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 counting 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. We show that in a memory-dependent task, network solutions that incorporate spiking dynamics can be less complex and easier to evolve than networks involving McCulloch-Pitts neurons. We identify and rigorously characterize two distinct properties of spiking dynamics in embedded agents: spikiness dynamic influence and spikiness functional contribution.

Keywords: evolutionary computation, spiking, counting, neurocontroller analysis

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 [8], 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, 9]. 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 counting task. The spiky neural networks are developed by a genetic algorithm [10] 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, 12]. 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, 11].

We rigorously address the questions of **what is a "spiky" network, and how to define and measure the spikiness level of each neuron**. We observe that a network with spiking neurons is not necessarily "spiky", in terms of integration of inputs over time, and of the spikiness functional contribution. 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 counting task, as this task requires memory, and cannot be solved 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.

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).

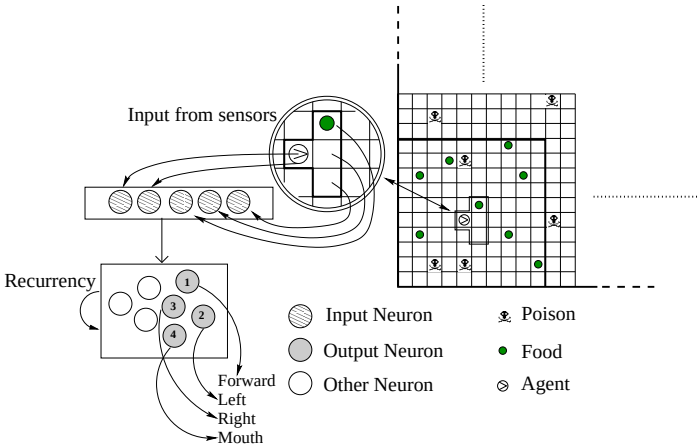

Figure 1: *The EAA environment.* An outline of the grid world and the agent's neurocontroller. The agent is marked by a small arrow on the grid, whose direction indicates its orientation. The curved lines indicate where in the arena each of the sensory inputs comes from.

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, 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. Closing its mouth 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 counting 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

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

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 counting 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, 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. Third, *even if the spikiness is utilized for firing, it does not necessarily contribute to the agent's performance*. 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.

Spiking Dynamic Factor (SDF). The first 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 SDF

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)). The fraction of time steps in which there is a difference between the activations of the spiky neuron and the corresponding "benchmark" MP neuron quantifies **the average percentage of lifetime steps in which the neuron's spiking dynamics "made a difference" in the firing of the neuron**.

Spikiness Relevance (SR). The second measurement for spikiness answers the following question: *how much are the spiking dynamics of a neuron needed for good performance of the agent?* Intuitively, 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. If, in contrast, the fitness of the agent is maintained, this neuron's spiking dynamics are functionally insignificant.

To quantify this type of spikiness we *lesion the memory factors of neurons* by clamping them to zero (and in fact turning them into MP neurons), leaving the rest of their dynamics unaltered. Fitness scores are measured when clamping together the memory factors of each subgroup of neurons. Based on these data, the Multi-perturbation Shapley value Analysis (MSA) [6, 7] determines **the causal importance of the spiking dynamics of each neuron to successful behavior**. The SDF and SR measures are normalized such that the sum over all neurons equals one.

4 Results

4.1 Performance Evaluation

Successful agents that solve the counting task were evolved with both MP and spiky networks. The evolution of the counting 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 average fitness score of the best evolutionary agent (Figure 3A). Evidently, the task is harder as the agent has to wait for a longer waiting 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 waiting period of 5 and 7 time steps, respectively. For S5, Figure 2A

