

Investigating the Evolution of a Neuroplasticity Network for Learning

Lin Wang^{id}, *Member, IEEE*, and Jeff Orchard

Abstract—The processes of evolution and learning interact. Learning is an evolved strategy that improves fitness, especially in a world where some aspects cannot realistically be encoded in the genome. We endeavored to see if evolution could sculpt a generic neuroplasticity mechanism into a learning rule that would give virtual organisms an advantage in a simulated foraging environment. Our virtual organisms have brains with nine neurons. The connections between those neurons are adjusted by a plasticity rule that is computed by another fixed neural network. Evolution experiments repeatedly found plasticity networks that conferred an adaptive advantage, even outperforming populations that were given a parametric Hebbian plasticity mechanism. Evolution also favored the inclusion of genetically encoded heterogeneity. We also investigate how behavior is influenced by various brain- and movement-related energy penalty terms in the fitness function.

Index Terms—Evolutionary computation, learning rules, neural networks.

I. INTRODUCTION

ANIMALS have evolved complex brains. They take in information about their environment, process that information, and then decide on an action that serves their interests. Presumably, those interests are derivative of evolutionary fitness, the ability to collect resources and turn those resources toward reproductive success (i.e., the number and quality of offspring).

However, most animals are not born with a fully developed central nervous system. The connections in the brain have to be tuned before the animal can implement the desired mapping from sensory input to motor output. The animal has to

learn. Different animals learn to varying degrees; fish can swim and eat from the moment they are hatched, but human babies require years of nurturing before their brains are set up to work effectively in their ecological niche (complex societies with language, etc.).

Despite the huge interspecies range in dependence on learning, neuroscientists are hopeful that the same underlying biological mechanisms are responsible for learning in all animals. Based on this principle, we ask how the evolution of that learning rule might have unfolded.

The capability to change the strength of connections between neurons is called *neuroplasticity*. In this paper, we study the interplay between evolutionary drivers and neuroplasticity. Just like other body parts, the brain can change over the lifetime of the animal. The brain's anatomical and biomolecular changes can manifest as changes in behavior. However, those brain changes depend on the neuroplasticity rule. Some rules might result in adaptive behaviors, while other rules might not.

In the set of evolutionary experiments presented in this paper, a species of primordial organisms is given a generic, flexible neuroplasticity mechanism, a mechanism for changing the connection weights between neurons in its brain. For the sake of biological plausibility, this plasticity mechanism for changing the weight of a connection is based solely on information locally available to that connection (the activity of the neurons it connects).

While the primordial species starts out with a generic plasticity rule (i.e., one that does not result in meaningful learning), we ask, *can evolution discover a way to use that mechanism so that—over evolutionary time—brains of more advanced organisms learn, bestowing an adaptive advantage over organisms with brains that do not learn?* Furthermore, how do different evolutionary pressures affect the learned behaviors?

In addition to evolving a learning rule, this report studies how different energy costs associated with learning and foraging influence the solutions of that evolutionary process. It also indicates that heterogeneity in the network—either in the form of including/excluding certain connections, or varying plasticity rules—seems to accelerate the evolution of the learning rule.

The remainder of this paper is split into five sections. In Section II, we outline the relevant background and related work. In Section III, we describe our foraging task in Section III. In Section IV, we describe our model, including our organisms and the evolutionary framework. Sections V and VI describe our experiments in detail, as well as the results

Manuscript received April 10, 2017; revised August 4, 2017; accepted September 17, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61573166, Grant 61572230, Grant 81671785, Grant 61373054, Grant 61472164, Grant 61472163, Grant 61672262, and Grant 61640218, in part by the Shandong Provincial Natural Science Foundation, China, under Grant ZR2015JL025 and Grant ZR2014JL042, in part by the Science and Technology Project of Shandong Province under Grant 2015GGX101025, in part by the Project of Shandong Province Higher Educational Science and Technology Program under Grant J16LN07, in part by the Shandong Provincial Key Research and Development Program under Grant 2016ZDJS01A12 and Grant 2016GGX101001, and in part by the Natural Sciences and Engineering Research Council of Canada. This paper was recommended by Associate Editor J.-H. Chou. (Corresponding author: Lin Wang.)

L. Wang is with the Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, Jinan 250022, China (e-mail: wangplanet@gmail.com).

J. Orchard is with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMC.2017.2755066

of those experiments. Finally, we conclude this paper in Section VII, and discuss future work.

II. RELATED WORKS

Neural networks are a powerful and flexible computational architecture. Their functionality depends mostly on the connection weights between the nodes in the network. Finding a set of connection weights can be difficult, but a number of strategies have emerged.

In supervised learning, both input and output of a feed-forward network are known, and the network can be trained using error backpropagation [1]. While this method has shown a lot of success, it is not biologically plausible since it depends on the ability to project backwards through the feed-forward connections [2]–[5].

In unsupervised neural learning, the output of the network is unspecified. Despite this lack of direction, the network can still tune itself to efficiently represent the statistically salient structure of its inputs. Many methods were proposed to train deep feed-forward neural networks [6], [7]. However, the deep structure and learning suffer from a time-consuming training process. A broad learning system [8] was proposed to offer an alternative way of learning in deep structure.

Evolutionary algorithms have also been used to find connection weights that enable the network to solve problems [9]–[12]. However, evolution and neural learning can interact in more subtle ways. For example, evolution generated organisms that were better at learning, but, additionally, organisms that learned to predict their future states evolved faster [9]. Moreover, learning can help smooth the evolutionary fitness landscape [13]. For a survey of research that combines evolution and neuroplasticity (see [14]).

But can evolution *discover* a learning rule? The earliest example is [15], in which they evolved the coefficients of a quadratic function that was used to adjust connection weights. The function took four inputs: the current connection weight, presynaptic activity, post-synaptic activity, and an error signal. While their networks evolved and learned to solve simple linear classification problems, they had only two layers, and were strictly feed-forward.

Since then, a number of papers evolving learning have appeared. Many of them [16]–[18] implement the neuroplasticity mechanism using a parameterized Hebbian rule

$$\Delta w_{ij} = \eta(Ar_i r_j + Br_i + Cr_j + D) \quad (1)$$

where r_j and r_i are the presynaptic and postsynaptic activities (respectively), and A – D and η are the parameters to be optimized by evolution.

Floreano and Mondadaab [19] evolved unsupervised, parametric-Hebbian learning rules that enabled a small robot to navigate around a track. Each connection had its own learning rate, and was assigned one of four variants of the Hebbian rule. They showed that the neuroplasticity rules they evolved could be used to establish a “dynamic equilibrium,” whereby the connection weights are updated dynamically throughout the robot’s life, adjusting the robot’s behavior in conjunction with the situation. Later, Floreano and Urzelai [20] used a

similar technique so that a simple robot could learn how to turn on a light switch and then approach the light. In this later work, however, they allowed evolution to select how the rules were encoded in the genome, and whether a rule is specified for a connection, or for an entire neuron. Both of those studies are examples of heterogeneous learning, in which different parts of the network have different plasticity rules.

Heterogeneous plasticity rules can be encoded in other ways. One method, called adaptive hyperNEAT [21], uses a compositional pattern producing network (CPPN) to encode neuroplasticity rules, as well as connectivity patterns. The update rules take the physical location of the presynaptic and postsynaptic neurons as input, as well as the presynaptic and postsynaptic activities. In an alternative formulation, a CPPN was used to encode heterogeneous Hebbian coefficients [as in (1)]. They evolved robots to negotiate a T-maze and collect rewards.

In other work [22], a plasticity rule was based on a parameterized version of the BCM learning method [23]. In their method, three parameters of the BCM rule were derived dynamically using a gene regulatory network (GRN). The 12 parameters that govern the dynamics of the GRN were optimized using covariance-matrix adaptation (CMA) [22]. While the learning capabilities of the resulting networks is impressive, their approach benefits from the preordained design of the BCM learning rule, the form of the GRN equations, and the biologically implausible CMA method. They also pretrain the input and output weights using error backpropagation. In this paper, we want nature to discover the plasticity rules without so much guidance, using biologically relevant evolutionary processes.

It is also possible to implement the synaptic-plasticity update function as a neural network. That is, the connection-weight adjustments for the original neural network are computed using *another fixed* neural network. The first report of this concept is [24]; they evolved a plasticity-rule neural network. This “evolved learning rule” was a fixed, two-layer, feed-forward network, whose output was used to adjust the connection weights of the original neural network. The input for the evolved plasticity rule was the presynaptic and postsynaptic activities, along with an error signal. The goal networks learned to perform linear classification on binary patterns.

The presence of an error signal is extremely helpful in training a neural network. According to [14], the presence of an error signal puts one in the category of “behavioral change.” However, instead of supplying the error signal, we wanted to see if evolution could engineer a solution to the foraging problem without an error signal. This puts us in the category of “behavioral robustness.”

Previous work looked at the evolution of a neural learning rule in the context of a foraging task [25]. The organism was given sensory input of the distance and angle to the nearest food item. However, the task was made more difficult by including an unknown offset in the angle. The neural network’s weights and biases were updated using two neural plasticity networks. The brain was a fully connected network of neurons, and the learning rule was homogeneous (the same at every synapse). Evolution was able to find plasticity networks that

outperformed nonlearning counterparts, as well as populations in which a Hebbian plasticity rule was evolved. Moreover, the neural plasticity network was stable over a much longer time frame than the Hebbian learning rule.

