Reservoir-Based Evolving Spiking Neural Network for Spatio-temporal Pattern Recognition

Stefan Schliebs[1], Haza Nuzly Abdull Hamed[1,2], and Nikola Kasabov[1,3]

[1] KEDRI, Auckland University of Technology, New Zealand
{sschlieb,hnuzly,nkasabov}@aut.ac.nz
www.kedri.info
[2] Soft Computing Research Group, Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Johor, Malaysia
haza@utm.my
[3] Institute for Neuroinformatics, ETH and University of Zurich, Switzerland

Abstract. Evolving spiking neural networks (eSNN) are computational models that are trained in an one-pass mode from streams of data. They evolve their structure and functionality from incoming data. The paper presents an extension of eSNN called reservoir-based eSNN (reSNN) that allows efficient processing of spatio-temporal data. By classifying the response of a recurrent spiking neural network that is stimulated by a spatio-temporal input signal, the eSNN acts as a readout function for a Liquid State Machine. The classification characteristics of the extended eSNN are illustrated and investigated using the LIBRAS sign language dataset. The paper provides some practical guidelines for configuring the proposed model and shows a competitive classification performance in the obtained experimental results.

Keywords: Spiking Neural Networks, Evolving Systems, Spatio-Temporal Patterns.

1 Introduction

The desire to better understand the remarkable information processing capabilities of the mammalian brain has led to the development of more complex and biologically plausible connectionist models, namely spiking neural networks (SNN). See [3] for a comprehensive standard text on the material. These models use trains of spikes as internal information representation rather than continuous variables. Nowadays, many studies attempt to use SNN for practical applications, some of them demonstrating very promising results in solving complex real world problems.

An evolving spiking neural network (eSNN) architecture was proposed in [18]. The eSNN belongs to the family of Evolving Connectionist Systems (ECoS), which was first introduced in [9]. ECoS based methods represent a class of constructive ANN algorithms that modify both the structure and connection weights of the network as part of the training process. Due to the evolving nature of the network and the employed fast one-pass learning algorithm, the method is able to accumulate information as it becomes available, without the requirement of retraining the network with previously

B.-L. Lu, L. Zhang, and J. Kwok (Eds.): ICONIP 2011, Part II, LNCS 7063, pp. 160–168, 2011.

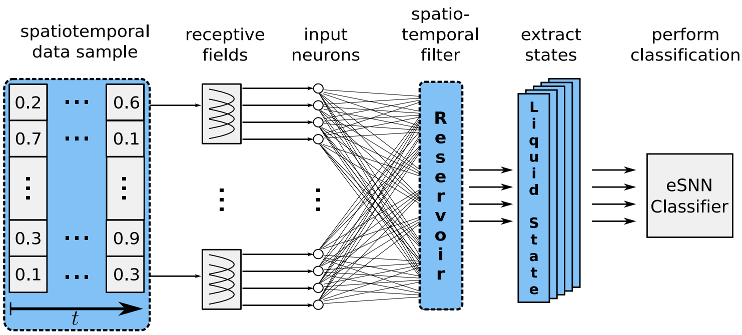


Fig. 1. Architecture of the extended eSNN capable of processing spatio-temporal data. The colored (dashed) boxes indicate novel parts in the original eSNN architecture.

presented data. The review in [17] summarises the latest developments on ECoS related research; we refer to [13] for a comprehensive discussion of the eSNN classification method.

The eSNN classifier learns the mapping from a single data vector to a specified class label. It is mainly suitable for the classification of time-invariant data. However, many data volumes are continuously updated adding an additional time dimension to the data sets. In [14], the authors outlined an extension of eSNN to reSNN which principally enables the method to process spatio-temporal information. Following the principle of a Liquid State Machine (LSM) [10], the extension includes an additional layer into the network architecture, *i.e.* a recurrent SNN acting as a reservoir. The reservoir transforms a spatio-temporal input pattern into a single high-dimensional network state which in turn can be mapped into a desired class label by the one-pass learning algorithm of eSNN.

In this paper, the reSNN extension presented in [14] is implemented and its suitability as a classification method is analyzed in computer simulations. We use a well-known real-world data set, *i.e.* the LIBRAS sign language data set [2], in order to allow an independent comparison with related techniques. The goal of the study is to gain some general insights into the working of the reservoir based eSNN classification and to deliver a proof of concept of its feasibility.

