Cognitive Processing, doi: 10.1007/s10339-016-0765-6

Predictive Coding as a Model of Cognition

M. W. Spratling
King's College London, Department of Informatics, London. UK.

Abstract

Previous work has shown that predictive coding can provide a detailed explanation of a very wide range of low-level perceptual processes. It is also widely believed that predictive coding can account for high-level, cognitive, abilities. This article provides support for this view by showing that predictive coding can simulate phenomena such as categorisation, the influence of abstract knowledge on perception, recall and reasoning about conceptual knowledge, context-dependent behavioural control, and naive physics. The particular implementation of predictive coding used here (PC/BC-DIM) has previously been used to simulate low-level perceptual behaviour and the neural mechanisms that underlie them. This algorithm thus provides a single framework for modelling both perceptual and cognitive brain function.

Keywords: perception; cognition; categorisation; reading; word-superiority effect; reasoning; behavioural control; naive physics

1 Introduction

Predictive coding is a leading theory of brain function (Bubic et al., 2010; Clark, 2013; Huang and Rao, 2011; Rao and Ballard, 1999), that has been shown to explain a great deal of neurophysiological and psychophysical data, such as the information processing performed in the retina and lateral geniculate nucleus (LGN; Hosoya et al., 2005; Jehee and Ballard, 2009; Laughlin, 1990; Srinivasan et al., 1982), orientation tuning, surround suppression and cross-orientation suppression in primary visual cortex (V1; Spratling, 2010, 2011, 2012a), the learning of Gabor-like receptive fields (RFs) in V1 (Jehee et al., 2006; Rao and Ballard, 1999; Spratling, 2012c), gain modulation as is observed, for example, when a retinal RF is modulated by eye position (De Meyer and Spratling, 2011, 2013), binocular rivalry (Denison et al., 2011; Hohwy et al., 2008), contour integration (Spratling, 2013b, 2014c), the modulation of neural response due to attention (Spratling, 2008a, 2014c), fMRI data related to stimulus expectation (Alink et al., 2010; Egner et al., 2010; Smith and Muckli, 2010; Summerfield and Egner, 2009), mismatch negativity (Garrido et al., 2009; Wacongne et al., 2012), habituation (Ramaswami, 2014), and the saliency of visual stimuli (Spratling, 2012b). As the preceding list illustrates, the predictive coding framework has been exceptionally successful at explaining low-level perceptual abilities and the neural processes that underlie them. It has been claimed that predictive coding can also account for higher-level cognitive processes like theory of mind (Koster-Hale and Saxe, 2013), mirror neurons (Kilner et al., 2007), emotions (Seth, 2013), aesthetics (de Cruys and Wagemans, 2011), self-awareness (Apps and Tsakiris, 2014), consciousness (Seth et al., 2011), and disorders of cognitive function such as schizophrenia (Lalanne et al., 2010) and autism (Lawson et al., 2014; van Boxtel and Lu, 2013). However, these claims are rather speculative consisting of verbal theories rather than explicit simulations.

This article makes an initial step in demonstrating that predictive coding can model cognition. As cognition is underpinned by the formation of conceptual knowledge, it is first shown that predictive coding can categorise perceptual information (section 3.1). The model is shown to successfully simulate a number of experiments exploring human categorisation behaviour. It is then shown that predictive coding can simulate the influence of higher-level knowledge on perception (section 3.2). Specifically, the model is used to simulate the influence of word knowledge on letter perception, and the results are shown to be consistent with a range of results obtained with human subjects. Predictive coding is then shown to be able to reason about conceptual knowledge (section 3.3), and to be able to perform context-dependent task switching (section 3.4). Finally, it is shown that predictive coding can perform reasoning about simple physics problems (section 3.5). Specifically, the model is used to infer the relative mass of objects from observations of the velocities before and after a collision. These simulation results are also in good agreement with human behavioural data.

Computational models exist which successfully explain all the phenomena simulated in this article. For example, there are numerous models of categorisation that can simulate some or all of the experiments discussed in section 3.1 (*e.g.*, Aha and Goldstone, 1992; Anderson, 1991; Anderson and Betz, 2001; Erickson and Kruschke, 1998; Kruschke, 1992; Love et al., 2004; Medin and Schaffer, 1978; Nosofsky and Johansen, 2000; Nosofsky et al., 1994; Sanborn et al., 2010). The IAC model (McClelland, 2014; McClelland and Rumelhart, 1981; Rumelhart and McClelland, 1982; Rumelhart et al., 1986) has previously been used to explain the influence of word knowledge on letter perception (as modelled in section 3.2), and to simulate recall and reasoning with the Jets and Sharks data (section 3.3). Salinas (2004a,b) has proposed a gain-modulated basis function network to model the context-dependent task switching experiments discussed in section 3.4. Intuitive understanding about collision physics, section 3.5, has been simulated by the "noisy Newton" model (Sanborn, 2014; Sanborn et al., 2013). In addition to these models of specific cognitive abilities, there also exists general-purpose modelling frameworks, such as connectionism, Bayesian modelling[1], and ACT-R (Anderson et al., 2004), all of which can simulate some or all of the experiments described here, and many others besides. This article does not therefore aim to show that the form of predictive coding studied here is unique in its ability to simulate the chosen cognitive tasks, nor to claim that the proposed framework for modelling cognition is better than existing ones. Instead, by showing that predictive coding can be used to build models of a diverse range of cognitive phenomena, the aim is to provide the first proof-of-concept demonstration that predictive coding can model cognition, and to establish predictive coding as a potential alternative method for constructing models of cognition. In the current article, the models have been constructed by hand with hard-coded, rather than learnt, synaptic weights. Such hand-designed networks suffice to demonstrate that predictive coding can perform cognitive tasks, but learning remains an important topic for future work.

2 Methods

All the simulations reported here are performed using a particular implementation of predictive coding called the PC/BC-DIM algorithm. PC/BC-DIM is a version of Predictive Coding (PC; Rao and Ballard, 1999) reformulated to make it compatible with Biased Competition (BC) theories of cortical function (Spratling, 2008a,b), and that is implemented using Divisive Input Modulation (DIM; Spratling et al., 2009) as the method for updating error and prediction neuron activations. DIM calculates reconstruction errors using division, which is in contrast to other implementations of PC that calculate reconstruction errors using subtraction (Huang and Rao, 2011).

PC/BC-DIM is a hierarchical neural network model. A single processing stage in a PC/BC-DIM hierarchy is illustrated in Fig. 1 and implemented using the following equations:

$$\mathbf{r} = \mathbf{V}\mathbf{y} \tag{1}$$

$$\mathbf{e} = \mathbf{x} \oslash [\mathbf{r}]_{\epsilon_2} \tag{2}$$

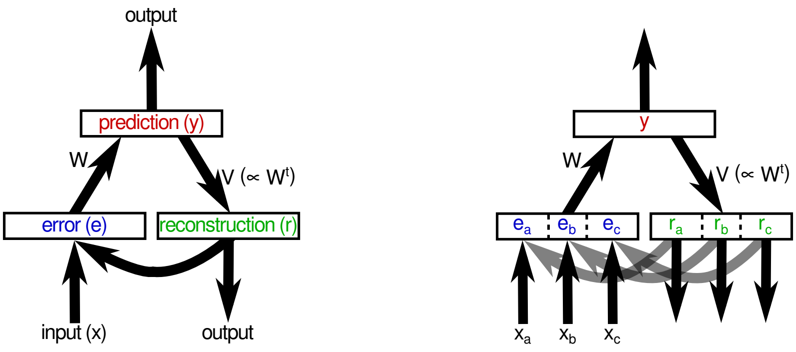
$$\mathbf{y} \leftarrow [\mathbf{y}]_{\epsilon_1} \otimes \mathbf{W}\mathbf{e} \tag{3}$$

Where $\mathbf{x}$ is a (m by 1) vector of input activations, $\mathbf{e}$ is a (m by 1) vector of error neuron activations; $\mathbf{r}$ is a (m by 1) vector of reconstruction neuron activations; $\mathbf{y}$ is a (n by 1) vector of prediction neuron activations; $\mathbf{W}$ is a (n by m) matrix of feedforward synaptic weight values; $\mathbf{V}$ is a (m by n) matrix of feedback synaptic weight values; $[v]_\epsilon = \mathtt{max}(\epsilon, v)$; $\epsilon_1 = 1 \times 10^{-6}$ and $\epsilon_2 = 1 \times 10^{-3}$ are parameters; and $\oslash$ and $\otimes$ indicate element-wise division and multiplication respectively. The matrix $\mathbf{V}$ is equal to the transpose of the $\mathbf{W}$, but each column is normalised to have a maximum value of one. Hence, the feedforward and feedback weights are simply rescaled versions of each other.

Initially the values of $\mathbf{y}$ are all set to zero, although random initialisation of the prediction neuron activations can also be used with little influence on the results. Equations 1, 2 and 3 are then iteratively updated with the new values of $\mathbf{y}$ calculated by equation 3 substituted into equation 1 and 3 to recursively calculate the neural activations. This iterative process was terminated after 75 iterations in all the experiments reported here.

The values of $\mathbf{y}$ represent predictions of the causes underlying the inputs to the network. The values of $\mathbf{r}$ represent the expected inputs given the predicted causes. The values of $\mathbf{e}$ represent the residual error between the reconstruction, $\mathbf{r}$, and the actual input, $\mathbf{x}$. The full range of possible causes that the network can represent are defined by the weights, $\mathbf{W}$ (and $\mathbf{V}$). Each row of $\mathbf{W}$ (which correspond to the weights targeting an individual prediction neuron) can be thought of as a "basis vector" or "elementary component" or "preferred stimulus", and $\mathbf{W}$ as a whole can be thought of as a "dictionary" or "codebook" of possible representations, or a model of the external environment. The activation dynamics described above result in the PC/BC-DIM algorithm selecting a subset of active prediction neurons whose RFs (which correspond to basis functions) best explain the underlying causes of the sensory input. The strength of activation reflects the strength with which each basis function is required to be present in order to accurately reconstruct the input. This strength of response also reflects the probability with which that basis function (the preferred stimulus of the active prediction neuron) is believed to be

[1] It is widely believed that predictive coding also emulates Bayesian inference (Deneve, 2008; Friston, 2005; Kilner et al., 2007; Lee, 2015; Spratling, 2016), hence, it should be expected that predictive coding can simulate cognitive function, however, as noted earlier this has yet to be demonstrated explicitly.



(a) (b)

Figure 1: (a) A single processing stage in the PC/BC-DIM neural network architecture. Rectangles represent populations of neurons and arrows represent connections between those populations. The population of prediction neurons constitute a model of the input environment. Individual neurons represent distinct causes that can underlie the input (*i.e.*, latent variables). The belief that each cause explains the current input is encoded in the activation level, $\mathbf{y}$, and is used to reconstruct the expected input given the predicted causes. This reconstruction, $\mathbf{r}$, is calculated using a linear generative model (see equation 1). Each column of the feedback weight matrix $\mathbf{V}$ represents an "elementary component", "basis vector", or "dictionary element", and the reconstruction is thus a linear combination of those components. Each element of the reconstruction is compared to the corresponding element of the actual input, $\mathbf{x}$, in order to calculate the residual error, $\mathbf{e}$, between the predicted input and the actual input (see equation 2). The errors are subsequently used to update the predictions (via the feedforward weights $\mathbf{W}$, see equation 3) in order to make them better able to account for the input, and hence, to reduce the error at subsequent iterations. The responses of the neurons in all three populations are updated iteratively to recursively calculate the values of $\mathbf{y}$, $\mathbf{r}$, and $\mathbf{e}$. The weights $\mathbf{V}$ are the transpose of the weights $\mathbf{W}$, but are normalised to so that the maximum value of each column is unity. Given that the $\mathbf{V}$ weights are fixed to the $\mathbf{W}$ weights there is only one set of free parameters, $\mathbf{W}$. References in the main text to the synaptic weights refer to the elements of $\mathbf{W}$. The activations of the prediction neurons or the reconstruction neurons may be used as inputs to other PC/BC-DIM processing stages. The inputs to this processing stage may come from the prediction neurons of this or another processing stage, or the reconstruction neurons of another processing stage, or may be external, sensory-driven, signals. The inputs can also be a combination of any of the above. (b) When inputs come from multiple sources, it is sometimes convenient to consider the population of error neurons to be partitioned into sub-populations which receive these separate sources of input. As there is a one-to-one correspondence between error neurons and reconstruction neurons, this means that the reconstruction neuron population can be partitioned similarly.

present, taking into account the evidence provided by the input signal and the full range of alternative explanations encoded in the RFs of the whole population of prediction neurons.

A simple two-stage hierarchical PC/BC-DIM network is illustrated in Fig. 7. The recurrent inputs provided by the reconstruction neurons in the second processing stage are used to provide top-down inputs to the first processing stage. Additional synaptic weights need to be defined to allow the recurrent inputs to influence the first-stage prediction neuron responses. These additional weights will form additional columns of $\mathbf{W}$ (and additional rows of $\mathbf{V}$). The exact form of the top-down weights will determine how the recurrent inputs affect the behaviour of the model. To perform simulations with a hierarchical model equations 1, 2 and 3 are evaluated for each processing stage in turn (starting from the lowest stage in the hierarchy), and this process is repeated to iteratively calculate the changing neural activations in each processing stage at each time-step.

In the hierarchical PC/BC-DIM network illustrated in Fig. 7 the first processing stage receives input from two distinct sources, the image and the second-stage reconstruction neurons. There are many other situations in which inputs may come from multiple sources. For example, in section 3.1 inputs come from image pixels and class labels (see Fig. 2b), and in sections 3.4 and 3.5 different parts of the input vector encode the values of different variables. When inputs come from multiple sources it is sometimes convenient to consider the input vector to be partitioned into separate sub-vectors representing these separate sources. Since there is a one-to-one correspondence between elements of the input vector and both error neurons and reconstruction neurons it is also possible to think of these neural populations as being partitioned into sub-populations (Fig. 1b). Each partition of the input will correspond to certain columns of $\mathbf{W}$ (and rows of $\mathbf{V}$). While it is conceptually convenient to think about separate partitions of the inputs, neural populations and synaptic weights (and to sometimes plot these values in separate sub-graphs), it does not in any way alter the mathematics of the model. In equations 1, 2 and 3, $\mathbf{x}$ is a concatenation of all partitions of the input, $\mathbf{e}$ and $\mathbf{r}$ represent the activations of all the error and reconstruction neurons; and $\mathbf{W}$ and $\mathbf{V}$ represent the synaptic weight values for all partitions.

Open-source software, written in MATLAB, which performs the experiments described in this article is available for download from: http://www.corinet.org/mike/Code/pcbc_cognition.zip.

3 Results

This section reports the results of applying the PC/BC-DIM algorithm to simulating a diverse range of tasks: categorisation (section 3.1), the influence of higher-level knowledge on perception (section 3.2), reasoning about conceptual knowledge (section 3.3), context-dependent task switching (section 3.4), and reasoning about collisions between objects (section 3.5). While the PC/BC-DIM algorithm (as described in section 2) stays the same throughout, the inputs to (and outputs from) the PC/BC-DIM network, the number of neurons in each population, and the synaptic weights, change from tasks to task. These details are therefore provided in the text that describes each experiment. Any missing implementation details can be found in the accompanying code (see section 2). Furthermore, to provide the reader with a visual summary of the structure and size of each network, a diagram accompanies each set of results. These summary diagrams appear within a box (like those shown on the left of Fig. 3). Given that there is a one-to-one correspondence between the neurons in the error and reconstruction populations, it is possible to draw a PC/BC-DIM network in a compact form with the error and reconstruction neurons superimposed. Each summary diagram has this more compact form, but otherwise uses the same conventions as described in the caption of Figure 1. The numbers above each neural population indicate the number of neurons in that population. When the error and reconstruction neurons are split into multiple partitions, the number of neurons in each partition is indicated.

