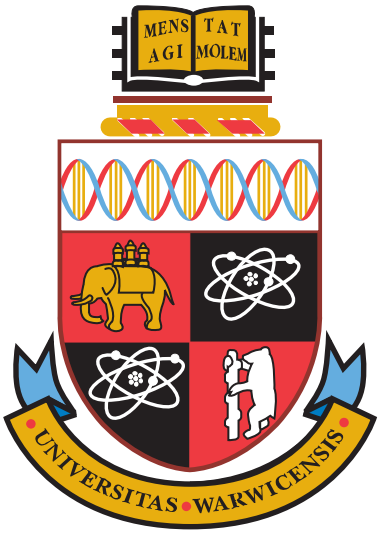


Invariant Object Recognition

Biologically Plausible and Machine Learning Approaches

by

Leigh Robinson MSc, MEng.

Submitted to the University of Warwick for the degree of
Doctor of Philosophy

Complexity Science Doctoral Training Centre
&
Department of Computer Science

2015

THE UNIVERSITY OF
WARWICK

Contents

List of Figures

List of Tables

Acknowledgements

Publications

Abstract

I Introduction

Introduction

Modelling Vision: Object Recognition

2.1 Introduction 5
2.2 Neurophysiology of Vision 5
2.3 Computational Models of Vision 8
2.3.1 Structural Models 9
2.3.2 View Models 9

II Biologically Plausible Approaches

Biological plausibility of VisNet and HMAX

3.1 Introduction 19
3.2 Methods 21
3.2.1 Overview of the architecture of the ventral visual stream model, VisNet . . 21

3.2.2 The network implemented in VisNet . . . 21
3.2.3 Competition and lateral inhibition in VisNet . . . 22
3.2.4 The VisNet trace learning rule . . . 24
3.2.5 The input to VisNet . . . 26
3.2.6 Recent developments in VisNet implemented in the research described here 27
3.2.7 The HMAX models used for comparison with VisNet . . . 27
3.2.8 Measures for network performance . . . 29
3.3 Results . . . 32
3.3.1 Categorization of objects from benchmark object image sets: Experiment 1 32
3.3.2 Performance with the Amsterdam Library of Images: Experiment 2 . . . 38
3.3.3 The effects of rearranging the parts of an object: Experiment 3 . . . 45
3.3.4 View invariant object recognition: Experiment 4 . . . 48
3.4 Discussion . . . 52
3.4.1 Overview of the findings on how well the properties of inferior temporal cortex neurons were met, and discussion of their significance . . . 52
3.4.2 The training method . . . 54
3.4.3 Representations of the spatial configurations of the parts . . . 55
3.4.4 Object representations invariant with respect to catastrophic view transformations . . . 55
3.4.5 How VisNet solves the computational problems of view invariant representations . . . 55
3.4.6 The approach taken by HMAX . . . 57
3.4.7 Some other approaches to invariant visual object recognition . . . 59
3.4.8 The properties of inferior temporal cortex neurons that need to be addressed by models in visual invariant object recognition . . . 59
3.4.9 Comparison with computer vision approaches to not only classification of objects but also to identification of the individual . . . 60
3.4.10 Outlook: some properties of inferior temporal cortex neurons that need to be addressed by models of ventral visual stream visual invariant object recognition 61
3.4.11 Conclusions . . . 62

III Machine Learning Approaches

Confusing Convolutional Networks

4.1 Introduction . . . 67
4.2 Implementation . . . 67
4.3 Image relabelling . . . 68
4.3.1 How robust is this process? . . . 69
4.3.2 Gaussian image synthesis . . . 70
4.4 Feature analysis . . . 71
4.5 Discussion . . . 72

Efficient Batchwise Dropout

5.1 Introduction . . . 73
5.2 Independent dropout . . . 73
5.3 Batchwise dropout . . . 74
5.4 Implementation . . . 76
5.4.1 Efficiency considerations . . . 77
5.5 Results for fully-connected networks . . . 78
5.5.1 MNIST . . . 78
5.5.2 CIFAR-10 fully-connected . . . 79
5.5.3 Artificial dataset . . . 79
5.6 Results for convolutional networks . . . 80
5.6.1 MNIST . . . 81
5.6.2 CIFAR-10 with varying dropout intensity . . . 82
5.6.3 CIFAR-10 with many convolutional layers . . . 82
5.7 Discussion . . . 83
5.7.1 Fast dropout . . . 84
5.7.2 Future Work . . . 84