A. Contributions

In this paper, our goal is to investigate the natural emergence of unsupervised learning through the process of evolution. Our organisms have a rudimentary brain as well as a plasticity mechanism that changes the neural connection weights in a nonspecific way (at the beginning). We show that this generic plasticity mechanism is indeed sculpted by evolution into a learning rule, enabling the organisms to out-perform competing species in a foraging task.

We demonstrate this evolutionary learning in a much more biologically rich context, extending [25] in at least five different ways. First, we use a single neural plasticity network to compute both weight and bias updates (rather than one network for the weights, and one for the biases). Second, we include a more difficult version of the foraging task, where the input is in Cartesian coordinates (x, y) instead of polar (ρ, θ) . This task is harder because the angular offset affects both x and y , whereas it only affected θ in the previous work. Third, we do not insist that the brain be fully connected, but allow evolution to determine the connectivity in the brain. Fourth, not only can some connections be excluded (pruned), but we add a nonhomogeneous factor that influences the neuroplasticity function—modeled after the possible modulatory influence that glial cells might exert on synapses. Finally, this paper investigates the influence of different energy penalties on the evolution of behavior.

Furthermore, our model is different from all those mentioned above. Our method does not use a training signal; the network has to evolve or learn its own error signals. Also, our learning rule does not depend on the locations of the neurons. In hyper-NEAT, their plasticity rule is tied to the spatial location of the presynaptic and postsynaptic neurons. Thus, two nearby neurons that receive input from the same (third) neurons must necessarily have similar update rules, and their connection weights will differ only depending on the presynaptic and postsynaptic neural activities. We believe that this is over-constraining, and that real brains have more flexibility. Our method does not use backpropagation, an algorithm that is not biologically plausible because it relies on weight copying or instantaneous error propagation. Finally, our method computes the plasticity updates using a neural network, proven to be able to approximate any nonlinear function. This gives evolution more flexibility than the manually designed update rules, such as the Hebbian and BCM rules.

III. TASK

Our simulation experiments consisted of virtual organisms foraging for food in a virtual world. The foraging task, similar to that used in [9], is described next.

To evolve our organisms, we used two slightly different foraging-like tasks. An organism forages as it moves around in

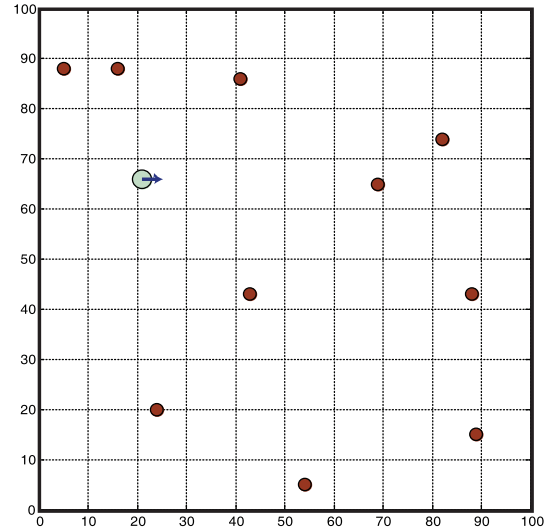


Fig. 1. Environment, organism, and food items in our foraging task.

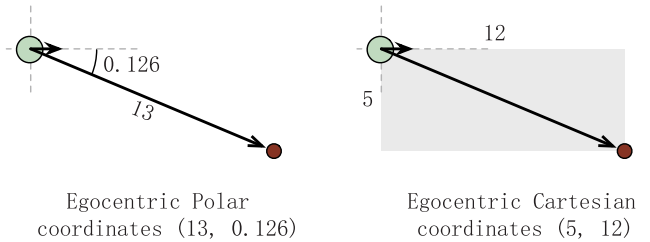


Fig. 2. Egocentric coordinates. Left: egocentric polar coordinates. Right: egocentric Cartesian coordinates.

a 100×100 cell arena, as shown in Fig. 1. Fifty food items are randomly distributed in this arena and are replaced every 50 action steps. Our organism has an orientation (what direction it is facing), and a location. The organism performs actions, stepping from grid cell to grid cell, and chooses its actions using a binary code: $(0, 0)$, stay still; $(1, 1)$, move forward; $(0, 1)$, turn right; or $(1, 0)$, turn left.

At each step, the organism receives sensory input about where the nearest food is.

In the simpler task, the sensory input is in the form of egocentric polar coordinates: distance and angle. The angles are measured in units of half-rotations, so are normalized to the range $[-1, 1]$, where an angle of 0 means the food is straight ahead of the organism, 0.5 means it is to its right, and ± 1 is directly behind it. Fig. 2 illustrates egocentric polar coordinates. More precisely, if 0° is straight ahead, and the angle is measured positive for clockwise rotations, then we can convert from Θ° to a normalized angle θ using the formula $\theta = (\Theta^\circ/180^\circ)$.

In the more difficult task, the sensory input is given in the form of egocentric Cartesian coordinates. For example, food 17 units straight ahead is indicated by input $(0, 17)$, while input of $(5, 12)$ indicates that the food item is 5 units to the right, and 12 units ahead, as shown in Fig. 2.

To make the task more difficult, we introduce an angular offset to the sensory input. At the beginning of the organism's life, a random value is chosen uniformly from the range

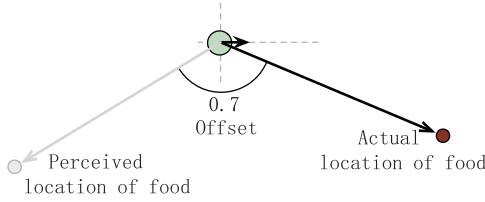


Fig. 3. Angular offset is applied to the egocentric location of the nearest food, so that the sensory input is systematically distorted. For this example, the true location of the food is shown in Fig. 2. But the sensory input is offset by an angle of 0.7, so the polar sensory input would be $(13, 0.826)$, and the Cartesian input would be $(6.8, -11.1)$.

$[-1.5, 1.5]$, and used as a constant angular offset throughout the individual's life. For example, for the polar task, if the nearest food item is actually at $(13, 0.126)$, and the sensory offset is 0.7, then the sensory input will be an angle of 0.826 (corresponding to an angle of 148.6°). This scenario is illustrated in Fig. 3. Notice that the angular offset affects only one of the inputs in the polar task.

The Cartesian task is made more difficult because both inputs are affected. The offset does not have a constant representation in Cartesian coordinates. Instead, both x - and y -coordinates of the sensory input change. Specifically, if ϕ is the offset angle (in normalized units), and the nearest food item is actually at the egocentric Cartesian coordinates (x, y) , then the sensory input will be

$$(\rho \cos(\pi(\theta + \phi)), \rho \sin(\pi(\theta + \phi))) \quad (2)$$

where $\rho = \sqrt{x^2 + y^2}$ and $\theta = \tan^{-1}(y, x)$. For example, an angular offset of 0.7 (corresponding to 126°) would change the true food location of $(5, 12)$ to sensory input of approximately $(6.8, -11.1)$, as illustrated in Fig. 3.

The food items that are collected supply energy, but the activity of the organism also consumes energy. There is an energy cost to moving around, and there is an energy cost to thinking (since neurons require energy to compute). Finally, learning consumes energy, since the metabolic changes in the synapses require energy. All these energy factors will be considered in the fitness function.

IV. METHODS

In this section, we describe our organisms, how they learn, and how they evolved.

A. Brain Model

The behavior of each organism is determined by its brain. The brains of our organisms have nine neurons, consisting of four inputs, two outputs, and three interneurons. Admittedly, this is a small number of neurons, but enough for our purposes.

Of the four inputs, two inputs receive the previous action, and two receive sensory inputs. The two output nodes indicate the next action. Which connections are present in the brain, their initial weights, and the plasticity rule that alters them, are all determined by evolution. Fig. 4 shows a schematic of the brain network.

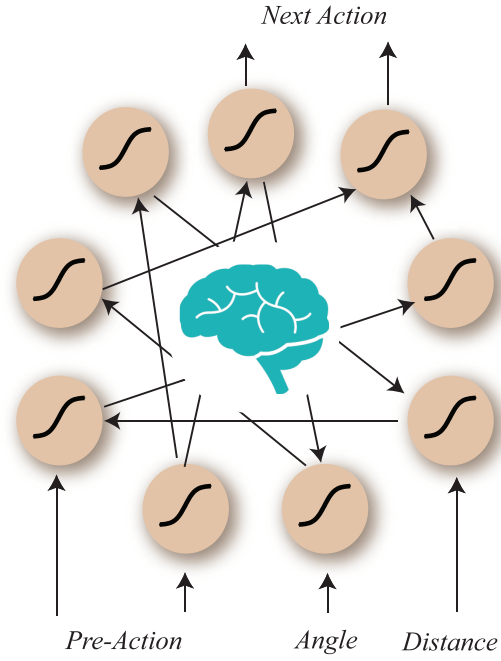


Fig. 4. Brain network. The connections, weights, and learning rule are determined by evolution. Note that not necessarily all possible connections are present.

