possibility of optimization in the case of event-driven simulation. This model is a generalization of the leaky integrate-and-fire model. The change in the membrane potential $u_i(t)$ of neuron i in the SRM model is defined by the following equation:

$$u_i(t) = \eta(t - \hat{t}_i) + \sum_{j=1}^{N_D} w_{i,j} \sum_{p_D=1}^{P_D} \varepsilon_{i,j}(t - \hat{t}_i, t - t_{j,p_D}), \quad (2.1)$$

where $u_i(t)$ is the membrane potential of the i th (postsynaptic) neuron; $w_{i,j}$ is the weight coefficient of the synapse between neurons j (presynaptic neuron D) and i (postsynaptic neuron A); t_{j,p_D} are the time instants of spikes at the synapse of neuron i from the presynaptic j neuron (D); P_D is the length of the spike train (of neurons D); N_D is the number of presynaptic neurons; $\eta(t)$ is the function of change of the membrane potential after generation of the action potential; $\varepsilon_{i,j}(t_1, t_2)$ is the function of change of the membrane potential on the arrival of the presynaptic spike at time $t_2 = t - t_{j,p_D}$ depending on the time interval up to the last action potential generated by the postsynaptic neuron i : $t_1 = t - \hat{t}_i$. Having calculated the values of the functions $\varepsilon_{i,j}(t_1, t_2)$ and $\eta(t)$ on some time interval, one can considerably decrease the amount of computation required for modeling. In the neuron model, an action potential is generated if the membrane potential $u_i(t)$ is equal to or exceeds the threshold value ν . The time of its generation is equal to t_{i,p_A} . The array of values t_{i,p_A} , $p_A \in [1, P_A]$ corresponds exactly to the time instants of spikes of postsynaptic (A) neurons. The condition of generation of an action potential is written as $t = t_{i,p_A}$: $u_i(t) \geq \nu$ [22].

In the definition of SRM (2.1) presented above the functions $\eta(t)$ and $\varepsilon_{i,j}(t_1, t_2)$ depend on the time of the last spike of neuron i : $\hat{t}_i$ (the last spike of the postsynaptic neuron). In order to take earlier spikes into account, especially at a high spike generation rate, the summation of these functions over the number of spikes generated by the postsynaptic neuron P_A was added to Eq. (2.1). In the changed model, the value $\hat{t}_i$ was replaced with an array of time instants of generation of the action potential t_{i,p_A} :

$$u_i(t) = \sum_{p_A=1}^{P_A} \eta(t - t_{i,p_A}) + \sum_{j=1}^{N_D} w_{i,j} \sum_{p_D=1}^{P_D} \sum_{p_A=1}^{P_A} \varepsilon_{i,j}(t - t_{i,p_A}, t - t_{j,p_D}), \quad (2.2)$$

P_A is the length of the spike train of the postsynaptic neuron (A) or the spike train of neuron i .

It is important to note that the values of N_D , P_D and P_A will differ most likely for each neuron. The number of presynaptic neurons, N_D , depends on the chosen network architecture, and the values of P_D and P_A depend additionally on the time of modeling t . However, to avoid going too much into detail, we will denote them as constants.

For further optimization of calculations, Eq. (2.2) was changed. Refractoriness and the function $\eta(t)$ were replaced with restriction of the maximal rate of generation of the action potential, $f_{\max}$, an invariable period of indispensable passivity of the cell. This made it possible to divide the time series into periods with duration $f_{\max}^{-1}$, resulting in a significant optimization of modeling and representation of the spike train for transmission. The activity in one such period for one cell characterizes one number that determines the time of a single spike or its absence. After expiration of this period, the condition for exceeding the threshold of the membrane potential for the cell is checked with a simulation sample rate of 44.1 kHz (This rate was chosen so that acoustic waves could later be processed using the neural network model).

The model of the neuron we use here includes synapses with different values of time delay. The existence of several synapses between two neurons with different delay times, the propagation of EPSP and IPSP, provides more tools for a neuron to change the sequence of generated spikes by controlling the time of individual spikes. As in [23], we change the EPSP function, the α -function, by adding time delay m and duration h : $\alpha(t, h, m) = m + (at/h)\exp(1 - (t+m)/bh)$; in this case the EPSP function for one synapse looks like $\sum_{m=0}^M \sum_{h=1}^H \alpha(t, h, m) \bar{w}_{h,m}$. Here $\bar{w}_{h,m}$ are the weight coefficients of only one synapse between neurons i and j . $m \in [0, M]$, $M = I_{\Delta} f_D$, where f_D is the sample rate. The quantity I_{Δ} is the maximal value of delay in seconds, and M is the number of discrete time steps, corresponding to the maximal delay. Instead of one synapse with an EPSP $\alpha(t)$ we use an array of synapses, with EPSP functions possessing different delays in the interval $[0, I_{\Delta}]$ on the arrival of an action potential. In addition, they can have different duration. If the synaptic weights are ranked so that the delay in them increases or decreases with a step identical to the sampling period (Δ_t), then we can replace the α -function by δ with a corresponding change in the weights:

$$\sum_{\bar{m}=0}^{\bar{M}} \sum_{h=1}^H \alpha(k, \bar{m}, h) \bar{w}_{h,m} = \sum_{m=0}^M \delta(k\Delta_t - m) w_m. \quad (2.3)$$

