In P. Maes, M. Mataric, J.-A. Meyer, J. Pollack and S. W. Wilson (eds.) *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, Cambridge, MA: The MIT Press/Bradford Books, 1996.

Generalization in Instrumental Learning

Christian Balkenius
Lund University Cognitive Science
Kungshuset, Lundagård
S-222 22 LUND, Sweden
christian.balkenius@fil.lu.se

Abstract

This paper shows how a representation at multiple scales can be used for generalization in instrumental learning. The use of such representations is an efficient way to code similarities and differences within a stimulus dimension, and allows a learning system to generalize easily between various situations. A neural network based model of classical and instrumental conditioning is presented and its ability to generalize using multi-scale representations is subsequently demonstrated in a number of simulations.

1 Introduction

Generalization, put simply, is the ability to behave in a new situation in a way that has been learned in other similar situations in the past. In nature, an animal will never encounter the exact same situation twice and if learning is to be of any use, it is necessary to generalize learned behavior to new situations. Must a study of learning not also be an inquiry into generalization?

I want to suggest that this is indeed the case, and we will see that the key concept in such an investigation is that of similarity. To determine whether it is appropriate to behave in some certain way, an animal must check if the new situation is similar to other situations in a suitable way. Generalization is only possible when this similarity can be recognized in a useful way.

When generalization is studied in the animal laboratory, an animal is first taught to respond to a specific stimulus, for example a tone. Later, it is presented a new stimulus that resembles the original in some sense, for example a tone of a different pitch (Mackintosh 1974). The amount of responding to the new stimulus is used as an index of the level of generalization. Given that some level of generalization is shown, what aspect of the animal nervous system makes this possible? Is this a phenomenon at receptor level or are more central mechanisms involved? Moreover, are there any similarities in the generalization within a single stimulus-dimension, as between two pitches, and in more complex cases, as, for example, when a child mistakes a horse for a dog?

In the next section, I show how stimuli can influence a neural network in a way that supports generalization along stimulus dimensions, although the dimensions themselves need not be explicitly represented. In sections 3 and 4, these constructions will be put to work in a model of conditioning that can generalize in a natural way. A number of simulation studies are presented that show how the model can describe generalization in instrumental learning. It is shown that the ability to generalize greatly increases the learning speed in a simple grid environment.

2 Neural Representation of Similarity

The notion of similarity that will be used here is based on the idea that a stimulus gives rise to a distributed pattern of activity in the nervous system whose components serve to represent different aspects of the stimulus. Such an activity pattern will be called a *sensory schema* (Balkenius 1994) and its different parts are assumed to directly code the values on various *stimulus dimensions*, such as shape and color. This idea is consistent with the view that the nervous system uses *representation by place*, that is, activity at a certain *location* in the nervous system corresponds to some specific stimulus property (Martin 1991).

For example, a specific set of neurons may react each time a tone of a certain pitch is heard. It follows that two

sensory schemata are similar to the extent that their neural realizations physically overlap (Balkenius 1994). Representations of this kind can be called *semi-local* since they share properties with both local and distributed coding schemes (Thorpe 1995). They are distributed in the sense that many nodes are necessary to represent the stimulus, but local in the sense that the activity of each node can be given meaning independently.

In Pavlov's (1927) classical account of generalization, it was assumed that each stimulus is represented by activity in a specific region of the cerebral cortex. Stimulus situations that were similar to each other would give rise to activity in overlapping regions of the cortex. This was the origin of the idea that generalization is based on common elements (Mackintosh 1974). According to this view, every stimulus can be analyzed into a number of components that each has the same properties as a whole stimulus. For example, a red ball can be though of as the compound of one stimulus component for the color red and one for the shape round.

Returning to the example of generalization along the pitch dimension, we see that this could be accomplished if the pitch of each tone is represented as a number of components where some of them are shared with tones of similar pitch. Such a coding is easily constructed if pitch is represented by the activity over a set of detectors, each of which is tuned to a specific frequency (figure 1).

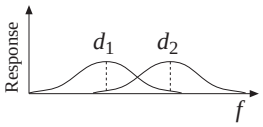

Figure 1 The response characteristics of two tuned detectors, d_1 and d_2. As a tone is moved along the frequency dimension (f), the responses of the detectors gradually change in the way specified by the tuning curve of each detector.

Since the tuning curves of the detectors overlap, each pitch will activate a whole number of detectors with strengths that depend on how well the stimulus pitch matches the preferred frequency for that detector. This classical account for generalization has a number of attractive properties (Mackintosh 1974), but without a more detailed characterization, it is impossible to say whether representations of this type can support generalization in more than the simplest cases.

For example, this coding scheme is biased toward a specific resolution in the representation of a dimension. It may be able to explain the generalization to neighbouring pitches, but it cannot explain how an animal can generalize to a tone of any pitch. Furthermore, it assumes that there exists an objective stimulus dimension to which the detector can be tuned. This may not be much of a problem for the representation of pitch, but what is the dimension underlying dog-ness or horse-ness?

