

# Evolving spiking neural networks for temporal pattern recognition in the presence of noise

Ahmed Abdelmotaleb<sup>1,2</sup>, Neil Davey<sup>1</sup>, Maria Schilstra<sup>1</sup>, Volker Steuber<sup>1</sup>, Borys Wróbel<sup>2,3</sup>

<sup>1</sup>Biocomputation Research Group, University of Hertfordshire, Hatfield, Herts AL10 9AB, UK

<sup>2</sup>Evolutionary Systems Laboratory, Faculty of Biology, Adam Mickiewicz University, Poznań, 61-712, Poland

<sup>3</sup>Systems Modeling Laboratory, IO PAS, Sopot, 81-712, Poland

a.m.abdelmotaleb@herts.ac.uk, n.davey@herts.ac.uk, m.j.l.schilstra@herts.ac.uk, v.steuber@herts.ac.uk, wrobel@evosys.org

## Abstract

Nervous systems of biological organisms use temporal patterns of spikes to encode sensory input, but the mechanisms that underlie the recognition of such patterns are unclear. In the present work, we explore how networks of spiking neurons can be evolved to recognize temporal input patterns without being able to adjust signal conduction delays. We evolve the networks with GReaNs, an artificial life platform that encodes the topology of the network (and the weights of connections) in a fashion inspired by the encoding of gene regulatory networks in biological genomes. The number of computational nodes or connections is not limited in GReaNs, but here we limit the size of the networks to analyze the functioning of the networks and the effect of network size on the evolvability of robustness to noise. Our results show that even very small networks of spiking neurons can perform temporal pattern recognition in the presence of input noise.

## Introduction

It is widely accepted that the brain employs temporal patterns of spikes to encode sensory input (for review, see Bialek et al., 1991; Gerstner et al., 1996; Laurent, 1996; Rieke et al., 1997; deCharms and Zador, 2000; Ahissar and Arieli, 2001; Huxter et al., 2003). Temporal coding forms the basis of sensory processing across different modalities, including hearing (Joris and Yin, 2007), vision (Thorpe et al., 1996), and olfaction (Isaacson, 2010). Nonetheless—in spite of the ubiquity of temporal coding in neuronal systems—the mechanisms that underlie the generation and recognition of temporal spike patterns are unclear.

It has been recognized that temporal pattern recognition can be performed by a system that contains an array of tuned delay lines and a coincidence detection mechanism (for example, Hopfield, 1995). Such a system, however, requires the existence of a developmental process or a learning algorithm that generates the appropriate delays, either by picking them from a spectrum of existing delays, or by adjusting them directly (Steuber and Willshaw, 1999; Steuber and De Schutter, 2002; Steuber and Willshaw, 2004).

In the present work, we evolve spiking neuron networks to recognize temporal input patterns without adjusting signal conduction delays. To evolve the networks, we modified

the GReaNs (the name stands for Gene Regulatory evolving artificial Networks; Wróbel and Joachimczak, 2011) artificial life platform to allow for spiking nodes (Wróbel et al., 2012a). GReaNs has been used previously to evolve gene regulatory networks for tasks including controlling multicellular development in three dimensions (e.g., Joachimczak and Wróbel, 2008, 2012a,b), processing signals (Joachimczak and Wróbel, 2010b), and controlling animats (Joachimczak and Wróbel, 2010a; Wróbel et al., 2012b).

The next section briefly describes how the spiking neural networks (SNNs) are encoded in an evolving linear genome in GReaNs. We then follow with the description of the task for which the networks were evolved—recognition of a pattern of spikes, with varying levels of noise affecting spike times. We end by discussing the results and the way in which the networks perform the computation.

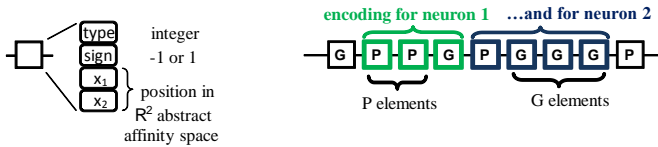
## Evolving spiking neural networks encoded in a linear genome

The encoding of spiking neural networks (SNNs) in GReaNs is inspired by the encoding of gene regulatory networks in biological linear genomes (Wróbel et al., 2012a). Linear genomes are lists of genetic elements (Fig. 1), in our case each is a vector of 4 numbers that can mutate during evolution (type, sign, coordinate 1, coordinate 2). There are two main types of elements, which we call: *P* and *G*. *P* elements correspond to post-synaptic terminals (and are inspired by promoters in biological genomes). *G* elements correspond to pre-synaptic terminals (and are inspired by genes). A chain of *P* elements followed by a chain of *G* elements encodes a node in the network (one spiking neuron). Each node must have at least one *P* element followed by at least one *G* element.