3.1 Perceptual Classification and Categorization

Classification and categorisation are fundamental to concept formation and human cognition (Goldstone and Kersten, 2003; Kruschke, 2005). It is straightforward to set-up a PC/BC-DIM network to recognise stimuli: by defining appropriate synaptic weights the prediction neurons will respond to specific patterns of input. However, to perform categorisation it is typically necessary to be able to generalise over changes in appearance, or to be able to group together perceptually dissimilar stimuli. If a subset of prediction neurons are defined to represent a range of stimuli falling within the same category, then it is necessary to “pool” the responses from this subset of prediction neurons to define a neuron that will responds to all members of the category. Figure 2 shows two ways in which this can be implemented. The first method (Fig. 2a) involves using a separate population of pooling neurons that are activated by the responses of the prediction neurons. This method has been used in previous work (Spratling, 2014a) and is consistent with (radial) basis function neural networks (Broomhead and Lowe, 1988; Kruschke, 1992; Pouget and Sejnowski, 1997) and with several hierarchical models of invariant object recognition which employ alternating layers of neurons to form more specialised representations in one layer, and more invariant representations in the next layer (Fukushima, 1980; LeCun et al., 2010; Riesenhuber and Poggio, 1999; Serre et al., 2007). The second method (Fig. 2b) involves defining additional neurons within the reconstruction neuron population that perform the same role as the pooling neurons in the first method. In this article the second method will be used. It has the advantage that: (1) it is slightly simpler to implement, as it is not necessary to introduce a new population of neurons governed by new equations; (2) it suggests that a single algorithm could be used to learn both the perceptual features and the classification labels when both are available as sensory-driven inputs to the network; and (3) if information about the expected class of the stimulus is available during recognition (for example, if there are inputs to the class partition of the input vector that are proportional in strength to the probability of each class), then this information can be combined with any available featural information to infer the best estimate of the class (*i.e.*, to perform cue combination; Spratling, 2016). More generally, while the first method (Fig. 2a) produces a uni-directional mapping from inputs to outputs, the proposed architecture (Fig. 2b) allows omni-directional mappings between the variables encoded by different partitions of the input and reconstruction neurons, removing the distinction between variables that are inputs and those that are outputs.

3.1.1 Image Recognition

Figure 3 shows two examples of using PC/BC-DIM to perform classification of images from standard datasets used in machine learning. To perform these experiments, each prediction neuron was given $\mathbf{W}_a$ weights (see Fig. 2b) equal to the pixel intensity values of one image from the training set, and $\mathbf{W}_b$ weights (see Fig. 2b) that

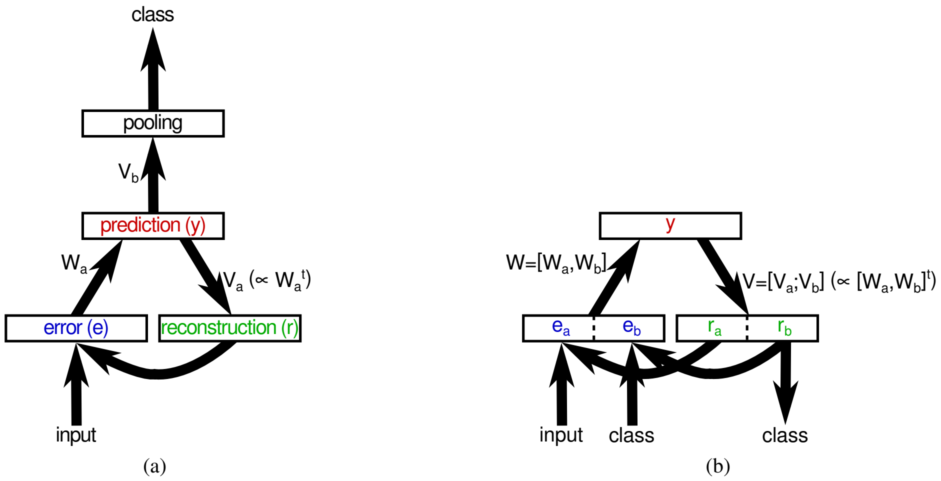


Figure 2: Methods of performing classification with a PC/BC-DIM network. The prediction neurons have RFs (defined by the $\mathbf{W}_a$ weights) that make them selective to specific input stimuli. (a) A population of pooling neurons receives input, via weights $\mathbf{V}_b$, from the prediction neurons. Each pooling neuron represents a class and receives non-zero weights from all prediction neurons that are selective to stimuli within that class. The responses of the pooling neurons, $\mathbf{z}$, are calculated as a linear weighted sum of their input, *i.e.*, $\mathbf{z} = \mathbf{V}_b\mathbf{y}$. (b) The PC/BC-DIM network receives two sources of input, one from the features of the stimuli that are to be categorised, and another source of input defining the corresponding class labels. For convenience, we can regard the vector of input signals, $\mathbf{x}$, the vector of error neuron activations, $\mathbf{e}$, and the vector of reconstruction neuron responses, $\mathbf{r}$, to be partitioned into two parts corresponding to these separate sources of input (see Fig. 1b). Dealing with the extra inputs requires the definition of additional columns of feedforward synaptic weights and additional rows of the feedback weights. The additional feedforward weights will be non-zero between the input representing a specific class label and all the prediction neurons that are selective to stimuli within that class. As with the $\mathbf{W}_a$ and $\mathbf{V}_a$ weights the additional feedback weights, $\mathbf{V}_b$, are rescaled versions of the corresponding additional feedforward weights, $\mathbf{W}_b$. Hence, a reconstruction neuron in the second partition will receive non-zero weights from all prediction neurons that are selective to stimuli within a single class. Each second partition reconstruction neuron will thus represent a class. Given the definition of the reconstruction neuron responses (see equation 1), it can be seen that the responses of the second partition of the reconstruction neurons, $\mathbf{r}_b$, will be identical to the responses of the pooling neurons in (a), *i.e.*, $\mathbf{r}_b = \mathbf{V}_b\mathbf{y}$. During classification only the features of the stimulus that is to be categorised would be presented as input (to the first partition), the second partition of the input would be blank, and the network's prediction of the class label would be read out from the responses of the reconstruction neurons in the second partition.

encoded the class label of that training image. There were thus as many prediction neurons as training images. It is also possible to learn prediction neuron weights from training images so that there are fewer prediction neurons than training images. However, the straight-forward, non-learning, method used here suffices to demonstrate that PC/BC-DIM can perform real-world image classification.

The first dataset used was the USPS hand-written digits dataset (Hull, 1994). This consists of 16-by-16 pixel greyscale images of hand-written digits divided into a training set containing 7291 images and a test set contains 2007 images. Figure 3a and b show the behaviour of the network in response to the presentation of two images from the test set. The predicted class label is defined by the maximum response of the second-partition reconstruction neurons (*i.e.*, the label of the highest bar in the histogram shown at the top-right of each sub-figure of Fig. 3). The PC/BC-DIM network's prediction of the class label was compared to the true class label for all 2007 images in the test set and was found to be correct in 94.8% of cases (this compares to 97.4% accuracy for human subjects (Chaaban and Scheessele, 2007)).

The second dataset used was the cropped and aligned version of the Extended Yale Face Database B (Georghiades et al., 2001; Lee et al., 2005). This contains images of faces taken under varying lighting conditions. There are approximately 64 images for each of 38 individuals. Half the images for each individual were used for training (*i.e.*, setting the weights) and the other half for testing. Images were downsampled from the 168-by-192 pixel originals to 48-by-42 pixels. Figure 3d and e show the behaviour of the network in response to the presentation of two images from the test set. Across all images in the test set the PC/BC-DIM network's prediction of the identity

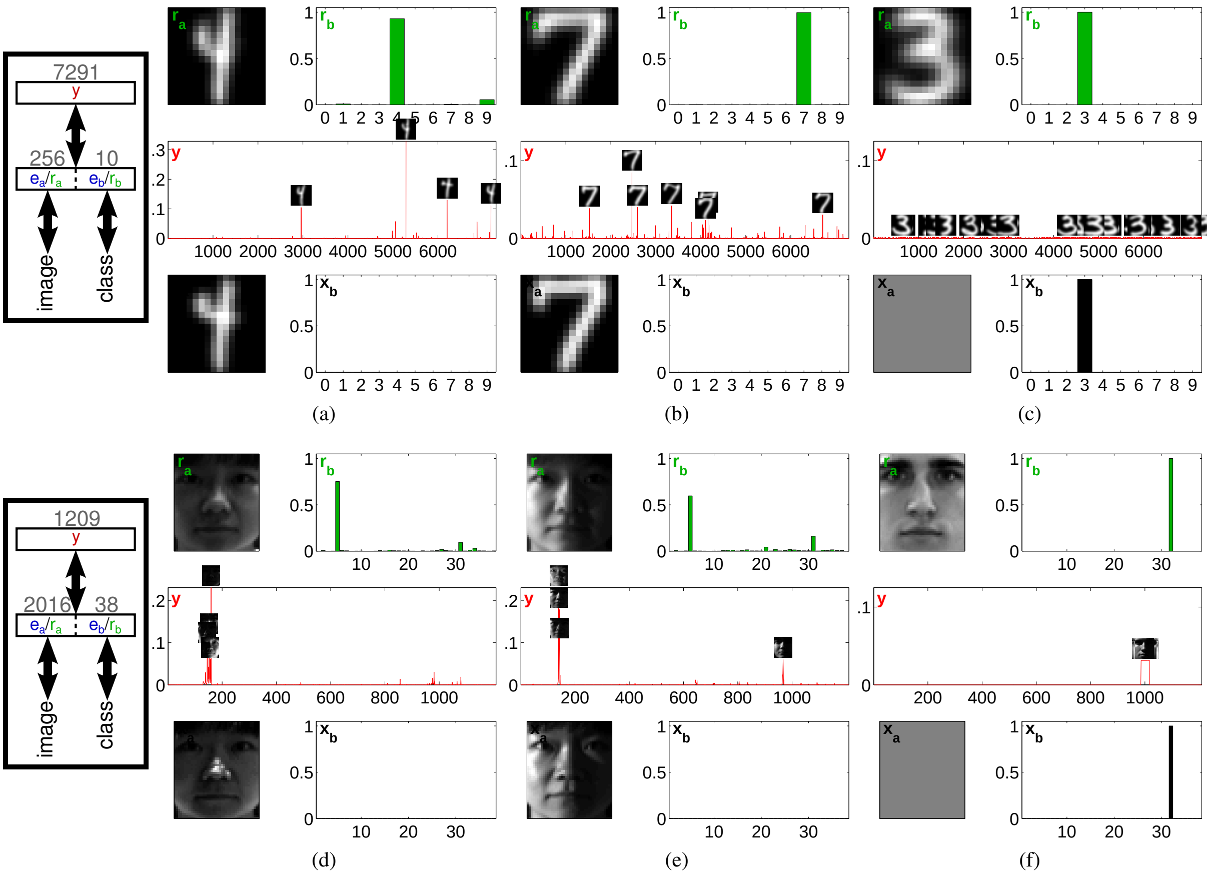


Figure 3: Classification of images. In each sub-figure the inputs to the PC/BC-DIM network are shown at the bottom. These inputs come from two sources (see Fig. 2b): an image (values shown as a 2D array of intensity values) and an array of category labels (values shown as a histogram). The prediction neuron responses are shown in the middle histogram (the x-axis represents neuron number and the y-axis represents response strength). The $\mathbf{W}_a$ weights (see Fig. 2b) of the most active prediction neurons are indicated by the images superimposed on the middle histograms. The reconstruction neuron responses are shown at the top. As for the inputs, there are two partitions. The first reconstructs the input image (shown as a 2D array of intensity values) and the second partition represents the predicted class label (shown as a histogram, where the x-axis represents class label and the y-axis represent response strength). The input images come from (a-c) the USPS hand-written digits dataset, and (d-f) the cropped and aligned version of the Extended Yale Face Database B (Georghiades et al., 2001; Lee et al., 2005). In both cases prediction neurons have been wired-up so that the $\mathbf{W}_a$ weights (the RFs corresponding to the first partition of the input) are equal to images from the training set, and the $\mathbf{W}_b$ weights (corresponding to the second partition) encode the class label of that training image. The first two columns show results when a novel image from the test set is presented to the network. In each case the network successfully predicts the correct class for this image. The last column shows results when no image is presented to the network, but a a class label is presented. In this case the reconstruction neuron responses represent the average image in that class.

of the individual was found to be correct 96.4% of the time (this compares to 91.0% accuracy for a k-nearest neighbours classifier, with a value of k optimised for the two datasets used here).

The previous paragraph describes experiments in which an image is presented to the PC/BC-DIM network, and the class label is read out from the second partition of reconstruction neurons. It is also possible to input a class label, and have the network reconstruct (in the first partition) an image corresponding to this class. For example, figure 3c shows the typical member of the class "number three" generated by the network wired up to represent the USPS dataset, and figure 3f shows the reconstruction of the face of one individual in the Yale Face dataset.

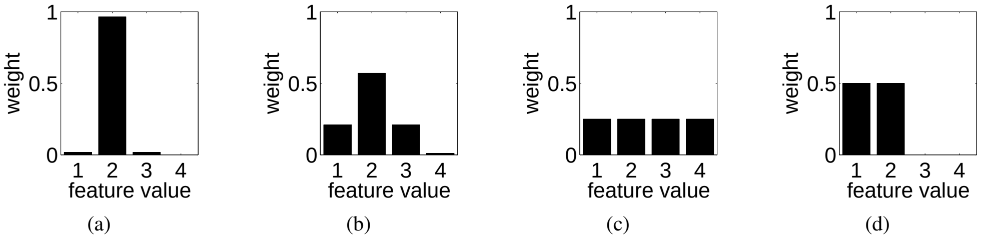


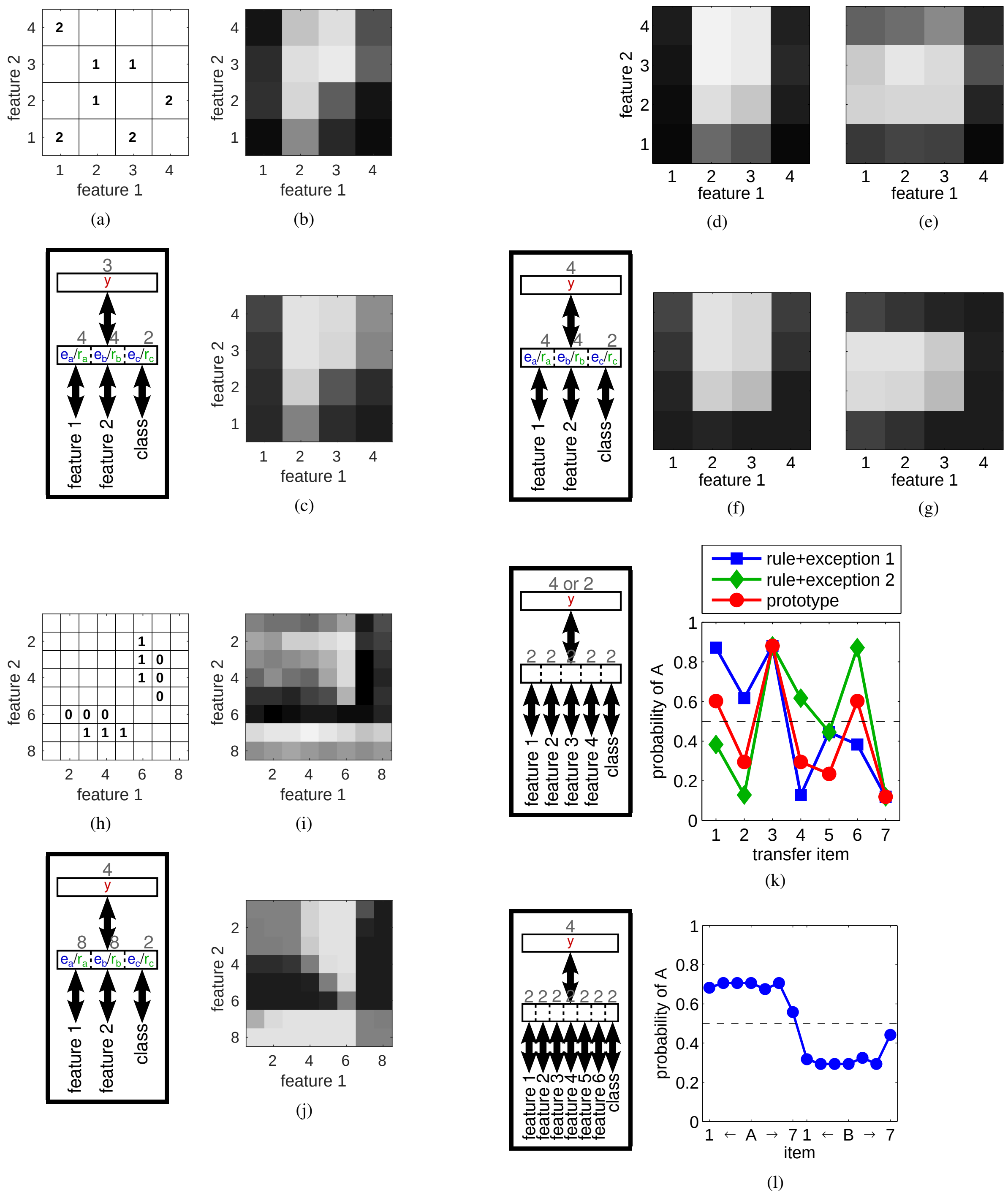
Figure 4: Examples of synaptic weights that could be used to define the RF of a prediction neuron to a single perceptual feature. (a) Weights that are highly selective for a particular feature value. (b) Weights that are selective for the same feature value but less narrowly tuned. (c) Weights that are uniformly distributed over all possible feature values, and hence, unselective for any particular value. (d) Weights that are equally selective for a sub-range of feature values.