Each node (neuron) in the brain network is updated by the discrete-time equation

$$r_i = f\left(\sum_j (w_{ij}r_j + \beta_{ij})\right) \quad (3)$$

where r_j is the activity of neuron j , w_{ij} and β_{ij} are the weight and bias (respectively) for the connection from neuron j to neuron i , and $f(\cdot)$ is the logistic function. Note that we assign a different bias current to each synapse, instead of a bulk bias current for each neuron. However, synaptic biases are more general than neuron biases since the bulk neuron bias for neuron i can be derived from the synaptic biases using $\beta_i = \sum_j \beta_{ij}$.

Using (3), the activities of all the neurons are updated simultaneously at each time step.

Each organism chooses an action based on the most recent input. But since the brain network is likely to be recurrent, the network might go through a sequence of transient states before attaining a steady state. In order for the decision to be based on the steady state, each organism performs deliberation steps before the state of its output nodes are used to select the next action. In our experiments, we tried different numbers of deliberations steps to see what impact this variable had on performance.

It is well known that the brain consumes a disproportionately large amount of energy (at least in humans). We include an energy penalty term to account for the metabolic cost of the brain. The brain metabolic cost is

$$\sum_{ij} |w_{ij}| + \sum_{ij} |\beta_{ij}|.$$

The first term accounts for the metabolic cost of maintaining many strong connections, while the second term accounts for the cost of the basal firing rate (or bias current).

B. Plasticity Model

In many of our experiments, we supplied a generic plasticity mechanism for changing synaptic connection weights. As a brain develops, experiences influence the connections. Specifically, a synapse is altered by some combination of the presynaptic activity, post-synaptic activity, and a small number of other potential factors. Trying to figure out how those factors combine into a useful synaptic update is an active area of research. Ideally, the experiences change the connection weights in a way that enables the brain network to solve a particular problem more effectively. This is the goal of neural learning. However, we want to arrive at that goal by natural means.

In an effort to avoid constraining the form of the plasticity function, we used *another* neural network as a computational architecture. Note that this new neural network is not the brain, but is a *different, additional* neural network, being used because neural networks are capable of computing a wide range of functions; this one's purpose is to compute weight updates. This feed-forward neural network (shown in Fig. 5) will be called the *synaptic plasticity network*, or “plasticity network,” emphasizing the word “network.” The inputs to the plasticity network are: the current weight of the synapse, the current bias of the synapse, and the presynaptic and postsynaptic neural activities, as well as a possible hidden variable (elaborated in Section VI-B). The synaptic plasticity network has two outputs: the weight adjustment Δw_{ij} , mapped to the range $[-0.003, 0.003]$, and the bias adjustment $\Delta \beta_{ij}$, also mapped to the range $[-0.003, 0.003]$. These ranges were arrived at by trial and error.

For the parametric *Hebbian plasticity formula* (here, emphasizing the word “formula”), the weight changes, Δw_{ij} , are governed by (1). The bias changes, $\Delta \beta_{ij}$, have an analogous rule, but with different parameter values. Thus, the parametric Hebbian learning is governed by a total of ten parameters (five for the weights, and five for the biases).

For any plasticity mechanism, changing the synaptic weights and biases of the brain network incurs an energy cost. Hence, we consider an energy penalty term of the form

$$\sum_{ij} (|\Delta w_{ij}| + |\Delta \beta_{ij}|).$$

Note that this penalty term involves the *changes* in the weights and biases, not the weights and biases themselves.

C. Evolution

Computational intelligence is a family of approaches which enables the computer to learn a specific task from data or experimental observations and aim at the development of novel problem-solving techniques [26], [27]. As one of them, evolutionary computation methods [28]–[31] are capable of discovering natural laws and designing systems automatically, without any prior knowledge [32]–[39]. In this paper, we use

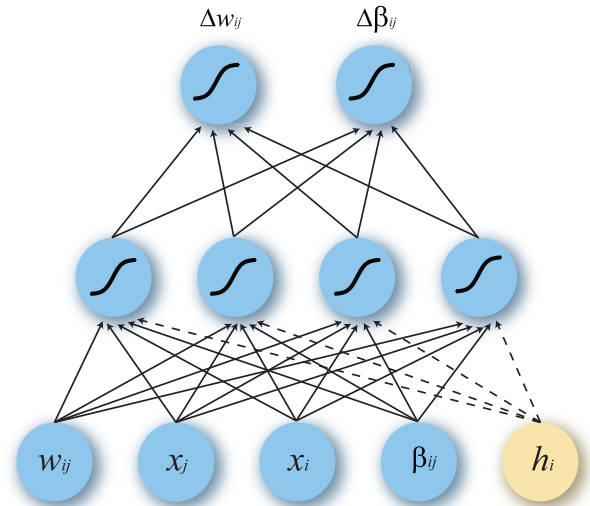


Fig. 5. Synaptic plasticity network. The outputs of this plasticity network determine the adjustments to the connection weights and biases in the brain network.

a genetic algorithm [28] to evolve the laws that govern the operation of the central nervous system—not only the connections and their initial weights, but also the plasticity rule that adjusts them.

We used a binary genetic algorithm to investigate the natural evolutionary processes that might have developed the central nervous system. The chromosomes encode a number of critical pieces of information about the brain and its plasticity rule.

For the brain network, the chromosome encodes the connection weights of the sensory inputs into the brain, a connection matrix (between neurons of the brain), and initial connection weights. Note that not all of our experiments use the initial weights, or connection matrix.

For the plasticity rule, obviously we have to encode different data for different plasticity mechanisms. For simulations that use the Hebbian plasticity formula, the chromosome encodes the values of the ten parameters [see (1)]. For simulations that use the synaptic plasticity network, the chromosome encodes the connection weights and biases that govern the plasticity neural network's output, thus determining the function that the plasticity network computes. The chromosome also encodes a synapse-specific “hidden variable” that can be used to modify the plasticity function on a synapse-by-synapse basis. Note that this hidden variable is not used in all experiments.

All the genes are encoded linearly into the chromosome using direct binary representation.

Our genetic algorithm performs crossover, mutation, selection, and fitness evaluation, in sequence, on a population of individuals. In all of our experiments, we used a uniform crossover rate with a probability of 0.9, and a mutation probability of 0.01. In order to accelerate convergence, we used a tournament selection operation with a tournament size of 32. Each population had 960 individuals (a multiple of 32); many of the solutions yield very low fitness, and having a larger population enables a broader search of the fitness landscape. Each evolutionary simulation was run for 200 generations, which seemed to be enough to allow the fitness to plateau.

An individual's fitness is based on the number of food items acquired over-and-above the energy expended in the process. The organism consumes energy through three processes: brain metabolism, learning, and moving. The amount of energy consumed by those processes is subtracted from the total energy collected, yielding the *surplus energy*

$$\text{Surplus} = \text{Food Eaten} - \text{Brain Metabolism} \\ - \text{Learning} - \text{Moving}.$$

Naturally, the surplus cannot be negative. Furthermore, there is a cap on the energy surplus, since an organism cannot store more than a fixed amount of energy. Hence, an organism's fitness is

$$\min(F_{\max}, \max(0, \text{Surplus}))$$

where F_{\max} is the maximum allowable energy surplus. Once an organism has collected enough food to reach its fitness maximum, the benefit of collecting more food is diminished. In our experiments, we scale F_{\max} so that it is proportional to the lifetime of the organism. Maintaining an F_{\max} value of 2% of the number of action steps (one food item for every 50 action steps, on average) gives the longer-living organisms a higher goal to aspire to.

D. Acceleration

Evolution can be a very time-consuming process; in our case, the entire life-long foraging task has to be performed for every chromosome in the population for each generation. The parallel and distributed computing has being widely used to solve computational expensive problems [40]–[42]. Therefore, we used a GPGPU-based high-performance computer, with a peak speed of 9TFlops. The computing platform includes four Kepler GK104 chips, each of which has 1536 stream processors with clock frequencies of 0.75 GHz.

Fig. 6 illustrates our adopted parallelization method. A master/slave architecture is used for parallelization, in which the general processes of the genetic algorithm are performed by the CPU. After mutation and crossover, the whole population is copied into the memory space of the GPGPU. Then, each GPGPU thread decodes a chromosome, selects a random offset, and simulates the life-long foraging task for an individual, returning the fitness to the CPU. Each individual is simulated in a variety of different sensory offsets, and its final fitness is its average fitness over all of its trials.

In the following two sections, we report on a series of experiments. The first section deals with how the neural plasticity rule performs compared to nonlearning and Hebbian-learning counterparts. The section after that investigates how some of the parameters of our model affect its behavior.