Locally Connected Deep Belief Networks

6.1 Introduction . . . 87
6.2 Deep Belief Network Architecture . . . 88
6.2.1 Restricted Boltzmann Machines . . . 88
6.2.2 Contrastive Divergence Learning . . . 89
6.2.3 Deep Belief Networks . . . 92
6.3 Extending DBNs with local connectivity . . . 92
6.4 Results . . . 93
6.4.1 MNIST classification experiments . . . 93
6.4.2 CIFAR-10 Orientation Maps . . . 94
6.5 Discussion . . . 98
6.5.1 Future Work . . . 98

IV Discussion & Conclusions

Discussion & Conclusions

7.1 Introduction . . . 103
7.1.1 Thesis Contributions . . . 103
7.2 Discussion . . . 104

V Bibliography

Bibliography

List of Figures

Figure 2.1 Caricature of the regions of the visual cortex and their major interconnections. The dorsal areas, MT (medial temporal), MST (medial superior temporal) and FST (fundus of superior temporal suculus) seem to be primarily concerned with motion and the location of objects. The red (bold) feed-forward ventral pathway is where the majority of object recognition processes are thought to take place. Adapted from Gross et al. (1993) 6

Figure 2.2 Examples of features identified by Kobatake and Tanaka (1994) grouped by the region of the visual hierarchy. Note that the stimuli become more visually complex as you traverse the ventral stream. (After Kobatake and Tanaka (1994)). 8

Figure 2.3 Examples of various symmetric (bottom) and anti-symmetric (top) Gabor filters at various scales (horizontal progression). 11

Figure 2.4 Gabor filters of various scales and orientations are applied to a sample image. 11

Figure 3.1 Convergence in the visual system. Right – as it occurs in the brain. V1, visual cortex area V1; TEO, posterior inferior temporal cortex; TE, inferior temporal cortex (IT). Left – as implemented in VisNet. Convergence through the network is designed to provide fourth layer neurons with information from across the entire input retina. 22

Figure 3.2 Sketch of the HMAX model of invariant object recognition (Riesenhuber and Poggio 1999a, 2000a). The model includes layers of 'S' cells which perform template matching (solid lines), and 'C' cells (solid lines) which pool information by a non-linear MAX function to achieve invariance (see text). (After (Riesenhuber and Poggio 1999a)) 29

Figure 3.3 Example images from the Caltech256 database for two object classes, hats and beer mugs. 33

Figure 3.4 Performance of HMAX and VisNet on the classification task (measured by the proportion of images classified correctly) using the Caltech-256 dataset and linear support vector machine (SVM) classification. The error bars show the standard deviation of the means over 3 cross-validation trials with different images chosen at random for the training set on each trial. There were 2 object classes, hats and beer-mugs, with the number of training exemplars shown on the abscissa. There were 30 test examples of each object class. All cells in the C2 layer of HMAX and layer 4 of VisNet were used to measure the performance. Chance performance at 50% is indicated. 34

Figure 3.5 Top: Firing rate of two output layer neurons of VisNet, when tested on two of the classes, hats and beer mugs, from the Caltech 256. The firing rates to 10 untrained (i.e. testing) exemplars of each of the two classes are shown. One of the neurons responded more to hats than to beer mugs (solid line). The other neuron responded more to beer mugs than to hats (dashed line). Middle: Firing rate of two C2 Tuned Units of HMAX when tested on two of the classes, beer mugs and hats, from the Caltech 256. Bottom: Firing rate of a View Tuned Unit of HMAX when tested on two of the classes, hats (solid line) and beer mugs (dashed line), from the Caltech 256. The neurons chosen were those with the highest single cell information that could be decoded from the responses of a neuron to 10 exemplars of each of the 2 objects (as well as a high firing rate) in the cross-validation design. 37

Figure 3.6 Example images from the two object classes within the ALOI database, (a) 293 (light bulb) and (b) 156 (clock). Only the 45 degree increments are shown. 39