2 Spatio-temporal Pattern Recognition with reSNN

The reSNN classification method is built upon a simplified integrate-and-fire neural model first introduced in [16] that mimics the information processing of the human eye. We refer to [13] for a comprehensive description and analysis of the method. The proposed reSNN is illustrated in Figure 1. The novel parts in the architecture are indicated by the highlighted boxes. We outline the working of the method by explaining the diagram from left to right.

Spatio-temporal data patterns are presented to the reSNN system in form of an ordered sequence of real-valued data vectors. In the first step, each real-value of a data vector is transformed into a spike train using a population encoding. This encoding distributes a single input value to multiple neurons. Our implementation is based on arrays of receptive fields as described in [1]. Receptive fields allow the encoding of continuous values by using a collection of neurons with overlapping sensitivity profiles.

As a result of the encoding, input neurons spike at predefined times according to the presented data vectors. The input spike trains are then fed into a spatio-temporal filter which accumulates the temporal information of all input signals into a single high-dimensional intermediate liquid state. The filter is implemented in form of a liquid or a reservoir [10], *i.e.* a recurrent SNN, for which the eSNN acts as a readout function. The one-pass learning algorithm of eSNN is able to learn the mapping of the liquid state into a desired class label. The learning process successively creates a repository of trained output neurons during the presentation of training samples. For each training sample a new neuron is trained and then compared to the ones already stored in the repository of the same class. If a trained neuron is considered to be too similar (in terms of its weight vector) to the ones in the repository (according to a specified similarity threshold), the neuron will be merged with the most similar one. Otherwise the trained neuron is added to the repository as a new output neuron for this class. The merging is implemented as the (running) average of the connection weights, and the (running) average of the two firing threshold. Because of the incremental evolution of output neurons, it is possible to accumulate information and knowledge as they become available from the input data stream. Hence a trained network is able to learn new data and new classes without the need of re-training already learned samples. We refer to [13] for a more detailed description of the employed learning in eSNN.

2.1 Reservoir

The reservoir is constructed of Leaky Integrate-and-Fire (LIF) neurons with exponential synaptic currents. This neural model is based on the idea of an electrical circuit containing a capacitor with capacitance C and a resistor with a resistance R, where both C and R are assumed to be constant. The dynamics of a neuron i are then described by the following differential equations:

$$\tau_m \frac{du_i}{dt} = -u_i(t) + R\, I_i^{\mathrm{syn}}(t) \tag{1}$$

$$\tau_s \frac{dI_i^{\mathrm{syn}}}{dt} = -I_i^{\mathrm{syn}}(t) \tag{2}$$

The constant $\tau_m = RC$ is called the membrane time constant of the neuron. Whenever the membrane potential u_i crosses a threshold ϑ from below, the neuron fires a spike and its potential is reset to a reset potential u_r. We use an exponential synaptic current I_i^{syn} for a neuron i modeled by Eq. 2 with τ_s being a synaptic time constant.

In our experiments we construct a liquid having a small-world inter-connectivity pattern as described in [10]. A recurrent SNN is generated by aligning 100 neurons in a three-dimensional grid of size $4 \times 5 \times 5$. Two neurons A and B in this grid are connected with a connection probability

$$P(A, B) = C \times e^{\frac{-d(A,B)}{\lambda^2}} \tag{3}$$

where $d(A, B)$ denotes the Euclidean distance between two neurons and λ corresponds to the density of connections which was set to $\lambda = 2$ in all simulations. Parameter C depends on the type of the neurons. We discriminate into excitatory (ex) and inhibitory (inh) neurons resulting in the following parameters for C: $C_{ex-ex} = 0.3$, $C_{ex-inh} = 0.2$, $C_{inh-ex} = 0.5$ and $C_{inh-inh} = 0.1$. The network contained 80% excitatory and 20% inhibitory neurons. The connections weights were randomly selected by a uniform distribution and scaled in the interval $[-8, 8]$nA. The neural parameters were set to $\tau_m = 30$ms, $\tau_s = 10$ms, $\vartheta = 5$mV, $u_r = 0$mV. Furthermore, a refractory period of 5ms and a synaptic transmission delay of 1ms was used.

