An Evolutionary Autonomous Agent with Visual Cortex and Recurrent Spiking Columnar Neural Network

Rich Drewes[1], James Maciokas[1], Sushil J. Louis[2], and Philip Goodman[1]

[1] Brain Computation Laboratory
Biomedical Engineering Program
University of Nevada, Reno NV 89557, USA
drewes@cs.unr.edu
http://brain.cs.unr.edu
[2] Evolutionary Computing Systems Lab
Department of Computer Science
http://ecsl.cs.unr.edu
University of Nevada, Reno NV 89557, USA

Abstract. Spiking neural networks are computationally more powerful than conventional artificial neural networks. Although this fact should make them especially desirable for use in evolutionary autonomous agent research, several factors have limited their application. This work demonstrates an evolutionary agent with a sizeable recurrent spiking neural network containing a biologically motivated columnar visual cortex. This model is instantiated in spiking neural network simulation software and challenged with a dynamic image recognition and memory task. Since the resulting agent or "virtual organism" initially contains many randomly and recurrently connected neurons, we use a genetic algorithm to evolve generations of this brain model that instinctively perform progressively better on the task. Our investigations lay the foundation for further experiments to resolve the question of whether autonomous agents with spiking neural networks can take advantage of the proven theoretical computational superiority afforded by spiking networks to outperform their conventional, non-spiking counterparts for dynamic cognitive tasks.

1 Introduction and Prior Work

Ruppin's recent review of evolutionary autonomous agent (EAA) research [1] reveals that practically all such work to date has involved agents composed of artificial neural networks (ANNs). The class of simulated neural networks called ANNs includes both feedforward and recurrent networks built from threshold or sigmoidal gates, such as McCulloch-Pitts networks [2], single and multilayer perceptron networks [3, 4], Hopfield networks [5, 6], and many others. All these models share the common feature that they do not model the timing of individual spike events. In contrast, spiking neural networks models (SNNs) do explicitly model the timing of spike events. SNNs are of course also "artificial" in the sense that they are simulated and are not really living things, and some researchers do classify SNNs as a subcategory of ANNs. However, in this paper we will use the term ANN to refer exclusively to simulated *non-spiking* neural networks, and SNN to refer to simulated *spiking* neural networks.

Maass [7] defines a taxonomy of three generations of neural networks. The first two of these generations map to different types of non-spiking ANNs which comprise the bulk of computer science research in neural networks, including nearly all EAA work. Maass' third generation is a certain subset of integrate-and-fire spiking neural network. Maass demonstrates that this third generation of spiking networks is strictly computationally more powerful than the first two generations of non-spiking networks [7].

Despite the fact that spiking neural networks are known to be more powerful than ANNs, most computer science and EAA research in neural networks continues to be conducted with ANNs. We think there are several reasons for this. First, the realization that spiking networks are more powerful than ANNs is relatively recent. Second, ANNs are easier to implement computationally, and they are much faster to run in simulation. Thus there is a tradeoff between the potential computational advantages of SNNs versus the cost of implementing and running them. Third, ANNs have proven to be useful and interesting and powerful in themselves, and they still provide rich topics for investigation. Fourth, demonstrations of the computational superiority of spiking networks over ANNs are currently mostly mathematical exercises with little proven practical consequence. The role of spike coding in biological processing remains tantalizingly obscure. Another way of saying this is that nobody yet knows a way to implement a practical image recognizer using SNNs that takes advantage of the theoretical computational superiority of spiking networks to result in a smaller, more accurate, more robust, or faster network than the comparable ANN. Finally, many central challenges of ANNs (including difficult network design and training problems, particularly for recurrent networks) are equally vexing with spiking networks, so there are no apparent simplifications to offset the other difficulties inherent in working with spiking networks. And yet the most powerful information processing neural networks available—human brains—are undeniably recurrent spiking neural networks. Although there is no conclusive proof that the spiking aspect of biological neural networks translates into their real-world computational advantages, this hypothesis is the underlying motivation of our own work, and this paper is a step toward exploring this idea.

