**Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife 2007)**

# Hebbian Learning using Fixed Weight Evolved Dynamical 'Neural' Networks

Eduardo Izquierdo-Torres
Centre for Computational Neuroscience and Robotics
University of Sussex
Brighton, U.K.
Email: e.j.izquierdo@sussex.ac.uk

Inman Harvey
Centre for Computational Neuroscience and Robotics
University of Sussex
Brighton, U.K.
Email: inmanh@sussex.ac.uk

***Abstract*— We evolve small continuous-time recurrent neural networks with fixed weights that perform Hebbian learning behavior. We describe the performance of the best and smallest successful system, providing an in-depth analysis of its evolved mechanisms. Learning is shown to arise from the interaction between the multiple timescale dynamics. In particular, we show how the fast-time dynamics alter the slow-time dynamics, which in turn shapes the local behavior around the equilibrium points of the fast components by acting as a parameter to them.**

## I. Introduction

Among the insights attributed to Hebb [1] are ideas about how the connection between two neurons should strengthen or weaken according to the correlation of their activity. The main principle behind his reasoning is that if one neuron stimulates another repeatedly then the strength of their link should increase. Hebb's postulate makes intuitive sense when the two components that are connected behave linearly. That is, the more one is active the more the other is as well. In fact, already half a century before him, James in his *Principles of Psychology* had written [2]: "*When two elementary brain-processes have been active together or in immediate succession, one of them, on reoccurring, tends to propagate its excitement into the other*" (p. 566, italics in original).

Following Hebb's insight a plethora of neural network models has been proposed to account for learning behaviors. Generally, these models postulate that learning takes place by modification of the efficacy of individual synapses; or similarly the modification of other parameters in the system (e.g. in [3] learning takes place in the neuron's threshold). This approach assumes that the learning-producing mechanisms and the rest of the internal mechanisms are separate. It also assumes that changes to the weight in the connection between two nodes will lead to changes to their correlation. Although this is the case in simple linear systems, it is not necessarily the case in more complex nonlinear systems (e.g. [4]).

A different approach has been taken within the Artificial Life field. The general idea has been to use continuous-time nonlinear dynamical systems (DS) as models of the internal mechanisms of an agent and to artificially evolve the parameters of these, specifying a behavioral task that the agent has to perform (see [5] for an overview of the approach). This approach allows the experimenter to synthesize systems that, when embodied and situated, result in adaptive interactions with their environment, whilst minimizing the assumptions about what the internal mechanisms need to be. This approach has demonstrated: (a) that learning can arise without parameter changing mechanisms (e.g. synaptic plasticity, threshold plasticity); and (b) that the mechanisms responsible for the learning and the ones responsible for the behavior are not necessarily separate (cf. [6], [7], [8], [9]).

This work involves the use of continuous-time recurrent neural networks (CTRNNs see methods section for further detail). They are convenient models for several reasons. First, they are universal: any smooth dynamical system can be approximated to any desired degree of accuracy by these systems with a sufficient number of nodes [10]. Second, they are tractable systems and some work studying their dynamics exists today (e.g. [11]). Finally, they have become increasingly used for generating adaptive behavior in the artificial life and evolutionary robotics literature.

There are (at least) two ways of conceptualizing CTRNNs: one is to regard them as a set of artificial neurons, interconnected by weights playing the role of artificial synapses. A more general view is to think of them as simple yet generic nonlinear DS, with no particular relationship to neurons and synapses. Here we consider both views, hence the quotes around the word neural in the title.

The motivation for this work is first to understand the effect of variations in the connection strength on the correlation between two nodes, in such nonlinear systems. Second, we are interested in knowing whether a fixed-weight network can generate Hebbian learning-like behavior. We are also interested in knowing whether it is possible to synthesize such systems using simple evolutionary techniques. If it is possible, then we would like to understand the evolved internal mechanisms.

In order to tackle these questions, we artificially evolve CTRNNs to perform a Hebbian learning task. The details of the methods are described in Section II. In Section III, we report on the evolutionary success for different sized circuits. We also describe the performance of the best and smallest successful circuit, providing some detail on the analysis of the evolved internal mechanisms. Towards the end we discuss the implications of our results for the different views on what CTRNNs represent.

**1-4244-0701-X/07/$20.00 ©2007 IEEE**

## II. METHODS

We use evolution to synthesize a continuous-time recurrent neural network that implements some version of a Hebb rule. The conventional rule is as follows: 2 nodes *A* and *B*, if the activations between these two nodes are highly correlated the weight of their connection increases; if not, it decays. However, we do not want to explicitly specify this in terms of what happens to the weights of the connections, since this carries built-in assumptions about the mechanisms and architecture required to do so. We want to define the task at a 'higher' behavioral level.

### A. Hebbian learning task

Suppose we have a black box and we have access to two of its variables: *A* and *B*. We can perturb these using time-varying inputs. The mechanism we seek is such that: (1) if there is no long-term correlation observed between the activity of nodes *A* and *B*, then experimental perturbations to *A* (in the absence of any perturbation to *B*) have no (or little) discernible influence on *B*. That is, the activation of nodes *A* and *B* become uncorrelated. (2) But if there is some long-term correlation between their activities, a stage is reached where experimental perturbations to *A* (in absence of any perturbations to *B*) have a correlated influence on the activation of *B*.

The task has two phases (see Figure 1). During the first phase time-varying perturbations are applied to the input of nodes *A* and *B*. This phase lasts 100 units of time and we will refer to it as *training*. During the second phase a time-varying perturbation is applied only to node *A*, also for 100 units of time. This phase we will refer to as *testing*. Before each phase a time delay is introduced where no perturbations are applied.

