*Network: Computation in Neural Systems*
June 2007; 18(2): 161–187

informa
healthcare



# Invariant object recognition with trace learning and multiple stimuli present during training

S. M. STRINGER, E. T. ROLLS, & J. M. TROMANS

*Department of Experimental Psychology, Oxford University, Centre for Computational Neuroscience, South Parks Road, Oxford OX1 3UD, England*

*(Received 4 May 2007; accepted 5 July 2007)*

**Abstract**
Over successive stages, the ventral visual system develops neurons that respond with view, size and position invariance to objects including faces. A major challenge is to explain how invariant representations of individual objects could develop given visual input from environments containing multiple objects. Here we show that the neurons in a 1-layer competitive network learn to represent combinations of three objects simultaneously present during training if the number of objects in the training set is low (e.g. 4), to represent combinations of two objects as the number of objects is increased to for e.g. 10, and to represent individual objects as the number of objects in the training set is increased further to for e.g. 20. We next show that translation invariant representations can be formed even when multiple stimuli are always present during training, by including a temporal trace in the learning rule. Finally, we show that these concepts can be extended to a multi-layer hierarchical network model (VisNet) of the ventral visual system. This approach provides a way to understand how a visual system can, by self-organizing competitive learning, form separate invariant representations of each object even when each object is presented in a scene with multiple other objects present, as in natural visual scenes.

**Keywords:** *Object recognition, inferior temporal visual cortex, competitive neural networks, trace learning, multiple objects, natural scenes*

## Introduction

Over successive stages, the ventral visual system develops neurons that respond with view, size and position (translation) invariance to objects including faces (Perrett et al. 1982; Desimone 1991; Perrett et al. 1991; Tanaka et al. 1991; Rolls 1992;

Correspondence: Prof. E. T. Rolls, Department of Experimental Psychology, Oxford University, South Parks Road, Oxford OX1 3UD, England. Tel: +44 1865 271348. Fax: +44 1865 310447. E-mail: edmund.rolls@psy.ox.ac.uk; url: http://www.cns.ox.ac.uk

ISSN 0954-898X print/ISSN 1361-6536 online 
DOI: 10.1080/09548980701556055

Logothetis et al. 1994; Rolls 2000; Rolls and Deco 2002; Rolls 2007). For example, it has been shown that the inferior temporal visual cortex has neurons that respond to faces and objects with translation (Kobotake and Tanaka 1994; Tovee et al. 1994; Ito et al. 1995; Op de Beeck and Vogels 2000; Rolls et al. 2003), size (Rolls and Baylis 1986; Ito et al. 1995), and view (Hasselmo et al. 1989; Booth and Rolls 1998) invariance.

In most previous studies of invariance learning in hierarchical networks that model the ventral visual stream, only one stimulus is presented at a time during training (Wallis and Rolls 1997; Rolls and Milward 2000; Elliffe et al. 2002; Perry et al. 2006; Rolls and Stringer 2006; Stringer et al. 2006). However, an important problem in understanding natural vision is how the brain can build invariant representations of individual objects even when multiple objects are present in a scene. In this article we start with a 1-layer competitive network, and show that it can learn separate representations of each object even with three objects always presented simultaneously during training. We show in the next section that an important factor in such learning is the number of objects in the training set, with transitions from learning about combinations of the objects present during training occurring as the number of objects in the training set is increased. In natural environments, there would be a large number of objects in the training set.

We then show that similar effects apply during the learning of invariant representations, using a trace learning rule. This is shown first in a 1-layer competitive network, in the third section. We then extend this by showing that similar principles can be applied to the training of a multilayer hierarchical model VisNet of the primate ventral visual system trained with a trace rule in the fourth section. We chose to apply this to VisNet (Wallis and Rolls 1997; Rolls and Milward 2000; Rolls and Stringer 2001, 2006) rather than to other hierarchical models of visual object recognition (Fukushima 1980, 1991, Bartlett and Sejnowski 1998; Riesenhuber and Poggio 1999, 2000) because VisNet has a set of hierarchically organized competitive nets to which the present principles should apply.

The concept of training separate object representations with multiple objects present simultaneously during training was investigated by Stringer and Rolls (2007). The present article makes important advances by showing that as the number of objects in the training set is increased, the network learns to form representations of first high-order combinations of the objects, then of low-order combinations of the objects, and then of individual objects; and by showing that invariant representations of individual objects with multiple objects in a scene can be formed using a temporal trace rule. Further, we present in this article for the first time VisNet simulations for a translation invariance problem with at least two objects always presented simultaneously during training, and in which the ten objects are shifted through overlapping retinal locations. This is a difficult test case, where the objects are represented by overlapping distributed representations.

## Training a 1-layer competitive network with input patterns that contain multiple independent stimuli

In this section, we investigate whether in a standard 1-layer competitive network (Hertz et al. 1991; Rolls and Deco 2002) separate representations of each object can

be formed when three objects are always simultaneously present during training (in contrast to forming neurons that respond to combinations of objects), and how this depends on the number of objects in the training set. Each object is presented to the network as a set of input features, which by occurring together define the object. To demonstrate the processes involved in learning representations of independent stimuli even with multiple stimuli present during training, we simulated them with the simplest model that would allow these processes to be investigated. This meant using nonoverlapping stimulus representations in a 1-layer network. Having demonstrated the principles involved in this section and in the third section, we proceed to extend the demonstration to a case in which the representations of each object overlap in space in the fourth section.

*Model*

The neural network architecture is shown in Figure 1. There is an input layer of cells with feedforward associatively modifiable synaptic connections onto an output layer of cells. At each timestep during learning, an input pattern is applied to the layer of input cells. Next, activity from the input layer is propagated through the feedforward connections to activate a set of cells in the output layer. Within the output layer there is competition implemented by feedback inhibition and the threshold non-linearity of neurons. Next, the synaptic weights between the active input cells and the active output cells are strengthened by associative (Hebbian) learning. The output cells self-organize to represent and thus categorize different patterns of activity in the input layer.

The competitive network contained 100 input cells and 100 output cells (unless otherwise specified), and was fully connected. Each simulation experiment used a set number $N$ of independent objects. The input patterns that represent each object consist of nonoverlapping contiguous blocks of $100/N$ active cells with binary (0, 1)

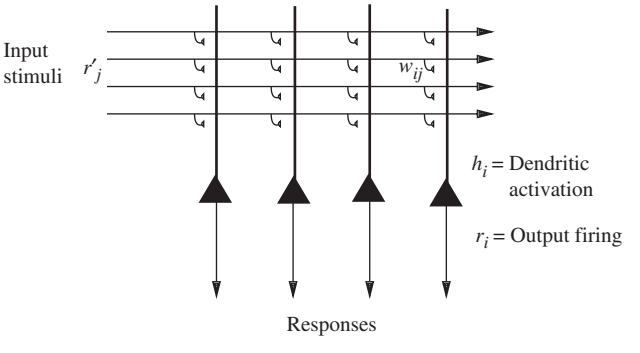


Figure 1. The architecture of a competitive neural network. There is an input layer of cells with feedforward associatively modifiable synaptic connections onto an output layer of cells. At each timestep during learning, an input pattern is applied to the layer of input cells. Next, activity from the input layer is propagated through the feedforward connections to activate a winning set of cells in the output layer. Within the output layer there is feedback inhibition, which ensures that only a small subset of output cells remains active. Next, the synaptic weights between the active input cells and the active output cells are strengthened. In this way, the output cells become associated with particular patterns of activity in the input layer.

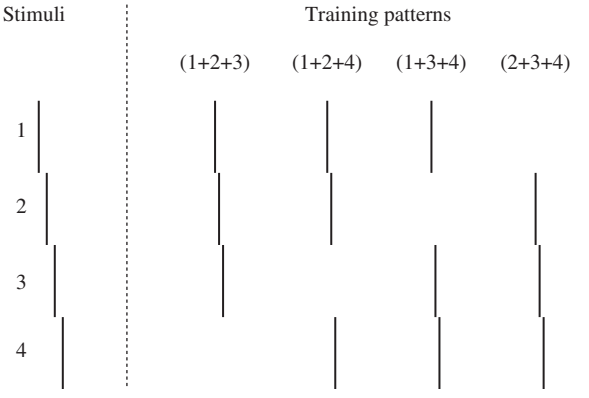


Figure 2. The input representations of the independent objects (in this case 4) used to test the 1-layer competitive network are shown on the left. On the right are shown the training patterns each of which consists of 3 objects presented simultaneously. All combinations of the three objects were used as training patterns. The competitive network has 100 input cells arranged in a column. Each simulation experiment used a set number $N$ of independent objects. The object patterns are represented by the activation of nonoverlapping contiguous blocks of $100/N$ cells. Object 1 is represented by cells 1 to $100/N$, object 2 by the next block of $100/N$ cells, and so on up to $N$ objects. The case of $N=4$ objects is represented in the figure. For the case of $N=4$ objects, there are a total of 4 triple-object training patterns as follows: $1+2+3$, $1+2+4$, $1+3+4$ and $2+3+4$. The cells that are active in each pattern are shown as black.

firing rates. Object 1 is represented by cells 1 to $100/N$, object 2 by the next block of $100/N$ cells, and so on up to $N$ objects. For the case $N=4$, Figure 2 (left) shows the input representations of the 4 independent objects. Figure 2 (right) shows the 4 input patterns that were used to train the 1-layer competitive network with 3 objects present simultaneously for the case $N=4$, in which the training patterns are: $1+2+3$, $1+2+4$, $1+3+4$, and $2+3+4$.

Each input pattern presented during training consisted of 3 objects presented simultaneously as shown in Figure 2 (right). The activity from the input layer is then propagated through the feedforward synaptic connections to activate a set of cells in the output layer. (The synaptic weights from the input to output cells are initially set to random values from a uniform distribution in the range 0–1, and then normalized to a vector length of 1. This is standard in competitive networks, and ensures that some firing of output cells will be produced by the inputs, with each output cell likely to fire at a different rate for any one input pattern.) The activations of the cells in the output layer are calculated according to

$$h_i = \sum_j w_{ij} r_j \tag{1}$$

where $h_i$ is the activation of output cell $i$, $r_j$ is the firing rate of input cell $j$, and $w_{ij}$ is the synaptic weight from input cell $j$ to output cell $i$.