3.1.2 Simulating Human Categorisation Experiments

Humans appear to employ a number of different strategies for performing classification, including rule-based, exemplar-based, and prototype-based strategies (Goldstone and Kersten, 2003). Different strategies are used by the same individual in different circumstances, and by different individuals on the same categorisation task. Furthermore, people seem capable of combining multiple strategies to perform a single categorisation task (Goldstone and Kersten, 2003). PC/BC-DIM is sufficiently flexible to be able to implement rule-based, exemplar-based, and prototype-based methods of classification, and to employ combinations of these methods to solve a single task. Previous theories have also proposed that the brain can use multiple categorisation strategies and combinations of strategies (Anderson and Betz, 2001; Ashby et al., 1998; Erickson and Kruschke, 1998; Smith et al., 1998). However, previous models of this type have used hybrid architectures containing multiple, distinct, modules to implement each categorisation method (*e.g.*, Erickson and Kruschke, 1998). In contrast, PC/BC-DIM can implement different categorisation strategies, and combinations of strategies, in a single neural network.

Imagine a set of stimuli that are defined by a single feature dimension which can take one of four values. The four possible feature values might be specific samples taken from a continuous feature space, or might represent distinct feature values in a discontinuous space, the same arguments apply to both cases. A prediction neuron selective to a particular feature value might have synaptic weights that are strongly selective to that value, as illustrated in Fig. 4a. A neuron tuned to the same feature value, but less selectively so, might have synaptic weights like those shown in Fig. 4b with the same mean but higher variance (or less precision). A neuron with no preference for the value of the feature might have uniform weights like those shown in Fig. 4c. A prediction neuron selective to a certain sub-range of possible feature values might have synaptic weights like those shown in Fig. 4d. Similar principles can be used to define weights in any number of feature dimensions, however, for simplicity imagine a set of stimuli defined over a two-dimensional feature space. A prediction neuron selective to a particular exemplar would be given weights that are strongly selective, in both dimensions, to the specific feature values of that exemplar (*i.e.*, weights like those shown in Fig. 4a for both dimensions, but not necessarily centred at the same value in each dimension nor necessarily having the same tuning width in each dimension). A prediction neuron selective to a prototype would have weights that peak at the average feature value in each dimension. However, it might be less narrowly tuned in order to represent a wider range of feature values (*i.e.*, it might have weights like those shown in Fig. 4b for both dimensions, but not necessarily centred at the same value in each dimension nor necessarily having the same tuning width in each dimension). A prediction neuron that implements a rule aligned with one of the feature dimensions, might be strongly selective to a specific feature value or range of values in that dimension, but be unselective to values in the other, rule-irrelevant, dimension (*e.g.*, it would have weights like those shown in Fig. 4a or Fig. 4d in one dimension, but weights like those shown in Fig. 4c in the other). Different prediction neurons can be given different weights. Hence, within the same PC/BC-DIM network different neurons can represent exemplars, prototypes, and rules.

In a classic experiment on human categorisation, Nosofsky et al. (1989) employed stimuli that varied over two dimensions each of which could take one of four values. Training stimuli from the two classes, 1 and 2, had the feature values indicated in Fig. 5a. Once trained on this task, human subjects were tested using all possible combinations of feature values and the probability with which each stimulus was categorised as class 1 is indicated by the lightness of the corresponding square in Fig. 5b. This pattern of results can be accounted for by the PC/BC-DIM model using a network with three prediction neurons. One prediction neuron is wired-up to represent the prototype of class 1: it has weights like those shown in Fig. 4b, but centred at feature values of 2.33 and 2.66 in the first and second dimensions respectively. The second prediction neuron represents the average features of the two



members of class 2 shown in the bottom right-hand corner of Fig. 5a: it has weights like those shown in Fig. 4a, but centred at feature values of 3.5 and 1.5 in the first and second dimensions respectively. This second neuron is intermediate between a neuron that represents a prototype and one that represents an exemplar as it is selective to the average features of a subset of exemplars in a class. This type of representation will be referred to as a "prototype/exemplar". The third prediction neuron implements a rule to place into class 2 those stimuli that have a value of one for feature one: it has narrowly tuned weights like those shown in Fig. 4a centred at a value of one in the first dimension, but is less selective for the value the second dimension, having weights like those shown in Fig. 4b but centred at a value of 2.5. As for the experiments shown in Fig. 3 the network's prediction of a stimulus' class is read out from reconstruction neurons in the last partition. As there are only two classes, only two such neurons are required. The first receives strong weights from the first prediction neuron, and the second receives strong weights from the second and third prediction neurons. For any given stimulus, both the reconstruction

Figure 5: (previous page) Simulation of human categorisation experiments. (a-c) Simulation of Nosofsky et al. (1989, expt. 1): (a) task structure, (b) human data, (c) PC/BC-DIM simulation results. (d-g) Simulation of Nosofsky et al. (1989, expt. 2): (d) human data for rule set 1, (e) human data for rule set 2, (f) PC/BC-DIM simulation results for rule set 1; (g) PC/BC-DIM simulation results for rule set 2. (h-j) Simulation of Aha and Goldstone (1992): (h) task structure, (i) human data, and (j) PC/BC-DIM simulation results. For (b-g), (i), and (j) the lightness of each square is proportional to the probability with which that combination of features was categorised as being in class 1: lightness ranges from black (zero probability of class 1) to white (probability of one that the stimulus is placed in class 1) (k) Simulation of Medin and Schaffer (1978, expt. 2). The graph shows the PC/BC-DIM network's estimate of the probability that each transfer item is in class A. Results are shown for three different sets of synaptic weights. Each pattern of categorisation predicted by the three networks are consistent with a common pattern of classification made by human subjects. (l) Simulation of Smith and Minda (1998, expt. 2). The graph shows for each item in the data set the PC/BC-DIM network's estimate of the probability that that item is in class A. The first seven items are in class A, the next seven are the items in class B. The two "exceptions" (items 7 and 14) are seen by the network as less probably members of their class, consistent with the human classification performance.

neurons that represent class labels may be simultaneously active. A simple measure of how certain the network is that the stimulus should be classified into class 1 was calculated by taking the response of the neuron representing class 1 and dividing this by the sum of the responses of the reconstruction neurons representing both class 1 and class 2. These values, for all possible stimuli, are shown in Fig. 5c and provide a good match to the human data.

In addition to the preceding experiment where subjects learnt the categories, Nosofsky et al. (1989) also performed an experiment where subjects were instructed to allocate an exemplar to class 2 if certain conditions were met. Two different sets of rules were used. The results of this experiment are shown in Fig. 5d for rule set 1 and Fig. 5e for rule set 2. To simulate these results with PC/BC-DIM it was assumed that class 1 was represented in the same way as in the previous experiment (using one neuron tuned to the prototype). Three further neurons were used to implement the three rules in each rule set, using weights analogous to those used to define the third neuron in the previous experiment. The simulation results are shown in Fig. 5f for rule set 1 and Fig. 5g for rule set 2.

Aha and Goldstone (1992) also employed two categories that were defined using a two-dimensional feature space, but where each feature ranged over eight possible values. The feature values of the training exemplars are indicated in Fig. 5h. Following training, human subjects were tested using all possible combinations of feature values and the probability with which each stimulus was categorised as class 1 is shown in Fig. 5i. Similar results are produced by the PC/BC-DIM model (Fig. 5j). This PC/BC-DIM network uses four prediction neurons, each of which is selective for a prototype/exemplar. One neuron, representing class 0, is centred at location (3,6) in feature space. The width of tuning over the first dimension is greater than the width of tuning over the second dimension, reflecting the distribution of the three training exemplars in class 0 that are clustered around this location. The other three prediction neurons have analogous weights to represent the other groups of three training items.

Medin and Schaffer (1978) devised a stimulus set in which stimuli had four features each of which could take one of two values, denoted by 0 or 1. Training stimuli were divided into two sets. Category A consisted of stimuli 1110, 1010, 1011, 1101, and 0111. Category B consisted of stimuli 1100, 0110, 0001, and 0000. Having been trained with these stimuli subjects were asked to classify seven novel transfer stimuli: 1001, 1000, 1111, 0010, 0101, 0011, and 0100. As each transfer stimulus can either be classified as A or B, the pattern of classifications made by a subject can be summarised as a string of As and Bs. The three most common patterns of transfer item categorisation were AAABBBB, BBAABAB, and ABABBAB (Nosofsky and Johansen, 2000; Nosofsky et al., 1994). By defining networks with different weights, PC/BC-DIM can simulate all three of these patterns, as illustrated in Fig. 5k.

Categorising the transfer items as AAABBBB suggests that the subject is employing a strategy in which a stimulus is placed in class A if the value of the first dimension is one, and in class B if the value is zero, but where the fifth item in class A and the first item in class B are treated as exceptions (Nosofsky and Johansen, 2000). This rule+exception strategy can be implemented in PC/BC-DIM using four prediction neurons. The first prediction neuron has weights that are selective for a value of one in the first feature dimension, but has equal weights for both possible values of the other three features, it also has a strong weight for class label A. This neuron implements the rule that if the first feature is a one then the class label is A. The second neuron has weights that represent the exemplar 0111 and class label A, it thus represents the exception for class A. The two other neurons have analogous weights in order to represent the rule that if the first feature is zero the class label is B and the exception to this rule. The network's estimate of the probability that a stimulus is in class A is plotted using square markers in Fig. 5k for all seven transfer items, and the pattern is consistent with the AAABBBB pattern of human subjects.

The second most common pattern of categorising the transfer items (BBAABAB) suggests that the subject is employing a strategy in which a stimulus is placed in class A if the value of the third dimension is one, and in class B if this value is zero, but where the fourth item in class A and the second item in class B are treated as exceptions (Nosofsky and Johansen, 2000). This strategy can be implemented in PC/BC-DIM in a way analogous to the previous strategy, and the results are consistent with the human subjects, see diamond markers in Fig. 5k. The third most common pattern (ABABBAB) can be accounted for by a prototype-based strategy. To simulate this strategy two prediction neurons are used. One represents the prototype of category A and the other the prototype of category B. The predicted category labels for the seven transfer items are plotted using circular markers in Fig. 5k, and the pattern is consistent with the human subjects.

Smith and Minda (1998) employed a two-class stimulus set in which stimuli had six binary-valued dimensions. Category A consisted of the stimuli 000000, 100000, 010000, 001000, 000010, 000001, and 111101. Category B consisted of the stimuli 111111, 011111, 101111, 110111, 111011, 111110, and 000100. Following training subjects were more accurate in classifying the first six items in each class, than the seventh item in each class. This pattern of results is consistent with a classification strategy in which the two classes are represented by the prototypes 000000 for class A, and 111111 for class B, plus two exceptions for the seventh item in both classes. A PC/BC-DIM network, with four prediction neurons, wired-up to implement this strategy produces the results shown in Fig. 5l. Compared to the neurons representing the prototypes, the prediction neurons representing the exemplars (the exceptions) have narrower tuning for perceptual features and/or wider tuning for category labels. Consistent with the human data (Nosofsky and Johansen, 2000; Smith and Minda, 1998) the network shows less certainty in its classification of the seventh item in each class, than it does for the other, more prototypical, items.

In all the preceding experiments PC/BC-DIM performs categorisation in the same way: the prediction neurons (or neuron) that best account for the stimulus are the most active, and each active prediction neuron in turn activates the reconstruction neuron in the last partition that represents its class label. The preceding experiments have demonstrated that PC/BC-DIM has the flexibility to account for a range of classification strategies used by humans. Specifically, PC/BC-DIM was used to implement a prototype + rule + prototype/exemplar method to simulate Nosofsky et al. (1989, expt. 1); used a prototype and multiple rules to simulate Nosofsky et al. (1989, expt. 2); employed multiple prototype/exemplars to simulate Aha and Goldstone (1992); a rule + exemplar method and a prototype method to simulate different behaviours for the task described in Medin and Schaffer (1978, expt. 2); multiple prototypes + exemplars were used to simulate Smith and Minda (1998, expt. 2); and an exemplar method was used to generate the results in section 3.1.1. These different results have been produced by simply varying the number of feature dimensions, the number of prediction neurons in the network, and the weights of those neurons. The algorithm that performs predictive coding with the PC/BC-DIM network, and hence, which generates the classification results has not changed. While it is important to demonstrate that PC/BC-DIM can account for the flexibility of human categorisation, such flexibility can only be used if these different strategies can be learnt. Investigating learning algorithms that can allow PC/BC-DIM network to learn categorisation will be the subject of future work.

3.2 Contextual Influences on Perception

3.2.1 The Influence of Word Knowledge on Letter Perception

Perceptual categorisation can be influenced by prior knowledge and contextual information. For example, the words in Fig. 6a can be easily read despite the middle letter in each word being formed by the same ambiguous shape (Selfridge, 1955). To achieve this, the brain combines knowledge about valid words in the English language with contextual information provided by the surrounding letters to perceive the same shape as a different letter in each context. Similarly, the word shown in Fig. 6b can be unambiguously identified even when parts of the constituent letters are obscured making the identity of those letters ambiguous (McClelland et al., 1986). Again, this seems to require knowledge of whole words to disambiguate perception of individual letters.

Contextual influences on letter perception have previously been simulated using the interactive activation and competition (IAC) neural network (McClelland, 2013; McClelland et al., 2014; McClelland and Rumelhart, 1981; Rumelhart and McClelland, 1982). PC/BC-DIM can also be used to simulate the same phenomena. The PC/BC-DIM model consists of a hierarchy of two processing stages as illustrated in Fig. 7. The input to the network comes from a set of 14 features representing the strokes forming alphanumeric characters at four possible locations (like segments of an LCD display). Inputs are given a value of one if the corresponding stroke is visible, and a value of zero otherwise. An input of zero could thus correspond to a stroke not forming part of a letter, to a letter being incomplete, or due to a letter being occluded; the model is not sophisticated enough to distinguish these cases. The arrangement of the strokes, and the font, used to define each letter was identical to that described in (McClelland and Rumelhart, 1981). Each prediction neuron in the first processing stage receives connections from

THE CAT
(a)
TRAPPED
(b)

Figure 6: (a) “the cat” written using an ambiguous shape for the middle letter of each word (redrawn from Fig.3 of Selfridge, 1955). (b) “trapped” written so that some letters are partially occluded resulting in their identities being ambiguous (inspired by Fig.2 of McClelland et al., 1986).