The values of the weight coefficients on the left-hand side of the equation of $\bar{w}_{h,m}$, a two-dimensional array of values, have been replaced by new weight coefficients, a one-dimensional array of w_m . In this equation we use discrete time steps, k . On the left-hand side of the equation the values of delay, $\bar{m}$, and maximal delay, $\bar{M}$, are purposely distinguished from the corresponding symbols on the right-hand side, m and M . The function $\alpha(k, \bar{m}, h)$ has a finite duration, and in order for $\alpha(k, \bar{M}, h)$ to be filled in time with the maximal value of delay $\bar{M}$, the value of M must be greater than that of $\bar{M}$. In Eq. (2.3), by replacing the sum of α -functions with different delay and duration values by the sum of δ -functions we only transform the superposition of the α -functions into a discrete array, which we can call a new EPSP function or simply a PSP function (function of a postsynaptic potential), but now not for several synapses, but only for one. The type of this new function is defined by an array of new weight coefficients w_m .

As a result, the SRM model (2.2) has been changed to the following model:

$$u_i(k) = \sum_{j=1}^{N_D} \sum_{p_D=0}^{P_D} \sum_{m=0}^M \delta(k\Delta_t - m - t_{j,p_D}) w_{i,j,m}. \quad (2.4)$$

In this equation, $w_{i,j,m}$ are weight coefficients between neurons i (postsynaptic neuron) and j (presynaptic neuron) with time delay m from the arrival of an action potential at the synapse.

Of course, it is reasonable to form a PSP function in the form of a spline with parameters for reduction of memory resources, however, for the purposes of the experiment we have deliberately used some redundancy both in the accuracy and in the duration of delays. The delay step is

equal to the sample rate of the model, and the maximal delay reaches 5 ms. At a sample rate of 44.1 kHz, 2205 values of weight coefficients correspond to one synapse. The activity is regulated using the mechanism of reducing the threshold, v , if there is no activity at the output for a long time.

Next, it was necessary to decide how such a neural network model should be trained. Recently a high accuracy of training is demonstrated in applying learning rules based on spike-timing-dependent plasticity (STDP) [24, 25]. This form of plasticity finds application for different architectures of multilayer neural networks with different strategies of local changes in weights [26–28]. The dynamics of changes in weights, the learning strategy depends also on the form of STDP, which is modulated by different signals [29]. For example, according to [30], dopamine is capable of replacing long-term inhibition by excitation in hippocampus neurons. Signals of the dopamine system are also regarded as a trigger of changes in weights by the STDP rule in the realization of cumulative statistics of pre- and postsynaptic spikes at the synapse of the neuron. This model was presented in the work of E.M. Izhikevich [31] and developed as a neo-Hebbian learning model in more recent work [32]. The results of biological experiments show how diverse the shapes of STDP and their dependences on modulating actions are. For one neural network they can depend on the stage of development and formation of the nervous system of the animal [33]. Along with numerous variations of the slowly developing form of STDP plasticity, which manifests itself in neurons after multiple presentations of pairs of pre- and postsynaptic spikes in the course of some minutes, more rapidly developing forms are observed. In them, STDP develops in neurons already after 5–10 presentations of pairs of pre- and postsynaptic spikes [34].

This evidence indicates universality of this mechanism of plasticity and changes in the synaptic weights for different neural network architectures. Therefore, in our model we consider synaptic weights in a neuron with time delays as a substrate of different rules of changing weights with different modulating signals.

When a spike (or action potential) is generated by neuron i at time instants t_{i,p_A} , all spikes generated by a presynaptic neuron j (denoted as time instants of spikes that have arrived at the synapse of neuron j with i : t_{j,p_D}) before the time of generation in the interval I_Δ lead to changes in the weight coefficients as follows:

$$\frac{dw_{i,j,m}}{dt} = \gamma s_P(t) \sum_{p_D=0}^{P_D} \mu(t_{i,p_A} - t_{j,p_D} - m), \quad (2.5)$$

where $s_P(t)$ is the reinforcement signal modulating the amplitude of the training function $\mu(t)$, and γ is the learning speed. In other words, Eq. (2.5) describes the algorithm of changing the weights during each generation of an action potential by the postsynaptic neuron i . When changing the weights one takes into account the time instants of spikes that have arrived at the synapse from neuron j no later than at time $t - I_\Delta$.

The signal of dopamine regulation in the nervous system requires a certain specificity and architecture of the neural network, which functions in the process of learning as a tool of model construction and prediction, and dopamine serves as a signal of confirmation of the model, as discussed in [35]. Since our neural network model has no such specificity, it uses a reinforcement signal in the form of some abstract signal which regulates the learning.

