

Maximizing the Number of Polychronous Groups in Spiking Networks

Roberto Santana
University of the Basque
Country (UPV/EHU)
P. Manuel de Lardizabal 1,
Donostia. CP.20018, Spain
roberto.santana@ehu.es

Concha Bielza
DIA, Universidad Politécnica
de Madrid
Madrid, Spain
mcbielza@fi.upm.es

Pedro Larrañaga
DIA, Universidad Politécnica
de Madrid
Madrid, Spain
pedro.larranaga@fi.upm.es

ABSTRACT

In this paper we investigate the effect of biasing the axonal connection delay values in the number of polychronous groups produced for a spiking neuron network model. We use an estimation of distribution algorithm (EDA) that learns tree models to search for optimal delay configurations. Our results indicate that the introduced approach can be used to considerably increase the number of such groups.

Categories and Subject Descriptors

I.5.4 [Computing Methodologies]: Pattern Recognition—Neural nets; G.3 [Mathematics of Computing]: Probability and Statistics; I.2.8 [Computing Methodologies]: Artificial Intelligence—Problem Solving, Control Methods, and Search

General Terms

Theory

Keywords

polychronization, spiking neural network, estimation of distribution algorithm

1. SPIKING NETWORKS AND POLYCHRONIZATION

It has been proved that conduction delays play a role in the computational power and learnability of SNNs [3]. Spiking network polychronization refers to reproducible time-locked but not synchronous firing patterns with millisecond precision. It was originally studied by Izhikevich [1] in a model of cortical spiking neurons with axonal conduction delays and spike-timing plasticity.

In this paper we study the influence of conduction delays in the polychronization process and in particular, we investigate whether the conduction delays can be biased to maximize the number of coexisting polychronous groups (PNGs). To find networks with optimal conduction delays we use an estimation of distribution algorithm (EDA) [2], a class of evolutionary algorithms that use probability models instead of genetic operators.

The Izhikevich spiking neuron model [1] is a 2-dimensional system of ordinary differential equations. It is summarized in Figure 1¹. In the equations shown in Figure 1, variable v represents the membrane potential of the neuron and u represents a membrane recovery variable, which accounts for the activation of K^+ ionic currents and the inactivation of Na ionic currents, and provides negative feedback to v . Parameters a , b , c , and d serve to describe different characteristic features of the model.

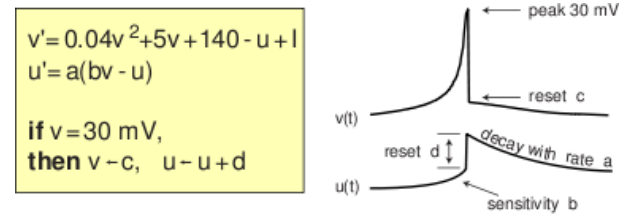


Figure 1: Izhikevich's spiking neuron model.

We use the spiking network model² introduced in [1]. It is a sparse network with 300 randomly connected spiking neurons, spike-timing dependent plasticity (STDP), and conduction delays. The network represents a cortical column or hypercolumn. It is composed of excitatory and inhibitory neurons, which are defined by assigning two different sets of values to the parameters of Izhikevich's spiking neuron model. All neurons in the network have parameters $(b, c) = (0.2, -65)$. Excitatory neurons are defined by parameters $(a, d) = (0.02, 8)$. Inhibitory neurons are defined by parameters $(a, d) = (0.1, 2)$.

The spiking network implementation sets a balance between the number of excitatory (80%) and inhibitory (20%) neurons which corresponds to ratios found in the mammalian cortex [1]. In our reduced implementation there are 240 excitatory and 60 inhibitory neurons. The network is sparse and each neuron is connected to 30 other neurons. The synaptic connections among neurons have fixed conduction delays, a delay value, dy , ranges from 1 to 20 ms. All inhibitory neurons have synaptic conductances with a homogeneous delay of $dy = 1$ ms. The delays of excitatory neurons range from

¹Electronic version of the figure and reproduction permissions are freely available at <http://www.izhikevich.com>

²This model is available from <http://senselab.med.yale.edu/modeldb/ShowModel.asp?model=115968>

1 ms to 20 ms. Of the total number of 30 neuron connections, each neuron has exactly 15 synapse with a delay value $dy, dy \in \{1, \dots, 20\}$. The simulation time was restricted to 5 hours, using a time resolution of 1 ms.