Using this configuration, the recorded liquid states did not exhibit the undesired behavior of over-stratification and pathological synchrony – effects that are common for randomly generated liquids [11]. For the simulation of the reservoir we used the SNN simulator Brian [4].

3 Experiments

In order to investigate the suitability of the reservoir based eSNN classification method, we have studied its behavior on a spatio-temporal real-world data set. In the next sections, we present the LIBRAS sign-language data, explain the experimental setup and discuss the obtained results.

3.1 Data Set

LIBRAS is the acronym for **LI**ngua **BRA**sileira de **S**inais, which is the official Brazilian sign language. There are 15 hand movements (signs) in the dataset to be learned and classified. The movements are obtained from recorded video of four different people performing the movements in two sessions. In total 360 videos have been recorded, each video showing one movement lasting for about seven seconds. From the videos 45 frames uniformly distributed over the seven seconds have then been extracted. In each frame, the centroid pixels of the hand are used to determine the movement. All samples have been organized in ten sub-datasets, each representing a different classification scenario. More comprehensive details about the dataset can be found in [2]. The data can be obtained from the UCI machine learning repository.

In our experiment, we used Dataset 10 which contains the hand movements recorded from three different people. This dataset is balanced consisting of 270 videos with 18 samples for each of the 15 classes. An illustration of the dataset is given in Figure 2. The diagrams show a single sample of each class.

3.2 Setup

As described in Section 2, a population encoding has been applied to transform the data into spike trains. This method is characterized by the number of receptive fields used for the encoding along with the width β of the Gaussian receptive fields. After some initial experiments, we decided to use 30 receptive fields and a width of $\beta = 1.5$. More details of the method can be found in [1].

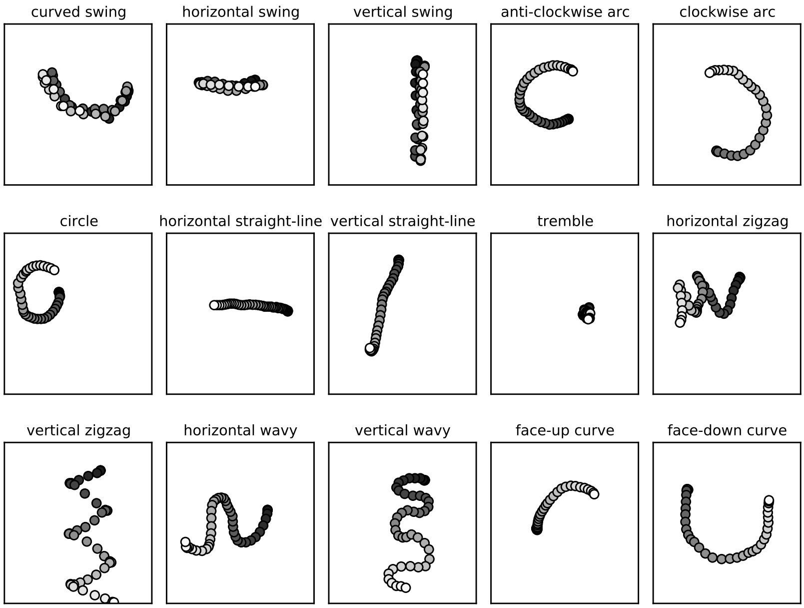


Fig. 2. The LIBRAS data set. A single sample for each of the 15 classes is shown, the color indicating the time frame of a given data point (black/white corresponds to earlier/later time points).

In order to perform a classification of the input sample, the state of the liquid at a given time t has to be read out from the reservoir. The way how such a liquid state is defined is critical for the working of the method. We investigate in this study three different types of readouts. We call the first type a cluster readout. The neurons in the reservoir are first grouped into clusters and then the population activity of the neurons belonging to the same cluster is determined. The population activity was defined in [3] and is the ratio of neurons being active in a given time interval $[t - \Delta_c t, t]$. Initial experiments suggested to use 25 clusters collected in a time window of $\Delta_c t = 10$ms. Since our reservoir contains 100 neurons simulated over a time period of $T = 300$ms, $T/\Delta_c t = 30$ readouts for a specific input data sample can be extracted, each of them corresponding to a single vector with 25 continuous elements. Similar readouts have also been employed in related studies [12].

