Hebbian reinforcement learning in a modular dynamic network

Emmanuel Daucé
UMR Movement and Perception
Faculty of sport sciences
University of the Mediterrannean
163 avenue de Luminy, CP 910
13288 Marseille cedex 9
France
dauce@esm2.imt-mrs.fr

Abstract

We present a multi-population dynamic neural network model with binary activation and a random interaction pattern. The weights parameters have been specified in order to distinguish excitatory populations from inhibitory populations. Under specific parameters, we design functional modules composed of two populations, one of excitatory neurons, one of inhibitory neurons. Such modules are found to display a weak chaotic activity, and to react toward incoming stimulations with increasing synchronization. We also present the design of topologically structured neural maps. We then combine such modules for the design of a perception/action network composed of one sensory module and two concurrent motor modules. Such a network is coupled with the dynamics of an inverted pendulum, showing spontaneous phase transitions toward various attractors, each attractor corresponding to a particular structural coupling within the agent/environment system. This spontaneous versatility is then exploited in a reinforcement learning paradigm where two separate reinforcement path are defined, one for the positive rewards, the other for the negative rewards. The learning experiment shows a fast adaptation to the constraints, followed by a slower phase where the behavior is improved. No degradation of the behavior is found for continuing learning, i.e. the learned behavior is preserved for long lasting time.

1. Introduction

Brain structures at different scales are in majority composed of self-feeding connections, and the connections between different structures/areas are always reciprocal (see for instance (Hupé et al., 1998) for the visual cortex). This massive recurrence suggests a major role of self-feeding dynamics in the processes of perceiving, acting, and learning, and more generally in maintaining the organism alive. Lot of simulation and theoretical work has been done about the spontaneous dynamic properties of massively connected recurrent neural networks (Sompolinsky et al., 1988, Doyon et al., 1993, Cessac, 1995, Ben-Yishai et al., 1995, Hansel and Sompolinsky, 1996, Camperi and Wang, 1998, Wang, 1999, Compte et al., 2000, Moynot and Samuelides, 2002, Izhikevich, 2003, Daucé, 2004), whereas lot remains to be done on the question of the *learning* processes taking place in such networks (Hertz and Prugel-Bennett, 1996, Daucé et al., 1998, Daucé et al., 2002).

Reinforcement learning offers a relevant framework for the study of such learning processes in large recurrent networks. Reinforcement learning is a well-known semi-supervised learning paradigm, where the environment occasionally sends to the agent reinforcement signals, which may be positive (reward) or negative (penalty). The aim of a reinforcement learning algorithm is to modify the agent's behavior in order to maximise the reward (and/or to minimize the penalty). This paradigm is of course well adapted in case an agent has to explore an unknown environment, where foes and traps have to be avoided and feeding sources and strategies have to be found. The agent is supposed to adapt by its own means, i.e. one can not select agents over generations, like in genetic algorithms. Every reinforcement learning method needs an *exploratory* process in order to test and experiment various structural couplings and a *selection* process in order to retain the most appropriate ones. For instance, in statistical reinforcement methods(Sutton, 1988, Watkins and Dayan, 1992), the transitions probabilities are initially random and uniform in order to explore the largest scope of interactions, so that learning tends to reduce the uncertainty, through

the selection of the most appropriate transitions.

The aim of this paper is to use some of the native dynamical properties of large networks, such as chaos, attractor switching and synchronization, as the core of the exploratory process. We also want to bring reinforcement methods one step closer to biological plausibility.

The paper is organized as follows. We first present the formalism of the multi-populations neural model, composed of several populations, which can either be purely excitatory or purely inhibitory (section 2.). We then present in section 3. some simulations showing the spontaneous dynamics of different "modules" composed of excitatory and inhibitory populations. Section 4. presents the controller architecture. This controller is composed of 3 functional modules : one sensory module an two motor modules. The last section presents our Hebbian reinforcement method, through different protocols (closed loop and on-line protocols), and simulations giving the convergence properties under various parameter settings.

2. A dynamic network model

The full model formalism is given here. We tried by the way to make this paper self-explanatory and testable[1].

2.1 Model setting

Our model is a discrete-time dynamical system with parallel update, where the state of the system at time t both depends on the previous state of the system $\mathbf{x}(t-1)$ and on the input $\mathbf{u}(t-1)$, i.e

$$\mathbf{x}(t) = f(\mathbf{x}(t-1), \mathbf{u}(t-1)) \tag{1}$$

where $\mathbf{x}$ is a state array, $\mathbf{u}$ is an input array and f is a global (nonlinear) operator. One can notice that our system is deterministic as soon as the input signal is set according to a deterministic process.

A network is defined as a pool of P interacting populations of neurons, of respective sizes $N^{(1)}, ..., N^{(P)}$. The global number of neurons is $N = \sum_{p=1}^{P} N^{(p)}$. The synaptic weights from population q toward population p are stored in a matrix $\mathbf{J}^{(pq)}$ of size $N^{(p)} \times N^{(q)}$ (possibly sparse). The state vector of population p at time t is $\mathbf{x}^{(p)}(t)$, of size $N^{(p)}$. The initial conditions $x_i^{(p)}(0)$ are set according to a random draw in $\{0, 1\}$.

At each time step $t \geq 1$, $\forall (p, q) \in \{1, .., P\}^2$,

$$\mathbf{h}^{(pq)}(t) = \mathbf{J}^{(pq)}\mathbf{x}^{(q)}(t-1) \tag{2}$$

is the *local field* array of population q toward population p.

We also consider spatio-temporal input signals $\mathbf{u}^{(p)} = \{\mathbf{u}^{(p)}(t)\}_{t=1..+\infty}$, where $\mathbf{u}^{(p)}(t)$ is an input array of size $N^{(p)}$. The input $\mathbf{u}^{(p)}(t)$ acts like a bias on each neuron[2]. Then, the global equation of the dynamics is :

$$\forall t \geq 1, \forall p \in \{1, ..., P\}$$
$$\mathbf{x}^{(p)}(t) = H\left(-\theta^{(p)} + \mathbf{u}^{(p)}(t-1) + \sum_{q=1}^{P} \mathbf{h}^{(pq)}(t)\right) \tag{3}$$

The activation potential, which corresponds to a linear combination of afferent local fields and input minus activation threshold $\theta^{(p)}$, is a real valued array. The activation function H is the Heaviside function so that the neuron state is a binary array, which takes its values in $\{0, 1\}$.