To simplify the presentation below, I will continue to assume that there exists some objective stimulus dimension. In section 5, however, this requirement will be relaxed and I will describe how generalization can be viewed as a mechanism that *constructs* subjective stimulus dimensions.

To allow a fuller representation of a stimulus dimension, the notion of a scale-hierarchy is required. Each dimension must be represented by a collection of scales, each of which codes for the same value on the corresponding stimulus-dimension but with varying resolution. Representations of this type have been used for a long time in artificial vision (Witkin 1983). The different scales correspond to the visual image projected onto different frequency domains. In the coarse scales, the image is 'blurred' and does not contain any fine details. Different objects thus become more similar. In the finer scales, more details are retained, which makes it simpler to discriminate between different objects although an object may be impossible to identify if it is viewed from a slightly altered angle. One of the key ideas of the present work is to investigate if this type of *multi-scale* representations can be used more generally.

Given an initial neural activity pattern at the sensory receptors, a stimulus representation at multiple scales can be generated in one of two ways. The first is to use sets of neural detectors that are tuned to each of the relevant scales. This is illustrated in figure 2a. The second strategy uses detectors only for the finest scale and derives the coarser scales using a network lattice as shown in figure 2b. To accomplish this, the nodes on each higher level should react when one of its two nodes in the finer scale is active, that is, it should react to the *disjunction* of each of its parts in the finer scale. However, this strategy has the disadvantage that it uses a hard-wired scale-hierarchy. Such a network can only be constructed if the topology of the sensory space is known *a priori*.

3 A Neural Network Model of Conditioning

Armed with the appropriate representations for generalization we must now consider a learning model capable of using this type of representations. It is now generally acknowledged that behavior in higher animals is controlled both by associative mechanisms related to those studied within the behavioristic tradition and by higher cognitive processes. Stimulus generalization plays a crucial role in both these types of control, but for the sake of simplicity, I will only consider the first type of learning here.

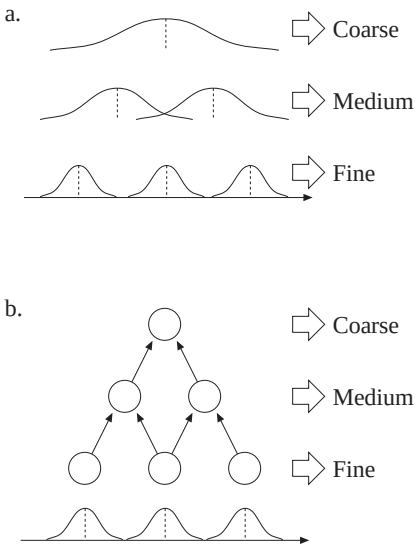

Figure 2 Two ways to derive stimulus representations at multiple scales. (a) Different sets of detectors are independently tuned to various scales. (b) The coarse representation is derived from a finer-scale representation in a feed-forward network where each node reacts to the disjunction of each of its parts at the finer level.

There are a number of requirements that a learning system must fulfill to use multi-scale representations. The first two requirements are shared with other models of instrumental learning:

1. It must be able to chain the responses required to move from the start to the goal.

2. Learning must be reversible.

The following four requirements makes sure the model can handle multi-scale representations:

3. It must operate with multiple-element stimulus representations, that is, sensory schemata.

4. The learning needs to be invariant to the size of the schemata. Learning should not be influenced by how many components are used for each schema.

5. The learning system should be able to discriminate between the different schema components. In this context, this means that it must be able to select the appropriate scale for its current problem.

6. The system must learn at different time scales. This final requirement is somewhat technical. Since the coarse representations do not necessarily change every time an action is performed, it is necessary that the system can learn even if there is a substantial time interval between changes in this representation.

Figure 3 shows an artificial neural network with the desired properties. It can also model many aspects of classical conditioning such as acquisition, extinction, conditioned inhibition, overshadowing, blocking (See Balkenius 1995 and 1996). Since my primary interest here is its ability to generalize, I will not dwell on these matters. However, it will be necessary to consider how the network models acquisition, extinction and secondary conditioning since these abilities are necessary for the generalization experiments described below. The network will be called a *reinforcement module* since its task is to calculate reinforcement, that is, signals that changes the state of the network, from changes in its input. The function of the module is related both to Gray's (1975) symmetrical model of conditioning and to Grossberg's (1987) dipole theory.

3.1 Acquisition

In the simplest form of a conditioning experiment, a conditioned stimulus, CS, is paired a number of times with an unconditioned stimulus, US, until the presentation of the CS on its own is able to produce a conditioned response, CR. The US can be either positive or negative depending on its consequences for the animal. Since the same stimuli can be used in instrumental conditioning as reward or punishment, they can alternatively be called Rew or Pun. In a typical acquisition experiment, the CS is paired with a rewarding US.

It has been shown in a number of studies that conditioning is easiest when there is a short interstimulus interval between the two stimuli (Mackintosh 1974). This is modelled in the network in the simplest possible way by demanding that the US should follow the CS by one time-step. This approximation is admittedly too crude for a more detailed modelling of interstimulus interval effects, but it will be sufficient for my present purposes.