Synaptic weights depend on the position in an abstract affinity space of points specified by elements' coordinates. If a *G* element and a *P* element specify points that overlap, the synaptic connection has maximum weight. The larger the distance, the lower the weight (using the inverse exponential function), provided the distance is below a pre-specified threshold (to prevent full connectivity). The ele-

ments' sign defines if the synapse is excitatory (if the signs are the same) or inhibitory (if they are different).

In addition to  $P$  and  $G$  elements,  $I$  and  $O$  elements in the genome encode inputs and outputs, respectively, working as clamped pre- and post-synaptic terminals. The connectivity between  $I$  and  $P$  elements, and  $G$  and  $O$  elements is specified as described above for the connections between  $G$  and  $P$ . No other connections (or directions) are allowed. Importantly, the number of pre-synaptic and post-synaptic terminals is not limited in GReaNs (neither for the network, nor individual neurons). The number of neurons in the network is also not limited in the model (but we have limited—and varied—it in the experiments described in this paper).



**Figure 1:** The structure of the genome in GReaNs. Each genetic element (left part) has integer value for type ( $P$ ,  $G$ ,  $I$ , or  $O$ ), sign (1 or -1; the signs of two elements determine if a synapse is inhibitory or excitatory), and the coordinates (which determine the weights of the synapses). The genome (right part) is a chain of genetic elements.

## Genetic algorithm

The genome is evolved with a genetic algorithm, in which genetic operators are applied on the level of the elements (here: with probability 0.0005—change of type, probability 0.0005—change of sign, with probability 0.02—sampling two random values from a Gaussian distribution centred at 0, and adding one value to each coordinate), and at the level of the whole genome (here: adding a chain of random elements, deleting a chain of elements, duplicating a chain elsewhere; each of these operators is considered at each element in the parent genome, with probability 0.1, and then the length of the chain is sampled from a logarithmic distribution with mean 10, and the site of duplication from a uniform distribution).

The genetic algorithm is initialized with a population with random genomes. In this paper, the population size was constant throughout all evolutionary runs (300 genomes). All elements in the initial populations had coordinates sampled from a Gaussian distribution, and all the initial genomes in each experiment encoded a network with five intraneurons, encoded by the chains of  $P$  and  $G$  elements with the length sampled from a Gaussian distribution (with mean = 1, and standard deviation = 1; if the length smaller than 1 was sampled, it was set to 1).

To calculate the fitness of the genomes (see below), the performance of the network encoded by each genome is

tested. After selection (size two tournament), a new population is created, by applying the genetic operators described above to the best genomes, plus recombination (here: with probability 0.5 per genome, one-point cross-over with a random genome from the parent population, sampled uniformly). In the experiments described here, we used elitism—the five best genomes were copied (when new generations were formed from the old) without applying any genetic operators. Crossover and mutations were applied to 100 genomes, and only mutations—to the remaining 195 genomes. All the evolutionary runs in this paper were halted after 1000 generations.

## Evolving networks of Leaky Integrate and Fire neurons for temporal pattern recognition

In this paper, we use Leaky Integrate and Fire (LIF) (Dayan and Abbott, 2005) neurons. GReaNs allows to use also Adaptive Exponential (Gerstner and Brette, 2005) neurons (Wróbel et al., 2012a), but they were not employed in the experiments described here.

LIF is one of the simplest and most commonly used spiking neural models. The rate of change of membrane potential is specified by the equation

$$\dot{V} = \frac{g_L(V_R - V) + g_E(E_{rev,E} - V) + g_I(E_{rev,I} - V)}{C} \quad (1)$$

where  $V$  is the membrane potential,  $V_R = -65.0$  mV is the resting potential,  $g_L = 0.05$   $\mu$ S is the leak conductance,  $g_E$  is the conductance of the excitatory synapses,  $g_I$  is the conductance of the inhibitory synapses,  $E_{rev,E} = 0$  mV is the reversal potential of the excitatory input,  $E_{rev,I} = -70.0$  mV is the reversal potential of inhibitory input, and  $C = 1$  nF is the capacitance of the membrane.

When the postsynaptic potential of a neuron reaches the threshold voltage ( $V_{th}$ ), the neuron fires a spike and the value of the membrane potential is changed to the reset voltage ( $V_{reset}$ ).

In the GReaNs implementation of LIF neurons, the postsynaptic conductance increases by a small value proportional to the weight of the synapse when the pre-synaptic neuron (or input) spikes, and then the conductance decays exponentially:

$$\dot{g} = \frac{-g}{\tau} \quad (2)$$

where  $\tau = 5.0$  ms, the decay time constant, is the same for both the excitatory and inhibitory synaptic conductance.

