Biologically Inspired Evolutionary Temporal Neural Circuits

Reza Derakhshani
reza@csee.wvu.edu

Stephanie A.C. Schuckers, PhD
scaswell@wvu.edu

Lane Department of Computer Science and Electrical Engineering
West Virginia University
P.O.Box 6109
Morgantown, WV, 26506-6109

Abstract - The success of classical artificial neural networks as well as the wealth of new data about neuronal functions encourages researchers to explore new possibilities for the creation of biologically inspired artificial neural circuits. Temporal sequence is an important factor which is underrepresented in classical artificial neural networks. We developed a computational model of a neuronal circuit that incorporates more biological resemblance in form of a fully interconnected temporal network. The proposed function of the circuit is crucial in selecting the parameters which control each neuron and their final arrangement. However, the mathematics behind the dynamics of nonlinear, time sensitive, interconnected neural circuits and their learning processes as well as the choice of temporal neural code is daunting. To solve this problem, we use evolutionary optimization methods to find the appropriate configuration of simulated neural circuits. The results of a preliminary simulated neural circuit demonstrate that pattern recognition tasks can be performed with robustness to noisy or distorted patterns.

I. OVERVIEW

Expansion of the knowledge of the function of biological neurons has motivated researchers to attempt to develop brain-like computational schemes. The advances of digital and analog VLSI technology and availability of cheap, but powerful, computers has made simulation of vast and complex systems more feasible. Progress in computer science, especially emergence of new optimization and search techniques like evolutionary computing have created answers to previously intractable engineering problems. This is especially true since evolutionary programs and genetic algorithms can easily be parallelized on multi-processor systems or PC clusters. In spite of detailed single-cell descriptions of neurons, one major impediment for creation of functional circuits out of complex artificial neural ensembles has been their interconnection, parameter selection and training, but with the methods mentioned above, one can hope to solve the existing problems and achieve brain-like computing.

Real neurons are powerful computational devices since they are actually complicated analog processors. Besides linear operations, much has been learned about a vast number of nonlinear operations which neurons perform including low and band-pass filtering, normalization, gain control, saturation, amplification, multiplication, and thresholding [1]. In this study, we developed a computational model of a neuron that begins to incorporate these aspects.

Single neurons have limitations too. For example, computations that require more than two recursive nonlinear interactions, finding the maximum of scalar integers, or any operation requiring precision would be difficult for a single neuron. Instead, neurons form massively interconnected parallel circuits to perform given tasks [2]. The engineering challenge is discovering the architecture and learning rules of these interconnected networks of neurons. We use evolutionary computing techniques to adapt the network to perform a predefined function.

The temporal association of synaptic inputs activates cellular mechanisms that underlie such diverse brain processes as learning, memory and coincidence detection for sound localization. Temporal factors can be built into real neural assemblies through repeating units of cellular architecture, as are most easily recognized in cortical territories, and tapped delays via branches of axons traversing the entire structure [3,4,5,6]. Temporal parameters are under-employed in classical artificial networks, but are an integral feature of the circuit we developed.

II. THE CIRCUIT AND ITS PARAMETERS

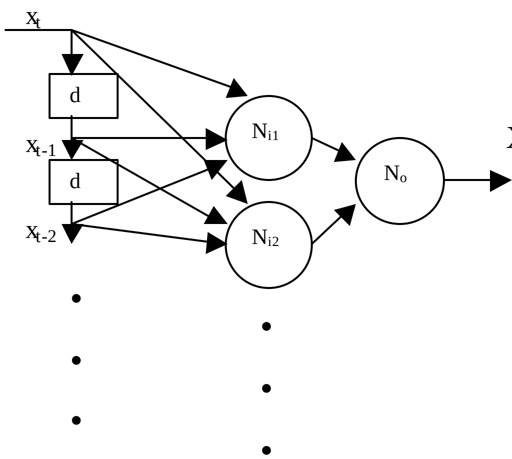
In this section development of a simple temporal architecture of a neural circuit is described. An important feature of our model is the adaptable temporal characteristics implemented mainly through path delays. The other temporal features are the cells' absolute and relative refractory periods. The latter has been modeled through an exponentially decreasing firing threshold.

0-7803-7278-6/02/$10.00 ©2002 IEEE

The main element in construction and training of our networks is an evolutionary algorithm which optimizes a series of real valued parameters which represent the network and its neurons. Exponential waveforms are used for action potentials because of their biological relevance and ease of analog circuit implementation in case of hardware implementation. Different aspects of our model are described in the next sections.

single neurons as well as their ensembles are crucial to both analysis and learning capabilities of neural circuits.

