Spikes That Count:
Rethinking Spikiness In Neurally Embedded Systems

Keren Saggie[1], Alon Keinan[1], Eytan Ruppin[1,2,†]

[1]School of Computer Sciences, Tel-Aviv University, Tel-Aviv, Israel
{*keren,keinan*}*@cns.tau.ac.il*, *ruppin@post.tau.ac.il*
[2]School of Medicine, Tel-Aviv University, Tel-Aviv, Israel

Abstract

Spiky neural networks are widely used in neural modeling, due to their biological relevance and high computational power. In this paper we investigate the usage of spiking dynamics in embedded artificial neural networks, that serve as a control mechanism for evolved autonomous agents performing a delayed-response task. The synaptic weights and spiking dynamics are evolved using a genetic algorithm. We compare evolved spiky networks with evolved McCulloch-Pitts networks, while confronting new questions about the nature of "spikiness" and its contribution to the neurocontroller's processing. On the behavioral level, we show that in a memory-dependent task, network solutions that incorporate spiking dynamics can be less complex and easier to evolve than neurocontrollers involving McCulloch-Pitts neurons. On the functional level, we identify and rigorously characterize two distinct properties of spiking dynamics in embedded agents: spikiness evident influence and spikiness functional contribution.

1 Introduction

Models of spiking neurons have been extensively studied in the neuroscience literature, in recent years. Spiky networks have a greater computational power then networks of sigmoidal and McCulloch-Pitts neurons [7], and are able to model the ability of biological neurons to convey information by the exact timing of an individual pulse, and not only by the frequency of the pulses [3, 8]. In this paper we investigate the usage of spiking dynamics in embedded neurocontrollers, that serve as the control mechanism for Evolved Autonomous Agents (EAAs) performing a delayed-response task. The spiky neural networks are developed by a genetic algorithm [9] to maximize a behavioral performance measure, and their resulting networks and dynamics are subjected to further study. EAAs are a very promising model for studying neural processing due to their simplicity, and their emergent architecture [5, 11]. Investigating spiky neural networks in this framework raises new questions, that were not raised using pre-designed spiky models. For Example, evolutionary robotics studies have previously analyzed whether the spiking dynamics result in a time-dependent or a rate-dependent computation, and investigated the effect of noise on the emerging networks [4, 10].

We rigorously address the questions of **what is a "spiky" network, and how to define and measure the spikiness level of each neuron**, for the first time. We observe that a network with spiking neurons is not necessarily "spiky", in terms of integration of inputs over time. Following this observation, we present two new fundamental ways by which we define and quantify the spikiness level of a neuron. The study of spiking neural networks is performed within a delayed-response task, as memory is needed to solve such tasks and spiking dynamics may hence be useful. Delayed response tasks are characterized by a significant delay between the stimulus and the corresponding appropriate response, which make them impossible to solve by a simple sensory-motor mapping. The rest of this paper is organized as follows: Section 2 describes the network architecture and the evolutionary procedure. In section 3 we present two basic properties of spikiness in embedded agents. Section 4 analyzes the evolved neurocontrollers and their dynamics. These results and their implications are discussed in section 5.

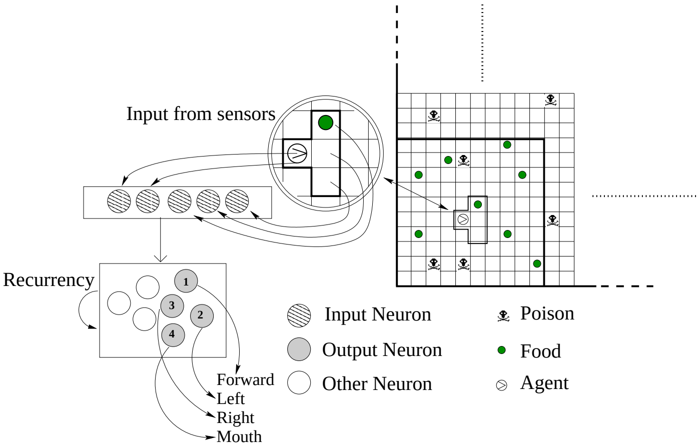

Figure 1: *The EAA environment.* An outline of the grid world and the agent's neurocontroller. The agent is marked by a small arrow on the grid, whose direction indicates its orientation. The curved lines indicate where in the arena each of the sensory inputs comes from.