In the genetic algorithm, we used the following error function as the fitness measure (rewarding lower values):

$$f_{err} = 1 - \frac{\min(S_{desired}, \alpha S_1) - \beta \sum_{i=2}^6 S_i}{S_{desired}} \quad (3)$$

$S_i$  is the number of spikes generated by a network when it is connected to the pattern number  $i$  during the period between 250 and 1000 ms,  $S_{desired}$  is the desired number of spikes to be generated by the network when it is connected with the correct pattern (here: 250), while  $\alpha$  and  $\beta$  are constant fractions (here:  $\alpha = 1$ , and  $\beta = 0.2$ ).

In other words, the task for which the networks are evolved is the recognition of the correct pattern, rewarded by the first term,  $min(S_{desired}, \alpha S_1)$ , in the numerator. The correct pattern used in the experiments described here comprised three spikes, each spike arriving from one input: one spike from input 1 at 50 ms, one from input 2 at 150 ms, and one from input 3 at 250 ms. We will call this pattern 1-2-3. Network activity in response to the other five patterns, with the same spike timings, but a different order (for example, 2-3-1, in which the first spike is from input 2), were penalized by the second term,  $\beta \sum_{i=2}^6 S_i$ .

When evolving pattern recognition in the presence of noise in the input, temporal noise was added to spike times (by adding a temporal shift, sampled from the Gaussian distribution centered at zero, and with the standard deviation 10, 20, or 30 ms). We performed 10 independent evolutionary runs for each setting (the size of the network, and the level of noise; 160 runs in total). During evolution, the networks were evaluated one time when they were evolved without noise (with all the patterns, six simulations in total), or 100 times when they were evolved with noise (600 simulations in total)—and the  $f_{err}$  value was calculated as the average of these 100 values. After the runs ended, the champions were tested in the same fashion, to obtain the performance of the best among the champions, the average (among 10) and the standard deviation.

## Results

To investigate the effect of network size on performance, we studied the evolution of networks limiting the maximum number of neurons allowed. In GReaNs, there are no constraints on the size of the genome, and thus the size of the network (even though available computational resources may restrict the feasible network size in practice). In the experiments described here, we simply ignored the rest of the genome after decoding the required number of nodes (interneurons, inputs, and outputs).

For the simplest network, with only one neuron (Fig. 2), there is only one possible topology, since the direct connection between inputs and output was not permitted. To detect the 1-2-3 pattern, this single neuron needs to have a strong excitatory connection to the output (so that the output will start spiking when there is a spike at 50 ms in input 1), and an excitatory recurrent connection from the intraneuron to itself so that it continues spiking. We could not, however, evolve networks with just one interneuron that would be robust to noise (Table 1)—for the reason that will become clear

after analyzing the behavior of robust networks with more interneurons.

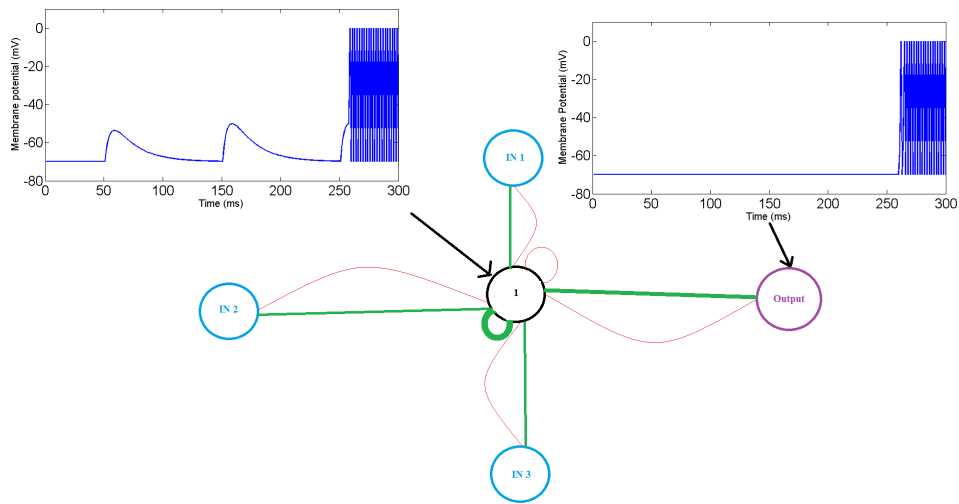
The networks evolved with the maximum number of two interneurons can evolve robustness to noise (Table 2) and their functioning can be analyzed in detail. Our analysis of champion networks from 10 independent evolutionary runs with the temporal noise of 10 ms allows us to classify them into three categories:

**Category 1** In 3 out of 10 champions, one interneuron (A) acts as a detector of a spike in input 2; once this interneuron starts firing at high frequency (because of a recurrent connection), the second interneuron (B) goes to a plateau state—the membrane potential is higher than the resting potential, but still sub-threshold—interneuron B will start spiking once a spike from input 3 arrives. Because of this plateau state, the activation of the interneuron B does not depend on the time between spikes in input 2 and input 3 (which allows for the robustness to noise). Input 1 inhibits both interneurons, so they remain silent with input patterns 2-3-1 or 2-1-3. With 1-2-3, interneuron B starts spiking at 350 ms (100 ms after the spike on input, at 250 ms), allowing for robustness to noise on input 1.