The main mechanism used by us in training is the search for values of weight coefficients which would lead to the generation of an action potential by a neuron at the required time. If one considers the value of the membrane potential, then at this time it must cross the threshold value v . A neuron which has some set of input presynaptic spikes in the interval I_Δ must

adjust for this an array of weight coefficients to obtain necessary postsynaptic potentials. These changes in the weight coefficients and the formation of postsynaptic potentials take place using one template of different amplitudes — the training functions $\mu(t)$. The change in the amplitude of this function is determined by the reinforcement signal $s_P(t)$, which can take positive and negative values. The learning speed γ is a constant that is chosen empirically from the modeling conditions or network architecture, which is invariable in the process of learning. In this paper, two types of training functions $\mu(t)$ are used:

- a bipolar function of synaptic plasticity STDP: $\mu_{SP1}(t) = (-t/\sigma) \exp(1 - |t/\sigma|)$;
- a Ricker wavelet or a “Mexican hat”: $\mu_R(t) = (-t^2/\sigma^2) \exp(1 - |t^2/(2\sigma^2)|)$.

As a result of transformation of the weights of single synapses with an individual delay value into an array of weight with a linearly variable delay, the function $\mu(t)$ is different for single synapses with an EPSP/IPSP function $\alpha(t)$ and for a linear array of weights without $\alpha(t)$, with a “direct” influence on the membrane potential $u_{PP}(t)$ with changing weights. To estimate these differences, we write the function of a postsynaptic potential $u_{PP}(t)$ on the arrival of a presynaptic spike at time t_1 as follows:

$$u_{PP}(t) = \lim_{I_\Delta \rightarrow \infty} \sum_{i=1}^{I_\Delta} w(t - t_1) \alpha(t + x_i - t_1) \Delta x = \int_x w(t - t_1) \alpha(t + x) dx, \quad (2.6)$$

at the instant of a discrete change in weights we can make the change of variables $w(t - t_1) = w_0(t - t_1) + s_P(t) \mu(t - t_1)$:

$$u_{PP}(t) = \int_x (w_0(t - t_1) + s_P(t) \mu(t - t_1)) \alpha(t + x) dx, \quad (2.7)$$

where $w_0(t)$ are the initial values of synaptic weights with delay t .

Under the initial modeling conditions, $w_0(t) = 0$ and the reinforcement signal is $s_P = 1$. In the modeling interval of interest, $s_P(t)$ can be taken to be constant. Then we need to find out how the postsynaptic potential $u_{PP}(t)$ changes during learning and whether we can replace the expression $\int_x \mu(t - t_1) \alpha(t + x) dx \approx \int_x a_1 \mu(x - t_1 - a_2) dx$, where a_1 and a_2 are the parameters. The graphs of the value of the integral $\int_x \mu(t - t_1) \alpha(t + x) dx$ for the Ricker wavelet $\mu_R(t)$ and the STDP function $\mu_{SP1}(t)$ are shown in Fig. 1. In the same graphs, the initial functions $\mu_R(t)$ and $\mu_{SP1}(t)$ are constructed. These functions differ in changes of amplitude and are offset in time. Without significantly changing the algorithm of changing the weights, we can reduce the function $u_{PP}(t)$ ($a_1 = 1$, $a_1 = 0$) to $u_{PP}(t) \approx \int_x w_0(t - t_1) + s_P \mu(x - t_1)$. These results hold only for a neuron without branching of dendrites and with delays at synapses situated in the interval sequentially and linearly. If dendrites have more complex structure, the form of the function $u_{PP}(t)$ can undergo additional changes.

The reinforcement signal $s_P(t)$ in the model depends on the error, which is calculated as follows:

$$s_P(t) = \frac{1}{T_P} \int_{t-T_P}^t (s_{p_i}(\tau) - s_{o_i}(\tau)) d\tau, \quad s_{p_i}(t) = t_{i,p_A} * \Delta(t); \quad s_{o_i}(t) = t_{i,p} * \Delta(t); \quad (2.8)$$

$$\Delta(t): t \in [-I_\Delta, I_\Delta]; \quad \Delta(t) = (|I_\Delta - t|/I_\Delta)^2$$

Fig. 1. Values of the integral $\int_x \mu(t-t_1)\alpha(t+x)dx$ for $\mu_R(t)$ (a) and $\mu_{SP1}(t)$ (b) (solid line, the amplitude scale on the right). The values of the functions $\mu_R(t)$ (a) and $\mu_{SP1}(t)$ (b) (dotted line, the amplitude scale on the left).

where $s_{p_i}(t)$ and $s_{o_i}(t)$ are signals received from the required and recorded output spike train with corresponding time instants of spikes t_{i,p_A} and $t_{i,p}$, and T_P is the duration of the time interval used to calculate the reinforcement signal. The estimate of the accuracy of training for data visualization was calculated as the Pearson correlation coefficient $r\{s_{p_i}(t), s_{o_i}(t)\}$.

3. Modeling

This paper considers two types of changes in synaptic weights. In one case, the reinforcement signal changes the form of the function $\mu(t)$, while in the other case induced activity along with the use of a given form of the function $\mu(t)$ is imposed on the output neurons of the network.