In our circuit, time delays are included between neurons which model path length or RC. Different time delays associated with each synapse model



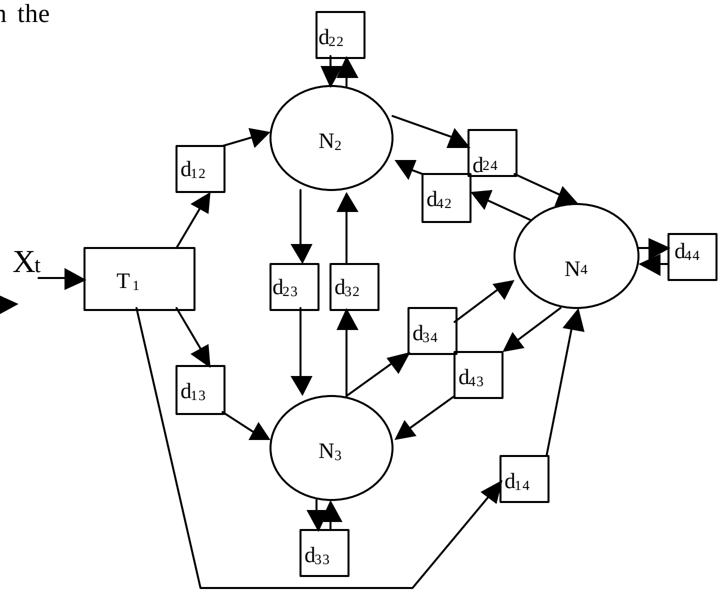


Figure 1. Comparison of classic neural networks inclusion of a time series (left) versus proposed neural circuit (right). While classical neural networks have a uniform tapped delay line, the path delays for each inter-neuron signal in the suggested model can be of any value. The transducer is represented by T, the neurons by N_i, and delays are represented by d_{ij}.

Transducer Neuron

It is considered that the afferent flow of the sensory (external world) data is inputted to a specialized neuron that acts as a transducer and thus converts the input signal level to appropriate instantaneous frequency. A similar type of neuron is included in our neural architecture that converts a time-series input to a train of action potentials. This cell dispatches the resulting spike train to the rest of the circuit. This cell does not receive feedback from other cells and is composed of one compartment without dendritic structure, as it does not take part in processing of the data. The transducer too is composed of a leaky integrator with absolute and relative refractory periods plus an exponential pre-amplification of the input signal.

Modeling of Time Delays

As mentioned, one of the main features that is ignored in most of existing artificial neural networks is the time factor and temporal coding. Temporal patterns of events in biological brains play a key role in perception of the external world and other processes [7,8,9]. Temporally sensitive functions of dendritic morphology. This has an important implication. Classic models store information in connection weights, but our proposed model learns in terms of, not only connection weights, but also different time delays, adding feedback between different layers as well as each neuron to itself (which also makes them capable of oscillation). The nearest analogy in dynamic systems would be a recurrent system described by n^{th} order difference equations, with n being proportional to the longest path delay in the system. This concept is different from existing recurrent neural networks such as Jordan and Elman networks [10] where only limited first order feedbacks are utilized. Moreover, the classical temporal neural networks' representation of time series are not realistic since they have the sampled signal moving vertically across the synapses instead of horizontal flow of each signal along each synaptic input, giving rise to problems such as determination of the temporal sample buffer size (see Figure 1).

Compartmental Model of Neuron

A compartmentalized model of a neuron is used. As has been suggested in various literatures

[11,12,13,14,15,16,17], the dendritic tree is divided into thresholding, neuron-like sub-trees, with up to a 100 sub-trees per each neuron. Their outputs are summed up in the main trunks and converge into the soma, where the soma is simply another similar compartment. Therefore each neuron is like a two-layer feed-forward network, with each sub-tree acting as a hidden node. These compartments have both absolute and relative refractory periods. In this study, we have only a single generic excitable dendritic sub-tree for each node. More complex models will be explored in future.