The conditioning network consists of four nodes, x^+, x^-, d^+, and d^- and a number of connections between them. Following Pavlov's (1927) account for conditioned inhibition, there are two symmetrical parts in the model. The excitatory part is responsible for the initial acquisition of a response and the inhibitory part is responsible for the build-up of a connection to the inhibitory part of the network during extinction. The role of the inhibitory system is thus to cancel a response that is no longer adaptive. This implies that the state of the network is different in its naive state and when it has first acquired but later extinguished a response.

There are two types of inputs to the network. The signals marked CS_1 and CS_2 represent stimulus components and can activate the two nodes x^+ and x^- through plastic connections.

The signals through each of these connections are multiplied with their weights, *w*, before they are added at *x*.

$$x^+(t) = \Sigma_{ij}\, CS^+_i(t)\, w^+_{ij}(t). \quad (1)$$

Initially, the weights on these connections are assumed to be very small. Before any learning, a stimulus will not activate either x^+ or x^-. The other type of input controls the learning process. These signals are denoted by Rew and Pun. The output of the network, the CR, is given by the difference between the activity on the excitatory side of the network at x^+ and the activity in the inhibitory side at x^-, i. e., $CR=x^+-x^-$.

The learning dynamics during acquisition are implemented by the static connections among the four nodes in the following way. (Since the network is symmetrical, the equations for the inhibitory part are easily derived from the ones describing the excitatory side. I will only describe the equations for the positive part here.) Given that no prior learning has occurred, all plastic weights are small.

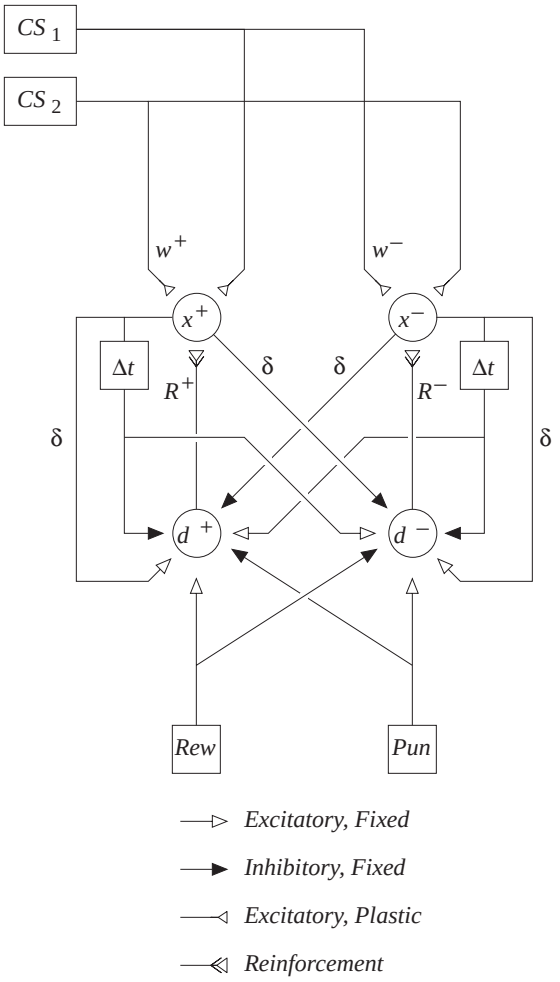

Figure 3 A neural-network model of classical conditioning. (See the text for explanation.)

The activation of, for example, CS_1 which is immediately followed by an US (or Rew) will generate activity at d^+, since,

$$d^+(t) = Rew(t) - x^+(t\text{-}1) + \delta\Delta x^+(t) + x^-(t\text{-}1) - Pun(t) - \delta\Delta x^-(t), \quad (2)$$

This equation describes a symmetrical extension of the delta-rule (Widrow and Hoff 1960/1988) which is also able to account for secondary conditioning using the $\delta\Delta x^+(t)$ and $\delta\Delta x^-(t)$ signals (see below). The activity at d^+ will generate a positive reinforcement signal.

$$R^+(t) = max(d^+(t), 0), \quad (3)$$

This signal will strengthen the connection between CS_1 and x^+ according to the following equation,

$$w^+_{ij}(t\text{+}1) = w^+_{ij}(t)+\gamma^+ R^+(t)CS_i(t\text{-}1), \quad (4)$$

where γ is the learning rate parameter.

Consistent with Rescorla-Wagner theory (Rescorla and Wagner 1972), this process will continue until the activity of x^+ becomes sufficiently strong to inhibit the reinforcement at d^+ through the time delayed connection from x^+. The signal from x^+ to d^+ can, thus, be seen as an expectation that the US will following CS_1.

The goal of the circuit is thus to let the output from x^+ agree with the reward given after the presentation of a CS. The learning process terminates when the output from x^+ has reached the same level as the Rew signal (See figure 4).

