Training Spiking Neurons by Means of Particle Swarm Optimization

Conference Paper · June 2011

DOI: 10.1007/978-3-642-21515-5_29 · Source: dx.doi.org

2 authors:

Roberto Antonio Vázquez
Universidad La Salle
79 PUBLICATIONS **639** CITATIONS
SEE PROFILE

Beatriz A Garro
Universidad Nacional Autónoma de México
28 PUBLICATIONS **311** CITATIONS
SEE PROFILE

用粒子群优化方法训练尖峰神经元

Training Spiking Neurons by Means of Particle Swarm Optimization

Roberto A. Vázquez[1] and Beatriz A. Garro[2]

[1] Intelligent Systems Group
Faculty of Engineering - La Salle University
Benjamín Franklin 47 Col. Condesa CP 06140 México, D.F.
[2] Center for Computing Research - IPN
Av. Juan de Dios Batiz sn, Col. Nueva Industrial Vallejo, CP 07738 México, D.F.
ravem@lasallistas.org.mx, bgarrol@ipn.mx

Abstract. Meta-heuristic algorithms inspired by nature have been used in a wide range of optimization problems. These types of algorithms have gained popularity in the field of artificial neural networks (ANN). On the other hand, spiking neural networks are a new type of ANN that simulates the behaviour of a biological neural network in a more realistic manner. Furthermore, these neural models have been applied to solve some pattern recognition problems. In this paper, it is proposed the use of the particle swarm optimization (PSO) algorithm to adjust the synaptic weights of a spiking neuron when it is applied to solve a pattern classification task. Given a set of input patterns belonging to K classes, each input pattern is transformed into an input signal. Then, the spiking neuron is stimulated during T ms and the firing rate is computed. After adjusting the synaptic weights of the neural model using the PSO algorithm, input patterns belonging to the same class will generate similar firing rates. On the contrary, input patterns belonging to other classes will generate firing rates different enough to discriminate among the classes. At last, a comparison between the PSO algorithm and a differential evolution algorithm is presented when the spiking neural model is applied to solve non-linear and real object recognition problems.

1 Introduction

James Kennedy and Russell C. Eberhart proposed the particle swarm optimization (PSO) algorithm in 1995 [11]. This algorithm is a method for combinatorial or numerical optimization problems. Several versions of this algorithm have been proposed in the last years; however, the principal characteristic of this algorithm is that it can optimize over non-linear and non-continuous search spaces. In addition, this PSO algorithm is based on the collective behavior of the bird flocking, fish schooling. This behavior consists on moving to another better place for eating or getting a better life.

The PSO algorithm has been applied in a wide range of problems, including those related to the field of artificial neural networks (ANN). These types of algorithms are non-gradient approaches; this characteristic makes them a promising learning strategy for the training (adjusting the synaptic weights) of an ANN [12], [13] and [14]. Furthermore, the PSO algorithm has been applied to design automatically feed-forward neural networks; this includes the design of the best topology, the transfer function of each neuron and the synaptic weights [1].

One of the most common applications of an ANN is a pattern classification task. However, though these models have been applied to solve several pattern recognition problems, they have some constrains that limits their applicability in real-life problems, particularly during the design of the ANN. For that reason, it is mandatory to explore new ways of implementing ANN.

Recently, a new type of neural model, called spiking neuron models, emerged. These models have been called the 3rd generation of artificial neural networks [2]. The spiking neuron models increase the level of realism in a neural simulation and incorporate the concept of time. They also have been applied in a wide range of areas from the field of computational neurosciences such as: brain region modeling, auditory processing [3], visual processing [4], robotics [5] and so on. Nonetheless, their application for solving pattern recognition problems is gaining popularity in the field of artificial intelligence.

In [17], the authors described how a genetic algorithm can be used during the learning process of a spiking neural network; however, the model was not applied to perform a pattern recognition problem. In [16], the authors show how a spiking neural network can be trained by a quantum PSO and then applied to a string pattern recognition problem. In [18] the authors trained a spiking neural network using a PSO algorithm, and then it is applied in three pattern recognition problems; however, in each problem, the authors had to design the topology of the network. In [10,15,19], the authors shown how only one spiking neuron such as Leaky-Integrate-and-Fire and Izhikevich models [6,7] can be applied to solve different linear and non-linear pattern recognition problems.