3.1. Learning with a reinforcement signal

The learning neural network is a layer of 50 neurons connected with five inputs. The input signal of the network constitutes regular pulses with a frequency of 22.2 Hz, which are fired to the network along five channels sequentially with a duration of 40 ms (the total duration is 200 ms). The frequency 22.2 Hz has been chosen so that one or two input spikes will arrive at any time in the 5 ms interval (at the input of the neuron) between the maximal and the minimal synaptic delays, i.e., these spikes will be within the limits of this interval. The required output signal is identical to the input signal, but is offset in time by 4.5 ms to allow for the delay in signal passage through the network. In accordance with equations (2.5) and (2.8), the error, the reinforcement signal $s_P(t)$ and new values of weight coefficients were calculated at intervals of 200 ms. For learning, two training functions, $\mu_R(t)$ and $\mu_{SP}(t)$, were applied. The values of the weight of each synapse were subjected to normalization — restriction of the minimal and maximal values and the sum of moduli of weights $\sum_{m=0}^M |w_{i,j,m}| = C$ between two neurons. The value of C was selected empirically.

Under identical conditions a change in the training function changes the pattern of activity of the network. The graphs in Fig. 2 show the learning process — changes in the correlation coefficient between the recorded and the reference correlation, the required output spike train for the training function $\mu_R(t)$ and $\mu_{SP1}(t)$. We can see advantages in this numerical experiment of application of the function $\mu_R(t)$, however, one can achieve high indices of training accuracy only for one input spike train out of five. As the number of inputs increases, the accuracy of training decreases dramatically, particularly for $\mu_{SP1}(t)$. The graphs for three inputs clearly demonstrate different learning dynamics for the first input. This is due to the presence of periods

Fig. 2. Evolution of the correlation coefficient r during training for two training functions: the Ricker wavelet $\mu_R(t)$, (graphs (c), (e), (g)) and STDP $\mu_{SP1}(t)$ (graphs (b), (d), (f), (h)). The figure shows results for two training protocols: the reinforcement signal comes only when a regular spike train appears at the first input out of five (graphs (a), (b)) and for three inputs out of five (graphs (c)–(h)). The graphs of the second protocol are arranged according to the number of the input from top to bottom: (c), (d) — input 1; (e), (f) — input 2; (g), (h) — input 3.

when spike trains arrive in the network from 4 and 5 channels, which are not taken into account in calculating the error and the reinforcement signal. As the number of inputs for training increases to five out of five, this effect disappears completely. The graph in Fig. 3 shows a fragment of a spike train in the process of training for the function $\mu_P(t)$ and three inputs. One can observe how the reinforcement signal, acting simultaneously on the whole network, leads to changes in the whole layer. As the number of channels increases, the diversity of reproducible regular spikes increases. Also, the formation of groups of synchronized neurons or polysynchronized neurons [36] with inputs depends on the form of the training function.

The training algorithm used allows one to train the neural network for output spike trains with limited complexity. Using only the modulation of the amplitude of the function of weight change with a long interval of adjustment of values of $s_P(t)$ or with a low updating rate, it becomes impossible for this neural network model to learn complicated output spike trains or activity patterns. Using only the tool of modeling of the amplitude of the function of weight change for this neural network model, it becomes impossible to choose a predetermined output spike train with specific frequencies and delays between the action potentials of individual neurons in the network from the whole range of all possible activity patterns of the neural network leading to a decrease in error, since the learning does not converge and the synaptic weights undergo constant fundamental changes.

3.2. The use of induced activity

One of the possibilities of optimizing the training algorithm is to impose induced activity on output neurons. Using small periods in which the activity of the output layer of the network will be under external control, we can synchronize the network with the input spike train. This will

Fig. 3. A fragment of a spike train during the training of a single-layer network with three inputs out of five using the training function $\mu_{\mathbf{R}}(t)$. Channels 0–5 are inputs of the network, channels 1–54 are outputs of neurons of the network. The error was calculated for the outputs of neurons (channels 54–51). During training, neuron 50 (channel 54) must generate a spike train of input 1 with a delay of 4.5 ms, and neurons 49 and 48 (channels 53 and 52) those of inputs 2 and 3, respectively (the outputs of the network must be a mirror reflection of inputs with a delay of 4.5 ms).

enable the network to learn complicated spike trains and activity patterns. During modeling in periods of induced activity the reinforcement signal was equal to the maximal value (unity), and at all other times it was equal to zero. The duration of five input patterns was identical to that of the previous experiment (40 ms). The period of induced activity lasted 200 ms followed by a period of 200 ms without changes in the weights. The weights changed for the whole network consisting of 10 neurons, and induced activity was applied only for five output neurons. Different modifications of the functions $\mu(t)$ (Fig. 4) were used to investigate the influence of the form of the function $\mu(t)$ on training.