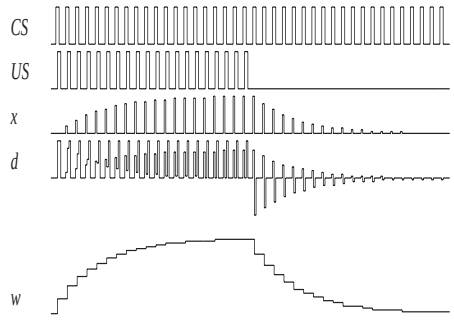

Figure 4 Computer simulation of acquisition and extinction in the neural network model described in the text. CS and US are the intensity of the stimuli and the values x and d are plotted as the difference between the activity of the two sides of the network, thus, CR = $x = x^+-x^-$, $d = d^+-d^-$. Similarly, the associative strength w is plotted as w^+-w^-.

3.2 Extinction

In an extinction experiment, an animal is first taught to respond to a conditioned stimulus, CS. Thereafter, the CS is presented on its own without the US. This will cause the CR to gradually disappear until the animal no longer responds (Mackintosh 1974).

During extinction, the role of the inhibitory side of the network becomes apparent. Since the network has previously learned that a reward follows CS_1, the presentation of CS_1 will cause the node x^+ to become active. This node will in turn activate d^- after a time delay of Δt. If this were an acquisition trial, the activity of d^- would be cancelled by the reward, but since it is now omitted, the activity of d^- will prevail and cause a negative reinforcement signal R^- to be generated. This signal will act as its positive counterpart, but this time, learning will take place between CS_1 and x^- and results in a decrease in responding.

The inhibitory learning will continue as long as the activity at d^- is larger than that at x^-. When the process comes to an end, the weight on the connection from CS_1 to x^- will have reached the same level as the weight from CS_1 to x^+, and no CR will be produced (See figure 4).

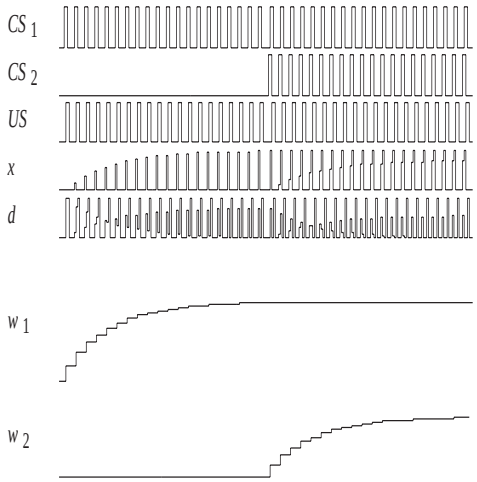

Figure 5 Secondary Conditioning. (See text for explanation.)

3.3 Secondary Conditioning

In secondary conditioning, a stimulus, say CS_1 is first paired with the US until a steady level of responding is achieved. At this point, a second stimulus CS_2 is introduced which is subsequently paired with CS_1 without the US being present. It is well known that this procedure causes CS_2 to produce the CR on its own (Mackintosh 1983). The typical explanation of this phenomenon is that CS_1 has acquired reinforcing properties during its previous pairing with the US and is now able to act as if it was a US.

This is modelled by the network in the following way (See figure 5). The activity generated at x^+ is sent to d^+ by two different pathways. The first, which was described above, is delayed by one time-step and is used to inhibit the reward. This signal causes the positive reinforcement to be shut off when the weight on the excitatory connection has reached its final value. The second pathway is excitatory with the weight δ. This signal is sent to the reinforcement node d^+ without delay and acts as secondary reward.

This signal, called Δx^+, is calculated as specified in equation (1) except that only positive *changes* in each CS is used and not its absolute level. In the simulations reported below, this means that Δx^+ is given a positive contribution only at the on-set of a CS component. In this way, the activity at x^+ is first used as secondary reward and immediately thereafter as expected reward. Since secondary reward and punishment are connected exactly as primary reward and punishment, they all share the same properties.

As discussed by Klopf (1988), it is necessary that CS_1 is followed by the US during secondary conditioning. If not, the reinforcing properties of CS_1 would gradually become extinguished. This, in turn, would cause the secondary association from CS_2 to vanish after some initially successful trials of secondary learning.

The connection weight δ, which should be less than 1, functions as a discount factor (cf. Sutton and Barto 1990). This weight makes sure that a secondary reward is never as large as the primary one. As a result, secondary reward will be worth δUS. More generally, an nth order reward will be worth $\delta^{(n-1)}US$. The result of this design is that the reinforcement module sets up a temporal gradient around the terminal event, the presentation of the US. In the next section, we will see how such a temporal gradient can be converted to a goal gradient that can guide sequential behavior.

3.4 Instrumental Learning

In an instrumental learning experiment, the animal is placed in the experimental apparatus and is allowed to behave freely. When the animal performs a certain response, it will be rewarded. This causes that specific response to become more likely when the animal finds itself in the same situation at a later time.