The second readout is principally very similar to the first one. In the interval $[t - \Delta_f t, t]$ we determine the firing frequency of all neurons in the reservoir. According to our reservoir setup, this frequency readout produces a single vector with 100 continuous elements. We used a time window of $\Delta_f t = 30$ resulting in the extraction of $T/\Delta_f t = 10$ readouts for a specific input data sample.

Finally, in the analog readout, every spike is convolved by a kernel function that transforms the spike train of each neuron in the reservoir into a continuous analog signal. Many possibilities for such a kernel function exist, such as Gaussian and exponential kernels. In this study, we use the alpha kernel $\alpha(t) = e\ \tau^{-1}\ t\ e^{-t/\tau}\Theta(t)$ where $\Theta(t)$ refers to the Heaviside function and parameter $\tau = 10$ms is a time constant. The

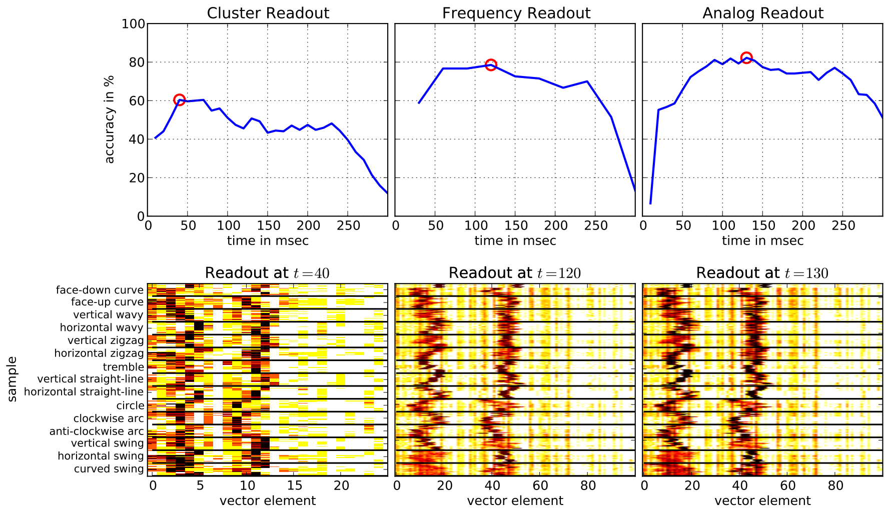


Fig. 3. Classification accuracy of eSNN for three readouts extracted at different times during the simulation of the reservoir (top row of diagrams). The best accuracy obtained is marked with a small (red) circle. For the marked time points, the readout of all 270 samples of the data are shown (bottom row).

convolved spike trains are then sampled using a time step of $\Delta_a t = 10$ms resulting in 100 time series – one for each neuron in the reservoir. In these series, the data points at time t represent the readout for the presented input sample. A very similar readout was used in [15] for a speech recognition problem. Due to the sampling interval Δ_a, $T/\Delta_a t = 30$ different readouts for a specific input data sample can be extracted during the simulation of the reservoir.

All readouts extracted at a given time have been fed to the standard eSNN for classification. Based on preliminary experiments, some initial eSNN parameters were chosen. We set the modulation factor $m = 0.99$, the proportion factor $c = 0.46$ and the similarity threshold $s = 0.01$. Using this setup we classified the extracted liquid states over all possible readout times.

3.3 Results

The evolution of the accuracy over time for each of the three readout methods is presented in Figure 3. Clearly, the cluster readout is the least suitable readout among the tested ones. The best accuracy found is 60.37% for the readout extracted at time 40ms, *cf.* the marked time point in the upper left diagram of the figure[1]. The readouts extracted at time 40ms are presented in the lower left diagram. A row in this diagram is the readout vector of one of the 270 samples, the color indicating the real value of the elements in that vector. The samples are ordered to allow a visual discrimination of the 15 classes. The first 18 rows belong to class 1 (curved swing), the next 18 rows to

[1] We note that the average accuracy of a random classifier is around $\frac{1}{15} \approx 6.67\%$.

class 2 (horizontal swing) and so on. Given the extracted readout vector, it is possible to even visually distinguish between certain classes of samples. However, there are also significant similarities between classes of readout vectors visible which clearly have a negative impact on the classification accuracy.