For this network, in addition to the normalization of minimal and maximal values of synaptic weights and instead of restriction of the sum of moduli of weights $\sum_{m=0}^M |w_{i,j,m}| = C$, a different, more efficient mechanism of suppression of undesirable high-frequency activity was used. When the size of the network grows, it manifests itself more actively. The values of weights in the model are adjusted every 200 ms on the basis of changes accumulated for the last 200 ms in accordance with the following rule:

$$\frac{dw_{i,j,m}}{dt} = \gamma \left(s_{\mathbf{R}}(t) \sum_{p_{\mathbf{D}}=0}^{P_{\mathbf{D}}} \mu(t_{i,p_{\mathbf{A}}} - t_{i,p_{\mathbf{D}}} - d) - (r_{\mathbf{N}i}(t)/b_{\mathbf{N}})^2 \right), \quad (3.1)$$

$$dr_{\mathbf{N}i}/dt = -a_{\mathbf{N}} r_{\mathbf{N}i}$$

where $r_{\mathbf{N}i}(t)$ is the variable regulating the activity of neuron i , and $b_{\mathbf{N}}$, $a_{\mathbf{N}}$ are the parameters: $a_{\mathbf{N}} = 0.1$, $b_{\mathbf{N}} = 1950$. In addition, at each instant of generation of a postsynaptic spike, except for the induced spike, the synaptic weights change during the following adjustment of weights:

$$\frac{dw_{i,j,m}}{dt} = w_{i,j,m} \left(\left| 1 - \frac{r_{\mathbf{N}i}}{r_{\mathbf{N} \max} + c_{\mathbf{N}}} \right| - 1 \right), \quad (3.2)$$

$$dr_{\mathbf{N}i}/dt = 1,$$

Fig. 4. Different modifications of the function of weight change which were applied for training. (a) Ricker wavelet $\mu_R(t)$; (b) STDP function $\mu_{SP1}(t)$; (c) STDP function with a delay of $72.6 \mu s$ $\mu_{SP1D}(t)$; (d) Ricker wavelet with an amplitude of negative values $\mu_{RP}(t)$ which was reduced by a factor of 10; (e) STDP function with an amplitude of negative values $\mu_{SP1P}(t)$ which was reduced by a factor of 10; (f) STDP function with an amplitude of negative values which was reduced by a factor of 10, and with a delay of $72.6 \mu s$ $\mu_{SP1PD}(t)$.

Fig. 5. Evolution of the correlation coefficient r during training of a single-layer network using the functions of change of weights: (a) — $\mu_R(t)$; (b) — $\mu_{RP}(t)$; (c) — $\mu_{SP1}(t)$; (d) — $\mu_{SP1P}(t)$; (e) — $\mu_{SP1D}(t)$; (f) — $\mu_{SP1PD}(t)$. The dotted line 1 is a dependence where the input of the network includes five regular successive spike trains with a frequency of 22.2 Hz. The solid line 2 is a dependence where the input of the network includes five regular successive spike trains with a frequency of 22.2 Hz and 45 random spike trains with uniform spike generation probability distribution in a frequency band of 0.33–100 Hz. The dotted line 3 is a dependence where the input of the network includes only 45 random spike trains with uniform spike generation probability distribution in a frequency band of 0.33–100 Hz.

where $r_{N \max}$ and c_N are the parameters: $r_{N \max} = 500$, $c_N = 50$. The training for iterations of a regular 5-channel spike train using the training protocol from the previous section presents no difficulty for this single-layer network. At the input there can be an even more complex signal, for example, a periodic 45-channel spike train with uniform time distribution between spikes (Fig. 5).

Fig. 6. A fragment of a spike train when the single-layer network learns 45 random spike trains with uniform spike generation probability distribution in a frequency band of 0.33–100 Hz using the training function $\mu_{\mathbf{R}}(t)$. Channels (a) from top to bottom — 45 random input spike trains. Channels (b) from bottom to top — 5 neurons with projection onto the outputs and the 5 remaining neurons. Channels (c) from top to bottom — 5 required output spike trains. The neurons have reverse numeration, which is necessary to avoid error in data analysis. Intervals 1 and 3 — the reinforcement signal is equal to zero, no changes of weights occur, there is no induced activity. Interval 2 — the reinforcement signal is equal to unity, and the induced activity on neurons with projection onto output is activated.

The graphs demonstrate the evolution of the correlation coefficient r between the recorded inputs and the required spike trains for different functions of change of weights, $\mu(t)$. A random 45-channel periodic spike train and a regular 5-channel spike train were sent, together or separately, to the input of the network. For all training functions, the required output is first observed to be reached fast, and then one can observe a decrease in accuracy, which is particularly manifest in the presence of a random spike train at the inputs. The initial values of weights are equal to zero, and such a fast growth synchronizes the network with the required output. As the weights change further, the normalization rules do not allow one to selectively return the synaptic weights to earlier, more exact values. As can be seen from the data obtained, some functions $\mu(t)$ are more stable with respect to such a decrease in accuracy. The best results have been obtained for the STDP function with a decreased amplitude of negative values (Fig. 5d, $\mu_{\text{SP1P}}(t)$, and Fig. 4e). The input that is the most complicated for training includes random spike trains (line 3, Fig. 5). At the input there are no inputs that could be used for generation of the required output spike trains. The decrease in the number of such inputs leads to a gradual decrease in accuracy, and vice versa, 45 inputs, for example, using the function $\mu_{\mathbf{R}}(t)$ (Fig. 4a), make it possible to form visually observable contours of the output. A fragment of a spike train for the instant and the graph shown by an arrow in Fig. 5a are presented in Fig. 6. This fragment shows the process of training of the network.