In this paper is shown how it is possible to apply a spiking neuron, trained with a PSO algorithm, in different linear and non-linear pattern recognition problems. Given a set of input patterns belonging to K classes, each input pattern is transformed into an input signal. After that, the spiking neuron is stimulated during T ms and the firing rate is computed. Once the synaptic weights of the neuron model were adjusted by the PSO algorithm, we expect that the input patterns which belong to the same class generate similar firing rate. On the contrary, patterns belonging to other classes generate firing rates different enough to discriminate among the classes. Finally, the proposed method is compared against the method described in [19] when the spiking neuron is applied to solve pattern recognition problems.

2 Spiking Neuron Model

A typical spiking neuron can be divided into three functionally distinct parts, called dendrites, soma, and axon. The dendrites play the role of the *input device* that collects signals from other neurons and transmits them to the soma. The soma is the *central processing unit* that performs an important non-linear processing step: if the total input exceeds a certain threshold, then an output signal is generated. The output signal is taken over by the *output device*, the axon, which delivers the signal (spike train) to other neurons.

Since all spikes of a given neuron look alike, the form of the action potential does not carry any information. Rather, it is the number and the timing of spikes, which matter.

In this paper we adopt the Izhikevich model

$$\begin{array}{ll} C\dot{v} = k\left(v - v_r\right)\left(v - v_t\right) - u + I & \text{if} v \geq v_{peak} \text{then} \\ \dot{u} = a\left\{b\left(v - v_r\right) - u\right\} & v \leftarrow c, u \leftarrow u + d \end{array} \tag{1}$$

which has only 9 dimensionless parameters. Depending on the values of a and b, it can be an integrator or a resonator. The parameters c and d take into account the action of high-threshold voltage-gated currents activated during the spike, and affect only the after-spike transient behavior. v is the membrane potential, u is the recovery current, C is the membrane capacitance, v_r is the resting membrane potential, and v_t is the instantaneous threshold potential. The parameters k and b can be found when one knows the neuron's rheobase and input resistance. The sign of b determines whether u is an amplifying ($b < 0$) or a resonant ($b > 0$) variable. The recovery time constant is a. The spike cutoff value is v_{peak} , and the voltage reset value is c. The parameter d describes the total amount of outward minus inward currents activated during the spike and affecting the after-spike behavior. A detail description of the model can be found in [6,7].

3 Proposed Method

It is important to point out that when the input current signal changes the response of the spiking neuron also changes, generating different firing rates. The firing rate is computed as the number of spikes generated in an interval of duration T divided by T. The neuron is stimulated during T ms with an input signal and fires when its membrane potential reaches a specific value generating an action potential (spike) or a train of spikes.

The proposed method is inspired in the method described in [19]. We expect that patterns which belong to the same class generate similar firing rates and patterns, which belong to other classes generate firing rates different enough to discriminate among the classes.

Let D=$\left\{\mathbf{x}^i, k\right\}_{i=1}^{p}$ be a set of p input patterns where $k = 1, \ldots, K$ is the class to which $\mathbf{x}^i \in \mathbb{R}^n$ belongs. First of all, each input pattern is transformed into an input signal I. The spiking neuron model is not directly stimulated with the input pattern $\mathbf{x}^i \in \mathbb{R}^n$, but with an injection current I. Since synaptic weights of the model are directly connected to the input pattern $\mathbf{x}^i \in \mathbb{R}^n$, the injection current generated with this input pattern can be computed as

$$I = \gamma \cdot \mathbf{x} \cdot \mathbf{w} \tag{2}$$

where $\mathbf{w}^i \in \mathbb{R}^n$ is the set of synaptic weights of the neuron model and $\gamma = 100$ is a gaining factor which guarantees that the neuron will fire.

After that, the spiking neuron is stimulated using I during T ms and then the firing rate of the neuron is computed. With this information, the average firing rate $\mathbf{AFR} \in \mathbb{R}^K$ of each class is computed.

At last, the class of an input pattern $\tilde{\mathbf{x}}$ is determined by means of the firing rates as follows:

$$cl = \arg\min_{k=1}^{K} \left(|AFR_k - fr| \right) \tag{3}$$

where fr is the firing rate generated by the neuron model stimulated with the input pattern $\tilde{\mathbf{x}}$.

The synaptic weights of the model, which are directly connected to the input pattern, determines the firing rate of the neurons. This means that our learning phases consist in generating the desired behavior by adjusting the synaptic weights of the neuron. The learning phase will be done by the particle swarm optimization algorithm.