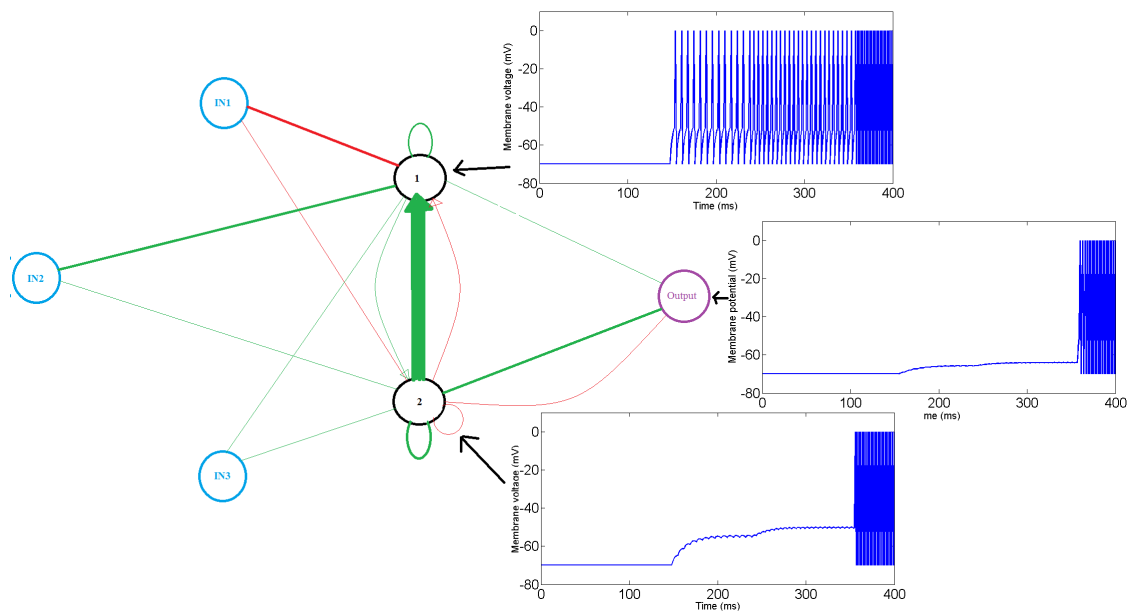
**Category 2** In 5 out of 10 champions, one interneuron (A) starts spiking after the spike in input 1, and thanks to the positive feedback loop, it keeps firing, putting the other interneuron (B) in a plateau state. When the spike from input 2 arrives, interneuron A starts firing at a higher rate, putting interneuron B in a higher plateau state. Only a small stimulus—a spike from input 3—is now required so that interneuron B starts spiking, and thus also the output. Here also the plateau state allows for robustness to noise—the times between spikes in input 1 and 2, or 2 and 3 have no effect on the pattern recognition.

**Category 3** In 2 champions out of 10, one interneuron (A) acts as a detector of the time interval between spikes in input 1 and 2. If this interval is higher than the network was evolved for, the pattern is not recognized. Otherwise, once interneuron A puts the second interneuron in the plateau state, any timing of the spike in input 3 is sufficient for correct pattern recognition. This means, however, that although the networks in category 3 have zero error when tested with 10 ms noise (for which they were evolved), they are robust to higher noise levels only on the timing of the spike in input 3—in contrast to the networks in the categories 1 and 2, which are robust to higher noise levels on the timing of spikes in all the three inputs.