The activation $h_i$ of each neuron was converted to the firing rate $r_i$ of each neuron using a threshold linear activation, and adjusting the threshold to achieve a prescribed sparseness of the representation in the output cells. (This represents a

process by which mutual inhibition between the output cells through inhibitory interneurons implements competition to ensure that there is only a small winning set of output cells left active.) The sparseness $a$ of the representation that was prescribed was defined, by extending the binary notion of the proportion of neurons that are firing, as follows

$$a = \frac{(\sum_{i=1}^{M} r_i/M)^2}{\sum_{i=1}^{M} r_i^2/M} \tag{2}$$

where $r_i$ is the firing rate of the $i$-th neuron in the set of $M$ neurons (Rolls and Treves 1990, 1998). In the simulations, the competition was achieved in an iterative cycle by feedback adjustment of the threshold for the firing of neurons until the desired sparseness was reached. The threshold was the same for all neurons. With the graded firing rates of the neurons, the result was that typically the proportion of neurons with nonzero firing rates after the competition was numerically somewhat larger than the sparseness value given as a parameter to the network.

Next, the synaptic weights between the active input cells and the active output cells are strengthened according to the associative Hebb learning rule

$$\delta w_{ij} = k r_i r_j \tag{3}$$

where $\delta w_{ij}$ is the change of synaptic weight, $r_i$ is the firing rate of output neuron $i$, $r_j$ is the firing rate of input neuron $j$, and $k$ is the learning rate constant which was set to 0.01 unless otherwise stated. To prevent the same few neurons always winning the competition, the synaptic weight vectors are set to unit length after each learning update for each training pattern, as is standard in competitive networks (Hertz et al. 1991; Rolls and Deco 2002). To implement weight vector normalization the synaptic weights were rescaled to ensure that for each output cell $i$ we have

$$\sqrt{\sum_j (w_{ij})^2} = 1, \tag{4}$$

where the sum is over all input cells $j$. Such a renormalization process may be achieved in biological systems through heterosynaptic long-term depression that depends on the existing value of the synaptic weight (Oja 1982; Rolls and Treves 1998; Rolls and Deco 2002). (Heterosynaptic long-term depression was described in the brain by Levy and colleagues (Levy 1985; Levy and Desmond 1985); see Brown et al. (1990)).

The presentation of all possible multi-object training patterns corresponded to one training epoch. For each experiment with fixed $N$, there were 1000 training epochs to ensure convergence of the synaptic weights. Similar results were obtained with a fixed order of presentation of the stimuli within an epoch as well as with random permutations in each epoch.

For each experiment, after training, the network was tested using $N$ single-object input patterns, each of which contained a different independent object as illustrated at the left of Figure 2. During this testing, for each output cell we recorded how many of the $N$ objects each cell responded to. This enabled each cell to be classified as responding to one, two, or three (or more) of the objects used as training stimuli. A neuron was classed as responsive if its firing was greater than 50% of the maximal

firing rate of any output cell to any test stimulus. We complemented this criterion by analysis of the synaptic weight matrices, as will be illustrated below.

*Simulation results*

The results described next showed that as the number of objects $N$ increases, there is a shift in the behavior of the network from representing the triple-object training patterns to representing pairs of stimuli. Then, as $N$ increases further, there is a shift from representing the stimulus-pairs to representing the $N$ independent stimuli. We note that with standard training of a competitive network on a set of patterns such as the triple objects illustrated in Figure 2 neurons might be expected to learn to respond to each triple-object training pattern. It was found that this happened only for small $N$, and that as $N$ increased, very different behavior was found.

For each experiment, the network was trained with all of the possible $N(N-1)(N-2)/6$ triple-object input patterns which can be formed from the $N$ stimuli, as illustrated in Figure 2.

In each simulation experiment we investigated a different fixed value of $N$. The sparseness of the output firing rates $a$ was set to 0.05.

The behavior of the competitive network for different values of $N$ can be understood from the cell firing rates and synaptic weight matrices shown in Figures 3 and 4. Figure 3 shows the firing rate responses of typical output cells after training for simulations with different values of $N$. The firing rates are shown when the individual objects are presented to the network after training with the triple-object patterns. The top row shows the firing rates of two typical output cells for a simulation with $N=4$. These cells have each learned to respond to three different objects, as shown also by the weight matrices in the top-right of Figure 4. These indicate that each neuron has increased its weights (indicated by black) to encode each triple-object training pattern. Because the neurons have learned to respond to a triple-object training pattern, each neuron has a response to each of the three objects presented separately. For the case $N=4$, a total of six simulation experiments were performed. Over 6 experiments, the average numbers of output cells that had learned to respond to 1, 2 or 3 stimuli were 0.3, 0.0, and 20.5. The standard errors of the means shown were in most cases less than 1. No cells were found that responded to 4 objects.

The middle row of Figure 3 shows the firing rates of two typical output cells for $N=10$. These cells have each learned to respond to two different objects, and this is also evident from the weight matrices in the lower left of Figure 4. Over 6 experiments, the average numbers of output cells which had learned to respond to 1, 2 or 3 objects were 0.0, 90.5, and 0.0. The standard errors of the means shown were in most cases less than 1. No cells were ever found which responded to 4 objects.

The bottom row of Figure 3 shows the firing rates of two typical output cells for $N=20$. These cells have each learned to respond to a single object, as is also evident from the weight matrices in the lower right of Figure 4. Over 6 experiments, the average numbers of output cells which had learned to respond to 1, 2 or 3 stimuli were 71.3, 7.0, and 0.0. The standard errors of the means shown were in most cases less than 2. No cells were ever found which responded to 4 or more stimuli. (Further analysis of the six simulation results with $N=20$ showed that in each case approximately equal numbers of output cells responded to each of the 20 objects.)

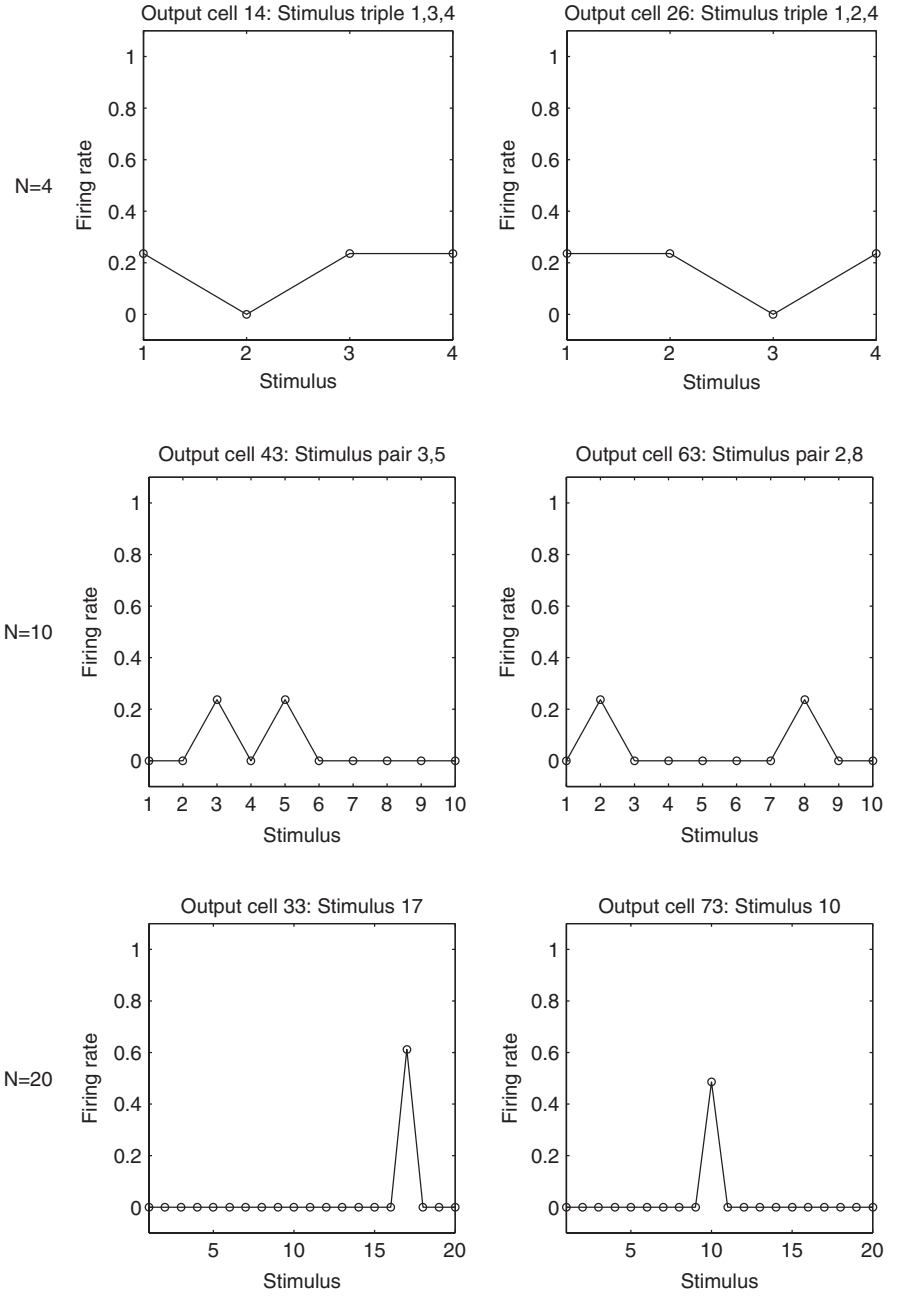


Figure 3. The learned firing rate responses of typical output cells for simulations with different values of $N$. The firing rates of a selection of typical cells are shown when the individual stimuli are presented to the network after training with the triple-stimulus patterns. The top row shows the firing rates of two typical output cells for a simulation with $N = 4$. These cells have each learned to respond to three different stimuli. The middle row shows the firing rates of two typical output cells for $N = 10$. These cells have each learned to respond to two different stimuli. The bottom row shows the firing rates of two typical output cells for $N = 20$. These cells have each learned to respond to a single stimulus.

