# Robust Very Small Spiking Neural Networks Evolved with Noise to Recognize Temporal Patterns

Muhammad Yaqoob[1], Borys Wróbel[1,2]

[1]Evolving Systems Laboratory, Adam Mickiewicz University in Poznan, Poland
[2]Systems Modeling Group, IOPAN, Sopot, Poland
yaqoob@evosys.org
wrobel@evosys.org

## Abstract

To understand how biological and bio-inspired complex computational networks can function in the presence of noise and damage, we have evolved very small spiking neural networks in the presence of noise on the membrane potential. The networks were built with adaptive exponential integrate and fire neurons. The simple but not trivial task we evolved the networks for consisted of recognizing a short temporal pattern in the activity of the network inputs. This task can be described in abstract terms as finding a specific subsequence of symbols (”ABC”) in a continuous sequence of symbols (”..ABCCCAAABCAC..”). We show that networks with three interneurons and one output neuron can solve this task in the presence of biologically plausible levels of noise. We describe how such a network works by mapping its activity onto the state of a finite state transducer—an abstract model of computation on continuous time series. We demonstrate that the networks evolved with noise are much more robust than networks evolved without noise to the modification of neuronal parameters and variation of the properties of the input. We also show that the networks evolved with noise are denser and have stronger connections than the networks evolved without noise. Finally, we demonstrate the emergence of memory in the evolved networks—sustained spiking of some neurons maintained thanks to the presence of self-excitatory loops.

## Introduction

Natural complex systems, including networks of biological neurons, maintain their functionality in the presence of noise and damage. Noise in natural neural networks originates from many sources, including thermal variations and small number of cellular components (for example, ion channels). These components, moreover, undergo constant turnover, and so do parts of the cell (such as dendrites), and the cells themselves. Including noise in models of biological systems helps in our understanding how reliable computation can be performed in the presence of noise, and in building more reliable artificial systems (Florian, 2003).

In biological neuronal networks and in artificial spiking neural networks, the timing of discrete events (spikes) represents the information received from the senses. Processing this information requires recognizing temporal patterns in neuronal activity by other neurons (Bialek et al., 1989; Gerstner et al., 1996; Laurent, 1996; Rieke, 1999; Decharms and Zador, 2000; Ahissar and Arieli, 2001; Huxter et al., 2003). Recognition of temporal patterns requires delays or maintaining the state of the network (Steuber and Willshaw, 1999; Steuber and De Schutter, 2002; Steuber and Willshaw, 2004; Steuber et al., 2006; Maex and Steuber, 2009). Intuitively, the necessity for precise synaptic delays seems a more fragile solution.

In this paper, we evolve very small spiking neural networks for simple pattern recognition in the presence of noise. We hope that analyzing the diverse solutions obtained using artificial evolution will allow us to identify the way robust pattern recognition can be accomplished. To represent the computation performed by the evolved spiking neural networks, we will use a formal computational model of the finite state transducer—a deterministic finite state automaton that receives a continuous sequence of symbols and produces a continuous output (Sipser, 1996). Our focus here is not the induction of a specific finite state automaton (as in Natschläger and Maass, 2002; Tiňo and Mills, 2005; Rutishauser and Douglas, 2009; where large recurrent multilayer spiking neural networks were used). Rather, we will use the formalism to illustrate how the evolved networks work. Finally, we will compare the functioning and structure of networks evolved in the presence of noise to networks evolved without it (which were the subject of our previous work, Yaqoob and Wróbel, 2017). Our preliminary analysis shows that even a relatively low level of noise during evolution results in much more robust networks, and that the networks evolved with noise are denser than the networks evolved without it.

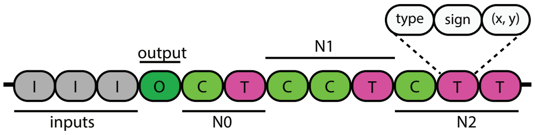

Figure 1: Encoding of a spiking neural network in a linear genome. See text for details.

## The Model

We use the model of evolution of spiking neural networks for temporal pattern recognition introduced previously in the artificial life software platform GReaNs (Wróbel et al., 2012; Wróbel, 2016; Yaqoob and Wróbel, 2017). The networks in this model are encoded in linear genomes built from genetic elements. Each element has a type (*input*, *output*, *cis* and *trans*; the biological inspiration for such a representation and these terms has been discussed previously; Fig. 1), sign and two coordinates. The elements encode the nodes in the network: three input nodes (encoded by an *input* element each), up to three interneurons (each encoded by a series of *cis* elements followed by a series of *trans* elements), and one output neuron (encoded by a single *output* element). To determine the connectivity, every pair of *input-cis*, *trans-cis* and *trans-output* elements is considered. If the coordinates of two elements in a pair are such that the Euclidean distance is below a threshold (equal to 5), the presence of such a pair contributes to the weight of the connection between two nodes; the contribution is positive if the signs of the two elements coincide and negative otherwise (the contribution is $s_i s_j \frac{2(5-d_{i,j})}{10d_{i,j}+1}$, where $s_i, s_j$ are signs and $d_{i,j}$ is distance; the threshold prevents full connectivity). If the overall sum of such contributions for two neurons is positive, the link (synapse) is excitatory, otherwise it is inhibitory.

