Neurocomputing 151 (2015) 310–318

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Improved perception-based spiking neuron learning rule for real-time user authentication ☆

Hong Qu *, Xiurui Xie, Yongshuai Liu, Malu Zhang, Li Lu

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, PR China

ARTICLE INFO

Article history:
Received 17 March 2014
Received in revised form
25 July 2014
Accepted 16 September 2014
Communicated by M.-J. Er
Available online 28 September 2014

Keywords:
Spiking neural networks
Perception-based spiking neuron learning rule
User authentication
Spatiotemporal pattern recognition

ABSTRACT

Spiking neural networks (SNNs) have been highly successful in spatiotemporal pattern recognition. As one of the most efficient supervised learning algorithms in spike sequences learning, the perceptron-based spiking neuron learning rule (PBSNLR) still has a relatively high computational complexity, which is difficult to use in a real-time system. In this paper, a novel method is presented to improve PBSNLR's efficiency without reducing its accuracy, and this method is applied to solve user authentication problem in real time. In our method, a user's behavioral biometric of sliding dynamic and finger pressure are selected as spatiotemporal features to recognize the user's identity. The temporal feature is obtained by the time coding of SNNs and the spatial feature is represented by the neurons' relative positions. Comprehensive experimental results demonstrate that our improved algorithm outperforms the traditional PBSNLR in terms of efficiency and exhibits excellent performance when identifying users of touch screen devices.

1. Introduction

As the third generation of artificial neural networks, spiking neural networks (SNNs) have been successfully applied to various domains such as human action recognition [1,2], image processing [3,4], path planning [5], and sound source localization [6]. The successful utilization of SNNs in these domains stems from their biological approach to processing information using temporal coding rather than rate coding [7], which offers a new way to represent information and allows the use of enormous computing power to process information [8].

To further explore the computational ability of SNNs, many advanced learning algorithms have been recently introduced in supervised learning. These algorithms can be categorized into two types: learning with single spike and learning with multi-spike sequences [9–11]. The former contains numerous methods [12,13] [14] among which SpikeProp [12] is the most typical method, combining a BP algorithm with SNNs to achieve supervised learning. Although single spike learning methods exhibit good performance when classifying problems, learning algorithms with multi-spikes increase network capacity and computing power.

The first multi-spike sequences learning algorithm is ReSuMe [15], which attains sequence learning, classification, and spike shifting using the Widrow-Hoff rule and spike-time-dependent plasticity (STDP). The ReSuMe was improved by Sporea [16]. After that, the perceptron-based spiking neuron learning rule (PBSNLR) was proposed by Yan [17]. It transforms the supervised learning task into a classification problem that can be solved using the perceptron learning rule. Although SNNs have numerous efficient learning methods, such as the PBSNLR, and can be successfully applied to various applications, their efficiencies are not high enough to meet the requirements of real-time applications. For this study, we improve the PBSNLR using dynamic learning rates and a dynamic selection method of training samples. Using these strategies, the PBSNLR's efficiency improves drastically. Our efforts are illustrated in the following sections.

We apply the improved algorithm to a user authentication system to demonstrate its high efficiency. This user authentication system is employed in real-time touch screen devices such as smartphones or PAD. In contrast to one-time unlock protection, which suffers from several limitations such as passwords being stolen, lost, or acquired by others [18], our real-time user authentication method protects user information all the time, even when devices are in use. A user's behavioral biometric of sliding motions and pressure are used as features in our spiking neural classifier. It has been illustrated that user authentication systems with biometric-based features possess more advantages over other security features, because they cannot be stolen or forgotten [19]. Two main contributions are discussed in

☆This work is supported by the National Science Foundation of China under Grant 61273308 and 61173171, the Fundamental Research Funds for Central Universities under Grant ZYGX2012J068 and ZYGX2012J072.

* Corresponding author.
E-mail address: hongqu@uestc.edu.cn (H. Qu).

http://dx.doi.org/10.1016/j.neucom.2014.09.034
0925-2312/

this paper: (1) The efficiency of the PBSNLR algorithm is improved to promote the learning efficiency of the supervised learning of SNNs. (2) The user authentication system is presented by a spiking neural classifier with the developed PBSNLR algorithm to meet real-time requirements.

The rest of this paper is organized as follows: Section 2 illustrates the manner in which the efficiency of the PBSNLR algorithm is improved. In Section 3, several numerical experiments are conducted to investigate learning performance of our improved algorithm compared with the traditional PBSNLR. The real time authentication system is proposed in Section 4, with its recognition results shown and analyzed in Section 5. Section 6 states conclusions and future research.

2. PBSNLR algorithm with dynamic parameters

2.1. PBSNLR algorithm

Supervised learning in time-encoded SNNs attempts to establish a link between input and target output sequences. The PBSNLR transforms this task into a classification problem using the perceptron learning rule [17] with the sample defined as follows.