We investigated further how the behavior of the system would scale up by simulating a case with object triples, but with $N = 50$, and accordingly in a system with a larger number of input neurons (200). The weight matrix after training (Figure 5) shows that almost all of the neurons have learned to respond to individual

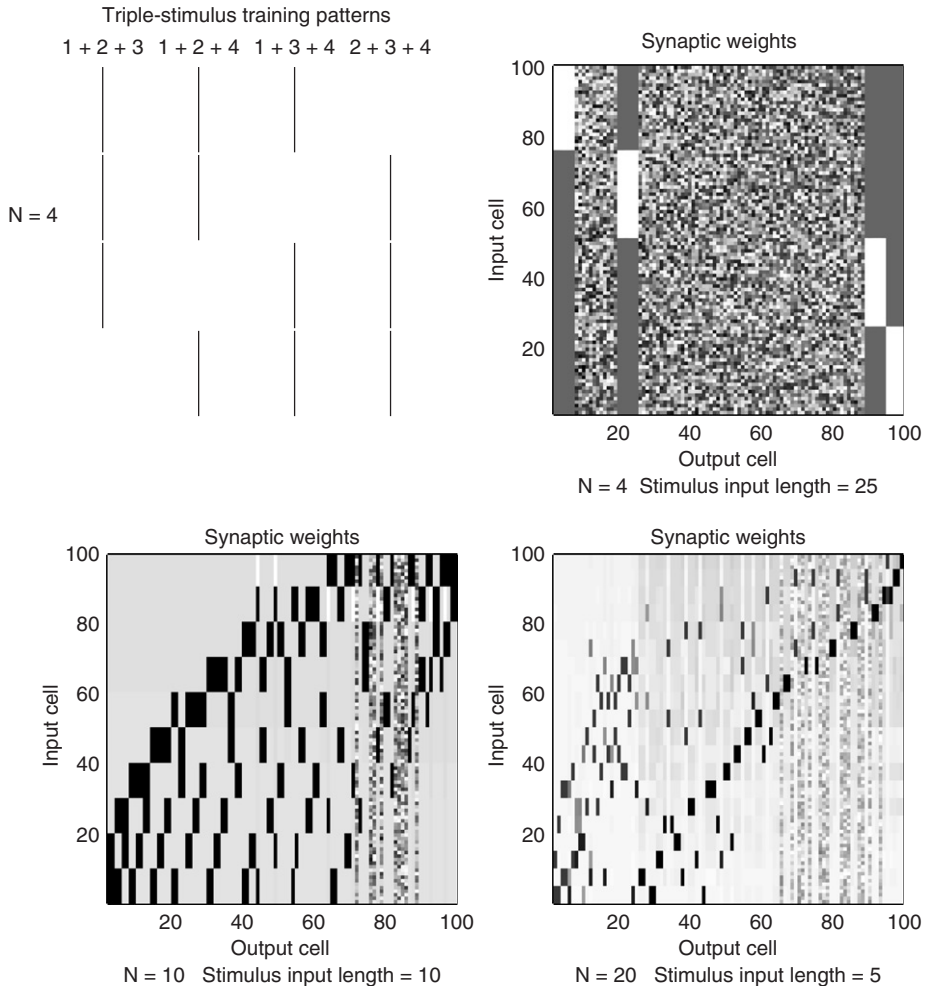


Figure 4. The synaptic weights after the network has been trained with triple-stimulus input patterns constructed from different numbers of stimuli $N$ (dark shading indicates high weight values). The top row shows results for the case $N = 4$ stimuli. On the left are shown the 4 triple-stimulus input patterns presented during training, while on the right is shown the synaptic weight matrix within the network after training. The output cells have been ordered to reveal the underlying weight structure developed during training. For the case $N = 4$, the input stimuli are 25 units long, and it can be seen that the output cells have typically learned to respond to triples of stimuli which were presented during training. The bottom left plot shows the synaptic weight matrix after training for the case $N = 10$. For this case the input stimuli are 10 units long, and it can be seen that the output cells have typically learned to respond to pairs of stimuli. The bottom right plot shows the synaptic weight matrix after training for the case $N = 20$. For this case the input stimuli are 5 units long, and the weight matrix shows that the output cells have typically learned to respond to single stimuli.

objects (196 of the 200 output neurons), and that only 4 neurons respond to object pairs, none learned to object-triples, even though the network had been trained with three objects always present during training.

The results of the experiments described in Figures 3–5 show that the transition from coding primarily for individual objects (with $N = 20$ and 50) to coding primarily for object-triples (with $N = 4$) was not sharp, with many neurons responding to object-pairs at $N = 10$. We explored intermediate values for $N$, and found intermediate behaviors. The important result is that as $N$ increases, there is a gradual transition, with neurons learning to encode single objects even when multiple objects are always present during training.

The reason why increasing the number of independent objects $N$ causes the competitive network to switch from learning to represent the triple-object training patterns to representing the pairs of objects present in the triples, and then to representing the independent objects may be understood by examining how often individual input neurons are co-active during training. The neurons from three different objects are coactive on only one occasion in each training epoch, in which all the triples are presented. In contrast, the number of times that the input neurons common to a particular pair of objects are co-active within a training epoch is of order $N$, and the number of times that the neurons within a single object are simultaneously active during one training epoch is of order $N^2$. Given that competitive networks effectively build categories based on correlations between sets of afferents to the network, as $N$ increases, a stage is reached where instead of representing the 3-object patterns actually presented during training, the network will instead form representations of the component object-pairs. Then, as $N$ is increased further, the network similarly switches from representing object-pairs to individual objects. The number of output neurons is clearly not a limiting factor that causes these transitions, for as shown in Figure 4, multiple neurons are allocated to any one representation, and some neurons remain unallocated to any pattern with their random initial weights evident.

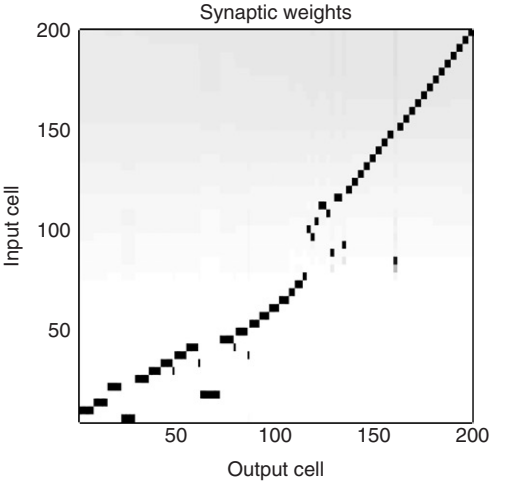


Figure 5. The synaptic weights after the network has been trained with triple-object input patterns constructed with $N = 50$ objects. Conventions as in Figure 4.

**Learning transform invariant representations in a 1-layer competitive network through trace learning**

An approach to invariance learning, *trace learning* (Földiák 1991; Wallis and Rolls 1997; Rolls and Milward 2000; Rolls and Stringer 2001), relies on the temporal continuity of visual stimuli in the real world. In this section we show that a 1-layer competitive network can combine the trace learning mechanism for learning transform-invariant representations with the learning dynamics described above in the second section for developing separate representations of independent stimuli even when multiple stimuli are shown simultaneously. Spatial continuity can be used to train invariant representations with multiple objects present simultaneously (Stringer and Rolls 2007), but this is the first time that this has been investigated with the more widely studied trace learning rule in a 1-layer competitive network.

*Trace learning*

Trace learning utilizes the temporal continuity of objects in the world (over short time periods) to help the learning of invariant representations. The concept here is that on the short time scale, of e.g. a few seconds, the visual input is more likely to be from different transforms of the same object, rather than from a different object. A theory used to account for the development of view invariant representations in the ventral visual system uses this temporal continuity in a *trace learning rule* (Földiák 1991; Rolls 1992; Wallis and Rolls 1997; Rolls and Milward 2000; Rolls and Stringer 2001). The trace learning mechanism relies on associative learning rules, which utilize a temporal trace of activity in the postsynaptic neuron (Földiák 1991; Rolls 1992). Trace learning encourages neurons to respond to input patterns which occur close together in time, which, given the statistics of natural visual inputs, are likely to represent different transforms (positions) of the same object.

The trace learning rule (Földiák 1991; Rolls 1992; Wallis and Rolls 1997; Rolls and Milward 2000) encourages neurons to develop invariant responses to input patterns that tended to occur close together in time, because these are likely to be from the same object. The particular rule in the simulations described next was

$$\delta w_{ij} = k\bar{r}_i^{\tau} r_j^{\tau} \tag{5}$$

where $\delta w_{ij}$ is the change of synaptic weight, $r_i^{\tau}$ is the trace value of the firing rate of output neuron $i$ at timestep $\tau$, $r_j^{\tau}$ is the firing rate of input neuron $j$ at timestep $\tau$, and $k$ is the learning rate constant which was set to 0.01 unless otherwise stated. The trace $\bar{r}_i^{\tau}$ is updated according to

$$\bar{r}_i^{\tau} = (1 - \eta) r_i^{\tau} + \eta \bar{r}_i^{\tau - 1} \tag{6}$$

where $\eta$ may be set anywhere in the interval [0, 1], and for the simulations described here $\eta$ was set to 0.9. To prevent the same few neurons always winning the competition, the synaptic weight vectors are set to unit length using equation 4 after each learning update for each training pattern.

An advantage of trace learning over continuous transform learning (Perry et al. 2006; Stringer et al. 2006) is that trace learning does not require the transforms of each stimulus to be similar or overlapping. Instead, trace learning is able to associate

together very different (i.e. dissimilar and nonoverlapping) transforms that occur close together in time because they are from the same object.

*Model simulations*

The neural network architecture is similar to that described in the second section above and shown in Figure 1. However, the input patterns are now altered to allow the objects to be presented in different nonoverlapping transforms, as described next.