V. EFFECTIVENESS OF PLASTICITY

How does the plasticity network perform? To answer this question, we compared the results from using the plasticity network against two other strategies, one that learns using a global Hebbian plasticity formula, and one that does no learning at all. We will denote these three different methods as “plasticity network,” “Hebbian formula,” and “nonlearning,”

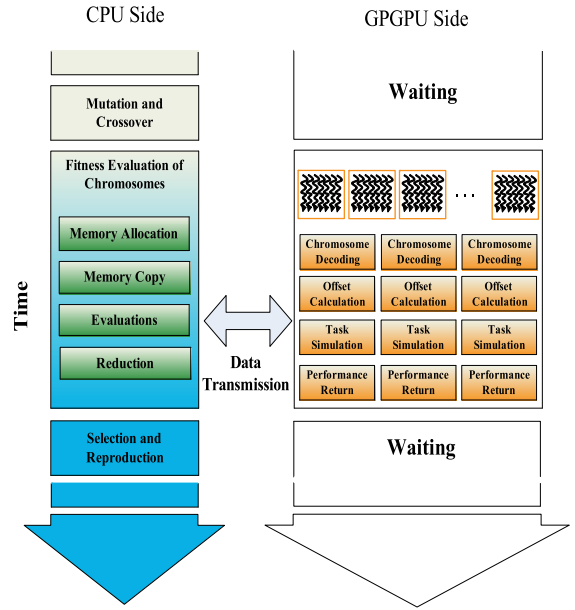


Fig. 6. Sequential relationship of GPGPU acceleration.

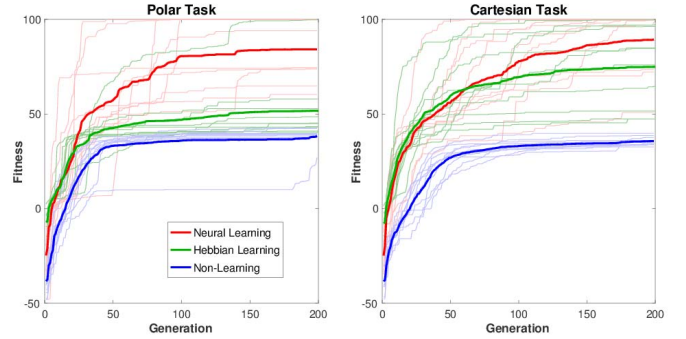


Fig. 7. Evolutionary convergence. The light-colored curves show the historical best for each of the ten trials for each method. The darker, thicker lines show the average over the ten evolutionary trials for each method. Best viewed in color.

respectively. Moreover, we simulated the process of evolution of these three different lineages on both polar task and Cartesian task.

For these experiments, we used the following parameter values: brain metabolic cost = 5×10^{-6} ; coefficient of learning cost = 2×10^{-5} ; and coefficient of movement cost = 3×10^{-3} . The plasticity-network organisms had only structural heterogeneity enabled. Thus, the plasticity-network update rule was homogeneous and global, and was the same for all synapses. Each organism had a lifetime of 5000 action steps, with 10 deliberation steps per action, and F_{\max} was set to 100.

A. Evolutionary Convergence

The plots in Fig. 7 show the fitness of the three methods over 200 generations. The plasticity network rule yielded the strongest performance, converging to a higher fitness than the other two methods.

The individual trial curves are also shown in Fig. 7, plotting the historical best fitness for each evolutionary trial. The populations using the neural plasticity rule exhibit more sudden jumps in fitness than the other populations. Each jump is

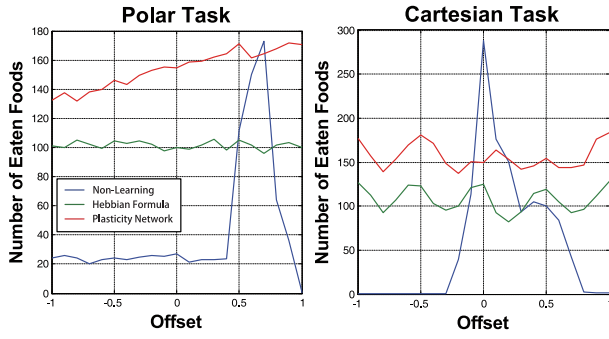


Fig. 8. Lifetime performance for different sensory offsets. For each organism type, the organism with the highest fitness over all ten evolutionary trials was chosen for this test. The apparent linear trend in the plasticity network's performance in the polar task is circumstantial, and not systematic. Best viewed in color.

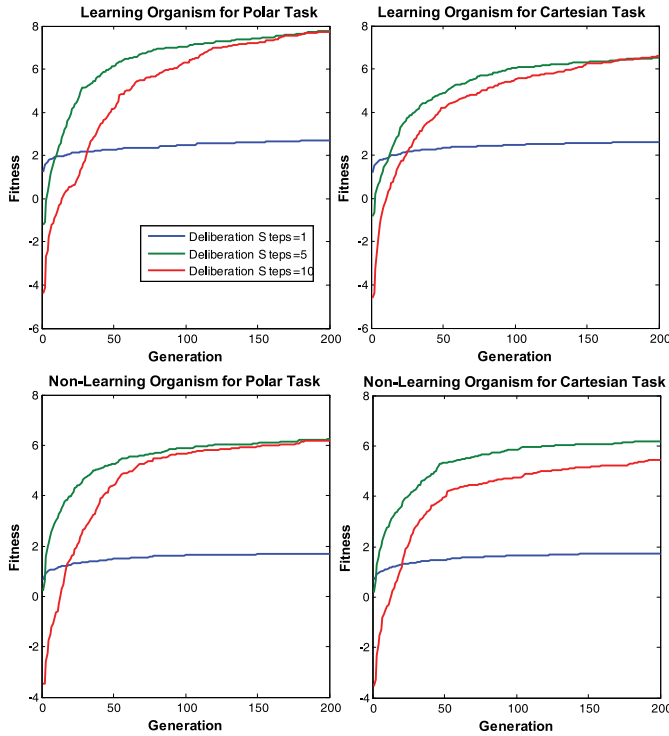


Fig. 9. Evolutionary progress of fitness using different numbers of deliberation steps for learning organisms (plasticity network) and nonlearning organisms on both tasks. Each curve is obtained by averaging over ten evolutionary trials. Best viewed in color.

an innovation in which a particularly advantageous mutation spreads throughout the population. This phenomenon suggests that the fitness landscape for the neural plasticity rule is less smooth than the fitness landscape for the other methods.

The figure also shows some differences between the methods when they are initialized randomly (at generation 0). The Hebbian learning rule tends to start at a higher fitness than the other methods, likely because the form of learning rule is already established, so evolution does not need to discover it.

The nonlearning populations tend to have the lowest fitness at generation 0. This, again, indicates that the fitness landscape is less smooth for the learning methods than for the nonlearning method. Randomly choosing organisms for the

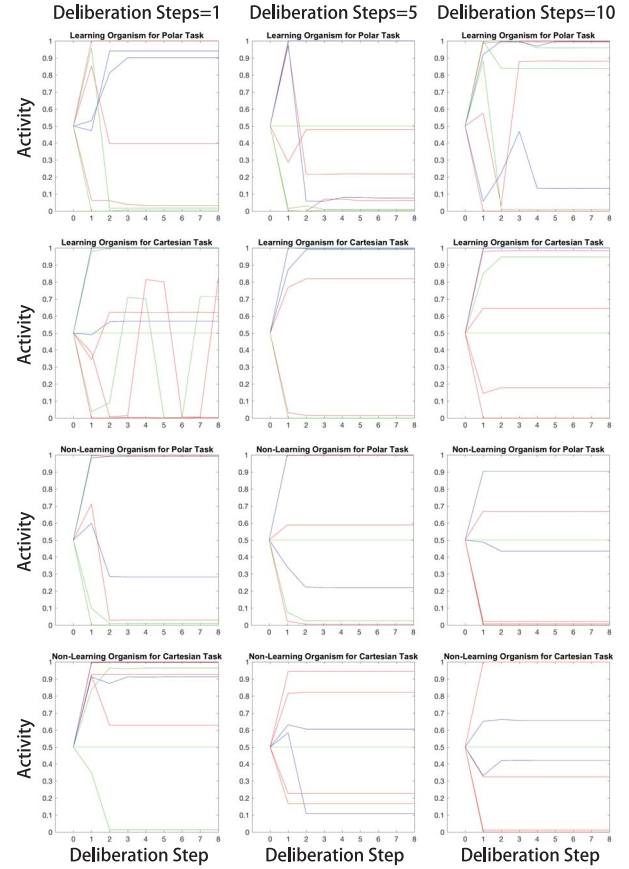


Fig. 10. Activities of all nine brain neurons over ten deliberation steps when input vector is fixed to $(0, 0, 0, 1)$. The four input neurons are plotted in red. The three interneurons are plotted in green, and the two output neurons are plotted in blue. The brains were evolved with either 1, 5, or 10 deliberation steps. The solution from the trial which fitness is the best one in all evolution trials is selected. Best viewed in color.

nonlearning populations yielded fairly low and consistent best-fitness values. However, for the learning populations, some of the randomly generated individuals exhibited relatively high fitness, suggesting that nearby points in the fitness landscape can yield drastically different fitness values.

B. Offsets Coverage

We wanted to see how well our best organisms could generalize their ability to cope with different sensory offsets. We chose the best organism from the nonlearning trials, from the Hebbian-formula learning trials, and from the plasticity-network learning trials, and tested each of them over 21 lifetimes, each lifetime with a different sensory offset.

Fig. 8 plots their performance over the various offsets. It shows that the best nonlearning organism is a specialist, achieving near-optimal performance over only a very limited range of offsets. For the other offsets, this organism performed very poorly.