In our previous work (Yaqoob and Wróbel, 2017) we have shown that two interneurons are sufficient for the simple pattern recognition task considered here, with no noise. We have subsequently found out that artificial evolution could not find any efficient solution when noise was present and the network size was limited to two interneurons, but could do so with three interneurons. In the evolutionary runs described here we have thus limited the number of interneurons in the decoded network to up to three, by ignoring the rest of the genetic elements. This restriction was imposed both to limit the search space and to ease the analysis of the networks. Similarly, superfluous input or output elements, introduced for example by unequal crossing over (see below) were ignored.

As in our previous work (Yaqoob and Wróbel, 2017), in this paper we used the adaptive exponential integrate and fire neuronal model for each interneuron and output neuron. Each adaptive exponential neuron has four state variables (membrane potential $V$, adaptation $w$, excitatory conductance $g_E$, and inhibitory conductance, $g_I$):

$$\begin{aligned}\frac{dV}{dt} &= \frac{1}{C}(g_E(E_E - V) + g_I(E_I - V) - w) \\ &\quad + \frac{1}{\tau_m}(E_L - V + \Delta_T e^{(\frac{V-V_T}{\Delta_T})}) \qquad (1)\end{aligned}$$

$$\tau_w \frac{dw}{dt} = a(V - E_L) - w \qquad (2)$$

$$\frac{dg_E}{dt} = \frac{-g_E}{\tau_E} \qquad (3)$$

$$\frac{dg_I}{dt} = \frac{-g_I}{\tau_I} \qquad (4)$$

The default values of the parameters we used here (Table 1) were the same as in our previous work (Yaqoob and Wróbel, 2017). They result in tonic spiking in response to constant input current (Naud et al., 2008).

When $V$ in the adaptive exponential neuron is high enough, $V$ quickly diverges to infinity because of the exponential term; this models a spike. For simulation purposes, the spike is cut at a finite value (here, 0 mV). After a spike occurs in a neuron, this neuron's $V$ is reset to $V_r$, and adaptation $w$ is incremented by $b$. In any neuron to which the neuron that spiked connects (in any postsynaptic neuron) with positive (negative) weight, the excitatory (inhibitory) conductance $g_E$ ($g_I$) is increased by synaptic gain ($gain_E$ or $gain_I$, respectively; here, 9 nS) multiplied by the absolute weight.

Modifying four parameters in the adaptive exponential neuron can bring qualitative change in neuronal behavior (qualitatively different responses to constant input current). These four bifurcation parameters (which are directly proportional to the four free parameters; Touboul and Brette, 2008; Naud et al., 2008) are: adaptation time constant $\tau_w$, adaptation conductance $a$, reset voltage $V_r$, and spike-triggered adaptation $b$. The remaining ones are scaling parameters: membrane capacitance $C$, threshold slope factor $\Delta_T$; three time constants, membrane ($\tau_m$), and excitatory/inhibitory ($\tau_E$/$\tau_I$); four potentials, effective rest ($E_L$), inhibitory ($E_I$) and excitatory ($E_E$) reverse, and effective threshold ($V_T$).

We used Euler integration with 1 ms step. When evolving with noise, a random value taken from a Normal distribution (mean 0, standard deviation 2 mV) was added to $V$ at every step. This level of noise is similar in magnitude to the level observed in biological neurons resulting from spontaneous or background synaptic activity (Paré et al., 1998; Destexhe and Paré, 1999; Anderson et al., 2000; Finn et al., 2007).

The task for the network was for the output neuron to spike at least once after the network received the activation from the input nodes in a specific order: first from input A, second from input B, third from input C. During both evolution and testing, input nodes were activated in a random order. Each such activation lasted for 6 ms. Only one input node could be active at a time. Each activation of an input node was followed by an interval of 16 ms during which no input node was active.

In abstract terms, this task corresponds to recognizing a pattern of symbols ("ABC") in a continuous stream of symbols {A, B, C} received in a random order. In terms of modeling, when a network receives a symbol, it means that six spikes, each one 1 ms apart, are received by all the interneurons to which the input node corresponding to the symbol connects.

Table 1: Robustness to change of neuronal parameters and properties of the input sequences for 10 best champions evolved with noise and the most robust champion evolved without noise. The values in square brackets show the range of robustness, the values above them show relative robustness (see text for details).