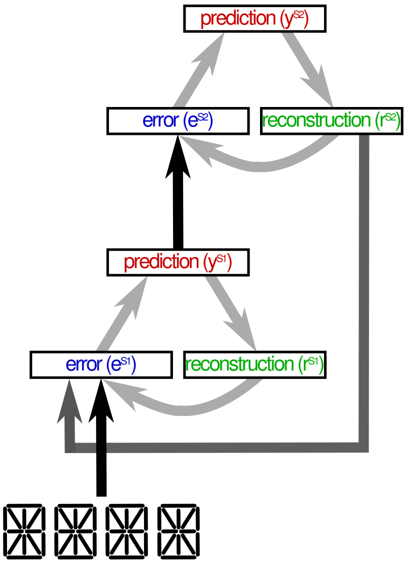
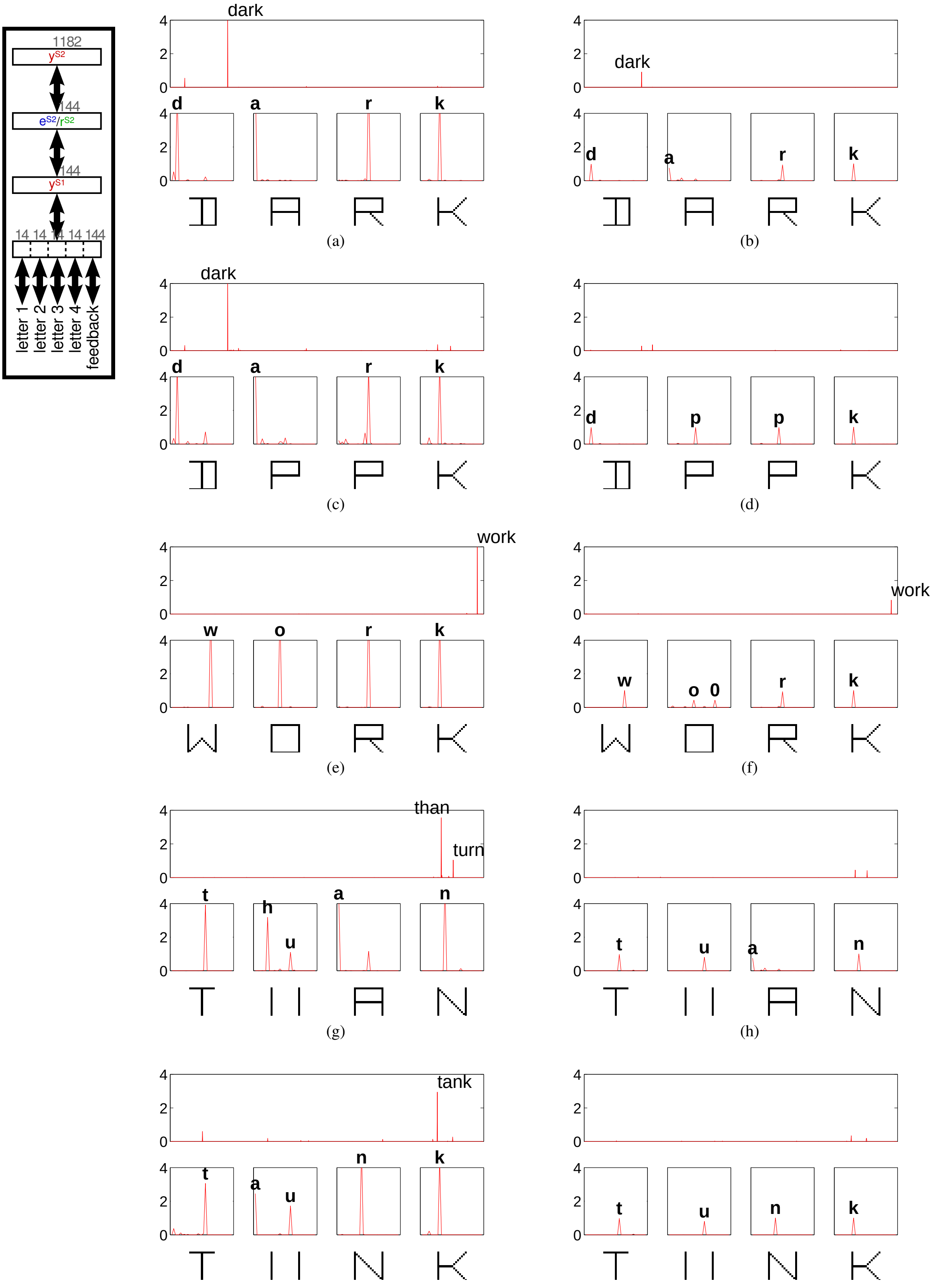


Figure 7: The two-stage hierarchical PC/BC-DIM network used to simulate the contextual influences of word knowledge on letter perception. Each processing stage in this hierarchy consists of a network of three neural populations, as illustrated in, and described in the caption to, Fig. 1. To improve clarity, connections within and between processing stages are shown in different shades of grey. All connections within a processing stage are shown in light-grey. The connections between processing stages can be classified as bottom-up or top-down. Bottom-up connections link the sensory inputs to the first processing stage and connect the first processing stage to the second processing stage, and are shown in black. Top-down connections exist from the second processing stage to the first, and are drawn in dark-grey.

a unique combination of features corresponding to the strokes forming a single character at a single location. The whole population of 144 prediction neurons represents all 26 letters plus the digits 0 to 9 at all four locations. Each prediction neuron in the second processing stage receives four connections from prediction neurons in the first processing stage. These four connections make each second stage prediction neuron represent a specific combination of letters across the four locations. The whole population of 1182 second-stage prediction neurons represent a corpus of English four-letter words. The reconstruction neurons in the second processing stage will represent the expected input to the second stage (*i.e.*, the expected outputs of the first-stage prediction neurons) given the predicted causes of this input (*i.e.*, the beliefs about words represented by the second-stage prediction neuron activations). This second-stage reconstruction is fed-back as an additional input to the first-stage. Each prediction neuron in the first processing stage receives (via the error neuron population) a single input from the corresponding element of the second-stage reconstruction. This provides top-down activation to prediction neurons representing beliefs about individual letters from second-stage prediction neurons representing beliefs about whole words.

When this PC/BC-DIM network is presented with four letters that form a known word (Fig. 8a), the first-stage prediction neurons that represent the individual letters become strongly active. The second-stage prediction neuron that represents the word is also strongly activated. When, as in this example, the input pattern is unambiguous the top-down connections from the second to the first processing stage have little influence (except on the scale of the response) as illustrated in Fig. 8b, which shows the same simulation with the top-down connections removed from the PC/BC-DIM model. However, if the input pattern is ambiguous then the top-down connections have a significant influence. For example, Fig. 8c shows a situation analogous to that shown in Fig. 6b where two letters are incomplete. In this example, the intact letters can activate neurons in the second processing stage


1182
y^S2
144
e^S2/r^S2
144
y^S1
14 14 14 14 144
letter 1
letter 2
letter 3
letter 4
feedback
4
2
0
dark
d a r k
work
w o r k
o 0
p p
than
turn
t h u a n
t u a n
tank
t a u n k
t u n k

(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

Figure 8: (previous page) Simulations of word and letter recognition. Each sub-figure shows the input to the PC/BC-DIM network (bottom) the activation of the first-stage prediction neurons (middle) and the activation of the second-stage prediction neurons (top). The responses of first-stage prediction neurons representing letters and digits at different positions are shown separately. The preferred stimuli of the most active prediction neurons are indicated by the labels. (a) and (b) show results when the input patterns are unambiguous. (c) and (d) show results when the pattern presented in the middle positions is incomplete and potentially consistent with several different letters. (e) and (f) show results when one of the input letters is perceptually identical to a digit. (g) to (j) show results when an ambiguous shape is presented in two different contexts. The left column shows results for an intact PC/BC-DIM network, the right column shown corresponding results for a network in which the top-down connections between the second and first processing stages have been removed.

that represent words consistent with these letters. The top-down connections allow this information about possible words to influence the response of the prediction neurons responding to the incomplete letters. The network comes to represent the most likely letters and words given the input and the knowledge about possible English words, encoded in the weights of the second-stage prediction neurons. In contrast, when the top-down connections are removed from the PC/BC-DIM network (Fig. 8d), the incomplete pattern is represented as the letter with which it shares most features, despite the fact that the resulting combination of letters does not correspond to a valid word.

A similar effect occurs without incomplete letters, due to certain letters and digits being perceptually identical. The result shown in Fig. 8e is analogous to that shown in Fig. 8a, but for another input word. However, when top-down connections are removed (Fig. 8f), the first-stage prediction neurons are unable to determine if the second character is letter “O” or the digit “0”, despite the surrounding context.

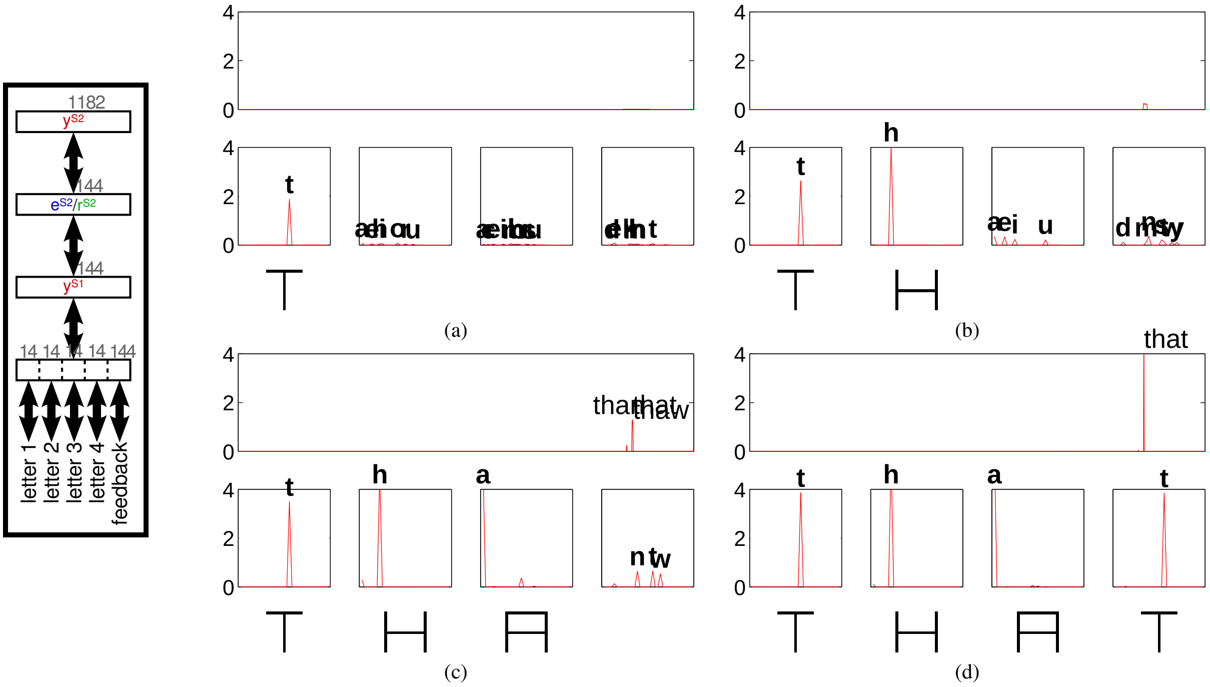
The influence of context and word knowledge, conveyed by the top-down connections, can also be illustrated by presenting an ambiguous shape to the network in two different contexts, analogous to the situation shown in Fig. 6a. In Fig. 8g the ambiguous shape (the second character) is perceived by the PC/BC-DIM network to be a letter H, whereas in Fig. 8i the same ambiguous shape is considered to most probably be a letter “A”. Without top-down connections the ambiguous shape, in either context (Fig. 8h and j), is perceived as the most perceptually similar letter (“U”): it differs by one feature (the bottom horizontal stroke) from the letter “U”, but by two features (the two central horizontal strokes) from the letter “H”, and by a three features (the two central horizontal strokes and the top horizontal stroke) from the letter “A”. It might be argued that a more graded response to each similar letter would be more desirable. However, these results are for the crippled version of the network: in the intact version top-down knowledge about words results in the ambiguous input being perceived as one of the less similar letters.

The influence of the knowledge about possible English words, encoded in the weights of the second-stage prediction neurons, can also be observed in the absence of any input at a particular letter location, as illustrated in Fig. 9. As the number of visible letters increases, so the range of possible words decreases (and fewer second-stage prediction neurons are active), and the range of possible letters that could fill the remaining spaces also decreases (and fewer first-stage prediction neurons are active).

3.2.2 Simulating Human Letter Perception Tasks

A major success of the IAC model was its ability to account for the improved identification of letters appearing in words and pseudowords compared to letters appearing in nonwords and in isolation (McClelland and Rumelhart, 1981). This word superiority effect can also be simulated using the PC/BC-DIM model. Fig. 10a shows the temporal response of a single prediction neuron in the first-stage of the hierarchy. The recorded neuron is selective for the the letter A as the second character in the input word. Its response has been recorded when this letter A appears in a variety of contexts. It can be seen that the response is strongest when the A appears in a word (CAVE) or a pronounceable pseudoword (MAVE), and is weakest when the A appears in a string of non-letter symbols (3A33), in a nonword (UAHB), or in isolation (_A__). This effect is due to the number of the word selective neurons in the second-stage that are activated by each contextual input, and the strength with which they are activated. This in turn affects the strength of top-down feedback received by the recorded first-stage prediction neuron, or by its rivals that represent other possible letters.

In the experiments with human subjects the perceptibility of the target letter was measured using the percentage of times that the letter was correctly identified. To be able to compare the human data directly with the model, it is necessary to define a measure of the models behaviour which correlates with letter perceptibility. For this purpose the mean response of first-stage prediction neuron selective for the target letter is used. The comparison of these measures of letter perceptibility in the model and in human subjects, for three conditions is presented in Fig. 10b.



(a)

(b)

(c)

(d)

Figure 9: Prediction of missing letters. The format of this figure is identical to, and explained in the caption of, Fig. 8. The number of letters presented to the network increases from subplot (a) to subplot (d). In each case the network predicts the missing letters. These predictions become more constrained as the number of missing letters, and hence the range of possible words, is decreased.

It can be seen that the model is in good agreement with the behavioural data.

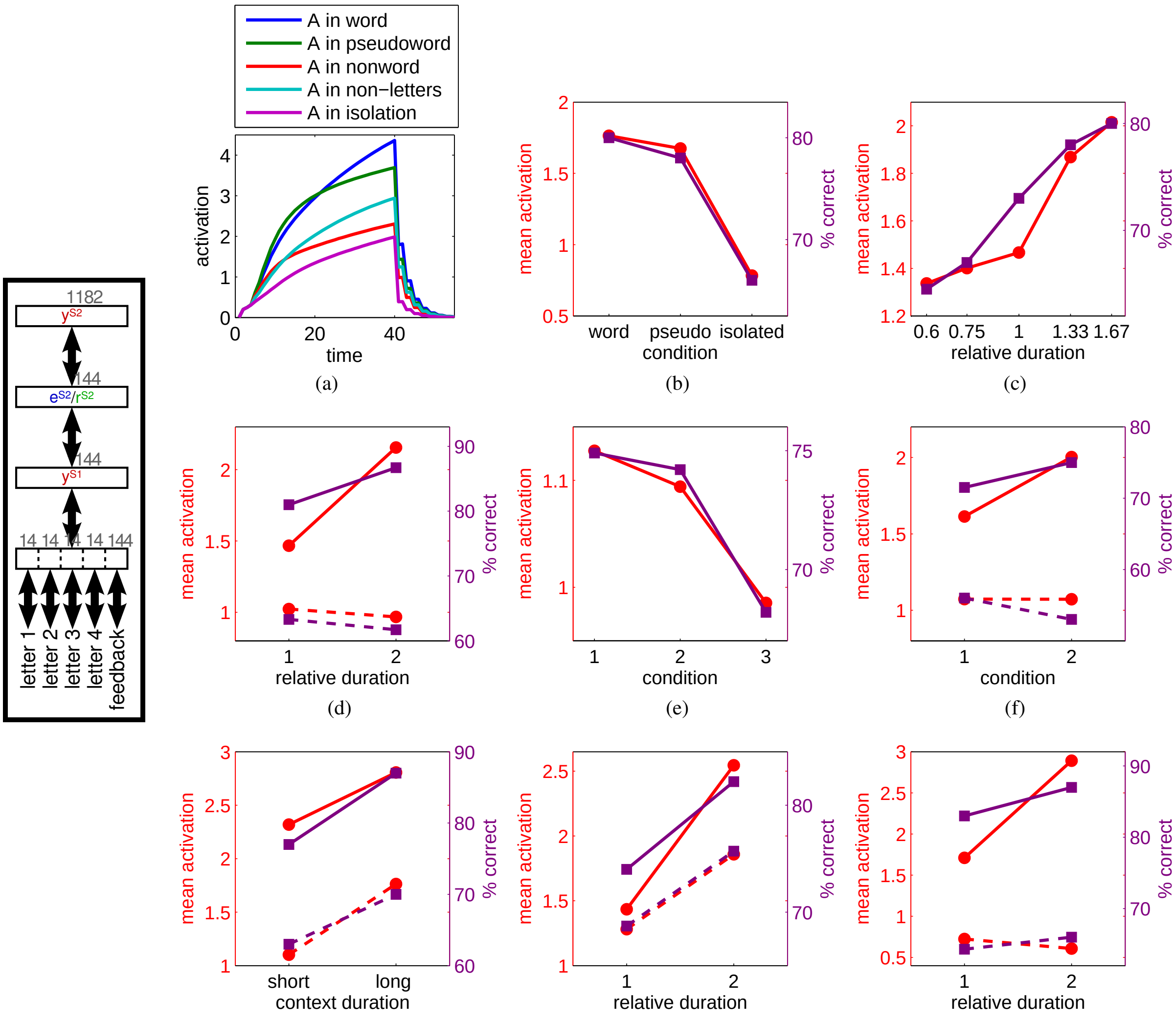
The IAC model was also extensively tested with a range of experiments which varied the relative onset and offset times of the target letter and its context (Rumelhart and McClelland, 1982). The behaviour of the IAC model was found to be in good agreement with the behaviour of human subjects. The behaviour of the PC/BC-DIM model on the same set of tasks is shown in Fig. 10c-i, and is also in close agreement with human behaviour. To simulate the onset and offset of stimuli in these experiments, the input to the PC/BC-DIM network was changed between iterations of equations 1, 2 and 3.