In the one-layer competitive network simulations described next with trace learning, there are $N = 10$ independent objects, each of which can occur in 4 transforms. Figure 6 shows schematically the input representations of the 4 transforms of each of the 10 independent objects. (The diagram explicitly shows the 4 transforms of objects 1, 2 and 10. The intervening objects 3 to 9 transform similarly.) The 10 object patterns are each represented by the activation of a contiguous block of 5 cells. Each object undergoes transformation by shifting the

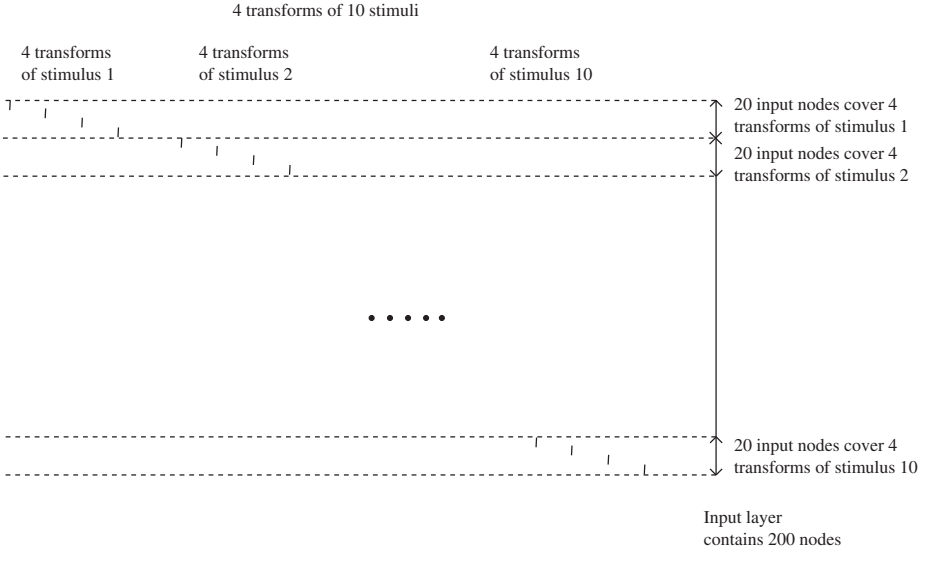


Figure 6. In the 1-layer competitive network simulations with trace learning there are $N = 10$ independent stimuli, each of which can occur in 4 transforms. Here we show the input representations of the 4 transforms of each of the 10 independent stimuli. (The diagram explicitly shows the 4 transforms of stimuli 1, 2, and 10. The intervening stimuli 3 to 9 transform similarly.) The 10 stimulus patterns are each represented by the activation of a contiguous block of 5 cells. Each stimulus undergoes transformation by shifting the location of the stimulus by one whole stimulus length at a time. This means that successive stimulus transforms do not overlap, which will ensure that continuous transformation learning is prevented from operating. During training, each stimulus is shifted through 4 nonoverlapping locations. Given that each stimulus is 5 neurons long, this means that the 4 transforms of each stimulus cover 20 input neurons. None of the transforms of each 5-neuron stimulus overlap. The 4 transforms of each stimulus cover a unique disjoint block of 20 input neurons. The input layer is thus set to contain 200 neurons in order to be able to represent all of the transforms of all of the stimuli.

location of the object by one whole object length at a time. This means that successive object transforms do not overlap, which will ensure that continuous transformation learning is prevented from operating. During training, each object is shifted through 4 nonoverlapping locations. Given that each object is 5 neurons long, this means that the 4 transforms of each object cover 20 input neurons. None of the transforms of each 5-neuron object overlap. The 4 transforms of each object cover a unique disjoint block of 20 input neurons. The input layer is thus set to contain 200 neurons in order to be able to represent all of the transforms of all of the objects.

During training, the network is presented with every possible pair of objects. The network was presented with a training sequence of 4 transforms for each object-pair, with the two objects transforming together. Examples of these object-pair training sequences are shown on the left of Figure 8. As an example, we have shown 4 transforms of object-pair $1+2$ and 4 transforms of object pair $1+3$. The presentation of the 4 transforms of all possible $N(N-1)/2$ object-pairs corresponded to one training epoch. The sparseness of the output firing rates was set to 0.2 during training and testing, the learning rate was set to $k=0.01$, and there were 1000 training epochs to ensure convergence of the synaptic weights. After training, the network was tested with all 4 transforms of each of the $N=10$ objects.

Figure 7 shows the firing rate responses of two typical output cells after the network has been trained with the trace rule and nonoverlapping object transforms. The upper two rows show the response of output cell 25 to each of the $N=10$ objects in each of their 4 transforms. It can be seen that output cell 25 responds to object 4 over all 4 transforms, but does not respond to any of the other objects in any location. The lower two rows show the response of output cell 73 to each of the $N=10$ objects in each of their 4 transforms. It can be seen that output cell 73 responds to object 6 over all 4 transforms, but does not respond to any of the other objects in any location.

Figure 8 shows the synaptic weights after the training. On the left are shown some of the paired-object training patterns. On the right is shown the synaptic weight matrix after training, where dark shading indicates a high weight value. For the weight matrix plot, the output cells have been ordered to reveal the underlying weight structure developed during training. It can be seen that the weight matrix has a block-diagonal structure, which clearly shows that the output cells have each learned to respond to all 4 transforms of one particular object, but have not learned to respond to any of the other objects. Since the blocks on the diagonal are almost square, the weight matrix also confirms that the output cells are distributed fairly evenly among the 10 objects, with disjoint subsets of approximately 10 output cells having learned to respond in a transform-invariant way to each of the 10 independent objects.

Six simulation runs were performed in total with different initial random synaptic weights. For all six of the simulations, each of the 100 output cells learned to respond to all transforms of one particular object, but did not respond to any of the transforms of the other objects. In this way, each of the output cells developed a transform-invariant representation of one of the objects. Furthermore, for each simulation, disjoint subsets of approximately 10 output cells learned to respond in a transform-invariant way to each of the independent objects. Thus, the $N=10$ input objects were all equally well-represented by the output layer of the network.

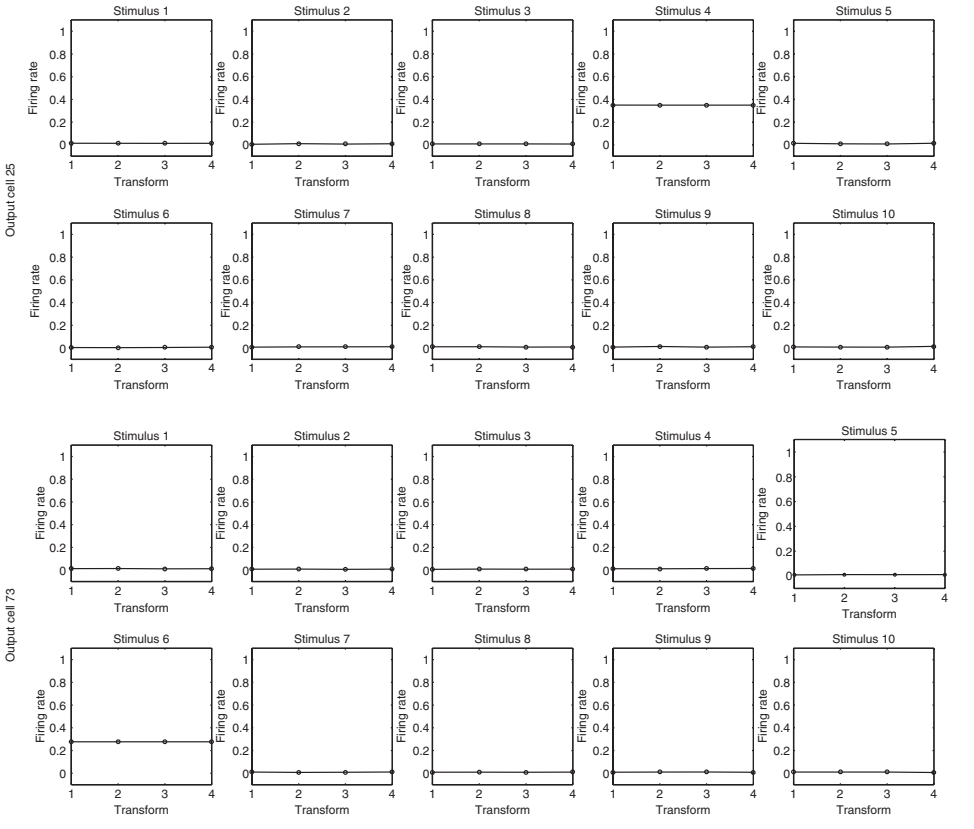


Figure 7. The firing rate responses of two typical output cells, after the network has been trained with the trace rule (5) on pairs of $N = 10$ stimuli, each of which was shifted through 4 nonoverlapping transforms (i.e. locations) during learning. For this test case, all 100 output cells learned to respond to all transforms of a single stimulus, but did not respond to any of the other stimuli. In this way, each of the output cells developed a transform-invariant representation of one of the stimuli. Furthermore, disjoint subsets of approximately 10 output cells learned to respond in a transform-invariant way to each of the independent stimuli. Thus, the $N = 10$ input stimuli were all equally well-represented by the output layer of the network. The upper two rows show the response of output cell 25 to each of the $N = 10$ stimuli in each of their 4 transforms. It can be seen that output cell 25 responds to stimulus 4 over all 4 transforms, but does not respond to any of the other stimuli in any location. The lower two rows show the response of output cell 73 to each of the $N = 10$ stimuli in each of their 4 transforms. It can be seen that output cell 73 responds to stimulus 6 over all 4 transforms, but does not respond to any of the other stimuli in any location.

The above simulations used 1000 training epochs. However, performance was reduced when only a few training epochs were used. Over six simulation runs with 5 training epochs, an average of 0.16 output cells (with standard error 0.16) learned to respond to all transforms of one particular object, but did not respond to any of the transforms of the other objects.

These results show for the first time that the trace rule can be used to train invariant representations of objects in a competitive network even when there is

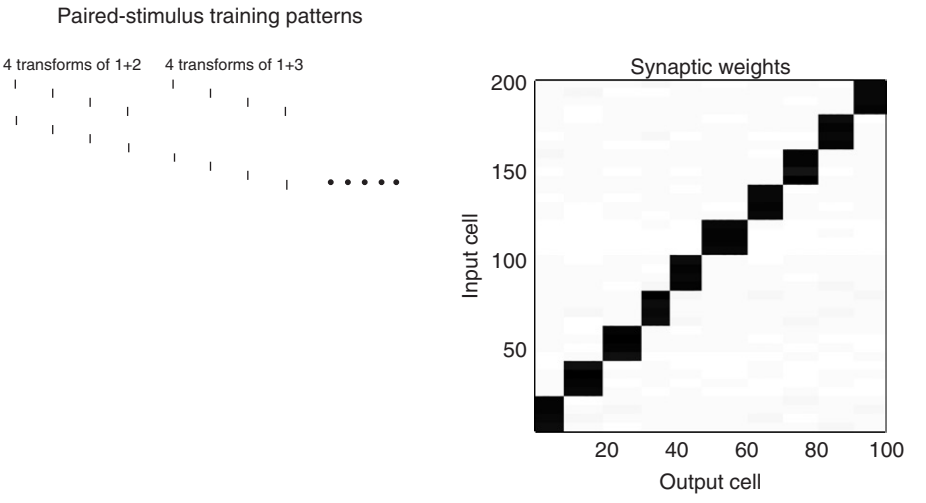


Figure 8. The synaptic weights after the network has been trained with the trace rule (5) on pairs of $N = 10$ stimuli, each of which was shifted through 4 nonoverlapping transforms (i.e. locations) during learning. On the left are shown some of the paired-stimulus training patterns. As an example, we have shown 4 transforms of stimulus-pair $1 + 2$ and 4 transforms of stimulus-pair $1 + 3$. On the right is shown the synaptic weight matrix after training, where dark shading indicates a high weight value. For the weight matrix plot, the output cells have been ordered to reveal the underlying weight structure which developed during training. It can be seen that the weight matrix has a block-diagonal structure, which clearly shows that the output cells have each learned to respond to all 4 transforms of one particular stimulus, but have not learned to respond to any of the other stimuli. Since the blocks on the diagonal are almost square, the weight matrix also confirms that the output cells are distributed fairly evenly among the 10 stimuli, with disjoint subsets of approximately 10 output cells having learned to respond in a transform-invariant way to each of the 10 independent stimuli.

more than one object (in this case two) always present during training. The competitive network is able to set up separate representations of each object if each object is seen on different occasions with objects that tend to be different on different trials, as would be the case in the real world.