In the single-layer network, the neuron model “uses” during training only a temporary window equal to the maximal delay at synapses I_{Δ} (in this paper, it is equal to 5 ms). For this reason, for a complex input spike train where one should compare the time instants of spikes longer than this delay, a multilayer network architecture is necessary.

At the next stage, after we have convinced ourselves that it is possible to classify input spike trains, we want to find out whether the training rules we apply are capable of teaching the neural network to generate a regular sequence of pulses of a certain frequency without a synchronizing input spike train.

3.3. Multilayer networks

To increase the diversity of trainable spike trains and activity patterns, we applied the same training algorithm to a three-layer network consisting of 150 neurons, Fig. 7b. The weight normalization algorithm was also identical to that used in Section 3.2.

Fig. 7. Neural network architectures in use. (a) — a three-layer network consisting of 150 neurons with feedbacks. Each layer contains 50 neurons. All connections shown in the diagram are projected onto all neurons of the layer. The input of the network consists of 50 channels, which are activated depending on the training protocol. The output of the network is formed by the activity of 5 neurons of the chosen layer. (b) — a five-layer network consisting of 50 neurons, 10 neurons in each layer. The type of connections, and the inputs and outputs are identical to those of the network (a).

The form of graphs of evolution of the accuracy of learning for different functions of weight change $\mu(t)$ the results of modeling of the three-layer network did not differ qualitatively from those obtained for the single-layer network. Therefore, we consider the results only for the function with the best results of training $\mu_R(t)$ (of the Ricker wavelet) shown in Fig. 8.

One can see how the learning dynamics has changed as compared to the single-layer network (Fig. 5). The types of input spike trains and the required output spike trains are identical to those of the single-layer network.

The volume of the network has been increased by a factor of 15, resulting in a longer learning. The attainable accuracy, the correlation coefficient r , decreases as well. In the presence of 45 random spike trains at the input, the same effect of decrease in accuracy is observed, but it is stretched in time. Feedbacks with other layers make it possible to obtain required output spike trains on the neurons of each layer without a significant decrease in attainable accuracy. In the presence of 45 random spike trains at the input, the learning dynamics on three layers does not differ greatly, but if we change the network architecture to one consisting of five layers, as shown in Fig. 7c, and perform the same series of modeling with the same input actions (Fig. 9), then, when sending a 45-channel random spike train, we will be able to see how the attainable accuracy of training decreases on each subsequent layer. For example, already on the fourth layer it is impossible to reproduce even a simple five-channel regular spike train. As the number of layers that are not connected with the input increases, one can observe a decrease in the diversity of activity, which could be associated with the induced activity. Nevertheless, these results confirm the possibility of associating during training the outputs of the network with input activity, with projection through one layer of the network.

Fig. 8. Evolution of the correlation coefficient r during training of a three-layer network using the Ricker wavelet as functions of weight change $\mu_{\mathbf{R}}(t)$. Graphs (a), (b), (c) — the first layer. Graphs (d), (e), (f) — the second layer. Graphs (g), (h), (i) — the third layer. For graphs (a), (d), (g) the input of the network includes five regular successive spike trains with a frequency of 22.2 Hz. For graphs (b), (e), (h) the input of the network includes five regular successive spike trains with a frequency of 22.2 Hz and 45 random spike trains with uniform spike generation probability distribution in a frequency band of 0.33–100 Hz. For graphs (c), (f), (i) the input of the network includes only 45 random spike trains with uniform spike generation probability distribution in a frequency band of 0.33–100 Hz.

Fig. 9. Evolution of the correlation coefficient r during training of a five-layer network using a Ricker wavelet as functions of weight change $\mu_{\mathbf{R}}(t)$. The graphs correspond to the following layers of the network: (a) — 1, (b) — 2, (c) — 3, (d) — 4, (e) — 5. The input of the network includes only 45 random spike trains with uniform spike generation probability distribution in a frequency band of 0.33–100 Hz.

4. Conclusion

The results obtained show the possibility of training a neural network with synaptic delays to reproduce required spike trains. Using a function of weight change selected for each problem, the network can also be trained to operate as a classifier on the basis of a single-layer

or multilayer network. An important result obtained in modeling is the possibility of associating (indirectly, through intermediary neurons by synchronization or polychronization) the induced activity at the output of the network with a random spike train at the input. This offers an opportunity to control and tune networks different in architecture in accordance with the required output. This association weakens fast as the the volume and the layers of the network increase, primarily due to the fact that the chosen algorithm of activity normalization often suppresses our training algorithm. However, such an approach opens up wide opportunities for optimization and a multifaceted interaction between neuron models, which will lead to a more pronounced synchronization of the activity of the network's neurons.

The redundant volume of synaptic weights with time delays has allowed us to qualitatively estimate the type and the relation of synaptic weights after training. It is quite possible to approximate the recorded synaptic weights without loss of function by several values using a linear function or a spline. In the future we intend to check this assumption in model experiments.