In Mowrer's (1960/1973) influential two-process theory of learning, instrumental conditioning was assumed to proceed in two steps. First, the positive emotional aspects of the rewarded situation would be learned by classical conditioning. In the second step, the rewarding properties of the situation were used to reinforce the appropriate behavior. Klopf, Morgan and Weaver (1993) presented a neural network architecture that can transform a model of classical conditioning into one of instrumental learning in this way. The method described is to implement the two-process idea in an artificial neural network.

Here, I will extend the reinforcement module presented above with such a network (figure 6). To admit instrumental learning, a second learning process is added between the

conditioned stimuli and responses. This second learning system is controlled by the reinforcement module described above. By shunting the connections from CS to *x* with the selected response, the activity in the classical conditioning network is determined by the conjunction of the currently perceived stimulus components *and* the response produced.

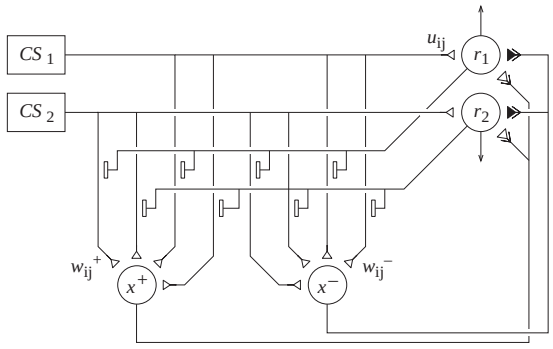

Figure 6 A neural network model of instrumental conditioning. A secondary learning system is added to the reinforcement module shown in figure 3. The signals from CS to *x* are shunted by the connections from the selected response. The nodes x^+ and x^- are the ones shown in figure 3.

This is modelled by replacing (1) above with,

$$w^+_{ij}(t+1) = w^+_{ij}(t)+\gamma^+ R^+(t)CS_i(t-1)r_j(t), \qquad (1')$$

where $r_j(t)$ is 1 if response j has been selected at time t, and 0 otherwise.

When the animat produces a certain behavior, the secondary learning system tries to make the connection strength between the current situation and the performed response approach that of the discounted reward of the subsequent response. In effect, instrumental learning can be seen as a case of secondary conditioning of a response to a specific stimulus configuration. Since classical conditioning is seen as an important part of instrumental conditioning, all the properties of classical conditioning carry over to this case.

The associations from stimuli to responses, u_{ij}, change according to the following equation,

$$u_{ij}(t+1) = u_{ij}(t)+\gamma[x^+(t)-x^-(t)]CS_i(t-1)r_j(t). \qquad (6)$$

Their role is to predict the discounted reward for each possible response in the current situation. These associations are used to select the response at each time step. First, the activation, ρ_j, of each response is calculated as,

$$\rho_j(t) = \Sigma_i\, CS_i(t)\, u_{ij}(t). \qquad (7)$$

The response to perform is subsequently selected according to a Bolzmann distribution generated by these activities ($\rho_j(t)$).

When the network learns a spatial behavior sequence, the associative strengths learned by the network can be seen as a goal-gradient (cf. Hull 1932). Responses closer to the goal receive larger associative strengths than responses further away. It is this decreasing gradient of secondary reward that serves to chain responses into a sequence. In this respect, the behavior of the network is very similar to the Q-learning algorithm (Watkins 1992) as well as to temporal-difference learning (Sutton and Barto 1990). The derivation of the model is described in more detail in (Balkenius 1996).

4 Simulations

A number of important questions can be asked about the model presented above. Can it exploit a multi-scale representation? Does generalization interfere with relearning? Is the network able to select different scales in different situations? To try to answer these questions, a number of simulations have been run of an animat moving in a simple grid-like environment.

The primary reason for using such an environment is that it has been much studied in the literature. It is also very easy to understand since it offers a description in terms of spatial orientation. However, this type of environment reflects the intricacies of spatial orientation only in a very superficial way, and the simulations are better interpreted in a more generic way.

To make generalization possible, each location is coded at multiple scales. A number of grids with varying resolutions generate the sensory input to the animat. The finest scale is a precise representation of the current location of the animat. This is the type of input usually employed in simulations of reinforcement learning. Grids of lower resolution generate the coarser representations (figure 7).

In the 8×8 environment, the current stimulus of the animat is coded by a binary array with 85 components, $CS_0 \ldots CS_{84}$. The finest scale uses 64 of these components to code the exact location of the animat. The next scale divides the environment into 16 regions for a slightly more blurred place representation, and so on. At all times, each scale sets exactly one of its stimulus components to 1. The remaining part of the input array is set to 0. Four components are thus active at any time.

The result of this coding is similar to the one generated by the network depicted in figure 2b. Using this coding strategy, the similarity of two locations is a function of the number of common regions at the coarser scales. Note that in this example, this is not the same as spatial proximity.