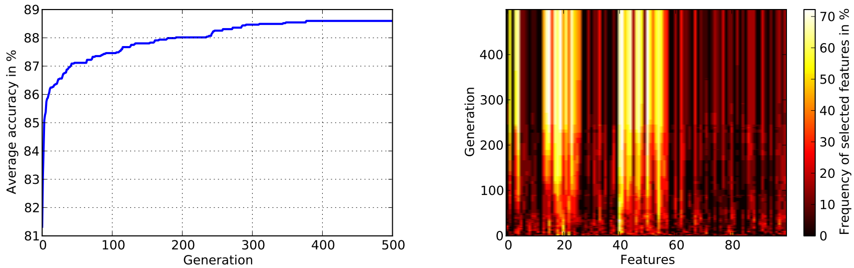
The situation improves when the frequency readout is used resulting in a maximum classification accuracy of 78.51% for the readout vector extracted at time 120ms, *cf.* middle top diagram in Figure 3. We also note the visibly better discrimination ability of the classes of readout vectors in the middle lower diagram: The intra-class distance between samples belonging to the same class is small, but inter-class distance between samples of other classes is large. However, the best accuracy was achieved using the analog readout extracted at time 130ms (right diagrams in Figure 3). Patterns of different classes are clearly distinguishable in the readout vectors resulting in a good classification accuracy of 82.22%.

3.4 Parameter and Feature Optimization of reSNN

The previous section already demonstrated that many parameters of the reSNN need to be optimized in order to achieve satisfactory results (the results shown in Figure 3 are as good as the suitability of the chosen parameters is). Here, in order to further improve the classification accuracy of the analog readout vector classification, we have optimized the parameters of the eSNN classifier along with the input features (the vector elements that represent the state of the reservoir) using the Dynamic Quantum inspired Particle swarm optimization (DQiPSO) [5]. The readout vectors are extracted at time 130ms, since this time point has reported the most promising classification accuracy. For the DQiPSO, 20 particles were used, consisting of eight update, three filter, three random, three embed-in and three embed-out particles. Parameter c_1 and c_2 which control the exploration corresponding to the global best (gbest) and the personal best (pbest) respectively, were both set to 0.05. The inertia weight was set to $w = 2$. See [5] for further details on these parameters and the working of DQiPSO. We used 18-fold cross validations and results were averaged in 500 iterations in order to estimate the classification accuracy of the model.

The evolution of the accuracy obtained from the global best particle during the PSO optimization process is presented in Figure 4a. The optimization clearly improves the classification abilities of eSNN. After the DQiPSO optimization an accuracy of 88.59% $(\pm 2.34\%)$ is achieved. In comparison to our previous experiments [6] on that dataset, the time delay eSNN performs very similarly reporting an accuracy of 88.15% $(\pm 6.26\%)$. The test accuracy of an MLP under the same conditions of training and testing was found to be 82.96% $(\pm 5.39\%)$.

Figure 4b presents the evolution of the selected features during the optimization process. The color of a point in this diagram reflects how often a specific feature was selected at a certain generation. The lighter the color the more often the corresponding feature was selected at the given generation. It can clearly be seen that a large number of features have been discarded during the evolutionary process. The pattern of relevant features matches the elements of the readout vector having larger values, *cf.* the dark points in Figure 3 and compare to the selected features in Figure 4.



(a) Evolution of classification accuracy

(b) Evolution of feature subsets

Fig. 4. Evolution of the accuracy and the feature subsets based on the global best solution during the optimization with DQiPSO

4 Conclusion and Future Directions

This study has proposed an extension of the eSNN architecture, called reSNN, that enables the method to process spatio-temporal data. Using a reservoir computing approach, a spatio-temporal signal is projected into a single high-dimensional network state that can be learned by the eSNN training algorithm. We conclude from the experimental analysis that the suitable setup of the reservoir is not an easy task and future studies should identify ways to automate or simplify that procedure. However, once the reservoir is configured properly, the eSNN is shown to be an efficient classifier of the liquid states extracted from the reservoir. Satisfying classification results could be achieved that compare well with related machine learning techniques applied to the same data set in previous studies. Future directions include the development of new learning algorithms for the reservoir of the reSNN and the application of the method on other spatio-temporal real-world problems such as video or audio pattern recognition tasks. Furthermore, we intend to develop a implementation on specialised SNN hardware [7,8] to allow the classification of spatio-temporal data streams in real time.

