Needs and motivations as mechanisms of learning and control of behaviour: interference problems with multiple tasks

Gianluca Baldassarre *
Department of Computer Science
University of Essex
CO4 3SQ Colchester, United Kingdom
email: gbalda@essex.ac.uk

Abstract

This work addresses the issue of agents that learn to carry out multiple (eventually) conflicting tasks by reinforcement learning, and learn to select the appropriate action according to their internal needs. The goal is to study interference problems that can arise when using "monolithic" neural networks for this purpose. The neural architecture used for the study is based on ideas drawn from the realm of needs and motivation in animals, and in particular from the homeostatic theory of regulation of physiological body variables. The task of the simulated organism consists in searching for two different resources according to its internal need state. The model studied is able to learn the behaviours necessary to accomplish the two tasks and to switch from one behaviour to the other according to the internal need state. The simulations show the existence of undesired interference problems between the two tasks, arising both during learning and during action selection. This suggests that innate or emergent neural modular architectures could be a better solution when multiple tasks are faced.

1 Introduction

Usually animals have the capacity to do many different things in response to different physiological needs: eating, drinking, mating, and so on. Superior animal species, like mammals, are endowed with mechanisms to build flexible behavioural responses to these needs, such as conditional learning and instrumental learning [Pavlov, 1927; Thorndike, 1911]. Instrumental learning allows the organisms to learn to produce appropriate behaviours that lead to primary reinforcers (e.g. to learn the "consummatory behaviour" of eating that leads to the ingestion of food). Also it allows the organisms to learn behaviours that lead to states that have acquired the property of secondary reinforcers through conditional learning (e.g. to learn the "appetitive behaviour" of approaching a particular kind of tree, seen from far away, that carries edible fruit) [Baldassarre and Parisi, 2000]. The presence of multiple physiological needs, and the opportunity to learn to achieve secondary reinforcers, implies that the neural systems underlying the learning processes should be capable of dealing with multiple and eventually conflicting tasks. From the point of view of the nervous system of animals, the construction of adaptive behaviours is based on both an "internal" input signal coming from the body and relative to the different physiological needs, and an "external" input signal coming from the world [McFarland, 1993]. The internal input signal contains information about which need the behaviour should satisfy. The external input signal contains information about the availability of primary and secondary reinforcers.

What should be the neural architecture of organisms capable of learning, say, to approach a fruit tree or a water pond, to eat or to drink? In particular, what are the consequences of having "monolithic" neural architecture to learn different behaviours that accomplish different tasks? Would it be useful to have an innate or emergent modularity, where different neural modules are dedicated to accomplish different behaviours? This paper addresses the issue of the interference problems that affect monolithic neural architectures. On the base of the simulations presented here, it is inferred that neural modular architectures could help to avoid interference problems.

The paper studies a simulated organism that has to search for food or water according to its physiological needs (the case of carrying out multiple tasks to achieve secondary reinforcers directed to the same need, is not considered here). Learning to carry out these two conflicting tasks implies the solution of two sub-problems, each of which could imply interference difficulties:

1. Learning the appropriate behaviour to satisfy each single need. The fact that the same computational resources (the same synaptic weights) are used for both tasks, can potentially generate interference problems. For example learning a behaviour alters the other behaviour.

* The research of the author is supported by a scholarship from the Department of Computer Science of the University of Essex.

2. Learning to select one of the two behaviours according to the current need state. The monolithic architecture could give rise to interference problems in selecting the two behaviours.

In order to test the existence and nature of these interference problems, a controller of the simulated organism has been designed that is supposed to capture some basic features of animal learning. The core of the model is a neural network implementation of the reinforcement learning actor-critic method [Lin, 1992; Barto *et al.*, 1990]. As shown in [Sutton and Barto, 1990; Shultz *et al.*, 1997; Baldassarre and Parisi, 2000] this model is capable of representing the basic features of conditional learning and instrumental learning. The input to the actor-critic system is pre-processed by a Kanerva-coding neural network that maps the input from the need system and the external sensors into a high number of "feature" units [Sutton and Whitehead, 1993]. Activities such as feeding, drinking and mating, involve control principles that can be represented in models employing the terminology and concepts of homeostatic theory [Mc Farland, 1993]. Hence a system based on homeostatic mechanisms has been designed to capture the main aspects of the regulation of energy and water in the organism's body.