2 The Model

2.1 The EAA Environment

The EAA environment is described in detail in [2]. The agents live in a discrete 2D grid "world" surrounded by walls. Poison items are scattered all around the world, while food items are scattered only in a "food zone" in one corner. The agent's goal is to find and eat as many food items as possible during its life, while avoiding the poison items. The fitness of the agent is proportional to the number of food items minus the number of poison items it consumes. The agent is equipped with a set of sensors, motors, and a fully recurrent neurocontroller of binary neurons.

Four sensors encode the presence of a resource (food or poison, without distinction between the two), a wall, or a vacancy in the cell the agent occupies and in the three cells directly in front of it (Figure 1A). A fifth sensor is a "smell" sensor which can differentiate between food and poison underneath the agent, but gives a random reading if the agent is in an empty cell. The four motor neurons dictate movement forward (neuron 1), a turn left (neuron 2) or right (neuron 3), and control the state of the mouth (open or closed, neuron 4).

In previous studies [2], eating occurs if the agent stands on a grid cell containing a resource for one step. Here, we have modified this task to include a delayed-response challenge: In order to eat food, the agent has to stand on a grid-cell containing a resource for a precise number of steps K, without moving or turning, and then consume it, by closing its mouth on the last waiting step. Eating after standing on a food item for more or less than K steps does not increase its fitness. Hence, in essence, the agent has to learn to count to K precisely. The agent's lifespan, defined by the number of sensorimotor steps available to it, is limited. Waiting steps are not counted as part of lifespan steps in order to facilitate the evolution of the delayed-response task.

2.2 The Neurocontrollers

All neurocontrollers are fully-recurrent with self-connections, containing 10 binary neurons (out of which 4 are motor neurons), and 5 sensor neurons that are connected to all network neurons. We compare between neurocontrollers with McCulloch-Pitts (MP) neurons, employed conventionally in most EAA studies, and ones with spiky *Integrate-And-Fire* neurons. In both types of networks, a neuron fires if its voltage exceeds a threshold. The spiking dynamics of an Integrate-And-Fire neuron i are defined by

$$V_i(t) = \lambda_{\mathbf{i}}(V_i(t-1) - V_{rest}) + V_{rest} + \frac{1}{N}\sum_{j=1}^{N} A_j(t)W(j,i), \qquad (1)$$

where $V_i(t)$ is the voltage of neuron i at time t, $\lambda_{\mathbf{i}}$ is a **memory factor** of neuron i (which stands for its membrane time-constant), $A_j(t)$ is the activation (firing) of neuron j at time t, $W(j,i)$ is the synaptic weight from neuron j to neuron i, N is the number of neurons including the input sensory neurons, and V_{rest} stands for the resting voltage (set to zero in all simulations). After firing, the voltage of a spiky neuron is reset to the resting voltage, with no refractory period.

The voltage of a spiky neuron results from an interplay between the history of its inputs and the current input field. The memory factor, which ranges between 0 and 1, determines the amount of integration over time that the neuron performs: The higher the memory factor, the more important is the neuron's history ($V_i(t-1)$), compared to the current input field (the last summand in Eq. (1)). The limit case of $\lambda_{\mathbf{i}}$=0 corresponds to a MP neuron, in which only the current input field determines the voltage. The memory factor is different for each neuron, as different neurons may have different roles, each demanding a different amount of integration over time. A **genetic algorithm** is used to evolve the synaptic weights $W(j,i)$ and, for spiky neurocontrollers, the memory factor parameters. Evolution is conducted over a population of 100 agents for 30000 generations, starting from random neurocontrollers, using a mutation rate of 0.2 and uniform point-crossover with rate of 0.35.

3 The Different Faces of "Spikiness"

We evolved agents that use spiking dynamics in order to successfully solve the delayed-response task. But are these agents really spiky, in the sense that they integrate their inputs over time? First, having encoded the neuronal memory factors in the genome gives rise to the possibility that the evolution will come out with non-spiky solutions. Second, **even if the memory factor is high (Eq. (1)), it does not ensure that the neuron indeed utilizes its "integration potential" in its firing**. For example, a neuron may receive a large excitatory input field in every time step and fire in a very high frequency, without performing any integration over its past input fields. That is, given its input field, its pattern of firing would be indistinguishable from a MP neuron. Essentially, we aim to distinguish between the observation that a given neuron has been assigned *spiking dynamics* by evolution, i.e. obtained a non-vanishing memory factor, and the true level of its *spikiness*, i.e., the amount by which it really "utilizes" its spiking dynamics. In this section we present two methods for measuring the spikiness level of a neuron, based on two fundamentally different perspectives.