Fig. 10c shows results for an experiment in which the target letter and its context form a valid English word (Rumelhart and McClelland, 1982, expt. 1). The target letter remains visible for a fixed period of time and disappears at the same time as the context. However, the context appears at varying times: later than the target when relative duration is less than one, and earlier than target when relative duration is greater than one. For both the PC/BC-DIM model and the human subjects the perceptibility of the target letter increases with the earlier onset of the context.

Fig. 10d shows results for an experiment that used two values for the relative duration of the context and target letter (Rumelhart and McClelland, 1982, expt. 2). For a relative duration of one, both context and target appeared and disappeared together. For a relative duration of two, the onset of the context was earlier than that of the target, such that the context was present for twice as long as the target, with both disappearing simultaneously. In contrast to the previous experiment two types of context were used. Firstly, a context that formed a valid word with the target letter (results shown with solid lines), and a context of non-letter stimuli (results shown with dashed lines). For both the PC/BC-DIM model and the human subjects the perceptibility of the target letter is improved when the context forms a word, and only in this condition does early context onset improve performance further.

Fig. 10e shows results for an experiment in which the target letter and its context form a valid English word, and both are visible for the same duration (Rumelhart and McClelland, 1982, expt. 3). In condition 1 the context appears first and disappears simultaneously with the onset of the target letter. In condition 2 both context and target letter have the same onset and offset times. In condition 3 the target letter appears first and disappears simultaneously with the onset of the context. At offset each letter was replaced by a mask in all three conditions. It can be seen that for both the PC/BC-DIM model and the humans the perceptibility of the target letter is reduced as the delay in the onset of the context increases.

Fig. 10f shows results for an experiment in which the context is displayed for twice as long as the target letter



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 10: Simulations of human letter perception tasks. (a) and (b) The word superiority effect. (a) shows the temporal response of a first-stage prediction neuron when its preferred stimulus is presented to the network in various different contexts. In all conditions both the preferred letter and the context were visible for 40 iterations. (b) The average response recorded in three of the conditions in (a), together with human accuracy data for the same task (data from expt. 1 of McClelland and Johnston, 1977). In this and all subsequent sub-figures circular markers show PC/BC-DIM simulation results (plotted against the left-hand scale) and square markers show corresponding human data (plotted against the right-hand scale). (c)-(i) Show simulations of, and human data from, the following experiments reported in Rumelhart and McClelland (1982): (c) expt. 1, (d) expt. 2, (e) expt. 3, (f) expt. 4, (g) expt. 5, (h) expt. 7, and, (i) expt. 8.

(Rumelhart and McClelland, 1982, expt. 4). In condition 1 the context and the target appear together so that the target disappears before the context. In condition 2 the context appears before the target and both disappear at the same time. Two types of context were used. Firstly, a context that formed a valid word with the target letter (results shown with solid lines), and a context of non-letter stimuli (results shown with dashed lines). For both the PC/BC-DIM model and the humans the perceptibility of the target letter is improved when the context forms a word, and only in this condition does early context onset improve performance further.

Fig. 10g shows results for an experiment in which the target letter and context form a valid English word (Rumelhart and McClelland, 1982, expt. 5). Both the target and context can be displayed either for a short duration or a long duration. In all conditions the target letter and the context disappear at the same time. Results for a long-duration target are shown using solid lines, and results for a short-duration target as shown using dashed lines. It can be seen that for both the the PC/BC-DIM model and the humans the perceptibility of the target letter is increased by increasing the duration for which the target is displayed, and that perceptibility is also increased

Name	Gang	Age	Education	Marital Status	Profession
Art	Jets	40s	JnrHigh	Single	Pusher
Al	Jets	30s	JnrHigh	Married	Burglar
Sam	Jets	20s	College	Single	Bookie
Clyde	Jets	40s	JnrHigh	Single	Bookie
Mike	Jets	30s	JnrHigh	Single	Bookie
Jim	Jets	20s	JnrHigh	Divorced	Burglar
Greg	Jets	20s	HighSch	Married	Pusher
John	Jets	20s	JnrHigh	Married	Burglar
Doug	Jets	30s	HighSch	Single	Bookie
Lance	Jets	20s	JnrHigh	Married	Burglar
George	Jets	20s	JnrHigh	Divorced	Burglar
Pete	Jets	20s	HighSch	Single	Bookie
Fred	Jets	20s	HighSch	Single	Pusher
Gene	Jets	20s	College	Single	Pusher
Ralph	Jets	30s	JnrHigh	Single	Pusher
Phil	Sharks	30s	College	Married	Pusher
Ike	Sharks	30s	JnrHigh	Single	Bookie
Nick	Sharks	30s	HighSch	Single	Pusher
Don	Sharks	30s	College	Married	Burglar
Ned	Sharks	30s	College	Married	Bookie
Karl	Sharks	40s	HighSch	Married	Bookie
Ken	Sharks	20s	HighSch	Single	Burglar
Earl	Sharks	40s	HighSch	Married	Burglar
Rick	Sharks	30s	HighSch	Divorced	Burglar
Ol	Sharks	30s	College	Married	Pusher
Neal	Sharks	30s	HighSch	Single	Bookie
Dave	Sharks	30s	HighSch	Divorced	Pusher

Table 1: The Jets and Sharks data set (McClelland, 2014).

by increasing the duration of the context.

Fig. 10h shows results for an experiment (Rumelhart and McClelland, 1982, expt. 7) which uses an identical procedure to that used to produce the results shown in Fig. 10d except for the context used to generate the results shown using the dashed lines. In Fig. 10d the dashed lines show results for a context of non-letter stimuli. In Fig. 10h the dashed lines show results for a context that forms a pseudoword with the target letter. It can be seen that a pseudoword context improves the perceptibility of the target letter, and this perceptibility is increased by early context onset.

Fig. 10i shows results for an experiment (Rumelhart and McClelland, 1982, expt. 8) which uses an identical procedure to that that used to produce the results shown in Fig. 10h except that dashed lines show results when the context and target letter form a non-word. For both the the PC/BC-DIM model and the humans, when the context forms a non-word with the target letter, the duration of the context has little effect on the perceptibility of the target.

3.3 Reasoning About Conceptual Knowledge

The IAC neural network was also proposed as a model of how the brain could store and reason about conceptual knowledge (McClelland, 2014; Rumelhart et al., 1986). These abilities of the IAC network were illustrated using the Jets and Sharks dataset, which lists the attributes of a number of individuals (table 1). A single-stage PC/BC-DIM network can be used to reason about this data. The PC/BC-DIM implementation is far simpler than the IAC implementation (as was the case for the PC/BC-DIM model of letter and word perception described previously), as it does not require neurons representing distinct attributes to be placed into separate pools, nor does it require separate inhibitory connection between neurons within a pool. The input to the PC/BC-DIM network consists of a 41-element vector representing all the possible attributes listed in Table 1 (27 names, 2 gangs, 3 age groups, 3 education types, 3 marital statuses, and 3 professions). A single prediction neuron is used to represent each individual, hence there are 27 prediction neurons. Synaptic weights take binary values. Each prediction neuron has a non-zero synaptic weight from all inputs corresponding to the attributes of the individual represented by that prediction neuron. For example, the prediction neuron representing the individual Art will have connections of weight equal to each one from the inputs “Art”, “Jets”, “40s”, “JnrHigh”, “Single” and “Pusher” and weights equal to zero (or equivalently no connection) from the other 35 inputs. The information represented in Table 1 is therefore directly, and straightforwardly, encoded in the weights of the network.

Activating the input corresponding to the name “Art” results in the single prediction neuron representing this individual being active. This active prediction neuron, in turn, produces a strong response from all the reconstruction neurons representing the attributes of the person Art (Fig. 11a). Hence, from information about a name it is possible to recall the age, occupation, *etc.* of the corresponding individual. If the same experiment is performed with IAC the neurons representing the person Art and the attributes “Jets”, “40s”, “JnrHigh”, “Single” and “Pusher” all become active. However, so do the neurons representing Clyde and Ralph. In fact the neuron representing the person Clyde is almost as active as the neuron representing the attribute “40s” (Bechtel and Abrahamsen, 1991). The behaviour of the PC/BC-DIM network is thus, arguably, superior to that of the IAC network in this particular case. However, it should be noted that in IAC networks the spreading of activation to related people and subsequently to the attributes of those people, is claimed as a feature of the IAC architecture as it allows the model to make generalisations (Kumaran and McClelland, 2012).

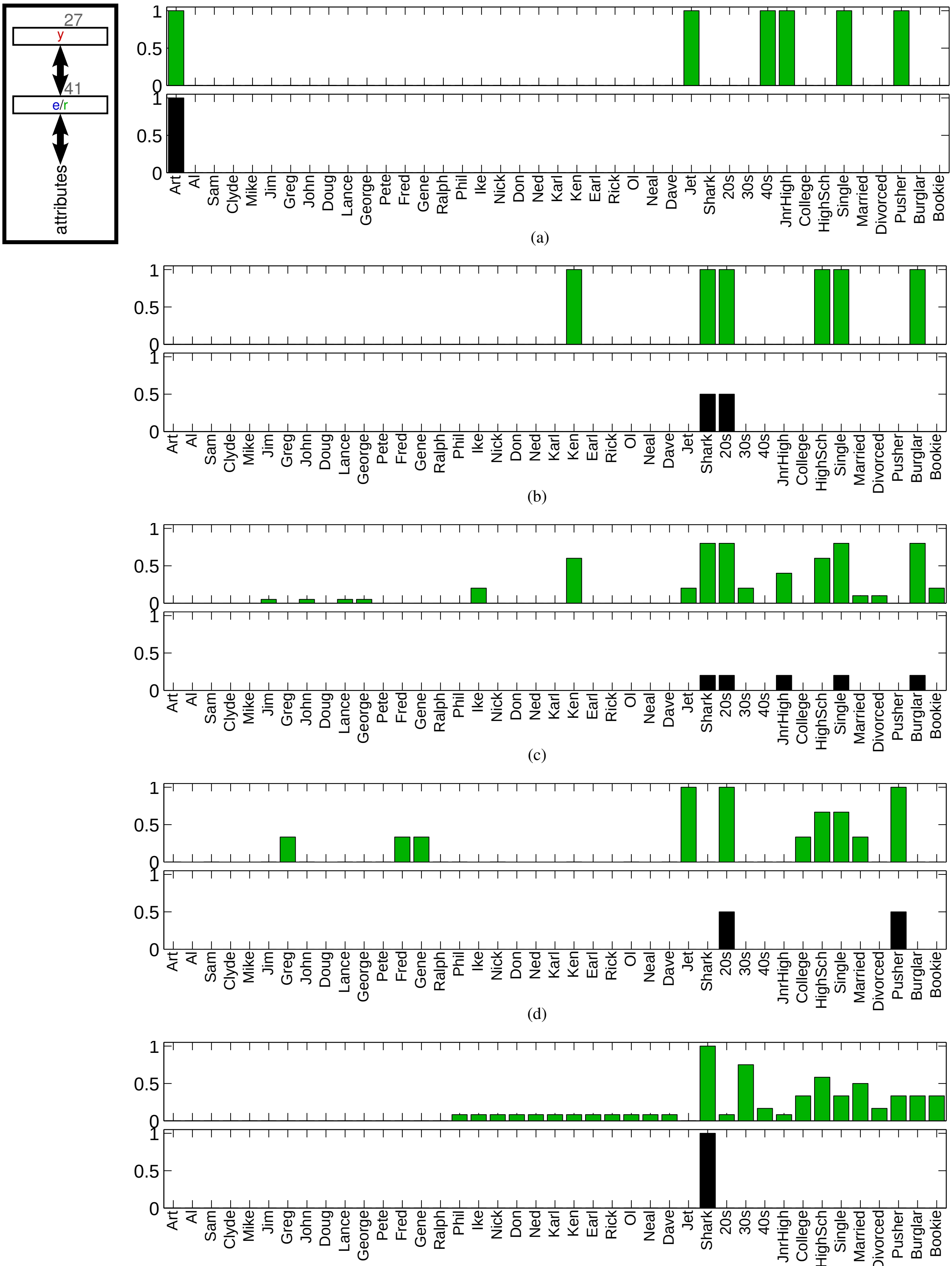
Given a partial description of an individual it is possible to retrieve their other attributes. For example, Fig. 11b shows the results of activating the inputs corresponding to the attributes “Shark” and “20s”. Ken is uniquely defined by these attributes, so the prediction neuron representing Ken becomes active (in isolation), and the name “Ken” and his other attributes (“HighSch”, “Single”, and “Burglar”) are represented by the reconstruction neuron responses generated by the active prediction neuron. Again, the PC/BC-DIM network arguably performs this task better than the IAC network. In the IAC network, inputting the attributes “Shark” and “20s” results in weak activation of all three neurons representing different professions (McClelland, 2014). The PC/BC-DIM network can also calculate the most likely person given incorrect or partially contradictory information. For example, if the input consists of all of Ken’s attributes except that “JuniorHigh” is presented instead of the correct value “HighSchool”, the response of the reconstruction neurons still identified Ken as the most likely individual being described (Fig. 11c).

In cases where the input attributes do not uniquely define an individual, the PC/BC-DIM network will determine the correct conditional probabilities associated with different possible values for the attributes. For example, the attributes “20s” and “Pusher” are shared by three individuals, hence, each prediction neuron representing those three individuals becomes active with strength $\frac{1}{3}$ (and all other prediction neurons have a response of zero). These three active prediction neurons in turn activate the reconstruction neurons representing the attributes of these three individuals. Specifically, the reconstruction neuron representing the names of those three individuals becomes active with a strength of $\frac{1}{3}$ (Fig. 11d). It can be seen from Table 1 that the conditional probability of “JnrHigh” (and of “Divorced”) given “20s” and “Pusher” is 0, the conditional probability of “College” (and of “Married”) given “20s” and “Pusher” is $\frac{1}{3}$, and the conditional probability of “HighSch” (and of “Single”) given “20s” and “Pusher” is $\frac{2}{3}$. These values are all correctly calculated by the responses of the reconstruction neurons. As all three possible individuals are in the same gang the reconstruction neurons represent this attribute “Jet” with a value of one. In contrast, for the same inputs an IAC network fails to activate the neurons representing “College” and “Married” but does activate the neuron representing the individual Pete (Bechtel and Abrahamsen, 1991).

Fig. 11e shows the output of the PC/BC-DIM network when the inputted attribute (in this example “Shark”) is shared by many individuals. The responses of the reconstruction neurons identify all the members of the Sharks gang. Additionally they represent the correct conditional probabilities associated with all the other attributes. For example, it can be seen from Table 1 that one member of the Sharks is in his 20s, nine are in their 30s, and two are in their 40s. Hence, the probability that a member of the sharks is in his “20s”, his “30s”, or his “40s” is $\frac{1}{12}$, $\frac{9}{12}$, and $\frac{2}{12}$ respectively. The responses of the reconstruction neurons representing the attributes “20s”, “30s”, and “40s” are exactly equal to these conditional probabilities. In contrast, IAC only activates one neuron representing an age attribute (“30s”) and it fails to activate the neurons representing several individuals who are members of the Sharks (Bechtel and Abrahamsen, 1991).