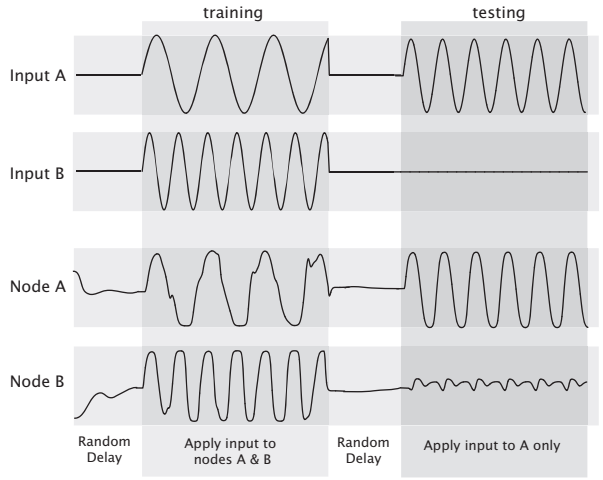


Fig. 1. The structure of an individual Hebbian learning trial. The trial starts with a random delay, where no input is applied. Then, inputs are applied to nodes *A* and *B* (training phase). The inputs can be correlated or uncorrelated. In the example shown, the perturbations have different frequencies. Here we show the minimum and maximum frequencies used: $f_A = 1$ and $f_B = 2$. This is followed by another delay. Perturbations are then applied to node *A* only (testing phase), while the correlation between the activities of *A* and *B* is evaluated.

The length of the delay is chosen at random between [10, 20] units of time. The time-varying input is a sine wave, with variable frequencies: $I_t = sin(kft)$, where $I_t$ is the external input at time $t$; $k$ (0.2) is a constant of proportionality; and $f$ determines the frequency of the wave, chosen uniformly at random between $[1, 2]$. The perturbations to the system during training can be of two different types: correlated or uncorrelated. For the correlated training, sine waves with similar frequencies are used, randomly chosen every time so that $|f_A - f_B| < 0.05$, where $f_i$ is the frequency given as input to node $i$. For uncorrelated training, perturbations are given with sine waves of different frequencies, also randomly chosen. It is ensured that frequencies are not too similar during uncorrelated training by constraining $|f_A - f_B| > 0.2$.

### B. Dynamical neural network

For the 'black box', we use a continuous-time recurrent neural network (CTRNN) with the following state equation [11]:

$$\tau_i \dot{y}_i = -y_i + \sum_{j=i}^{N} w_{ji}\sigma\left(y_j + \theta_j\right) + s_i I_i \qquad (1)$$

where $y$ is the activation of each node; $\tau$ is its time constant; $w_{ji}$ is the strength of the connection from the $j^{th}$ to the $i^{th}$ node; $\theta$ is a bias term; $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic activation function; $I$ represents the external input (e.g. from a sensor); $N$ represents the number of nodes in the network; and $s_i$ is a 'sensory' weight for the external input. Only the two nodes that receive external input (*A* and *B*) have sensory weights. In simulation, node activations are calculated forward through time by straightforward time-slicing using Euler integration with a time-step of $0.1$. The network is fully connected.

### C. Evolutionary algorithm

The parameters of each circuit (i.e. weights, biases and time-constants for each node) are evolved using a version of the microbial genetic algorithm [12]. These are encoded in a genotype as a vector of real numbers over the range [0, 1]. Offspring of microbial tournaments are generated as a mutation of the winner of the tournament (i.e. no recombination). The mutation is implemented as a random displacement on every gene drawn uniformly from a Gaussian distribution with mean 0 and variance 0.05. Each gene is forced to be in [0, 1]: when a mutation takes a gene out of this range it is reflected back. The offspring replaces the loser of the tournament. Genes are mapped to network parameters linearly between [-10, 10] for biases and inter-node weights, and to [0, 10] for sensory weights. Time-constants are mapped exponentially to $[e^0, e^5]$. The size of the population used is 50. We define a generation as the time it takes to generate 50 new individuals. A minimal 1D wrap-around geography [13] with demes of size 10 is used: such that only nearby individuals can compete in tournaments.

### D. Fitness evaluation

The fitness of a circuit is obtained by maximizing the correlation coefficient on trials where inputs have similar

frequencies and minimizing the absolute value of the correlation coefficient on trials where the inputs have different frequencies. We define the correlation of two time series, $Y$ and $X$, during phase $p$ ($p = 1$ for training and 2 for testing) using the Pearson product-moment correlation coefficient:

$$r(Y, X, p) = \frac{\sum_{i=ts}^{te}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=ts}^{te}(x_i - \bar{x})}\sqrt{\sum_{i=ts}^{te}(y_i - \bar{y})}} \quad (2)$$

where $ts$ and $te$ denote the start and end of phase $p$; $x_i$ and $y_i$ are the values in the time series X and Y; $\bar{x}$ and $\bar{y}$ are the average activations of X and Y, respectively. The correlation is calculated only if both of the standard deviations are nonzero, otherwise correlation zero is given.

The fitness is composed of six parts and it is calculated by evaluating the circuit on $R$ independent trials. The fitness is, thus, given by the multiplication of the averages of each component as follows:

$$f = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot (1 - \bar{d}) \cdot (1 - \bar{e}_1) \cdot (1 - \bar{e}_2) \quad (3)$$

The first three components measure the correlation between the input signals and the activities of the nodes: $a = r(A, I_A, 1)$ is the correlation between the input to *A* and the activity of *A* during training; $b = r(B, I_B, 1)$ is the correlation between the input to *B* and the activity of *B* also during training; $c = r(A, I_A, 2)$ is the correlation between the input to *A* and the activity of *A* during testing. We would like to maximize these. Negative values are clipped to 0.

