Pulse Neuron Learning Rules for Processing of Dynamical Variables Encoded by Pulse Trains

Vladimir Bondarev(✉)

Sevastopol State University, 299053 Sevastopol, Russian Federation
bondarev@sevsu.ru

Abstract. The paper deals with a model of pulse neural network that is applicable for solving of various tasks of processing sensory information. These tasks relate to dynamical variables processing. The distinctive feature of the problem statement is that dynamical variables are represented by pulse (spike) trains. We propose two supervised temporal learning rules for pulse neural network executing the required linear dynamic transformations of variables represented by pulse trains. To generate the required output of the network model we used a reference system with desired properties. The rules minimize the difference between the actual and required pulse train in a local window. The first temporal learning rule was named WB-FILT as it uses the filtered values of errors between binary vectors representing the desired and actual pulse sequences. The second rule was named WB-INST as it uses instantaneous value of the error, which is the difference of the desired and the actual elements of binary vectors. We demonstrated rule's properties by computer simulation of the mappings of the regular and the dynamical pulse trains. It has been shown that proposed rules are able to configure the simple network that implements a linear dynamic system.

Keywords: Pulse neuron · Pulse train · Supervised learning · Dynamic system

1 Introduction

Now much attention is paid to the pulse neural networks (PNN) for processing of dynamical variables [1, 2]. In PNN the dynamical variables are encoded by pulse (spike) trains. Development of supervised learning rules for functional PNN which implements the required processing of dynamical variables during the mapping process of the input pulse trains to the desired output pulse trains is considered as an important problem in neuroinformatics [3].

Various temporal supervised learning rules providing the desirable mappings of the pulse trains and using precise time of pulses are proposed in [4–7]. However, in most cases they are oriented on pattern classification problems and are not aimed to the direct application in adaptive real-time systems where processing of the dynamical variables represented by the multi-pulse trains is required.

The vector-matrix digital model of the pulse neuron (PN) and the supervised learning rule for real-time adaptive signal processing were proposed in [8, 9].

B. Kryzhanovsky et al. (eds.), *Advances in Neural Computation, Machine Learning, and Cognitive Research*, Studies in Computational Intelligence 736,
DOI 10.1007/978-3-319-66604-4_8

The purpose of this paper is the extension of the scope of the PN vector–matrix model [8, 9] that provides direct realization of the required linear transformations of dynamical variables based on the input and output pulse sequences of the PN.

2 Problem Formulation

We will consider the adaptive modeling scheme of the linear dynamic system appearing as a reference system which performs the required linear transformation (mapping) of the input dynamical variable $u(t)$ to the output variable $y_d(t)$ represented (encoded) by means of desired pulse sequence $s_d(t)$. We want to construct the PNN model which reproduces the dynamics of the reference system based on the desired (required) pulse train $s_d(t)$.

To solve the problem, we will use the multi-input PN model that was considered in [8]. It is assumed that bipolar input pulse trains $u_i(t)$ generated by the encoding presynaptic neurons arrive at inputs of the PN linear filters with pulse responses $h_i(t)$. Filter reactions $x_i(t)$ are weighted with synaptic weights w_i and summarised to form the summary postsynaptic potential $y_o(t)$ of the PN. If the integral of the module of $y_o(t)$ exceeds a threshold then an output pulse of the PN with the sign corresponding to the sign of $y_o(t)$ is emitted and integrator state is nullified. The specified chain of the conversions corresponds to the LIF-neuron.

If we calculate the values of $y_o(t)$ at discrete time $t_n = n\Delta t$, where Δt is a time sampling step, then [8, 9]

$$y_o(n) = \mathbf{w}^T\mathbf{x}(n),\ x_i(n) = \mathbf{b}_i^T(n)\mathbf{h}_i, \tag{1}$$

where $\mathbf{w}^{\mathrm{T}} = (w_0, w_1, \ldots, w_{I-1})$ is synaptic weight vector, $\mathbf{b}_i^{\mathrm{T}}(n)$ is sliding binary vector whose elements are equal to signs of the input pulses at time moments t_n, $\mathbf{h}_i$ denotes the impulse response vector $\mathbf{h}_i = (h_i(0), h_i(1), \ldots, h_i(K-1))^{\mathrm{T}}$. In this case, we can use the supervised learning rule in the form of Widrow-Hoff [9]:

$$\Delta\mathbf{w}(n) = \mu\mathbf{x}(n)e(n), \tag{2}$$

Where μ is a learning rate, $e(n) = y_d(n) - y_0(n)$ is an error.