2.2 Weights setting

2.2.1 Random Recurrent Neural Networks

Our network belongs to the category of Random Recurrent Neural Networks (RRNNs), so that the weights obey to a random draw. The principal consequence of this setting is the "almost sure" non-symmetry of the connectivity pattern, so that one can not ensure the convergence of the dynamics toward a fixed point attractor. Autonomous RRNN's (i.e. $\forall t, \mathbf{u}(t) = 0$ in eq.(1)) are discrete time dynamical systems, that can for instance display a generic quasi-periodicity route to chaos while progressively increasing the gain of a continuous transfer function (Doyon et al., 1993).

Each family of weights $\mathbf{J}^{(pq)}$ (weights from population q toward population p) are set according to a uniform distribution. The main parameters are $\bar{J}^{(pq)}$ (weights mean), $\sigma_{\mathrm{J}}^{(pq)}$ (weights standard deviation) and $\rho^{(pq)}$ (weights sparsity), such that the expectation of the weights is $E\left(J_{ij}^{(pq)}\right) \simeq \frac{\bar{J}^{(pq)}}{N^{(q)}}$ and the variance is $\mathrm{var}\left(J_{ij}^{(pq)}\right) \simeq \frac{\left(\sigma_{\mathrm{J}}^{(pq)}\right)^2}{N^{(q)}}$. The precise weights settings are given in Annexe A.

The activation threshold $\theta^{(p)}$ have scalar values (they are identical for every neuron of a given population).

2.2.2 Biological constraints

In this paper, in order to remain coherent with elementary biological requirements, we rule the weights of a given population $q \in \{1, ..., P\}$ to be either purely excitatory or purely inhibitory. This constraint implies additional dependencies between $\bar{J}^{(pq)}$, $\sigma_{\mathrm{J}}^{(pq)}$, $N^{(q)}$ and $\rho^{(pq)}$. The parameter $\rho^{(pq)}$ is derived from $\bar{J}^{(pq)}$ and $\sigma_{\mathrm{J}}^{(pq)}$ such that :

- When $\bar{J}^{(pq)} > 0$, the weights lower bound is 0, such that every weight is positive or null.

[1] Matlab code can be obtained by simple demand at `dauce@esm2.imt-mrs.fr`.

[2] On the contrary to Hopfield system (Hopfield, 1982), the input doesn't correspond to the initial state $\mathbf{x}^{(p)}(0)$ of the network.

- When $\bar{J}^{(pq)} < 0$, the weights upper bound is 0, such that every weight is negative or null.

Details are in Annexe A.

3. Excitatory-inhibitory interactions

Most of the cortical an sub-cortical layers are composed of interacting populations of excitatory and inhibitory neurons. We present in this section a large scale structure of interaction that we further call a "module". In order to take into account some basic physiological features, we suppose that our two-populations module grossly models a cortical column, such that :

- local dynamics dominate incoming signals.
- excitatory neurons can receive and send signals, inhibitory neurons only act locally.
- excitatory neurons represent 70-90% of the total population.

The parameters setting globally defines the way the two populations interact, independently of the population sizes (i.e. the dynamic properties are invariant with the sizes, provided the sizes are larger enough, i.e. >50). Two different parameter settings are given in the following sections.

3.1 Simple modules

The excitatory population is population 1. The inhibitory population is population 2. Inputs are only displayed on the excitatory layer, i.e. $\forall i, \forall t, u_i^{(2)}(t) = 0$ (see eq.(3)). We take binary inputs such that $\forall i, \forall t, u_i^{(1)}(t) \in \{0, 1\}$. We set the population sizes to $N^{(1)} = 1000$, $N^{(2)} = 200$.

We define two global parameters k and d which help to define the interaction pattern. The way the two populations interact are characterized by the *asymmetry* k between excitatory and inhibitory influences. Parameter k acts on the mean value of weights distributions. The *eccentricity* d mostly acts on the scattering/standard deviation of the weights such that :

$$\bar{J} = \begin{pmatrix} \frac{1}{2} & -\frac{k}{2} \\ \frac{k}{2} & -\frac{k}{2} \end{pmatrix} \quad \sigma_{\mathrm{J}} = \begin{pmatrix} \frac{1}{2d} & \frac{\sqrt{k}}{2d} \\ \frac{\sqrt{k}}{2d} & \frac{\sqrt{k}}{2d} \end{pmatrix}$$

The activation threshold relates to the individual excitabilities. The thresholds are small, so that a small excitation can initiate the neuron to spike. The thresholds are of the order of $1/10$ of the expanse of the weights interval, i.e. $\theta^{(1)} = 0.1$, $\theta^{(2)} = 0.1k$.

We present on figure 1 the reaction of a network toward input presentation. Once the system defined, the dynamical system (2-3) is initialized with a binary random vector taking values in $\{0, 1\}$. The spontaneous activity (without input) is displayed from $t = 1$ to $t = 100$. This activity is weak, irregular, with a slight synchronization (synchronous bursts appear as small peaks on the mean activity). The synchronization is stronger on inhibitory population. Then, an input is sent on 15 excitatory neurons. This input appears as a black band on first population activity. The network then tends to become more synchronous with almost periodic bursts of activity. This synchrony appears to be stronger in the inhibitory population. When the input is removed, the network turns back to its initial spontaneous weak activity.

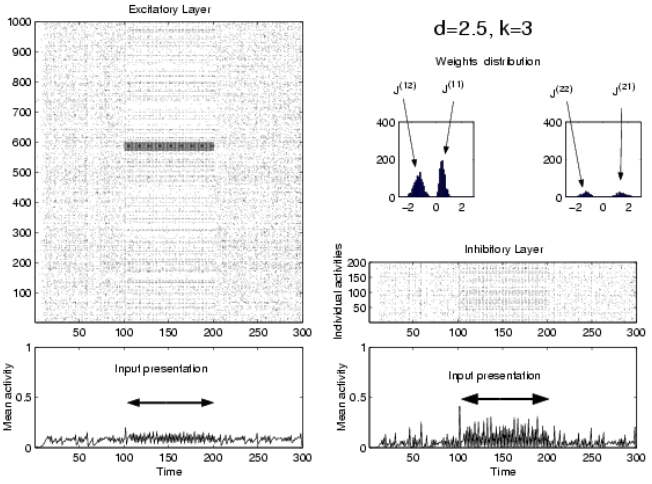

Figure 1: **Input presentation**. The network is composed of $N^{(1)} = 1000$ neurons and $N^{(2)} = 200$ neurons. The effective sparsity are $\rho^{(11)} = 0.8\%$, $\rho^{(12)} = 11\%$, $\rho^{(21)} = 2\%$, $\rho^{(22)} = 11\%$. An input is sent at time 100-200 on neurons 585-600. **Upper left** : global activity of the excitatory layer. **Lower left** : mean activity on excitatory layer. **Middle right** : global activity of the inhibitory layer. **Lower right** : mean activity on inhibitory layer. **Upper right insets** : distribution of the sum of incoming weights from excitatory and inhibitory populations, for 1000 excitatory neurons (left) and 200 inhibitory neurons (right).