While the model does not offer a definition of "emotion", it offers a definition of "motivation". In the model "motivation" is the information about the gap, called "need" in the paper, between the optimal and the actual level of a controlled physiological variable (e.g. the level of water in the body). This information is sent to the critic (responsible for the conditional learning) and the actor (responsible for instrumental learning, i.e. for selecting the appropriate behaviour of searching for food or water and for executing these behaviours in details). See [Cecconi and Parisi, 1992] for a similar definition of motivation, and [Maes, 1990] for another example of motivation as action selection. The model contains another aspect related to the "affective sphere" of animals. The level of need modulates the reward that is perceived by the organism when a consummatory behaviour is accomplished: the smaller the need, the smaller the perceived reward (cf. [Humphrys, 1997] for this idea and for a survey of the "action selection" literature).

Section two of the article presents the details of the controller and the scenario of the simulations. Section three describes the experiments accomplished and the interpretation of the results obtained. Section four discusses the potentiality of modular architectures when multiple tasks have to be accomplished.

2 Scenario, organism and neural architecture

The environment of the simulations is a 1x1 unit toroidal square arena. In this arena there are 30 elements of "red" food and 30 elements of "blue" water, each represented as a circle with a radius of 0.005. In a first experimental condition the food is randomly spread in the left half of the arena and the water in the right half. In a second experimental condition, water and food are randomly spread in the arena. Figure 1 shows the second experimental condition. In both conditions food and water elements are at least 0.1 distant from the borders of the arena. The organism is represented as a circle with a radius of 0.01. The simulation takes place in discrete time cycles. If in one cycle the organism steps on an element of food (water), it eats (drinks) the food (water) element that then disappears. When a resource element is consumed, a new element is introduced in a random location. The organism is endowed with two ingestion sensors, one for food and one for water. They get an activation of 1 if the correspondent resource is ingested, with 0 otherwise.

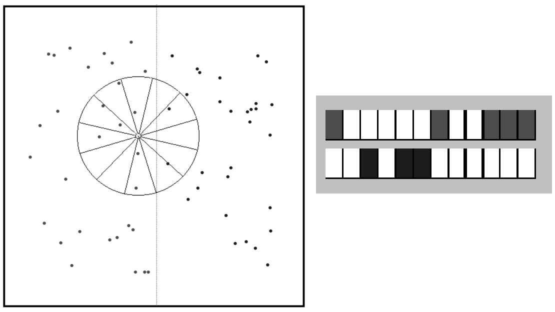

Figure 1. Left: the environment, the organism and its visual field. Right: the current activation of the organism's sensors.

Each organism has a one-dimension “retina” of 12 double non-overlapping aligned sensors, sensitive to two colours, red and blue. These sensors receive information from a 360° visual field. Each sensor has a scope of 30° and a depth limited to 0.2. A sensor takes an activation of 1 if an element of food (water) or part of it is within its field, 0 otherwise (figure 1).

The organism has two legs, both moved in each cycle. The effect of these steps is equal to the one you would have with a two-wheel robot. By controlling the length of the left and right step, the organism can go straight (same length for both left and right step) or turn (different lengths). The length of left and right steps can be either 0 or 0.02 (so there are four possible actions: do not move, turn left, turn right, go straight). The components of the organism's controller are represented in Figure 2.

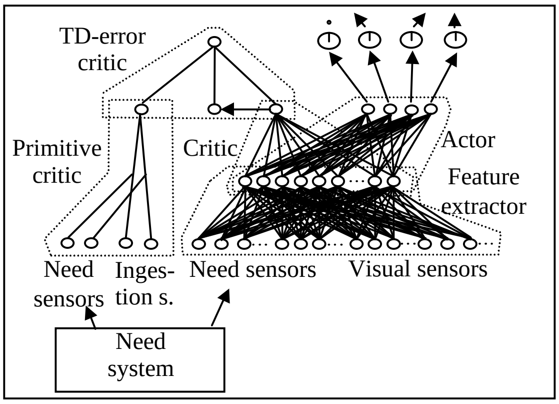

Figure 2. The main components of the neural architecture of the organism's controller.