4.1 Initial Learning

A number of simulations were run with environments with 16, 64, and 256 states to compare the learning speed with and without multi-scale representations. In all simulations, the start was placed in the upper left corner of the square environment. To make generalization as useful as possible, the initial simulations used a goal in the upper right corner. The shortest path from start to goal were 4, 8, and 16 respectively. With this placement of start and goal, only the coarsest scale is necessary to solve the problem. Since the correct strategy is to move to the right at all locations along the path from the start to the goal, the finer scales are not necessary.

In all simulations, four responses where available to the animat: move-north, move-south, move-east, and move-west. During the simulation, the animat was moved back to the start location every time it entered the goal region. Each sequence of movements from the start to the goal would count as one trial.

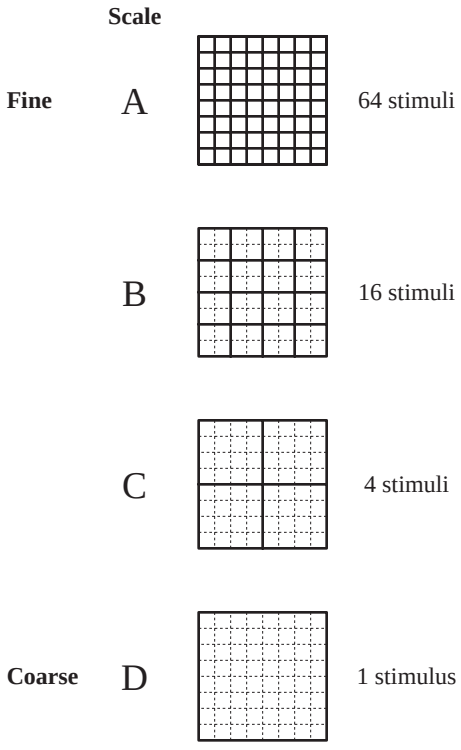

Figure 7 The coding of location at multiple scales makes generalization possible. A grid environment is simultaneously coded at four spatial resolutions. In the finest scale, 64 different stimuli are generated. At the most coarse scale, all locations in the environment fall in the same region and are coded by the same stimulus.

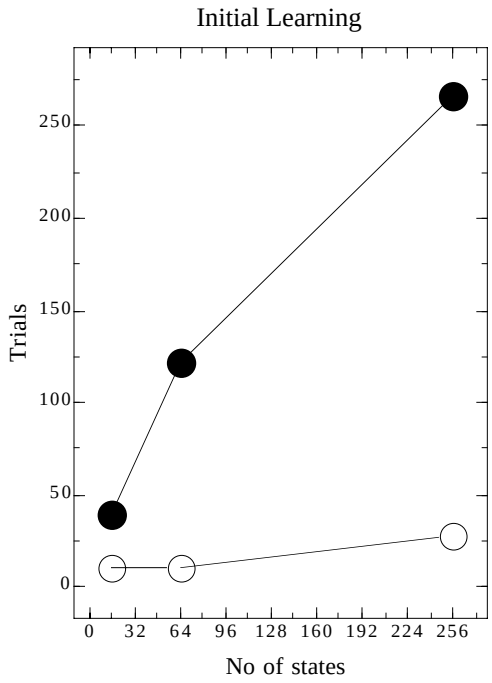

Figure 8 Learning speed with and without multi-scale representationas a function of the size of the environment (○ with multi-scale representation, ● without multi-scale representation). Each data point represents the average time needed to learn the environment calcualted over 10 simulations each.

Figure 8 shows the number of trials required for the animat to learn the shortest path from the start to the goal with and without generalization. The best strategy was found very fast when generalization was used. This simulation thus shows that the network is able to exploit the coarser representations when possible.

To avoid any possible advantage resulting from the larger number of components in the input array, simulations were also run in which the stimulus vector had been normalized before it was sent to the neural network. This had no effect on the learning speed which shows that the learning system satisfies the fourth requirement above in section 3.

Less satisfactory is the sensitivity to the temperature of the Bolzmann distribution. It turned out to be impossible to find a set of parameters that would work for both single-scale and multi-scale learning in the larger environment. In the simulation shown in figure 8, the temperature parameter was set to T=0.2 for the multi-scale learning and to T=0.02 when only the finest scale was used. When the scale-hierarchy is generated internally by the animat, this is not too much of the problem, but this is still a difficulty that needs to be addressed in future research.

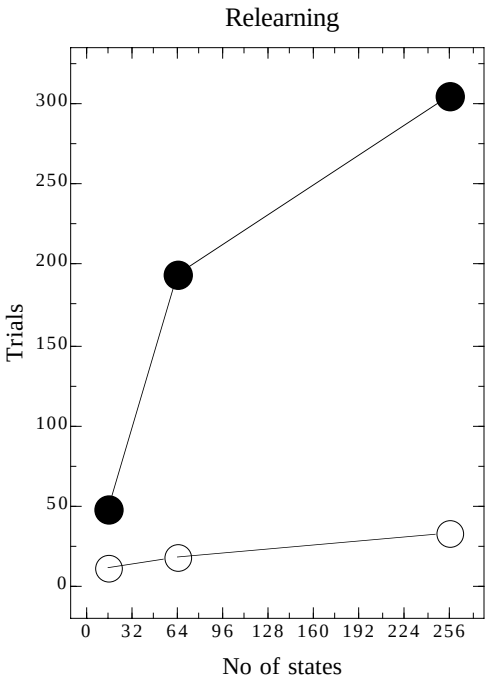

Figure 9 Speed of relearning after the goal was moved with and without multi-scale representation (○ with multi-scale representation, ● without multi-scale representation). Each data point represents the average time needed to relearn the environment and is the average of 10 simulations each.

