

Invariant object recognition in the visual system with error correction and temporal difference learning

Edmund T Rolls¹ and Simon M Stringer

Department of Experimental Psychology, Oxford University, South Parks Road, Oxford
OX1 3UD, UK

E-mail: Edmund.Rolls@psy.ox.ac.uk

Received 20 July 2000, in final form 23 February 2001

Abstract

It has been proposed that invariant pattern recognition might be implemented using a learning rule that utilizes a trace of previous neural activity which, given the spatio-temporal continuity of the statistics of sensory input, is likely to be about the same object though with differing transforms in the short time scale. Recently, it has been demonstrated that a modified Hebbian rule which incorporates a trace of previous activity but no contribution from the current activity can offer substantially improved performance. In this paper we show how this rule can be related to error correction rules, and explore a number of error correction rules that can be applied to and can produce good invariant pattern recognition. An explicit relationship to temporal difference learning is then demonstrated, and from this further learning rules related to temporal difference learning are developed. This relationship to temporal difference learning allows us to begin to exploit established analyses of temporal difference learning to provide a theoretical framework for better understanding the operation and convergence properties of these learning rules, and more generally, of rules useful for learning invariant representations. The efficacy of these different rules for invariant object recognition is compared using VisNet, a hierarchical competitive network model of the operation of the visual system.

1. Introduction

1.1. Background

There is now much experimental evidence to suggest that over a series of successive cortical stages the visual system develops neurons that are able to respond with considerable view, size and position invariance to objects or faces (Desimone 1991, Rolls 1992, Rolls and Tovee 1995, Tanaka *et al* 1991, Rolls and Treves 1998, Booth and Rolls 1998). Rolls has proposed (Rolls 1992, 1994, 1995) that such transform-independent selectivity could arise from the following:

¹ Corresponding author.

(i) A series of competitive networks, organized in hierarchical layers, and incorporating mutual inhibition over a short range within each layer as part of the local competition. These networks allow combinations of features or inputs occurring in a given spatial arrangement to be learned by neurons, ensuring that higher-order spatial properties of the input stimuli are represented in the network. (ii) A convergent series of connections from a localized population of cells in preceding layers to each cell of the following layer, thus allowing the receptive field size of cells to increase through the visual processing areas or layers. (iii) A modified Hebb-like learning rule incorporating a temporal trace of each cell's previous activity, which, it is suggested, will enable the neurons to learn transform invariances (see also Földiák (1991) and Wallis *et al* (1993)). The trace rule is designed to enable neurons to learn from the spatio-temporal statistics of the natural visual inputs, which in short time periods are likely to be about the same object. (Earlier work by Sutton and Barto (1981) had explored the use of a trace learning rule for modelling the temporal properties of classical conditioning. They investigated incorporation of the trace in the presynaptic neuron or in the post-synaptic neuron, or both. Mitchison (1991) investigated the use of a different, anti-Hebb, learning rule to remove the effects of temporal variation in the input.) The proposal was tested by Wallis and Rolls (1997) in a model (VisNet) of ventral stream cortical visual processing. It was found that the network could produce view-invariant neurons that responded to some but not other stimuli as long as the Hebbian learning rules incorporated a trace of recent cell activity, where the trace is a form of temporal average. Further work with this trace rule is presented in Wallis (1996) and Wallis and Baddeley (1997).

Recently, it has been demonstrated (Rolls and Milward 2000) that a modified Hebbian rule which incorporates a trace of activity only from the preceding time step can offer substantially improved performance over the standard trace rules described in Földiák (1991) and Wallis and Rolls (1997). In this paper we show the relation of the modified Hebbian rule presented by Rolls and Milward (2000) to a particular form of error correction rule. This leads to an investigation of a number of more general forms of error correction learning rules, all of which produce substantially improved performance. We note that one of the rules is closely related to a rule developed in Peng *et al* (1998) for invariance learning. Some rules can offer high performance using an exponential trace of neuronal firing rate from the current time step, while other rules do not need to use an exponential trace of the neuronal firing rate at all but only a trace based on the activity of the immediately preceding trial. Next, we are able to make explicit a relationship to temporal difference learning (Sutton 1988, Sutton and Barto 1998), and from this we derive further temporal-difference-inspired learning rules. This relationship to temporal difference learning allows us to begin to exploit established temporal difference analyses to provide a theoretical framework for better understanding the operation and convergence properties of learning rules which incorporate a trace of previous activity in order to perform invariance learning.

The efficacy of these different learning rules is compared using VisNet, a model of the operation of the ventral visual system in invariant object recognition, which incorporates some of the architectural features of the primate ventral visual system (Wallis and Rolls 1997, Rolls 2000).

1.2. Methods—VisNet, training and test procedure

In this section we give an overview of the VisNet model; full details are provided by Wallis and Rolls (1997) and Rolls and Milward (2000). In particular, the simulations performed in this paper use the latest version of the VisNet model (VisNet2) with the same parameter values as given in Rolls and Milward (2000). The model consists of a feedforward hierarchical series of four layers of competitive networks (with 1024 neurons per layer), corresponding in the

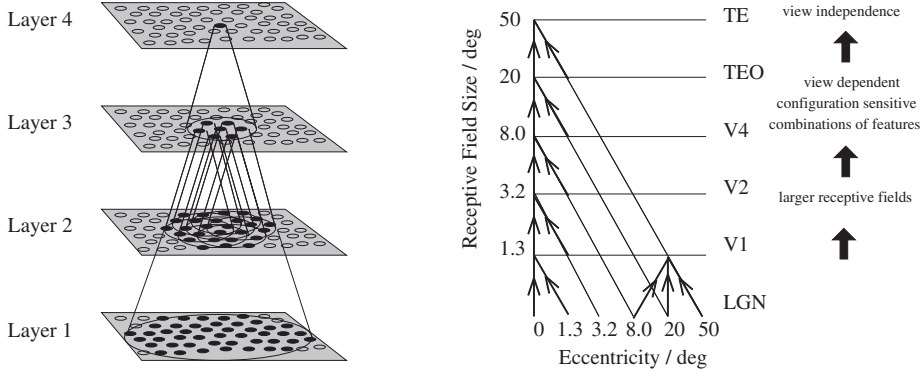


Figure 1. Left: stylized image of the VisNet four-layer network. Convergence through the network is designed to provide fourth-layer neurons with information from across the entire input retina. Right: convergence in the visual system—adapted from Rolls (1992). V1: visual cortex area V1; TEO: posterior inferior temporal cortex; TE: inferior temporal cortex (IT).

primate visual system to V2, V4, the posterior inferior temporal cortex, and the anterior inferior temporal cortex, as shown in figure 1. The first layer of the model receives its inputs from an input layer which provides a representation comparable to that found in V1. The forward connections to individual cells are derived from a topologically corresponding region of the preceding layer, using a Gaussian distribution of connection probabilities. Within each layer competition is graded rather than winner-take-all, and is implemented in two stages. First, to implement lateral inhibition the activation of neurons within a layer is convolved with a local spatial filter which operates over several pixels. Next, contrast enhancement is applied by means of a sigmoid activation function where the sigmoid threshold is adjusted to control the sparseness of the activities or firing rates to values that are approximately 0.01 for the first two layers, and 0.1 for layers 3 and 4 (for details see Rolls and Milward (2000)). (We note that the outputs of the neurons are scalar variables that could be called neuronal activities or neuronal responses, and which would be directly related to the firing rate of the neuron in an integrate-and-fire model. When we refer to firing rates in this paper, we are referring to the values of these scalar variables.) The sparseness of the firing within a layer is defined by Rolls and Treves (1998) as

$$a = \frac{(\sum_i y_i/n)^2}{\sum_i y_i^2/n}$$

where n is the number of neurons in the layer.