Now the general features of the system are described. Refer to [Baldassarre and Parisi, 2000] for details. The feature extractor has 48 input units divided into 4 groups. The first 12 units encode the level of the need for food (the higher the level, the higher the number of units that assume an activation of 1) and the second 12 units encode the level of the need for water. Within this paper these 24 units are called "motivational units" and the flow of information that goes from the need system to the motivational units is called "motivation". The third and fourth 12 unit blocks encode the activation of the red and blue sensors respectively. The feature extractor has 200 output units (feature units). The feature extractor implements a "Kanerva re-coding" of the input [Sutton and Whitehead, 1993]. Its weights are randomly drawn in the set {-1, +1}. Each feature unit takes an activation of 1 if the Hamming distance between the input pattern and the "prototype" encoded by its weights is bigger then 0.6, of 0 otherwise.

The actor is a perceptron [Widrow and Hoff, 1960], that takes the activation of the feature units of the feature extractor as input pattern. It has 4 sigmoidal output units that locally encode four actions (do not move; go left; go right; go straight). In order to select one action, the activation p_k of the four output units is used for a stochastic winner-takes-all competition. The probability P[.] that a given action a_g among the a_k actions becomes the winner action a_w is computed as follows (using the more complicated soft-max formula commonly used in the literature made no difference in the speed of learning):

$$P\left[a_g = a_w\right] = p_g \Big/ \sum_k p_k$$

The critic is a perceptron that takes the activation of the feature units as input and has one linear output unit. The critic has to learn to give as output an estimation V' of the evaluation V of the current state s_t, defined as the expected discounted sum of all future rewards r, given the current action-selection policy π:

$$V^{\pi}\left[s_t\right] = E^{\pi}\left[\gamma^0 r_{t+1} + \gamma^1 r_{t+2} + \gamma^2 r_{t+3} + ...\right]$$

where $\gamma \in$ [0, 1] is the discount factor (set to 0.95 in the simulations).

The TD-error network is an implementation in neural terms (its weights are hardwired) of the computation of the Temporal-Difference error E defined as:

$$E_t = \left(r_{t+1} + \gamma\, V'^{\pi}\left[s_{t+1}\right]\right) - V'^{\pi}\left[s_t\right]$$

The critic is trained with a Widrow-Hoff algorithm [Widrow and Hoff, 1960] that uses as error the error signal coming from the TD-critic. The weights w_j are updated so that the estimation $V'^{\pi}[s_t]$ of $V^{\pi}[s_t]$, expressed at time t by the critic, tends to become closer to the target value (r_{t+1}+ γ $V'^{\pi}[s_{t+1}]$). This target value is a more precise evaluation of s_t being it expressed at time t+1 on the base of the observed r_{t+1} and the new estimation $V'^{\pi}[s_{t+1}]$:

$$\Delta w_j = \eta\ E_t\, y_j$$

where η is a learning rate set to 0.01 in the simulations, and y_j. is the activation of the feature units.

The actor is trained according to the error signal coming from the critic. Given that the critic learns to produce an evaluation $V'^{\pi}[s_t]$ of s_t according to the average action that the actor selects with s_t, if $E_t > 0$ it means that the winning action a_w has positively "surprised" the critic, so its probability of being selected is increased. If $E_t < 0$ the probability is decreased. This is done by updating the weights of the unit correspondent to a_w as follows (ζ is a learning rate set to 0.02 in the simulations):

$$\Delta w_{wj} = \ \zeta\, E_t\, y_j$$

The "need" system depends on the physiological level of energy and water in the organism's body. In a first experimental condition both needs for food and water are constantly kept at the maximum value, 1. In a second experimental condition the two needs dynamically change according to the level of energy and water in the body. The need for water works on the basis of the same principles as the need for food, so only the later is described. These principles are summarised in figure 3.