## Training a multi-layer feedforward network (VisNet) with multiple 3-dimensional objects shifting through space

We now show how the trace-based learning mechanisms with multiple objects described above can operate in a more biologically realistic model of transform (e.g. location or view) invariant object recognition in the ventral visual processing stream, VisNet (Wallis and Rolls 1997; Rolls and Milward 2000) to enable learning of individual objects even with multiple objects present during training. Moreover, we show that trace learning can do this in a situation when continuous transformation learning cannot occur, that is when the different transforms of an object do not overlap.

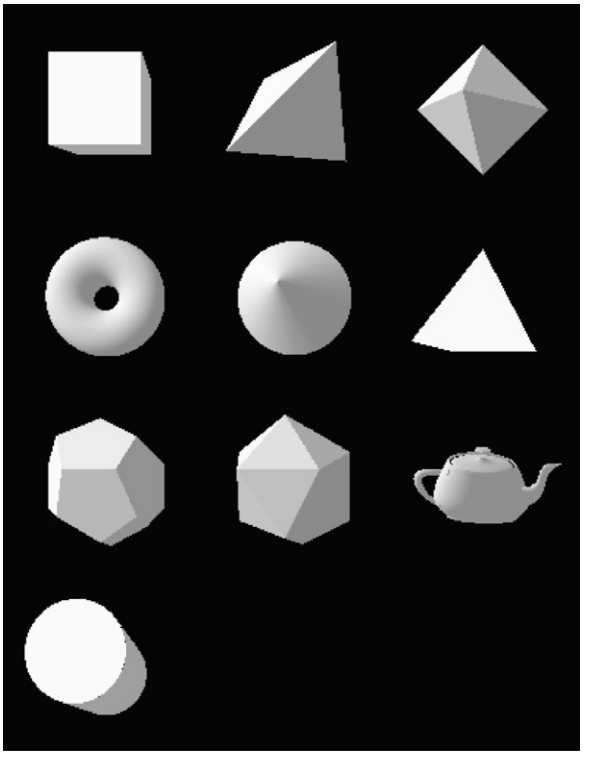

Figure 9. The ten stimuli used to train the VisNet network. The stimuli were 3D objects, each shown in a $2 \times 2$ grid of 4 locations in a test of location-invariant recognition. The effect of the ambient lighting and a single diffuse light source to allow different surfaces to be shown with different intensities is illustrated. In rows from left to right the stimuli were a cube, tetrahedron, octahedron, torus, cone, pyramid, dodecahedron, icosahedron, teapot, and a cylinder. During training, the network was presented with every possible pair of stimuli simultaneously shifting clockwise around the 4 locations in the grid. After training, the network was tested with image sequences of the ten individual stimuli shifting clockwise around the 4 locations in the grid.

To ensure these conditions, we trained on a shift invariance problem, in which each object was a 3D object shown in a $2 \times 2$ grid of 4 nonoverlapping locations. (OpenGL was used to build a 3D representation of the objects, and then to project different views onto a 2D image.) There were ten objects as shown in Figure 9. For ten objects, there are 45 possible ways of pairing the objects during training. During training, each of the 45 possible pairs of objects were shown shifting together clockwise around the four locations in the grid. A typical training sequence of four transforms is shown in Figure 10, where we show a cube and a pyramid shifting around the grid together. Whenever one object is presented in its sequence of locations, another object is also presented simultaneously shifting around the grid in the same way. (It can be seen from Figure 10 that as each object shifts around the grid, there is also an additional slight change in perspective.) The nature of the object pairing was of the same form described above, that is each object was paired during training with every other object. Thus, for example, object 1 was paired with object 2 during all four transforms of the type shown in Figure 10. Then object 1

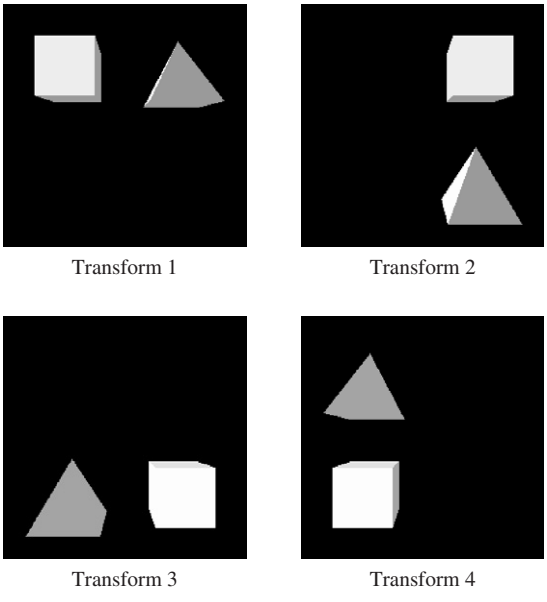


Figure 10. An example of a 4-frame training image sequence. The image sequence shows the cube and the pyramid shifting clockwise around the $2 \times 2$ grid of 4 training locations. In total, there are 45 such image sequences used during training, with each image sequence corresponding to one of the 45 possible pairs of stimuli shifting together clockwise around the 4 locations in the grid.

was paired with object 3 for all transforms, etc until object 1 had been paired through all transforms for every other object. An analogous procedure was carried out with all 45 possible pairs.

The model architecture (VisNet) (Wallis and Rolls 1997) is based on the following: (i) A series of hierarchical competitive networks with local graded inhibition. (ii) Convergent connections to each neuron from a topologically corresponding region of the preceding layer, leading to an increase in the receptive field size of neurons through the visual processing areas. (iii) Synaptic plasticity based on a trace learning rule to allow trace invariance learning.

The model consists of a hierarchical series of four layers of competitive networks, corresponding to V2, V4, the posterior inferior temporal cortex, and the anterior inferior temporal cortex, as shown in Figure 11. The forward connections to individual cells are derived from a topologically corresponding region of the preceding layer, using a Gaussian distribution of connection probabilities. These distributions are defined by a radius which will contain approximately 67% of the connections from the preceding layer. The values used are given in Table I.

Before objects are presented to the network's input layer they are pre-processed by a set of input filters which accord with the general tuning profiles of simple cells in V1. The input filters used are computed by weighting the difference of two Gaussians by a third orthogonal Gaussian according to the following:

$$\Gamma_{xy}(\rho,\theta,f) = \rho\left[ e^{-((x\cos\theta+y\sin\theta)/(\sqrt{2}/f))^2} - \frac{1}{1.6} e^{-((x\cos\theta+y\sin\theta)/(1.6\sqrt{2}/f))^2} \right] e^{-((x\sin\theta-y\cos\theta)/(3\sqrt{2}/f))^2} \quad (7)$$

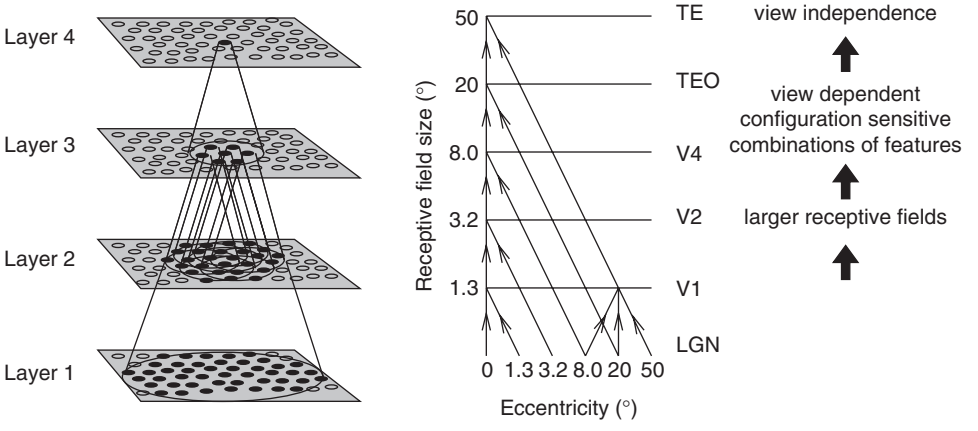


Figure 11. Left: Stylized image of the 4-layer network. Convergence through the network is designed to provide fourth layer neurons with information from across the entire input retina. Right: Convergence in the visual system V1: visual cortex area V1; TEO: posterior inferior temporal cortex; TE: inferior temporal cortex (IT).

Table I. Network dimensions showing the number of connections per neuron and the radius in the preceding layer from which 67% are received.

| | Dimensions | Number of connections | Radius |
|---|---|---|---|
| Layer 4 | $32 \times 32$ | 100 | 12 |
| Layer 3 | $32 \times 32$ | 100 | 9 |
| Layer 2 | $32 \times 32$ | 100 | 6 |
| Layer 1 | $32 \times 32$ | 272 | 6 |
| Retina | $128 \times 128 \times 32$ | – | – |

where $f$ is the filter spatial frequency, $\theta$ is the filter orientation, and $\rho$ is the sign of the filter, i.e. $\pm 1$. Individual filters are tuned to spatial frequency (0.0625 to 0.5 cycles/pixel); orientation ($0^o$ to $135^o$ in steps of $45^o$); and sign ($\pm 1$). The number of layer 1 connections to each spatial frequency filter group is given in Table II.

The activation $h_i$ of each neuron $i$ in the network is set equal to a linear sum of the inputs $y_j$ from afferent neurons $j$ weighted by the synaptic weights $w_{ij}$. That is,

$$h_i = \sum_j w_{ij} y_j \tag{8}$$

where $y_j$ is the firing rate of neuron $j$, and $w_{ij}$ is the strength of the synapse from neuron $j$ to neuron $i$.