The mechanism for transform-invariant object recognition proposed by Földiák (1991) and Rolls (1992) relies on the spatio-temporal statistics of natural visual input. In particular, in the real world different views of the same object are likely to occur in temporal proximity to each other. Then, if synaptic learning rules are utilized that encourage neurons to respond invariantly to temporally proximal input patterns, such neurons should learn to respond invariantly to different views of individual stimuli. The original trace learning rule used in the simulations of Wallis and Rolls (1997) took the form

$$\Delta w_j = \alpha \bar{y}^\tau x_j^\tau \quad (1)$$

where the trace \bar{y}^τ is updated according to

$$\bar{y}^\tau = (1 - \eta)y^\tau + \eta\bar{y}^{\tau-1} \quad (2)$$

and we have the following definitions:

- x_j : j th input to the neuron.
- \bar{y}^τ : Trace value of the output of the neuron at time step τ .
- w_j : Synaptic weight between j th input and the neuron.
- y : Output from the neuron.
- α : Learning rate. Annealed between unity and zero.
- η : Trace value. The optimal value varies with presentation sequence length.

The parameter $\eta \in [0, 1]$ controls the relative contributions to the trace \bar{y}^τ from the instantaneous firing rate y^τ and the trace at the previous time step $\bar{y}^{\tau-1}$, where for $\eta = 0$ we have $\bar{y}^\tau = y^\tau$ and equation (1) becomes the standard Hebb rule

$$\Delta w_j = \alpha y^\tau x_j^\tau. \quad (3)$$

Rolls and Milward (2000) demonstrated that VisNet's performance could be greatly enhanced with a modified Hebbian learning rule that incorporated a trace of activity from the preceding time steps, with no contribution from the activity being produced by the stimulus at the current time step. This rule took the form

$$\Delta w_j = \alpha \bar{y}^{\tau-1} x_j^\tau. \quad (4)$$

In our simulations η is set to 0.8. Also, to avoid interference between different stimuli, the trace \bar{y}^τ is reset to zero each time the stimulus is changed, although this does not generically affect the operation of VisNet (Rolls and Milward 2000). This issue is considered further in the discussion.

The stimuli used for training and testing in VisNet simulations in this paper are a set of seven faces previously used in Wallis and Rolls (1997) and Rolls and Milward (2000). To train the network, a stimulus was presented in a sequence of nine locations in a square grid across the retina, starting at a random position within its sequence. At each location, VisNet calculated the activation of the neurons, then their firing rates using a sigmoid activation function, and then updated the synaptic weights. After all the training locations for a given stimulus had been visited, another stimulus was selected at random and the procedure was repeated. Training in this way for all the stimuli in a set comprised one training epoch, and the numbers of training epochs for layers 1–4 were 50, 100, 100 and 75 respectively, as used by Rolls and Milward (2000).

The performance of the network was assessed with two information-theoretic measures, as described by Rolls and Milward (2000). The measures assess the extent to which either a single cell, or a population of cells, responds to the same stimulus invariantly with respect to its location, yet responds differently to different stimuli. The measures effectively show what one learns about which stimulus was presented from a single presentation of the stimulus at any randomly chosen location. Results for top (fourth) layer cells are shown in this paper. High information measures thus show that cells fire similarly to the different transforms of a given stimulus (object), and differently to the other stimuli. The single-cell stimulus-specific information, $I(s, R)$, is the amount of information the set of responses, R , has about a specific stimulus, s (see Rolls *et al* (1997) and Rolls and Milward (2000)). $I(s, R)$ is given by

$$I(s, R) = \sum_{r \in R} P(r|s) \log_2 \frac{P(r|s)}{P(r)} \quad (5)$$

where r is an individual response from the set of responses R of the neuron. For each cell the performance measure used was the maximum amount of information a cell conveyed about

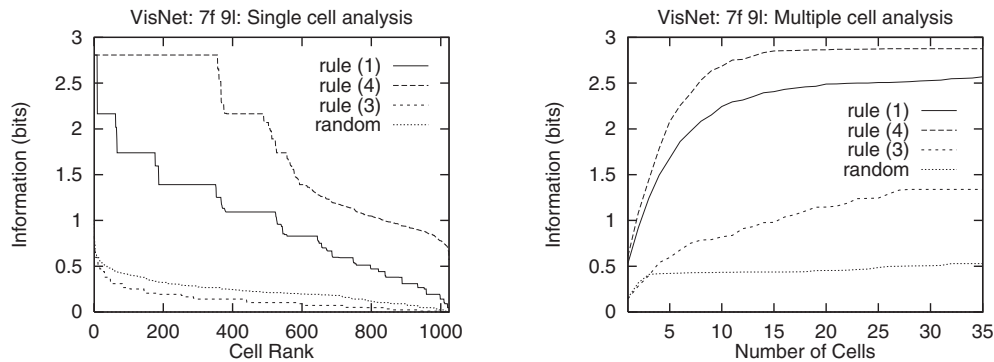


Figure 2. Numerical results with the standard trace rule (1), learning rule (4), the Hebb rule (3) and random weights, trained on seven faces in nine locations: single-cell information measure (left), multiple-cell information measure (right).

any one stimulus. This (rather than the mutual information, $I(S, R)$ where S is the whole set of stimuli s) is appropriate for a competitive network in which the cells tend to become tuned to one stimulus. ($I(s, R)$ has more recently been called the stimulus-specific surprise, see DeWeese and Meister (1999). Its average across stimuli is the mutual information $I(S, R)$.)

If all the output cells of VisNet learned to respond to the same stimulus, then the information about the set of stimuli S would be very poor, and would not reach its maximal value of \log_2 of the number of stimuli (in bits). The second measure that is used here is the information provided by a set of cells about the stimulus set, using the procedures described by Rolls *et al* (1997) and Rolls and Milward (2000). The multiple-cell information is the mutual information between the whole set of stimuli S and of responses R calculated using a decoding procedure in which the stimulus s' that gave rise to the particular firing rate response vector on each trial is estimated. (The decoding step is needed because the high dimensionality of the response space would lead to an inaccurate estimate of the information if the responses were used directly, as described by Rolls *et al* (1997) and Rolls and Treves (1998).) A probability table is then constructed of the real stimuli s and the decoded stimuli s' . From this probability table, the mutual information between the set of actual stimuli S and the decoded estimates S' is calculated as

$$I(S, S') = \sum_{s, s'} P(s, s') \log_2 \frac{P(s, s')}{P(s)P(s')}. \quad (6)$$

This was calculated for the subset of cells which had as single cells the most information about which stimulus was shown. In particular, we calculated the multiple-cell information from the first five cells for each stimulus that had maximal single-cell information about that stimulus: that is, from a population of 35 cells given that there were seven stimuli.

Baseline results of VisNet runs trained on seven faces at each of nine locations are given in figure 2. On the left are the single-cell information measures for all top-layer neurons ranked in order of their invariance, while on the right are multiple-cell information measures which rise steadily as the number of cells in the sample is increased. Given that the maximum single-cell information measure is 2.8 (i.e. $\log_2 7$) as there are seven face stimuli, we can see that no cells reach this level of performance with a Hebb rule or in the untrained condition with random weights, that a small number of cells achieve optimal performance with the standard trace rule (1), and that many cells reach optimal performance with the rule described by Rolls and Milward (2000), equation (4). The fact that many single cells reached optimal performance

with rule (4) indicates that the rule is better for training. The rule is better in the sense that single cells are more likely to show invariant representations after the training. In addition, the multiple-cell measures show that in practice the rules which produce many well performing single cells also tend to lead to populations of cells which together provide high levels of information about all the stimuli, although with the limited number of stimuli used here, the multiple-cell information measure can saturate (see Rolls and Treves (1998)).

2. Modified Hebbian trace rules, error correction and temporal difference learning

2.1. The modified Hebbian trace rule and error correction

Rule (4) corrects the weights using a post-synaptic trace obtained from the previous firing (produced by other transforms of the same stimulus), with no contribution to the trace from the current post-synaptic firing (produced by the current transform of the stimulus). Indeed, insofar as the current firing y^τ is not the same as $\bar{y}^{\tau-1}$, this difference can be thought of as an error. This leads to a conceptualization of using the difference between the current firing and the preceding trace as an error correction term, as developed next. Although trace learning has been related to error correction learning by Sutton and Barto (1981), this was in the context of studies of the temporal aspects of classical conditioning; the current conceptualization is developed in relation to invariant recognition learning.