4.2 Relearning

The second set of simulations investigated whether the multi-scale representation would interfere with relearning when the environment was changed. In these simulations, the initial location of the goal was first learned as above. When the animat had reached its optimal behavior, the goal was moved to the lower left corner of the environment and the network was allowed to relearn its location. As can be seen in figure 9, the ability to generalize was useful also during relearning. The network using the multi-scale representations learned the new location of the goal much faster that the one using only the fine scale. Multi-scale representations are thus useful both during learning and relearning.

4.3 Scale Discrimination

The first two simulations show that the network can utilize the coarsest scale when the best behavior strategy can be found at this scale. For generalization to be useful, however, the network must find the best scale in each situation. The final set of simulations was run to check if the network could discriminate between different scales and select the appropriate ones in a number of different environments. Each environment was learned with a variable number of active scale-representations to find how much each scale contributed to the solution of the problems.

Figure 10 shows the different environments and the changes in learning speed for the different number of active scales. Each diagram can be considered as a *scale-profile* for the corresponding environment. It shows how much each scale is used in the learning task. Note that such scale-profiles depend both on the structure of the environment *and* how that structure is represented by the animat. Recall that the sensory representation of each state is given by the arrangement described in figure 7. The environment contributes its share to the scale-profile through the location of the start and the goal together with the obstacles that constrain movement in the environment.

In the first simulation, the role of the different scales in an empty environment was tested. Although the coarsest scale (D) would in principle be sufficient to solve this problem, all scales were in fact used. As shown in figure 10, all the involved scales had an effect on the learning speed.

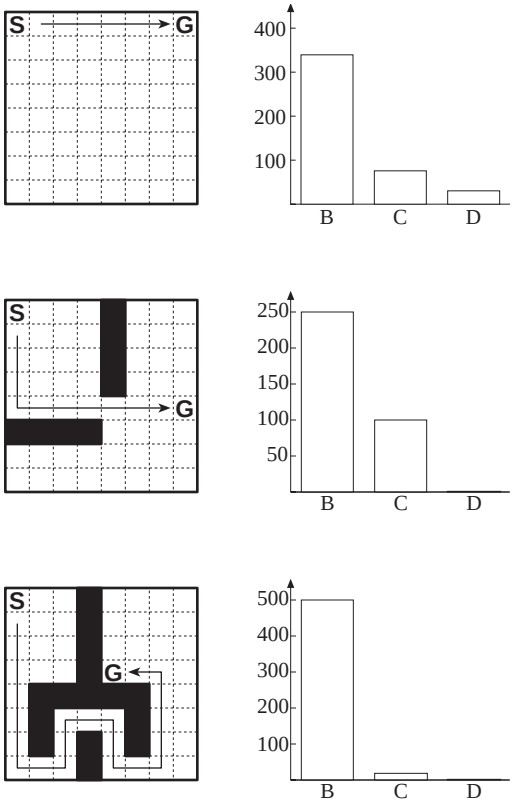

Figure 10 Scale-profiles for three different environment. The diagrams show how much faster learning became when each new scale was activated. The first bar shows how much faster the network learned when scale B was added to scale A (see figure 7). The second bar shows how much faster learning became when scale C was added to A and B, and so on.

Since the behavior performed by the animat is drawn from a probability distribution, there will always be some probability that it will deviate from the optimal path. In these cases, it can use the finer scales to direct it back to the optimal path.

In the second environment, two obstacles were introduced to see what effect they would have on the learning. As could be expected, the coarsest scale could no longer be used. However, all the other scales contributed to the solution. Finally, the model was tested in an environment with much fine structure. Here, only the two finest scales (A and B) were used to any large extent.

These profiles obviously give rather crude pictures of how the different scales are used, but they clearly show that multi-scale representations can be exploited by the learning model in instrumental learning. They do not, however, show the many complex interactions between the different scales at different locations. To investigate these mechanisms further, more advanced methods of analysis must be developed.

5 Discussion

The simulations reported above show that multi-scale representations can be very useful in instrumental learning. The important property of this type of representation is that it generates a rich set of similarities and differences (Balkenius 1994). More generally, two stimuli can be learned as similar if there exists a scale in which both are coded by the same node. The same two stimuli can also be learned as different if there exists a scale in which they are coded by different nodes.