The current limited state of knowledge of how complex recurrent neural networks represent and process information makes them well suited for exploration with the EAA approach. That is, if thinking machines are too complex to design explicitly, then perhaps we can *evolve them* by computer. And if we cannot understand how these machines represent and process information internally, we can at least treat them as *agents* and learn about their internals indirectly by conducting controlled experiments that lead to changes in behavioral capabilities.

The evolutionary design of neural networks (EDNN) is worth reviewing briefly as a general topic, not just in the context of autonomous agents. Designing an ANN for a particular problem is a complex nonlinear search task. How many hidden nodes should be present? What input features should be selected? What values should the connection weights take? Certain search strategies work satisfactorily for certain aspects of this problem, such as back-propagation for training connection weights in feedforward ANNs with certain smooth transfer functions. However, a more general approach to ANN design, and SNN design, is to use evolutionary strategies such as genetic algorithms for the search. A good survey of the state of EDNN research is given in [8] and more recently [9]. Both authors create taxonomies according to what features of the NN are being evolved: connection weights, architecture, or both. Subclasses are also defined for recurrent and feedforward network evolutions. However, both of these surveys are restricted to evolutionary design of ANNs only—there is no treatment of evolutionary design of SNNs specifically. This was not merely a question of the authors' selection of focus for their papers; there is very little work on this topic at all. Exploring this area is another goal of our work.

2 Method

The next sections describe an evolutionary autonomous agent experiment designed to provide a framework for exploring the computational power of certain biological features, such as spiking networks and columnar organization, in dynamic cognitive tasks.

2.1 The Delayed Matching Task

Because our long term goal is to replicate features of biological cognition, the task we have chosen to challenge our neural agents is modeled after a psychological recognition and memory test rather than a more familiar ANN mapping task such as image or pattern recognition. We considered such a mapping recognition task to be a poor candidate for our goal of advancing understanding of cognition for several reasons. First, these tasks are not *dynamic*. By this we mean that there is no time constraint on the response. The input is presented, and when the network settles, the answer is available at an output node. Indeed, conventional ANNs have no notion of time implicit in them at all. (A number of extensions to ANN models have imposed sequenced processing to permit recurrence, for example [10], but these are still distinct from the actual notion of *time*.) In contrast, biological organisms make decisions in real-time in response to the environment. Second, ANNs in conventional mapping recognition tasks perform well even though they are strictly feedforward; nothing is gained for them in simple recognition tasks by being recurrent (and in fact making them recurrent would disrupt most training paradigms). Real biological neural networks are highly recurrent. To require agents to evolve functional recurrent networks, we picked a task that requires transient memory. Recognition alone is a valuable skill for an organism to have, no doubt; but combining recognition with short-term memory is certainly essential for any truly *cognitive* system.

Many explorations of human and animal visual working memory involve a delayed matching task [11]. In our variant of the delayed matching task, an image is presented to the test subject momentarily and then removed. A short time later, a second image is presented and the test subject must decide if the image was the *same* as or *different* than the first image. The test subject is not being asked to identify the images, or categorize them, but merely to determine if the first image presented matches the last one presented. To make the test more difficult, a "distractor" image can be interposed between the first and second images. The complete set of inputs and correct outputs is shown in table 1. From the table it is apparent that this task can be viewed as a temporal exclusive-or problem.

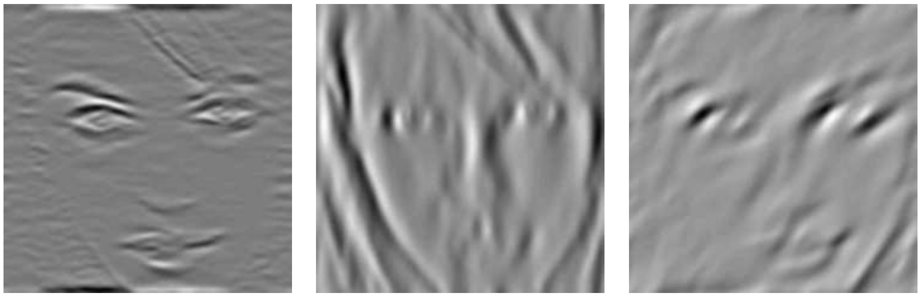
Table 1. Four possible input sequences for the delayed matching task

Input Sequence Number	First Image	Second Image	Third Image	Correct Output
1	Face	Noise	Face	"Same"
2	Face	Noise	Tree	"Different"
3	Tree	Noise	Face	"Different"
4	Tree	Noise	Tree	"Same"

Each agent is presented the four input sequences as shown, one sequence per simulation run, and that agent's reproductive success is determined by whether it answers with the correct "same" or "different" response shown. The preparation of the spiking input signals from these input image sequences is the topic of the next section.