First, we re-express rule (4) in an alternative form as follows. Suppose we are at time step τ and have just calculated a neuronal firing rate y^τ and the corresponding trace \bar{y}^τ from the trace update equation (2). If we assume $\eta \in (0, 1)$, then rearranging equation (2) gives

$$\bar{y}^{\tau-1} = \frac{1}{\eta}(\bar{y}^\tau - (1 - \eta)y^\tau) \quad (7)$$

and substituting equation (7) into (4) gives

$$\begin{aligned} \Delta w_j &= \alpha \frac{1}{\eta}(\bar{y}^\tau - (1 - \eta)y^\tau)x_j^\tau \\ &= \alpha \frac{1 - \eta}{\eta} \left(\frac{1}{1 - \eta} \bar{y}^\tau - y^\tau \right) x_j^\tau \\ &= \hat{\alpha}(\hat{\beta}\bar{y}^\tau - y^\tau)x_j^\tau \end{aligned} \quad (8)$$

where $\hat{\alpha} = \alpha \frac{1 - \eta}{\eta}$ and $\hat{\beta} = \frac{1}{1 - \eta}$. The modified Hebbian trace learning rule (4) is thus equivalent to equation (8) which is in the general form of an error correction rule (Hertz *et al* 1991). That is, rule (8) involves the subtraction of the current firing rate y^τ from a target value, in this case $\hat{\beta}\bar{y}^\tau$.

To understand the behaviour of the modified Hebbian rule (4) we need to consider the effect of scaling $\hat{\beta}$ in equation (8). To do this, we replace $\hat{\beta}$ with $\beta = k\hat{\beta}$ where $k > 0$ is some scaling coefficient, giving a new target $\beta\bar{y}^\tau$. This scaling controls the primary factor governing the performance of the learning rule, namely the resultant y^τ term in the brackets in equation (8) (which is produced by subtracting the $-y^\tau$ term from the y^τ component of the target $\beta\bar{y}^\tau$). As β is scaled up (i.e. set $k > 1$), the part of y^τ left after subtracting the $-y^\tau$ term from the target will be positive, and the term $\hat{\alpha}\beta\bar{y}^\tau x_j^\tau$ will come to dominate the term $-\hat{\alpha}y^\tau x_j^\tau$. Consequently, rule (8) will tend towards the behaviour of the standard trace rule (1), because the post-synaptic term includes as a component the current firing rate y^τ . In contrast, as β is scaled down (i.e. set $k < 1$), the resultant component of y^τ after subtracting the $-y^\tau$ term from the target is negative. Then rule (8) will function like an error correction method, continually comparing the current firing rate y^τ with the target $\beta\bar{y}^\tau$, and adapting the

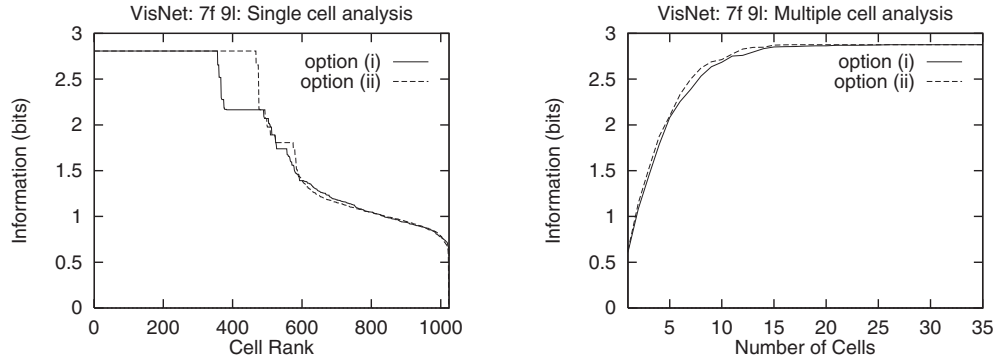


Figure 3. Numerical results with error correction learning rule (8) with two options: (i) $\hat{\beta} = \frac{1}{1-\eta} = 5.0$ and (ii) $\hat{\beta}$ replaced by $0.5 \times \hat{\beta} = 2.5$. These two rules are trained on seven faces in nine locations: single-cell information measure (left), multiple-cell information measure (right).

weights to set the current firing rate equal to a function of the cell's recent activity given by the target. In figure 3 we show numerical results for learning rule (8) with two options. In option (i), k is set to 1 so that β is equal to $\hat{\beta}$ (and is 5 because η is 0.8 and $\hat{\beta} = \frac{1}{1-\eta}$). This produces conditions equivalent to the modified Hebbian rule (4). The performance shown in figure 3 is identical to that shown for rule (4) in figure 2. In option (ii), we set k to move the operation of equation (8) into an error-correction-like region. (This was achieved simply by setting k to 0.5 with no attempt to optimize the value of k . With $k = 0.5$ there is genuine error correction since the resultant y^τ term in brackets in equation (8) becomes negative. For this value of k , there is a resultant component of $-0.5y^\tau$ after subtracting the $-y^\tau$ term from the target $2.5\bar{y}^\tau$.) It can be seen in figure 3 that option (ii) offers significantly better performance with the single-cell information measure than option (i), thus demonstrating the effectiveness of the error correction paradigm described above.

The above analysis shows that the trace-modified Hebbian rule (4) sits in a continuum of learning rules ranging from reasonably performing trace-modified Hebbian rules at one end, to a superior error correction regime at the other. The primary factor governing the performance of a rule within this continuum is the sign and relative magnitude of the resultant component of y^τ in equation (8) with $\hat{\beta}$ replaced by β . When β is scaled up there is a positive resultant component of y^τ , and then rule (8) is Hebbian-like and associates the current cell firing rate y^τ with the current cell inputs x_j^τ . When β is scaled down there is a negative resultant component of y^τ , and then rule (8) is a form of error correction. However, for rule (4) the resultant component of y^τ in equation (8) with $\beta = \hat{\beta}$ is zero, and this stops the current cell firing y^τ from being part of what is associated with the current cell inputs x_j^τ . In other words, the purely Hebbian, non-trace-related, component of the learning is removed. The Hebbian component normally would have the deleterious effect of encouraging neurons to learn position-dependent responses (because there is no trace to link different locations, see Wallis and Rolls (1997)), and thus with $\beta = \hat{\beta}$, the deleterious component is not present. Although above we have referred to rule (4) as a modified Hebbian rule, in fact it is only associative in the sense of associating *previous* cell firing with the current cell inputs. When considering the role of the current firing rate y^τ in learning rule performance, we note that rule (4) in fact lies on the boundary of the two regimes, outperforming the standard trace rule (1) with a positive resultant component of y^τ , but performing less well than error correction rules with a negative resultant component of y^τ . In the next section we continue to explore the error correction paradigm, examining five alternative examples of this sort of learning rule.

2.2. Five forms of error correction learning rule

Error correction learning rules are derived from gradient descent minimization (Hertz *et al* 1991), and continually compare the current neuronal output to a target value t and adjust the synaptic weights according to the following equation at a particular time step τ :

$$\Delta w_j = \alpha(t - y^\tau)x_j^\tau. \quad (9)$$

In this usual form of gradient descent by error correction, the target t is fixed. However, in keeping with our aim of encouraging neurons to respond similarly to images that occur close together in time it seems reasonable to set the target at a particular time step, t^τ , to be some function of cell activity occurring close in time, because encouraging neurons to respond to temporal classes will tend to make them respond to the different variants of a given stimulus (Földiák 1991, Rolls 1992, Wallis and Rolls 1997). For this reason, in this paper we explore a range of error correction rules where the targets t^τ are based on the trace of neuronal activity calculated according to equation (2). We note that although the target is not a fixed value as in standard error correction learning, nevertheless the learning rules described here perform gradient descent on each time step, as elaborated in the discussion. Although the target may be varying early on in learning, as learning proceeds the target is expected to become more and more constant, as neurons settle to respond invariantly to particular stimuli. The first set of five error correction rules we shall discuss are as follows:

$$\Delta w_j = \alpha(\beta\bar{y}^{\tau-1} - y^\tau)x_j^\tau, \quad (10)$$

$$\Delta w_j = \alpha(\beta y^{\tau-1} - y^\tau)x_j^\tau, \quad (11)$$

$$\Delta w_j = \alpha(\beta\bar{y}^\tau - y^\tau)x_j^\tau, \quad (12)$$

$$\Delta w_j = \alpha(\beta\bar{y}^{\tau+1} - y^\tau)x_j^\tau, \quad (13)$$

$$\Delta w_j = \alpha(\beta y^{\tau+1} - y^\tau)x_j^\tau \quad (14)$$

where updates (10)–(12) are performed at time step τ , and updates (13) and (14) are performed at time step $\tau + 1$. (The reason for adopting this convention is that the basic form of the error correction rule (9) is kept, with the five different rules simply replacing the term t .) It may be readily seen that equations (11) and (14) are special cases of equations (10) and (13) respectively, with $\eta = 0$.