The rule (2) assumes that PN input signals are pulses and the output signal of the PN is represented by sample values of the dynamical variable $y_o(t)$. Therefore, the rule (2) cannot be used directly for training of a PNN where not only input signals, but also output signals are represented by pulse sequences.

We will derive the supervised PN learning rules for a case when the required output signal $y_d(t)$ of the reference dynamic system and the actual output signal $y_o(t)$ of the PN model (1) are represented by the pulse trains. We will call such rules that are driven directly by the time of pulses as temporal rules.

3 Temporal Learning Rules of the Pulse Neuron

In order to calculate the error $e(n)$ we will use the known similarity measures of the pulse trains [10, 11]. The most often used measure convolves of the pulse trains with some positive smooth localized kernel $h_r(t)$. In accordance with (1) the convolution of an actual output pulse train $s_o(t)$ and desired pulse trains$s_d(t)$ with a kernel $h_r(t)$ can be written as follows

$$\tilde{y}_o(n) = \mathbf{b}_o^{\mathrm{T}}(n)\mathbf{h}_r,\ \tilde{y}_d(n) = \mathbf{b}_d^{\mathrm{T}}(n)\mathbf{h}_r, \tag{3}$$

where $\mathbf{h}_r$ is the vector of samples of a kernel $h_r(t)$, $\mathbf{b}_o^{\mathrm{T}}(n)$ and $\mathbf{b}_d^{\mathrm{T}}(n)$ are the binary sliding vectors containing of M elements and corresponding to the pulse sequences $s_o(t)$ and $s_d(t)$. Length M is selected considering dynamics of the reference system and the processed signals. In fact, binary vectors fix some temporal prehistory of pulses.

Variables $\tilde{y}_o(n)$ and $\tilde{y}_d(n)$ can be interpreted as the result of conversions of dynamical variables $y_o(n)$ and $y_d(n)$ to the pulse trains s_o and s_d, and then back to the origin form for the purpose of restoration of these variables from the pulse sequences. If we perform replacement of $y_o(n)$ and $y_d(n)$ by variables $\tilde{y}_o(n)$ and $\tilde{y}_d(n)$ in the learning rule (2) we will derive the temporal learning rule

$$\Delta\mathbf{w}(n) = \mu\mathbf{x}(n)\left[\left(\mathbf{b}_d^{\mathbf{T}}(n) - \mathbf{b}_o^{\mathbf{T}}(n)\right)\mathbf{h}_r\right]. \tag{4}$$

Having compared (2) and (4), we conclude that the error $e(n)$ in the rule (4) corresponds to the difference of binary vectors representing the desired and actual pulse sequences. At the same time this error is smoothed by a window (by a filter) with weights $\mathbf{h}_r$. We will name this temporal rule WB-FILT, as it compares filtered binary vectors (by analogy with [7]).

The window $\mathbf{h}_r$ is often selected so that the pulses (elements of binary vectors) which were formed later will have the greater weight. If the length of the window is restricted to a single sample then from (4) we derive the simple learning rule

$$\Delta\mathbf{w}(n) = \mu\mathbf{x}(n)(b_d(n) - b_0(n)), \tag{5}$$

where $b_d(n)$ and $b_o(n)$ are the elements of binary vectors. This temporal rule uses the instantaneous value of the error, which is equal to the difference of binary vectors elements. Therefore, we will name it WB-INST (by analogy with [7]).

4 Computer Simulation

During the simulation, the simple model of bipolar IF-neuron with single input was used as the model of an encoding neuron. The encoding neuron converts an input signal $u(t)$ to a pulse train. This pulse train simultaneously arrives to all inputs of the PN. Pulse responses of the PN filters were identical in the form, but shifted in time for the sampling step, i.e. $h_i(t) = \exp(-(t - i\Delta t)/\tau_s)$, where τ_s is the time constant.