3.4 Context Dependent Task Switching

Many behaviours are contingent on circumstances. A classic example is the behaviour of a person in response to a ringing telephone or a ringing door-bell. This will depend on whether the person is in their own home, or someone else’s house (Cohen and Newsome, 2008; Salinas, 2004a), or if the sound is coming from the television or elsewhere. Another example, proposed by Salinas (2004a), is a hypothetical laboratory experiment in which the participant maintains fixation at a coloured spot while a target is flashed either to the left or right of the fixation spot. The participant must then make an eye movement either to the location of the target or to a location an equal distance from the fixation spot but on the opposite side to where the target appeared. Whether the participant



(a)

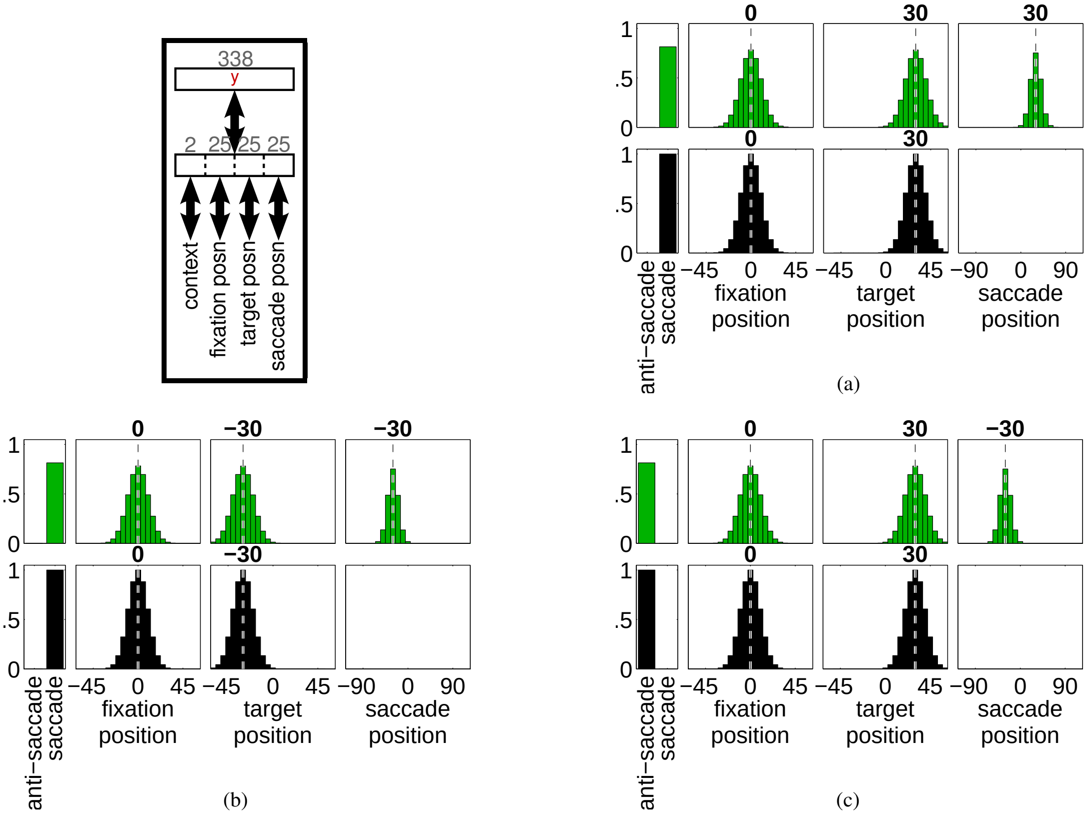
(b)

(c)

(d)

(e)

Figure 11: Simulations with the Jets and Sharks data. In each figure the lower histogram show the input to the PC/BC-DIM network and the upper histogram shows the response of the reconstruction neurons. In all cases the input vector is normalised to sum to unity, which allows the reconstruction neurons to represent the conditional probabilities exactly. Without normalisation of the input vector, the reconstruction neuron activities are proportional to the conditional probabilities. (a) Retrieving properties from a name. (b) Retrieving properties from a unique partial description. (c) Retrieving probable properties from an incorrect description. (d) Retrieving the conditional probabilities of properties from an ambiguous partial description. (e) Retrieving typical properties of “Sharks”.



(a)

(b)

(c)

Figure 12: Simulations of the context-dependent behaviour selection task proposed in Salinas (2004a). In each figure the lower histogram show the input to the PC/BC-DIM network and the upper histogram shows the response of the reconstruction neurons. The network receives input from multiple sources, representing the context (saccade or anti-saccade), the fixation position, and the target position. Each of these three variables are represented by a separate partition of the input, and a corresponding partition of the reconstruction neuron population. A fourth partition represents the output of the network: the required position to which a saccade is to be made. Positions are represented by Gaussian population codes, and the mean of this distribution is indicated by the number above the corresponding histogram. If the variable A represents the fixation position, the variable B represents the target position, and the variable C represents the saccade position, then the network has been wired-up to approximate C=A-B in the first context (anti-saccade), and to approximate C=A+B in the second context (saccade). When three inputs representing the context, fixation position, and target position are presented (lower histograms), the reconstruction neurons generate an output (upper histograms) that represents the correct value of the saccade position (as well as outputs representing the given values of the other three variables).

does the former (a saccade) or the latter (an anti-saccade) is determined by the colour of the fixation spot. A PC/BC-DIM network was used to simulate this saccade/anti-saccade task.

The PC/BC-DIM network employed four partitions of the input and the reconstruction neurons. The first partition consisted of two elements that represented the context (saccades or anti-saccades). The second partition represented the position of the fixation point and the third partition represented the position of the target. In both cases the position information was represented by a Gaussian population code centred at the true location. The final partition represented the required action, which was also encoded using a Gaussian population code centred at the location of the intended saccade. The PC/BC-DIM network was wired-up so that each prediction neuron represented a particular combination of context, fixation position and target location. Each prediction neuron also had weights to the final partition that represented the position of the required saccade, given the values of the other three variables. A total of 338 prediction neurons were used to cover the full range of possible contexts, fixation positions and target locations.

In the saccade condition, the position of the target needs to be added to the position of fixation to determine the correct saccade location. Figure 12a and b show, for two different target locations, that the network succeeds in doing this. In the anti-saccade condition, the position of the target needs to be subtracted from the position

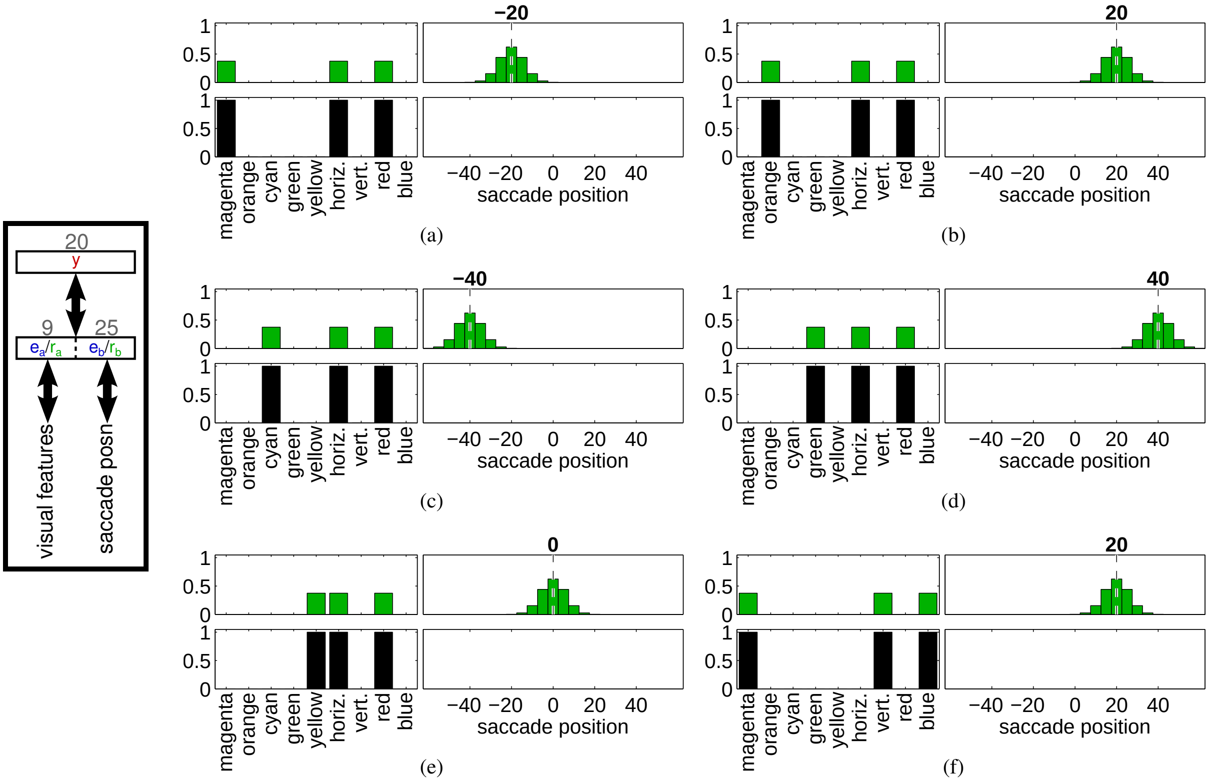


Figure 13: Simulations of the context-dependent behaviour selection task proposed in Salinas (2004b). In each figure the lower histogram show the input to the PC/BC-DIM network and the upper histogram shows the response of the reconstruction neurons. The network receives inputs representing the fixation spot colour (magenta, orange, cyan, green, yellow) and the attributes of the stimulus (horizontal, vertical, red, blue). A second partition represents the output of the network: the required position to which a saccade is to be made. This position is represented by a Gaussian population code, and the mean of this distribution is indicated by the number above the corresponding histogram. The network has been wired-up to calculate the required eye-movement for each possible combination of context and stimulus features. When inputs representing the context, the stimulus orientation and the stimulus colour are presented (lower histograms), the reconstruction neurons generate an output (upper histograms) that represents the correct value of the saccade position (as well as outputs representing the given values of the other variables).

of fixation to determine the correct saccade location. Figure 12c shows that the network succeeds in doing this task. In this example, the location for both fixation and target are identical to those used in Fig. 12a, however, the change in context, causes a significant change in eye movement.

In each condition, the PC/BC-DIM algorithm selects the subset of prediction neurons that best explain the input. For example, for the condition shown in Figure 12a the inputs cause responses in the subset of prediction neurons with connections to the "saccade" input and with Gaussian RFs centred near 0^o in the second partition and near 30^o in the third partition. Each of these prediction neurons has a Gaussian RF centred near 30^o in the last partition. The reconstruction neuron responses are a linear combination of all the active prediction neuron RFs, and hence, will peak at the appropriate places in each of the four partitions. In the condition shown in Figure 12c a different set of prediction neurons are activated by the input. These prediction neurons also have RFs in the second and third partitions centred around 0^o and 30^o, but have a selectivity for the other context (in the first partition), and an RF centred near -30^o in the last partition.

Salinas (2004b) proposed a second, more complex, hypothetical experiment to illustrate context-dependent behaviour. In this second task there are five contexts, that are again indicated by the colour of the fixation spot. The participant is required to make an eye-movement to report one feature of a separate stimulus. When the fixation spot is magenta, the participant makes an eye-movement to report the orientation of the stimulus, saccadding to a proximal location to the left of the fixation spot if the stimulus is horizontal, or to a proximal location to the right of the fixation spot if the stimulus is vertical. When the fixation spot is orange, the participant again makes an eye-movement to report the orientation of the stimulus, but the locations for the saccades are reversed compared to the previous condition. When the fixation spot is cyan, the participant makes an eye-movement to report the colour of the stimulus, saccading to a distal location to the left of the fixation spot if the stimulus is red, or to a distal location to the right of the fixation spot if the stimulus is blue. When the fixation spot is green, the participant again makes an eye-movement to report the colour of the stimulus, but the locations for the saccades are reversed compared to the previous condition. In the fifth context, when the fixation spot is yellow, the participant is required to maintain fixation at the central fixation spot (the no-go condition).

To simulate this task a PC/BC-DIM network was wired-up so that a population of 20 prediction neurons represented each possible combination of context, stimulus orientation and stimulus colour. Each prediction neuron was also given weights to represent to correct eye-movement associated with the combination of context and stimulus properties that it represented. These saccade position weights were defined using a Gaussian centred at the correct position. Figure 13a-e show the behaviour of the network when the stimulus remains constant but the context changes. When the context is a magenta fixation spot, the network generates an eye movement that corresponds to the orientation of the stimulus (Fig. 13a). The same is true when the context is orange, but now the mapping between stimulus orientation and saccade position has been reversed, so that the eye movement is to the right rather than the left (Fig. 13b). When the context is cyan, the eye movement reports the colour of the stimulus (Fig. 13c), and this eye movement is reversed when the context is green (Fig. 13d). In the no-go condition, a yellow context, the saccade position is zero. Figure 13f illustrates that a different saccade position is produced in the magenta context when the stimulus attributes change. In each condition, a single prediction neuron is activated by the visual attributes of the fixation spot and the stimulus. This single prediction neuron has an RF in the second partition centred at the appropriate saccade position, and hence, generates a response in the second partition reconstruction neurons at this location.

Neural networks that could simulate the above two tasks were also described in Salinas (2004a,b). Those networks are similar to the PC/BC-DIM networks presented here. They consist of a population of neurons representing all combinations of possible inputs (a basis function population) the responses of which are mapped onto the required output. However, one significant difference is that in Salinas (2004a,b), each neuron has two distinct types of input connection: one set of connections are driving and the other set of connections are modulatory. The different types of connections are implemented using distinct mathematical operations. In contrast, the current model treats all inputs identically. Despite this, the prediction neurons in PC/BC-DIM models can display gain modulated responses, like those in Salinas (2004a,b), without the need for one set of inputs to explicitly have a multiplicative influence on the response (De Meyer and Spratling, 2011; Spratling, 2014c).

3.5 Reasoning About Collision Physics

Humans have an intuitive understanding about the behaviour of objects in the physical world. This enables people to reason about the likely causes or consequences of physical events, or about the physical properties of the objects involved. One particular aspect of this ability has been explored in a number of experiments investigating intuitive understanding of the relation between velocity and mass in collision events (Gilden and Proffitt, 1989; Runeson and Vedeler, 1993; Sanborn, 2014; Sanborn et al., 2013; Todd and Warren, 1982; Vicovaro and Burigana, 2014). In these experiments, subjects are shown the collision between two objects (A and B) simulated by moving dots on a computer screen. Following the collision, the subject is asked to determine which of the objects had the greater mass. The ability to identify the heavier object is tested as the relative mass of the two objects is varied. Typically, these experiments are repeated using collisions with different coefficients of restitution.

To simulate these experiments a PC/BC-DIM network was used that had six partitions. Input to the first four partitions represented, using Gaussian population codes, the velocities of the two objects before and after collision. The fifth partition was used to encode the coefficients of restitution, and the sixth partition was used to encode relative mass of the two objects. The network was wired up so that each prediction neuron represented a different combination of velocities and coefficient of restitution and had a weight to the sixth partition to represent the true relative mass for that combination of velocities. A total of 525 prediction neurons were used to cover the range of possible pre- and post-collision velocities and coefficients of restitution used in the experiments.