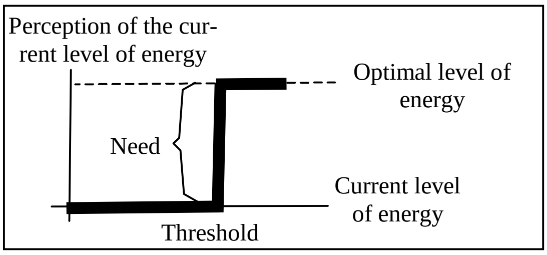

Figure 3. Main aspects of the need for food.

The level of energy has the following dynamics:

$$L_t = \alpha\, L_{t-1} + \beta\, I$$

where L is the level of energy, α is a decay coefficient set to 0.999 in the simulations, β is the content of energy of one element of food, set to 0.03 in the simulations, and I is the activation of the food ingestion sensor (the activation is 1 when food is ingested, 0 otherwise). The organism has a "perception" of the current level of energy that is a function of the actual level of energy (only the case of a step function has been explored). "Need" is defined as the difference between the optimal level of energy and the perception of its current level. The level of the need is used to activate the correspondent unit of the primitive critic (it has a continuous activation between 0 and 1) and a proportional number of need-for-food units of the feature extractor (they have an activation of 0 or 1).

If the current higher need always gets the control, and if the food and water are concentrated in different zones of the space, soon the organism runs out of energy and water. In fact it would search for food, eat a little bit, become thirsty, search for water, drink a little bit, start to be hungry again, etc. Minsky [1986] called this problem "dithering". To avoid dithering, a simple solution has been adopted, inspired by Minsky's idea that each need gets control for some minimum amount of time. The perceived water and food needs have a reciprocal inhibition, so that in each moment if one perceived need has the maximum value, 1, the other has the minimum of 0. Once one need gets control, it loses it only when the correspondent level of energy (water) reaches a threshold thanks to the activity of eating (drinking). When this event occurs, the other need gets control.

The primitive critic is a network that maps the signals coming from the world, correspondent to the activation of the ingestion sensors of food and water, into an internal reward signal. The internal reward signal is the activation of an "internal reward unit" that computes the sum of the signals coming from the two ingestion sensors. In the simulations the weights of the connections between the ingestion sensors and the internal reward unit are set to +1. Each of the signals coming from an ingestion sensor is multiplied by the signal coming from the correspondent need sensor that encodes the level of the need. The effect of this multiplication is that the higher the need the higher the reward perceived.

3 Experiments and results

The simulations have been done under four experimental conditions:

1) The food and water are respectively concentrated in the left and right halves of the environment. Both needs are constantly set to 1.
2) The food and water are respectively concentrated in the left and right halves of the arena. The needs have the dynamics described in section two.
3) Food and water are randomly spread in the environment. Both needs are constantly set to 1.
4) Food and water are randomly spread in the environment. The needs have the dynamics described in section two.

In the experimental conditions 1) and 3), the dependent variable measured has been the steps taken to reach indifferently an element of food or water. A moving average on the last 100 elements reached has been used. An organism that has not undergone the learning process has a performance of about 110 average steps taken to reach an element of food or water. This measure is useful as a baseline to judge the performance of organisms that undergo the learning process. In the experimental conditions 2) and 4), the same moving average has been used, but an element of food or water reached has been counted as "valid" only if the correspondent need was equal to 1. With this second measure of performance, the performance of an organism that has not undergone a learning process is about 212. Each result shown is an average of 9 simulations run with different random seeds.

3.1 Concentrated food and water, needs set to 1

When the elements of food are concentrated in the left side of the arena, the elements of water are concentrated in the right side, and the needs are constantly set to 1, the organism has the performance shown in figure 4.

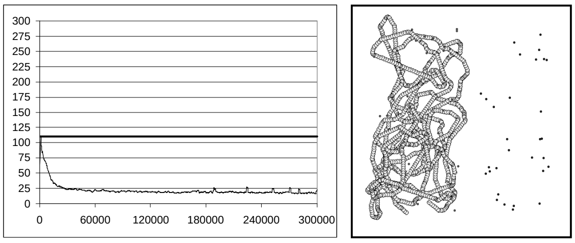

Figure 4. Left: the thin line plots the performance against cycles of an organism whose needs are set to 1. The bold line is the average performance of an organism following a random walk. Right: path of an organism specialized in searching for food.