2.2 Preparation of Input Stimuli

We selected an image of a human face and an image of a tree to be the two test images for this experiment. These are natural images (as opposed to synthetic images like line drawings), and visually distinct. The images were pre-processed as follows to get them into a form accessible to the agent neural network models. First, the face and tree images were equalized to help ensure that the images could not easily be distinguished by the networks based on spectral energy alone. The images were then whitened to simulate the effect that retinal ganglion cells and the lateral geniculate nucleus have on visual signals on the way to visual cortex [12]. Spatial Gabor filtering is then performed to model current theory of early visual processing [13]. The filtering is done at three orientations (horizontal, vertical, and diagonal) and two scales (small and large). At this point in the processing, representative images appear as depicted in figure 1. Each small region of the filter outputs is then converted into a variable frequency spike train. An entire input stimulus sequence consists of the concatenation of these sets of spike trains. These spike trains are stored in data files for insertion, during simulation, into the columns of the input area of the brain model as described in section 2.3. Note that all work up to this point are performed algorithmically in a preprocessing step, not via the agents' spiking neural networks during the actual agent simulation. This preprocessing step reduces the simulation time for the agents and frees up computational resources for the later stages.



(a) Horizontal orientation, small filter scale

(b) Vertical orientation, large filter scale

(c) Diagonal orientation, large filter scale

Fig. 1. The face image shown at three representative Gabor filter scales and orientations

The entire input sequence lasts 300 ms of simulation time. At time 100 ms, the first test image is presented to the agent for a duration of 100 ms. This is the face image for sequences 1 and 2, and the tree image for sequences 3 and 4. At 200 ms, another distractor image is presented for a duration of 100 ms. At 300 ms, the second test image is presented for a duration of 100 ms. This is the face image for sequences 1 and 3, and the tree image for sequences 2 and 4. For the time period 300 ms to 400 ms, the spike outputs of the model's two motor areas (the "same" area and the "different" area) are recorded. These are used to determine the model's fitness, as described in section 2.5.

2.3 Brain Model

The neural network models in our autonomous agents are designed to be like their biological counterparts, to the extent of current knowledge and given computational constraints. Our assumption is that many or most of the biological structures we are emulating exist because they are good at what they do—even if we do not yet understand how it is they do it. It is certainly also possible that some biological aspects we are emulating are merely incidental features that do not contribute to function; which features fall in which category is a question we hope to explore in future experiments.

For these experiments, the agents are recurrent column-structured SNNs with about 14000 total neurons and about ten times that many synapses. The ratio of synapses per cell in our model is considerably lower than in typical real biological networks. Conductivity constants have been adjusted upward to compensate; the resulting network behaves in a fairly realistic fashion in isolated tests, but almost certainly our network lacks the memory capacity of a more connected network. It may also be less capable in other respects.

Mammalian neocortex is, roughly, a 2mm thick multi-layered sheet of neurons that is crumpled and folded around the rest of the brain [14, 15]. Many investigators view neocortex as a six layer structure, though the physical reality is not quite so tidy. Different layers of neocortex do seem to have different characteristic patterns of connections, with, for example one layer having the bulk of input connections from sensory organs, another layer having the bulk of output connections to other cortical areas, and other layers having interconnects with neighboring cells of characteristic proximity. Our agents' neural networks also share this layered structure, which we simplify to three functional layers (roughly: input, output, and interconnect). In portions of mammalian cortex, particularly visual cortex, groups of neighboring cells share certain patterns of increased interconnectivity. These groups of more interconnected cells are called cortical columns [16]. Groups of these structures seem to combine into even larger structures called hypercolumns with a radius of up to about 1mm.