The dynamical transition toward input presentation can be compared to a phase transition, i.e. the crossing of a bifurcation point in the parametric space. This kind of phase transition during input presentation has been observed for long time in neurobiology, see for instance (Skarda and Freeman, 1987), and also (Mac Leod and Laurent, 1996). More generally, synchronizing behaviors in unitary delays networks depend on the asymmetry k, i.e. inhibition has to dominate excitation for the network to produce synchrony. This point has still be noted in simulation works, see for instance (Bush and Sejnowski, 1996, Izhikevich, 2003).

3.2 Topologically organized modules

Topologically structured dynamical systems are defined by a *space* (or a *map*, which defines a distance), a *field*, possibly representing the states of uniformly distributed units, and a *process* by which those units interact. The nature of this interaction process is supposed to depend on the distance. Topologically structured dynamical systems have been introduced in neural modeling with the Neural Field of Amari (Amari, 1977).

Our maps are derived from Amari's neural map. We thus introduce a topology in the weight structure: We define a parameter $r^{(pq)}$ which represents the neighborhood density from population q toward population p (also called neighborhood radius). The weights are thus adapted according to the (normalized) distance between neuron i and neuron j. The smaller is $r^{(pq)}$, the tighter is the neighborhood. Close links are strongly enhanced (according to the initial random draw) whereas distant links fade to zero. One can notice that introducing a neighborhood factor tends to increase the sparsity of the weights. In concrete terms, every link such that $d_{ij}^{(pq)} > \pi r^{(pq)}$ is suppressed, i.e. set to zero, such that for $r^{(pq)} \leq 1$, the effective sparsity is ${\rho^{(pq)}}' = r^{(pq)}\rho^{(pq)}$. We take a simple $1D$ closed ring geometry. Map settings are in Annexe B.

Topological maps are used as perceptual modules in the next section. More details on random recurrent neural maps (as models of short term memory) can be found in (Daucé, 2004).

4. Perception-action network

The 2-population networks we have described in previous sections are now combined in order to build a perception/action network. A functional module is composed of 2 populations of neurons : one excitatory population and one inhibitory population. As previously said, a module can be seen as a rough approximation of a cortical column. A module can own a topology or not :

- topologically structured modules are associated with the perception processes;
- unstructured modules are associated with the action processes.

Designing a perception-action network also means to specify the environment through which the system interacts, and also to specify a task. As a first lookahead, we made the choice to minimize the environment complexity and apply our system on a very well known and documented task : the control of an inverted pendulum.

The environment is thus composed of 2 variables : the angular position ϑ and the angular velocity $\dot{\vartheta}$, and its dynamics is given by :

$$\begin{cases} \frac{d\vartheta}{dt} = \dot{\vartheta} \\ \frac{d\dot{\vartheta}}{dt} = g\sin(\vartheta) - 2\dot{\vartheta} + F \end{cases}$$

where $g = 9.81$ is the earth attraction and F is an external force (coming by the control system). We suppose that a measure of angular position ϑ can be done every 5 ms. The force F is also updated every 5 ms. The system stable fixed point is $\vartheta = \pi$. The task consists in maintaining the angular position within given bounds, say $[-\pi/15, \pi/15]$, whereas the natural pendulum tendency is to fall toward $\vartheta = \pi$.

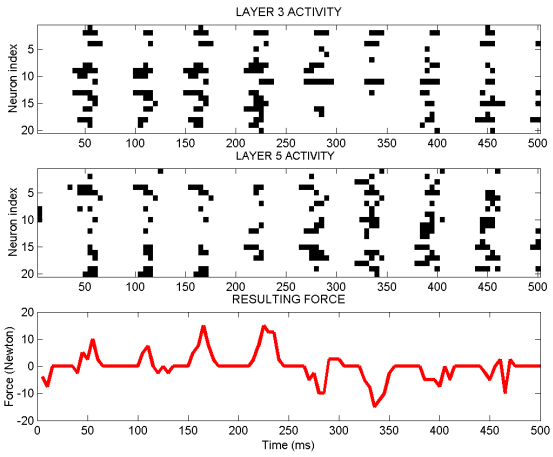

Figure 2: **Motor output production.** The upper figure gives an example of layer 3 activity, in case layer 3 is composed of 20 neurons. The middle figure presents layer 5 activity in the same conditions. In that simulation, the two layers are synchronized. The lower figure presents the resulting force, which comes from the difference of population activities. Time is in ms.

The control system is composed of 3 modules, each module owning two populations. In order to allow fast simulations, we limited the module sizes to 260 neurons (i.e. 200 excitatory neurons and 60 inhibitory neurons).

- The sensory module "S1" corresponds to populations 1 (excitatory) and 2 (inhibitory). Its topology is defined by $r^{(11)} = 0.2$, $r^{(12)} = 0.6$. The input is sent on layer 1. It activates 2% of the neuron around the reference position $\left\lfloor N^{(1)} \times \left(\frac{15\vartheta(t)}{2\pi} + \frac{1}{2}\right)\right\rfloor$.
- Two motor modules are defined. Module "M1" corresponds to populations 3 (exc.) and 4 (inh.), and module "M2" corresponds to populations 5 (exc.) and 6 (inh.). The force F is defined as the difference between the mean activities of populations 3 and 5, i.e.

$$F(t) = 50\left(m^{(3)}(t) - m^{(5)}(t)\right)$$

where $m^{(p)}(t)$ is the mean activity on layer p ($m^{(p)}(t) \in [0,1]$). Modules M1 and M2 act concurrently, one leading to the left, the other leading to the right. An example of network output is given on figure 2.