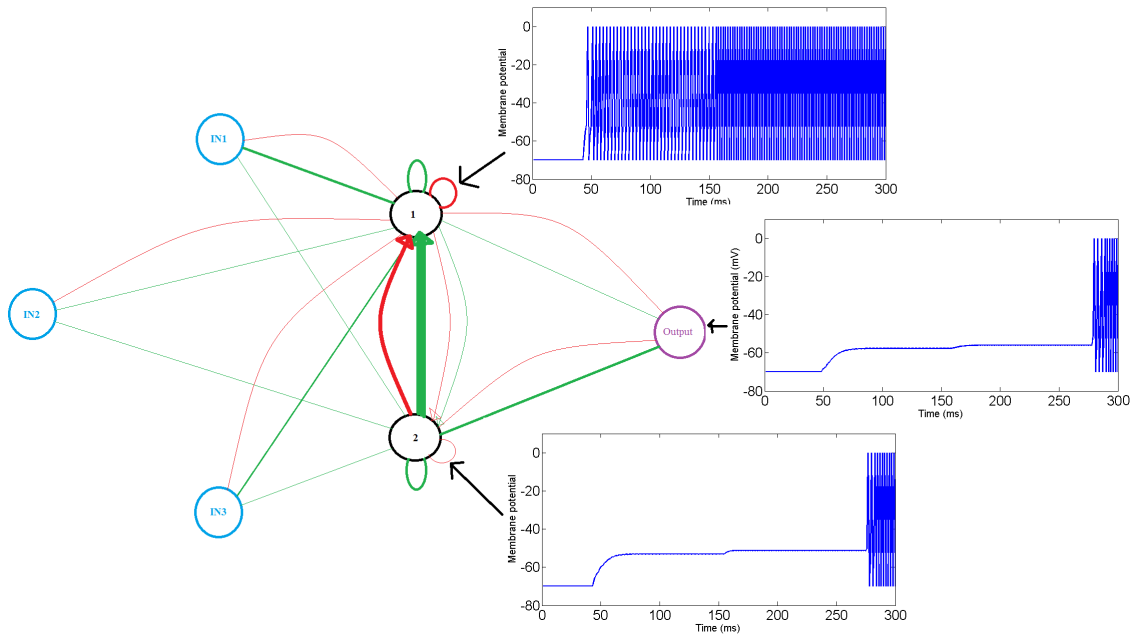
Even though just two interneurons allow for robustness to noise, allowing for more increases the performance (Tables 3, 4). Although these networks are too large to allow for the analysis as detailed as above, their functioning depends, similarly, on recurrent feedback loops that put some neurons in a high firing rate, and others at sub-threshold plateaus, so



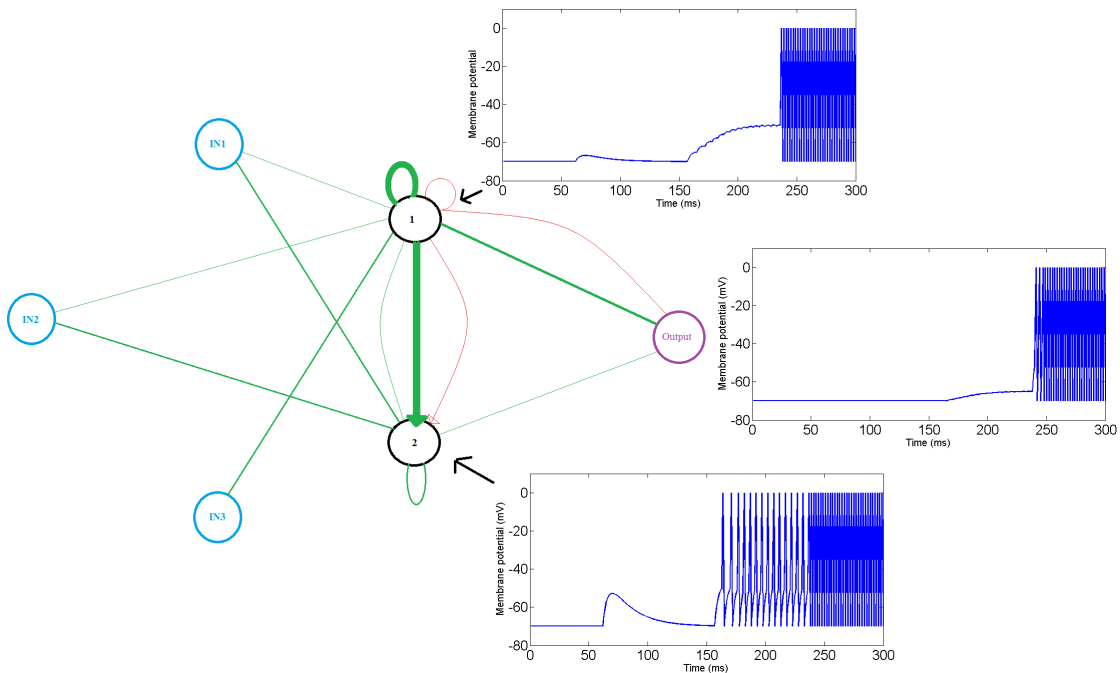
**Figure 2:** The final SNN with three inputs (cyan) each with only one spike at the times: 50 ms, 150 ms, and 250 ms, respectively, one interneuron (black), and one output (purple). The neurons are connected with both excitatory connections (green) and inhibitory connections (red). Line thickness indicates total synaptic weight.



**Figure 3:** A champion SNN with two interneurons that belongs to category 1. The network has three inputs (cyan), which spike (in the correct pattern) at times 50 ms, 150 ms, and 250 ms, respectively; two interneurons (black), and one output (purple). The neurons are connected with both excitatory connections (green) and inhibitory connections (red). Line thickness indicates total synaptic weight.