It is the principle of training the network that is of importance and constitutes the novelty of the results obtained. The neuron model used in training controls the time of spike generation rather than the general average activity — changes in the average frequency of generated spikes and unaveraged activity of population. This opens up an opportunity to apply such a model of the network and training to control fast processes where the time instants of individual spikes are critical. This involves a spectrum of problems such as analysis of data from sensors, cameras, recognition and reproduction of sound, and real-time control of actuators in robotic applications. The novelty of the model is the method of representing an array of weight coefficients with different synaptic delays in the form of an arbitrary function, a substrate for formation of different activity templates. The results obtained show that the training of the network is possible when the PSP function is changed using a modification of the STDP rule. The confirmation of the ability of the network to learn opens up for different architectures wide possibilities of further development and improvement of training such a type of neural network models with time synaptic delays and the search for an optimal algorithm of formation of the PSP function.

References

- [1] Brooks, R. A., From Earwigs to Humans, *Robot. Auton. Syst.*, 1997, vol. 20, nos. 2–4, pp. 291–304.
- [2] Brooks, R. A., Breazeal, C., Marjanović, M., Scassalatti, B. and Williamson, M. M., The Cog Project: Building a Humanoid Robot, in *Computation for Metaphors, Analogy, and Agents (CMAA, 1998)*, Ch.L.Nehaniv (Ed.), Lecture Notes in Comput. Sci., vol. 1562, Berlin: Springer, 1999, pp. 52–87.
- [3] Adams, B., Breazeal, C., Brooks, R. A., and Scassalatti, B., Humanoid Robots: A New Kind of Tool, *IEEE Intell. Syst.*, 2000, vol. 15, no. 4, pp. 25–31.
- [4] Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., Ogino, M., and Yoshida, C., Cognitive Developmental Robotics: A Survey, *IEEE Trans. Auton. Mental Develop.*, 2009, vol. 1, no. 1, pp. 12–34.
- [5] Kawato, M., From “Understanding the Brain by Creating the Brain” towards Manipulative Neuroscience, *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 2008, vol. 363, pp. 2201–2214.
- [6] Morimoto, J. and Kawato, M., Creating the Brain and Interacting with the Brain: An Integrated Approach to Understanding the Brain, *J. R. Soc. Interface*, 2015, vol. 12, no. 104, pp. 1–15.
- [7] D’Angelo, E., Mapelli, L., Casellato, C., Garrido, J. A., Luque, N., Monaco, J., Prestori, F., Pedrocchi, A., and Ros, E., Distributed Circuit Plasticity: New Clues for the Cerebellar Mechanisms of Learning, *Cerebellum*, 2016, vol. 15, no. 2, pp. 139–151.