| | | Top 10 individuals evolved with noise | | | | | | | | | | most robust |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Parameter | default value | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | without noise |
| $E_L$ | -70 mV | 0.77 | 1.00 | 0.48 | 0.61 | 0.68 | 0.77 | 0.58 | 0.29 | 0.48 | 0.32 | 0.04 |
| | | [-81, -57] | [-91, -60] | [-76, -61] | [-78, -59] | [-86, -65] | [-88, -64] | [-78, -60] | [-75, -66] | [-77, -62] | [-73, -63] | [-71, -70] |
| $V_r$ | -58 mV | 0.42 | 0.79 | 0.46 | 0.71 | 0.50 | 0.46 | 0.21 | 0.79 | 0.58 | 1.00 | 0.13 |
| | | [-63, -53] | [-63, -44] | [-63, -52] | [-62, -45] | [-60, -48] | [-60, -49] | [-60, -55] | [-64, -45] | [-61, -47] | [-65, -41] | [-60, -57] |
| $V_T$ | -50 mV | 0.40 | 1.00 | 0.40 | 0.80 | 0.80 | 0.80 | 0.80 | 0.40 | 0.60 | 0.20 | 0.00 |
| | | [-51, -49] | [-53, -48] | [-51, -49] | [-52, -48] | [-51, -47] | [-51, -47] | [-52, -48] | [-52, -50] | [-51, -48] | [-51, -50] | [-50, -50] |
| $\tau_m$ | 20 ms | 0.39 | 1.00 | 0.95 | 0.55 | 0.35 | 0.42 | 0.53 | 0.17 | 0.96 | 0.22 | 0.06 |
| | | [8, 44] | [6, 99] | [12, 100] | [8, 59] | [6, 39] | [7, 46] | [8, 57] | [10, 26] | [11, 100] | [9, 29] | [16, 21] |
| $\Delta_T$ | 2 mV | 0.28 | 0.89 | 1.00 | 0.83 | 0.83 | 0.83 | 0.83 | 0.44 | 0.89 | 0.28 | 0.13 |
| | | [1.7, 2.2] | [1.3, 2.9] | [1.5, 3.3] | [1.3, 2.8] | [1.6, 3.1] | [1.7, 3.2] | [1.5, 3.0] | [1.4, 2.2] | [1.6, 3.2] | [1.6, 2.1] | [2.0, 2.2] |
| $C$ | 0.2 nF | 0.40 | 1.00 | 0.50 | 0.50 | 0.50 | 0.60 | 0.30 | 0.40 | 0.60 | 0.20 | 0.00 |
| | | [0.17, 0.21] | [0.15, 0.25] | [0.17, 0.22] | [0.18, 0.23] | [0.17, 0.22] | [0.17, 0.23] | [0.19, 0.22] | [0.16, 0.20] | [0.17, 0.23] | [0.19, 0.21] | [0.2, 0.2] |
| $a$ | 2 nS | 0.60 | 0.87 | 0.37 | 0.87 | 1.00 | 0.87 | 0.90 | 0.27 | 0.40 | 0.73 | 0.03 |
| | | [-4, 14] | [-7, 19] | [-4, 7] | [-5, 21] | [-2, 28] | [-5, 21] | [-8, 19] | [-1, 7] | [-3, 9] | [-9, 13] | [2, 3] |
| $b$ | 0 pA | 0.60 | 0.74 | 0.28 | 0.79 | 1.00 | 0.68 | 0.82 | 0.28 | 0.47 | 0.39 | 0.013 |
| | | [0, 34] | [0, 42] | [0, 16] | [0, 45] | [0, 57] | [0, 39] | [0, 47] | [0, 16] | [0, 27] | [0, 22] | [0, 7] |
| $\tau_E$ | 5 ms | 0.73 | 0.91 | 0.64 | 1.00 | 0.64 | 0.73 | 0.91 | 0.45 | 0.55 | 0.64 | 0.18 |
| | | [4.7, 5.5] | [4.7, 5.7] | [4.6, 5.3] | [4.4, 5.5] | [4.5, 5.2] | [4.6, 5.4] | [4.4, 5.4] | [4.9, 5.4] | [4.7, 5.3] | [4.9, 5.6] | [4.9, 5.1] |
| $\tau_I$ | 5 ms | 0.43 | 0.57 | 1.00 | 0.95 | 1.00 | 0.76 | 0.43 | 0.29 | 0.71 | 0.43 | 1.00 |
| | | [4.5, 5.4] | [4.6, 5.8] | [4.5, 6.6] | [4.5, 6.5] | [4.8, 6.9] | [4.8, 6.4] | [4.7, 5.6] | [4.8, 5.4] | [4.6, 6.1] | [4.5, 5.4] | [4.1, 6.1] |
| $E_E$ | 0 mV | 0.50 | 0.83 | 0.75 | 0.92 | 1.00 | 0.83 | 0.75 | 0.58 | 0.67 | 0.50 | 0.08 |
| | | [-2, 4] | [-4, 6] | [-4, 5] | [-4, 7] | [-8, 4] | [-6, 4] | [-5, 4] | [-1, 6] | [-4, 4] | [0, 6] | [-1, 0] |
| $E_I$ | -70 mV | 0.33 | 0.56 | 0.89 | 0.78 | 0.67 | 0.56 | 0.33 | 0.33 | 0.44 | 0.22 | 1.00 |
| | | [-71, -68] | [-72, -67] | [-76, -68] | [-74, -67] | [-75, -69] | [-74, -69] | [-72, -69] | [-71, -68] | [-73, -69] | [-70, -68] | [-74, -65] |
| $gain_E$ | 9 nS | 0.50 | 1.00 | 0.75 | 0.95 | 0.95 | 0.85 | 0.75 | 0.65 | 0.80 | 0.50 | 0.05 |
| | | [8.6, 9.6] | [8.2, 10.2] | [8.2, 9.7] | [8.3, 10.2] | [7.7, 9.6] | [7.9, 9.6] | [8.1, 9.6] | [8.8, 10.1] | [8.2, 9.8] | [8.9, 9.9] | [8.9, 9.0] |
| $gain_I$ | 9 nS | 0.21 | 0.40 | 0.92 | 0.46 | 0.46 | 0.35 | 0.28 | 0.26 | 0.40 | 0.15 | 1.00 |
| | | [8.0, 9.8] | [7.2, 10.6] | [7.9, 15.7] | [7.1, 11] | [8.2, 12.1] | [8.2, 11.2] | [7.7, 10.1] | [7.7, 9.9] | [7.7, 11.1] | [8.0, 9.3] | [7.1, 15.6] |
| $noise$ | 2 mV | 0.85 | 1.00 | 0.93 | 0.85 | 0.78 | 0.78 | 0.89 | 0.74 | 0.93 | 0.74 | 0.04 |
| | | [0, 2.5] | [0, 2.9] | [0, 2.7] | [0, 2.5] | [0, 2.3] | [0, 2.3] | [0, 2.6] | [0, 2.2] | [0, 2.7] | [0, 2.2] | [0, 0.1] |
| $silence$ | 16 ms | 0.95 | 0.25 | 0.25 | 1.00 | 0.20 | 0.20 | 0.70 | 0.55 | 0.20 | 0.60 | 0.16 |
| | | [13, 32] | [14, 19] | [13, 18] | [10, 30] | [12, 16] | [13, 17] | [11, 25] | [12, 23] | [14, 18] | [10, 22] | [15, 18] |
| $signal$ | 6 ms | 0.50 | 0.50 | 0.50 | 1.00 | 1.00 | 0.50 | 0.50 | 0.00 | 1.00 | 0.00 | 1.00 |
| | | [6, 7] | [5, 6] | [6, 7] | [5, 7] | [4, 6] | [5, 6] | [5, 6] | [6, 6] | [6, 8] | [6, 6] | [6, 8] |
| Average relative robustness | | 0.52 | 0.78 | 0.65 | 0.80 | 0.73 | 0.65 | 0.62 | 0.41 | 0.63 | 0.42 | 0.30 |