Within each layer, competition is graded rather than winner-take-all, and is implemented in two stages. First, to implement lateral inhibition the activation $h$ of neurons within a layer is converted to firing rates $r$ using a linear activation function followed by convolution with a spatial filter, $I$, where $\delta$ controls the contrast and

Table II. Layer 1 connectivity. The number of connections from each spatial frequency set of filters are shown. The spatial frequency is in cycles per pixel.

| Frequency | 0.5 | 0.25 | 0.125 | 0.0625 |
|---|---|---|---|---|
| Number of connections | 201 | 50 | 13 | 8 |

Table III. Lateral inhibition parameters.

| Layer | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Radius, $\sigma$ | 1.38 | 2.7 | 4.0 | 6.0 |
| Contrast, $\delta$ | 1.5 | 1.5 | 1.6 | 1.4 |

Table IV. Sigmoid parameters.

| Layer | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Percentile | 99.2 | 98 | 88 | 91 |
| Slope $\beta$ | 190 | 40 | 75 | 26 |

$\sigma$ controls the width, and $a$ and $b$ index the distance away from the centre of the filter in orthogonal directions

$$I_{a,b} = \begin{cases} -\delta e^{-(a^2+b^2)/\sigma^2} & \text{if } a \neq 0 \text{ or } b \neq 0, \\ 1 - \sum_{\substack{a\neq 0 \\ b\neq 0}} I_{a,b} & \text{if } a = 0 \text{ and } b = 0. \end{cases} \tag{9}$$

The lateral inhibition parameters are given in Table III.

The modeling of lateral inhibition also incorporates contrast enhancement (related to nonlinearity in the neurons) which is applied by means of a sigmoid activation function

$$y = f^{\text{sigmoid}}(r) = \frac{1}{1 + e^{-2\beta(r-\alpha)}} \tag{10}$$

where $r$ is the firing rate after the lateral inhibition filter defined above, $y$ is the firing rate after contrast enhancement, and $\alpha$ and $\beta$ are the sigmoid threshold and slope respectively. The parameters $\alpha$ and $\beta$ are constant within each layer, although $\alpha$ is adjusted to control the sparseness of the firing rates. For example, to set the sparseness to, say, 5%, the threshold is set to the value of the 95th percentile point of the activations within the layer. The parameters for the sigmoid activation function are shown in Table IV.

At each timestep, the activity due to the object on the retina is propagated in a feedforward fashion through the network, stimulating patterns of activity in the later layers. Once the activity patterns have been computed in the various layers including competitive lateral inhibition as described above, the synaptic weights of the forward connections between the layers are updated by the trace learning rule (5) with the postsynaptic trace from the previous timestep as is standard in VisNet simulations. The value of $\eta$ was 0.8. To bound the growth of each neuron's synaptic weight vector, its length is normalized at the end of each timestep during training according to Equation 4.

*Training and test procedure*

In the experiment, ten objects were used because this was found to be sufficient in the third section earlier to allow the 1-layer network to develop representations of individual objects when the network was trained on object-pairs. The objects and pairing during training are described above and in Figures 9 and 10.

To train the network each pair of objects is presented to the network in a temporal sequence of the 4 transforms (i.e. locations). At each presentation the activation of individual neurons is calculated, then their firing rates are calculated, and then the synaptic weights are updated. The presentation of all the object-pairs across all transforms constitutes 1 epoch of training. In this manner the network is trained one layer at a time starting with layer 1 and finishing with layer 4. In the investigation described here, the numbers of training epochs for layers 1–4 were 50, 100, 100, and 75 respectively.

After training the network, we tested it by recording the firing rates of the neurons in the 4th (output) layer of the network as the network was presented with image sequences of the ten individual objects each presented in every location. We used an information theoretic approach described next to measure whether a neuron had a similar response to every transform of an object, but little response to any transform of the other objects (Rolls and Milward 2000). For each cell the single cell information measure used was the amount of information a cell conveyed about the most effective object. This is computed using the following formula with details given by Rolls et al. (1997), Rolls and Milward (2000) and Rolls (2008). The stimulus-specific information or surprise $I(s, R)$ is the amount of information the set of responses $R$ has about a specific stimulus $s$, and is given by

$$I(s, R) = \sum_{r \in R} P(r|s) \log_2 \frac{P(r|s)}{P(r)} \tag{11}$$

where $r$ is an individual response from the set of responses $R$. The maximum single cell information measure is

$$\text{Maximum single cell information} = \log_2(\text{Number of stimuli}), \tag{12}$$

where in this case the number of objects is 10. This gives a maximum single cell information measure of 3.32 bits, and is achieved when all the responses to the different transforms of a stimulus are similar, that is when the representation is invariant, and when the neuron does respond to any other stimulus.

A multiple cell information measure, the average amount of information that is obtained about which stimulus was shown from a single presentation of a stimulus from the responses of all the cells, enabled measurement of whether across a population of cells information about every object in the set was provided. Procedures for calculating the multiple cell information measure are given elsewhere (Rolls et al. 1997; Rolls and Milward 2000; Stringer et al. 2006). In the experiments presented later, the multiple cell information was calculated from the subset of 50 output cells that gave the highest single cell information values for each object (with the 5 most selective for each object being used).

*VisNet simulation results*

After the network had been trained on the 45 pairs of objects each shifting clockwise around the 2 × 2 training grid, a number of cells in the 4th (output) layer were found to have learned to respond to one of the ten objects over all four locations, but did not respond to any of the other objects in any location, as described next. In particular, some neurons that responded to a particular object after training had no response to any of the other objects, even though the other objects had been presented simultaneously with the particular object during training.

Figure 12 shows the firing rate responses of cell (6,12) in the 4th (output) layer to all 4 locations of the ten individual objects before training. The ten plots show the

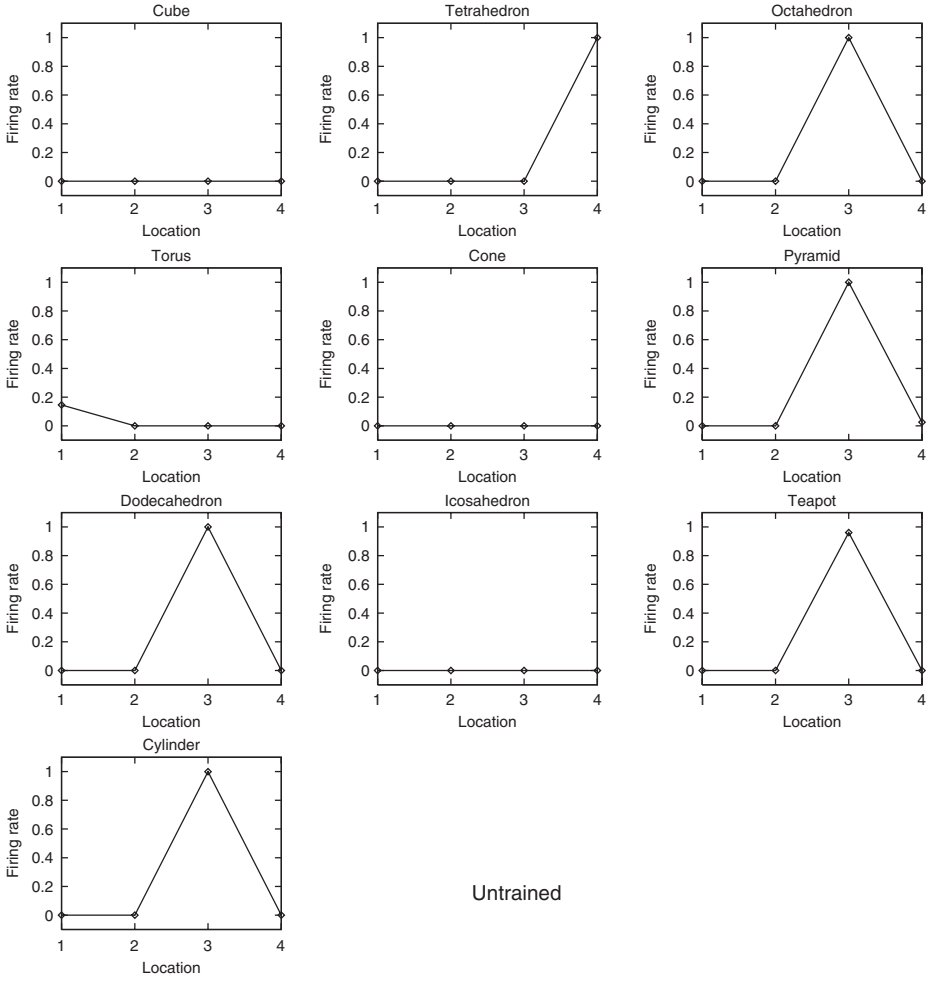


Figure 12. The firing rate responses of cell (6, 12) in the 4th (output) layer to all 4 locations of the ten individual stimuli before training. The ten plots show the responses of cell (6, 12) to each of the ten stimuli as they are shifted clockwise around the 4 locations in the grid. It can be seen that before training, the cell responds randomly to various of transforms of the different stimuli.

responses of cell (6,12) to each of the ten objects as they are shifted clockwise around the 4 locations in the grid. It can be seen that before training the cell responds randomly to various of the transforms of the different objects. Figure 13 shows the firing rate responses of cell (6,12) in the 4th (output) layer to all 4 locations of the ten individual objects after training. It is evident that after training, the cell has learned to respond to the pyramid over all four locations, but does not respond to any of the other objects in any locations. The cell has thus learned to respond location invariantly to the pyramid, even though during training the pyramid was presented on different trials with other members of the training set of objects.