Relevant Spikiness (RS). The first measurement for spikiness answers the following question: **how much are the spiking dynamics of a neuron needed for good performance of the agent?** If while abolishing the spiking dynamics of a neuron, the agent's performance deteriorates considerably, then its spiking dynamics contribute to the agent's behavior and one can claim that this neuron is really "spiky". If in contrast the fitness of the agent is maintained, clearly this neuron is not really "spiky".

To quantify this type of spikiness we use a lesioning paradigm [1], where numerous multiple lesions are afflicted upon a network, the corresponding performance score of the agent is recorded, and the causal importance (contribution) of each lesioned element to the behavioral task is assessed. We define a new lesioning method: The **λ-lesioning** method *lesions only the memory factor of the neuron*, leaving the rest of its dynamics unaltered. Lesioning the memory factor is done by clamping it to zero, which turns the neuron into an MP one. Fitness scores are measured for all multi-lesion configurations with the λ-lesioning method, and a Multi-lesion Shapley value Analysis (MSA) [6] is utilized to quantify **the contribution of the neuronal memory factor of each neuron to successful behavior**.

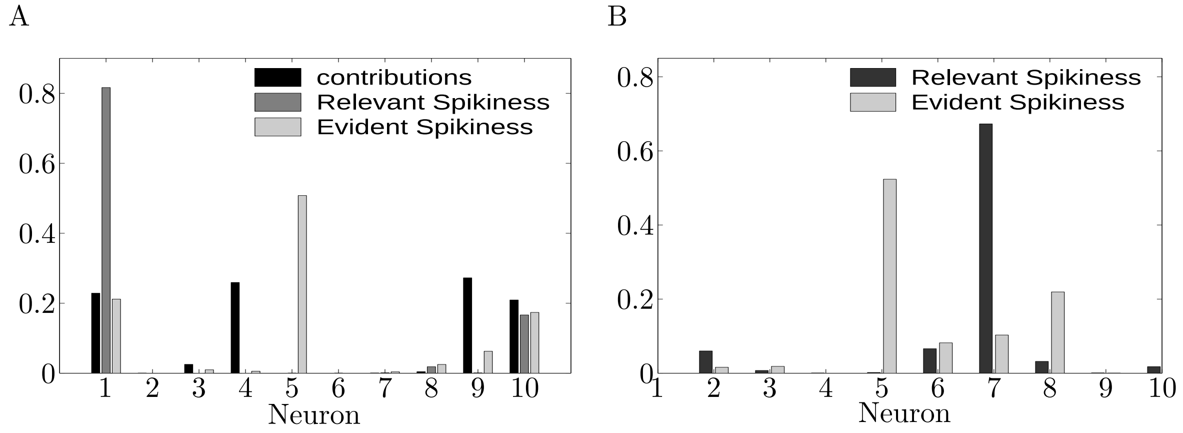

Figure 2: *Comparison between spikiness measurements.* (A). RS and ES scores for the neurons of agent S5, along with their general neuronal contribution values. (B). RS and ES scores for agent S7.

Evident Spikiness (ES). The second index of spikiness measures **how much do the spiking dynamics of a neuron influence its firing**: If the firing pattern of a neuron stays the same regardless of whether it possesses spiking dynamics or not, then we can consider it as non-spiky. The ES index is calculated by tracing the firing pattern of a spiky neuron and comparing its actual firing to that of a MP neuron receiving an identical current input field on each time step (last summand in Eq. (1)). We calculate the fraction of time steps in which there is a difference between the activations of the spiky neuron and a corresponding "benchmark" MP neuron, quantifying **the average percentage of lifetime steps in which the neuron's spiking dynamics "made a difference" in the firing of the neuron examined**. (A formal description is omitted due to space considerations). The RS and ES measures are normalized such that the sum over all neurons equals one.