The genomes in the initial population were created randomly as described previously (Yaqoob and Wróbel, 2017). Each evolutionary run had a constant population size (300), with size two tournaments, elitism (10 individuals) and crossover (30 individuals in each generation; there was no crossover in Yaqoob and Wróbel, 2017).

Multi-point crossover was implemented in the following manner. First, two parents (A and B) are selected from population as winners of two independent size two tournaments. A cursor pointing to the genetic elements to be copied is initiated at the first element for both genomes. Then, one of the four schemes is chosen: copy an element from parent (i) A or (ii) B to offspring, advance cursor on both genomes, or copy from (iii) A or (iv) B, advance cursor only on the template copied from (each with probability 0.03). The probabilities of choosing actions (i) and (ii) were equal and 4 times larger than the (again, equal) probabilities of choosing (iii) or (iv). After an element is copied, the scheme stays the same as previously with probability 0.7, and otherwise a scheme is re-chosen (the same one can be chosen again), maintaining the ratio between the probabilities as above.

If an element was chosen for a point mutation (per element probability of 0.1), the coordinates were changed so that the associated point was moved in a random direction (drawn from a uniform distribution) by a distance drawn from a normal distribution (mean 0, standard deviation 1). Duplications (probability of 0.001 per genome) occurred twice as often as deletions (probability 0.0005). The starting element and the insertion site were chosen randomly (each element had the same chance of being chosen). The length of duplication/deletion was drawn from a geometric distribution with mean 11.

Each individual in the population was evaluated on six random sequences (different from each other, for each indidvidual, and in each generation), each with 500 symbols. Four sequences were generated with equiprobable occurrence of A, B, and C (and thus contained about 16 ABC subsequences each), and two other continuous sequences were constructed by concatenating, in random order, ABC with ABB, and ABA (two subsequences that are the most problematic to discriminate from ABC; Yaqoob and Wróbel, 2017).

The fitness function rewarded for spike(s) after the target subsequence and penalized for spikes elsewhere:

$$f_{fitness} = 1 - R + 4P \tag{5}$$

$R$ (for reward) is the fraction of time intervals when the input nodes are silent (each such silence is 16 ms long) after the last C in "ABC" and the output spikes at least once. In other words, it is the fraction of instances in which the output spikes correctly. $P$ (for penalty) is the fraction of instances in which the output spikes incorrectly. These instances can happen either in (i) 16 ms silence intervals not after ABCs, or in (ii) 6 ms time intervals in which one of the input nodes is active. Although $P$ in principle could reach 1, in practice it was always quite small, and the $f_{fitness}$ was below 1. We call an individual a perfect recognizer if $f_{fitness} = 0$. The number by which $P$ is multiplied, 4, was chosen after preliminary exploration to find a value that resulted in the highest evolvability (number of evolutionary runs that ended with perfect recognizers).

## Results and Discussion

Among 100 independent evolutionary runs without noise, 33 ended with perfect recognizers—champions with $f_{fitness} = 0$ when re-evaluated (tested) on 500 random input sequences (thus different than the sequences experienced during evolution) with equal probability of each symbol (in Yaqoob and Wróbel, 2017 we used different settings for artificial evolution—in particular, no cross over, different probabilities of duplications and deletions—and the yield was much lower; other difference were: time during which outputs were active were 4 ms, with 8 silences; the fitness function was more complex; excitatory/inhibitory gain was 5 nS; the output neuron had an offset current).