The interesting fact about these experimental conditions is that in the 9 simulations run with different random seeds, the organisms specialise in searching for either one or the other resource (in the simulations 4 organisms have specialised in searching for food and 5 in searching for water). The reason is that the reward is given for both resources and the motivational units have always the same activation. Once the organism starts to specialise to search for one resource and all the computational means are dedicated to this task (for example food, see figure 4), it is attracted on the area where this resource can be found and ignores the other resource.

3.2 Concentrated food and water, dynamic needs

When the resources are concentrated in different regions of the arena and the needs are dynamic, the organism performs as showed in figure 5.

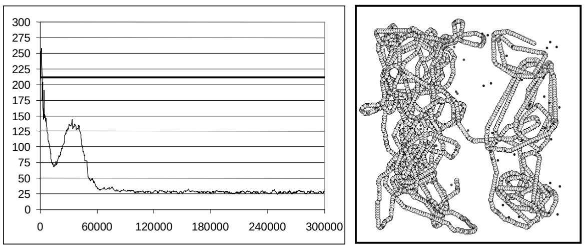

Figure 5. Left: performance of an organism with dynamic needs. Right: path followed by the organism.

The performance reaches a level of 25 steps per element of food/water collected when the correspondent need is 1, indicating that the organism is capable of learning to carry out the two tasks. The interesting fact is that once the organism has learned the two tasks, a change in the state of needs provokes a sudden change of the behaviour, thanks to the motivational signal that goes from the need system to the feature extractor. The motivational signal succeeds in selecting the correct behaviour. This fact is shown on the right side of figure 5. Figure 6 shows the percentage of elements of food or water that are collected when the correspondent or the other need has the control.

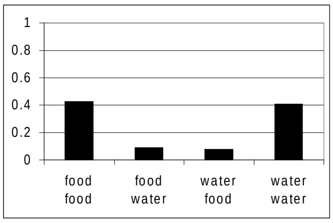

Figure 6. Food and water collected (as percentage of total elements collected) when the two different needs are active ("food food" means need for food, food collected; "food water" means need for food, water collected; etc.)

Figure 5 shows that the performance of the organism becomes temporary bad at about 40000 cycles. An explanation for this can be given by observing the behaviour of the organism. At the beginning the organism has a particular need, say for food, so it learns to search for food. When the need for food is satisfied, the need for water takes control. At this point even if the motivation signal has changed, the organism appears to continue to be guided by the sight of food. Probably this happens because some feature units that encode the position of food continue to have an activation of 1 even if the input is partially different because of the different motivational signal (the behaviour "connected" with these units is precisely the behaviour of searching for food). After some time, given that the ingestion of food does not produce a positive reward anymore, the critic starts to produce a signal that progressively erase the searching-for-food behaviour connected to the features units that generalise in a wrong way. This problem is a quite general problem of interference between multiple tasks. Neural networks have the capacity of generalising the behaviour to similar input patterns, and this turns out to be an advantage when similar answers have to be given to similar input patterns. When different answers have to be given to otherwise identical input patterns, dissimilar only in the part that select for different tasks, the capacity of generalisation can cause interference. For example this happens when different motivational input signals require completely different answers to identical visual input patterns.

3.3 Distributed food and water, needs set to 1

When food and water elements are distributed in the whole arena, and both needs are constantly set to 1, the organism's performance is shown in figure 7.

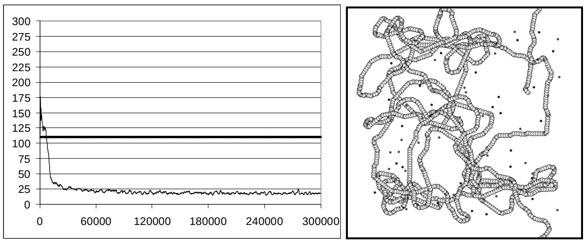

Figure 7. Performance of an organism with needs set to 1 in an environment with distributed resources.