In contrast, the learning organisms were able to handle all offsets effectively. That means that their plasticity rules enabled them to adapt to environments with different sensory offsets. In addition, the plasticity-network rule showed higher performance than the Hebbian-formula rule.

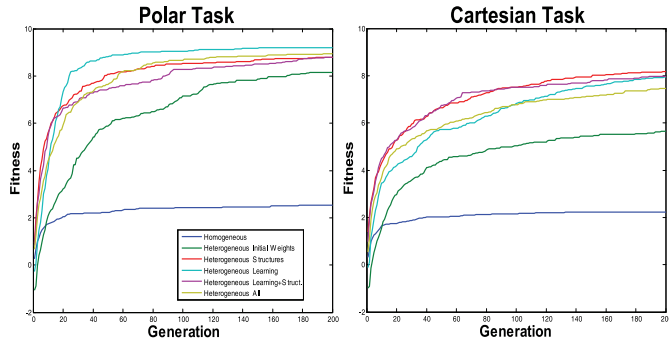


Fig. 11. Evolutionary progress with different combinations of network heterogeneity. Each curve is obtained by averaging ten evolutionary trials. Best viewed in color.

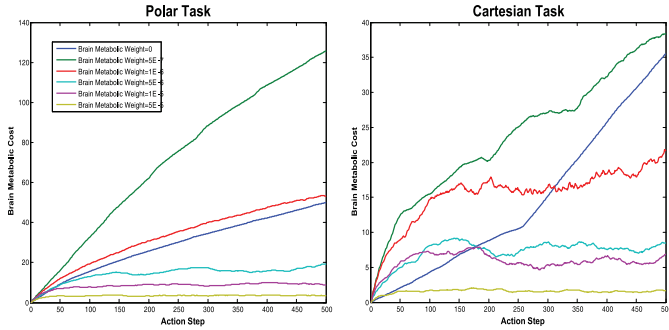


Fig. 12. Brain metabolic cost for different weighting coefficients. Each curve is obtained by averaging over ten evolutionary trials. Best viewed in color.

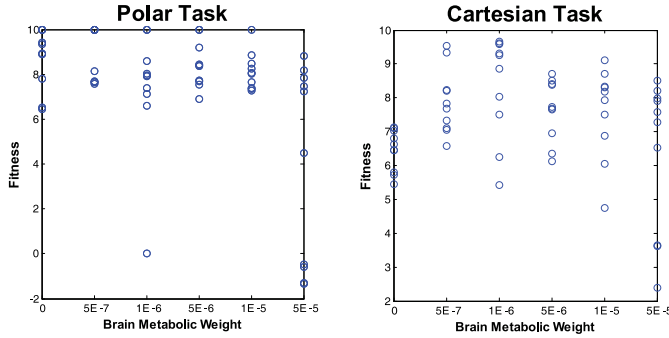


Fig. 13. Fitness for different weights of brain metabolic cost.

VI. SENSITIVITY EXPERIMENTS

We also ran a number of evolutionary lineages using the plasticity-network populations, but varied the different model parameters. This sensitivity analysis could shed some light on how evolution arrives at slightly different solutions under different circumstances. The behaviors that result should reflect, to some extent, the challenges faced during evolution. The following sections outline the different experiments and their results. Unless specified otherwise, the parameter values were: brain metabolic cost = 5×10^{-6} ; coefficient of learning cost = 2×10^{-5} ; coefficient of movement cost = 3×10^{-3} ; structural heterogeneity (homogeneous update rule and initial weights); lifespan = 500 action steps; ten deliberation steps per action step; and $F_{\max} = 10$. In all of our experiments, we set F_{\max} to be 2% of the number of action steps.

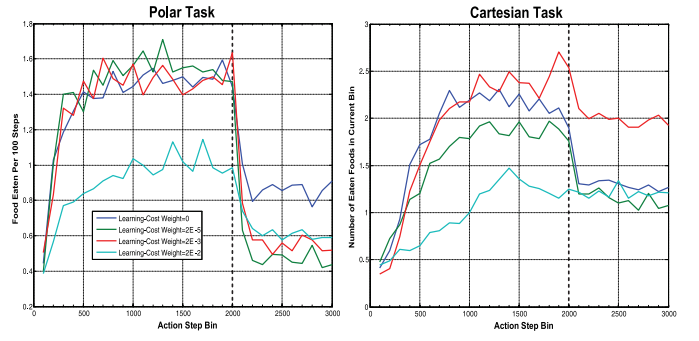


Fig. 14. Food consumption rate during and after learning. Learning (plasticity) is switched off after 2000 action steps. Each curve is obtained by averaging ten evolutionary trials, and each time bin includes 100 action steps. Best viewed in color.

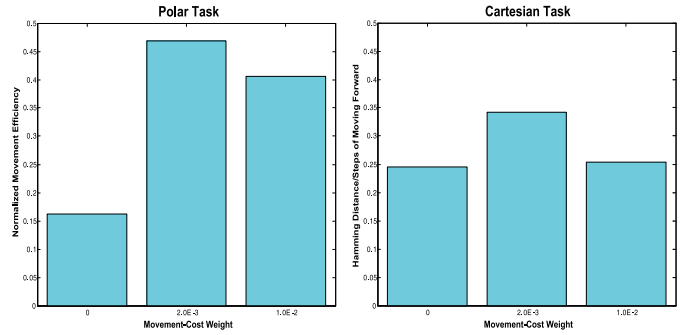


Fig. 15. Movement efficiency for different movement-cost weights. The efficiency is defined by the ratio of the Manhattan distance to the food, divided by the number of forward steps taken to reach the food. The results are obtained by averaging ten evolutionary trials.

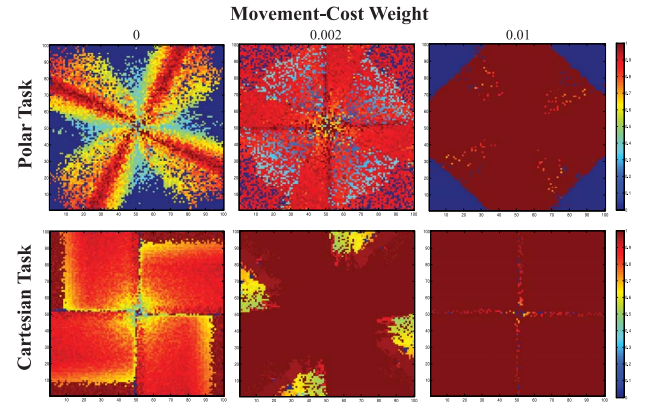


Fig. 16. Movement efficiency map for different movement-cost weights. The organism starts at the center. The color at each location indicates the efficiency of reaching a food item there. The orientation offset is -1 throughout. For each task, the best organism over all ten evolutionary trials was chosen for this test. Best viewed in color.

A. Influence of Deliberation Steps

We hypothesized that it is helpful for the brain network to come to a stable state before choosing an action. Fig. 9 plots the evolutionary performance of the learning and nonlearning populations in the two tasks. It shows that the performance improves if we allow at least five deliberation steps. Forcing an action after just one step causes fitness to plateau at a low level.

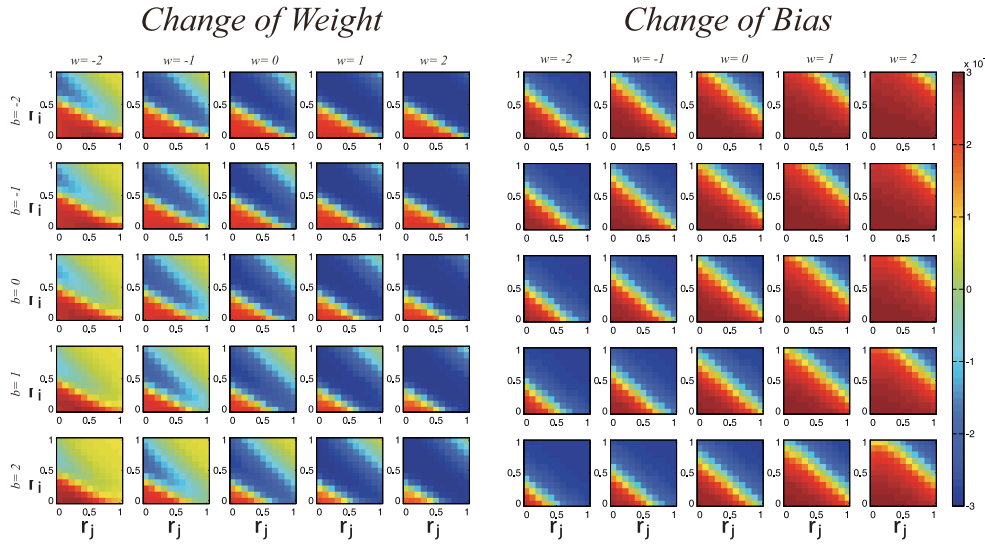


Fig. 17. Plasticity network function for polar task for the individual with the highest fitness over all ten trials.

Fig. 9 shows the same deliberation effect on both plasticity-network learning and nonlearning populations. Hence, the effect is due to thinking, not learning.

Fig. 10 plots the activity of the brain's nine neurons over ten deliberation steps. We see that even the organisms that evolved with only one deliberation step (the left column of the figure) do not typically reach a steady state after just one step. We did not specifically investigate if the action chosen after one step was better than the subsequent actions arrived at after more deliberation, but Fig. 9 suggests that too little deliberation is disadvantage.