**Figure 4:** A champion SNN with two interneurons that belongs to category 2. The network has three inputs (cyan), which spike (in the correct pattern) at times 50 ms, 150 ms, and 250 ms, respectively; two interneurons (black), and one output (purple). The neurons are connected with both excitatory connections (green) and inhibitory connections (red). Line thickness indicates total synaptic weight.



**Figure 5:** A champion SNN with two interneurons that belongs to category 3. The network has three inputs (cyan), which spike (in the correct pattern) at times 50 ms, 150 ms, and 250 ms, respectively; two interneurons (black), and one output (purple). The neurons are connected with both excitatory connections (green) and inhibitory connections (red). Line thickness indicates total synaptic weight.

**Table 1:** Evolution of robustness to noise in temporal pattern recognition in LIF networks with one intraneuron. The networks evolved with various levels of noise in the input (Gaussian noise added to spike times, centered at zero and with indicated standard deviation, sd) were tested with the same (or different) level of noise that (than) was used during evolution. The first value shows the performance of the best network in 10 runs, the values in brackets are averages  $\pm$  sd.

Testing	Temporal noise during evolution			
	sd=0 ms (no noise)	sd=10 ms	sd=20 ms	sd=30 ms
sd=0 ms (no noise)	0 (0.09 $\pm$ 0.06)	0.05 (0.056 $\pm$ 0.012)	0.05 (0.07 $\pm$ 0.04)	0.049 (0.08 $\pm$ 0.076)
sd=10 ms	0.017 (0.48 $\pm$ 0.21)	0.198 (0.25 $\pm$ 0.06)	0.2 (0.23 $\pm$ 0.1)	0.196 (0.235 $\pm$ 0.1)
sd=20 ms	0.042 (0.55 $\pm$ 0.2)	0.29 (0.39 $\pm$ 0.05)	0.22 (0.27 $\pm$ 0.09)	0.216 (0.271 $\pm$ 0.093)
sd=30 ms	0.11 (0.56 $\pm$ 0.18)	0.4 (0.49 $\pm$ 0.06)	0.25 (0.37 $\pm$ 0.09)	0.229 (0.361 $\pm$ 0.081)

**Table 2:** Evolution of robustness to noise in temporal pattern recognition in LIF networks with two intraneurons. The networks evolved with various levels of noise in the input (Gaussian noise added to spike times, centered at zero and with indicated standard deviation, sd) were tested with the same (or different) level of noise that (than) was used during evolution. The first value shows the performance of the best network in 10 runs, the values in brackets are averages  $\pm$  sd.

Testing	Temporal noise during evolution			
	sd=0 ms (no noise)	sd=10 ms	sd=20 ms	sd=30 ms
sd=0 ms (no noise)	0 (0.026 $\pm$ 0.052)	0 (0.04 $\pm$ 0.07)	0.05 (0.18 $\pm$ 0.2)	0 (0.08 $\pm$ 0.12)
sd=10 ms	0 (0.36 $\pm$ 0.22)	0 (0.12 $\pm$ 0.1)	0 (0.18 $\pm$ 0.066)	0 (0.145 $\pm$ 0.09)
sd=20 ms	0.04 (0.41 $\pm$ 0.21)	0 (0.21 $\pm$ 0.16)	0.01 (0.2 $\pm$ 0.1)	0 (0.17 $\pm$ 0.11)
sd=30 ms	0.08 (0.47 $\pm$ 0.19)	0.05 (0.31 $\pm$ 0.17)	0.06 (0.28 $\pm$ 0.11)	0.07 (0.27 $\pm$ 0.11)

**Table 3:** Evolution of robustness to noise in temporal pattern recognition in LIF networks with five intraneurons. The networks evolved with various levels of noise in the input (Gaussian noise added to spike times, centered at zero and with indicated standard deviation, sd) were tested with the same (or different) level of noise that (than) was used during evolution. The first value shows the performance of the best network in 10 runs, the values in brackets are averages  $\pm$  sd.

Testing	Temporal noise during evolution			
	sd=0 ms (no noise)	sd=10 ms	sd=20 ms	sd=30 ms
sd=0 ms (no noise)	0 (0 $\pm$ 0)	0 (0.06 $\pm$ 0.13)	0.05 (0.09 $\pm$ 0.11)	0.05 (0.06 $\pm$ 0.02)
sd=10 ms	0 (0.4 $\pm$ 0.25)	0 (0.07 $\pm$ 0.09)	0.006 (0.18 $\pm$ 0.06)	0.03 (0.18 $\pm$ 0.05)
sd=20 ms	0 (0.42 $\pm$ 0.25)	0 (0.13 $\pm$ 0.12)	0.03 (0.23 $\pm$ 0.07)	0.07 (0.2 $\pm$ 0.05)
sd=30 ms	0.04 (0.5 $\pm$ 0.23)	0.054 (0.24 $\pm$ 0.15)	0.07 (0.33 $\pm$ 0.01)	0.08 (0.29 $\pm$ 0.08)

**Table 4:** Evolution of robustness to noise in temporal pattern recognition in LIF networks with 10 intraneurons. The networks evolved with various levels of noise in the input (Gaussian noise added to spike times, centered at zero and with indicated standard deviation, sd) were tested with the same (or different) level of noise that (than) was used during evolution. The first value shows the performance of the best network in 10 runs, the values in brackets are averages  $\pm$  sd.