4 Results

4.1 Performance Evaluation

Successful agents that solve the delayed-response task were evolved with both MP and spiky networks. The evolution of the delayed-response task is fairly difficult, and many evolutionary runs ended (i.e. the performance has converged) without yielding successful agents. We measure the difficulty of each task as the fraction of runs that ended successfully (Figure 3A). Evidently, the task is harder as the agent has to wait for a longer delay period. More important, successful spiky neurocontrollers evolve more easily than MP networks.

4.2 Spikiness Analysis

We examine the spikiness level of the neurocontrollers evolved with spiking neurons, focusing on two agents: S5 and S7, with a delay period of 5 and 7 time steps, respectively. For S5, Figure 2A compares the general contributions of the neurons (yielded by the MSA [6]) with the RS values, indicating how much the spikiness of each neuron contributes to successful behavior. Notably, neurons 1, 4, 9 and 10 contribute significantly to the agent's behavior, as shown by their general contribution MSA values, while the spikiness of only neurons 1 and 10 has a significant contribution, according to their RS values. Figure 2A also presents the ES values. Clearly, the two methods for measuring "spikiness" yield different results: Neuron 5 receives a very high ES score, and a near-zero RS score. A more pronounced difference is apparent in Figure 2B, which shows both measures for agent S7. In this case, the seventh neuron gets the highest RS value, but receives a

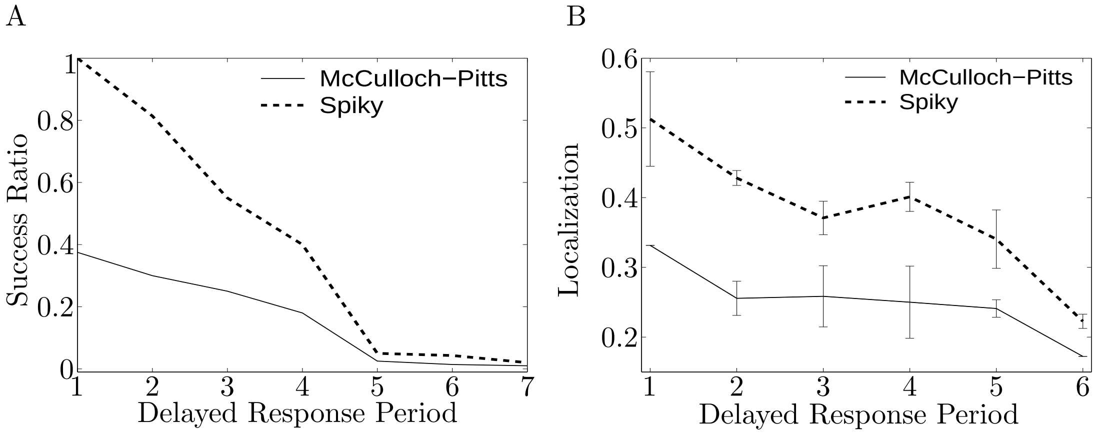

Figure 3: (A). Task difficulty vs. delayed response period: Success ratio for a task is the fraction of evolutionary runs, out of 30, in which the best agent achieved a behavioral fitness above a pre-defined threshold. (B). Localization score (mean and standard deviation across several successfully evolved agents) vs. delayed response period.

pretty low ES score. Neurons 5 and 8 are the most spiky ones according to the ES measure, but receive low RS values.

The difference between the results of the two spikiness measurements originates in their different nature: The RS method is a functional measurement, which considers as spiky only neurons that their spikiness contributes to behavior. Usually such neurons will also have high neuronal contributions, and will be generally important to behavior. The ES method measures the role that spiking dynamics themselves play in determining the firing of a neuron *irrespectively of that neuron's functional and behavioral role.* In the case of agent S7, further analysis of its behavior and activation patterns revealed that neuron 7's spikiness plays a pivotal role in the agent's counting ability. When this neuron is λ-lesioned, the agent cannot eat, explaining the very high RS value assigned to it. However, since the ratio of steps in which the spiking dynamics influence the activation of neuron 7 is only about 3% (the counting steps), this neuron receives a low ES score.