The larger scale organization and function of mammalian visual cortex remains only partially understood, but a great deal has been learned about the early stages [17]. We have attempted to incorporate some of this knowledge into our model. Once in the brain, visual processing appears to be accomplished roughly in stages, with some feedback from latter stages to earlier stages. Further processing takes place in the brain with each stage occurring in a certain portion of the visual cortex called a visual area. Conventionally, areas of the visual system are numbered V1, V2, and so on. In our model, the spike trains from the preprocessed input sequence are presented to hypercolumns organized in a 5 by 5 array. This is the agent's simplified V1, the first visual area of cortex. Each portion of the input visual field is mapped to one of these 25 hypercolumns. Within each hypercolumn is a further subarray of 2 by 3 columns. Each of these columns receives input from a subportion of the Gabor filtered image at one filter scale and filter orientation. This models the known selectivity of biological visual cortex [18]. From the model's V1 area, connections are made to a modeled visual assocation area called VA. VA is also organized as an array of hypercolumns, each containing a subarray of columns. Connections are made from V1 to VA in an expanding pattern inspired by biological data that allows the model to correlate features detected at different orientations and scales. In real visual cortex further visual processing areas follow, but limited knowledge about how those later areas function makes further modeling difficult.

The agent's modeled VA area connects back to itself to provide the recurrence believed necessary for memory and dynamic behavior. The VA area in the model also contains feedforward connections to the two "motor" output areas, one for the "same" result and one for the "different" result. Why these areas are called motor areas, and how they are used in the model, is conceptually an important part of our experimental design. We acknowledge the very limited state of knowledge of how spike coding is used for information processing in the brain. Nevertheless, it is possible to interpret the end result of an organism's information processing simply by *looking at how it acts*; that is, by looking at its rate-coded motor output, where differential activation of one motor area results in, for example, a forward rather than backward movement of a limb. At the end of a simulation run, the number of spikes in the agent's two motor areas are tallied and compared. The output motor area with the larger number of spikes is considered the agent's response.

All synaptic connections in the model have their initial strength (technically, their utilization of synaptic efficacy, or USE value [19, 20]) copied from the chromosome to that model's input file before the agent is simulated. These USE values roughly correspond to the synaptic weight in ANN networks. For the experiments described in this paper, there is no architectural variation in the models from generation to generation (though that is planned for future experiments). However, all synaptic connections were modifiable, so certain sub-regions could become better connected or even selectively disconnected according to the synapses that reached them. In this sense, some limited architectural variation is actually possible, without the complication of variable chromosome size.

Connection strengths change from these starting USE values during simulation of the brain through the modeled processes of biological synaptic facilitation and depression. In this work, no long-term changes in the base USE values (Hebbian learning) take place. (The neural network simulator supports Hebbian learning, but this feature is disabled for the current experiments.) An individual agent is not being trained within its lifetime on the recognition task. All training is done on the evolutionary timescale through crossover and mutation, and any improvements in ability to perform on the delayed matching task happen from generation to generation, not within a generation. Thus, this experiment is seeking to breed an "instinct" to perform well on the delayed matching task, not an organism capable of learning to improve its performance on the delayed matching task from trial to trial. Each organism gets one and only one chance to perform the delayed matching task in its lifetime. Later experiments may focus on the issue of learning within a generation.

2.4 Chromosome

Each USE value is encoded in the chromosome as a two bit binary number, allowing for initial connection strengths with three intermediate values plus zero (disconnected). The chromosome encoded 10824 USE strengths for these experiments. Note that because of technical limitations of the NCS input file format, each USE value actually specifies the connection strength of a small *group* of related probabilistic synaptic connections from one cell group to another cell group, rather than specifying the strength of an individual synapse. Our experimental setup also readily supports chromosomal storage of other values that affect the simulation, for example global conductivity values. The experiments reported here have those values fixed, however.

2.5 Fitness Function

The goal of the experiment is to evolve agents that can get the answer to all four input sequences correct on the delayed matching task. The design of the GA fitness function to accomplish that end is complicated by the discrete nature of the task: an agent could get no answers correct out of the four input sequences, one correct, two, three, or all four. A fitness function with only five steps would not provide much guidance to an organism to help it improve its performance to the next number of correct responses. To be most effective, the fitness landscape for genetic search should be smooth.