The performance reaches a good level, similar to the condition with concentrated food. Differently from the later case, however, the organism learns to search both for food and water instead of specialising. This difference of behaviour can probably be explained as follows. The organism repeatedly encounters both food and water, so that the updating of the weights of the critic and of the actor for the first task alternates with the updating for the second task. As a consequence, none of the two tasks takes up all the computational resources.

3.4 Distributed food and water, dynamic needs

When the food and water are distributed on the whole arena and the needs are dynamic, the organism has a performance like the one shown in figure 8. The performance reaches a level of about 28. Again the performance becomes temporarily worse around cycle 50000. Probably the explanation of this fact is similar to the one given in for the condition of concentrated resources, but contrary to the later case, this time the direct observation of behaviour did not furnish clear evidence.

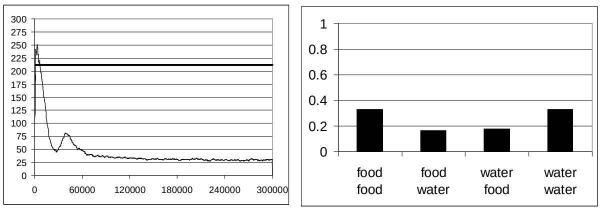

Figure 8. Performance of organism with distributed resources and dynamic needs

The right part of figure 8 shows the percent of elements of food and water reached when the correspondent or the different need has control. The organism has a certain capacity of focussing on the resource that satisfy the current prevailing need. However a direct observation of the behaviour shows that in some circumstances the organism reaches for one resource, say water, even if it has a need for food and an element of food is in sight.

Why the condition with distributed resources is more challenging than the one with concentrated resources? A possible explanation is that the later condition allows the organism to learn a given task for a certain time in "clean" perceptual conditions, i.e. without the activation of the sensors of the "wrong" resource. In these conditions when the actor's weights are changed, say, to learn to search for food, the feature units that encode the vision of water are off. The consequence is that the weights of the connections from these feature units to the units of the actions are not updated and the searching-for-water behaviour is not distorted. In the condition with distributed resources, the sensors for food and for water are often activated simultaneously, so the weights that allow the accomplishment of one task are changed even if it is the other task that is being learned. Similar interference problems happen at the level of the critic that has to learn to predict the value of a given input.

4 Discussion and conclusion

This work has presented a neural model of an organism that has to accomplish two different tasks. The model presented has been used to study the interference problems that can arise in a "monolithic" neural network architecture when a need state and a motivational signal are used to learn and select among different behaviours. It has been shown that the neural network capacity to generalize turns out to be useful when similar input patterns require similar answers. On the other hand this property can turn out to have undesired effects when slightly different input patterns require completely different answers. This work has shown the occurrence of this problem when the input patterns differs in the motivational signal.

In cases where the different tasks to be accomplished are known a-priori, it could be useful to use a modular architecture where different neural modules are dedicated to different tasks (see for example [Lin, 1993]). This is the way that natural evolution has followed in some cases. For example Alcock [1998] reports a study on the praying mantis where different innate neural modules (command centers) are dedicated to mating, foraging, etc. In this and similar cases the content of behaviour is itself innate.

In other cases the behaviour is learned (like in the model presented here), but it is still potentially possible that motivation selects for different innate modules that contain learned behaviours. Consider the superior species like mammals. The homeostatic regulatory mechanisms (the primitive actor and the need system of the model) that control the needs and motivations underlying activities such as feeding, drinking and mating, are innate [McFarland, 1993]. For this reason there is the possibility of having a system made of different innate modules dedicated to different needs/motivations. The motivation signal could go to a neural network (a selector) capable of choosing which module should be triggered each time, or could directly select for the suitable module.

In cases where the tasks cannot be known a-priori, the modularity has to be emergent. This situation includes all the cases that involve secondary reinforcers or punishers. In these cases "what to do", i.e. the tasks, are themselves learned. For example, within the same need of feeding, the organism should learn to trigger an avoidance behaviour when exposed to the sight of a poisonous food ("second punisher") while should learn to trigger an approaching behaviour when exposed to an edible food ("second reinforcer"). In these cases, since the motivational signal is fixed, the choice among the different emergent modules/behaviours should be done on the basis of the external input (see for example [Nolfi, 1997]).