4 Adjusting Synapses of the Neuron Model

Synapses of the neuron model $\boldsymbol{w}$ are adjusted by means of the PSO algorithm. As we already said, the PSO algorithm is a powerful and an efficient technique for optimizing non-linear and non-differentiable continuous space functions. This algorithm works with a population of particles $\mathbf{x}_i$ that represent solutions (positions). Each particle changes its position with the time t. They are guided by the knowledge of the best particle from whole population and also by its own knowledge as so to search the optimums value. In this case, each one is guided by a velocity function that evolves a social component $\mathbf{p}_g$ (the best particle of all) and a cognitive component $\mathbf{p}_i$ (the best position of each particle). The velocity function limited to the range $[v_{min}, v_{max}]$ help to find the best positions in the search space. This function is shown in the next equation:

$$\mathbf{v}_i(t+1) = \omega \mathbf{v}_i(t) + c_1 r_1 (\mathbf{p}_i(t) - \mathbf{x}_i(t)) + c_2 r_2 (\mathbf{p}_g(t) - \mathbf{x}_i(t)) \tag{4}$$

where ω is the inertia weight $[1, 0)$ that changes each iteration, c_1 and c_2 are acceleration coefficients; $r_1 \sim U(0,1)$ and $r_2 \sim U(0,1)$ are uniformly distributed random numbers called the craziness After that, each particle i has to update its new position computing by the next equation:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{5}$$

Moreover, each possible solution is measured by means of fitness function in order to know its aptitude. It is desired that this solution be the optimums or the nearest to it.

Finally, the PSO algorithm for the real version could be performed as follows:

```
Given a population of $\mathbf{x}_i \in \mathbb{R}^D, i = 1, \ldots, M$ individuals

1. Initialize the population at random.
2. Until a stop criteria is reached:
   (a) For each individual $\mathbf{x}_i$ evaluate their fitness
```

```
(b) For each individual $i$, update its best position $\mathbf{p}_i$.
(c) From all individual $i$, update the best individual $\mathbf{p}_g$.
(d) For each individual $i$, compute the velocity update equation $\mathbf{v}_i(t+1)$
    and then compute the current position $\mathbf{x}_i(t+1)$.
```

To find the synaptic weights that maximize the accuracy of the spiking neural model during a pattern recognition task, the next fitness function was proposed:

$$f(\boldsymbol{w}, D) = 1 - performance(\boldsymbol{w}, D) \tag{6}$$

where $\boldsymbol{w}$ are the synapses of the model, D is the set of input patterns and $performance(\boldsymbol{w}, D)$ is a function which follows the steps described in above section and computes the classification rate given by the number of patterns correctly classified divided by the number of tested patterns.

5 Experimental Results

Several experiments were performed in order to evaluate the accuracy of the proposed method. Five of them were taken from the UCI machine learning repository [8]: iris plant, glass, diabetes, liver-bupa and wine datasets. In addition, another dataset was generated from a real object recognition problem [9].

The parameters for the Izhikevich neuron were defined as $C = 100$, $v_r = -60$, $v_t = -40$, $v_{peak} = 35$, $k = 0.7$, $a = 0.03$, $b = -2$, $c = -50$, and $d = 100$. The Euler method was used to solve the differential equation of the model with $dt = 1$. The parameter to compute input current I from the input pattern was set to $\theta = 100$ with a duration of $T = 1000$. For the case of the particle swarm optimization algorithm, $NP = 40$, $MAXGEN = 1000$, $VMAX = 4$, $VMIN = -4$, $c_1 = 2$, $c_2 = 2$, $XMAX = 10$, $XMIN = -10$ and ω was varied in the range $[0.95 - 0.4]$ throught the generations.

The accuracy (classification rate) achieved with the proposed method was computed as the number of input patterns correctly classified divided by the total number of tested patterns. To validate the accuracy of the method 10 experiments over each dataset were performed. Something important to notice is

Table 1. Average accuracy provided by the methods using different databases

Dataset	Method using DE [19]		Proposed method using PSO	
	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.
Wine	0.9796	0.8744	0.9782	0.8879
Iris plant	0.9933	0.9833	0.9933	0.97
Glass	0.8158	0.7411	0.8178	0.7457
Diabetes	0.8038	0.7371	0.7990	0.7619
Liver	0.7620	0.6870	0.7591	0.6754
Object rec.	1	0.9850	1	0.9950

Tr. cr = Training classification rate, Te. cr. = Testing classification rate.