Action potentials and their rate

A useful but simple way to convert an accumulated level in a compartment is the use of a leaky integrate-and-fire model. This model not only does the job of level-to-frequency conversion but also helps in coincidence detection. The leaky integrator fires action potentials when it is charged to its threshold. The initial values for these integrators were taken from auditory circuit neurons and later subjected to Gaussian mutation for evolution. Action potentials are composed of an exponential attack and an exponential decay phase, modeling the RC circuit of the membrane. The initial excitatory postsynaptic potential (ePSP) is composed of a fast exponential attack and a slower exponential decay. The inhibitory postsynaptic potential (iPSP) is the same but with a negative. It is interesting to note that dendritic sub-trees can use PSPs as building blocks and linearly combine them to get different waveforms, as in the auditory system, and the resulting sharper waveforms will yield better temporal resolution [18]. There is biological evidence (e.g. in cortical neurons) supporting such basis function approach to sensorimotor transformations [19].

The Connection Weight Matrix W_{NxN}

W is an NxN matrix where N is the number of compartments in the circuit, with each member w_{ij} defining the connection weight between the presynaptic neuron/dendritic tree i to its target synapse/neuron j. W_{NxN} constitutes an important part of learning by memorizing the optimized connection weights, similar to those of classical neural networks.

The Delay (or path length) Matrix D_{NxN}

D_{NxN} is composed of elements d_{ij} which represent the path delays between the presynaptic neuron/dendritic tree i to target synapse/neuron j. This matrix memorizes the important temporal configuration learned through evolutionary optimization. N is the number of compartments in the circuit.

III. EVOLUTION PROCEDURE

The last section discussed the details of the model for the neural circuit. Evolution is used to adapt the parameters in circuits' chromosome such that the network can perform a chosen task. A fitness function determines the success of the circuit. During the evolution cycle, first the initial population is generated. Each representation is then converted to the corresponding architecture. The chromosome for each circuit is composed of the delay and weight matrices, D_{NxN} and W_{NxN}. Search operators are applied in order to produce offspring, and the fittest from both parents and offspring are chosen for next cycle. Since an evolutionary search is performed on a large population and enjoys the element of randomness to some extent, unlike gradient descent, there is less probability of being trapped in local minima.

Evolutionary Procedure Details

Mutation: Mutation will be applied through Gaussian deviations so the offspring would generally follow their parents. During initial genesis of each network, the weight and delay matrices generated by picking their values randomly but within biologically plausible ranges.

No crossover was used since the knowledge is spread across the network, and its application to weight or delay matrices would destroy the accumulated knowledge in those formations.

Fitness: Number of offspring for each circuit is determined by its fitness, which is defined as its degree of success in achieving the defined goal. To quantify this, the algorithm observes the emergence of a neural code (the way neurons create or interpret spiking patterns), which is predefined for recognition of specific input pattern. In this study, the code is simply firing rate maximized for one input waveform and minimized for the other one. This ratio of max-to-min firing rates will constitute the fitness of an individual network. A cell must be selected as the output or "grandmother" cell. Because of the symmetry of fully interconnected network, we do not need to assume one initially. Instead the program measures the triangle to square wave output frequency ratio for each cell and chooses the highest as that network's fitness.

0-7803-7278-6/02/$10.00 ©2002 IEEE

Training and Simulation

Over-fitting can occur if the number of examples is high and their variance low, since the network might precisely lock into the training pattern. Training sets used in this research are in the form of time series representing triangle and square waveforms. Invariance is tested by compressing or stretching the input waveforms in time by a non-rational factor. Robustness is tested by inclusion of noise with different characteristics.

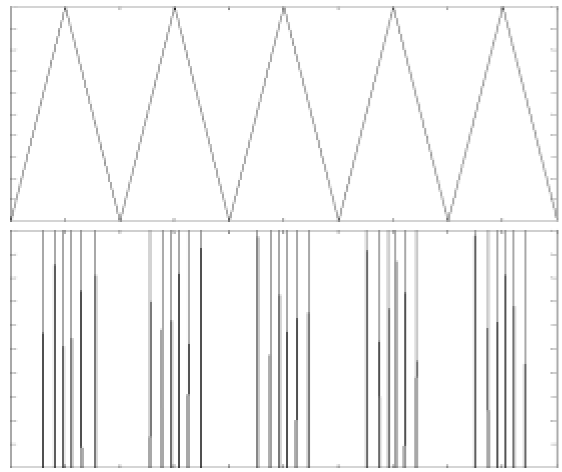
In order to determine the level of discrimination the circuit has achieved in training, the algorithm dynamically chooses an output cell with the highest occurrence of the predefined neural code, here maximum spike frequency. This offers a practical solution to neural code and the so-called "grand mother cell" problems. In other words, all the neurons have the potential to be output cells.