The appropriate weight for the sixth partition was determined using the equation for the conservation of momentum. Specifically, if u represents initial velocity, and v represents final velocity, then if $(v_B - u_B) > (u_A - v_A)$ there was a strong weight to one element of the sixth partition that represented object A being heaviest. If $(v_B - u_B) = (u_A - v_A)$ then there was a strong weight to a second element of the sixth partition that represented the masses being equal, and if $(v_B - u_B) < (u_A - v_A)$ then the there was a strong weight to a third element of the sixth partition that represented the mass of object B being larger. The three elements of the last partition thus act to represent class labels, exactly as in the models presented in section 3.1. In each simulation, velocity information was provided as input to the first four partitions of the network, and the decision about relative

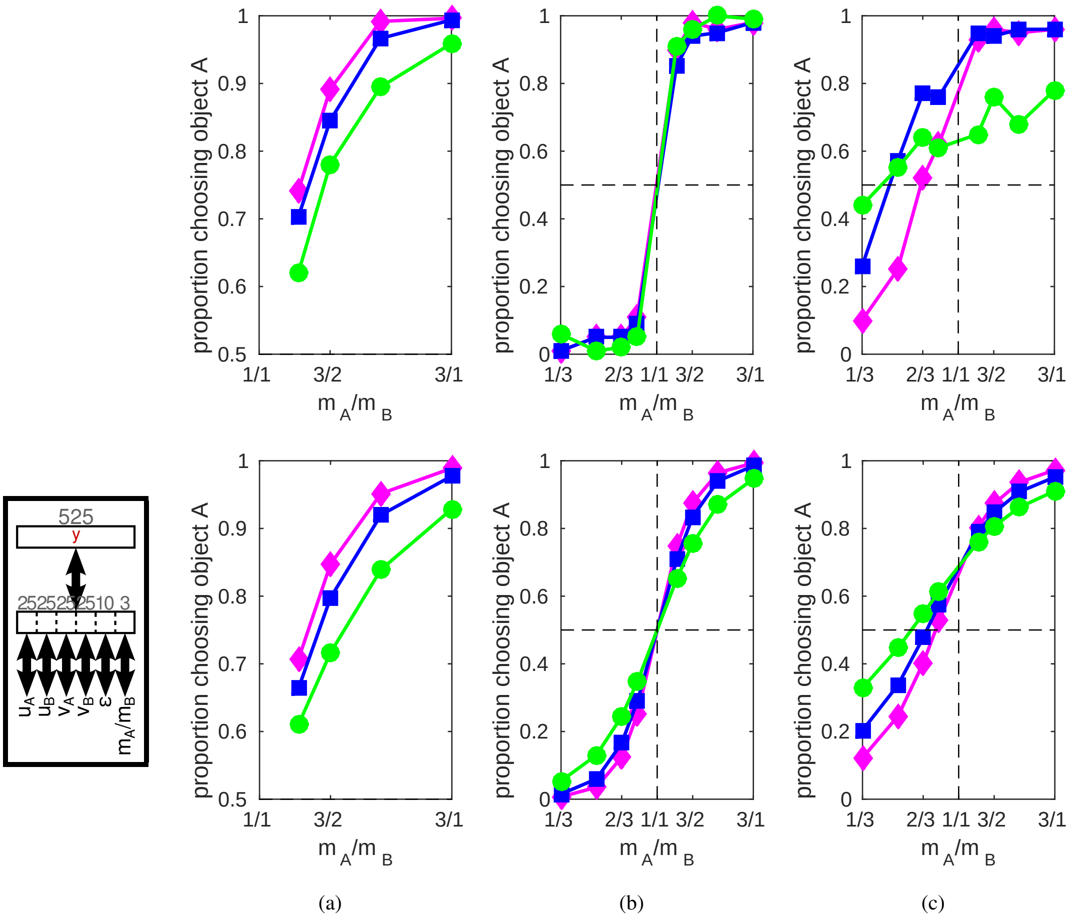
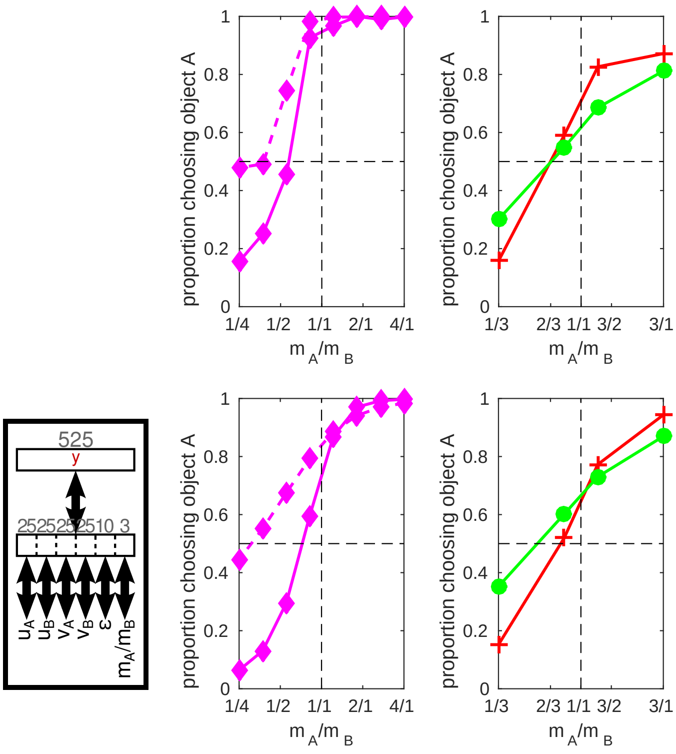


Figure 14: Reasoning about collision physics. Top row shows psychophysical data, bottom row shows corresponding simulation results. In each sub-figure, the coefficient of restitution (ϵ) used in the experiment is indicated by the line marker shape: diamond markers for ϵ=0.9, square markers for ϵ=0.5, circle markers for ϵ=0.1. (a) Todd and Warren (1982, expt. 1). (b) Todd and Warren (1982, expt. 2) moving condition. (c) Todd and Warren (1982, expt. 2) stationary condition.

mass was read out from the reconstruction neuron responses in the sixth partition. The probability that object A was chosen as being the heavier was calculated by taking the squared response of the first reconstruction neuron in the sixth partition and dividing this by the sum of the squared responses of all three sixth partition reconstruction neurons.

Figure 14a shows results for an experiment in which prior to the collision both objects moved towards each other (Todd and Warren, 1982, expt. 1). The initial velocities were varied but the speed with which the objects approached each other remained constant (*i.e.*, as the initial speed of object A increased, the initial speed of object B was decreased in proportion). As with the human subjects, the model is accurate at determining the relative masses of the objects, but with a decline in performance as the coefficient of restitution is reduced. In Todd and Warren (1982, expt. 2) a larger range of relative masses was used. In one condition (the moving condition), as in expt. 1, both objects initially moved towards each other, however, the velocities were fixed. In this condition, the simulation results are in close agreement with the psychophysical results, as shown in Fig. 14b. In the second condition (the stationary condition) one object was initially stationary and the other object had a fixed initial speed. The simulation results for this condition are not such a good fit to the data (Fig. 14c), but they do reflect some key features of the psychophysical results. Specifically, there is a bias for choosing object A as the heavier one (the curves cross the dashed horizontal line slightly to the left of centre), and also the results for the lowest coefficient of restitution are flatter than the others. The fit of the PC/BC-DIM model is very similar to that of the "noisy Newton" model proposed in Sanborn et al. (2013).

Runeson and Vedeler (1993, expt. 2) performed similar collision experiments but, in one condition, both objects were invisible prior to the collision. Hence, in this condition the initial velocities of the objects were unavailable to (or "occluded" from) the observers. To simulate this result with PC/BC-DIM, the input was set



(a)

(b)

(c)

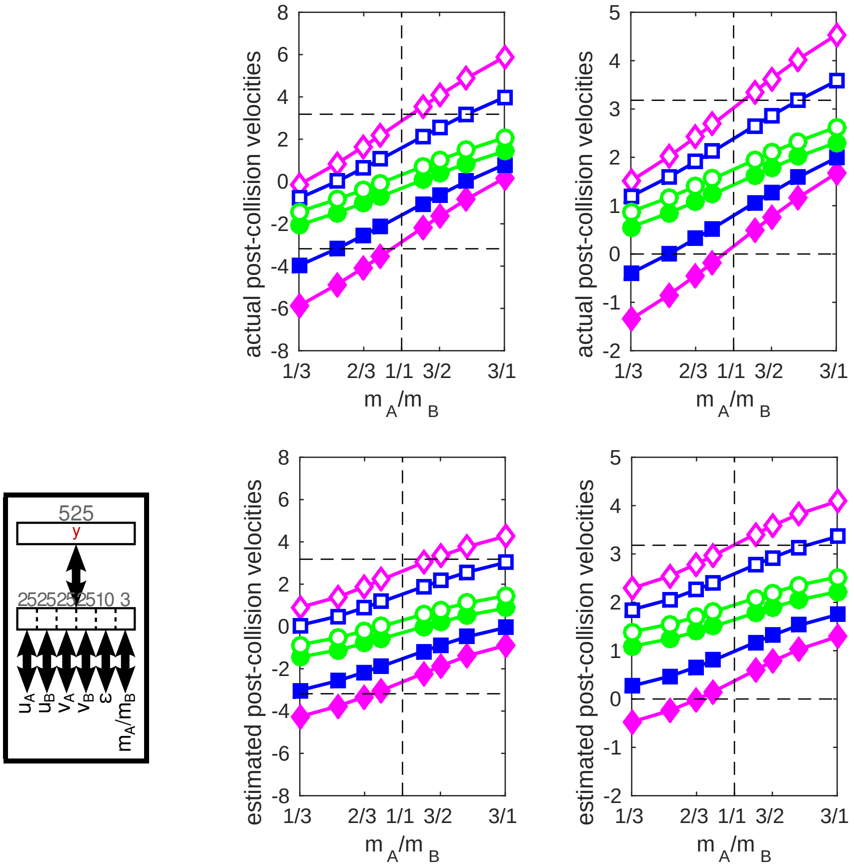
(d)

Figure 15: Reasoning about collision physics with occluded velocities. Top row shows psychophysical data, bottom row shows corresponding simulation results. In each sub-figure, the coefficient of restitution (ϵ) used in the experiment is indicated by the line marker shape: plus markers for ϵ=1, diamond markers for ϵ=0.9, circle markers for ϵ=0.1. (a) Runeson and Vedeler (1993, expt. 2). The solid line indicates results for when both pre- and post-collision velocities are visible, the dashed line indicates results for when only post-collision velocities are visible. (b-d) Experiment shown in Sanborn (2014, Fig. 3): (b) both pre- and post-collision velocities are visible, (c) object B invisible post-collision, (d) object A invisible post-collision.

equal to zero for those partitions of the input vector that represented the initial velocities. As with the human subjects, the model is strongly biased to select object A as the heavier one when the initial velocities are occluded (Fig. 15a).

Rather than occluding the initial velocities, Sanborn (2014) ran experiments in which one or other of the post-collision velocities were occluded. One object was initially stationary (as in the stationary condition of Todd and Warren, 1982, expt. 2), and the initial velocity of the other object varied. Hence, when there was no occlusion (Fig. 15b) the results were similar to those shown in Fig. 14c. Figure 15c shows results for when the post-collision velocity of the initially stationary object (object B) was occluded, and figure 15d shows results for when the post-collision velocity of the initially moving object (object A) was occluded. To simulate these experiments with PC/BC-DIM, the input corresponding to the occluded velocity was not set to zero (in contrast to when simulating the experiments with occluded initial velocities). Instead, the occluded final velocities were set equal to the pre-collision velocities, but the strength of these inputs was made weaker (20% of the amplitude of the pre-collision inputs). The model thus makes the assumption that when an object disappears its velocity is weakly perceived to remain unchanged. For both the human subjects and the PC/BC-DIM model, when the collision has a low coefficient of restitution the disappearance of either object post-collision has relatively little effect on the judgement of relative mass. In contrast, for collisions with a high coefficient of restitution the two occlusion conditions have more significant, and opposite, effects on the perceived masses of the objects: occluding object B increases the likelihood that object B is perceived as being more massive, whereas occluding object A increases the likelihood that object A is perceived as being heavier.

The method of setting the weights in the PC/BC-DIM network has encoded knowledge about possible collisions in the prediction neuron RFs. In all the above experiments, the velocity information causes a subset of prediction neurons to become active. These active prediction neurons are those with RFs most consistent with the



(a)

(b)

Figure 16: Predicted post-collision velocities as a function of the relative masses of the two objects. Top row shows the expected velocities calculated using the equation for the conservation of momentum, the bottom row shows the corresponding estimates of velocity found by the PC/BC-DIM network. In each sub-figure, the coefficient of restitution (ϵ) used in the experiment is indicated by the line marker shape: diamond markers for ϵ=0.9, square markers for ϵ=0.5, circle markers for ϵ=0.1. Closed markers are used to plot the post-collision velocity of object A and open markers are used for object B. (a) The initial velocity of object A is 3.18, and the initial velocity of object B is -3.18; as in Todd and Warren (1982, expt. 2 moving condition), and figure 14b. (b) The initial velocity of object A is 3.18, and object B is initially stationary; as in Todd and Warren (1982, expt. 2 stationary condition), and figure 14c.

given velocities. In many cases a majority of the active prediction neurons will connect to the same reconstruction neuron in the last partition, and hence, cause the network to have a strong belief that object A is more massive, or that object B is more massive, or that they have the same weight. In other conditions, the combination of velocities may be consistent with possible collisions in which object A is heavier as well as collisions in which object B is heavier. In these cases, the active prediction neurons will cause activation of the last partition of the reconstruction neurons that is more ambiguous, but still lead to little preference for choosing object A as the heavier one.

In the preceding experiments, information about velocity has been presented to the PC/BC-DIM network, and an estimate of the relative mass of the two objects has been read out from the last partition of reconstruction neurons. It is also possible to input the relative mass and the coefficient of restitution together with some velocities and have the network estimate the missing velocities. For example, it is possible to input the initial velocities of the two objects, their relative masses and the coefficient of restitution[2] and have the PC/BC-DIM network predict the final velocities of the objects after collision. Results for such an experiment are shown in figure 16. It can be seen that the PC/BC-DIM network produces reasonable estimates of the post-collision velocities of both objects, however, these estimates tend to be closer to those that would have been produced by equally weighted objects *i.e.*, the network underestimate the influence of differences in mass on the final velocity.

[2] The network can estimate the coefficient of restitution from observing a previous colllision between the same objects.

4 Discussion

Predictive coding models are often implemented as neural networks (*e.g.*, De Meyer and Spratling, 2011; Jehee and Ballard, 2009; Jehee et al., 2006; Rao and Ballard, 1999; Spratling, 2010; Wacongne et al., 2012). However, they are specific neural network architectures that are constrained in numerous ways compared to neural networks in general. For example, in the PC/BC-DIM model used here, the mathematical functions performed by the neurons are defined (equations 1– 3) to implement predictive coding and can not be changed in order to match the behaviour of the network to the data being simulated. Similarly, all the inputs to a PC/BC-DIM network are treated identically, and it is not possible to assign different roles to different inputs (for example, to define some inputs as being modultory or inhibitory) in order to perform different tasks. The connectivity between the neurons is also prescribed so that it is not possible to add additional connections (for example, from the reconstruction neurons to the prediction neurons, or between different neurons in the prediction neuron population, or between neurons in the reconstruction population) in order to reproduce the desired behaviours. Furthermore, the neurons are all constrained to have non-negative firing rates and the connections are all defined to have non-negative synaptic weights. While it is well known that a neural network can be defined to simulate any data or perform any computation, the constraints placed on PC/BC-DIM mean that it is not certain that it has the flexibility to simulate any behaviour. The aim of this work was, therefore, to demonstrate that predictive coding (at least the PC/BC-DIM version of predictive coding) is capable of simulating a range of cognitive abilities, and hence, to provide more concrete support for previous speculation about the possible role of of predictive coding in cognition.

The simulations described here necessarily cover only a small subset of abilities that are relevant to cognition, but they were chosen to be illustrative and to cover a range of behaviours that, on the surface, appear to have little in common. A single modelling method, PC/BC-DIM, was shown to be able to account for all these diverse cognitive abilities. Added to this, PC/BC-DIM has previously been shown to account for a very large range of perceptual abilities and the neural mechanisms that underlie them (De Meyer and Spratling, 2011, 2013; Spratling, 2008a, 2010, 2011, 2012a,b,c, 2013b, 2014c). As PC/BC-DIM is a particular implementation of predictive coding (Clark, 2013; Huang and Rao, 2011; Rao and Ballard, 1999; Spratling, 2014b) the current results suggest that a single computational principle, the predictive coding principle, could underlie functions ranging from low-level perceptual processes to high-level cognitive abilities. What other phenomena can, or can not, be simulated using PC/BC-DIM, and hence, the limits of predictive coding as an explanation of brain function remain to be explored in future work.

Here, it has been shown, in principle, that a single mechanism can account for a wide range of cognitive phenomena. However, the models presented here have been hard-wired. This leaves unanswered the arguably more difficult question of how the brain is wired-up to perform these tasks? There are at least two issues here. Firstly, given a set of inputs to a PC/BC-DIM network, how could appropriate synaptic weights be learnt to perform the task? For example, in a classification task, given input vectors that represent the stimulus features and the associated class labels for a number of exemplars, how could appropriate synaptic weights be learnt in order to perform classification? Previous work that has addressed learning in PC/BC-DIM networks (De Meyer and Spratling, 2011; Spratling, 2012c; Spratling et al., 2009), and other work that has derived biologically-plausible learning rules from formal, information theoretic, principles in a closely related model (Kay and Phillips, 1997, 2011), provide grounds for optimism concerning the possibility of solving the first issue, but additional work is required to show that learning could be used to wire-up networks like those described in this article. Furthermore, as tasks become more complex and the corresponding networks become larger, it becomes impractical to hand-design the connection weights (as is also the case for deep neural network architectures). Solving the first issue is therefore important to enable the simulation of more complex tasks.

The second issue, is how do the inputs get to right places in the first place? For example, in a classification task, how could the inputs representing the appropriate stimulus features and the class labels all get routed to the input of a single PC/BC-DIM network? As a second example, consider the task proposed by (Salinas, 2004b) for which the simulation results are shown in Fig. 13. Here representations of fixation point colour, stimulus orientation and saccade position all need to be brought together so that the appropriate context-dependent mapping can be learnt. Such stimulus-task mappings are entirely arbitrary, for example, we could define a task in which the context was defined by shape (rather than colour) and the stimulus was defined by a sound (rather than the visual appearance) and where the appropriate response was to press a button (rather than perform a saccade). How such an arbitrary array of sensory-motor representations can be brought together to allow rapid learning of the task is a hard question to answer. This is another question that needs to be addressed by future work. To help address this question, a single PC/BC-DIM network (Fig. 1a) can form a building block that can be plugged together with other PC/BC-DIM networks (one simple example is shown in Fig 7) to create large-scale PC/BC-DIM networks that can be used both to simulate systems-level models of cortex and be used to explore learning and self-organisation in such networks.