Whenever the neurons in a group do fire with the spike-timing pattern determined by the connectivity and delays, it is said that the group is activated and the corresponding neurons *polychronize* [1]. The activated groups may have different sizes, lengths, and time spans.

Following [1], after the simulation of a given SNN has been completed, the spikes trains are searched to detect persistent spike-timing patterns that emerge and reoccur with millisecond precision. The next step is to recognize all the groups of neurons that participate in each pattern. To detect PNGs, all groups are filtered and only those with a length greater than 40 and at least 6 neurons involved in the firing pattern are selected.

2. MODIFYING THE NETWORK DYNAMICS USING TREE-EDA

By varying the number of excitatory neurons with homogeneous conduction delays and analyzing the influence of this change on the number of detected PNGs we study the effect of conduction delays. In our representation, there are n_e variables, where n_e is the number of excitatory neurons. Since the inhibitory neurons retain their original homogeneous delays, they are not part of the optimization process. $x_i = 1, i \in \{1, \dots, n_e\}$ means that neuron i will have a homogeneous conduction delay value for all the axons. The selected delay value is $D = 1$ ms. $x_i = 0$ means that neuron i will keep the same axonal conduction delay assignment that in the reference network.

The objective or fitness function $f(\mathbf{x})$ assigned to a solution $\mathbf{x}$ is the number of PNGs found after finishing the simulation. We would optimize this function using and EDA.

Tree-EDA [4] uses a tree probabilistic model in which each variable may depend on at most another variable, which is called the parent. The algorithm uses a population size of 100 individuals and 85 generations. The population size and number of generations was considerably constrained by the cost of the optimization function that is very expensive. Truncation selection is used with truncation parameter $T = 0.5$. For the optimization experiments, 25 runs of the EDAs are executed.

We apply the optimization algorithm to modify the number of homogeneous delays. Figure 2 shows the average fitness of the new generated solutions at different generations of Tree-EDA. The average fitness steadily increases with generations. Figure 3 shows the relationship between the number of neurons with homogeneous delay values and the number of PNGs for random and optimized networks. It can be seen in figures 2 and 3 that the EDA consistently improves the number of PNGs.

3. CONCLUSIONS

Although it has been recognized that polychronization can be used as a tool for exploiting the computational power of synaptic delays and for monitoring the topology and activity of a spiking neuron network [3], more research is needed to determine the influence of the delays values in the number of PNGs and other features of the networks. In this paper we have shown that SNNs can be evolved to maximize the

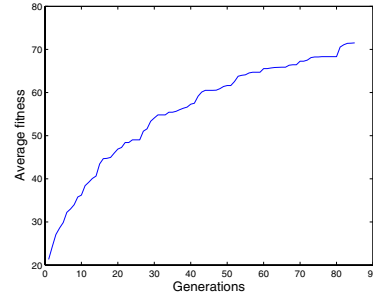


Figure 2: Average fitness of generated solutions.

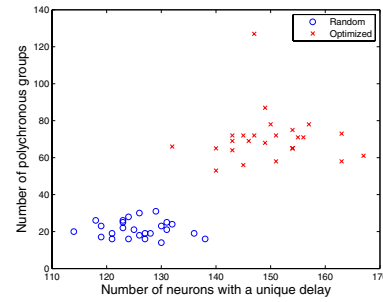


Figure 3: Number of neurons with homogeneous delay values against the number of PNGs for random and optimized networks.

number of PNGs. In future work it would be possible to investigate to what extent the structural information learned by the probabilistic model is able to capture details of the SNN connectivity patterns.

Acknowledgments

This work has been partially supported by the Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government), Cajal Blue Brain, TIN-2008-06815-C02-02, TIN2007-62626, TIN2010-14931 and Consolider Ingenio 2010 - CSD 2007 - 00018 projects (Spanish Ministry of Science and Innovation) and COMBIOMED network in computational biomedicine (Carlos III Health Institute).

4. REFERENCES

- [1] E. M. Izhikevich. Polychronization: Computation with spikes. *Neural Computation*, 18(2):245–282, 2006.
- [2] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.
- [3] H. Paugam-Moisy, R. Martinez, and S. Bengio. Delay learning and polychronization for reservoir computing. *Neurocomputing*, 71(7-9):1143–1158, 2008.
- [4] R. Santana, A. Ochoa, and M. R. Soto. The mixture of trees factorized distribution algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2001*, pages 543–550, San Francisco, CA, 2001. Morgan Kaufmann Publishers.