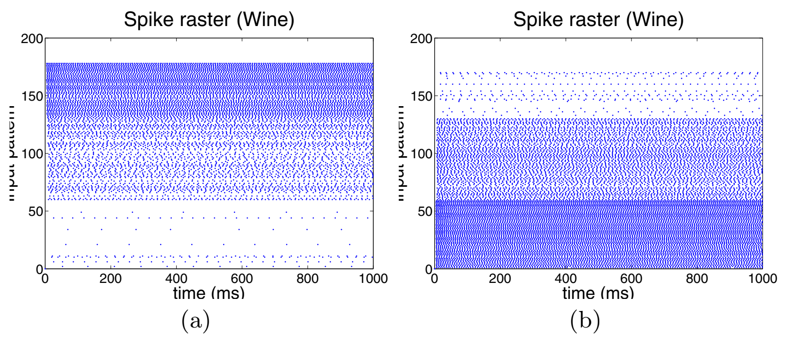

Fig. 1. Two of the experimental results obtained with the wine dataset. Notice that 3 different firing rates which correspond to 3 classes can be observed.

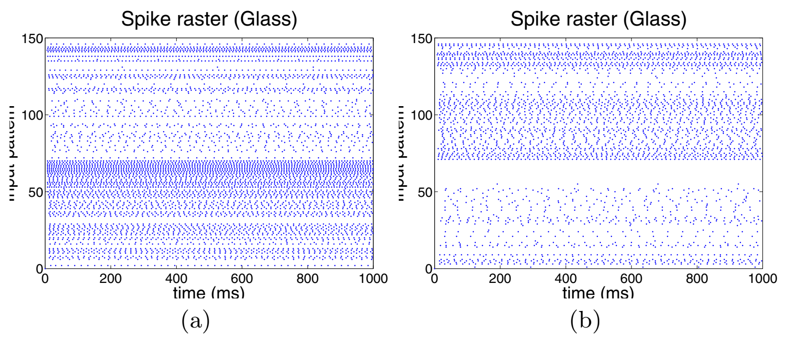

Fig. 2. Two of the experimental results obtained with the glass dataset. Notice that 2 different firing rates which correspond to 2 classes can be observed.

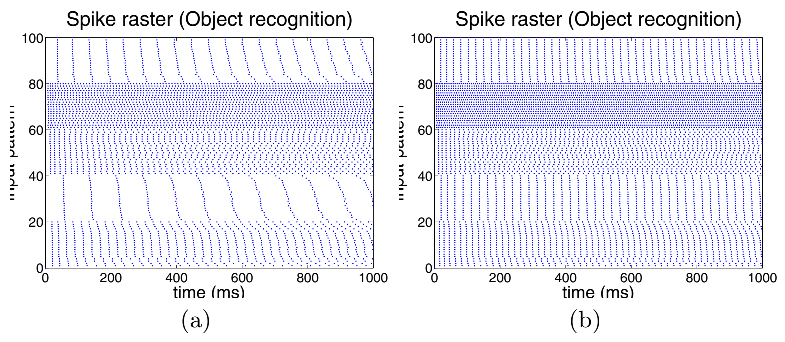

Fig. 3. Two of the experimental results obtained with the object recognition dataset. Notice that 5 different firing rates which correspond to 5 classes can be observed.

that in each experiment the datasets were randomly split into two new partitions: 50% of the patterns for training and the remain for testing.

Due to space only some of the experimental results achieved with the proposed method are shown in Fig 1 - Fig 3. As can be appreciated from these figures, the set of synaptic weights found with the PSO algorithm provokes that the spiking neuron generates almost the same firing rate when it is stimulated with input patterns from the same class. On the contrary, the spiking neuron generates firing rates different enough to discriminate among the different classes when it is stimulated with input patterns which belong to different classes.

The average classification rate computed from all experimental results is shown in Table 1. These preliminary results suggest that spiking neurons trained with a PSO algorithm can be considered as an alternative way to perform different pattern recognition tasks. As the reader can appreciate, only one spiking neuron was enough to solve a pattern recognition problem with a high acceptable accuracy. In general, the accuary of the proposed method was comparable to the method described in [19]. Both methods provide similar results; however, in some problems the proposed method was slightly better than the other, and vice verse.

We can also conjecture that if only one neuron is capable of solving pattern recognition problems, perhaps several spiking neurons working together can improve the experimental results obtained in this research. However, that is something that should be proven.