Notice that the advantage of having a modular architecture to avoid interference would be diminished by the advantage of using common pieces of behaviour to satisfy different needs.

Future work will explore these issues.

Acknowledgments

I thank Prof. Jim Doran and Prof. Paul Scott of the Department of Computer Science of the University of Essex, and Prof. Parisi of the Institute of Psychology - Italian National Research Council, for their precious help and suggestions.

References

[Alcock, 1998] John Alcock. *Animal Behavior: An Evolutionary Approach*. Sinauer Associates, Sunderland, Massachusetts, 1998.

[Baldassarre and Parisi, 2000] Gianluca Baldassarre and Domenico Parisi, Classical Conditioning in Adaptive Organisms, Submitted to: Jean-Arcady Meyer, Alain Berthoz, Dario Floreano, Herbert L. Roitblat, Stewart W. Wilson, *From Animals to Animats 6: Proceedings of the 6th International Conference on the Simulation of Adaptive Behaviour,* Cambridge, Mass., 2000. The MIT Press.

[Barto *et al*., 1990] Andrew G. Barto, Richard S. Sutton, Christopher J. C. H. Watkins, Learning and sequential decision making. In Michael R. Gabriel and John W. Moore, editors, *Learning and Computational Neuroscience: Foundations of Adaptive Networks*. The MIT Press, Cambridge, Mass., 1990.

[Cecconi and Parisi, 1992] Federico Cecconi and Domenico Parisi, Neural networks with motivational units. In Jean-Arcady Meyer, Herbert L. Roitblat, Stewart W. Wilson, editors, *From Animals to Animats 2: Proceedings of the 2nd International Conference on the Simulation of Adaptive Behaviour*, pages 346-355, Cambridge, Mass., 1992. The MIT Press.

[Humphrys, 1997] Mark Humphrys. Action Selection Methods Using Reinforcement Learning. PhD Thesis, Technical Report No. 426, University of Cambridge, Cambridge, 1997.

[Lin, 1992] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8:293-391, 1992.

[Lin, 1993] Long-Ji Lin. Hierarchical learning of robot skills by reinforcement, In Enrique H. Ruspini, editor, *IEEE - Proceedings of the international conference on neural networsks*, pages 181-186, New York, 1993. IEEE.

[Maes, 1990] Pattie Maes. A Bottom-Up Mechanism for Behavior Selection in an Artificial Creature. In Jean-Arcady Meyer and Stewart W. Wilson, editors, *From Animals to Animats: Proceedings of the First International Conference on the Simulation of Adaptive Behavior*, pages 238-246, Cambridge, Mass., 1990. The MIT Press.

[McFarland, 1993] David McFarland. *Animal Behaviour*. Longman, New York, 1993.

[Minsky, 1986] Marvin Minsky. *The Society of Mind*, Simon and Schuster, New York, 1986.

[Nolfi, 1997] Stefano Nolfi. Using emergent modularity to develop control system for mobile robots. *Adaptive Behavior*, 5:343-364, 1997.

[Pavlov, 1927] Ivan P. Pavlov. *Condition reflexes.* Oxford University Press, Oxford, 1927.

[Shultz *et al*., 1997] Wolfram Shultz, Peter Dayan, Read P. Montague, A neural substrate of prediction and reward, *Science*, 275:1593-1599, 1997.

[Sutton and Barto, 1990] Richard S. Sutton and Andrew G. Barto. Time-Derivative Models of Pavlovian Reinforcement. In Michael R. Gabriel and John W. Moore, editors, *Learning and Computational Neuroscience: Foundations of Adaptive Networks*. The MIT Press, Cambridge, Mass., 1990.

[Sutton and Whitehead, 1993] Richard S. Sutton and Steven D. Whitehead. Online learning with random representations, *Proceedings of the Tenth International Conference on Machine Learning*, pages 314-321, Morgan Kaufmann.

[Thorndike, 1911] Edward L. Thorndike. *Animal Intelligence*. Macmillan, New York, 1911.

[Widrow and Hoff, 1960] B. Widrow and M. E. Hoff, Adaptive switching circuits, *IRE WESCON Convention Record,* Part IV, pages 96-104, 1960.