In almost all evolved spiky agents, most of the network's neurons have low spikiness functional contributions (RS). Usually, neurons with high RS scores are involved in the counting process, which utilizes the memory abilities of a spiky neuron. The two spikiness measurements show that the evolved agents are truly spiky, both in using past history to determine the activation patterns of some neurons, and in utilizing the spiking dynamics for successful behavior.

4.3 Delayed Response And Processing Localization

It is interesting to compare between the spiky and the non-spiky networks, in terms of localization of processing. The localization score [1] is a measure of how distributed is a given task (in our case, the counting task) in the network, the lower it is, the processing involved in counting is more evenly spread in the network across many neurons. We have already shown that the difficulty of evolving the counting-task increases with the delayed response period (section 4.1). Is there a correlation between the difficulty of evolving a network that solves a task, and the localization level of the resulting network? Figure 3B plots the average localization score as a function of the delayed response period. In both types of agents, the network's localization decreases with the length of the delayed response period, and MP agents are less localized than the spiky ones. Interestingly, the correlation coefficient between the average localization score and the task difficulty (the evolution success ratio) is high: 0.86 for the spiky networks, and 0.84 for the MP ones.

The lower localization levels observed in MP networks compared with the spiky ones result from the fact that in the MP networks, in order to count to K, the network has to pass through K distinct activation states. In a spiky network, the same network activation state can be repeated several times during the counting process, since the state of a neuron consists also of its accumulated voltage. Therefore, a spiky network can theoretically count with a single neuron, that accumulates voltage over $K-1$ steps, and fires on the Kth step. The evolved spiky networks do not posses such efficient counting, but usually a small number of neurons use their spikiness to accumulate voltage and "count" for a few steps, and as a result, less neurons are needed compared with MP networks. By using incremental evolution techniques we evolved agents with spiking dynamics that count up to delay periods of 35! Such agents utilize a very efficient counting method that involves only two spiky neurons with high functional (RS) contributions.

5 Discussion

The study of spiky neural networks in the context of embedded evolutionary agents brings forward basic questions regarding spiking dynamics that have not yet been raised. The simplicity and concreteness of EAA models makes them a promising model for computational neuroscience research, and specifically to study the spikiness properties of neurocontrollers. We have shown that the presence of evolved spiking dynamics does not necessarily transcribe to actual spikiness in the network, and that the spikiness level can be defined and quantified in several functionally different ways. Specifically, the spikiness functional contribution (RS) and the spikiness evident influence (ES) each point to different neurons in the neurocontrollers studied. On a behavioral level we have shown that in tasks possessing memory-dependent dynamics network solutions that involve spiking neurons can be less complex and easier to evolve, compared with MP networks.

References

[1] R. Aharonov, L. Segev, I. Meilijson, and E. Ruppin. Localization of function via lesion analysis. *Neural Computation*, 15(4), 2003.

[2] R. Aharonov-Barki, T. Beker, and E. Ruppin. Emergence of memory-driven command neurons in evolved artificial agents. *Neural Computation*, 13:691–716, 2001.

[3] G. Bugmann. Biologically plausible neural computation. *Biosystems 40, 11–19*, 1997.

[4] D. Floreano and C. Mattiussi. Evolution of spiking neural controllers for autonomous vision-based robots. *Evolutionary Robotics IV, Berlin, Springer-Verlag*, 2001.

[5] L.H. Hartwell, J.J Hopfield, S. Leibler, and A.W. Murray. From molecular to modular cell biology. *Nature*, 402(6761):C47–C52, 1999.

[6] A. Keinan, C. C. Hilgetag, I. Meilijson, and E.Ruppin. Fair attribution of contribution: Shapley value analysis of neurocontrollers. *preprint*, 2003.

[7] W. Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks 10, 1656-1671*, 1997.

[8] W. Maass and B. Ruf. On computation with pulses. *Information and Computation 148(2), 202–218*, 1999.

[9] M. Mitchell. *An introduction to genetic algorithms*. MIT Press, Cambridge, Massachusetts, 1996.

[10] E. A. Di Paolo. Spike-timing dependent plasticity for evolved robot control: neural noise, synchronization and robustness. *To appear in Adaptive Behavior*, 10(3/4), 2003.

[11] E. Ruppin. Evolutionary autonomous agents: A neuroscience perspective. *Nature Reviews Neuroscience*, 3:132–141, 2002.