Figure 3.7 Performance of VisNet and HMAX C2 units measured by the percentage of images classified correctly on the classification task with 8 objects using the Amsterdam Library of Images dataset and measurement of performance using a pattern association network with one output neuron for each class. The training set consisted of 4 views of each object spaced 90 degrees apart; or 9 views spaced 40 degrees apart; or 18 views spaced 20 degrees apart. The test set of images was in all cases a cross-validation set of 18 views of each object spaced 20 degrees apart and offset by 10 degrees from the training set with 18 views and not including any training view. The 10 best cells from each class were used to measure the performance. Chance performance was 12.5% correct. 40

Figure 3.8 Top: Firing rate of one output layer neuron of VisNet, when trained on 8 objects from the Amsterdam Library of Images, with 9 views of each object spaced 40 degrees apart. The firing rates on the training set are shown. The neuron responded to all 9 views of object 4 (a light bulb), and to no views of any other object. The neuron illustrated was chosen to have the highest single cell stimulus-specific information about object 4 that could be decoded from the responses of the neurons to all 72 exemplars shown, as well as a high firing rate to object 4. Middle: Firing rate of one C2 Unit of HMAX when trained on the same set of images. The unit illustrated was that the highest mean firing rate across views to object 1 relative to the firing rates across all stimuli and views. Bottom: Firing rate of one View Tuned Unit (VTU) of HMAX when trained on the same set of images. The unit illustrated was that the highest firing rate to view 1 of object 1. 42

Figure 3.9 Top: Firing rate during cross-validation testing of one output layer neuron of VisNet, when trained on 8 objects from the Amsterdam Library of Images, with 9 exemplars of each object with views spaced 40 degrees apart. The firing rates on the cross-validation testing set are shown. The neuron was selected to respond to all views of object 4 of the training set, and as shown responded to 7 views of object 4 in the test set each of which was 20 degrees from the nearest training view, and to no views of any other object. Middle: Firing rate of one C2 Unit of HMAX when tested on the same set of images. The neuron illustrated was that the highest mean firing rate across training views to object 1 relative to the firing rates across all stimuli and views. The test images were 20 degrees away from the test images. Bottom: Firing rate of one View Tuned Unit (VTU) of HMAX when tested on the same set of images. The neuron illustrated was that the highest firing rate to view 1 of object 1 during training. It can be seen that the neuron responded with a rate of 0.8 to the two training images (1 and 9) of object 4 that were 20 degrees away from the image for which the VTU had been selected. 43

Figure 3.10 Similarity between the outputs of the networks between the 9 different views of 8 objects produced by VisNet (top), HMAX C2 (middle), and HMAX VTUs (bottom) for the Amsterdam Library of Images test. Each panel shows a similarity matrix (based on the cosine of the angle between the vectors of firing rates produced by each object) between the 8 stimuli for all output neurons of each type. The maximum similarity is 1, and the minimal similarity is 0. 44

Figure 3.11 Examples of images used in the scrambled faces experiment. Top: Two of the 8 faces in 2 of the 5 views of each. Bottom: examples of the scrambled versions of the faces. 45

Figure 3.12 Top. Effect of scrambling on the responses of a neuron in VisNet. This VisNet layer 4 neuron responded to one of the faces after training, and to none of the other 7 faces. The neuron responded to all the different view exemplars 1–5 of the unscrambled face (exemplar normal). When the same neuron was then tested with the randomly scrambled versions of the same face stimuli (exemplar scrambled), the firing rate was zero. Bottom. Effect of scrambling on the responses of a neuron in HMAX. This View Tuned Neuron of HMAX was chosen to be as discriminating between the 8 face identities as possible. The neuron responded to all the different view exemplars 1–5 of the unscrambled face. When the same neuron was then tested with the randomly scrambled versions of the same face stimuli, the neuron responded with similarly high rates to the scrambled stimuli. 47

Figure 3.13 View invariant representations of cups. The two objects, each with four views. 48

Figure 3.14 Top. View invariant representations of cups. Single cells in the output representation of VisNet. The two neurons illustrated responded either to all views of one cup (labelled 'Bill') and to no views of the other cup (labelled 'Jane'), or vice versa. Middle. Single cells in the C2 representation of HMAX. Bottom. Single cells in the View Tuned Unit output representation of HMAX. 51

Figure 4.1 This figure shows the appearance of the gradients that are computed on the first iteration when relabelling an image from class 10, to class 348. Notice that the for the gradient passed through the sgn function (4.1(b)) the values are much more spread out spatially, and each colour channel are either -1/255, 0 or 1/255. This stops "hot spots" appearing in the image from the large clustered values apparent in 4.1(a) when used in the gradient descent. 69