The last three components measure the relation between the correlations of the two nodes during training, $x = r(A, B, 1)$, and testing, $y = r(A, B, 2)$, phases: $d = |x - y|$ is the absolute difference between the correlation of nodes *A* and *B* during training and testing; $e_1 = \sqrt{(1-x)^2 + (1-y)^2}$ for trials where similar frequencies are applied we would like the activities of *A* and *B* to be highly correlated; $e_2 = \sqrt{x^2 + y^2}$ for trials where different frequencies are applied we would like the activities to be as uncorrelated as possible. These components we seek to minimize.

### *E. Incremental evolution*

During the first stage of evolution, individuals in the population are evaluated only on one consecutive trial after initialized. This is repeated 100 times per circuit ($R = 100$). During this stage the state of the nodes are initialized to 0 at the start of the trial. Once any individual in the population obtains a fitness greater than 0.75, the task becomes harder. During the second stage, the nodes are initialized at random in [-10, 10] and the task involves evaluating the circuit on two consecutive trials without reinitializing the circuit's state. To keep evaluation times similar $R = 50$, also the variance in the mutation is decreased to 0.01. The third and last stage consists of 5 subsequent trials and $R = 20$.

## III. RESULTS

### *A. Preliminary study: effect of varying weights on correlation*

Before we begin the main part of our results it will be useful to understand what is the effect of changing the strength of the connection from node *A* to *B* on the correlation of their activities in nonlinear dynamical networks. In order to gain some insight into this question, we analysed 1000 randomly chosen 2-node CTRNNs from the same range as that used for the evolutionary experiments. For each, we evaluated the correlation between the activities of both nodes for 141 weights connecting *A* to *B* evenly spaced between -7 and 7. Each circuit is first integrated for 200 units of time from a random initial activation (between [-10, 10]). This allows the system to settle in its long-term state. The correlation is measured for the following 100 units of time, as node *A* is perturbed by a sine wave with a randomly chosen frequency. This is repeated 1000 times for each circuit and for each weight. The thick black line in Figure 2 shows the mean correlation over all generated circuits. As can be observed, when the weight is negative there is likely to be some negative correlation between the two nodes. Analogously, when the weight is positive there is a good chance of some positive correlation. Note that the correlation, on average, does not grow particularly stronger or weaker as the absolute value of the weight changes. Also, note that (on average) the correlation drops to 0 only when the weight is exactly 0.

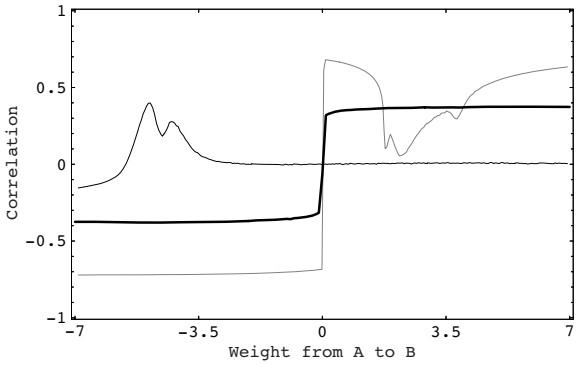


Fig. 2. Effect of weight changes on the correlation between node activity. Average correlation coefficient as the weight from node *A* to *B* is varied (thick line). Average taken from 1000 randomly generated 2-node CTRNNs. Example effects of the weight on the correlation from two of those circuits (thin black and gray lines).

However, the most relevant insight in this preliminary study is not to be obtained from the average over hundreds of circuits, but in particular examples of these nonlinear systems. The thinner lines show two examples out of all randomly generated. In the example shown in the thin black line, the activities are most correlated at a particular inhibitory value for the weight, but decreases to near zero correlation when weakened until there is virtually no weight. Interestingly, the activity remains almost completely uncorrelated after strengthened. Also, increasing the strength of the inhibitory connection results in slightly stronger negative correlations. In the second example (gray line) the correlation is zero when there is no weight. The slightest excitatory weight makes the activity highly correlated. However, increasing the strength results in less correlation, not more. This is until a minimum is hit

and the correlation starts going up again. On the other hand, setting the weight to any inhibitory strength results in a steady strong negative correlation, regardless of the strength. These individual examples show just how complex and unintuitive the relation between increasing/decreasing the weight and the correlation between the activities of both nodes can be.

### *B. Evolutionary success using small circuits*

Is it possible to evolve a non-weight-changing circuit to perform Hebbian learning behavior? To answer this question we performed evolutionary runs using 3 and 4 node circuits. No 3-node circuits out of 20 advanced to the second stage. Much more successful were populations of circuits with extra components. After 10000 generations, we found $11/20$ 4-node populations that reached the last stage of the incremental protocol. The best fitness of $9/11$ of those populations was greater than 0.78 at the end of the runs. Fitness greater than 0.78 is regarded as successful from analysis of their performance. Figure 3 shows the run that led to this circuit. During the first hundreds of generations the population's fitness remains in stasis around 0.28 Around generation 1700 the population finds a portal to higher fitness. At generation 3141 (first dashed line) the task changes to the second stage after the best performing circuit in the population reaches the 75% threshold. At this point, the fitness of the best individual (as well as the average) drops significantly. This means that the circuits in the current population had not generalised to more than one consecutive trial. After less than 1000 generations more, the circuit reaches (second dashed line) again the 75% threshold on the task involving 2 consecutive trials. At this point the task changes to 5 consecutive trials and the circuit's performance does not drop as much, meaning that the circuit generalized well. All of the features mentioned for this evolutionary run are qualitatively similar for the rest of the successful runs. It is the best circuit evolved in this run that we analyse in detail in the rest of this paper. The best evolved network achieved a fitness of 0.83 on a more thorough fitness evaluation performed at the end of its evolutionary run. This involved $10^6$ tests using randomly chosen frequencies, time delays and initial activations on 5 consecutive training and testing trials. This was performed using a time-step of integration an order of magnitude smaller than that used during evolution (0.01) to avoid time integration errors.