Testing	Temporal noise during evolution			
	sd=0 ms (no noise)	sd=10 ms	sd=20 ms	sd=30 ms
sd=0 ms (no noise)	0 (0 $\pm$ 0)	0 (0.02 $\pm$ 0.06)	0 (0.06 $\pm$ 0.08)	0 (0.16 $\pm$ 0.08)
sd=10 ms	0.03 (0.4 $\pm$ 0.29)	0 (0.02 $\pm$ 0.06)	0 (0.05 $\pm$ 0.08)	0 (0.165 $\pm$ 0.08)
sd=20 ms	0.14 (0.46 $\pm$ 0.25)	0 (0.04 $\pm$ 0.08)	0 (0.08 $\pm$ 0.1)	0.01 (0.18 $\pm$ 0.08)
sd=30 ms	0.4 (0.54 $\pm$ 0.2)	0.07 (0.14 $\pm$ 0.01)	0.02 (0.17 $\pm$ 0.1)	0.05 (0.24 $\pm$ 0.09)

that they start firing when a spike from the next input arrives in the correct order.

## Discussion

We can conclude from the analysis of the behavior of the SNNs with various numbers of interneurons that the feedback loops play a crucial role in transferring the network from one plateau state to another, until it becomes active. We can now also understand why the networks with just one interneuron cannot reach 0 error when noise is present during the evolution.

Importantly, once robustness evolved, the networks remained robust to higher and lower noise levels in the input than they experienced during evolution (Tables 2, 3, 4). This is in contrast to the results of the experiments on the robustness of multicellular development to noise on the activity of non-spiking nodes in the artificial gene regulatory networks evolved with GReaNs (Joachimczak and Wróbel, 2012a). In future work we plan to evolve robustness to noise introduced to the membrane potential of interneurons and/or the delays in the connections between them. Another interesting issue for further research is the relation between the evolvability and robustness to noise. Evolvability requires that mutations result in small phenotypic changes—a property related to robustness to noise, and to graceful degradation in the presence of damage (see e.g., Voigt et al., 2005).

The ability to recognize temporal input patterns is a characteristic that is shared by biological organisms at all levels of complexity. Temporal patterns of sensory inputs range from spatio-temporal gradients of nutrients in bacteria and protozoa to speech recognition in humans and echolocation in bats. In particular, vertebrate brains are believed to perform a large number of diverse temporal pattern recognition tasks (for review, see Bialek et al., 1991; Gerstner et al., 1996; Laurent, 1996; Thorpe et al., 1996; Rieke et al., 1997; deCharms and Zador, 2000; Ahissar and Arieli, 2001; Huxter et al., 2003; Joris and Yin, 2007; Isaacson, 2010).

Often, neural systems that perform temporal pattern recognition are thought to combine a coincidence detection mechanism with the ability to generate arrays of finely tuned time delays (Hopfield, 1995; Steuber and Willshaw, 1999; Steuber and De Schutter, 2002; Steuber and Willshaw, 2004; Steuber et al., 2006; Maex and Steuber, 2009). Animals are able, however, to recognize temporal patterns in the presence of large amounts of noise, and it is unclear how robust an array of finely tuned delay lines would be against noise in the temporal inputs patterns.

In the present work, we have investigated potential mechanisms of noise-robust temporal pattern recognition by evolving spiking neural networks for pattern recognition tasks, both in the presence and absence of noise. Our main result is that a spiking neural network can develop as a temporal pattern recognition device without being able to adjust signal propagation delays. Even without providing delay

lines, the temporal pattern recognition in the evolved networks is based on coincidence detection. But it is a particular form of coincidence detection—one that involves the sequential activation of a small number of neurons in the network. Moreover, positive feedback loops are the essential elements in these networks that provide robustness against temporal noise. Our results show that very small networks of spiking neurons can be able to perform surprisingly complex computational tasks in a noisy environment.

## Acknowledgements

This work was supported by the Polish National Science Center (project EvoSN, UMO-2013/08/M/ST6/00922). AA is supported by the the Foundation for Polish Science, co-financed by EU Regional Development Fund (Innovative Economy Operational Programme 2007-2013). Computational resources were provided by the Tri-City Academic Computer Centre (TASK), and the Interdisciplinary Centre for Molecular and Mathematical Modeling (ICM, University of Warsaw; project G33-8). We are grateful to Michał Joachimczak for his help in debugging GReaNs.