In the presence of noise, 1000 generations were needed to obtain 10 champions in 100 independent runs that were perfect recognizers when re-evaluated as above without noise—with noise they failed to produce a spike on output at most after 1 in 100 ABCs.

All perfect recognizers evolved without noise always produced only 1 spike in output after an ABC in the input sequence. The perfect recognizers evolved with noise belonged to two classes: when evaluated without noise, either (i) the output always spiked once, or (ii) always twice after each ABC. When evaluated with noise, the output neurons in both classes spiked once after some ABCs, and twice after the other ABCs in the same input sequence, but never more times.

To measure the robustness, we first analyzed what was the range of robustness for each parameter. In this preliminary analysis, only one parameter was changed at a time. The range of robustness was defined as the largest continuous set of parameter values around the default value for which a given network had the true positive rate of at least 99% and the false discovery rate of at most 5%. We define here the true positive rate and the false discovery rate as follows. The true positive rate is the average number of recognized ABCs (the number of 16 ms intervals after ABCs in which the output neuron spiked, correctly) divided by the actual number of ABCs in the input sequence. The false discovery rate is the average number of intervals (6 ms or 16 ms) in which the output spiked incorrectly (not in the interval of silence after an ABC), divided by the total number of intervals in which the output spiked. Since we are interested in temporal pattern recognition in a continuous input sequence, we actually evaluated the champions on 600-symbol input

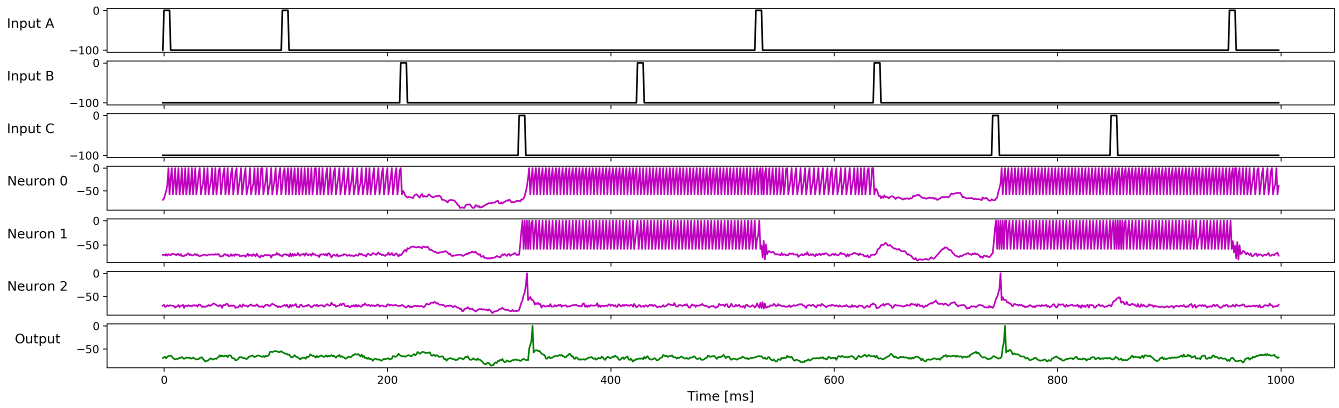

Figure 2: Network activity of the champion 3 evolved with noise on membrane potential. The individual is tested for signal length 6 ms, silence interval 100 ms (it evolved for 16 ms). The output neuron spikes after ABC around 330 ms and 700 ms

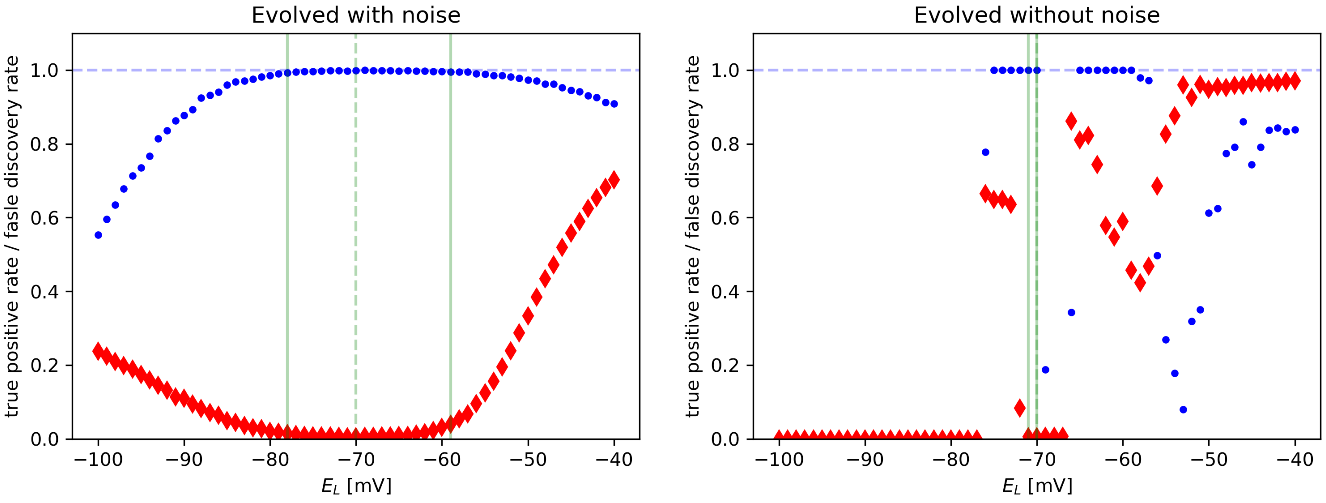

Figure 3: Example of robustness of network performance to change of a parameter (here, $E_L$). Blue circles show the true positive rate, red triangles show the false discovery rate (see text for more details). The range of robustness is showed by two continuous vertical lines, the dashed vertical line shows the default value, -70 mV. The network evolved (and tested) with noise (left; the network of champion 3) shows graceful degradation of network performance and a larger range of robustness than the network evolved (and tested) without noise (right; the network of the most robust champion evolved without noise).