### *C. Best evolved 4-node circuit*

What can we say about the best evolved architecture? A graphical depiction of the parameters are depicted in Figure 4. There are a number of important things to note. First, components *A*, *B* and *D* are fast-acting, while the *C* component is significantly slower. Node *A* has a high threshold and strongly inhibits itself. Nodes *A* and *B* have similarly strong connections to *D*, but one of them excitatory and the other inhibitory. *D* has virtually no self-connection but a strong excitatory connection to the slower node in turn. The slow node has a very low threshold and highly excites itself. In general, the connections between *A*, *B* and *C* are very weak.

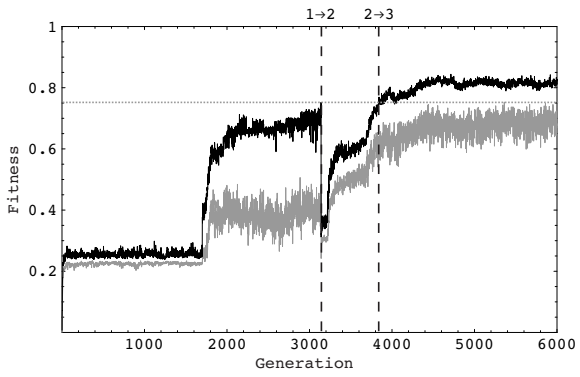


Fig. 3. Plot of the best (black) and average (gray) fitness of the population versus generation resulting in the best evolved 4-node circuit. Only the first 6000 generations are shown. The transition between the stages in the incremental evolutionary protocol are marked with dashed lines. Transitions occur when the best fitness exceeds 0.75 (dotted gray line). The evolutionary run shown here is representative of a typical successful one.

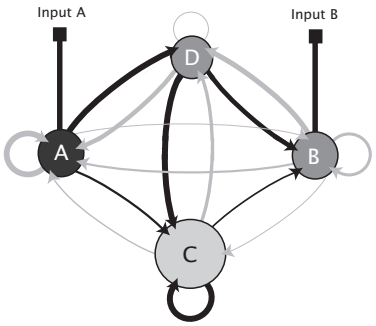


Fig. 4. A graphical depiction of the best evolved CTRNN. The shading scheme is similar to that used in [14]. Nodes are shaded according to their bias, with higher threshold nodes (which require more excitation to activate) shaded darker. Excitatory connections are shaded black and inhibitory connections are shaded gray, with the width of the line proportional to the strength of the connection. The time-constant parameter for each node is represented by the size of the circle, with larger circles representing slower nodes (integrating over longer periods of time). The slow node (*C*) should be displayed 7 times larger than the fast nodes (*A*, *B* and *D*), but this is impractical for this figure.

Thus, most of their interactions are mediated by node *D*. These factors play an important role in the generation of the circuit's learning behavior as we will see ahead.

### *D. Performance of best evolved circuit*

What does the circuit do? In order to visualize the circuit's behavior, we can record the activity of each of the components of the circuit as well as the external perturbations applied during an example trial. Figure 5 shows this in a trial where the input to *A* and *B* are first correlated and later when the input is uncorrelated, separated by a dashed line. We can observe how, after correlated perturbations have been applied, the activities of *A* and *B* remain highly correlated: every time node *A* 'fires', so does node *B*. In contrast, after uncorrelated perturbations are applied, the activities of these nodes becomes highly uncorrelated. In the figure we can also see how the slower node (*C*) adopts different states for each of the scenarios; similarly node *D* adopts different patterns of activity altogether.

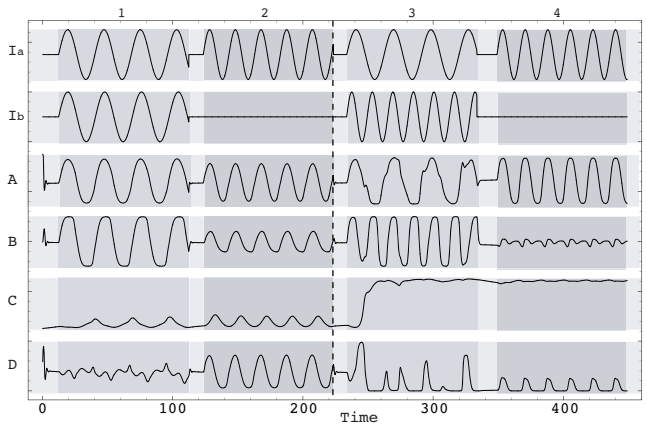


Fig. 5. Example sequence trial. Top two rows show the external perturbations to nodes *A* and *B*. The bottom four rows show the sigmoided activity of each of the nodes in the circuit. The areas shaded in light gray depict training phases. The areas shaded in darker gray depict testing phases. Time proceeds from left to right along the horizontal axis. The circuit is first trained using similar time-varying perturbations (1). After a delay, the circuit is evaluated while receiving a different frequency input on *A* (2). The correlation between the activations of node *A* and *B* is 0.98. Subsequently, the circuit is trained on two random but different frequencies (3). When evaluated (4), the correlation between the activities of *A* and *B* is -0.02. The 'behavioral connection' between nodes *A* and *B* has strengthened during the first part of the trial (1, 2) and then weakened during the second part of the trial (3, 4).