These rules are all similar except for their targets t^τ , which are all functions of a temporally nearby value of cell activity. In particular, rule (12) is directly related to rule (8), but is more general in that the parameter $\beta = \frac{1}{1-\eta}$ is replaced by an unconstrained parameter β . In addition, we also note that rule (10) is closely related to a rule developed in Peng *et al* (1998) for view invariance learning. The above five error correction rules are biologically plausible in that the targets t^τ are all local cell variables (Rolls and Treves 1998). In particular, rule (12) uses the trace \bar{y}^τ from the current time level τ , and rules (11) and (14) do not need exponential trace values \bar{y} , instead relying only on the instantaneous firing rates at the current and immediately preceding time steps. However, all five error correction rules involve decrementing of synaptic weights according to an error which is calculated by subtracting the current activity from a target.

Numerical results with the error correction rules trained on seven faces in nine locations are presented in figure 4. For all the results shown in this paper the synaptic weights were clipped to be positive during the simulation, because it is important to test that decrementing synaptic weights purely within the positive interval $w \in [0, \infty)$ will provide significantly enhanced performance. That is, it is important to show that error correction rules do not necessarily require possibly biologically implausible modifiable negative weights. For each of the rules (10)–(14), the parameter β has been individually optimized to the following respective

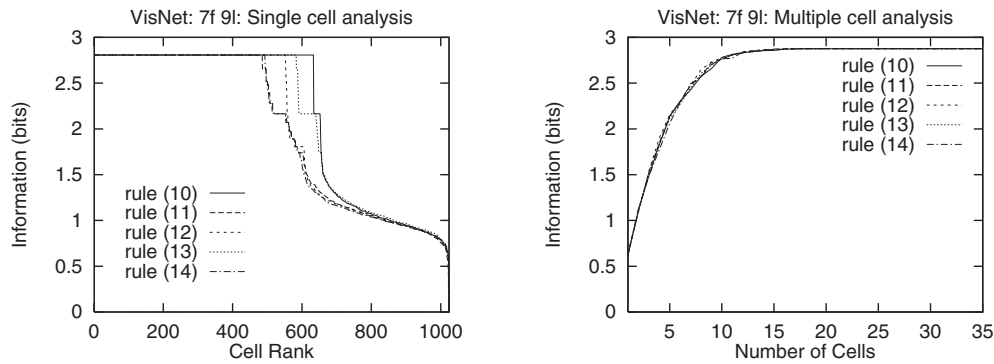


Figure 4. Numerical results with the five error correction rules (10)–(14), (with positive clipping of synaptic weights) trained on seven faces in nine locations. On the left are single-cell information measures, and on the right are multiple-cell information measures.

values: 4.9, 2.2, 2.2, 3.8, 2.2. On the left and right are results with the single- and multiple-cell information measures, respectively. Comparing figure 4 with figure 2 shows that all five error correction rules offer considerably improved performance over both the standard trace rule (1) and rule (4). From the left-hand side of figure 4 it can be seen that rule (10) performs best, and this is probably due to two reasons. Firstly, rule (10) incorporates an exponential trace $\bar{y}^{\tau-1}$ in its target t^τ , and we would expect this to help neurons to learn more quickly to respond invariantly to a class of inputs that occur close together in time. Hence, setting $\eta = 0$ as in rule (11) results in reduced performance. Secondly, unlike rules (12) and (13), rule (10) does not contain any component of y^τ in its target. If we examine rules (12) and (13), we see that their respective targets $\beta\bar{y}^\tau$, $\beta\bar{y}^{\tau+1}$ contain significant components of y^τ . Indeed, comparing rules (12) and (13) shows that the target of rule (12) contains the largest contribution from y^τ , and this rule performs least well out of the two. Results with these five error correction rules without positive clipping of weights showed that there were only minor differences in performance.

In the above simulations one might have expected the optimal β value to be equal to 1 for each of the five error correction rules; this would imply that the error correction rules were attempting to adapt the weights such that $y^\tau = t^\tau$ at convergence. However, in practice performance is poor for $\beta = 1$, with the best results obtained with values of β greater than 1. This is due to effects related to the transition between two stimuli, as shown by the following. If the trace is reset between stimuli (the procedure used in this paper), then the magnitude of the trace for the first few presentations of the next stimulus is too low. Setting β greater than 1 compensates for this and therefore produces better performance. (The issue of trace reset between stimuli should not be thought of as a major issue in VisNet, in that an exponentially decaying trace minimizes interfering effects from previous stimuli (Wallis and Rolls 1997, Wallis and Baddeley 1997), and in that a trace might be reset biologically between stimuli due to a large saccade to a different stimulus (Wallis and Rolls 1997).)

2.3. Relationship to temporal difference learning, and five forms of temporal-difference-inspired learning rules

So far we have discussed the relationship of rule (4) to error correction, and we now consider how the error correction rules shown in equations (10)–(14) are related to temporal difference learning (Sutton 1988, Sutton and Barto 1998).

Sutton (1988) described temporal difference methods in the context of prediction learning. These methods are a class of incremental learning techniques that can learn to predict final outcomes through comparison of successive predictions from the preceding time steps. This is in contrast to traditional supervised learning, which involves the comparison of predictions only with the final outcome. Consider a series of multi-step prediction problems in which for each problem there is a sequence of observation vectors, $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m$, at successive time steps, followed by a final scalar outcome z . For each sequence of observations temporal difference methods form a sequence of predictions y^1, y^2, \dots, y^m , each of which is a prediction of z . These predictions are based on the observation vectors \mathbf{x}^τ and a vector of modifiable weights \mathbf{w} ; i.e. the prediction at time step τ is given by $y^\tau(\mathbf{x}^\tau, \mathbf{w})$, and for a linear dependency the prediction is given by $y^\tau = \mathbf{w}^T \mathbf{x}^\tau$. The problem of prediction is to calculate the weight vector \mathbf{w} such that the predictions y^τ are good estimates of the outcome z .

The supervised learning approach to the prediction problem is to form pairs of observation vectors \mathbf{x}^τ and outcome z for all time steps, and compute an update to the weights according to the gradient descent equation

$$\Delta \mathbf{w} = \alpha(z - y^\tau) \nabla_{\mathbf{w}} y^\tau \quad (15)$$

where α is a learning rate parameter. However, this learning procedure requires all calculation to be done at the end of the sequence, once z is known. To remedy this, it is possible to replace method (15) with a temporal difference algorithm that is mathematically equivalent but allows the computational workload to be spread out over the entire sequence of observations. Temporal difference methods are a particular approach to updating the weights based on the values of successive predictions, $y^\tau, y^{\tau+1}$. Sutton (1988) showed that the following temporal difference algorithm is equivalent to method (15):

$$\Delta \mathbf{w} = \alpha(y^{\tau+1} - y^\tau) \sum_{k=1}^{\tau} \nabla_{\mathbf{w}} y^k \quad (16)$$

where $y^{m+1} \equiv z$. However, unlike method (15) this can be computed incrementally at each successive time step since each update depends only on $y^{\tau+1}, y^\tau$ and the sum of $\nabla_{\mathbf{w}} y^k$ over previous time steps k . The next step taken in Sutton (1988) is to generalize equation (16) to the following final form of temporal difference algorithm, known as ‘TD(λ)’:

$$\Delta \mathbf{w} = \alpha(y^{\tau+1} - y^\tau) \sum_{k=1}^{\tau} \lambda^{\tau-k} \nabla_{\mathbf{w}} y^k \quad (17)$$

where $\lambda \in [0, 1]$ is an adjustable parameter that controls the weighting on the vectors $\nabla_{\mathbf{w}} y^k$. Equation (17) represents a much broader class of learning rules than the more usual gradient-descent-based rule (16), which is in fact the special case TD(1).