- [8] Casellato, C., Antonietti, A., Garrido, J. A., Ferrigno, G., D'Angelo, E., and Pedrocchi, A., Distributed Cerebellar Plasticity Implements Generalized Multiple-Scale Memory Components in Real-Robot Sensimotor Tasks, *Front. Comput. Neurosci.*, 2015, vol. 9, no. 24, pp. 1–9.
- [9] Ijspeert, A. J., Crespi, A., Ryczko, D., and Cabelguen, J.-M., From Swimming to Walking with a Salamander Robot Driven by a Spinal Cord Model, *Science*, 2007, vol. 315, no. 5817, pp. 1416–1420.
- [10] Migalev, A. S., Searching Algorithm for Sound Wave Transform in to Spike Sequence, in *Neuroinformatics-2017: Proc. of the 19th Conf. on Artificial Neural Networks: P. 1*, Moscow: National Research Nuclear University MEPhI, 2017, pp. 60–70 (Russian).
- [11] Schrauwen, B. and Van Campenhout, J., BSA, a Fast and Accurate Spike Train Encoding Scheme, in *Proc. of the Internat. Joint Conf. on Neural Networks (Portland, Ore., 2003)*, pp. 2825–2830.
- [12] Rieke, F., Warland, D., de Ruyter van Steveninck, R. R., and Bialek, W., *Spikes: Exploring the Neural Code*, Cambridge, Mass.: MIT, 1999.
- [13] Bialek, W., Rieke, F., de Ruyter van Steveninck, R., and Warland, D., Reading a Neural Code, *Science*, 1991, vol. 252, no. 5014, pp. 1854–1857.
- [14] Bialek, W. and Rieke, F., Reliability and Information Transmission in Spiking Neurons, *Trends Neurosci.*, 1992, vol. 15, no. 11, pp. 428–434.
- [15] Gonzalez-Bellido, P. T., Peng, H., Yang, J., Georgopoulos, A. P., and Olberg, R. M., Eight Pairs of Descending Visual Neurons in the Dragonfly Give Wing Motor Centers Accurate Population Vector of Prey Direction, *Proc. Natl. Acad. Sci. USA*, 2013, vol. 110, no. 2, pp. 696–701.
- [16] Maas, W. and Bishop, C. M., *Pulsed Neural Networks*, Cambridge, Mass.: MIT, 2001.
- [17] Schwemmer, M. A., Fairhall, A. L., Denève, S., and Shea-Brown, E. T., Constructing Precisely Computing Networks with Biophysical Spiking Neurons, *J. Neurosci.*, 2015, vol. 28, no. 35, pp. 10112–10134.
- [18] Mel, B. W., Structural Plasticity at the Axodendritic Interface: Some Functional Implications, in *Modeling Neural Development*, A. Van Ooyen (Ed.), Cambridge, Mass.: MIT, 2003, pp. 273–290.
- [19] Case, R. D., Humphries, M., and Gutkin, B., Passive Dendrites Enable Single Neurons to Compute Linearly Non-Separable Functions, *PLoS Comput. Biol.*, 2013, vol. 9, no. 2, pp. 1–15.
- [20] Migliore, M., Messineo, L., and Ferrante, M., Dendritic I_k Selectively Blocks Temporal Summation of Unsynchronized Distal Inputs in CA1 Pyramidal Neurons, *J. Comput. Neurosci.*, 2004, vol. 1, no. 16, pp. 5–13.
- [21] Sinyavskiy, O. Yu. and Kobrin, A. I., Reinforcement Learning of a Spiking Neural Network in the Task of Control of an Agent in a Virtual Discrete Environment, *Nelin. Dinam.*, 2011, vol. 7, no. 4, pp. 859–875 (Russian).
- [22] Gerstner, W., *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge, Mass.: MIT, 2002.
- [23] Migalev, A. S., Application of Custom Synaptic Plasticity Model on Spiking Neural Networks, in *Neuroinformatics-2018: Proc. of the 21st Internat. Conf. on Artificial Neural Networks: P. 2*, Moscow: National Research Nuclear University MEPhI, 2018, pp. 154–161 (Russian).
- [24] Thiele, J. C., Bichler, O., and Dupret, A., Event-Based, Timescale Invariant Unsupervised Online Deep Learning with STDP, *Front. Comput. Neurosci.*, 2018, vol. 12, pp. 1–13.
- [25] Tavanaei, A., Masquelier, T., and Maida, A., Representation Learning Using Event-Based STDP, *Neural Netw.*, 2018, vol. 9, no. 2, pp. 1–15.
- [26] Diehl, P. U. and Cook, M., Event-Based, Timescale Invariant Unsupervised Online Deep Learning with STDP, *Front. Comput. Neurosci.*, 2018, vol. 12, pp. 1–13.
- [27] Demin, V. A. and Nekhaev, D. V., Spiking Neural Networks Learning Based on a Neuron Activity Maximizing Principle, in *Neuroinformatics-2018: Proc. of the 21st Internat. Conf. on Artificial Neural Networks: P. 2*, Moscow: National Research Nuclear University MEPhI, 2018, pp. 54–64 (Russian).

- [28] Demin, V. A. and Nekhaev, D. V., Recurrent Spiking Neural Network Learning Based on a Competitive Maximization of Neuronal Activity, *Front. Neuroinform.*, 2018, vol. 12, pp. 1–13.
- [29] Zappacosta, S., Mannella, F., Mirolli, M., and Baldassarre, G., General Differential Hebbian Learning: Capturing Temporal Relations between Events in Neural Networks and the Brain, *PLoS Comput. Biol.*, 2018, vol. 14, no. 8, pp. 1–30.
- [30] Zhang, J.-C., Lau, P.-M., and Bi, G.-Q., Gain in Sensitivity and Loss in Temporal Contrast of STDP by Dopaminergic Modulation at Hippocampal Synapses, *Proc. Natl. Acad. Sci. USA*, 2009, vol. 106, no. 31, pp. 13028–13033.
- [31] Izhikevich, E. M., Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling, *Cereb. Cortex*, 2007, vol. 17, no. 10, pp. 2443–2452.
- [32] Kuśmierz, Ł., Isomura, T., and Toyozumi, T., Learning with Three Factors: Modulating Hebbian Plasticity with Errors, *Curr. Opin. Neurobiol.*, 2017, no. 46, pp. 170–177.
- [33] Gerstner, W., Lehmann, M., Liakoni, V., Corneil, D., and Brea, J., Eligibility Traces and Plasticity on Behavioral Time Scales: Experimental Support of NeoHebbian Three-Factor Learning Rules, *Front. Neural Circuits*, 2018, vol. 12, Art. 053, 16 pp.
- [34] Cui, Y., Paillé, V., Xu, H., Genet, S., Delord, B., Fino, E., Berry, H., and Venance, L., Endocannabinoids Mediate Bidirectional Striatal Spike-Timing-Dependent Plasticity, *J. Physiol.*, 2018, vol. 593, no. 13, pp. 2833–2849.
- [35] Quartz, S. R., Modeling the Neural Basis of Cognitive Development, in in *Modeling Neural Development*, A. Van Ooyen (Ed.), Cambridge, Mass.: MIT, 2003, pp. 291–313.
- [36] Izhikevich, E. M., Polychronization: Computation with Spikes, *Neural Comput.*, 2006, vol. 18, no. 2, pp. 245–282.