Acknowledgements. The work on this paper has been supported by the Knowledge Engineering and Discovery Research Institute (KEDRI, www.kedri.info). One of the authors, NK, has been supported by a Marie Curie International Incoming Fellowship with the FP7 European Framework Programme under the project "EvoSpike", hosted by the Neuromorphic Cognitive Systems Group of the Institute for Neuroinformatics of the ETH and the University of Zürich.

References

1. Bohte, S.M., Kok, J.N., Poutré, J.A.L.: Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing 48(1-4), 17–37 (2002)
2. Dias, D., Madeo, R., Rocha, T., Biscaro, H., Peres, S.: Hand movement recognition for brazilian sign language: A study using distance-based neural networks. In: International Joint Conference on Neural Networks IJCNN 2009, pp. 697–704 (2009)

3. Gerstner, W., Kistler, W.M.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge (2002)
4. Goodman, D., Brette, R.: Brian: a simulator for spiking neural networks in python. BMC Neuroscience 9(Suppl 1), 92 (2008)
5. Hamed, H., Kasabov, N., Shamsuddin, S.: Probabilistic evolving spiking neural network optimization using dynamic quantum-inspired particle swarm optimization. Australian Journal of Intelligent Information Processing Systems 11(01), 23–28 (2010)
6. Hamed, H., Kasabov, N., Shamsuddin, S., Widiputra, H., Dhoble, K.: An extended evolving spiking neural network model for spatio-temporal pattern classification. In: 2011 International Joint Conference on Neural Networks, pp. 2653–2656 (2011)
7. Indiveri, G., Chicca, E., Douglas, R.: Artificial cognitive systems: From VLSI networks of spiking neurons to neuromorphic cognition. Cognitive Computation 1, 119–127 (2009)
8. Indiveri, G., Stefanini, F., Chicca, E.: Spike-based learning with a generalized integrate and fire silicon neuron. In: International Symposium on Circuits and Systems, ISCAS 2010, pp. 1951–1954. IEEE (2010)
9. Kasabov, N.: The ECOS framework and the ECO learning method for evolving connectionist systems. JACIII 2(6), 195–202 (1998)
10. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation 14(11), 2531–2560 (2002)
11. Norton, D., Ventura, D.: Preparing more effective liquid state machines using hebbian learning. In: International Joint Conference on Neural Networks, IJCNN 2006, pp. 4243–4248. IEEE, Vancouver (2006)
12. Norton, D., Ventura, D.: Improving liquid state machines through iterative refinement of the reservoir. Neurocomputing 73(16-18), 2893–2904 (2010)
13. Schliebs, S., Defoin-Platel, M., Worner, S., Kasabov, N.: Integrated feature and parameter optimization for an evolving spiking neural network: Exploring heterogeneous probabilistic models. Neural Networks 22(5-6), 623–632 (2009)
14. Schliebs, S., Nuntalid, N., Kasabov, N.: Towards Spatio-Temporal Pattern Recognition Using Evolving Spiking Neural Networks. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010, Part I. LNCS, vol. 6443, pp. 163–170. Springer, Heidelberg (2010)
15. Schrauwen, B., D'Haene, M., Verstraeten, D., Campenhout, J.V.: Compact hardware liquid state machines on fpga for real-time speech recognition. Neural Networks 21(2-3), 511–523 (2008)
16. Thorpe, S.J.: How can the human visual system process a natural scene in under 150ms? On the role of asynchronous spike propagation. In: ESANN. D-Facto public (1997)
17. Watts, M.: A decade of Kasabov's evolving connectionist systems: A review. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 39(3), 253–269 (2009)
18. Wysoski, S.G., Benuskova, L., Kasabov, N.K.: Adaptive Learning Procedure for a Network of Spiking Neurons and Visual Pattern Recognition. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 1133–1142. Springer, Heidelberg (2006)