## References

- Ahissar, E. and Arieli, A. (2001). Figuring space by time. *Neuron*, 32:185–201.
- Bialek, W., Rieke, F., de Ruyter van Steveninck, R. R., and Warland, D. (1991). Reading a neural code. *Science*, 252:1854–1857.
- Dayan, P. and Abbott, L. (2005). *Theoretical Neuroscience*. MIT Press.
- deCharms, R. C. and Zador, A. (2000). Neural representation and the cortical code. *Neuroscience*, 23:613–647.
- Gerstner, W. and Brette, R. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Neurophysiology*, 94:3637–3642.
- Gerstner, W., Kempter, R., van Hemmen, J. L., and Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383:76–78.
- Hopfield, J. (1995). Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376:33–36.
- Huxter, J., Burgess, N., and O’Keefe, J. (2003). Independent rate and temporal coding in hippocampal pyramidal cells. *Nature*, 425:828–832.
- Isaacson, J. (2010). Odor representations in mammalian cortical circuits. *Current Opinion in Neurobiology*, 20:328–331.
- Joachimczak, M. and Wróbel, B. (2008). Evo-devo *in silico*: a model of a gene network regulating multicellular development in 3D space with artificial physics. In *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 297–304. MIT Press.

- Joachimczak, M. and Wróbel, B. (2010a). Evolving gene regulatory networks for real time control of foraging behaviours. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Simulation and Synthesis of Living Systems*, pages 348–355. MIT Press.
- Joachimczak, M. and Wróbel, B. (2010b). Processing signals with evolving artificial gene regulatory networks. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Simulation and Synthesis of Living Systems*, pages 203–210. MIT Press.
- Joachimczak, M. and Wróbel, B. (2012a). Evolution of robustness to damage in artificial 3-dimensional development. *Biosystems*, 109:498–505.
- Joachimczak, M. and Wróbel, B. (2012b). Open ended evolution of 3d multicellular development controlled by gene regulatory networks. In *Artificial Life XIII: Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems*, pages 67–74. MIT Press.
- Joris, P. and Yin, T. C. T. (2007). A matter of time: Internal delays in binaural processing. *Trends in Neurosciences*, 30:70–78.
- Laurent, G. (1996). Dynamical representation of odors by oscillating and evolving neural assemblies. *Trends in Neurosciences*, 19:489–496.
- Maex, R. and Steuber, V. (2009). The first second: Models of short-term memory traces in the brain. *Neural Networks*, 22:1105–1112.
- Rieke, F., Warland, D., de Ruyter van Steveninck, R. R., and Bialek, W. (1997). *Spikes: Exploring the Neural Code*. MIT Press.
- Steuber, V. and De Schutter, E. (2002). Rank order decoding of temporal parallel fibre input patterns in a complex purkinje cell model. *Neurocomputing*, 44:183–188.
- Steuber, V. and Willshaw, D. (1999). Adaptive leaky integrator models of cerebellar purkinje cells can learn the clustering of temporal patterns. *Neurocomputing*, 26:271–276.
- Steuber, V. and Willshaw, D. (2004). A biophysical model of synaptic delay learning and temporal pattern recognition in a cerebellar purkinje cell. *Journal of Computational Neuroscience*, 17:149–164.
- Steuber, V., Willshaw, D., and van Ooyen, A. (2006). Generation of time delays: simplified models of intracellular signalling in cerebellar purkinje cells. *Computation in Neural Systems*, 17:173–191.
- Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 381:520–522.
- Voigt, C. A., Mayo, S. L., Wang, Z.-G., and Arnold, F. (2005). Directing the evolvable: Utilizing robustness in evolution. In Jen, E., editor, *Robust Design: Repertoire of Biological, Ecological, and Engineering Case Studies*, pages 105–134. Oxford University Press.
- Wróbel, B., Abdelmotaleb, A., and Joachimczak, M. (2012a). Evolving networks processing signals with a mixed paradigm, inspired by gene regulatory networks and spiking neurons. In *Proceedings of the 7th International Conference on Bio-Inspired Models of Network, Information, and Computing Systems (BIONETICS)*, page in press. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.
- Wróbel, B. and Joachimczak, M. (2011). Using GReaNs (genetic regulatory evolving artificial networks) for 3-dimensional asymmetrical pattern formation and morphogenesis. In *Proceedings of DevLeaNN: A Workshop on Development and Learning in Artificial Neural Networks*, pages 26–28, Paris, France.
- Wróbel, B., Joachimczak, M., Montebelli, A., and Lowe, R. (2012b). The search for beauty: Evolution of minimal cognition in an animat controlled by a gene regulatory network and powered by a metabolic system. In *From Animals to Animats 12: The 12th International Conference on the Simulation of Adaptive Behavior (SAB 2012)*, volume 7426 of *Lecture Notes in Computer Science*, pages 198–208. Springer.