Fig. 10 also shows that organisms that evolved with five or ten deliberation steps reach steady state within about five steps. The behavior of the learning organisms seems more complex than that of the nonlearning. Perhaps their neural computation is more complex (and requires more time to converge), or maybe the neuroplasticity rule is changing the brain's connections dynamically, as in the dynamic equilibrium in [19].

B. Heterogeneous Factor

In our experiments, we wanted to investigate if variations from a global learning rule would be beneficial. We evolved several lineages allowing different heterogeneous factors. We enabled three types of genetically encoded heterogeneity.

- 1) *Structure*: Neural connections to include in the network.
- 2) *Learning*: A connection-specific modifying factor that changes the behavior of the learning rule for each connection.
- 3) *Initialization*: Encoding of initial weights.

Fig. 11 shows how these heterogeneous factors affect the progress of evolution over the course of 200 generations. The lineages in which heterogeneity was absent showed slower evolution. Genetic encoding of initial connection weights also showed slow evolution, compared to the lineages that allow heterogeneity in connections and plasticity rules. In general, it seems that allowing heterogeneity in *either* connections or

the hidden neuroplasticity factor enabled the populations to evolve more quickly, attaining a much higher fitness level by the end of the 200 generations.

C. Brain Metabolic Cost

Energy considerations can have a profound effect on the evolution of a species. We start our analysis by looking at the metabolic cost of the brain.

We computed the brain metabolic cost using

$$\sum_{ij} |w_{ij}| + \sum_{ij} |\beta_{ij}| \quad (4)$$

where the first term accounts for the cost of maintaining synapses, and the second term accounts for the basal firing rate (or bias current). Larger connection weights imply larger/more synapses.

Fig. 12 plots the brain metabolic cost over the organism's lifetime. With a small coefficient, the weights and biases continue to climb over an extended period of time. With a larger coefficient, the weights and biases stabilize quickly. In general, a larger coefficient results in smaller connection weights and biases.

An interesting exception to that rule occurred when the value for the coefficient was zero. The corresponding brain metabolic energy was lower than for that for some higher coefficient values. For example, in both plots in Fig. 12, a coefficient value of 5×10^{-7} yielded higher weights and biases than a coefficient value of 0 (compare the green curve with the dark-blue curve). Irregularities like this could be the result of a complex fitness landscape that changes as the coefficient changes value. Further investigation of this phenomenon is needed.

From Fig. 13, we see that there is an intermediate value for the coefficient of the brain metabolic cost that yields higher performance than extreme values. Not surprisingly, very high weights for the brain metabolic cost result in lower fitness, since this energy term is subtracted from the food eaten to

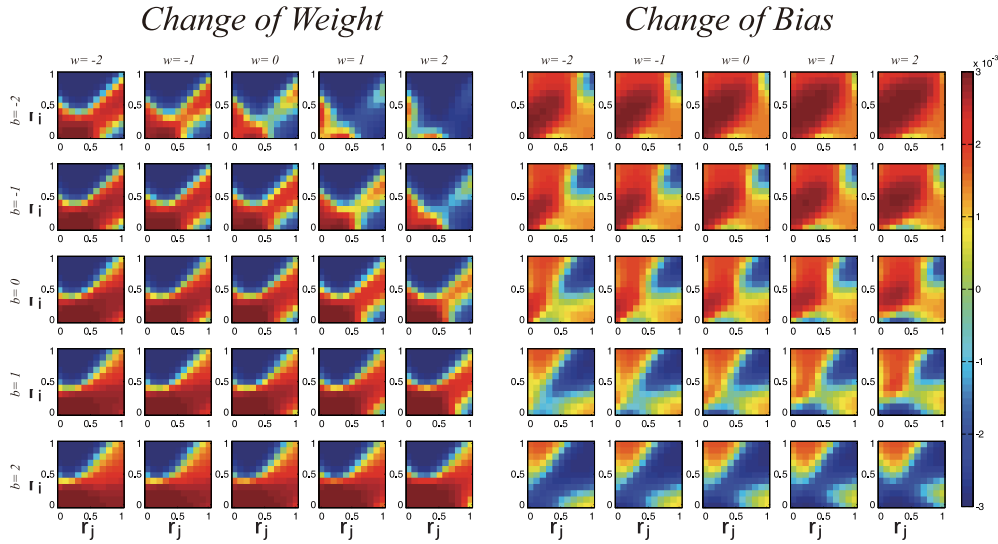


Fig. 18. Plasticity network function for Cartesian task for the individual with the highest fitness over all ten trials. Best viewed in color.

arrive at the fitness. Somewhat more surprising is the fact that very low (or zero) weight can also reduce fitness. According to Fig. 13, a brain metabolic weight of zero yields poorer performance than an intermediate value for the Cartesian task. This phenomenon is likely linked to the one mentioned above, but more investigation is needed to establish this connection.

The energy term in (4) is similar to the L_1 -norm constraint that is characteristic of the lasso method [43]. For multilayer perceptrons, the lasso method has been shown to make the weight matrix more sparse [44]. Moreover, the use of fewer connections yielded more generalizable solutions [44].

D. Energy Consumption of Learning

We can alter the energy cost of synaptic plasticity on our populations. Fig. 14 shows the food consumption rate (over a lifetime) for various weightings of the learning energy cost. During evolution, these organisms live for 2000 action steps, in a world in which F_{\max} is set to 40. However, for testing, we also wanted to see how the organisms fare when learning is disabled. Hence, for testing, each organism learns for the first 2000 action steps, but then learning is turned off for an additional 1000 steps. We see in Fig. 14 that turning the learning off causes a drastic decrease in food consumption rate for organisms evolved with a small learning energy coefficient. The effect is less pronounced, or absent, for organisms evolved with a larger coefficient.

First of all, this finding is consistent with the intuitive notion that a higher cost on neuroplasticity will result in organisms that depend less on continual synaptic adjustments. The figure also shows that the performance for the highest learning-cost (the cyan curves) is the lowest. It seems that the organisms sacrifice some eating in order to reduce the energy penalty that a higher consumption rate would require. Interestingly, the highest consumption rate does not correspond to a zero learning-cost weight. This suggests that there might be an optimal learning-cost weight for food consumption rate.

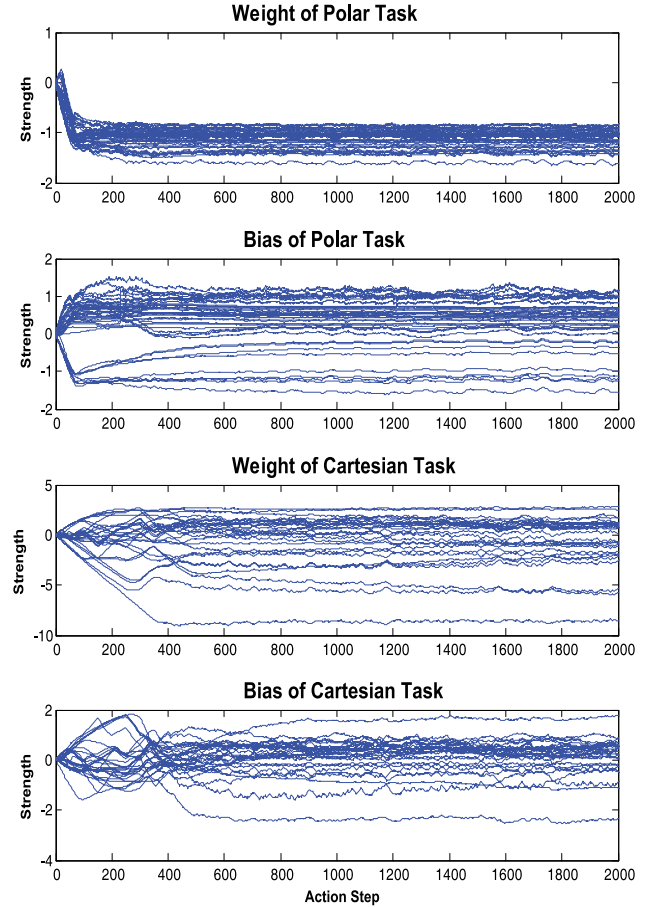


Fig. 19. Lifelong time course for all brain weights and biases. The plasticity-network organism with the highest fitness over all ten evolutionary trials was chosen for this test. Best viewed in color.