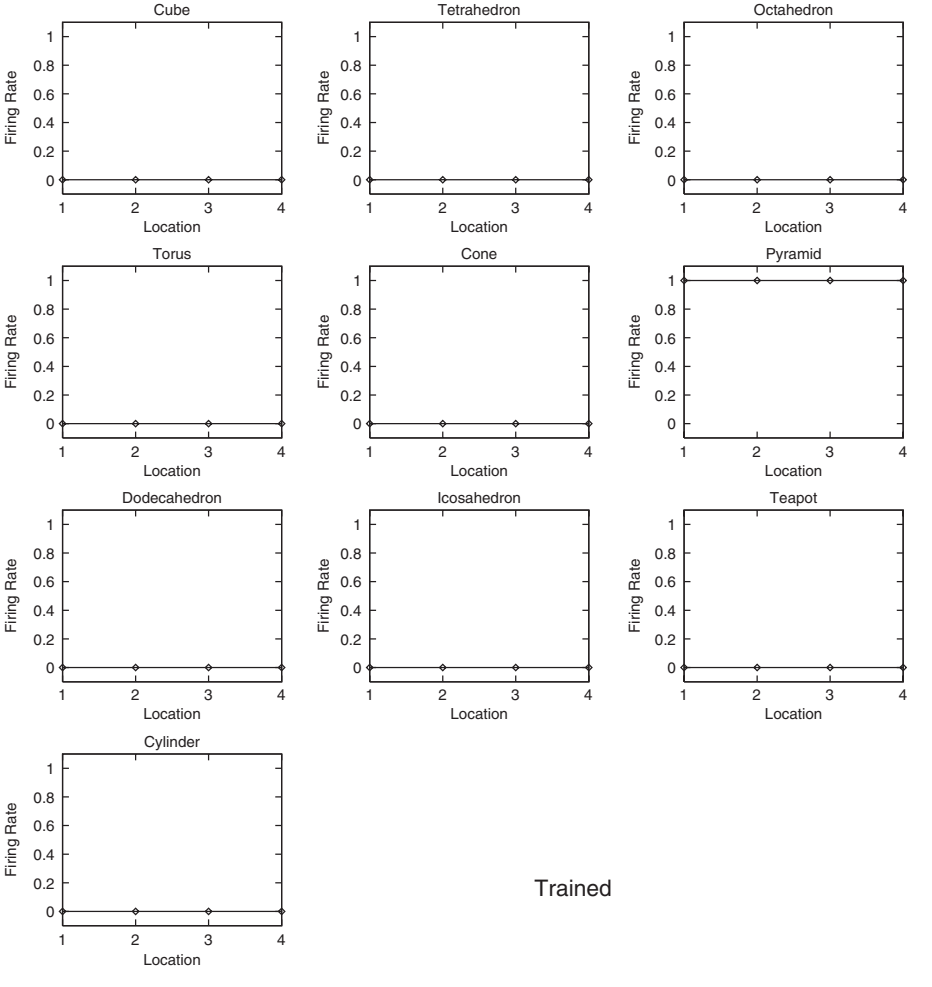


Figure 13. The firing rate responses of cell (6, 12) in the 4th (output) layer to all 4 locations of the ten individual stimuli after training. The ten plots show the responses of cell (6, 12) to each of the ten stimuli as they are shifted clockwise around the 4 locations in the grid. It is evident that after training, the cell has learned to respond to the pyramid over all four locations, but does not respond to any of the other stimuli in any locations. The cell has thus learned to respond location invariantly to the pyramid.

Figure 14 shows the information results obtained when VisNet was tested with the ten individual objects shifting clockwise around the grid. Results are presented after training the network with image sequences of all 45 possible object-pairs (unbroken line), and with a random untrained network (dashed line). The single cell

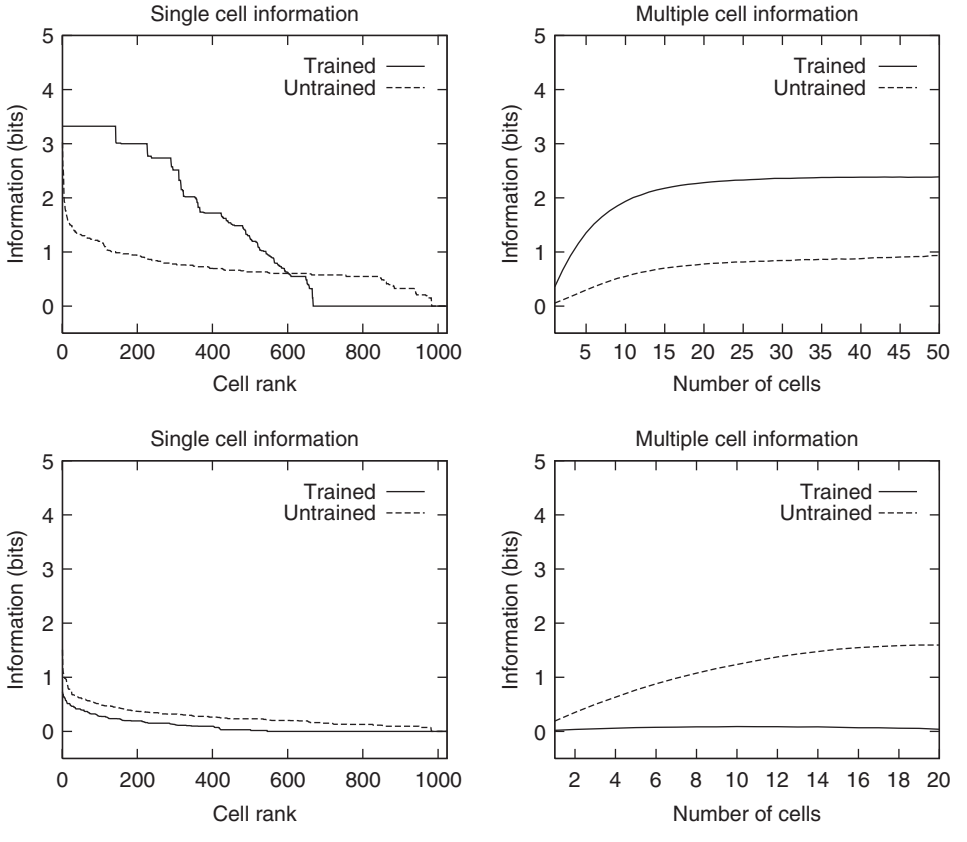


Figure 14. Top: Information about which object was shown invariantly with respect to position when VisNet was tested with the ten individual objects shifting clockwise around the 4 locations in the grid. Results are presented after training the network with image sequences of all 45 possible object-pairs (unbroken line), and with a random untrained network (dashed line). Top left: The single cell information measures for all top (fourth) layer neurons ranked in order of their invariance to the objects are shown. It can be seen that training the network on the object pairs has led to many top layer neurons attaining the maximal level of single cell information of 3.32 bits. Top right: Multiple cell information measure showing that a population of 50 cells has more information about which object was shown invariantly with respect to position after training with the trace rule than with the initial untrained random connectivity. These results show that training the network on the object pairs has led to many output neurons learning to respond to one of the ten individual objects over all possible 4 locations. Bottom: Analysis of the information about the location where the image was shown. Bottom left: single cell information. Bottom right: multiple cell information. Before training, the random connectivity allowed some information to be read out about in which of the 4 possible locations the object had been shown. (The maximum value possible is thus 2 bits.) After training to form position invariant representations of the objects, there was very little information about where the object had been shown.

information measures for all top (fourth) layer neurons ranked in order of their invariance to the objects are shown. It can be seen that training the network on the object-pairs has led to many top layer neurons attaining the maximal level of single cell information of 3.32 bits. This corresponds to a neuron responding to only one of the 10 objects, and indeed producing a similar response to every transform of that object. If a neuron had responded to only some of the transforms of that object, or had responded to any of the transforms of another object, the single cell information would have been less than 3.32 bits. Thus the information theory analysis shown in Figure 14 shows that many of the top layer neurons have learned to respond to all transforms (translations) of one object, and to no transforms of any other object, even though each object was being trained in the presence of one other object in the scene.

Analysis of the responses showed that different output cells had learned to respond invariantly to different objects. For example, three of the objects had output cells with perfect recognition performance (3.32 bits), half of the training objects had neurons that gave high levels of information (greater than 3 bits) about the presence of that object, and 9 objects had neurons that provided with more than half the maximal level of information. For comparison, in the control (untrained) condition, no objects had neurons that reached half the maximal level of information.

To confirm that the training produced position invariant representations of objects, and did not instead result in neurons that encoded the positions of the objects, we show in Figure 14 (bottom) the information about the location in which an object had been shown. The single cell and multiple cell measures show that before training, the random connectivity allowed some information to be read out about in which of the 4 possible locations the object had been shown. After the training described earlier to form position invariant representations of the objects, there was very little information about where the object had been shown.

We have previously shown that the sparseness of the output representation is not a crucial factor in forming independent representations of individual objects in a 1-layer network even with multiple objects present during training (Stringer and Rolls 2007). Indeed, the 1-layer network operated in this way even when the sparseness $a$ varied in the range 0.01–0.5. To confirm that the sparseness of the output representation is not a crucial factor in forming independent representations in VisNet, we varied the sparseness by altering the value of $\alpha$ in Equation 10 in layer 4, and showed that VisNet also produced independent object representations over a wide range of output sparseness values (equivalent to $a$ in the range 0.09–0.5). Thus the crucial factor in determining whether VisNet too produces representations of individual objects even when multiple objects are present simultaneously during training is not the sparseness of the output representation, but the number $N$ of different objects.

Learning to individual objects even with multiple objects present during training occurs in this hierarchical network model of the ventral visual stream because during training of any one object, the features of that object always co-occur, and this leads the competitive networks in the hierarchy to form representations of these co-active inputs. Many neurons do not learn to respond to the other objects being presented simultaneously. This occurs because there are 9 other objects, each of which is only sometimes presented with the first object, so that the features of these other objects

are not presented very frequently with the features of the first object. Moreover, this is a particularly hard problem, because the transforms of 10 different objects overlap in every training position. This is the first time that overlapping stimuli have been used in a multiple object training problem.

## Discussion

In natural vision an important problem is how the brain can build invariant representations of individual objects even when multiple objects are present in a scene. What processes enable the learning to proceed for individual objects rather than for the combinations of objects that may be present during training (Spratling 2005)? In this article we have made important advances to an approach to this that relies on the statistics of natural environments (Stringer and Rolls 2007). In particular, the features of a given object tend to co-occur with high probability with each other during training, whereas because during training this object may be seen with different objects on different occasions, the features of the first object are seen with lower probability with the features of each of these other objects. In this article we have shown that competitive networks operate usefully in this scenario when three objects may be present at a time during learning. That is, the networks learn primarily to form representations that reflect the high probability of co-occurrence of features from one object and do not reflect the features of 2 other objects presented simultaneously during training if the object being trained is seen much more frequently than it is presented with any one other object. This occurs when the number of objects in the training set is greater than approximately 20. Moreover, we show that the transition from representing pairs of objects to representing triples of objects occurs with a lower number of training objects, for example when 4 are used. Thus we show here that as the number of objects in the training set is increased, the network learns to form representations of first high-order combinations of the objects, then of low-order combinations of the objects, and then of individual objects. This leads us to conjecture that the concept will also scale up when the number of objects presented simultaneously is four or more, although the transition points are unknown.

