Izhikevich Neuron Model and its Application in Pattern Recognition

Roberto A. Vázquez

Escuela de Ingeniería - Universidad La Salle
Benjamín Franklin 47 Col. Condesa CP 06140 México, D.F.
ravem@lasallistas.org.mx

Abstract. In this paper is shown how an Izhikevich neuron can be applied to solve 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, then the Izhikevich neuron is stimulated during T ms and finally the firing rate is computed. After adjusting the synaptic weights of the neural model, input patterns belonging to the same class will generate almost the same firing rate and input patterns belonging to different classes will generate firing rates different enough to discriminate among the different classes. At last, a comparison between a feed-forward neural network and the Izhikevich neural model is presented when they are applied to solve non-linear and real object recognition problems.

1 Introduction

Spiking neuron models have been called the 3rd generation of artificial neural networks [2]. These models increase the level of realism in a neural simulation and incorporate the concept of time. Spiking models have been applied in a wide range of areas from the field of computational neurosciences [3] such as: brain region modeling [4], auditory processing [5, 6], visual processing [7, 8], robotics [9, 10] and so on.

In this paper is shown how an Izhikevich neuron [11–13] can be applied to solve 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, then the spiking neuron is stimulated during T ms and finally the firing rate is computed. After adjusting the synaptic weights of the neuron model, we expect that input patterns belonging to the same class generate almost the same firing rate; on the other hand, we also expect that input patterns belonging to different classes generate firing rates different enough to discriminate among the different classes. Finally, a comparison against a feed-forward neural network trained with the well-known backpropagation and Levenberg-Marquartd algorithms, and the proposed method is presented when they are applied to solve non-linear and real object recognition problems.

2 Izhikevich 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.

The Izhikevich model

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

has only nine dimensionless parameters. Depending on the values of a and b, it can be an integrator or a resonator. The parameters c and d do not affect steady-state sub-threshold behavior. Instead, they 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 [13].

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.

Various choices of the parameters result in various intrinsic firing patterns including [11]: RS (*regular spiking*) neurons are the most typical neurons in the cortex; IB (*intrinsically bursting*) neurons fire a stereotypical burst of spikes followed by repetitive single spikes; CH (*chattering*) neurons can fire stereotypical bursts of closely spaced spikes; FS (*fast spiking*) neurons can fire periodic trains of action potentials with extremely high frequency practically without any adaptation(slowing down); and LTS (*low-threshold spiking*) neurons can also fire high-frequency trains of action potentials, but with a noticeable spike frequency adaptation.

3 Proposed method

Before describing the proposed method applied to solve pattern recognition problems, it is important to notice that when the input current signal changes, the response of the Izhikevich 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.

Let D=$\{\mathbf{x}^i, k\}_{i=1}^{p}$ 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, 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 can be computed.

During training phase, the synaptic weights of the model, which are directly connected to the input pattern, are adjusted by means of a differential evolution algorithm.

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

$$cl = \arg\min_{k=1}^{K} (|AFR_k - fr|) \tag{2}$$

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

Izhikevhic 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

$$I = \gamma \cdot \mathbf{x} \cdot \mathbf{w} \tag{3}$$

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.

3.1 Adjusting synapses of the neuron model

Synapses of the neuron model $\boldsymbol{w}$ are adjusted by means of a differential evolution algorithm. Evolutionary algorithms not only have been used to design artificial neural networks [1], but also to evolve structure-function mapping in cognitive neuroscience [14] and compartmental neuron models [15].

Differential evolution (DE) is a powerful and efficient technique for optimizing non-linear and non-differentiable continuous space functions [16]. DE has a lower tendency to converge to local maxima with respect to the conventional genetic algorithm, because it simulates a simultaneous search in different areas of solution space. Moreover, it evolves populations with a smaller number of individuals and have a lower computation cost.

DE begins by generating a random population of candidate solutions in the form of numerical vectors. The first of these vectors is selected as the target vector. Next, differential evolution builds a trial vector by executing the following sequence of steps:

1. Randomly select two vectors from the current generation.
2. Use these two vectors to compute a difference vector.
3. Multiply the difference vector by weighting factor F.
4. Form the new trial vector by adding the weighted difference vector to a third vector randomly selected the current population.

The trial vector replaces the target vector in the next generation if and only if the trial vector represents a better solution, as indicated by its measured cost value computed with a fitness function. DE repeats this process for each of the remaining vectors in the current generation. Then, it replaces the current generation with the next generation, and continues the evolutionary process over many generations.