How well does the circuit learn? In Figure 6a we visualize the relation between the correlation of nodes *A* and *B* during training (horizontal axis) versus the correlation during testing (vertical axis). The figure shows 1000 examples using randomly chosen frequencies, time-delays and initial activations for similar (black points) and different (gray points) time-varying perturbations. As can be observed, application of similar frequency inputs generates correlation in the activities of *A* and *B* during training and this leads to correlation during the test phase, even though no input is applied to *B*. Similarly, presentation of different frequencies generates uncorrelated activity during training, which in turn leads to uncorrelated activity during testing.

There are two other questions of interest in the context of a circuit that learns. The first is to do with the plasticity of the circuit. How many times can the link between *A* and *B* re-strengthen and re-weaken? Does the plasticity harden after some trials? Figure 6b shows the correlation on the $i^{th}$ trial following a correlated training phase (black triangles) or following an uncorrelated training phase (gray boxes), when all previous trials were randomly chosen to be either correlated or uncorrelated with 50% chance. As can be seen, the circuit is capable of maintaining high performance even after 100 re-learning experiences. This demonstrates generalized behavioral plasticity.

The second question of interest concerns the circuit's memory. How sensitive is the circuit to longer time delays? We address this by testing its performance on a bigger range of time delays than it was evolved for. Figure 6c shows the average correlation coefficient for delays between 0 and 100

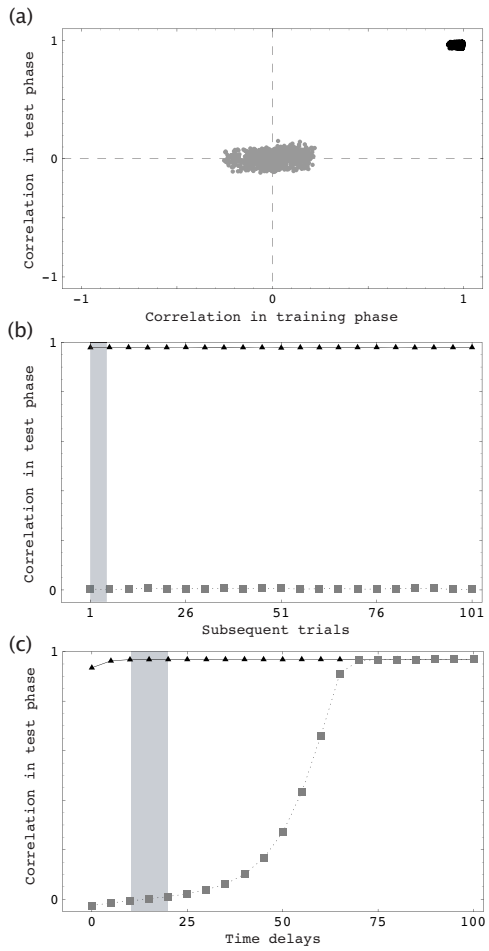


Fig. 6. Performance of best circuit. **(a)** Correlation during training versus correlation during testing, for trials with different time-varying perturbations (gray) and similar perturbations (black). **(b)** Relearning ability. Each point shows the average correlation coefficient on the $i^{th}$ trial over $10^5$ different runs. Black triangles correspond to the correlation coefficients after correlated training (good performance should be near 1). Gray squares correspond to the correlation coefficients after uncorrelated training (good performance should be near 0). The gray vertical bar represents the range of trials the circuit was evolutionary trained for [1-5]. **(c)** Robustness to time-delays. Average correlation coefficient of the activity of node *A* and *B* during the testing phase for $10^5$ repetitions with randomly chosen train and test inputs, for correlated input (black line) and uncorrelated input (gray line) using different time-delays between training and testing phases. The gray area represents the range of delays the circuit was evolutionary trained on [10-20].

units of time. The circuit is evaluated during the first trial $10^5$ times for correlated as well as uncorrelated trials using different initial activations and frequencies of sine waves. As can be seen, the memory for the learned correlated input does not decay across the range measured. In contrast, as the time range increases after uncorrelated training, the activities of the two nodes cease to be uncorrelated gradually. Eventually, it becomes indistinguishable from the activity after correlated training. In other words, after extended delays, the circuit 'forgets' how to 'behave' following uncorrelated training; yet 'remembers' perfectly well what to do after correlated training. We will come back to this point in the dynamical analysis.