PC/BC-DIM is an abstract, mathematical, model that aims to explore the computational, rather than the biophysiological, mechanisms which underlie cortical function (Spratling, 2011). However, it is possible to speculate about the potential biological implementation of the model. There are many different ways in which the simple circuitry of PC/BC-DIM model could potentially be implemented in the much more complex circuitry of the cortex (Spratling, 2008b, 2011, 2012b, 2013a). However, the most straightforward explanation would equate prediction neurons with the sub-population of cortical pyramidal cells (mostly found in cortical layers II and III) whose axon projections form the feedforward connections between cortical regions, and to equate reconstruction neurons with the sub-population of cortical pyramidal cells (mostly found in cortical layer VI) whose axon projections form the feedback connections between cortical regions. This is consistent with the connectivity between PC/BC-DIM processing stages illustrated in figure 7, and with previous work showing that the behaviour of prediction neurons can explain the response properties of cortical pyramidal cells (De Meyer and Spratling, 2011; Spratling, 2010, 2011, 2012a). It is possible to equate the error-detecting neurons with the spiny-stellate cells in cortical layer IV, which are the major targets of cortical feedforward connections and sensory inputs. However, it is also possible that the error-detection is performed in the dendrites of the superficial layer pyramidal cells (Spratling and Johnson, 2003) rather than in a separate neural population; or via synaptic depression (Rothman et al., 2009) which can produce the specific form of divisive inhibition required by the error-neurons in the PC/BC-DIM model; or that the error neurons reside in the thalamus, individual regions of which receive connections from layer VI pyramidal cells (putative reconstruction neurons) as well as either sensory input or input from lower cortical regions.

Acknowledgements

Thanks to the organisers of, and the participants at, the Lorentz Centre Workshop on Perspectives on Human Probabilistic Inference (May 2014) for discussions that inspired this work. Additional thanks to Bill Phillips for helpful comments on an earlier draft of this paper.

References

Aha, D. W. and Goldstone, R. L. (1992). Concept learning and flexible weighting. In *Proceedings of the 14th annual conference of the Cognitive Science Society*, pages 534–9.

Alink, A., Schwiedrzik, C. M., Kohler, A., Singer, W., and Muckli, L. (2010). Stimulus predictability reduces responses in primary visual cortex. *The Journal of Neuroscience*, 30:2960–6.

Anderson, J. R. (1991). The adaptive nature of human categorization. *Psychological Review*, 98:409–29.

Anderson, J. R. and Betz, J. (2001). A hybrid model of categorization. *Psychonomic Bulletin and Review*, 8(4):629–47.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4):1036–60.

Apps, M. A. J. and Tsakiris, M. (2014). The free-energy self: A predictive coding account of self-recognition. *Neuroscience and Biobehavioral Reviews*, 41:85–97.

Ashby, F. G., Alfonso-Reese, L. A., Turken, A. U., and Waldron, E. M. (1998). A neuropsychological theory of multiple systems in category learning. *Psychological Review*, 107:442–81.

Bechtel, W. and Abrahamsen, A. (1991). *Connectionism and the Mind: An Introduction to Parallel Processing in Networks*. Basil Blackwell, Oxford.

Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–55.

Bubic, A., von Cramon, D. Y., and Schubotz, R. I. (2010). Prediction, cognition and the brain. *Frontiers in Human Neuroscience*, 4(25):1–15.

Chaaban, I. and Scheessele, M. R. (2007). Human performance on the USPS database. Technical report, Indiana University, South Bend, Indiana.

Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(03):181–204.

Cohen, M. R. and Newsome, W. T. (2008). Context-dependent changes in functional circuitry in visual area MT. *Neuron*, 60(1):162–73.

de Cruys, S. V. and Wagemans, J. (2011). Putting reward in art: A tentative prediction error account of visual art. *i-Perception*, 2:1035–62.

De Meyer, K. and Spratling, M. W. (2011). Multiplicative gain modulation arises through unsupervised learning in a predictive coding model of cortical function. *Neural Computation*, 23(6):1536–67.

De Meyer, K. and Spratling, M. W. (2013). A model of partial reference frame transforms through pooling of gain-modulated responses. *Cerebral Cortex*, 23(5):1230–9.

Deneve, S. (2008). Bayesian spiking neurons I: Inference. *Neural Computation*, 20(1):91–117.

Denison, R. N., Piazza, E. A., and Silver, M. A. (2011). Predictive context influences perceptual selection during binocular rivalry. *Frontiers in Human Neuroscience*, 5(166).

Egner, T., Monti, J. M., and Summerfield, C. (2010). Expectation and surprise determine neural population responses in the ventral visual stream. *The Journal of Neuroscience*, 30(49):16601–8.

Erickson, M. A. and Kruschke, J. K. (1998). Rules and exemplars in category learning. *Journal of Experimental Psychology: General*, 127(2):107–40.

Friston, K. J. (2005). A theory of cortical responses. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 360(1456):815–36.

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

Garrido, M. I., Kilner, J. M., Kiebel, S. J., and Friston, K. J. (2009). Dynamic causal modeling of the response to frequency deviants. *Journal of Neurophysiology*, 101:2620–31.

Georghiades, A. S., Belhumeur, P. N., and Kriegman, D. J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–60.

Gilden, D. L. and Proffitt, D. R. (1989). Understanding collision dynamics. *Journal of Experimental Psychology: Human Perception and Performance*, 15(2):372–83.

Goldstone, R. L. and Kersten, A. (2003). Concepts and categorization. In Healy, A. F. and Proctor, R. W., editors, *Comprehensive Handbook of Psychology*, volume 4, pages 599–621. Wiley, NJ, USA.

Hohwy, J., Roepstorff, A., and Friston, K. (2008). Predictive coding explains binocular rivalry: An epistemological review. *Cognition*, 108(3):687–701.

Hosoya, T., Baccus, S. A., and Meister, M. (2005). Dynamic predictive coding by the retina. *Nature*, 436(7047):71–7.

Huang, Y. and Rao, R. P. N. (2011). Predictive coding. *WIREs Cognitive Science*, 2:580–93.

Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–4.

Jehee, J. F. M. and Ballard, D. H. (2009). Predictive feedback can account for biphasic responses in the lateral geniculate nucleus. *PLoS Computational Biology*, 5(5):e1000373.

Jehee, J. F. M., Rothkopf, C., Beck, J. M., and Ballard, D. H. (2006). Learning receptive fields using predictive feedback. *Journal of Physiology – Paris*, 100:125–32.

Kay, J. and Phillips, W. A. (1997). Activation functions, computational goals and learning rules for local processors with contextual guidance. *Neural Computation*, 9(4):895–910.

Kay, J. W. and Phillips, W. A. (2011). Coherent infomax as a computational goal for neural systems. *Bulletin of Mathematical Biology*, 73:344–72.

Kilner, J. M., Friston, K. J., and Frith, C. D. (2007). Predictive coding: an account of the mirror neuron system. *Cognitive Processing*, 8(3):159–66.

Koster-Hale, J. and Saxe, R. (2013). Theory of mind: A neural prediction problem. *Neuron*, 79(5):836–48.

Kruschke, J. K. (1992). ALCOVE: an exemplar-based connectionist model of category learning. *Psychological Review*, 99(1):22–44.

Kruschke, J. K. (2005). Category learning. In Lamberts, K. and Goldstone, R. L., editors, *The Handbook of Cognition*, pages 183–201. Sage, London, UK.

Kumaran, D. and McClelland, J. L. (2012). Generalization through the recurrent interaction of episodic memories: A model of the hippocampal system. *Psychological Review*, 119(3):573–616.

Lalanne, L., van Assche, M., and Giersch, A. (2010). When predictive mechanisms go wrong: Disordered visual synchrony thresholds in schizophrenia. *Schizophrenia Bulletin*, 38(3):506–13.

Laughlin, S. (1990). Coding efficiency and visual processing. In Blakemore, C., editor, *Vision: Coding and Efficiency*, chapter 2, pages 25–31. Cambridge University Press.

Lawson, R. P., Rees, G., and Friston, K. J. (2014). An aberrant precision account of autism. *Frontiers in Human Neuroscience*, 8(302).

LeCun, Y., Kavukvuoglu, K., and Farabet, C. (2010). Convolutional networks and applications in vision. In *Procedings of the International Symposium on Circuits and Systems (ISCAS10)*. IEEE.

Lee, K. C., Ho, J., and Kriegman, D. (2005). Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–98.

Lee, T. S. (2015). The visual system's internal model of the world. *Proceedings of the IEEE*, 103(8):1359–1378.

Love, B. C., Medin, D. L., and Gureckis, T. M. (2004). SUSTAIN: A network model of category learning.

Psychological Review, 111:309–32.

McClelland, J. L. (2013). Integrating probabilistic models of perception and interactive neural networks: A historical and tutorial review. *Frontiers in Psychology*, 4(503).

McClelland, J. L. (2014). *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*. https://web.stanford.edu/group/pdplab/pdphandbook/, 2nd edition.

McClelland, J. L. and Johnston, J. C. (1977). The role of familiar units in perception of words and nonwords. *Perception and Psychophysics*, 22(3):249–61.

McClelland, J. L., Mirman, D., Bolger, D. J., and Khaitan, P. (2014). Interactive activation and mutual constraint satisfaction in perception and cognition. *Cognitive Science*, 38:1139–89.

McClelland, J. L. and Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. an account of basic findings. *Psychological Review*, 88:375–407.

McClelland, J. L., Rumelhart, D. E., and Hinton, G. E. (1986). the appeal of parallel distributed processing. In Rumelhart, D. E., McClelland, J. L., and The PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Volume 1: Foundations*, chapter 1, pages 3–44. MIT Press, Cambridge, MA.

Medin, D. L. and Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, 85:207–38.

Nosofsky, R. M., Clark, S. E., and Shin, H. J. (1989). Rules and exemplars in categorization, identification, and recognition. *Journal of Experimental Psychology: Learning, memory, and cognition*, 15(2):282–304.

Nosofsky, R. M. and Johansen, M. K. (2000). Exemplar-based accounts of multiple-system phenomena in perceptual categorization. *Psychonomic Bulletin and Review*, 7(3):375–402.

Nosofsky, R. M., Palmeri, T. J., and McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychological Review*, 101(1):53–79.

Pouget, A. and Sejnowski, T. J. (1997). Spatial transformations in the parietal cortex using basis functions. *Journal of Cognitive Neuroscience*, 9(2):222–37.

Ramaswami, M. (2014). Network plasticity in adaptive filtering and behavioral habituation. *Neuron*, 82(6):1216–29.

Rao, R. P. N. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87.

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–25.

Rothman, J., Cathala, L., Steuber, V., and Silver, R. A. (2009). Synaptic depression enables neuronal gain control. *Nature*, 457:1015–8.

Rumelhart, D. E. and McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. the contextual enhancement effect and some tests and extensions of the model. *Psychological Review*, 89(1):60.

Rumelhart, D. E., McClelland, J. L., and The PDP Research Group, editors (1986). *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Volume 1: Foundations*. MIT Press, Cambridge, MA.

Runeson, S. and Vedeler, D. (1993). The indispensability of precollision kinematics in the visual perception of relative mass. *Perception and Psychophysics*, 53(6):617–632.

Salinas, E. (2004a). Context-dependent selection of visuomotor maps. *BMC Neuroscience*, 5:47.

Salinas, E. (2004b). Fast remapping of sensory stimuli onto motor actions on the basis of contextual modulation. *The Journal of Neuroscience*, 24(5):1113–8.

Sanborn, A. (2014). Testing bayesian and heuristic predictions of mass judgments of colliding objects. *Frontiers in Psychology*, 5(938).

Sanborn, A. N., Griffiths, T. L., and Navarro, D. J. (2010). Rational approximations to rational models: Alternative algorithms for category learning. *Psychological Review*, 117(4):1144–67.

Sanborn, A. N., Mansinghka, V. K., and Griffiths, T. L. (2013). Reconciling intuitive physics and newtonian mechanics for colliding objects. *Psychological Review*, 120(2):411–37.

Selfridge, O. G. (1955). Pattern recognition and modern computers. In *Proceedings of the western joint computer conference*, pages 91–93. ACM.

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–26.

Seth, A. K. (2013). Interoceptive inference, emotion, and the embodied self. *Trends in Cognitive Sciences*, 17(11):656–63.

Seth, A. K., Suzuki, K., and Critchley, H. D. (2011). An interoceptive predictive coding model of conscious presence. *Frontiers in Psychology*, 2(395).

Smith, E. E., Patalano, A. L., and Jonides, J. (1998). Alternative strategies of categorization. *Cognition*, 65:167–96.

Smith, F. W. and Muckli, L. (2010). Nonstimulated early visual areas carry information about surrounding context. *Proceedings of the National Academy of Sciences USA*, 107(46):20099–103.

Smith, J. D. and Minda, J. P. (1998). Prototypes in the mist: The early epochs of category learning. *Journal of Experimental Psychology: Learning, memory, and cognition*, 24:1411–36.

Spratling, M. W. (2008a). Predictive coding as a model of biased competition in visual selective attention. *Vision Research*, 48(12):1391–408.

Spratling, M. W. (2008b). Reconciling predictive coding and biased competition models of cortical function. *Frontiers in Computational Neuroscience*, 2(4):1–8.

Spratling, M. W. (2010). Predictive coding as a model of response properties in cortical area V1. *The Journal of Neuroscience*, 30(9):3531–43.

Spratling, M. W. (2011). A single functional model accounts for the distinct properties of suppression in cortical area V1. *Vision Research*, 51(6):563–76.

Spratling, M. W. (2012a). Predictive coding accounts for V1 response properties recorded using reverse correlation. *Biological Cybernetics*, 106(1):37–49.

Spratling, M. W. (2012b). Predictive coding as a model of the V1 saliency map hypothesis. *Neural Networks*, 26:7–28.

Spratling, M. W. (2012c). Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. *Neural Computation*, 24(1):60–103.

Spratling, M. W. (2013a). Distinguishing theory from implementation in predictive coding accounts of brain function [commentary]. *Behavioral and Brain Sciences*, 36(3):231–2.

Spratling, M. W. (2013b). Image segmentation using a sparse coding model of cortical area V1. *IEEE Transactions on Image Processing*, 22(4):1631–43.

Spratling, M. W. (2014a). Classification using sparse representations: a biologically plausible approach. *Biological Cybernetics*, 108(1):61–73.

Spratling, M. W. (2014b). Predictive coding. In Jaeger, D. and Jung, R., editors, *Encyclopedia of Computational Neuroscience*, pages 1–5. Springer, New York, NY.

Spratling, M. W. (2014c). A single functional model of drivers and modulators in cortex. *Journal of Computational Neuroscience*, 36(1):97–118.

Spratling, M. W. (2016). A neural implementation of bayesian inference based on predictive coding. *submitted*.

Spratling, M. W., De Meyer, K., and Kompass, R. (2009). Unsupervised learning of overlapping image components using divisive input modulation. *Computational Intelligence and Neuroscience*, 2009(381457):1–19.

Spratling, M. W. and Johnson, M. H. (2003). Exploring the functional significance of dendritic inhibition in cortical pyramidal cells. *Neurocomputing*, 52-54:389–95.

Srinivasan, M. V., Laughlin, S. B., and Dubs, A. (1982). Predictive coding: A fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 216(1205):427–59.

Summerfield, C. and Egner, T. (2009). Expectation (and attention) in visual cognition. *Trends in Cognitive Sciences*, 13(9):403–9.

Todd, J. T. and Warren, W. H. (1982). Visual perception of relative mass in dynamic events. *Perception*, 11:325–35.

van Boxtel, J. J. A. and Lu, H. (2013). A predictive coding perspective on autism spectrum disorders: a general comment on Pellicano and Burr (2012). *Frontiers in Psychology*, 4(19).

Vicovaro, M. and Burigana, L. (2014). Intuitive understanding of the relation between velocities and masses in simulated collisions. *Visual Cognition*, 22(7):896–919.

Wacongne, C., Changeux, J.-P., and Dehaene, S. (2012). A neuronal model of predictive coding accounting for the mismatch negativity. *The Journal of Neuroscience*, 32(11):3665–78.