sequences and considered the response to the last 500 symbols, over 500 such sequences. This is a very conservative approach as in practice discarding the response to the first symbol or a few at most would work equally well.

Second, we compared the ranges (differences between maximum and minimum value) for a given parameter across champions, defining relative robustness as the fraction of the maximum range for a given parameter among the champions (thus the champion with the largest range has the relative robustness 1.00, and the one with, say, half that range, has 0.50).

Third, we calculated the average relative robustness for each champion (Table 1; only the most robust champion evolved without noise is shown for simplicity). All the champions evolved with noise were robust to setting $\tau_w$ in the range from 1 to more than 1000 ms (the default value was 30 ms). Only the best champion evolved without noise was equally robust to changes of $\tau_w$, other champions evolved without noise had smaller ranges. When the integration step was changed from 1 ms to 0.5 ms, the champions evolved with noise displayed a drop in the true positive rate (from 0.99 to 0.98), while the false discovery rate remained unaffected. The networks evolved without noise were not robust to such a change.

The comparison between champions evolved and re-evaluated with noise to the champions evolved and re-evaluated without noise is conservative, as champions evolved with noise have much larger ranges of robustness when re-evaluated in the absence of noise. Moreover, the average relative robustness is a crude measure, as robustness to change of a particular parameter may be highly correlated to the robustness to change of other one(s); a more refined measure would give lower weights to relative robustness for parameters that belong to such a group.

We expected that noise will promote robustness to damage (here, change of neuronal parameters), as has been observed before in GReaNs for evolving gene regulatory networks, were damage affected an artificial developmental process (Joachimczak and Wróbel, 2012), and spiking neural networks (Wróbel, 2016).

In accordance with our expectations, the networks evolved with noise were much more robust to change in neuronal parameters than networks evolved without noise (Table 1; the second best, in terms of robustness, champion evolved without noise had average robustness 0.13, the rest had average robustness below 0.1).

Moreover, the networks evolved with noise showed graceful degradation beyond the range of robustness (for most of the parameters; $E_L$ is shown as an example in Fig. 3, taking champion 3 as the one evolved with noise and the most robust champion evolved without noise for comparison).

In contrast to gene regulatory networks regulating artificial development, networks evolved with noise functioned very well (in fact, better) when tested without noise. The most robust champion evolved with noise (champion 3) was, in particular, the most robust to lengthening the silences between the activity of the input nodes, and when this champion was tested without noise, these silences could be extended with no discernible limit, indicating that this network is able to maintain its states forever (Fig. 2).

Table 2: States of the finite state transducer corresponding to states of the network of the champion 3 evolved with noise

| | S | hA | hAB | hABC |
|---|---|---|---|---|
| Neuron 0 | 330 Hz | 333 Hz | 0 | 331 Hz |
| Neuron 1 | 333 Hz | 0 | 0 | 333 Hz |
| Neuron 2 | 0 | 0 | 0 | 1 spike |
| Output | 0 | 0 | 0 | 1 spike |

We can describe how this network functions by mapping the network activity to the states in a finite state transducer (Table 2, Fig. 4). Let us first assume that the network has already received some symbols (some input subsequence). If this subsequence ends with a C that did not follow AB or with a B that did not follow an A, the network is in the state S (starting state), in which (inter)neuron 0 and neuron 1 spike continuously, at high frequency. If an A is received, the network goes to a state hA ("had A"), in which only neuron 0 spikes in this fashion. If this A is followed by a B, all interneurons do not spike (state hAB, "had AB"). If this AB is followed by a C, neurons 0 and 1 again start spiking continuously, while neuron 2 and then the output neuron produce one spike each, immediately after the C (the state hABC, "had ABC").

Although this is not actually relevant for computing the output in response to a continuous input sequence, if the network receives a B with no previous history, it goes, essentially, to the state S. However, no previous history (and no activity in the network) is indistinguishable in this network from the state hAB. This is why the output neuron of this network will incorrectly spike after receiving a C with no previous history—in other words, if an input sequence starts with a C.

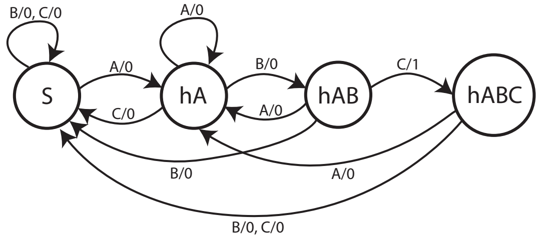

Figure 4: Minimal finite state transducer for recognizing ABC; the nodes represent states and edges represents transition from one state to another state on receiving an input symbol {A,B,C} and with producing an output {0: no spike(s), 1: spike(s) of the output neuron}.