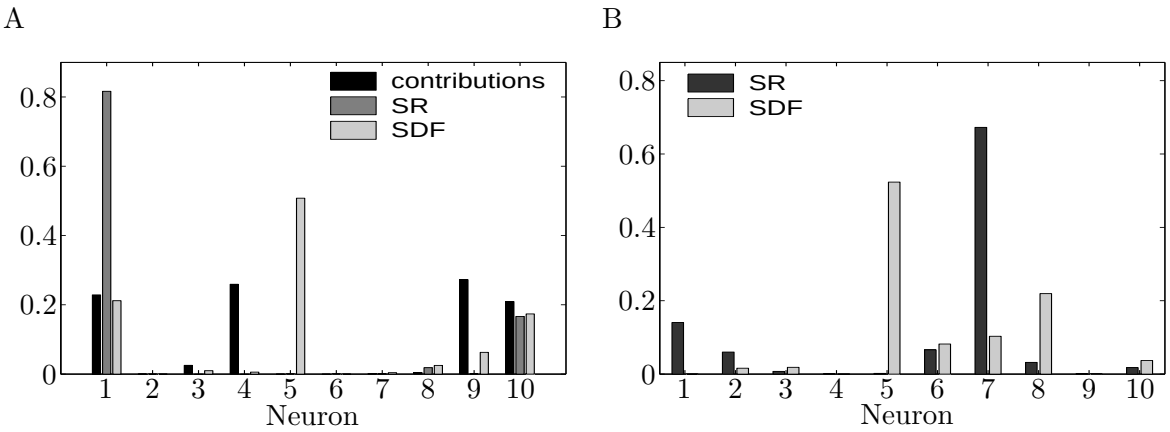

Figure 2: *Comparison between spikiness measurements.* (A). SR and SDF scores for the neurons of agent S5, along with their general neuronal contributions. (B). SR and SDF scores for agent S7.

compares the SR values of the different neurons with the contributions of the neurons yielded by the MSA based on all possible multi-lesions of subgroups of neurons. These contributions quantify how much each neuron contributes to successful behavior [6, 7]. Notably, neurons 1, 4, 9 and 10 contribute significantly to the agent's behavior, as shown by their general MSA contributions, while the spikiness of only neurons 1 and 10 has a significant contribution, according to their SR values. Figure 2A also presents the SDF values. Clearly, the two methods for measuring "spikiness" yield different results: Neuron 5 receives a very high SDF score, and a near-zero SR 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 SR value, but receives a pretty low SDF score. Neurons 5 and 8 are the most spiky ones according to the SDF measure, but receive low SR values.

The difference between the results of the two spikiness measurements originates in their different nature: The SDF method measures the role that spiking dynamics play in determining the firing of a neuron *irrespectively of that neuron's functional and behavioral role.* The SR method is a functional measurement, which considers as spiky only neurons whose spikiness contributes to behavior. Usually such neurons will also have high neuronal contributions, and will be generally important to behavior. Additionally, since influencing the firing pattern of the neuron is a prerequisite of having a behavioral contribution, these neurons must have a non-zero SDF score. In the case of agent S7, further analysis of its behavior and activation patterns revealed that the spikiness of neuron 7 plays a pivotal role in the agent's counting ability. When the memory-factor of this neuron is clamped, the agent cannot eat, explaining its very high SR value. However, since the