In order to find the set of synaptic weights, which maximize the accuracy of the Izhikevich neural model during a pattern recognition task, the next fitness function was proposed

$$f(\boldsymbol{w}, D) = 1 - performance(\boldsymbol{w}, D) \tag{4}$$

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 computes the classification rate given by the number of patterns correctly classified divided by the number of tested patterns.

4 Experimental results

To evaluate the accuracy of the proposed method, several experiments using three datasets were performed. Two of them were taken from the UCI machine learning benchmark repository [17]: iris plant and wine datasets. The other one was generated from a real object recognition problem.

The iris plant dataset is composed of three classes and each input pattern is composed of four features. The wine dataset is composed of three classes and each input pattern is composed of 13 features. For the case of the real object recognition problem, a dataset was generated from a set of 100 images which contains five different objects whose images are shown in Fig. 1 [18]. Objects were not recognized directly from their images, but by an invariant description of each object. Several images of each object in different positions, rotations and scale changes were used. To each image of each object, a standard thresholder [19] was applied to get its binary version. Small spurious regions were eliminated from each image by means of a size filter [20]. Finally, the seven well-known Hu moments invariant, to translations, rotations and scale changes [21], were computed to build the object recognition dataset.

(a) (b) (c) (d) (e) (f)

Fig. 1. (a)-(c) Some of the images used to train the proposed method. (d)-(f) Some of the images used to test the proposed method.

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 differential evolution algorithm, $NP = 40$, $MAXGEN = 1000$, $F = 0.9$, $XMAX = 10$, $XMIN = -10$ and $CR = 0.8$.

The classic back-propagation and Levenberg-Marquardt algorithms were used to train the feed-forward neural network. The number of generations was set to 10000 and learning rate $\alpha = 0.01$. Concerning to the architecture of the feed-forward neural network, one hidden layer composed of 13 *hyperbolic tangent* neuron and an output layer composed of *linear* neurons were used in all experiments. The stop criterion for the three algorithms was the number of generations or the minimum error which was set to $e = 0$.

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 input patterns. To validate the accuracy of the proposed method 20 experiments over each dataset were performed. The same metric and number of experiments was used to measure the accuracy of the feed-forward neural network trained with the two different algorithms. Something important to notice is that in each experiment a new set of partitions over each dataset was generated by means of the 5-fold-cross validation strategy.

The experimental results, obtained with the iris, wine and object recognition datasets, are shown in Fig 2, Fig 3 and Fig 4, respectively. As can be appreciated from these figures, the set of synaptic weights found with the DE algorithm provokes that the Izhikevich neuron generates almost the same firing rate when it is stimulated with input patterns from the same class; in the contrary, the Izhikevich 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. The results obtained with the spiking neuron model, trained with the proposed method, improve the results obtained with feed-forward neural networks. Something that should be remarked is that while the feed-forward neural networks were composed of more than 13 neurons, the proposed method only used one Izhikevich neuron.

On the other hand, we also compared the accuracy of the proposed method using the Izhikevich neuron against the method described in [22] which uses a Leaky-Integrate-and-Fire (LIF) neuron. The accuracy of both models was comparable; however, the results achieved with the Izhikevich model were slightly better.

These preliminary results suggest that spiking neurons can be considered as an alternative way to perform different pattern recognition tasks.

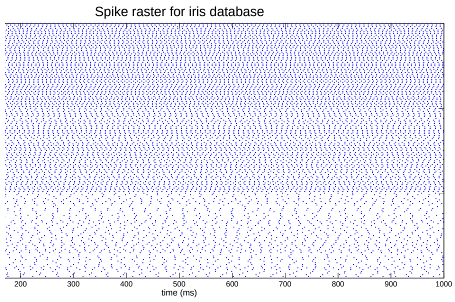

Fig. 2. Experimental results obtained with the iris plant dataset. Notice that three different firing rates which correspond to three different classes can be observed.

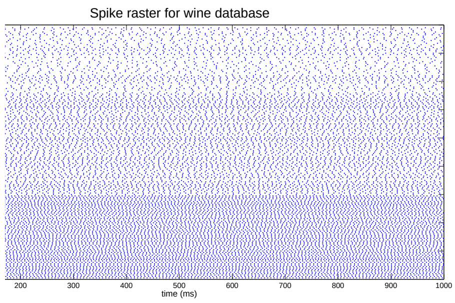

Fig. 3. Experimental results obtained with the wine dataset. Notice that three different firing rates which correspond to three different classes can be observed.

We can also conjecture that if only one neuron is capable to solve 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 proved.