The analysis of the network of the champion 3 evolved with noise (Fig. 5) shows that all interneurons connect to one another and each connects to itself (each has a self-loop). Two interneurons, neuron 0 and 1, have excitatory self-loops, which seem to be responsible for maintaining the continuous high-frequency spiking of these two neurons in states S, hA, hAB, and hABC. On the other hand, the recurrent inhibitory connections between interneurons (neuron 1 excites neuron 0, while 0 inhibits 1; neuron 2 excites 0, while 0 inhibits 2; neurons 1 and 2 inhibit one another) seem to, together with inhibitory connections from the input nodes to interneurons, bring the end of continuous spiking of neuron 0, or both neuron 0 and neuron 1, that corresponds to state transitions from S to hA, from hA to hAB, and from hABC to S. Finally, inhibitory connections from neurons 0 and 1 to the output neuron, together with an excitatory connection from neuron 2 to the output neuron, ensure that a spike in the output neuron is possible only after neuron 2 spikes (state hABC).

If we count the number of connections in the networks disregarding the associated weights (in other words, giving each one the same weight, 1), the 10 champion networks evolved with noise show significantly higher density (average 18.5, standard deviation 1.6) than the 10 champion

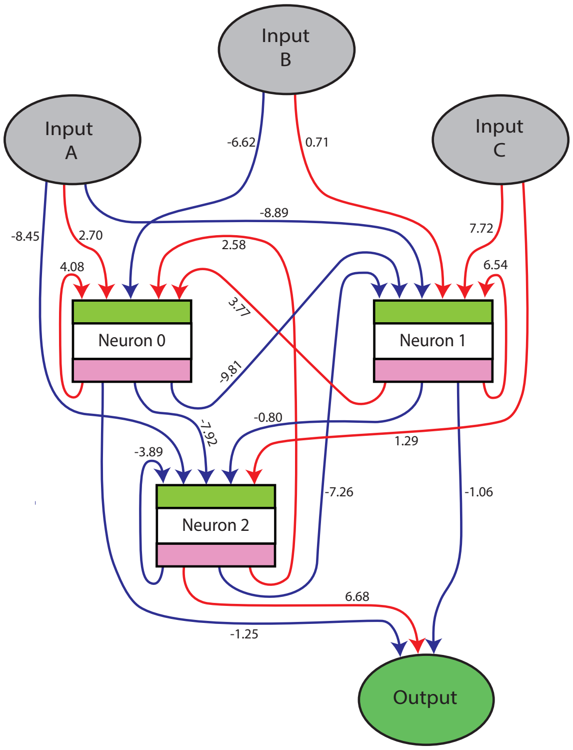

Figure 5: The topology of the network of the champion 3 evolved with noise. Blue lines are inhibitory connections, red lines are excitatory, numbers next to the lines show the weights.

networks evolved without noise (16.2, 2.1, respectively; $p = 0.01$, Mann-Whitney U test). Summing the absolute weight for all the edges gives an even more striking, more than 2-fold, difference (average for champions evolved with noise: 82.1, standard deviation: 13.8; without noise: 34.2, 11.8, respectively; $p = 0.0012$).

## Conclusions and future work

Our results show that evolving networks with noise leads to high robustness to modifying the neuronal parameters and variations of the input. In addition, the results show that networks evolved with noise are capable of maintaining their internal state infinitely. This memory seems to be kept thanks to excitatory self-loops, while switching state is possible thanks to inhibitory recurrent connections.

In future work, we plan to further investigate the robustness of the spiking neural networks evolved with noise to modifying parameters. The adaptive exponential model itself has in fact only four free parameters (Touboul and Brette, 2008), which are directly proportional to four bifurcation parameters in the model with the parametrization used here ($\tau_w$, $a$, $V_r$, and $b$; Naud et al., 2008). Similarly, we may expect that robustness to changes in synaptic gains ($gain_E$ or $gain_I$), $C$, $E_E$, and $E_I$ will be related, and networks robust to changing them will be expected to be robust to variation of synaptic weights. Therefore, it should be possible to change more than one parameter at a time, perhaps provided that certain relationships between them is maintained (for example, their quotient would have to stay within a certain range).

It will be also interesting if the networks are robust to changes of the parameters which actually result in the changed behavior of the neuron in terms of the response to constant input current. For example, we could evolve the networks with neurons showing tonic spiking, and then change parameters so that the behavior is tonic bursting or delayed spiking (Naud et al., 2008). Our preliminary results, incidentally, show that it is possible to evolve networks with neurons displaying all types of responses to the constant input current demonstrated for this model (Naud et al., 2008). It will be interesting to see if the networks evolved with noise in neurons showing other behavior than tonic spiking are equally robust to parameter changes.

We also plan to analyze robustness to introducing synaptic delays and absolute refractory periods. Such changes, similarly to changes in some neuronal parameters, could lead to lower firing rates in neurons whose continued spiking provides memory.

We would also like to know if the networks will be robust to changes in parameters affecting each neuron in the network independently (in this paper a particular change affected all neurons in the network in the same fashion). This last type of robustness is particularly interesting for transferring an evolved network able to perform a particular computation to analog neuromorphic hardware, in which setting particular (and the same) parameters for all the neurons in the network may be problematic.