IV. SIMULATION RESULTS

For preliminary work, a small population of ten simulated neural circuits was evolved, each consisting of six fully connected neurons. Zero-mean Gaussian mutations with a fixed standard deviation of 10ms are used for all elements in the delay matrix. The same distribution but with a standard deviation of 10^{-9} (about 10% of initial values) is used for mutations of the weight matrix. The code was written in Matlab 5 and executed on a single processor, so the size of the network was kept small to achieve higher speed. The network was evolved for five generations, with each test simulating a time equal to 100 ms.

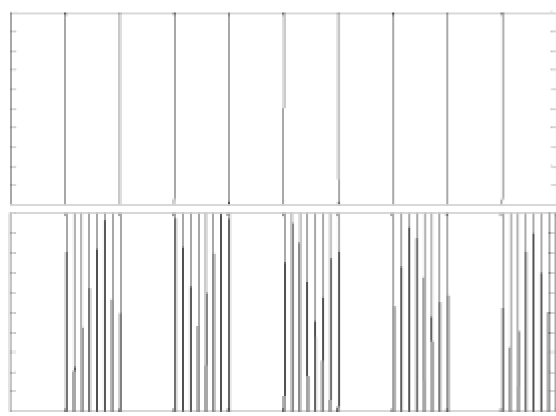
Integrate and fire compartments have an RC of 10ms, absolute refractory period of 1ms, and relative refractory period of 2ms. One cell is designated as a transducer, receiving a waveform from outside, with an exponential input amplification [20]. Triangle and square waves were used both in training (evolving) and testing the circuit. Figure 2 shows the triangular input waveform and the resulting train of action potentials at the transducer's output. Figure 3 shows the same for a square wave. The circuits then undergo an evolution process to find the best (fittest) circuit which distinguishes a triangle from a square wave. Figure 4 shows the output cell's spiking activity for a square and triangle wave for the circuit with highest fitness. Note that even though a square wave produces more spikes at the transducer cell, the circuit has successfully blocked it out at the output cell. It appears that the ensemble is detecting the temporally different patterns of triangle spikes injected by the transducer cell. Only the last half of the output (the last 500 ms) was used in fitness evaluations in order to let the circuit reach steady state.

After evolutionary training, the circuit was subjected to 4 test cases. The first two were triangle and square waves with added low passed Gaussian noise with standard deviations of 0.1 and 0.25. As it can be seen from Figure 5, the network robustly distinguishes a triangle from a square wave with additive noise with standard deviation of 0.10, shown by firing at the output neuron. Similar results were seen for the second case (Gaussian noise, SD=0.25).



Time (0-100ms)

Figure 2. Input waveform, a 50 Hz triangle wave, (top) and output after transducer (bottom).



Time (0-100ms)

Figure 3. Input waveform, a 50 Hz square wave, (top) and output after transducer (bottom).

Similarly, this output pattern was not affected by time distortion where each waveform was expanded by square root of 2, a non-rational number. The same results were seen when the network was exposed to waveforms compressed by factor of square root of two.

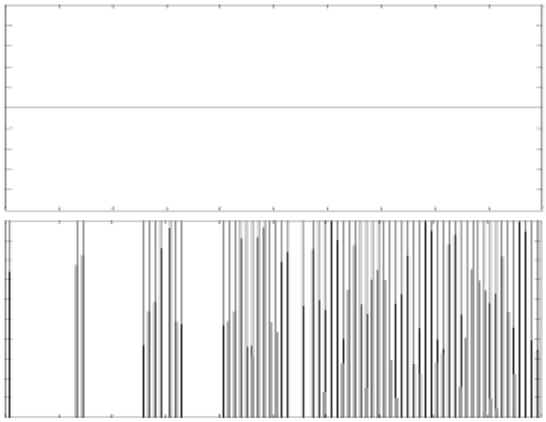


Figure 4. Output for square wave (top), and output of triangle wave (botom), after evolution.

Notice that this robust recognition capability has been achieved through just five generations of the evolutionary algorithm, and holds promise for more difficult pattern recognition problems. Repeated runs created comparable performance.

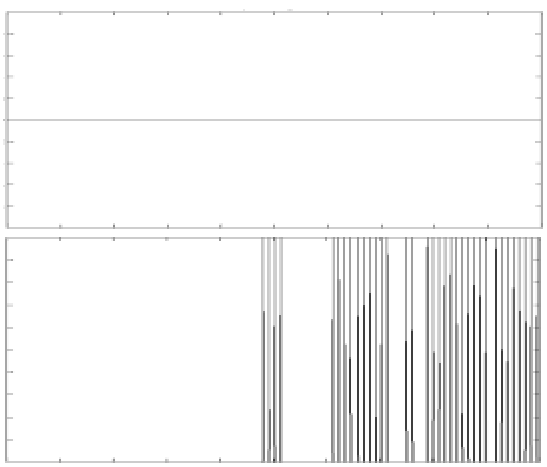


Figure 5. Related output for a square wave plus Gaussian noise with SD 0.1 (top) and triangle with the same noise (bottom).