Fig. 14 also suggests that the updates generated by the plasticity network can be used to perform part of the computation in a dynamic manner. In other words, the plasticity is enabling a dynamic equilibrium similar to that reported in [19], possibly

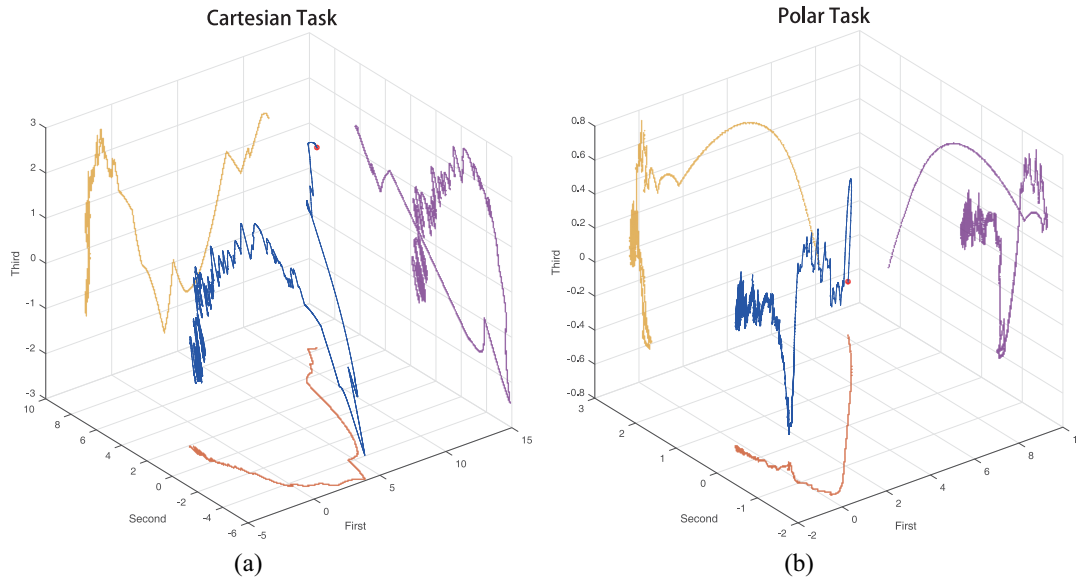


Fig. 20. State-space representation of dynamics of weights and biases during 2000 actions plotted as trajectory within the space of the first three principal components. The red spot represents the initial point.

adjusting the weights and biases for each newly targeted food item. In this sense, our simulations fall under the “behavioral robustness” category suggested in [14].

E. Energy Consumption of Moving

Movement also incurs an energy penalty. Fig. 15 plots movement efficiency for three different values of the weight of the movement cost term. The organism’s lifetime is 2000 action steps and F_{\max} is set to 40. We computed the movement efficiency as the ratio of the Manhattan distance to the food, divided by the number of forward movements taken to reach the food. A value of 0.5 means that the organism took double the number of forward steps to reach the food compared to the optimal.

Interestingly, intermediate values of the movement cost weight seem to yield the best efficiency. A movement cost weight of zero means that the organisms have evolved in a world in which movement is cost-free, and they move around a lot. In this case, they exhibit low efficiency as expected. However, the highest value of the coefficient did not yield the most efficient behavior. We conjecture that this may be due to the fact that a higher coefficient separates the peaks in the fitness landscape with deeper valleys, making the search for the optimal solution more difficult. Hence, it seems that there is an ideal range of values for the movement cost weight. A student’s t -test confirms that the means shown in Fig. 15 are significantly different with 95% confidence.

Fig. 16 shows a spatial map of efficiency for the two tasks for three different movement cost weights. Each map shows the efficiency of the best organism for reaching a food item at that location, with the organism starting from the center of the map. As the movement cost increases (left-to-right in the figure), the efficiency of the best organism improves. Note that this trend does not necessarily indicate the entire population is more efficient *on average*, since Fig. 15 shows that the mean

population fitness does not always increase. Thus, there must be many organisms that evolved in the Cartesian task that are not efficient (i.e., more variability in those cases).

It is interesting to note that the polar task with the highest movement cost exhibits a no-go region (the corners of the arena). We conjecture that the organism has learned that a food item more than 50 spaces away is not worth moving toward (since it will disappear before it can reach it; recall that all food items are replaced every 50 action steps).

F. Analysis of the Plasticity-Network Rule

The plasticity-network rule has four (or five) inputs and two outputs. Its input consists of: current weight, current bias, presynaptic activity, postsynaptic activity, and (optionally) an additional heterogeneous factor. Its output is the weight adjustment, and the bias adjustment. Visualizing this high-dimensional function is challenging. Figs. 17 and 18 offer a visualization of the plasticity-network function for the polar task and the Cartesian task, shown for the top individual for each task.

Fig. 19 shows the time-course of the connection weights and biases over the lifetime of an individual evolved in a population where lifetime was 2000 action steps and F_{\max} was 40. For both tasks, the weights and biases change drastically over the first 400 or 500 action steps, but change little after that. There are, however, small fluctuations that persist throughout the organism’s lifetime. Compared with manually designed dynamic systems [45]–[47], this figure shows that the evolved dynamics is also able to lead the system to stable states.

Given that the weights and biases are fairly constant over time (once the individual has matured), we can focus our attention around the zero-crossings of the plasticity function.

If a weight does not change, then the plasticity function must have generated a weight adjustment of 0. The same is true for the bias. Hence, we can expect that the locus

of activity in the plasticity function must be around zero-crossings. Thus, the typical combinations of inputs (current weight and bias, presynaptic and postsynaptic activities, and hidden factor, where applicable) hover around zero-crossings of the plasticity function. The zero-level sets of the plasticity functions shown in Figs. 17 and 18 (indicated by the color green) form complicated hyper-surfaces in the 5-D space, offering a variety of steady states with different presynaptic and postsynaptic activities, weights, and biases (and possibly heterogeneous factors).

Fig. 20 shows the state of the network weights and biases during a 2000-action lifetime, projected onto the space of the three principal components. Starting from the red dot (showing the initial weights/biases state), the weights and biases migrate toward a region of convergence. However, similar to Fig. 19, the weights and biases do not converge to a constant, but rather converge to a region where they continue to exhibit small fluctuations. Those fluctuations appear to be the interaction of the plasticity function with the varying sensory input, resulting in a dynamic equilibrium similar to that reported in [19]. Recall that for this experiment, the weight of the learning energy cost is 2×10^{-5} .

VII. CONCLUSION

In this paper, we demonstrated that evolution can take advantage of a flexible synaptic plasticity mechanism to accelerate adaptation. The plasticity-network rule evolved individuals with a higher fitness value than the parametric Hebbian-formula rule, as well as a nonlearning individuals. The individuals that evolved with learning also had the flexibility to adapt to different sensory offsets; the nonlearning populations competed by having mediocre fitness scores on average, but upon closer inspection, their behavior was rather inflexible and specialized for a small range of sensory offsets.

As expected, including different energy terms in the fitness function influenced the behaviors of the learning individuals. The organisms adapted to different environments. Sometimes the adaptations were expected, such as the decreased weights and biases that accompany an increase in the coefficient for the brain metabolic cost. Some influences, however, were counter-intuitive. For example, a brain metabolic cost coefficient of zero yielded lower brain metabolic cost than when that coefficient was set to 5×10^{-7} . This phenomenon warrants further investigation.

Another energy term that could be added to the fitness function is brain activity. The brain activity cost could be computed as the sum of the activities of the neurons in the brain. The more active the neurons, the more energy the brain consumes. We predict that this term might encourage sparse neural solutions, where only a subset of neurons are active at a time. These future investigations should probably involve brains with more than nine neurons.

We found that learning individuals from populations that evolved with a small (or zero) coefficient for learning cost tended to dynamically involve the neuroplasticity rule in their computational strategy. This was made evident by the sudden drop in performance when the neuroplasticity rule was

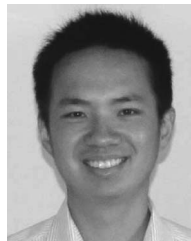
turned off (see Fig. 14). We conjecture that we are observing the same type of dynamic equilibrium previously pointed out by Floreano and Mondadaab [19], but further investigation is needed to verify this hypothesis.

An interesting extension of this paper is to compare the Hebbian and plasticity network learning rules to other update functions, such as hyperNEAT [21] or BCM [23]. Another avenue for future work is to investigate the impact of a modulatory network that affects the learning rule. Finally, understanding the plasticity-network learning rule is a complex and difficult undertaking. A full examination of it is beyond the scope of this paper, and is left for future work.