A further special case of equation (17) is for $\lambda = 0$, i.e. TD(0), as follows:

$$\Delta \mathbf{w} = \alpha(y^{\tau+1} - y^\tau) \nabla_{\mathbf{w}} y^\tau. \quad (18)$$

But for problems where y^τ is a linear function of \mathbf{x}^τ and \mathbf{w} , we have $\nabla_{\mathbf{w}} y^\tau = \mathbf{x}^\tau$, and so equation (18) becomes

$$\Delta \mathbf{w} = \alpha(y^{\tau+1} - y^\tau) \mathbf{x}^\tau. \quad (19)$$

If we assume that the prediction process is being performed by a neuron with a vector of inputs \mathbf{x}^τ , synaptic weight vector \mathbf{w} and output $y^\tau = \mathbf{w}^T \mathbf{x}^\tau$, then we see that the TD(0) algorithm (19) is identical to the error correction rule (14) with $\beta = 1$. In understanding this comparison with temporal difference learning, it may be useful to note that the firing at the end of a sequence of the transformed exemplars of a stimulus is effectively the temporal difference target z . This

establishes a link to temporal difference learning. Further, we note that from learning epoch to learning epoch, the target z for a given neuron will gradually settle down to be more and more fixed as learning proceeds.

We now explore in more detail the relation between the error correction rules described above and temporal difference learning. For each sequence of observations with a single outcome the temporal difference method (19), when viewed as an error correction rule, is attempting to adapt the weights such that $y^{\tau+1} = y^\tau$ for all successive pairs of time steps—the same general idea underlying the error correction rules (10)–(14). Furthermore, in Sutton and Barto (1998), where temporal difference methods are applied to reinforcement learning, the TD(λ) approach is again further generalized by replacing the target $y^{\tau+1}$ by any weighted average of predictions y from arbitrary future time steps (e.g., $t^\tau = \frac{1}{2}y^{\tau+3} + \frac{1}{2}y^{\tau+7}$), including an exponentially weighted average extending forward in time. So a more general form of the temporal difference algorithm has the form

$$\Delta \mathbf{w} = \alpha(t^\tau - y^\tau)\mathbf{x}^\tau, \quad (20)$$

where here the target t^τ is an arbitrary weighted average of the predictions y over future time steps. Of course, with standard temporal difference methods the target t^τ is always an average over *future* time steps $k = \tau + 1, \tau + 2$, etc. But in the five error correction rules this is only true for the last exemplar (14). This is because with the problem of prediction, for example, the ultimate target of the predictions y^1, \dots, y^m is a final outcome $y^{m+1} \equiv z$. However, this restriction does not apply to our particular application of neurons trained to respond to temporal classes of inputs within VisNet. Here we only wish to set the firing rates y^1, \dots, y^m to the same value, not some final given value z . However, the more general error correction rules clearly have a close relationship to standard temporal difference algorithms. For example, it can be seen that equation (11) with $\beta = 1$ is in some sense a temporal mirror image of equation (19), particularly if the updates Δw_j are added to the weights w_j only at the end of a sequence. That is, rule (11) will attempt to set y^1, \dots, y^m to an *initial* value $y^0 \equiv z$. This relationship to temporal difference algorithms allows us to begin to exploit established temporal difference analyses to investigate the convergence properties of the error correction methods, and this is discussed in section 2.5.

Although our main aim in relating error correction rules to temporal difference learning is to begin to exploit established temporal difference analyses, we note in the following that the most general form of temporal difference learning, TD(λ), in fact suggests an interesting generalization to the existing error correction learning rules for which we currently have $\lambda = 0$. Assuming $y^\tau = \mathbf{w}^T \mathbf{x}^\tau$ and $\nabla_{\mathbf{w}} y^\tau = \mathbf{x}^\tau$, the general equation (17) for TD(λ) becomes

$$\Delta \mathbf{w} = \alpha(y^{\tau+1} - y^\tau) \sum_{k=1}^{\tau} \lambda^{\tau-k} \mathbf{x}^k \quad (21)$$

where the term $\sum_{k=1}^{\tau} \lambda^{\tau-k} \mathbf{x}^k$ is a weighted sum of the vectors \mathbf{x}^k . This suggests generalizing the original five error correction rules (10)–(14) by replacing the term x_j^τ by a weighted sum $\hat{x}_j^\tau = \sum_{k=1}^{\tau} \lambda^{\tau-k} x_j^k$ with $\lambda \in [0, 1]$. In Sutton (1988) \hat{x}_j^τ is calculated according to

$$\hat{x}_j^\tau = x_j^\tau + \lambda \hat{x}_j^{\tau-1} \quad (22)$$

with $\hat{x}_j^0 \equiv 0$. This gives the following five temporal-difference-inspired error correction rules:

$$\Delta w_j = \alpha(\beta \bar{y}^{\tau-1} - y^\tau) \hat{x}_j^\tau, \quad (23)$$

$$\Delta w_j = \alpha(\beta y^{\tau-1} - y^\tau) \hat{x}_j^\tau, \quad (24)$$

$$\Delta w_j = \alpha(\beta \bar{y}^\tau - y^\tau) \hat{x}_j^\tau, \quad (25)$$

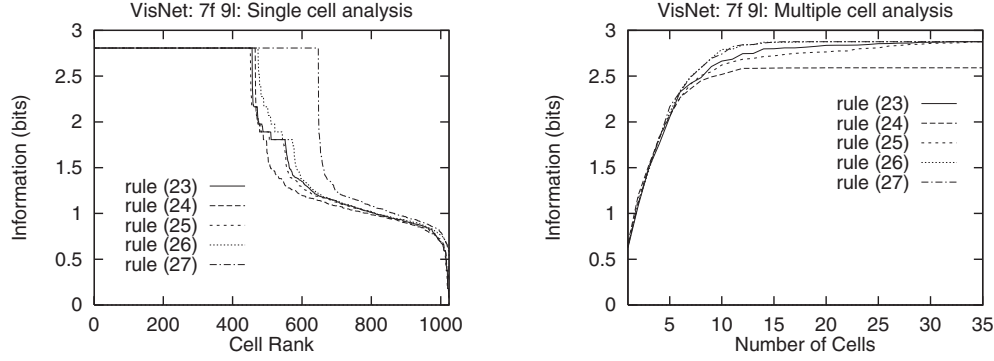


Figure 5. Numerical results with the five temporal-difference-inspired error correction rules (23)–(27), and \hat{x}_j^τ calculated according to equation (22) (with positive clipping of synaptic weights) trained on seven faces in nine locations. On the left are single-cell information measures, and on the right are multiple-cell information measures.

$$\Delta w_j = \alpha(\beta \bar{y}^{\tau+1} - y^\tau) \hat{x}_j^\tau, \quad (26)$$

$$\Delta w_j = \alpha(\beta y^{\tau+1} - y^\tau) \hat{x}_j^\tau \quad (27)$$

where it may be readily seen that equations (24) and (27) are special cases of (23) and (26) respectively, with $\eta = 0$. As with the trace \bar{y}^τ , the term \hat{x}_j^τ is reset to zero when a new stimulus is presented. These five rules can be related to the more general TD(λ) algorithm, but continue to be biologically plausible using only local cell variables. Setting $\lambda = 0$ in rules (23)–(27), gives us back the original error correction rules (10)–(14), which may now be related to TD(0).