Although the level of noise we have used here is biologically plausible, it is on a lower end of the spectrum observed for biological neurons (Paré et al., 1998; Destexhe and Paré, 1999; Anderson et al., 2000; Finn et al., 2007). It will be interesting to find out if higher, still biologically plausible, levels of noise (for example, noise as in this paper but with standard deviation of 4 mV) allow for evolvability, and if possibly result in more robustness to changes in neuronal parameters—and if lower levels of noise will result in lower robustness. Moreover, other models of noise (for example, an Ornstein-Uhlenbeck process; commonly used to model noise in neuroscience) on neuronal variables could be introduced—also, the presence of noise on input (such as variable signal or silence length). We wonder if different models of noise lead to similar levels of robustness to changes in neuronal parameters.

Another possible direction of future work could be evolution of networks for more complex tasks involving temporal pattern recognition (for example: more symbols, regular expressions, recognition of several patterns by the same network, with several output neurons). Our preliminary results indicate that our model scales up for more complex tasks. This direction of possible future research would allow us to find out what are the relationships between the complexity of the tasks and the size (and complexity) of the minimal-size spiking neural networks necessary to solve them.

## Acknowledgements

This work was supported by the Polish National Science Center (project EvoSN, UMO-2013/08/M/ST6/00922). MY acknowledges the support of the KNOW RNA Research Center in Poznan (No. 01/KNOW2/2014). We are grateful to Volker Steuber and Neil Davey for discussion and suggestions.

## References

Ahissar, E. and Arieli, A. (2001). Figuring space by time. *Neuron*, 32:185–201.

Anderson, J. S., Lampl, I., Gillespie, D. C., and Ferster, D. (2000). The contribution of noise to contrast invariance of orientation tuning in cat visual cortex. *Science*, 290:1968–1972.

Bialek, W., Rieke, F., van Steveninck, R. R. d. R., Warland, D., et al. (1989). Reading a neural code. In *Neural Information Processing Systems*, pages 36–43.

Decharms, R. C. and Zador, A. (2000). Neural representation and the cortical code. *Annual Review of Neuroscience*, 23:613–647.

Destexhe, A. and Paré, D. (1999). Impact of network activity on the integrative properties of neocortical pyramidal neurons in vivo. *Journal of Neurophysiology*, 81:1531–1547.

Finn, I. M., Priebe, N. J., and Ferster, D. (2007). The emergence of contrast-invariant orientation tuning in simple cells of cat visual cortex. *Neuron*, 54:137–152.

Florian, R. V. (2003). Biologically inspired neural networks for the control of embodied agents. *Center for Cognitive and Neural Studies (Cluj-Napoca, Romania), Technical Report Coneural-03-03*.

Gerstner, W., Kempter, R., van Hemmen, J. L., and Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383:76.

Huxter, J., Burgess, N., and O'Keefe, J. (2003). Independent rate and temporal coding in hippocampal pyramidal cells. *Nature*, 425:828–832.

Joachimczak, M. and Wróbel, B. (2012). Evolution of robustness to damage in artificial 3-dimensional development. *Biosystems*, 109:498 – 505.

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.

Natschläger, T. and Maass, W. (2002). Spiking neurons and the induction of finite state machines. *Theoretical Computer Science*, 287:251–265.

Naud, R., Marcille, N., Clopath, C., and Gerstner, W. (2008). Firing patterns in the adaptive exponential integrate-and-fire model. *Biological Cybernetics*, 99.

Paré, D., Shink, E., Gaudreau, H., Destexhe, A., and Lang, E. J. (1998). Impact of spontaneous synaptic activity on the resting properties of cat neocortical pyramidal neurons in vivo. *Journal of Neurophysiology*, 79:1450–1460.

Rieke, F. (1999). *Spikes: exploring the neural code*. MIT press.

Rutishauser, U. and Douglas, R. J. (2009). State-dependent computation using coupled recurrent networks. *Neural Computation*, 21:478–509.

Sipser, M. (1996). *Introduction to the Theory of Computation*, page 87. International Thomson Publishing, 1st edition.

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. (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. *Network: Computation in Neural Systems*, 17:173–191.

Steuber, V. and Willshaw, D. J. (1999). Adaptive leaky integrator models of cerebellar Purkinje cells can learn the clustering of temporal patterns. *Neurocomputing*, 26:271–276.

Tiňo, P. and Mills, A. (2005). *Learning Beyond Finite Memory in Recurrent Networks of Spiking Neurons*, pages 666–675. Springer Berlin Heidelberg, Berlin, Heidelberg.

Touboul, J. and Brette, R. (2008). Dynamics and bifurcations of the adaptive exponential integrate-and-fire model. *Biological Cybernetics*, 99:319.

Wróbel, B. (2016). Evolution of spiking neural networks robust to noise and damage for control of simple animats. In *Parallel Problem Solving from Nature – PPSN XIV*, pages 686–696.

Wróbel, B., Abdelmotaleb, A., and Joachimczak, M. (2012). Evolving networks processing signals with a mixed paradigm, inspired by gene regulatory networks and spiking neurons. In *BIONETICS*, volume 134 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 135–149. Springer.

Yaqoob, M. and Wróbel, B. (2017). Very small spiking neural networks evolved to recognize a pattern in a continuous input stream. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 3496–3503.