5 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 an Izhikevich neuron when is stimulated. The firing rate is computed as the number of spikes generated in an interval of duration T divided by T.

The training phase of the neuron model was done by means of a differential evolution algorithm. After training, we observed that input patterns, which belong to the same class, generate almost the same firing rates (low standard

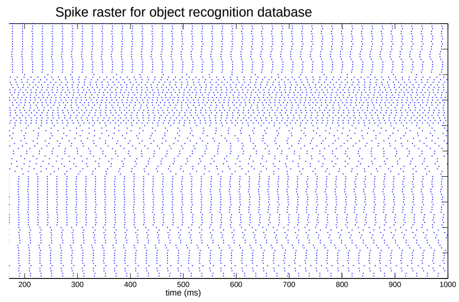

Fig. 4. Experimental results obtained with the object recognition dataset. Notice that five different firing rates which correspond to five different classes can be observed.

deviation) and input patterns, which belong to different classes, generate firing rates different enough (average spiking rate of each class widely separated) to be discriminate among the different classes.

Through several experiments, we observed that on the one hand the proposed method significantly improves the results obtained with feed-forward neural networks; on the other hand, this methodology slightly improves the results compared against those provides using a LIF neuron. Finally, we can conclude that spiking neurons can be considered as an alternative tool to solve pattern recognition problems.

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 Izhikevich 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.

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

Dataset	Back-propagation algorithm		Levenberg-Marquartd algorithm		Proposed method using LIF		Proposed method using IZ	
	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.
Iris plant	0.8921	0.8383	0.8867	0.7663	0.9988	0.9458	1	0.9308
Wine	0.4244	0.3637	1	0.8616	0.9783	0.7780	0.9993	0.8319
Object rec.	0.4938	0.4125	0.6169	0.4675	0.8050	0.7919	1	0.9912

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

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. Rieke, F. et al.: Spikes: Exploring the Neural Code. Brad-ford Book (1997)
4. Hasselmo, M. E., Bodelon, C. et al.: A Proposed Function for Hippo-campal Theta Rhythm: Separate Phases of Encoding and Retrieval Enhance Re-versal of Prior Learning. Neural Computation 14, 793–817 (2002)
5. Hopfield, J. J., Brody, C. D.: What is a moment? Cortical sensory integration over a brief interval. PNAS 97(25), 13919–24 (2000)
6. 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)

7. Azhar, H., Iftekharuddin, K. et al.: A chaos synchronization-based dynamic vision model for image segmentation. IJCNN 5, 3075–3080 (2005)
8. Thorpe, S. J., Guyonneau, R. et al.: SpikeNet: Real-time visual processing with one spike per neuron. Neurocomputing 58(60), 857–864 (2004)
9. Di Paolo, E. A.: Spike-timing dependent plasticity for evolved robots. Adaptive Behavior 10(3), 243–263 (2002)
10. Floreano, D., Zufferey, J. et al.: From wheels to wings with evolutionary spiking neurons. Artificial Life 11(1-2), 121–138 (2005)
11. Izhikevich, E. M.: Simple model of spiking neurons. IEEE Trans. on Neural Networks 14(6), 1569–1572 (2003)
12. Izhikevich, E. M.: Which model to use for cortical spiking neurons? IEEE Trans. on Neural Networks 15(5), 1063–1070 (2004)
13. Izhikevich, E.M.: Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. The MIT press (2007)
14. Frias-Martinez, E. Gobet, F.: Automatic generation of cognitive theories using genetic programming. Minds and Machines 17(3), 287–309 (2007)
15. Hendrickson, E., et al.: Converting a globus pallidus neuron model from 585 to 6 compartments using an evolutionary algorithm. BMC Neurosci. 8(s2), P122 (2007)
16. Price, K., Storn, R. M., Lampinen, J. A.: Diffentential evolution: a practical ap-proach to global optimization. Springer (2005)
17. Murphy, P. M., Aha, D. W.: UCI repository of machine learning databases. Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, (1994)
18. Vazquez, R.A., Sossa, H.: A new associative model with dynamical synapses. Neural Processing Letters, 28(3), 189–207 (2008)
19. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. on SMC, 9(1), 62–66 (1979)
20. Jain, R. et al.: Machine Vision McGraw-Hill (1995)
21. Hu, M. K.: Visual pattern recognition by moment invariants. IRE Trans. on Information Theory 8, 179–187 (1962)
22. Vazquez, Roberto A., Cachon, Aleister: Integrate and fire neurons and their application in pattern recognition. Proceedings of the 7th International Conference on Electrical Engineering, Computing Science and Automatic Control, 424–428 (2010)