Numerical results with error correction rules (23)–(27), and \hat{x}_j^τ calculated according to equation (22) with $\lambda = 1$, with positive clipping of weights, trained on seven faces in nine locations are presented in figure 5. For each of the rules (23)–(27), the parameter β has been individually optimized to the following respective values: 1.7, 1.8, 1.5, 1.6, 1.8. On the left and right are results with the single- and multiple-cell information measures, respectively. Comparing these five temporal-difference-inspired rules, it can be seen that the best performance is obtained with rule (27) where many more cells reach the maximum level of performance possible with respect to the single-cell information measure. In fact, this rule offered the best such results in this paper. This may well be due to the fact that this rule may be directly compared to the standard TD(1) learning rule, which itself may be related to classical supervised learning for which there are well known optimality results. This is fully discussed in section 2.5. We note here that the use of \hat{x} implies a presynaptic temporal trace, which is biologically plausible (though λ as high as 1 might be considered less plausible because it implies no decay of the presynaptic trace). We have shown in this paper that invariant learning in neural networks can be produced with values of λ as extreme as 0 and 1, within which the biologically plausible values will lie. In additional simulations we confirmed similar invariant learning with intermediate values of λ . A surprising result shown in figure 5 is the relatively poor performance of rule (24) for the multiple-cell information measure, which asymptotes to a sub-optimal value. It is not clear why this rule has behaved in this way. Overall, the temporal-difference-related learning rules with a presynaptic trace (23)–(27) produce good performance which is comparable to that produced with error correction rules.

Sutton (1988) discusses the possibility of using different weightings for the vectors $\nabla_w y^k = \mathbf{x}^k$. Therefore, an alternative approach would be to replace \hat{x}_j^τ in equations (23)–(27)

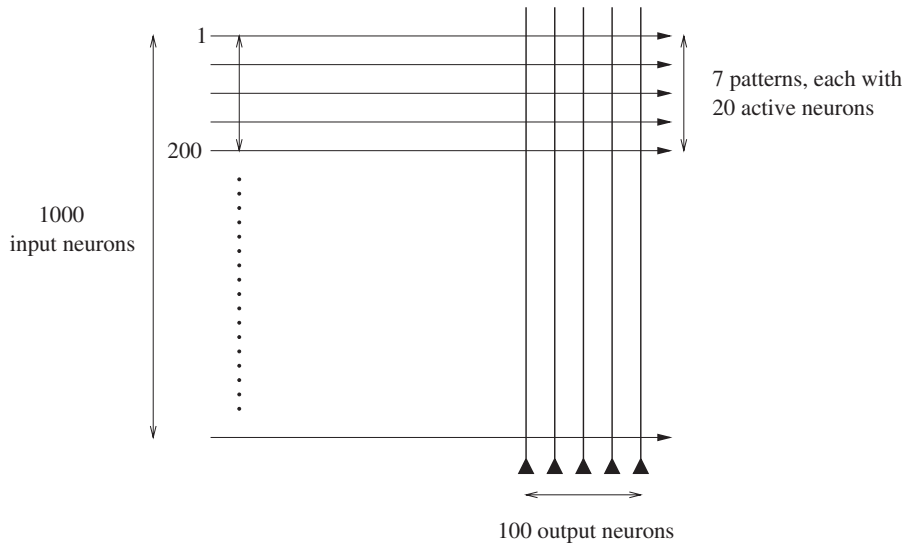


Figure 6. The architecture of the one-layer network. The shift-invariant problem consisted of learning to produce the same output firing for each of the shifted versions of the seven training patterns. Each test pattern was generated by setting at random 20 of the first 200 input firing rates to 1. Each of the nine shifted versions of each training pattern were produced by shifting the training pattern by 100 units across the 1000 input neurons. There were 100 output neurons.

with the traced average, \bar{x}_j^t , calculated according to equation (2). Although not presented here, the results we obtained with such rules also showed good performance.

2.4. Speed of convergence

It may be expected that the different learning rules described in this paper produce different convergence histories, with both the optimal performance achieved and the rate at which that optimal performance is achieved differing between the different learning rules. In the runs with VisNet, a relatively low learning rate which tended to produce good performance with the different rules was used, and the network was allowed to learn for a standard number of epochs for each layer. To explore in more detail the performance of the different rules, and especially their speed of convergence, investigations were performed with a smaller, one-layer, network to enable the performance to be investigated without the large simulation overhead of VisNet. (The simulations with VisNet allow the operation of these rules to be studied in this much fuller model of several stages of operation of the visual system, but these are much larger and more computationally intensive simulations. For the present section, it was of interest to restrict the scale of the simulations to one-layer nets, to allow intensive exploration of the particular issue of the speed of convergence of the different learning rules described and in some cases introduced in this paper.)

The one-layer net consisted of 100 output neurons with firing rates y_i each receiving 1000 inputs with firing rates x_j . The seven input stimuli each consisted of binary firing rate patterns of 20 active neurons, chosen randomly from the first 200 neurons in the input array. Translation invariance was studied by choosing one of the seven input stimuli, and presenting it successively during learning in nine locations, each location translated across a distance corresponding to 100 input units (see figure 6). At each location, the activation h of a neuron was computed by the inner product of the input firing vector and the synaptic weight vector

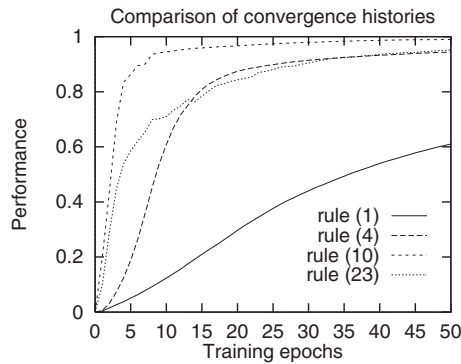


Figure 7. Convergence histories showing the performance of a simplified one-layer object recognition network with the following four learning rules: standard trace rule (1), learning rule (4), error correction rule (10), and temporal-difference-inspired rule (23).

as follows:

$$h = \sum w_j x_j. \quad (28)$$

Then the output firing was produced by a linear threshold activation function that ensured that the sparseness of the output firing of the population of 100 neurons had a sparseness (see above) of 0.1 (Rolls and Treves 1998). Then learning took place according to the learning rule being tested. After the stimulus had been presented this way in all nine locations, the trace was then reset, and another stimulus was chosen, and presented successively during learning in nine locations. The training algorithm was thus very similar to that used by VisNet, and the aim of the learning was to produce output neurons that could be activated by any of the translated versions of one of the stimuli, and by none of the other stimuli.

In these experiments with a one-layer competitive network we implemented a correlation measure, which shows the similarity of the output firing pattern of the 100 neurons to all the translated versions of a given stimulus, relative to the output patterns produced by all the other stimuli in all locations. In particular, we measured the average cosine of the angle between the output firing rate pattern vector to the translated versions of a particular stimulus, and subtracted from this the average cosine of the angle between the output pattern and all output patterns to other stimuli. Given that the cosine of the angle between aligned vectors is 1, and between orthogonal vectors is zero, the overall translation invariance performance measure will have a maximal value of 1, and a minimal value clipped at zero. Results with this performance measure are presented in figure 7, which shows the convergence histories of the performance measure for the one-layer network with the following four learning rules: standard trace rule (1), learning rule (4), error correction rule (10) and temporal-difference-inspired rule (23). It can be seen that the error correction rule (10) converges much more rapidly than the other rules, and by the end of 50 epochs has achieved almost perfect performance. However, the results shown are for a comparatively low value of the learning rate parameter used in the above equations of 0.01 which was found to produce smooth convergence with all the learning rules. If the learning rate is increased to too high a value, then oscillatory performance may result. The conclusion of the investigations with the one-layer net is thus that the learning rules do differ with respect to the efficiency with which convergence occurs. The error correction rule operates relatively quickly and effectively, the temporal difference and the asymmetric trace rule (4) both operate intermediately, and the standard trace rule (1) operates more slowly.