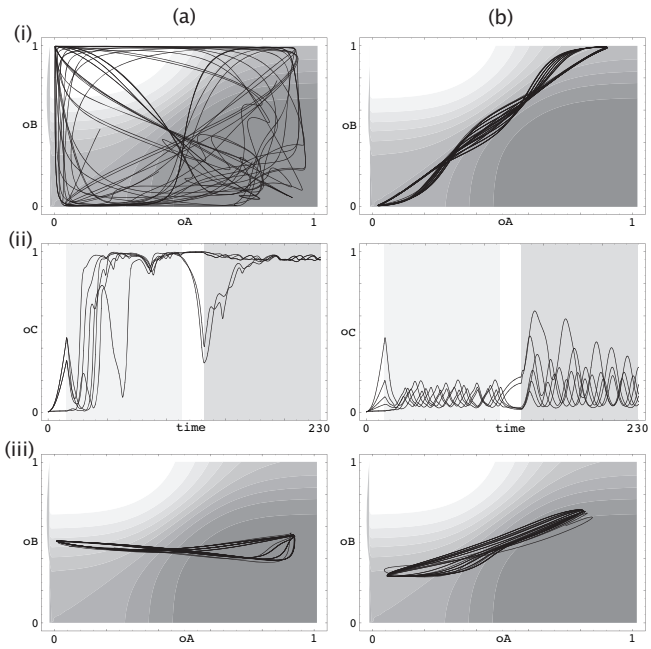


Fig. 7. Interaction of multi-timescale dynamics during *uncorrelated* training **(a)** and *correlated* training **(b)**. **(i)** Activity in output state space of the fast nodes *A* and *B* during a set of training examples with (a)(i) different-frequency perturbations to both nodes and (b)(i) same frequency perturbations. The shades of gray represent the state of node *D* for each possible state of nodes *A* and *B*. White represents *D*=0 and the darkest shade of gray represents *D*=1. **(ii)** Activity of slow dynamics (output of node *C*) through time for the same set of training examples. The region shaded in light gray corresponds to the training phase and the region in darker gray corresponds to the testing phase. Delays are not shaded. **(iii)** Activity in state space of fast nodes after training, while node *A* is being perturbed (testing phase).

### *E. Analysis of the multiple time-scale dynamics*

How does this nonplastic circuit produce Hebbian learning behavior? We know dynamics are occurring at two different time-scales. The inputs change the activations of *A* and *B* almost instantaneously because both are fast acting. These change *D* in turn. We also know that *C*'s slower dynamics is influenced mostly by the activity of *D*. In order to understand this circuit's mechanisms, we need to understand (a) how the fast dynamics influence the slow dynamics and (b) how the changed slower dynamics affect the fast dynamics, in turn. In order to address these questions, we visualize the activity of the circuit in its fast and slow state spaces using a set of trials with different frequencies covering the whole range. We are particularly interested in the differences between the structure of the activities when the perturbations during training are correlated and when they are not.

Figures 7ai and 7bi show the trajectories of the sigmoided output of nodes *A* versus *B* during the last 80 units of time of the training during uncorrelated perturbations (7ai) and during correlated perturbations (7bi). The shades of gray represent the state of node *D* for each possible state of nodes *A* and *B*. The cancellation of activities from *A* and *B* during correlated activity means that node *D* is maintained in a mid-level of activation (7bi). In contrast, during uncorrelated activity node *D* oscillates with the difference between nodes *A* and *B* (7bi).

How does this affect the slower component? Figures 7aii and 7bii show the activity of *C* throughout each of the trials, the light gray region represents the training phase and the darker gray represents the testing phase. During uncorrelated input (7aii), the firing of node *D* results in the saturation of *C*. In contrast, during correlated input (7bii), the absence of activity in *D* allows *C* to rest in its off state. As a result of training, *C* is left in two different states. More importantly, this difference in state persists during the testing phase (see darker shaded regions in 7aii and 7bii).

The long term changes in *C* modulate the fast dimensional dynamics in turn. Figures 7aiii and 7biii show the activity of the fast nodes during the testing phase. Even though the perturbations to *A* at this point are the same, the patterns of activity are significantly different. In fact, the only difference in the system is maintained by *C*. The different patterns correspond to the correlation coefficient in a straightforward manner: activity along the diagonal means that both nodes are firing in phase (7biii). This results in an average correlation coefficient of 0.99. In contrast, activity parallel to the horizontal axis corresponds node *A* having little or no influence on node *B* (7aiii), resulting in an average correlation coefficient of 0.01. Finally, it is important to note that the *C* component does not *only* modulate the dynamics of *A* and *B*. It also plays an active role in the generation of correlated activity, as can be seen from Figure 7bii.

### *F. Dynamical analysis of best evolved circuit*

We can use tools from DS theory to further understand the evolved mechanisms. We will focus on two questions: what are the equilibrium points of the evolved system? and how do the trajectories flow into these equilibrium points for each of the different scenarios?

The global dynamics of the evolved system are rather simple: with just one equilibrium point (EP). When unperturbed, this point attracts all trajectories in the phase plane. In the context of the Hebbian learning task, it is near this EP where the system arrives at after training with correlated input. Hence, determining the local behavior (how the trajectories proceed into this point) corresponds to the system's performance during the testing phase after correlated training. We will refer to this as the correlated EP.

What about the system after uncorrelated training? Is there not a long-term behavior associated with this scenario? There isn't. At least not in the strict sense. This explains why, when longer time delays are introduced, the performance of the system falls back into what it would be had been trained with correlated input. However, it is here where the system's operation on two rather different time-scales allows it to have a 'temporary equilibrium point'. Although for the time-scale of the slow node this point is merely a transient, for the much faster subset of the dynamics the state will arrive at this point in the relative long term. From the previous section, we know *C* is near one after training with uncorrelated input. Hence,

its influence on the states of the fast nodes is nearly constant. We can approximate the relatively-long-term behavior of the system after training with uncorrelated signals by studying the dynamics of the modified circuit that assumes $C$ is always one. Similar to the correlated case, the reduced circuit has only one EP. This is also a stable attractor, shifted on the $AB$ plane. We will refer to it as the uncorrelated EP.

While there is no global difference in the dynamics between the two scenarios, we can still ask about the local behavior around their associated EPs. How do the trajectories proceed in their vicinities? This is determined by the eigenvalues of the Jacobian matrix of the system evaluated at these points [11]. Both EPs are *mixed*: the trajectories proceed in a straight line towards the fixed point in some dimensions but proceed in spirals in other. Figures 8a and 8b show the flow around the correlated and uncorrelated fixed points, respectively. It is the qualitative difference in the local behavior around the relative-long-term behavior of the two EPs that generates the two different correlations patterns between $A$ and $B$.