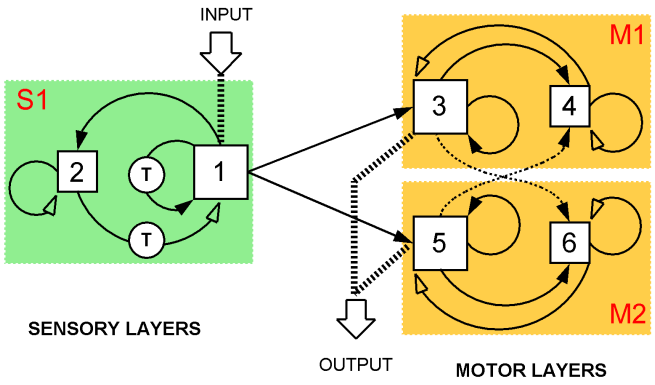

Figure 3: **Interconnection pattern for the perception-action network** The network is composed of 6 populations. Filled arrows represent excitatory connections. Unfilled arrows represent inhibitory connections. A functional module is composed of two strongly interconnected populations, one excitatory and one inhibitory. Sensory module **S1** is composed of populations 1 and 2, motor module **M1** of populations 3 and 4, motor module **M2** of populations 5 and 6. Module S1 owns a topology in its interconnection pattern (topological links are mentioned with symbol "T"). The 3 modules are interconnected through excitatory links (the inhibitory neurons only act locally). Module S1 sends (excitatory) signals toward modules M1 and M2. Modules M1 and M2 can inhibit each other through the excitation of their neighbor inhibitory layer (dotted links). Those lateral interactions are initially set to zero.

The global interconnection pattern is given on figure 3. The weights and threshold settings are comparable with the ones of sections 3.. Weights parameters are $k = 3$ and $d = 6$ with :

$$\bar{J} = \begin{pmatrix} \frac{1}{2} & -\frac{k}{2} & 0 & 0 & 0 & 0 \\ \frac{k}{2} & -\frac{k}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & -\frac{k}{2} & 0 & 0 \\ 0 & 0 & \frac{k}{2} & -\frac{k}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & -\frac{k}{2} \\ 0 & 0 & 0 & 0 & \frac{k}{2} & -\frac{k}{2} \end{pmatrix}$$

$$\sigma_{\mathrm{J}} = \begin{pmatrix} \frac{1}{2d} & \frac{\sqrt{k}}{2d} & 0 & 0 & 0 & 0 \\ \frac{\sqrt{k}}{2d} & \frac{\sqrt{k}}{2d} & 0 & 0 & 0 & 0 \\ \frac{1}{2d} & 0 & \frac{1}{d} & \frac{\sqrt{k}}{d} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{k}}{d} & \frac{\sqrt{k}}{d} & 0 & 0 \\ \frac{1}{2d} & 0 & 0 & 0 & \frac{1}{d} & \frac{\sqrt{k}}{d} \\ 0 & 0 & 0 & 0 & \frac{\sqrt{k}}{d} & \frac{\sqrt{k}}{d} \end{pmatrix}$$

We present on figure 4 the spontaneous behavior of the network/pendulum system. The initial pendulum position is randomly set in interval $[-\frac{\pi}{30}, \frac{\pi}{30}]$ rad, and the velocity is randomly set in interval $[-0.2, 0.2]$ rad/s. Some remarks can be made :

- Layer 1 activity directly adapts to the input, and develops a local "bundle" attractor in the vicinity of the stimulation. This attractor is driven by the input. Its radius of this is of the order of 10% of the map.

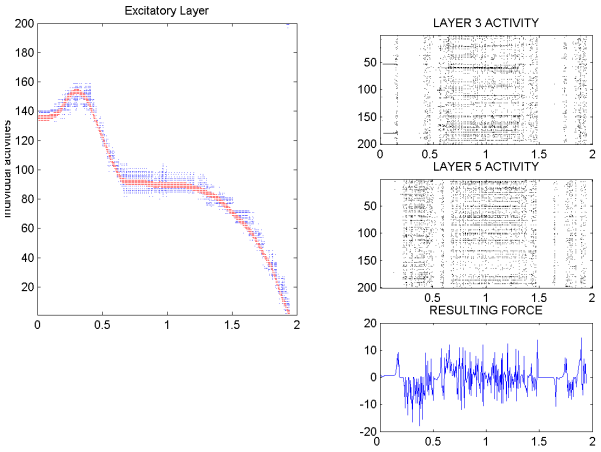

Figure 4: **Spontaneous dynamics of the network/pendulum system.** Left : layer 1 activity. The input (red trace) corresponds to the pendulum position which is measured every 5 ms (i.e. one time step corresponds to 5 ms). The pendulum overtakes the perceptual field around $t = 2$s. Upper right : layer 3 activity. Lower right : layer 5 activity. Middle right : network motor output, which is the difference between layer 3 mean activity and layer 5 mean activity. Time is in s.

- The activity of layers 3 and 5 organize under layer 1 influence. As in figure 1, They tend to display a synchronized activity when they get stimulated. The property of synchrony allows to reach a significant global activity, which defines the force to be sent to the output. The mean force amplitude is of the order of 5 N. The maximum force amplitude is of the order of 20 N.

- Even synchronized, the activity of layers 3 and 5 remains chaotic, and their pattern of activity is both driven by layer 1 activity *and* local reentrant activity (the activity of layer 5 between 0.2 and 0.5 s is thus mostly a self-feeding activity). This strong *versatility* in motor activity allows the system to explore and experiment various motor responses.

- The activities of layers 3 and 5 are balanced, i.e. the probability of a left command is almost the same as the probability of a right command. There may be only one layer to react to a given perceptual configuration, which can cause fast angular variations (for instance between 0.2 and 0.5 s).

More generally, the network/pendulum system is found to display a complex activity profile, where phase transition from one chaotic attractor to the other can be observed within the pattern of activity (i.e. it is not caused by an external parametric change). Such dynamics are often called *itinerant chaotic dynamics*, see

for instance (Tsuda, 1991). This property is of course of great interest in the perspective of a sensory-motor exploration where every attractor may correspond to a *structural coupling* between the agent and the environment (Varela et al., 1991).