V. CONCLUSION AND FUTURE WORK

Biologically inspired neural networks have the potential to address different pattern recognition problems and possibly achieve some sort of intelligence, through a network of neurons that follow biological models. These time-sensitive neural circuits can be adapted through the use of evolutionary techniques in order to overcome unsolved problems like creation of functional and neural codes. The results of our preliminary work as well as biological evidence of comparable phenomena show a promising future for this approach.

Future work will include augmenting the neuron model by inclusion of short-range sub-threshold interaction of PSPs in dendritic trees, spike-timing dependent synaptic plasticity, synaptic redistribution and scaling, and improving the evolution process by dividing the circuit parameters into local and global sub-groups.

References

[1] Koch, C.: Biophysics of Computation, Oxford Univ. Press, New York, 1998.

[2] Koch, C. and Segev, I., “The Role of Single Neurons in information processing,” Nature Neuroscience Supplement, Vol. 3, pp. 1171-1177, November 2000.

[3] Medina, J. and Mauk, M., “Computer simulation of cerebellar information processing,” Nature Neuroscience Supplement, Vol. 3, pp. 1205-1211, November 2000.

[4] Voogd J, Glickstein M, “The anatomy of the cerebellum,” Trends Neurosci 21:370– 375, 1998.

[5] Eccles, J. C., and Ito, M. & Szentágothai, J.: The Cerebellum as a Neuronal Machine, Springer, 1967.

[6] Ito, M.: The Cerebellum and Neural Control, Raven, New York, 1984.

[7] Carr, C.E., and Friedman, M.A.: Evolution of Time Coding Systems, Neural Computation, Vol. 11, 1-20, MIT press, 1999.

[8] Rao, S. M. et al, “The Evolution of Brain Activation During Temporal Processing,” Nature Neuroscience, Vol.4, No.3, pp 317-323, March 2001.

[9] Zacks J. M. et al , “Human Brain Activity Time-Locked to Perceptual Event Boundaries,” Nature Neuroscience, , Vol.4, No.6, pp 651-655, June 2001.

[10] Principe JC, Euliano NR, Lefebvre WC, Neural and Adaptive Systems, John Wily, New York, 2000.

[11] Koch, C., Poggio, T., and Torre, V., “Retinal ganglion cells: A functional interpretation of dendritic morphology,” Phil. Trans. R. Soc. Lond. B, 298, 227-264, 1982.

[12] Shepherd, G., Brayton, R., Miller, J., Segev, I., Rinzel, J., and Rall, W., “Signal enhancement in distal cortical dendrites by means of interactions between active dendritic spines,”Proc. Natl. Acad. Sci. USA, 82, 2192-2195, 1985.

[13] Rall, W., and Segev, I., “Functional possibilities for synapses on dendrites and on dendritic spines,” In Edelman, G., Gall, W., & Cowan, W. (Eds.): Synaptic Function, pp. 605-636.Wiley, New York, 1987.

[14] Mel B. W., “Synaptic integration in an excitable dendritic tree.” J Neurophys 70:1086-1101, 1993.

[15] Mel, B. W., “NMDA-based pattern discrimination in a modeled cortical neuron,” Neural Comp., 4, 502-516, 1992.

[16] Mel, B. W., “The clusteron: Toward a simple abstraction for a complex neuron,” In Moody, J., Hanson, S., and Lippmann, R. (Eds.): Advances in Neural Information Processing Systems, vol. 4, pp. 35-42. Morgan Kaufmann, San Mateo, CA, 1992.

[17] Mel B.W., Ruderman D.L., and Archie K.A., “Translation-invariant orientation tuning in visual `complex' cells could derive from intradendritic computations,” J Neurosci, 17:4325-4334, 1998.

[18] Spirou, G., West Virginia University, Personal Communications, 2001.

[19] Pouget, A. and Snyder, L., “Computational approaches to sensorimotor transformations,” Nature Neuroscience Supplement, Vol. 3, November, pp. 1192-1198, 2000.

[20] Mead C.: Analog VLSI and Neural Systems, Reading, MA, Addison-Wesley, 1989.