6 Conclusions

In this paper a new method to apply a spiking neuron in a pattern recognition task was proposed. This method is based on the firing rates produced with a spiking neuron when is stimulated. The training phase of the neuron model was done by means of a particle swarm optimization algorithm. After training, we observed that input patterns, which belong to the same class, generate almost the same firing rates and input patterns, which belong to different classes, generate firing rates different enough to be discriminate among the different classes.

Through several experiments, we can conclude that spiking neurons can be considered as an alternative tool to solve pattern recognition problems. Concerning to the strategy adopted to adjust the synaptic weights, we could observe that the results achieved with the particle swarm optimization algorithm are comparable to those provided by the differential evolution algorithm.

Nowadays, we are developing a methodology to calculate the maximum number of categories that the spiking neuron can classify. Furthermore, we are researching different alternatives of combining several types of spiking neurons in one network to improve the results obtained in this research and then apply it in more complex pattern recognition problems such as face, voice and 3D object recognition.

Acknowledgements

The author thanks Universidad La Salle for the economical support under grant number ULSA I-113/10.

References

1. Garro, B.A., Sossa, H., Vazquez, R.A.: Design of Artificial Neural Networks using a Modified Particle Swarm Optimization Algorithm. IJCNN, 938–945 (2009)
2. Maass, W.: Networks of spiking neurons: the third generation of neural network models. Neural Networks 10(9), 1659–1671 (1997)
3. Loiselle, S., Rouat, J., Pressnitzer, D., Thorpe, S.: Exploration of rank order coding with spiking neural networks for speech recognition. IJCNN 4, 2076–2080 (2005)
4. Thorpe, S.J., Guyonneau, R., et al.: SpikeNet: Real-time visual processing with one spike per neuron. Neurocomputing 58(60), 857–864 (2004)
5. Di Paolo, E.A.: Spike-timing dependent plasticity for evolved robots. Adaptive Behavior 10(3), 243–263 (2002)
6. Izhikevich, E.M.: Simple model of spiking neurons. IEEE Trans. on Neural Networks 14(6), 1569–1572 (2003)
7. Izhikevich, E.M.: Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. The MIT press, Cambridge (2007)
8. Murphy, P.M., Aha, D.W.: UCI repository of machine learning databases. Dept. Inf. Comput. Sci., Univ. California, Irvine (1994)
9. Vazquez, R.A., Sossa, H.: A new associative model with dynamical synapses. Neural Processing Letters 28(3), 189–207 (2008)
10. Vazquez, R.A., Cachon, A.: Integrate and fire neurons and their application in pattern recognition. In: Proceedings of the 7th CCE, pp. 424–428 (2010)
11. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. IV, pp. 1942–1948 (1995)
12. Zhao, L., Yang, Y.: PSO-Based Single Multiplicative Neuron Model for Time Series Prediction. Expert Systems with Applications 36, 2805–2812 (2009)
13. Yu, J., et al.: An Improved Particle Swarm Optimization for Evolving Feedforward Artificial Neural Networks. Neural Processing Letters 26, 217–231 (2007)
14. Gudise, V.G., Venayagamoorthy, G.K.: Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. In: Proceedings of the IEEE Swarm Intelligence Symposium, pp. 110–117 (2003)
15. Vázquez, R.A.: Pattern recognition using spiking neurons and firing rates. In: Kuri-Morales, A., Simari, G.R. (eds.) IBERAMIA 2010. LNCS, vol. 6433, pp. 423–432. Springer, Heidelberg (2010)
16. Hamed, H.N., Kasabov, N., Michlovský, Z., Shamsuddin, S.M.: String Pattern Recognition Using Evolving Spiking Neural Networks and Quantum Inspired Particle Swarm Optimization. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009. LNCS, vol. 5864, pp. 611–619. Springer, Heidelberg (2009)
17. Kamoi, S., et al.: Pulse Pattern Training of Spiking Neural Networks Using Improved Genetic Algorithm. In: Proceedings of the IEEE CIRA, pp. 977 – 981 (2003)
18. Hong, S., et al.: A Cooperative Method for Supervised Learning in Spiking Neural Networks. In: 14th CSCWD, pp. 22–26 (2010)
19. Vazquez, R.A.: Izhikevich Neuron Model and its Application in Pattern Recognition. Australian Journal of Intelligent Information Processing Systems 11, 35–40 (2010)