The output of the agent consists of spikes in the "same" and "different" motor areas. Whichever area contains the most output spikes during the result period (300-400 ms) is that agent's response to that input sequence. The number of spikes in each output area during this period varies from about 10 to 1300 for each group of output cells. This high degree of variability could provide a smoother range of potential fitness values if we consider the agent's response not merely to be correct or incorrect, depending on which output has more spikes, but instead to be the ratio of the number of spikes found on the cells in the correct output area to the number of spikes in the incorrect output area during the measurement period. Call this quantity the recognition ratio f. There are four such responses, one for each input sequence, which we call f_1, f_2, f_3, and f_4. Naively, it seems that a good continuous fitness function might simply be given by the sum of all f_n; however, this is not the case. Randomly initialized models invariably favor one of the two outputs over the other because of stronger random synaptic connections to that area. This results in the agent initially answering always "same" or always "different", and getting two answers right and two wrong. Thus two f_n will be large, and two small. During genetic search, the simplest tactic for an agent to increase fitness using this naive fitness function is to further strengthen the synaptic connections to the favored areas, and decrease connections to the other area. This results in two even larger f_n and two very small f_n, and fairly high fitness, even though the agent is only getting two of the four responses correct. This naive fitness function is unacceptable.

Because of the random bias that ends up favoring one of the outputs over the other in initial, randomly created agents, it is actually quite difficult to develop an organism that is capable of getting more than two answers correct. In order to perform better than this, agents must evolve to have less of a gross bias toward one output region or the other so that selected features in the input can be detected and converted into enough output spikes in the *correct* output region to offset any remaining gross bias toward the other region.

To prevent extremely large ratios on two of the input sequences from overwhelming the overall fitness, g_n is defined as follows:

$$g_n = f_n - (f_n - 1.0)^2 \tag{1}$$

These g_n have a maximum when the f_n ratio is 1.5; thus, increasing ratios are only rewarded to a point. Furthermore, since the slope of g_n is steeper in the neighborhood of 1.0 for values of f_n less than 1.0 than for values greater than 1.0, an agent's fitness increases more by moving slightly closer to one more correct answer than it does for the agent to get an existing correct answer "more" correct (that is, for further increasing its f_n ratio above 1.0 for that n). It is desirable for the recognition ratios f_n to increase above 1.0 if possible, because that makes the output less ambiguous; however, it is more important to first get another answer *correct* by bringing the f_n above 1.0 for any remaining incorrect responses. Finally, to give agents with more answers correct an extra fitness advantage, the final fitness value is computed by adding together all g_n and multiplying by the fraction of correct answers.

The overriding design goal of the fitness function is to reward individuals who get more responses correct (have the most recognition ratios greater than 1.0). Among those individuals with the same number of responses correct, somewhat greater fitness should be given to those who discriminate better among the two responses; that is, those individuals with higher recognition ratios f_n.

2.6 Genetic Algorithm

The genetic algorithm used in this experiment is a generic implementation with roulette selection. The mutation rate is one per hundred USE elements per generation. The crossover probability is 0.66 (there is a one-third chance the parents would be passed to the next generation with only mutation and not crossover). Crossover is multipoint and variable, with between 1 and 100 points per offspring. Population size is 16, and elitist selection preserves the four most fit individuals from each generation unchanged.

2.7 NCS: The NeoCortical Simulator

Once the brain models are built, they are run in the neocortical simulation software package NCS [21] on a parallel cluster computer. NCS simulates networks of tens of thousands of neurons and millions of synapses at the level of templated spikes with modeled membrane channels and voltages and synaptic facilitation and depression and Hebbian learning. NCS is an open-source C language application written using the MPI multi processing library running on a commodity GNU/Linux cluster.

3 Results and Discussion

At the outset it was far from clear to us whether any agents would arise that could get all four answers correct in the delayed matching task. For a time it appeared none would. After considerable experimentation we eventually found a single agent that managed the feat after 800 lengthy generations of evolution. Once this was achieved, further adjustments in the model architecture and parameters—by and large consisting of simplifications—led comparatively quickly to the much more adept agents described in this paper. These more compelling current results are shown in figure 2. Any fitness value over 16 indicates that the agent got answers to all four input stimulus sequences correct with a recognition ratio greater than 1.0. A fitness of exactly 25 is the best possible, indicating the maximum ratio of spikes in the correct output region to spikes in the incorrect output region. During one of the runs, depicted in figure 3, an agent surprisingly achieved very close to the maximum fitness in only 60 generations.