2.5. Convergence analyses

In section 2.3 it was shown that the error correction rules discussed in section 2.2 can be related to temporal difference methods. This now allows us to begin to apply existing convergence analyses from temporal difference theory (Sutton 1988, Sutton and Barto 1998) and supervised learning (Widrow and Stearns 1985). In this paper we focus on two results from Sutton (1988) relating to perhaps two of the most simple rules, which are examples of TD(1) and TD(0) respectively. These analyses shed light on the basic operation of these two types of learning rule, helping to explain their relative efficacies. However, the results we discuss below apply to the rather idealized situation of sequences of input vectors, $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m$ presented to an isolated neuron with firing rates $y^\tau = \mathbf{w}^T \mathbf{x}^\tau$ for $\tau = 1, \dots, m$, and y^{m+1} kept fixed to some constant value z . This is the basic temporal difference setup for the prediction problem described in Sutton (1988) and section 2.3. Of course, these assumptions are not fully met in VisNet, where the individual firing rates of neurons are affected by other neurons through lateral inhibition. In addition, in VisNet y^{m+1} will depend on the synaptic weights and the input vector \mathbf{x}^{m+1} , which in turn will depend on the presentation order. However, the aim of the following analyses is to help reveal how error correction rules cause individual neurons to learn to respond similarly to classes of inputs that occur close together in time. The underlying processes behind this are in fact complemented by these additional model details: lateral inhibition simply helps to select which neurons respond to a particular stimulus, and the re-ordering of views should act to reduce the possibility of particular views dominating the learning process just by virtue of their place in the sequence. These two effects then blend with the fundamental operation of error correction rules which adapt synaptic weights to encourage neurons to respond similarly to inputs that occur close together in time. It is this aspect of the error correction learning rules that the following analyses help to illuminate.

Firstly, we consider the TD(1) rule derived from setting $\lambda = 1$ and $\beta = 1$ in equation (27):

$$\Delta w_j = \alpha(y^{\tau+1} - y^\tau) \sum_{k=1}^{\tau} x_j^k. \quad (29)$$

In Sutton (1988) it is shown that if all synaptic weight updates are carried out at the end of a sequence, then rule (29) is in fact equivalent to

$$\Delta w_j = \alpha(z - y^\tau)x_j^\tau \quad (30)$$

which is classical supervised learning where y^τ is compared directly with $y^{m+1} \equiv z$. Widrow and Stearns (1985) provide an account of the theory of this rule. In particular, when continually presented with new data series, rule (30) converges to the true expected value $E\{z|x\}$ for each input vector x . In contrast, under repeated presentations of a finite data set, rule (30) minimizes the root mean square error between the predictions y and the actual outcomes z in the training set.

Next we consider the TD(0) rule derived from setting $\lambda = 0$ and $\beta = 1$ in equation (27), which is equivalent to equation (14) with $\beta = 1$, and may be written as

$$\Delta w_j = \alpha(y^{\tau+1} - y^\tau)x_j^\tau. \quad (31)$$

In Sutton (1988) it is shown that when continually presented with new data series, rule (31) performs similarly to rule (30), converging to the true expected value $E\{z|x\}$ for each input vector x . However, under repeated presentations of a finite data set, rule (31) converges to what can be considered the optimal estimates for coping with future experience—those consistent with the maximum likelihood estimate of an assumed underlying Markov process. So, under repeated presentations of a finite set of input sequences, the TD(1) and TD(0) rules, (29) and (31) respectively, perform different types of optimization, and it is this difference that may

be responsible for the apparent disparity in the performance of these two rules within VisNet, as discussed below.

From our earlier simulations it appears that the form of optimization described above associated with TD(1) rather than TD(0) leads to better performance within VisNet. Comparing figures 4 and 5 shows that the TD(1)-like rule (27) with $\lambda = 1.0$ and $\beta = 1.8$ gives considerably superior results to the TD(0)-like rule (14) with $\beta = 2.2$. In fact, the former of these two rules provided the best single-cell information results presented in this paper. We hypothesize that these results are related to the fact that only a finite set of image sequences is presented to VisNet, and so the type of optimization performed by TD(1) for repeated presentations of a finite data set is more appropriate for this problem than the form of optimization performed by TD(0).

In this section we have exploited some analyses from temporal difference methods and supervised learning to provide convergence results for two error correction rules (29) and (31) that are examples of TD(1) and TD(0), respectively. These initial results have begun to provide a theoretical framework to better understand the operation of the learning rules discussed in this paper, and may now open the way for the transfer of further convergence results from the temporal difference literature.

3. Discussion

In this paper we have established a link between trace learning rules and both error correction and temporal difference learning, and have discussed the relationships above. In terms of biological plausibility, we note the following. First, all the learning rules described in this paper are local learning rules, and in this sense are biologically plausible (see Rolls and Treves (1998)). (The rules are local in that the terms used to modify the synaptic weights are potentially available in the pre- and post-synaptic elements.)

Second, we note that all the rules do require some evidence of the activity on one or more previous stimulus presentations to be available when the synaptic weights are updated. Some of the rules (for example, learning rule (12)) use the trace \bar{y}^τ from the current time level, while rules (11) and (14) do not need to use an exponential trace of the neuronal firing rate, but only the instantaneous firing rates y at two successive time steps. It is known that synaptic plasticity does involve a combination of separate processes each with potentially differing time courses (Koch 1999), and these different processes could contribute to trace rule learning. Another mechanism suggested for implementing a trace of previous neuronal activity is the continuing firing for often 300 ms produced by a short (16 ms) presentation of a visual stimulus (Rolls and Tovee 1994) which is suggested to be implemented by local cortical recurrent attractor networks (Rolls and Treves 1998).

Third, we note that in utilizing the trace in the targets t^τ , the error correction (or temporal-difference-inspired) rules perform a comparison of the instantaneous firing y^τ with a temporally nearby value of the activity, and this comparison involves a subtraction. The subtraction provides an error, which is then used to increase or decrease the synaptic weights. This is a somewhat different operation from long-term depression as well as long-term potentiation, which are *associative* changes which depend on the pre- and post-synaptic activity. However, it is interesting to note that an error correction rule which appears to involve a subtraction of current firing from a target might be implemented by a combination of an associative process operating with the trace, and an anti-Hebbian process operating to remove the effects of the current firing. For example, the synaptic updates $\Delta w_j = \alpha(t^\tau - y^\tau)x_j^\tau$ can be decomposed into two separate associative processes, $\alpha t^\tau x_j^\tau$ and $-\alpha y^\tau x_j^\tau$, that may occur independently. (The target, t^τ , could in this case be just the trace of previous neural activity from the preceding

trials, excluding any contribution from the current firing.) Another way to implement an error correction rule using associative synaptic modification would be to force the post-synaptic neuron to respond to the error term. Although this has been postulated to be an effect which could be implemented by the climbing fibre system in the cerebellum (Ito 1984, 1989, Rolls and Treves 1998), there is no similar system known for the neocortex, and it is not clear how this particular implementation of error correction might operate in the neocortex.