Next we visualize the system's dynamical properties in a 3D projection of its 4D activation space. Figure 8c shows the circuits EPs and the behavior of the system around its EPs as it is systematically perturbed through node $A$. The perturbations are in the form of positive and negative parts of a sine wave with different frequencies. The black disk represents the correlated EP. Although the uncorrelated EP only has $A$, $B$ and $D$ coordinates, we can visualize it where the sigmoided output of $C$ nears 1. This is represented by the gray disk. The trajectory in gray depicts the flow of the system from the uncorrelated to the correlated EP after longer delays. The black trajectories to the left depict the system in activation state space when perturbed away from the correlated EP. The state of the system remains at all times very near a 2-dimensional plane that cuts the $A$ and $B$ plane diagonally. This results in their activity being correlated. As remarked earlier, it is important to note that $C$ plays a role in maintaining the activities of $A$ and $B$ on this plane. The black trajectories around the uncorrelated EP for the same pattern of perturbations show a completely different structure. Most importantly the activity in the $B$ plane is minimized. This results in uncorrelated activity. Hence, the 'weakening' and 'strengthening' of the behavioral connection between nodes $A$ and $B$ corresponds to the shift between these two distinct regions of dynamics: the global EP (black disk) and the 'temporary' EP (gray disk).

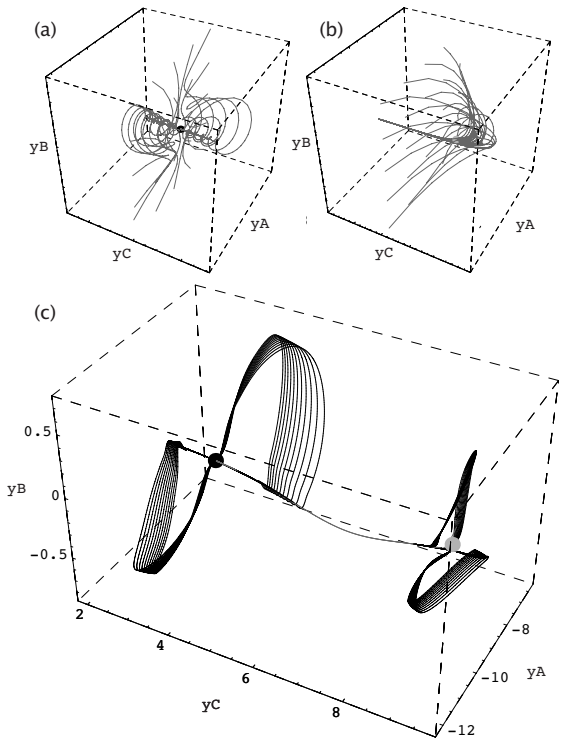


Fig. 8. 3D projection of the circuit's state space. **(a)** Flow (gray trajectories) around the correlated EP (black disk). **(b)** Flow around the uncorrelated EP (light gray disk). **(c)** The light gray disk represents the system's temporary EP associated with the state after uncorrelated training. The black disk represents the system's EP associated with the state after correlated training. Trajectories in black depict the behavior of the system in activation state space when systematically perturbed away from the equilibria through node $A$. The gray trajectory depicts the movement between the two points corresponding to the strengthening and weakening of the 'connection' between $A$ and $B$.

### *G. Evolved time parameters in successful circuits*

Most of the analysis in this paper deals only with the internal mechanisms of the best evolved and smallest possible circuit (sections C through F). Is there something in common in the parameters of all successfully evolved circuits? Due to the high-dimensionality of parameter space, we will not attempt to characterize successful circuits in general. Instead, we will focus only on the time-constants because of their relevance in the analyzed circuit. Figure 9 shows the average and standard deviation of the evolved time-constants for all successful 4-node circuits. The $C$ node was arbitrarily chosen as the slowest of the two extra components. As can be seen similar to the analyzed circuit, all $A$, $B$ and $D$ nodes evolved to be as fast as possible. In contrast, one of the extra components ($C$) is, on average, much slower acting. More importantly, we observed the behavior of two of these other successful circuits and found much similar mechanisms at play.

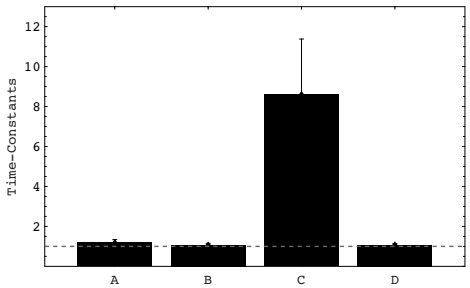


Fig. 9. Evolved time-constants for the successful 4 node circuits. Averages for each of the nodes indicated by the bars and standard deviations with the lines. Dashed gray line indicates the smallest (i.e. fastest) time-constants allowed.

## IV. Summary and Discussion

In this paper, we have (a) shown how unintuitive the relation can be, in nonlinear systems, between changing a weight and the correlation between the two nodes connected by that weight; (b) demonstrated that CTRNNs with fixed weights can be evolved to produce Hebbian-like learning; and (c) analyzed the behavior and dynamics of the best and smallest evolved circuit. Learning is shown to arise from the multi-timescale dynamics. In particular, the evolved mechanism requires two extra components: one fast and the other slow. The fast node fires when the activities of nodes $A$ and $B$ are different, and the firing of that node influences the slow one which then settles into two different persistent states. The changed slow component, by acting as a parameter to the fast components, shapes their patterns of activity in turn.