REFERENCES

- [1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [2] D. G. Stork, "Is backpropagation biologically plausible?" in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 2. Washington, DC, USA, 1989, pp. 241–246.
- [3] Y. Bengio, D. Lee, J. Bornschein, and Z. Lin, "Towards biologically plausible deep learning," *CoRR*, vol. abs/1502.04156v1, 2015. [Online]. Available: <http://arxiv.org/abs/1502.04156>
- [4] W. He *et al.*, "Multiagent systems on multilayer networks: Synchronization analysis and network design," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1655–1667, Jul. 2017, doi: [10.1109/TSMC.2017.2659759](https://doi.org/10.1109/TSMC.2017.2659759).
- [5] C. L. P. Chen, J. Wang, C.-H. Wang, and L. Chen, "A new learning algorithm for a fully connected neuro-fuzzy inference system," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1741–1757, Oct. 2014.
- [6] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [7] C. L. P. Chen, C.-Y. Zhang, L. Chen, and M. Gan, "Fuzzy restricted Boltzmann machine for the enhancement of deep learning," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 2163–2173, Dec. 2015.
- [8] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: [10.1109/TNNLS.2017.2716952](https://doi.org/10.1109/TNNLS.2017.2716952).
- [9] S. Nolfi, D. Parisi, and J. L. Elman, "Learning and evolution in neural networks," *Adapt. Behav.*, vol. 3, no. 1, pp. 5–28, 1994.
- [10] S. Schliebs and N. Kasabov, "Evolving spiking neural network—A survey," *Evol. Syst.*, vol. 4, no. 2, pp. 87–98, 2013.
- [11] L. Wang, B. Yang, Y. Chen, X. Zhang, and J. Orchard, "Improving neural-network classifiers using nearest neighbor partitioning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2255–2267, Oct. 2017.
- [12] C.-T. Lin, M. Prasad, and A. Saxena, "An improved polynomial neural network classifier using real-coded genetic algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 11, pp. 1389–1401, Nov. 2015.
- [13] G. E. Hinton and S. J. Nowlan, "How learning can guide evolution," *Complex Syst.*, vol. 1, no. 3, pp. 495–502, 1987. [Online]. Available: http://www.complex-systems.com/abstracts/v01_i03_a06.html
- [14] J.-B. Mouret and P. Tonelli, "Artificial evolution of plastic neural networks: A few key concepts," in *Growing Adaptive Machines: Combining Development and Learning in Artificial Neural Networks*, vol. 557, T. Kowaliw, N. Bredeche, and R. Doursat, Eds. Heidelberg, Germany: Springer, 2014, pp. 251–261.
- [15] D. J. Chalmers, "The evolution of learning: An experiment in genetic connectionism," in *Proc. Connectionist Models Summer School*, 1990, pp. 1–20. [Online]. Available: <http://consc.net/papers/evolution.pdf>
- [16] Y. Niv, D. Joel, I. Meilijson, and E. Ruppin, "Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors," *Adapt. Behav.*, vol. 10, no. 1, pp. 5–24, 2002.
- [17] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, "Evolving adaptive neural networks with and without adaptive synapses," in *Proc. Congr. Evol. Comput.*, vol. 4. Canberra, ACT, Australia, 2003, pp. 2557–2564. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1299410

- [18] P. Tonelli and J.-B. Mouret, "On the relationships between synaptic plasticity and generative systems," in *Proc. 13th Annu. Conf. Genet. Evol. Comput. (GECCO)*, Dublin, Ireland, 2011, pp. 1531–1538. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=2001576.2001782>
- [19] D. Floreano and F. Mondadaab, "Evolutionary neurocontrollers for autonomous mobile robots," *Neural Netw.*, vol. 11, nos. 7–8, pp. 1461–1478, 1998.
- [20] D. Floreano and J. Urzelai, "Evolutionary robots with on-line self-organization and behavioral fitness," *Neural Netw.*, vol. 13, nos. 4–5, pp. 431–443, 2000.
- [21] S. Risi and K. O. Stanley, *Indirectly Encoding Neural Plasticity As a Pattern of Local Rules* (LNCS 6226). Heidelberg, Germany: Springer, 2010, pp. 533–543.
- [22] J. Yin, Y. Meng, and Y. Jin, "A developmental approach to structural self-organization in reservoir computing," *IEEE Trans. Auton. Mental Develop.*, vol. 4, no. 4, pp. 273–289, Dec. 2012.
- [23] E. L. Bienenstock, L. N. Cooper, and P. W. Munro, "Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex," *J. Neurosci.*, vol. 2, no. 1, pp. 32–48, 1982.
- [24] T. P. Runarsson and M. T. Jonsson, "Evolution and design of distributed learning rules," in *Proc. IEEE Symp. Combinations Evol. Comput. Neural Netw.*, San Antonio, TX, USA, 2000, pp. 59–63.
- [25] J. Orchard and L. Wang, "The evolution of a generalized neural learning rule," in *Proc. Int. Joint Conf. Neural Netw.*, Vancouver, BC, Canada, 2016, pp. 4688–4694.
- [26] L. Peng, H. Zhang, B. Yang, and Y. Chen, "A new approach for imbalanced data classification based on data gravitation," *Inf. Sci.*, vol. 288, pp. 347–373, Dec. 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025514005118>
- [27] J. Zhou, L. Chen, C. L. P. Chen, Y. Zhang, and H.-X. Li, "Fuzzy clustering with the entropy of attribute weights," *Neurocomputing*, vol. 198, pp. 125–134, Jul. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231216003155>
- [28] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Ann Arbor, MI, USA: MIT Press, 1992.
- [29] C. L. P. Chen, T. Zhang, L. Chen, and S. C. Tam, "I-Ching divination evolutionary algorithm and its convergence analysis," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 2–13, Jan. 2017.
- [30] L. Wang, B. Yang, and Y. Chen, "Improving particle swarm optimization using multi-layer searching strategy," *Inf. Sci.*, vol. 274, pp. 70–94, Aug. 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025514002564>
- [31] L. Wang, B. Yang, and J. Orchard, "Particle swarm optimization using dynamic tournament topology," *Appl. Soft Comput.*, vol. 48, pp. 584–596, Nov. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494616303763>
- [32] S. He *et al.*, "Cooperative co-evolutionary module identification with application to cancer disease module discovery," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 874–891, Dec. 2016, doi: [10.1109/TEVC.2016.2530311](https://doi.org/10.1109/TEVC.2016.2530311).
- [33] J. Branke, S. Nguyen, C. W. Pickardt, and M. Zhang, "Automated design of production scheduling heuristics: A review," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 110–124, Feb. 2016.
- [34] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, 2009.
- [35] E. Muñoz, J. M. Cadenas, Y. S. Ong, and G. Acampora, "Memetic music composition," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 1–15, Feb. 2016.
- [36] M. A. Rashid, F. Khatib, M. T. Hoque, and A. Sattar, "An enhanced genetic algorithm for Ab initio protein structure prediction," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 627–644, Aug. 2016.
- [37] H. Al-Sahaf, A. Al-Sahaf, B. Xue, M. Johnston, and M. Zhang, "Automatically evolving rotation-invariant texture image descriptors by genetic programming," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 83–101, Feb. 2017, doi: [10.1109/TEVC.2016.2577548](https://doi.org/10.1109/TEVC.2016.2577548).
- [38] L. Wang, B. Yang, S. Wang, and Z. Liang, "Building image feature kinetics for cement hydration kinetic equation from observed data of Portland cement," *Sci. China Technol. Sci.*, vol. 53, no. 6, pp. 1540–1553, Jun. 2010. [Online]. Available: <https://doi.org/10.1007/s11431-010-3228-9>
- [39] L. Wang, B. Yang, X. Zhao, Y. Chen, and J. Chang, "Reverse extraction of early-age hydration kinetic equation from observed data of Portland cement," *Sci. China Technol. Sci.*, vol. 53, no. 6, pp. 1540–1553, Jun. 2010. [Online]. Available: <https://doi.org/10.1007/s11431-010-3228-9>
- [40] C. Chen, K. Li, A. Ouyang, Z. Tang, and K. Li, "GPU-accelerated parallel hierarchical extreme learning machine on flink for big data," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 10, pp. 2740–2753, Oct. 2017.
- [41] Y. Xun, J. Zhang, and X. Qin, "FiDooP: Parallel mining of frequent item-sets using mapreduce," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 3, pp. 313–325, Mar. 2016.
- [42] Z. Yu, Y. Liu, X. Yu, and K. Q. Pu, "Scalable distributed processing of K nearest neighbor queries over moving objects," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1383–1396, May 2015.
- [43] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.
- [44] M. A. Costa and A. P. Braga, "Optimization of neural networks with multi-objective LASSO algorithm," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Vancouver, BC, Canada, 2006, pp. 3312–3318.
- [45] T. Li and Y. V. Rogovchenko, "Oscillation criteria for even-order neutral differential equations," *Appl. Math. Lett.*, vol. 61, pp. 35–41, Nov. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893965916301367>
- [46] S.-Y. Han, Y.-H. Chen, and G.-Y. Tang, "Fault diagnosis and fault-tolerant tracking control for discrete-time systems with faults and delays in actuator and measurement," *J. Frankl. Inst.*, vol. 354, no. 12, pp. 4719–4738, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0016003217302624>
- [47] T. Li and Y. V. Rogovchenko, "Oscillation of second-order neutral differential equations," *Mathematische Nachrichten*, vol. 288, no. 10, pp. 1150–1162, 2015.



Lin Wang (M'12) was born in Shandong Province, China, in 1983. He received the B.Sc. and master's degrees in computer science and technology from the University of Jinan, Jinan, China, in 2005 and 2008, respectively, and the Ph.D. degree in computer science and technology from the School of Computer Science and Technology, Shandong University, Jinan, in 2011.

He is currently an Associate Professor with the Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan.

His current research interests include classification, hybrid computational intelligence, and mathematical modeling.



Jeff Orchard received the B.Math. degree in applied mathematics from the University of Waterloo, Waterloo, ON, Canada, the M.Sc. degree in applied mathematics from the University of British Columbia, Vancouver, BC, Canada, and the Ph.D. degree in computing science from Simon Fraser University, Burnaby, BC, Canada, in 2003.

Since then, he has been a Faculty Member with the David R. Cheriton School of Computer Science, University of Waterloo. He has published research in image processing and medical imaging. His current

research interests include computational neuroscience, using mathematical models and computer simulations of neural networks in an effort to understand how the brain works, learning methods for deep perceptual networks, network dynamics, decision-making, spatial navigation, and neural population coding.