In section 2.2 we describe five learning rules as error correction rules. We now discuss an interesting difference of these error correction rules from error correction rules as conventionally applied. It is usual to derive the general form of error correction learning rule from gradient descent minimization in the following way (Hertz *et al* 1991). Consider the idealized situation of a single neuron with a number of inputs x_j and output $y = \sum_j w_j x_j$, where w_j are the synaptic weights. We assume there are a number of input patterns and that for the k th input pattern, $\mathbf{x}^k = [x_1^k, x_2^k, \dots]^T$, the output y^k has a target value t^k . Hence an error measure or cost function can be defined as

$$e(\mathbf{w}) = \frac{1}{2} \sum_k (t^k - y^k)^2 = \frac{1}{2} \sum_k \left(t^k - \sum_j w_j x_j^k \right)^2. \quad (32)$$

This cost function is a function of the input patterns \mathbf{x}^k and the synaptic weight vector $\mathbf{w} = [w_1, w_2, \dots]^T$. With a fixed set of input patterns, we can reduce the error measure by employing a gradient descent algorithm to calculate an improved set of synaptic weights. Gradient descent achieves this by moving downhill on the error surface defined in \mathbf{w} space using the update

$$\Delta w_j = -\alpha \frac{\partial e}{\partial w_j} = \alpha \sum_k (t^k - y^k) x_j^k. \quad (33)$$

If we update the weights after each pattern k , then the update takes the form of an error correction rule

$$\Delta w_j = \alpha (t^k - y^k) x_j^k \quad (34)$$

which is also commonly referred to as the delta rule or Widrow–Hoff rule (see Widrow and Hoff (1960) and Widrow and Stearns (1985)). Error correction rules continually compare the neuronal output with its pre-specified target value and adjust the synaptic weights accordingly. In contrast, the way we have introduced in this paper of utilizing error correction is to specify the target as the activity trace based on the firing rate at nearby time steps. Now the actual firing at those nearby time steps is not a pre-determined fixed target, but instead depends on how the network has actually evolved. This effectively means the cost function $e(\mathbf{w})$ that is being minimized changes from time step to time step. Nevertheless, the concept of calculating an error, and using the magnitude and direction of the error to update the synaptic weights, is the similarity we wish to draw out to gradient descent learning. Further investigations of this concept may be of interest in the future. For example, with each update taking one gradient descent step in a direction that helps to minimize the current cost function, it might be the case that there is an overall cost function that is minimized. The relatively good performance of the error correction rules investigated in this paper would suggest that this possibility deserves further investigation. In a similar way, the temporal difference learning rules introduced in this paper do not use fixed final outcomes z that can be pre-specified, but instead the traditional temporal difference concept is extended to include a temporally varying outcome $y^{m+1} \equiv z$.

The simulations were run with the memory trace \bar{y}^τ being reset between stimuli. The issue of trace reset between stimuli should not be thought of as a major issue in VisNet, in that an exponentially decaying trace minimizes interfering effects from previous stimuli (Wallis and

Rolls 1997, Wallis and Baddeley 1997), and in that a trace might be reset biologically between stimuli due to a large saccade to a different stimulus (Wallis and Rolls 1997). However, we did run additional simulations to investigate whether trace reset had effects that were more prominent with the rules investigated in this paper. First, we found that for rule (4) not resetting the trace between stimuli produced very similar results to those obtained with trace reset. Second, we found that for error correction rule (10) not resetting the trace between stimuli produced somewhat less good performance than with trace reset. This effect may be related to the fact that as the error correction rule is more powerful, then large changes to the weights are made on the first one or two presentations of a new stimulus which are in the direction of the stimulus that was being shown previously. The changes in the synaptic weights on the first few trials of a new stimulus thus tended with the error correction rule to strongly encourage neurons to respond as they had to the previous stimulus. We also showed that this effect can be reduced by, for example, sweeping through the locations more than once whenever a stimulus is selected. This minimizes between-stimulus interference effects, as the same stimulus is presented successively more times. In summary, the lack of a trace reset may introduce some performance cost with the more powerful learning rules such as the error correction rules presented here, but these effects can be minimized by using longer presentations of each stimulus during training. Resetting of the trace by, for example, large saccades to a new stimulus (Wallis and Rolls 1997) might be especially beneficial in a system that used the more powerful error correction learning rules described here.

To conclude, we see then the error correction and temporal difference rules explored in this paper as providing interesting approaches to help understand invariant pattern recognition learning. Although we do not know whether the full power of these rules is expressed in the brain, we have provided suggestions about how they might be implemented. At the same time, we note that the original trace rule used by Földiák (1991), Rolls (1992) and Wallis and Rolls (1997) is a simple associative rule, is therefore biologically very plausible, and while not as powerful as many of the other rules introduced here, can nevertheless solve the same class of problem. We would also like to emphasize that although we have demonstrated how a number of error correction and temporal difference rules might play a role in the context of view-invariant object recognition, they may also operate elsewhere where it is important for neurons to learn to respond similarly to temporal classes of inputs that tend to occur close together in time.

Acknowledgments

This research was supported by the Medical Research Council, grant PG9826105, by the MRC Interdisciplinary Research Centre for Cognitive Neuroscience, and by the Wellcome Trust.

References

- Booth M C A and Rolls E T 1998 View invariant representations of familiar objects by neurons in the inferior temporal visual cortex *Cereb. Cortex* **8** 510–23
- Desimone R 1991 Face-selective cells in the temporal cortex of monkeys *J. Cognitive Neurosci.* **3** 1–8
- DeWeese M R and Meister M 1999 How to measure the information gained from one symbol *Network* **10** 325–40
- Földiák P 1991 Learning invariance from transformation sequences *Neural Comput.* **3** 194–200
- Hertz J, Krogh A and Palmer R G 1991 *Introduction to the Theory of Neural Computation* (Wokingham: Addison-Wesley)
- Ito M 1984 *The Cerebellum and Neural Control* (New York: Raven)
- 1989 Long-term depression *Ann. Rev. Neurosci.* **12** 85–102
- Koch C 1999 *Biophysics of Computation* (Oxford: Oxford University Press)

- Mitchison G 1991 Removing time variation with the anti-Hebbian differential synapse *Neural Comput.* **3** 312–20
- Peng H C, Sha L F, Gan Q and Wei Y 1998 Energy function for learning invariance in multilayer perceptron *Electron. Lett.* **34** 292–4
- Rolls E T 1992 Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical areas *Phil. Trans. R. Soc. B* **335** 11–21
- 1994 Brain mechanisms for invariant visual recognition and learning *Behav. Process.* **33** 113–38
- 1995 Learning mechanisms in the temporal lobe visual cortex *Behav. Brain Res.* **66** 177–85
- 2000 Functions of the primate temporal lobe cortical visual areas in invariant visual object and face recognition *Neuron* **27** 205–18
- Rolls E T and Milward T 2000 A model of invariant object recognition in the visual system: learning rules, activation functions, lateral inhibition and information-based performance measures *Neural Comput.* **12** 2547–72
- Rolls E T and Tovee M 1994 Processing speed in the cerebral cortex, and the neurophysiology of visual masking *Proc. R. Soc. B* **257** 9–15
- Rolls E T and Tovee M J 1995 Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex *J. Neurophysiol.* **73** 713–26
- Rolls E T and Treves A 1998 *Neural Networks and Brain Function* (Oxford: Oxford University Press)
- Rolls E T, Treves A and Tovee M J 1997 The representational capacity of the distributed encoding of information provided by populations of neurons in the primate temporal visual cortex *Exp. Brain Res.* **114** 177–85
- Rolls E T, Treves A, Tovee M and Panzeri S 1997 Information in the neuronal representation of individual stimuli in the primate temporal visual cortex *J. Comput. Neurosci.* **4** 309–33
- Sutton R S 1988 Learning to predict by the methods of temporal differences *Machine Learning* **3** 9–44
- Sutton R S and Barto A G 1981 Towards a modern theory of adaptive networks: expectation and prediction *Psychol. Rev.* **88** 135–70
- 1998 *Reinforcement Learning* (Cambridge, MA: MIT Press)
- Tanaka K, Saito H, Fukada Y and Moriya M 1991 Coding visual images of objects in the inferotemporal cortex of the macaque monkey *J. Neurophysiol.* **66** 170–89
- Wallis G 1996 Using spatio-temporal correlations to learn invariant object recognition *Neural Networks* **9** 1513–9
- Wallis G and Baddeley R 1997 Optimal unsupervised learning in invariant object recognition *Neural Comput.* **9** 883–94
- Wallis G and Rolls E T 1997 A model of invariant object recognition in the visual system *Prog. Neurobiol.* **51** 167–94
- Wallis G, Rolls E T and Foldiak P 1993 Learning invariant responses to the natural transformations of objects *Int. Joint Conf. on Neural Networks* vol 2 pp 1087–90
- Widrow B and Hoff M E 1960 Adaptive switching circuits *1960 IRE WESCON Convention Record* vol 4 (New York: Institute of Radio Engineers) pp 96–104
- Widrow B and Stearns S D 1985 *Adaptive Signal Processing* (Englewood Cliffs, NJ: Prentice-Hall)