In connectionist artificial neural networks, it is common for all neurons to be constrained to operate at a single time-scale. It is for this reason that longer term changes to the behavior of the system have to be introduced as additional parameter changing rules. These rules are generally applied to the weights of the connections. One example is backpropogation learning, another is Hebb's rule. It is now known that neurons can act over a range of different time-scales, and so can changes in synaptic efficacy. Furthermore, the time-scales of activity of the two can overlap (see [15] for a review). If we think of CTRNNs as models of interconnected neurons, then we are essentially allowing neurons to act in a range of different time-scales. Under this view, this work suggests: (a) that Hebbian forms of learning can arise without synaptic plasticity; and perhaps more importantly: (b) that it can do so via neurons (or sets of neurons) interacting over multiple time-scales. There are at least two ways in which this is feasible in nervous systems: single cell (i.e. changes to their intrinsic properties) and network (i.e. active reverberation in recurrently connected circuits) mechanisms. Both are likely to co-operate in generating persistent activity [16]. The circuit analyzed in this paper resembles more closely the former.

In the broader 'CTRNN as a dynamical system' view, each of the nodes can evolve to instantiate any possible component, with certain intrinsic properties and with particular rules for interacting with the rest of the nodes. Some nodes could resemble, for example, neurons; others: population of neurons, a range of neurotransmitters, synapses, and so on. In fact, these need not be constrained to things you find in brains, but also things in other systems: plants, for example. In this perspective, our work suggests that Hebbian forms of learning can be present in a broader range of systems, regardless of whether they have synapses and neurons or not. The interesting question becomes: can we extract general principles about the properties of the components required for learning? More importantly still, we believe further understanding of these systems could lead to practical suggestions to the community studying learning and memory in living organisms. Suggestions to study components, processes or even organisms not generally considered.

This work raises several interesting directions for future research. First, the task could be further developed in several directions. Of particular interest would be to alter the task to allow a continuum of different strengths of correlation. At present, it suffices for the system to have just two modes of long-term dynamics: fully correlated or nil correlated. Also interesting would be to increase the range of delays experienced to encourage the evolution of systems that do not easily forget. Second, we need to understand better how the slower components relate to synaptic plasticity mechanisms. One question that we can ask is, can we extract a weight-changing rule (or any parameter-changing rule) from the evolved dynamics of one of these circuits? Finally, although all of the successful circuits evolved slower-acting extra components, slow components are not the only way to generate slower-time dynamics, as remarked earlier. An interesting direction of research for the future is to encourage the evolution of reverberatory dynamics by evolving larger systems and/or constraining components to a smaller range of time-constants.

## Acknowledgment

We thank Thomas Bührmann, Peter Fine, and Eldan Goldenberg for their feedback on an earlier draft of this paper.

## References

[1] D. Hebb, *The Organization of Behavior*. Wiley & Sons, 1949.
[2] W. James, *The Principles of Psychology*. MacMillan & Co., 1901, vol. 1.
[3] G. Tesauro, "A plausible neural circuit for classical conditioning without synaptic plasticity," *PNAS*, vol. 85, pp. 2830–2833, 1998.
[4] J. E. Rubin, "Surprising effects of synaptic excitation," *Journal of Computational Neuroscience*, vol. 18, no. 3, pp. 333–342, 2005.
[5] I. Harvey, E. Di Paolo, R. Wood, M. Quinn, and E. A. Tuci, "Evolutionary robotics: A new scientific tool for studying cognition," *Artificial Life*, vol. 11, no. 1–2, pp. 79–98, 2005.
[6] B. Yamauchi and R. Beer, "Sequential behavior and learning in evolved dynamical neural networks," *Adaptive Behavior*, vol. 2, no. 3, pp. 219–246, 1994.
[7] E. Tuci, M. Quinn, and I. Harvey, "An evolutionary ecological approach to evolving learning behavior using a robot based model," *Adaptive Behavior*, vol. 10, no. 3/4, pp. 201–221, 2003.
[8] E. Izquierdo-Torres and I. Harvey, "Learning on a continuum in evolved dynamical node networks." in *Proc. of the Tenth Int. Conf. on the Simulation and Synthesis of Living Systems*. MIT Press, 2006.
[9] P. Phattanasri, H. Chiel, and R. Beer, "The dynamics of associative learning in evolved model circuits," *Adaptive Behavior*, Submitted.
[10] K. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks." *Neural Networks*, vol. 6, pp. 801–806, 1993.
[11] R. Beer, "On the dynamics of small continuous-time recurrent neural networks," *Adaptive Behavior*, vol. 3, no. 3, pp. 469–509, 1995.
[12] I. Harvey, "Artificial evolution: a continuing SAGA," in *Evolutionary Robotics: From Intelligent Robots to Artificial Life.*, T. Gomi, Ed. Springer-Verlag LNCS 2217, 2001.
[13] L. Spector and J. Klein, "Trivial geography in genetic programming," in *Genetic Programming Theory and Practice III*, T. Yu, R. Riolo, and B. Worzel, Eds. Kluwer Academic Publishers, 2005, pp. 109–124.
[14] R. Beer, "The dynamics of active categorical perception in an evolved model agent," *Adaptive Behavior*, vol. 11, no. 4, pp. 209–243, 2003.
[15] M. Toledo-Rodriguez, A. El Manira, P. Wallen, G. Svirskis, and J. Hounsgaard, "Cellular signalling properties in microcircuits," *Trends in Neurosciences*, vol. 28, no. 10, pp. 534–540, 2005.
[16] G. Major and D. Tank, "Persistent neural activity: prevalence and mechanisms," *Current Opinion in Neurobiology*, vol. 14, pp. 675–684, 2004.