5. A Hebbian reinforcement learning paradigm

5.1 Hebbian reinforcement in dynamic neural networks

Our *exploratory* process relies on the self-generated chaotic activity. We thus have to define a *selection* process through which the better configurations will be stabilized (see introduction). We use for this purpose the principle of the *Hebbian trace*. In an active network, a Hebbian reinforcement factor can be defined according to the pre-synaptic and post-synaptic activities, i.e. $\forall (p,q) \in \{1,...,p\}^2$

$$\mathcal{H}_{\mathrm{J}}^{(pq)}(t) = \frac{\alpha^{(pq)}}{N_{\mathrm{aff}}^{(pq)}} \mathbf{x}^{(p)}(t) \mathbf{x}^{(q)}(t-1)^T$$

where $\mathbf{x}^{(q)}(t-1)^T$ is the transpose of $\mathbf{x}^{(q)}(t-1)$. This Hebbian term represents the correlated activities from population q toward population p. In order to refine the weight selection, we add a local factor called the *cooperative limitation factor*. This factor is based on the local field of the target neuron, i.e. $\forall (p,q) \in \{1,...,p\}^2$

$$\mathcal{H}_{\mathrm{J}}^{(pq)}(t) = \frac{\alpha^{(pq)}}{N_{\mathrm{aff}}^{(pq)}} \left[\left(1 - H(\mathbf{h}^{(pq)} - \theta^{(p)}) \right) . \mathbf{x}^{(p)}(t) \right] \mathbf{x}^{(q)}(t-1)^T \quad (4)$$

where the "." operator represents a term to term product. It namely means that a weight adaptation will be allowed if (and only if) the activation of the target neuron relies on the cooperation between several local fields, i.e. for instance a cooperation between local recurrent stimulation and distant stimulation. If the source neuron is the only cause of target neuron activation, then no adaptation is done (for it is not necessary to reinforce a still strong connection). This factor reduces by 80-90% the number of selected links.

Those Hebbian terms are calculated every time step, and stored in a Hebbian trace with a slow decay, i.e.

$$\mathcal{T}_{\mathrm{J}}^{(pq)}(t) = \beta \mathcal{T}_{\mathrm{J}}^{(pq)}(t-1) + \mathcal{H}_{\mathrm{J}}^{(pq)}(t)$$

with $\beta = 0.95$, so that the "half-life" of a Hebbian trace is 20 time steps, i.e. 100 ms. The Hebbian trace thus memorizes the most recent correlated activities from layer q toward layer p.

The Hebbian trace being stored, the most delicate aspect of Hebbian reinforcement is the choice of the effective weights reinforcement according to positive or negative reward. In case of positive reward, the rule has to simulate and stabilize the current attractor. In case of negative reward, the rule has to favor the transition toward a different pattern of activity, i.e. to "inhibit" the current attractor. We thus defined two reinforcement path, one devoted to positive rewards, the other devoted to negative rewards (see figure 5) :

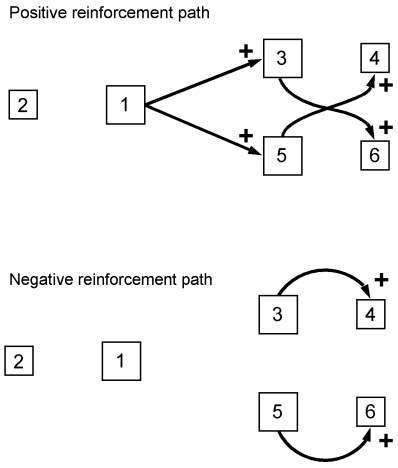

Figure 5: **The reinforcement path.**

- In case of positive reward, we both reinforce the excitatory path between the sensory module and the motor modules (visuo-motor path), and also the lateral connection toward the neighbor module inhibitory population, in order to reinforce the most active module in case of asymmetric activity (motor lateral path).
- In case of negative reward, we only reinforce the path corresponding to the motor module self inhibition, in order to weaken the most active module in case of asymmetric activity (local path).

Those reinforcements *only take place on excitatory links*, and the learning rule is such that the selected links can only be strengthened. The learning parameters are small enough for the weights adaptation to remain weak according to the weights initial values. We concretely set $\alpha^{(31)} = 0.1$, $\alpha^{(51)} = 0.1$, $\alpha^{(63)} = 0.15$, $\alpha^{(45)} = 0.15$ for the positive reinforcement path, and $\alpha^{(43)} = -0.15$, $\alpha^{(65)} = -0.15$ for the negative reinforcement path. The other $\alpha^{(pq)}$'s are equal to zero.

5.2 Closed loop protocol

The closed-loop learning protocol is the following. The angular position and velocities are randomly set according in intervals $[-\frac{\pi}{30}, \frac{\pi}{30}]$ and $[-0.2, 0.2]$. Then, the network/pendulum dynamics is iterated until a reinforcement signal is generated. The reinforcement signal mainly relies on the pendulum velocity. The aim is to maintain the velocity as small as possible i.e. :

- $R(t) = 1$ if $t > 300$ ms and $|\dot{\vartheta}(t)| < 0.05$ rad/s.
- $R(t) = -1$ if $|\dot{\vartheta}(t)| > 0.5$ rad/s or $|\vartheta(t)| > \frac{\pi}{15}$

The weights are then modified according to the Hebbian trace :

$$\text{If } \left(R(t)\mathcal{T}_{\mathrm{J}}^{(pq)}(t) > 0\right)$$
$$\mathbf{J}^{(pq)} \leftarrow \mathbf{J}^{(pq)} + R(t)\mathcal{T}_{\mathrm{J}}^{(pq)}(t)$$

Due to the decay term, only the most recent correlations get reinforced through the current Hebbian trace. The system is resetted after every trial. This series of operations is repeated for numerous trials, while the the weights get progressively modified according to the rewards.