All target time points are regarded as positive samples and all no-target time points are negative samples. These negative samples are selected by a uniform distribution on the neurons' running time (removing target time points). Positive samples are misclassified if there is no spike at that time, and conversely, a negative sample is misclassified if the voltage of output neuron at that time exceeds threshold [17]. All of these misclassified samples are trained using the perceptron learning rule in the following equations:

$$W_i(n) = \begin{cases} W_i(n-1) - \alpha U_t & \text{if } d_t = 0 \quad \text{and} \quad a_t = 1, \\ W_i(n-1) + \alpha U_t & \text{if } d_t = 1 \quad \text{and} \quad a_t = 0, \\ W_i(n-1) & \text{if } d_t = a_t, \end{cases} \tag{1}$$

where $W_i(n)$ is the weight of the ith synapse at generation n, α is the learning rate, and U_t is the transmit voltage of the synapse at time t. $d_t = 0$ indicates that time t is not the target time, and $a_t = 0$ means that the voltage of the output neuron is below threshold at time t.

2.2. The improved PBSNLR algorithm

The PBSNLR algorithm has been shown to be effective in classification problems, however, it still suffers from two limitations:

(1) The learning process is instructed by only one learning parameter α, and after its assignment, it is fixed in a learning process. Consequently, weight changes drastically or slowly for both negative and positive samples in every learning period, which leads to low learning efficiency. Employing only one fixed learning parameter here is not an optimal option because it cannot detect surrounding environment to determine its proper value to complete learning efficiently.

(2) The method that introduced in PBSNLR to select negative samples [17] is inserting time points uniformly and detecting their voltage. Since the continuity of the voltage function at most time (except when a spike is emitted), this sample selective method is easy to cause the situation that numerous misclassified negative samples clustered, which is shown in Fig. 1(a). While weight modification of so much clustered negative samples will affect the voltage of their next nearest positive sample significantly, and then lead to accuracy oscillation. Fig. 1 shows this situation.

Here we propose two strategies to improve the PBSNLR. First, dynamic learning parameters are applied. The time distance item D_t is introduced, which is calculated using Eq. (3). It expresses the time distance of the current negative sample t and its nearest positive sample t_{dt} with $t_{dt} > t$ (for example, in Fig. 1(a), the distance between t_1 and its next nearest positive sample t_d is denoted by D_{t_1}). Weights modification at misclassified negative points are calculated according to the following equations:

$$\Delta w_{neg} = \beta_1 D_t U_t, \tag{2}$$

with

$$D_t = \| t - t_{dt} \|. \quad t = 0, \Delta t, 2\Delta t, \ldots T. \tag{3}$$

Using this time distance parameter, weights change slowly if negative sample t is close to its next positive sample t_{dt}, and conversely, weights vary drastically. This dynamic parameter can adjust its value according to the learning requirements, and mitigates the interference of the negative samples' training with the positive to improves learning efficiency.

Similarly, to decrease the impact of the positive points' weight modification with the negative, a dynamic parameter D_v is presented for each misclassified positive sample t_{dt}. It is expressed by Eq. (5) to denote the voltage difference of threshold ϑ and voltage at time t_{dt}, with $U_{t_{dt}}$ calculated using the simplified spike response model (SRM_0) of spiking neural networks (see appendix for details of models). D_v can avoid excessive weight modification in time t_{dt} if its voltage is close to threshold. Weights at misclassified positive samples are adjusted according to the following equations:

$$\Delta w_{posi} = \beta_2 D_v U_t, \tag{4}$$

with

$$D_v = \| \vartheta - U_{t_{dt}} \|. \tag{5}$$

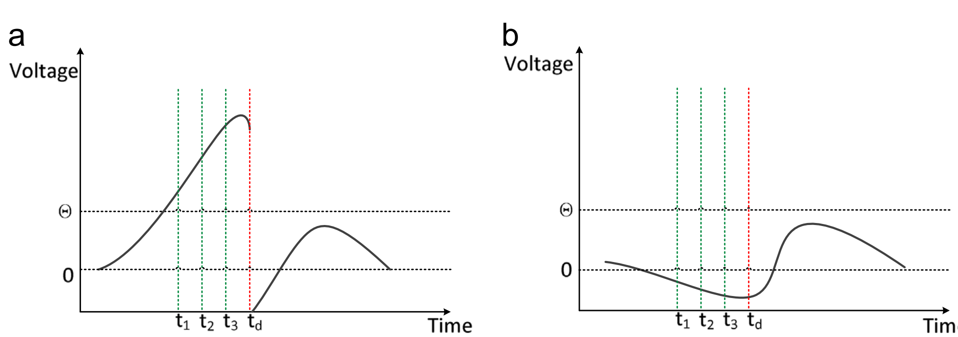

Fig. 1. Training with traditional PBSNLR. (a) Voltage of the output neuron before training. t_1,t_2,t_3 are no-target time but their voltage exceeds threshold, then they are all chosen as misclassified negative samples. t_d is a positive sample which is classified correctly because its voltage exceeds threshold. (b) Voltage of the output neuron after training. Clearly, the training of negative samples t_1,t_2,t_3 leads to misclassification of t_d. Conversely, the strengthened weights at t_d with a large α also lead to misclassification of negative samples.