To keep the shape of signals the finite symmetric exponent was used as a kernel function. The kernel function was shifted for the half of its length to provide the linear phase characteristic. Such kernel function creates the time delay equal to $(M-1)/2$ (if M is odd) that requires the correction of the rule (4):

$$\Delta \mathbf{w}(n) = \mu \mathbf{x}(n - (M-1)/2)\left[\left(\mathbf{b}_d^{\mathbf{T}}(n) - \mathbf{b}_o^{\mathbf{T}}(n)\mathbf{h}_r\right)\right]. \quad (6)$$

In the first computational experiment (I = 401, K = 64, M = 129, Δt = 0.5 ms), we run the training process to map the regular input pulse train with the period of 12.5 ms to the desired pulse train with the period of 20.5 ms. The specified pulse sequences were created by the encoding neurons when their inputs are constant signals with amplitude $u(t)$ = 0.08 and $y_d(t)$ = 0.05. It provided one pulse within the significant duration of the pulse response $h_i(t)$. In this case, the filter reactions to pulses in the separate channels of PN are not accumulated. It allows tracing the learning dynamics of PN visually.

The actual output pulse train in the form of a raster and the diagram of the mean-square error after training of the PN with the help of WB-INST rule (5) are illustrated in Fig. 1. The raster was created from the actual pulse train by cutting it into segments.

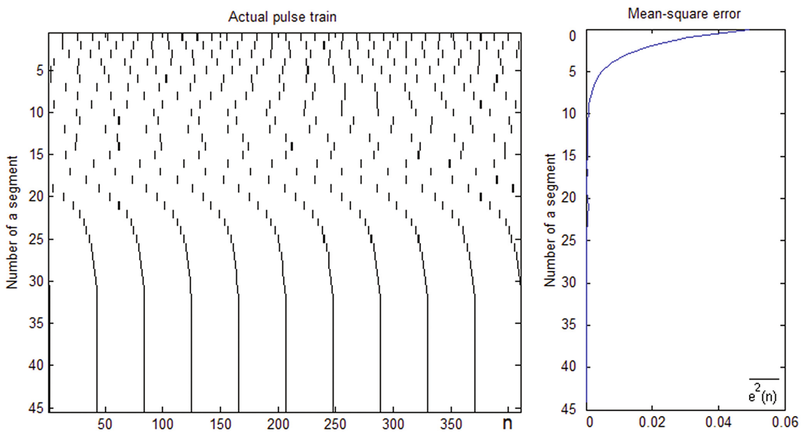

Fig. 1. Results of the mapping of the regular pulse sequences

Similar results also turn out using the WB-FILT rule (6). However, in case of a mapping of the regular pulse trains the WB-FILT rule provides faster convergence in comparison with the WB-INST rule due to averaging of the error $e(n)$ by $\mathbf{h}_r$.

In the second experiment (I = 2001, K = 10, M = 65, Δt = 10 ms), the training of a mapping of the dynamical pulse sequences was carried out. We want to build a PNN with the dynamics defined by the dynamics of the reference system which implements double integration of the dynamical input variable $u(t)$. The similar problem arises in the case of signal processing of accelerometers [12].

During the training, the input signal $u(t)$ equal to the sum of sine signals with the multiple frequencies was applied to the input of the encoding neuron, and the corresponding desired (reference) signal $y_d(t)$ arrived at the input of other encoding neuron. The desired output signal $y_d(t)$ is calculated with the help of normalized values of frequency response of the reference double integrator [12].

The distributions of the weight vector elements after training of PN are shown in Fig. 2. Interpreting **w** as a pulse response, it is possible to obtain the frequency response of the PNN model which corresponds to the double integrator in the bandpass range (curves 2 and 3). The frequency response of the reference double integrator (curve 1) was set in 30 uniformly distributed frequency points. Figure 2 shows that the mean square of the error $e(n)$ is decreasing with growth of n and the frequency response of the synthesized PNN model approximates the frequency response of the reference double integrator well. Pulse periodic behavior of the error is explained by periodicity of used signals.

It is interesting to note that despite the differences in the nature of elements distribution of the vector **w** for two rules (5) and (6) the frequency responses obtained with their help are the very close (curves 2 and 3).