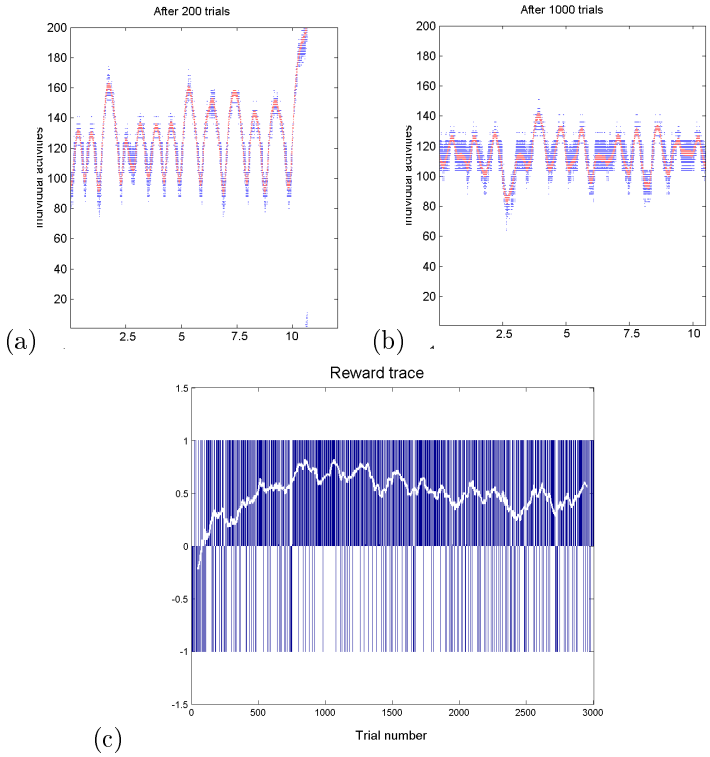

Figure 6: **Control adaptation during a learning experiment.** (a) Layer 1 activity displaying the pendulum control after 200 trials. Time is in s on the x-axis. (b) Layer 1 activity displaying the pendulum control after 1000 trials. (c) Rewards evolution during 3000 trials with their mean value on a sliding window of 100 trials (white line).

The result of a typical learning simulation is presented on figure 6. The experiment is repeated for 3000 trials. The successive reward values are shown on figure 6(c). The main observations are :

- The mean reward rapidly increases during the first steps of the experiment. Figure 6(a) shows that the network is already controlling the pendulum after 200 trials, which is a rather fast result, even if the task complexity is not too strong. The pendulum trajectory owns a periodicity, with various amplitudes. The network has learned to associate a negative force with the upper map activity (i.e. layer 5 dominates layer 3), and a positive force with the lower map activity (i.e. layer 3 dominates layer 5), whereas the two layers activities remain balanced in the middle of the map. The pendulum amplitude is found to increase so that it reaches the bound after 10 s.

- Then the mean reward slowly increases and seems to reach a plateau between 1000 and 1500 trials. This period corresponds to a control improvement. The resulting control is more accurate and tight, as shown on figure 6(b). Most of the activity takes place on the center of the map, where the two motor modules balance each other, with small fluctuations on one side or the other, so that the pendulum remains in the vicinity of $\pi = 0$ for unbounded time.

- The control can be said to be achieved after 1000 trials. However, we extended the number of trials in order to test the preservation of the learned behavior. Despite the slow reward decrease, we observed no significant alteration of the pendulum control accuracy.

This experiment illustrates the feasibility of a reinforcement learning protocol on a dynamic neural network using two separate Hebbian path which tend to stabilize favorable interaction patterns. The learning of the pendulum control is found to be very reproducible and robust, and the learning protocol, although relying on a simple Hebbian process, is found to be fast, giving significant results in 100-200 trials, and although capable of a slow improvement.

5.3 On-line protocol

For the on-line protocol, we allow several rewards during one trial, and there is no difference between exploration and exploitation processes. The only reset condition is the crossing of the angular bounds, so that the trial duration is equal to the control duration. In order to experiment various situations, we however limited the trial duration to 1000 time steps (i.e. 5s), so that our on-line protocol is not a "pure" on-line protocol.

Three improvements are moreover introduced in order to avoid weight drift :

- The rewards are adaptive, i.e. the reward amplitude depends on its probability of appearance (*habituation* principle). When most of the rewards are negative, a positive reward is more significant. On the contrary, when most of the rewards are positive, a negative reward is more significant. We thus define a mean reward trace r, initially equal to 0. Then,

$r' = 0.9r + 0.1$ in case of positive reinforcement, and $r' = 0.9r - 0.1$ in case of negative reinforcement, such that $R(t) = (1 - r')/(r' + 1)$ in case of positive reinforcement, and $R(t) = (1 + r')/(r' - 1)$ in case of negative reinforcement. Then, r is updated with the value of r', and the process goes on until the next reinforcement. A typical series of reinforcement values is given on figure 7.

- A slight forgetting term is added on the Hebbian rule, i.e.

$$\text{If } \left(R(t)\mathcal{T}_{\mathrm{J}}^{(pq)}(t) > 0\right)$$
$$\Delta\mathbf{J}^{(pq)} \leftarrow \left(1 - \frac{R(t)}{1000}\right) \times \Delta\mathbf{J}^{(pq)} + R(t)\mathcal{T}_{\mathrm{J}}^{(pq)}(t)$$
$$\mathbf{J}^{(pq)} \leftarrow \mathbf{J}_0^{(pq)} + \Delta\mathbf{J}^{(pq)}$$

  with $\Delta\mathbf{J}^{(pq)}$ initially set to 0 and $\mathbf{J}_0^{(pq)}$ is the initial weight matrix.

- The minimal interval between 2 different rewards is 20 time steps i.e. 100 ms.

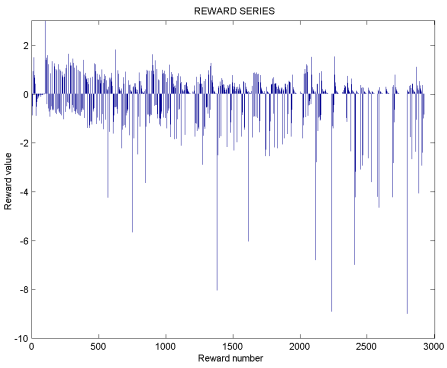

Figure 7: **Series of reward values in a typical on-line learning session.**

With those improvements, we empirically tested the convergence properties of our method, under three condition. Every condition has been tested on 20 networks. The first condition corresponds to the reinforcement path given in figure 5. In the second condition, the positive path is limited to the visuo-motor path (no lateral reinforcement). In the third condition, the positive path is limited to the motor lateral path (no visuo-motor reinforcement). Under those three conditions, we tested the median control duration over the 20 networks, for increasing trial numbers (i.e. ongoing learning process). the median control duration is measured over 20 networks, in a 10 trials width window (i.e 100^{th} control duration over 200 control duration values around trial number +/-5). Note that every simulation was stopped after 5s of control, so that the mean would not be representative of the real control duration. Results are given on figure 8.