An interesting change of perspective is to view any neural network in which a number of disjunctive nodes are connected in series as a network for multi-scale representations. When representations are viewed in this way, it becomes possible to describe a stimulus dimension, not as an objective dimension outside the animal, but instead as a collection of subjective similarities and differences. Any such collection of similarities and differences that is sufficiently dense would act as a subjective stimulus dimension, although no explicit representation of that dimension is necessary in the network. Such dimensions need not necessarily correspond in a simple way to a single observable stimulus dimension.

Another aspect of the proposed representations is that behavior can be seen as a number of overlaid behavior tendencies. At each location in the environment, every scale contributes with its own preferred behavior. The response performed is determined by the sum of these, possibly conflicting, behavior tendencies.

In the simulations above, both the environment and the responses where discrete. In the continuous case, this aspect of the model becomes even more attractive. In this approach, the goal of the learning system is to coordinate the continuous outputs of behavior modules instead of generating fixed responses, and the output from the learning system functions as parameters for an arbitration process using *additive composition* (Balkenius 1995).

This suggests that the ideas presented above could be generalized to a learning method in which the animat would first learn behavior at the coarsest scale. In situations where this would not produce the desired behavior, the finer scales could be superimposed to adjust the behavior strategy further. For each finer scale that is included, the residual error would decrease until performance reaches the desired level.

6 Conclusion

I have shown that instrumental learning can merit a lot from the introduction of scale-hierarchies in the representation of stimuli. When stimuli are represented in a way that supports a rich set of similarities and differences, generalization and discrimination become possible between a large set of stimuli.

Computer simulations have shown that this allows a neural network to learn the appropriate behavior in a grid environment much faster that if all states are coded as completely different. When similarity is represented in a scale-hierarchy, learned behavior is automatically generalized also to states that have not yet been visited. When the generalization is correct, learning will proceed much faster. If it is not, the fine scales will be used to correct the over–generalization.

In none of the performed simulations did this over-generalization make learning slower than without generalization. At worst, the generalizing system appears to do as well as the same learning system without generalization.

Acknowledgements

The support from the Swedish National Board for Industrial and Technical Development (NUTEK) is gratefully acknowledged. Thanks to Peter Gärdenfors, Lars Kopp, Janne Morén and the rest of the Cognitive Science group at Lund University for helpful discussions.

References

Balkenius, C., (1994), "Some properties of neural representations". In L. F. Niklasson and M. B. Bodén (eds.) *Connectionism in a broad perspective*, 79-88, New York: Ellis Horwood.

Balkenius, C., (1995), "Natural intelligence in artifical creatures", *Lund University Cognitive Studies 37*.

Balkenius, C., (1996), "A neural network model of classical conditioning I: The dynamics of learning", *Lund University Cognitive Studies , to appear*.

Gray, J. A., (1975), *Elements of a two-process theory of learning*, London: Academic Press.

Grossberg, S., (1987), *The adaptive brain*, Amsterdam: North-Holland.

Hull, C. L., (1932), "The goal-gradient hypothesis and maze learning", *Psychological Review*, **39**, 25-43.

Klopf, A. H., (1988), "A neuronal model of classical conditioning", *Psychobiology*, **16**, 85-125.

Klopf, A. H., Morgan, J. S. & Weaver, S. E., (1993), "A hierarchical network of control systems that learn: modelling nervous system function during classical and instrumental conditioning", *Adaptive Behavior*, **1**, 263-319.

Mackintosh, N. J., (1974), *The psychology of animal learning*, New York: Academic Press.

Mackintosh, N. J., (1983), *Conditioning and associative learning*, Oxford: Oxford University Press.

Martin, J. H., (1991), "Coding and processing of sensory information". In E. R. Kandel, J. H. Schwartz and T. M. Jessel (eds.) *Principles of neural science*, 329-340, New York: Elsevier.

Mowrer, O. H., (1960/1973), *Learning theory and behavior*, New York: Wiley.

Pavlov, I. P., (1927), *Conditioned reflexes*, Oxford: Oxford University Press.

Rescorla, R. A. & Wagner, A. R., (1972), "A theory of Pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement". In A. H. Black and W. F. Prokasy (eds.) *Classical conditioning II: current research and theory*, 64-99, New York: Appleton-Century-Crofts.

Sutton, R. S. & Barto, A. G., (1990), "Time-derivative models of Pavlovian reinforcement". In M. Gabriel and J. Moore (eds.) *Learning and computational neuroscience: foundations of adaptive networks*, 497-538, Cambridge, MA: MIT Press.

Thorpe, S., (1995), "Localized versus distributed representations". In M. A. Arbib (ed.) *The handbook of brain theory and neural networks*, Cambridge, MA: MIT Press.

Watkins, C. J. C. H., (1992), "Q-learning", *Machine Learning*, **8**, 279-292.

Widrow, B. & Hoff, M. E., (1960/1988), "Adaptive switching circuits". In J. A. Anderson and E. Rosenfeld (eds.) *Neurocomputing: foundations of research*, 123-134, Cambridge, MA: Mit Press.

Witkin, A., (1983), "Scale-space filtering". *8th International Joint Conference on Artificial Intelligance*, 1019-1022.