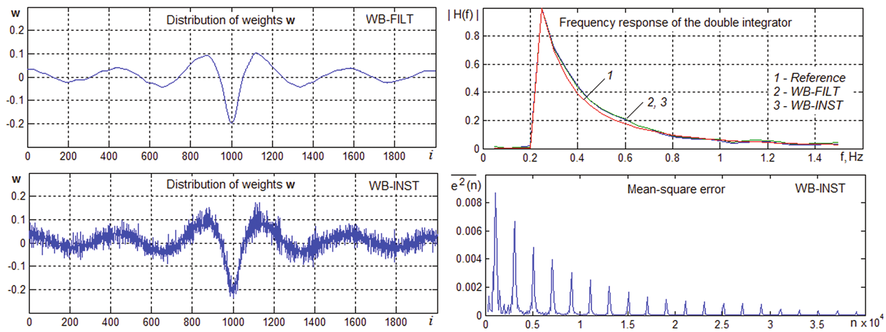

Fig. 2. Results of the mapping of the dynamical pulse sequences

5 Conclusions

The presented temporal supervised learning rules WB-INST and WB-FILT are applicable for using in digital adaptive systems with the reference PNN that performs the required linear transformations of the dynamical variables represented by pulse sequences.

The quantitative changes of synaptic weights are proportional to an error and reactions of the PN synaptic connections to input pulses. In such common formulation, the proposed temporal learning rules are similar to the known rules: ReSuMe [4], SPAN [5], PSD [6], INST and FILT [7].

However, an important distinction of the proposed temporal learning rules is that they are formulated in the discrete time in a general view. It allows deriving further variations of these rules oriented on specifics of processing tasks of dynamical variables. In addition, the offered PN model and rules due to the sparsity of binary vectors are quite effective from computational point of view.

References

1. Boerlin, M., Machens, C.K., Denève, S.: Predictive coding of dynamical variables in balanced spiking networks. PLoS Comput. Biol. **9**(11), 1–16 (2013). doi:10.1371/journal.pcbi.1003258
2. Memmesheimer, R.M., Rubin, R., Ölveczky, B.P., Sompolinsky, H.: Learning precisely timed spikes. Neuron **82**(4), 925–938 (2014). doi:10.1016/j.neuron.2014.03.026
3. Abbott, L.F., DePasquale, B., Memmesheimer, R.-M.: Building functional networks of spiking model neurons. Nat. Neurosci. **19**(3), 350–355 (2016)
4. Ponulak, F., Kasinski, A.: Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting. Neural Comput. **22**(2), 467–510 (2010). doi:10.1162/neco.2009.11-08-901
5. Mohemmed, A., Schliebs, S., Matsuda, S., Kasabov, N.: SPAN: Spike pattern association neuron for learning spatio-temporal spike patterns. Int. J. Neural Syst. **22**(4), 1–17 (2012). doi:10.1142/S0129065712500128
6. Yu, Q., Tang, H., Tan, K.C., Li, H.: Precise-Spike-Driven synaptic plasticity: learning hetero-association of spatiotemporal spike patterns. PLoS ONE **8**(11), 1–16 (2013). doi:10.1371/journal.pone.0078318
7. Gardner, B., Grüning, A.: Supervised learning in spiking neural networks for precise temporal encoding. PLoS ONE **11**(8), 1–28 (2016). doi:10.1371/journal.pone.0161335
8. Bondarev, V.N.: Pravila obucheniya impul'snogo nejrona dlya adaptivnoj obrabotki signalov (Training rules of pulse neuron for the adaptive signal processing). In: Proceedings of the XVIII All-Russian scientific and technical conference "Neuroinformatics-2016", part 2, pp. 192–202. NIYaU MIFI Publ., Moscow (2016)
9. Bondarev, V.: Vector-matrix models of pulse neuron for digital signal processing. In: Cheng, L., Liu, Q., Ronzhin, A. (eds.) Advances in Neural Networks—ISNN 2016. Lecture Notes In Computer Science, vol. 9719, pp. 647–656. Springer, Cham (2016). doi:10.1007/978-3-319-40663-3_74
10. Rusu, C.V., Florian, R.V.: A new class of metrics for spike trains. Neural Comput. **26**(2), 306–348 (2014). doi:10.1162/NECO_a_00545
11. Rossum, M.C.W.: Novel Spike Distance. Neural Comput. **13**, 751–763 (2001)
12. Bondarev, V.N., Smetanina, T.I.: Adaptivnyj sintez cifrovogo fil'tra dlya akselerometrich-eskogo volnografa (Adaptive synthesis of the digital filter for accelerometer wave gage). Sistemy kontrolya okruzhayushchej sredy. **2**(22), 25–28 (2015)