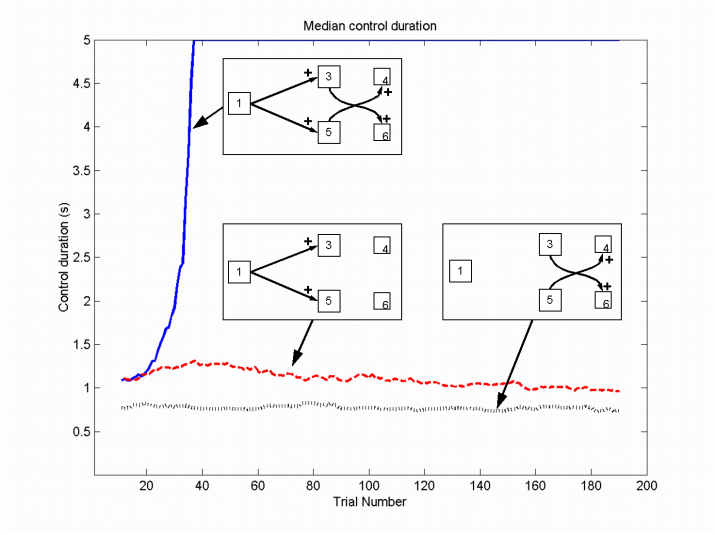

Figure 8: **Median control duration under three reinforcement conditions.** Trial number is on the x-axis, median control duration over 20 networks is on the y-axis (in seconds). The plain line corresponds to the full positive reinforcement path condition. The dashed line corresponds to the single visuo-motor path. The dotted line corresponds to the single motor lateral path.

This figure is self explanatory. First of all, the dual learning path (visuo-motor/motor-motor) is robust and operant over every network we tested. 40 to 50 trials are necessary for displaying a robust pendulum control (often less!). Second, single local path reinforcement or single visuo-motor path reinforcement do not display any significant increase of the control duration. A tight cooperation between the two processes is thus found to be necessary for the process to converge. The activation of the two path thus appears as a necessary condition for the learning process to be effective.

The nature of the process taking place during learning is not fully highlighted at present time. We can however give some clues :

- The negative reinforcement path doesn't seem to take much part in the reinforcement process. This mostly relies on limitations induced by the cooperative factor (eq. 4). Indeed, the local excitatory population is initially the only source of activation for the inhibitory population, until the learning process allows lateral inhibition to be activated. In the current settings of the network, the negative reinforcement path is not fully operant in the first steps of the learning process.

- The selection of a proper motor command relies on an exploration/selection process, where the exploration grounds on the versatility of spontaneous chaotic activity. The positive rewards often occur when one of

the two motor layers overtakes the other, so that a light inbalance tends to be twice consolidated by the process : the stronger layer receives a supplementary activation from the sensory layer through the visuo-motor path, and simultaneously weakens the activity of its neighbor throuh the motor lateral path.

- There are however some severe limitation in the competition process which avoid the learning process to diverge. First, the cooperative limitation factor prevents the motor layers to be too strongly driven by the sensory layer (the sensory layer activation can not overtake local activation). Second, the indirect neighbor inhibition prevents to extinguish the neighbor activity. It must indeed be known that local excitatory-inhibitory interactions tend to produce synchrony. An indirect effect of the lateral inhibition reinforcement is to increase the synchrony (and thus the cooperation) between the two motor layers.

- Why is one path such necessary to the other for the full process to collapse in case one is missing? One can just conjecturate that the two path may balance each other, as the visuo-motor path feeds the motor excitatory layers and the lateral path feeds the inhibitory layers, so that the global level of activity may remain approximately equal. When one of the two path is missing, the network activity may diverge, either with an excess of excitation or inhibition.

6. Outlook

So the main results of our simulation experiment is to demonstrate the feasibility of reinforcement learning using the intrinsic dynamical properties of large random recurrent networks, and a biologically compatible Hebbian learning rule. The reinforcement experiment we presented can be seen as a first lookahead toward more complex and realistic learning tasks, possibly taking place on real robotic agents. There is a need of intensive simulation toward various constraints, environments and reinforcement signals in order to stabilize the parameters, learning protocol and operational/generalization capabilities. One can also ask whether several concurrent behaviors may be learned in the same network. One can for instance imagine a robotic agent with several sensory modules exciting several concurrent motor modules, each one devoted to the control of a particular muscle flexion or extension.

More generally, this article aimed to demonstrate the relevance of a dynamic system framework in neural modeling and control, which may provide a unified approach at different scales, including the neural level, the structural level and the agent/environment interaction level (Guillot and Daucé, 2002). Such an approach could help to shed new lights on biological functions and structures, and also to give new protocols and methodologies for the design of artificial life-like systems. This work is thus to be completed.

Annexe A : Weights settings

For two given populations p and q, we set $\rho_0 = \frac{\left(\bar{J}^{(pq)}\right)^2}{3\left(\sigma_{\mathrm{J}}^{(pq)}\right)^2 N^{(q)}}$ and $\rho^{(pq)} = \frac{4\rho_0}{1+3\rho_0}$, which is the sparsity. We also set $\sigma^* = \frac{\sigma_{\mathrm{J}}^{(pr)}}{\sqrt{4-3\rho^{(pq)}}}$, which is the effective weights deviation and $N_{\mathrm{aff}}^{(pq)} = \rho^{(pq)} N^{(q)}$ represents the expectation of the number of afferent weights arriving from population q. $N_{\mathrm{aff}}^{(pq)}$ is such that $\left|\frac{\bar{J}^{(pq)}}{N_{\mathrm{aff}}^{(pq)}}\right| - \sqrt{\frac{3}{N_{\mathrm{aff}}^{(pq)}}}\sigma^* = 0$, i.e. $N_{\mathrm{aff}}^{(pq)} = \frac{\left(\bar{J}^{(pq)}\right)^2}{3(\sigma^*)^2}$.

A given weight $J_{ij}^{(pq)}$ is then equal to 0 with probability $1-\rho^{(pq)}$, and takes its value randomly in interval $\left[\frac{\bar{J}^{(pq)}}{N_{\mathrm{aff}}^{(pq)}} - \sqrt{\frac{3}{N_{\mathrm{aff}}^{(pq)}}}\sigma^*, \frac{\bar{J}^{(pq)}}{N_{\mathrm{aff}}^{(pq)}} + \sqrt{\frac{3}{N_{\mathrm{aff}}^{(pq)}}}\sigma^*\right]$, with probability $\rho^{(pq)}$, i.e.

$$J_{ij}^{(pq)} = \begin{cases} \frac{\bar{J}^{(pq)}}{N_{\mathrm{aff}}^{(pq)}} + \frac{\sigma^*}{\sqrt{N_{\mathrm{aff}}^{(pq)}}} b & \text{with probability } \rho^{(pq)} \\ 0 & \text{with probability } (1-\rho^{(pq)}) \end{cases} \tag{5}$$

where b is set according to $\mathcal{U}(0,1)$, which is a uniform distribution in $[-\sqrt{3}, \sqrt{3}]$.