The simulations described with three objects presented during training exhibited richer behavior that with two objects as previously investigated by Stringer and Rolls (2007). In the present simulations with three objects present, it was found that the network tended to develop a more heterogeneous population of output cells, which had learned to respond to a mixture of different numbers of objects. For example, for $N = 20$ there was a mixture of output neurons that had learned to respond to either 1 or 2 objects. It was only by raising $N$ to the more extreme value of 50 that the output neurons responding to pairs of objects were eliminated to leave only neurons responding to individual stimuli. Furthermore, there were no clear cut values of $N$ at which the behavior of the network switched from representing object-triples to object-pairs, or from representing object-pairs to individual objects. Instead, the behavior of the network appeared to change more gradually with $N$, with the distribution of output neurons responding to triples, pairs and singles changing more smoothly. These results are helpful for understanding how neurons at different stages of the ventral visual system in the brain might develop to encode

different combinations of visual features or objects from the visual environment. Firstly, at any stage we might anticipate a somewhat heterogeneous range of neurons which respond to different numbers of visual features or objects. Secondly, the numbers of the features which are represented by neurons at each stage of the visual system will depend on the size of the receptive fields at that stage, and the relative frequencies with which visual features or objects are combined in the real world during learning over the receptive field size. It will be of interest to investigate this further in future research, in which hierarchical models of the ventral visual processing stream will be trained using video input from the real world, as well as visual input from computer-generated 3D virtual reality environments.

If we consider further the effects of receptive field size at each stage of the ventral visual stream, neurophysiological data suggest that the mechanism described here works to separate out feature combinations of such size that only a relatively small number of such feature combinations are typically co-present during training, say of order 5–10. For example, cells in V1 learn to represent edges and bars. However, the very small receptive field sizes of V1 neurons mean that only a small number of edges may be seen together. Similarly, neurons in inferotemporal cortex (IT) receive inputs from across the retina. However, the neural representations which develop in IT are at the object (or face) level, and again only a limited number of objects will be seen together in a natural scene. We further note that in complex, cluttered, natural scenes, with multiple objects present, the receptive fields of inferior temporal cortex neurons shrink to be sufficiently large to include only a few (perhaps of order 5) objects (Rolls et al. 2003; Aggelopoulos and Rolls 2005). The implication of this is that the mechanisms described in this paper might not need to operate with more than of order 5–10 at most objects present simultaneously, and we have already shown that the concept can operate with up to 3 objects present simultaneously.

In this article, we not only showed in single layer networks that trace learning can operate to build invariant representations of individual objects when multiple objects are presented during training, but also extended the results to a 4-layer model of hierarchical processing in the ventral visual system, VisNet. This is the first time that trace learning with multiple objects present during training has been tested with VisNet, and the results are consistent with the hypothesis that the ventral visual system in the brain could use this method to learn invariant representations of individual objects presented in complex scenes with multiple objects present at any one time. We further note that this is the first time that VisNet has been trained with multiple objects present during training in a paradigm requiring translation invariance learning, and in a paradigm which moreover has multiple objects overlapping in every location. We note that all this is achieved without supervised learning, in a purely feedforward network that utilizes the interesting statistics of objects in natural scenes to separate out individual objects from other objects present sometimes in the scene.

The effects described here could operate at every level of a feature hierarchy network model of the ventral cortical processing stream for object recognition. This means that potentially every layer of a hierarchical neural network like the ventral visual system may perform the same operations of separating independent inputs and developing transform invariant representations of those inputs. The kinds of stimuli represented would depend on which stage the layer was in the hierarchy. In the early layers of the network, invariance learning may occur for simple visual

features such as corners. In the later layers, which receive inputs from across the retina, invariance learning may occur for complete stimuli such as objects including faces. The learning mechanisms described in this article may thus operate in every layer of the hierarchy.



## Acknowledgements

This research was supported by the Wellcome Trust.

## References

Aggelopoulos NC, Rolls ET. 2005. Natural scene perception: Inferior temporal cortex neurons encode the positions of different objects in the scene. Eur J Neurosci 22:2903–2916.

Bartlett MS, Sejnowski TJ. 1998. Learning viewpoint-invariant face representations from visual experience in an attractor network. Network: Comput Neural Syst 9:399–417.

Booth MCA, Rolls ET. 1998. View-invariant representations of familiar objects by neurons in the inferior temporal visual cortex. Cerebral Cortex 8:510–523.

Brown TH, Kairiss EW, Keenan CL. 1990. Hebbian synapses: Biophysical mechanisms and algorithms. Ann Rev Neurosci 13:475–511.

Desimone R. 1991. Face-selective cells in the temporal cortex of monkeys. J Cog Neurosci 3:1–8.

Elliffe MCM, Rolls ET, Stringer SM. 2002. Invariant recognition of feature combinations in the visual system. Biol Cybern 86:59–71.

Foldiak P. 1991. Learning invariance from transformation sequences. Neural Comput 3:194–200.

Fukushima K. 1980. Neocognitron: A self–organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol Cybern 36:193–202.

Fukushima K. 1991. Neural networks for visual pattern recognition. IEEE Trans E 74:179–190.

Hasselmo ME, Rolls ET, Baylis GC, Nalwa V. 1989. Object-centered encoding by face-selective neurons in the cortex in the superior temporal sulcus of the monkey. Exp Brain Res 75:417–429.

Hertz J, Krogh A, Palmer RG. 1991. Introduction to the theory of neural computation. Wokingham, UK: Addison Wesley.

Ito M, Tamura H, Fujita I, Tanaka K. 1995. Size and position invariance of neuronal response in monkey inferotemporal cortex. J Neurophysiol 73:218–226.

Kobotake E, Tanaka K. 1994. Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex. J Neurophysiol 71:856–867.

Levy WB. 1985. Associative changes in the synapse: LTP in the hippocampus. In: Levy WB, Anderson JA, Lehmkuhle S, editors. Synaptic modification, neuron selectivity, and nervous system organization. Chapter 1. Hillsdale, NJ: Erlbaum. pp 5–33.

Levy WB, Desmond NL. 1985. The rules of elemental synaptic plasticity. In: Levy WB, Anderson JA, Lehmkuhle S, editors. Synaptic modification, neuron selectivity, and nervous system organization. Chapter 6. Hillsdale, NJ: Erlbaum. pp 105–121.

Logothetis NK, Pauls J, Bulthoff HH, Poggio T. 1994. View dependent object recognition by monkeys. Curr Biol 4:401–414.

Oja E. 1982. A simplified neuron model as a principal component analyser. J Math Biol 15:267–273.

Op de Beeck H, Vogels R. 2000. Spatial sensitivity of macaque inferior temporal neurons. J Comp Neurol 426:505–518.

Perrett DI, Rolls ET, Caan W. 1982. Visual neurons responsive to faces in the monkey temporal cortex. Exp Brain Res 47:329–342.

Perrett DI, Oram MW, Harries MH, Bevan R, Hietanen JK, Benson PJ. 1991. Viewer-centered and object centered coding of heads in the macaque temporal cortex. Exp Brain Res 86:159–173.

Perry G, Rolls ET, Stringer SM. 2006. Spatial vs temporal continuity in view invariant visual object recognition learning. Vision Res 46:3994–4006.

Riesenhuber M, Poggio T. 1999. Hierarchical models of object recognition in cortex. Nature Neurosci 2:1019–1025.

Riesenhuber M, Poggio T. 2000. Models of object recognition. Nature Neurosci Suppl 3:1199–1204.

Rolls ET. 1992. Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical visual areas. Philosophical Trans R Soc 335:11–21.

Rolls ET. 2000. Functions of the primate temporal lobe cortical visual areas in invariant visual object and face recognition. Neuron 27:205–218.

Rolls ET. 2007. The representation of information about faces in the temporal and frontal lobes of primates including humans. Neuropsychologia 45:124–143.

Rolls ET. 2008. Memory, attention, and decision-making. Oxford: Oxford University Press.

Rolls ET, Baylis GC. 1986. Size and contrast have only small effects on the responses to faces of neurons in the cortex of the superior temporal sulcus of the monkey. Exp Brain Res 65:38–48.

Rolls ET, Deco G. 2002. Computational neuroscience of vision. Oxford: Oxford University Press.

Rolls ET, 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 Comp 12:2547–2572.

Rolls ET, Stringer SM. 2001. Invariant object recognition in the visual system with error correction and temporal difference learning. Network: Comput Neural Syst 12:111–129.

Rolls ET, Stringer SM. 2006. Invariant visual object recognition: A model with lighting invariance, J Physiology – Paris 100:43–62.

Rolls ET, Treves A. 1990. The relative advantages of sparse versus distributed encoding for associative neuronal networks in the brain. Network 1:407–421.

Rolls ET, Treves A. 1998. Neural networks and brain function. Oxford: Oxford University Press.

Rolls ET, Treves A, Tovee MJ. 1997. The representational capacity of the distributed encoding of information provided by populations of neurons in the primate temporal visual cortex. Exp Brain Res 114:149–162.

Rolls ET, Aggelopoulos NC, Zheng F. 2003. The receptive fields of inferior temporal cortex neurons in natural scenes. J Neurosci 23:339–348.

Rolls ET, Treves A, Tovee M, Panzeri S. 1997. Information in the neuronal representation of individual stimuli in the primate temporal visual cortex. J Comput Neurosci 4:309–333.

Spratling MW. 2005. Learning viewpoint invariant perceptual representations from cluttered images. IEEE Trans Pat Anal Mach Intel 27:753–761.

Stringer SM and Rolls ET. 2007. Learning transform invariant object recognition in the visual system with multiple stimuli present during training. Neural Networks (submitted).

Stringer SM, Perry G, Rolls ET, Proske JH. 2006. Learning invariant object recognition in the visual system with continuous transformations. Biol Cybern 94:128–142.

Tanaka K, Saito H, Fukada Y, Moriya M. 1991. Coding visual images of objects in the inferotemporal cortex of the macaque monkey. J Neurophysiol 66:170–189.

Tovee MJ, Rolls ET, Azzopardi P. 1994. Translation invariance and the responses of neurons in the temporal visual cortical areas of primates. J Neurophysiol 72:1049–1060.

Wallis G, Rolls ET. 1997. Invariant face and object recognition in the visual system. Progress in Neurobiol 51:167–194.