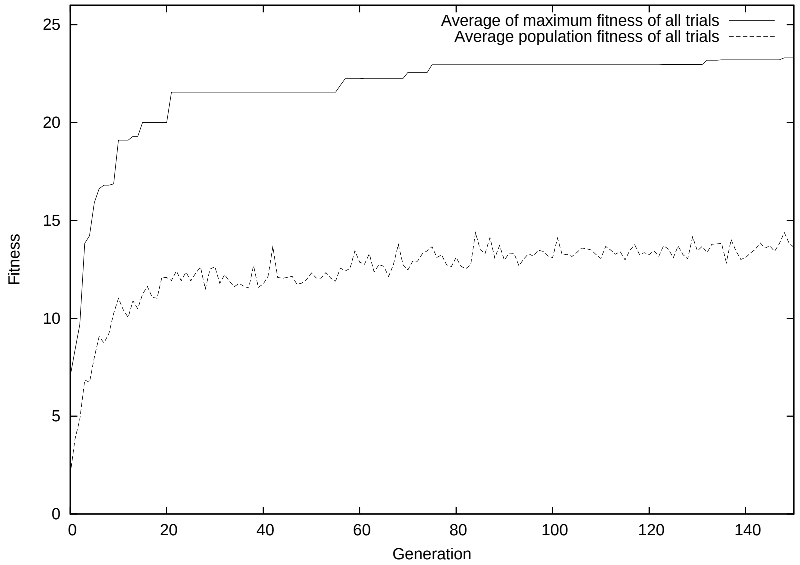


Fig. 2. Average of maximum fitness, and average of population fitness, over five trials. Population size: 16

It is interesting to reconsider this experiment from the point of view of one of the evolved agents. The agent begins its existence at time 0. The environment is dark, and there is only transient spiking activity for the first 100 ms. At 100 ms, the agent experiences the beginnings of spikes in the input columns of its visual cortex. These spikes represent the first image in the input sequence. Using its evolved instinct, the agent impresses some representation of the image in its recurrent memory. At 200 ms, the incoming image changes to white noise. The agent "knows" in some sense that this is just a distraction. Somehow, in the face of the incoming noise, the agent keeps some representation of the *first* image alive in the recurrent spiking network of its visual association area. After 100 ms of this noise, the spikes from the third and final image are presented to the agent's visual cortex. The agent instinctively compares this new input in some way to the memory of the first image it has retained in order to determine if the two images are the same or different. If they are the same, the agent triggers increased spiking activity in the first motor output (it moves left, perhaps). If the images are different, the agent triggers increased spiking activity in the second motor

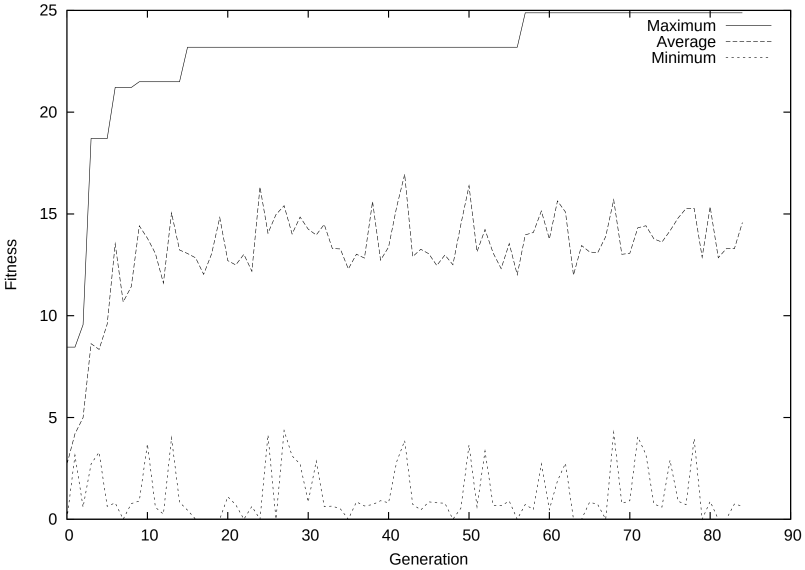


Fig. 3. In this trial an agent achieved very close to the maximum fitness of 25 in only 60 generations

output (it moves right). Furthermore, the disparity in the number of output spikes is not a small percent difference; in the most evolved individuals, it is a compelling 50% difference for each of the responses—the maximum difference for which the agent would be rewarded.