Annexe B : Map settings

$\forall i \in \{1,...,N^{(p)}\}, \forall j \in \{1,...,N^{(q)}\}$, we calculate the distance

$$\delta_{ij}^{(pq)} = 2\pi \times \min\left(\left|i/N^{(p)} - j/N^{(q)}\right|, 1 - \left|i/N^{(p)} - j/N^{(q)}\right|\right)$$

so that $\delta_{ij} \in [0,\pi]$, and then we calculate a gaussian normalized neighborhood factor

$$\nu^{(pq)}(\delta) = \frac{\sqrt{2\pi}}{r^{(pq)}} \exp\left(-\frac{1}{2}\left(\delta/r^{(pq)}\right)^2\right)$$

We finally modify the weights according to this neighborhood factor

$$\tilde{J}_{ij}^{(pq)} = J_{ij}^{(pq)} \times \nu^{(pq)}\left(\delta_{ij}^{(pq)}\right)$$

Knowing that

- $\int_0^\infty \nu^{(pq)}(\delta)d\delta = \pi$, and, for small $r^{(pq)}$, $\int_0^\pi \nu^{(pq)}(\delta)d\delta \simeq \pi$.
- The repartition of $\delta_{ij}^{(pq)}$'s is uniform in $[0,\pi]$,

For $r^{(pq)} < 1$, weights enhancements and weights decays are globally balanced, so that the expectation of the weights remains unchanged by this transformation. On the contrary, the weights standard deviation is increased by a factor of the order of $1/r^{(pq)}$. In order to avoid too large weights distortion, we define a adaptation factor which is designed in order to approach $1/r^{(pq)}$ for $r^{(pq)} < 1$, and 1 for $r^{(pq)} \geq 1$, i.e. $\kappa^{(pq)} = 1 + \frac{\exp\left(-\left(r^{(pq)}\right)^2\right)}{r^{(pq)}}$ and the weights standard deviation is set to $\sigma^{(pq)\prime} = \frac{\sigma^{(pq)}}{\sqrt{\kappa^{(pq)}}}$

Acknowledgments

This work is supported by ACI "neurosciences intégratives et computationnelles", thème "temps et cerveau" and by the ACI "Robea".

References

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87.

Ben-Yishai, R., Lev Bar-Or, R., and Sompolinsky, H. (1995). Theory of orientation tuning in visual cortex. *Proc. Nat. Acad. Sci. USA*, 92:3844–3848.

Bush, P. and Sejnowski, T. (1996). Inhibition synchronizes sparsely connected cortical neurons within and between columns in realistic network models. *J. Comput. Neurosci.*, 3(2):91–110.

Camperi, M. and Wang, X.-J. (1998). A model of visuospatial working memory in prefrontal cortex: Recurrent network and cellular bistability. *J. of Computational Neuroscience*, 5:383–405.

Cessac, B. (1995). Increase in complexity in random neural networks. *Journal de Physique I*, 5:409–432.

Compte, A., Brunel, N., Goldman-Rakic, P. S., and Wang, X.-J. (2000). Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral Cortex*, 10:910–923.

Daucé, E. (2004). Short term memory in recurrent networks of spiking neurons. *Natural Computing*. in press.

Daucé, E., Quoy, M., Cessac, B., Doyon, B., and Samuelides, M. (1998). Self-organization and dynamics reduction in recurrent networks: stimulus presentation and learning. *Neural Networks*, 11:521–533.

Daucé, E., Quoy, M., and Doyon, B. (2002). Resonant spatiotemporal learning in large random recurrent networks. *Biological Cybernetics*, 87:185–198.

Doyon, B., Cessac, B., Quoy, M., and Samuelides, M. (1993). Control of the transition to chaos in neural networks with random connectivity. *Int. J. of Bif. and Chaos*, 3(2):279–291.

Guillot, A. and Daucé, E. (2002). *Approche dynamique de la cognition artificielle*. Lavoisier.

Hansel, D. and Sompolinsky, H. (1996). Chaos and synchrony in a model of a hypercolumn in visual cortex. *J. Comp. Neurosc.*, 3:7–34.

Hertz, J. and Prugel-Bennett, A. (1996). Learning synfire chains: turning noise into signal. *Int. J. Neural Systems*, 7:445–450.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci.*, 79:2554–2558.

Hupé, J.-M., James, A., Payne, B., Lomber, S., Girard, P., and Bullier, J. (1998). Cortical feedback improves discrimination between figure and background by v1, v2 and v3 neurons. *Nature*, 394:784–787.

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE trans. on neural networks*, 14(6):1569–1572.

Mac Leod, K. and Laurent, G. (1996). Distinct mechanisms for synchronization and temporal patterning of odor-encoding cell assemblies. *Science*, 274:976–979.

Moynot, O. and Samuelides, M. (2002). Large deviations and mean-field theory for asymmetric random recurrent neural networks. *PTRF*, (123-1):41–75.

Skarda, C. and Freeman, W. (1987). How brains make chaos in order to make sense of the world. *Behav. Brain Sci.*, 10:161–195.

Sompolinsky, H., Crisanti, A., and Sommers, H. (1988). Chaos in random neural networks. *Phys. Rev. Lett.*, 61:259–262.

Sutton, R. (1988). Learning to predict by the method of temporal differences. *Machine learning*, 3:9–44.

Tsuda, I. (1991). Chaotic itinerancy as a dynamical basis of hermeneutics in brain and mind. *World Futures*, 32:167–184.

Varela, F., Thompson, E., and Rosch, E. (1991). *The Embodied Mind*. MIT Press.

Wang, X.-J. (1999). Synaptic basis of cortical persistent activity: the importance of nmda receptors to working memory. *The Journal of Neuroscience*, 19(21):9587–9603.

Watkins, C. and Dayan, P. (1992). Q-learning. *Machine learning*, 8:279–292.