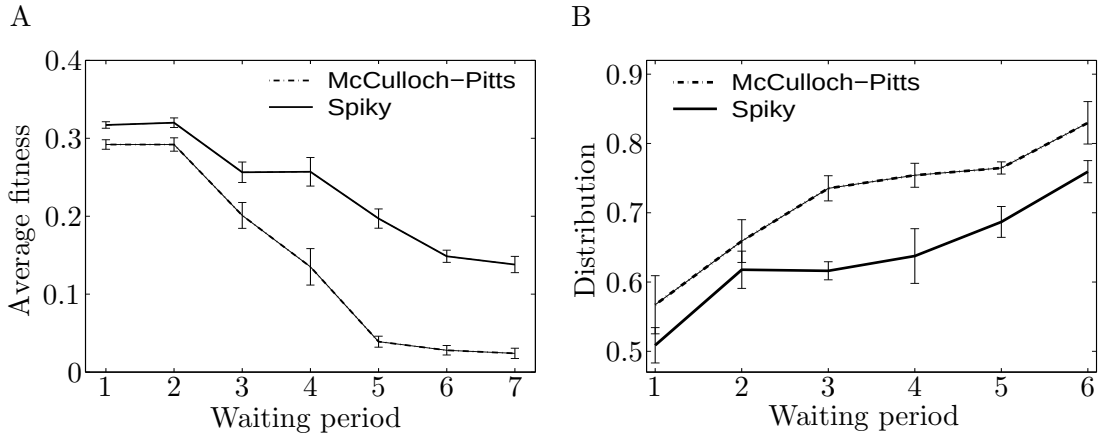

Figure 3: (A). Average fitness vs. waiting period: The fitness score of the best evolutionary agent averaged over 30 evolutionary runs, along with the standard deviation of the mean. (B). Distibution index (mean and standard deviation of the mean across several successfully evolved agents) vs. waiting period.

fraction 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 SDF score.

In almost all evolved spiky agents, most of the network's neurons have low spikiness functional contributions (SR). Usually, neurons with high SR 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 Counting Analysis and Processing Distribution

It is interesting to compare between the spiky and the non-spiky networks, in terms of distribution of processing. The distribution index [1] measures how distributed is a given task in the network, the higher it is, the more the processing is evenly spread in the network across many neurons. We have already shown that the difficulty of evolving the counting-task increases with the waiting period (section 4.1). Is there a correlation between the difficulty of evolving a network that solves a task, and the distribution level of the resulting network? Figure 3B plots the average distribution score as a function of the waiting period. In both types of agents, the network's distribution increases with the length of the waiting period, and MP agents are more distributed than the spiky ones. Interestingly, the correlation coefficient between the distribution index and the average fitness score is high: 0.8 for the spiky networks, and 0.9 for the MP ones.

The higher distribution levels observed in MP networks compared with the spiky ones result

from a different counting network processing: In order to count to K, an MP 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 waiting periods of 35! Such agents utilize a very efficient counting method that involves only two spiky neurons with high functional (SR) contributions.

5 Discussion

The simplicity and concreteness of EAA models makes them a promising framework for computational neuroscience research. 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. Apparently, the presence of evolved spiking dynamics does not necessarily transcribe to actual spikiness in the network. We have presented two ways by which the spikiness level of each neuron can be defined and quantified. Specifically, the spikiness dynamic influence (SDF) and the spikiness functional contribution (SR) each point to different neurons in the neurocontrollers studied. 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.

Acknowledgments

We acknowledge the valuable contributions of Hezi Avraham and Shay Cohen. This research was supported by the Adams Super Center for Brain Studies in Tel-Aviv University and by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities.

References

[1] R. Aharonov, L. Segev, I. Meilijson, and E. Ruppin. Localization of function via lesion analysis. *Neural Computation*, 15(4):885–913, 2003.

[2] R. Aharonov-Barki, T. Beker, and E. Ruppin. Emergence of memory-driven command neurons in evolved artificial agents. *Neural Computation*, 13(3):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. Hilgetag, I. Meilijson, and E. Ruppin. Causal localization of neural function: The shapley value method. In *Twelfth Annual Computational Neuroscience Meeting (CNS*2003)*, 2003.

[7] A. Keinan, C. C. Hilgetag, I. Meilijson, and E. Ruppin. Fair attribution of functional contribution in artificial and biological networks. *Preprint,* `http://www.cns.tau.ac.il/~keinan`.

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

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

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

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

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

Keren Saggie learned her undergraduate studies in the Interdisciplinary Program for Fostering Excellence of Tel-Aviv University, during which, through an individually designed interdisciplinary curriculum, acquired knowledge in computational neuroscience. She is currently an M.Sc. student at the school of computer science of Tel-Aviv University, under the guidance of Prof. Eytan Ruppin.

Alon Keinan earned his B.Sc. degree *summa cum laude* in computer science, statistics and operations research from Tel Aviv University in 1997. He is currently a Ph.D. student at the school of computer science, Tel Aviv University, working in the computational neuroscience lab, under the guidance of Eytan Ruppin and Isaac Meilijson.