We have created a neural network-based evolutionary autonomous agent with considerably more biological realism than other work. The biological realism of our model extends to the columnar organization of the visual cortex, the Gabor filter-modeled first stage of visual processing, and perhaps most importantly the spiking network itself. We are incorporating biological features into our models in the hopes that we will be able to gather evidence for the hypothesis that some of these biological features lead to greater computational power on dynamic cognitive tasks. We have not yet provided conclusive evidence for that hypothesis, however the ease with which the spiking agents in this experiment evolved to perform the rather complex dynamic memory task is encouraging. Forthcoming experiments will attempt to generalize on many aspects of the one reported here, with more complex dynamic tasks for the agents and an exploration of the lower limits on the size of the models still capable of performing them. More direct comparisons to ANN capabilities will also be attempted.

References

1. Ruppin, E.: Evolutionary autonomous agents: A neuroscience perspective. Nature Reviews Neuroscience **3** (2002) 132–141

2. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics **5** (1943) 115–133
3. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review **65** (1965) 386–408
4. Minsky, M., Papert, S.: Perceptrons; an introduction to computational geometry. MIT Press, Cambridge, MA (1969)
5. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Science **79** (1982) 2254–2258
6. Hopfield, J.: Neurons with graded response have collective computational properties like those of two-state neurons. Proceedings of the National Academy of Science **81** (1984) 3088–3092
7. Maass, W.: Networks of spiking neurons: The third generation of neural network models. Neural Networks **10** (1997) 1659–1671
8. Balakrishnan, K., Honavar, V.: Evolutionary Design of Neural Architectures: A Preliminary Taxonomy and Guide to Literature. Technical report, Department of Computer Science, Iowa State University, Ames, Iowa (1995)
9. Yao: Evolving artificial neural networks. Proceedings of the IEEE **87** (1999) 1423–1447
10. Elman, J.L.: Finding structure in time. Cognitive Science **14** (1990) 179–211
11. Miller, E., Erickson, C., Desimone, R.: Neural mechanisms of visual working memory in prefrontal cortex of the macaque. J Neurosci **Aug 15** (1996) 5154–5167
12. Dan, Y., Atick, J., Reid, R.C.: Efficient coding of natural scenes in the lateral geniculate nucleus: Experimental test of a computational theory. J Neurosci **May 15** (1996) 3351–3362
13. Olshausen, B., Field, D.: Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Research **3311–3325** (1997)
14. Schmitt, F.O., Worden, F.G., Adelman, G., Dennis, S.G.: Anatomy of cerebral cortex: columnar input-output organization. In Jones, E., ed.: The Organization of the Cerebral Cortex. (1981)
15. Hubel, D., Wiesel, T.: Functional architecture of macaque visual cortex. Proc. Roy. Soc. (London) **198B** (1977) 1–59
16. Mountcastle, V.B.: The columnar organization of the neocortex. Brain **120** (1997) 701–722
17. Callaway, E.M.: Local circuits in primary visual cortex of the macaque monkey. Annu. Rev. Neurosci **21** (1998) 47–74
18. Maldonado, P.E., Godecke, I., Gray, C.M., Bonhoeffer, T.: Orientation selectivity in pinwheel centers in cat striate cortex. Science **276** (1997) 1551–1555
19. Tsodyks, M.V., Markram, H.: The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. Proc. Natl. Acad. Sci. USA **94** (1997) 719–723
20. Senn, W., Markram, H., Tsodyks, M.: An algorithm for modifying neurotransmitter release probability based on pre- and postsynaptic spike timing. Neural Computation **13** (2001) 35–67
21. Wilson, E.C., Goodman, P.H., Harris, Jr., F.C.: Implementation of a biologically realistic parallel neocortical-neural network simulator. In: Proc. of the 10th SIAM Conference on Parallel Processing. (2001) 11