Figure 4.2 Examples of class relabelling on GoogLeNet. Left: Original images that are correctly classified as 'Shih-Tzu' and 'Half-track'. Right: relabelled images such that the classification output is reversed - i.e the 'Shih-Tzu' is now strongly classified as 'Half-track' and vice-versa. Centre: Pixel differences multiplied by 10 and scaled to mean-level for visibility. The distortion introduced in the top relableling was 0.00864 and the bottom 0.00712. These are a random pair of images taken from the above classes that were chosen to be qualitatively "maximally different". As we show in Section 4.3.1 relabelling is not sensitive to the initial or target class. 70

Figure 4.3 The class probabilities for each of the four images in Fig 4.2. Notice that after relabelling the uncertainty is also reduced. 71

Figure 4.4 These figures show a random gaussian image (4.4(a)) and the same image after being re-labeled to class 348 (4.4(b)). The bottom two graphs show the class probabilities before and after the relabelling process. The distortion introduced in this relabelling process was 0.03 - which is much higher than typically required to relabel natural images. 72

Figure 5.1 Example of applying dropout to a fully connected neural network 74

Figure 5.2 Left: MNIST training time for three layer networks (log scales) on an NVIDIA GeForce GTX 780 graphics card. Right: Percentage reduction in training times moving from no dropout to batchwise dropout. The time saving for the 500N network with minibatches of size 100 increases from 33% to 42% if you instead compare batchwise dropout with independent dropout. 77

Figure 5.3 Dropout networks trained using a restricted the number of dropout patterns (each $\times$ is from an independent experiment). The blue line marks the test error for a network with half as many hidden units trained without dropout. 79

Figure 5.4 Results comparing the performance of independent and batchwise dropout on the CIFAR-10 dataset using fully-connected networks of different sizes. 80

Figure 5.5 Results comparing the performance of independent and batchwise dropout on the artificial dataset using networks of different sizes. 100 classes each corresponding to noisy observations of a one dimensional manifold in $\{0,1\}^{1000}$. 81

Figure 5.6 MNIST test errors, training repeated three times for both dropout methods. . 82

Figure 5.7 CIFAR-10 results using a convolutional network with dropout probability $p \in (0, 0.4)$. Batchwise dropout produces a slightly lower minimum test error. 83

Figure 6.1 The general structure of a Restricted Boltzmann Machine (RBM) is comprised of a bipartite graph of nodes which are labelled visible, $\mathbf{v}$ and hidden $\mathbf{h}$. The pair-wise interactions of nodes between layers is defined by a connectivity matrix, $\mathbf{W}$. 88

Figure 6.2 This diagram shows one step contrastive divergence (CD) learning. At time $t = 0$ we sample the states of the hidden units having clamped the visible units to an exemplar in the dataset. At time $t = 1$ we then sample the visible units given the previous hidden node samples which in turn allows a re-sampling of the hidden units. This iterative process can be computed more times to get better estimates, but commonly one is sufficient for gradient estimates good enough to learn with. 91
Figure 6.3 The MNIST handwritten digit database. The clean images (a) and the four corrupted versions: random noise (b), rotations (c), occlusions (d) and translations (e). . . . 93
Figure 6.4 Error rates of DBN models with various receptive field sizes when tasked to classify MNIST images corrupted with increasing amounts of noise. Notice that while the fully connected model (image sized receptive fields - 28x28) very quickly degrades to random chance, the others do not. 94
Figure 6.5 Error rates of LCDBN models with various receptive field sizes when tasked to classify MNIST images corrupted with translations (a), rotations (b) and occlusions (c). Unlike with noise corruption (Fig 6.4) in these cases local connectivity does not seem to offer any benefits. 95
Figure 6.6 Example images drawn from four of the classes from the CIFAR-10 dataset. The rows correspond to the classes (from the top) *airplane*, *automobile*, *bird*, and *cat*. Each image is 32×32 in size. 96
Figure 6.7 Some exemplar receptive fields of units in the first (a) and second layer (b) of a LCDBN trained on the CIFAR-10 dataset with 11x11 Gaussian receptive fields. 97
Figure 6.8 Orientation maps computed for the first (a) and second (b) layers of a 11x11 sized receptive field LCDBN. Notice that the second layer is more selective for orientation as suggested by the receptive fields. This image is coloured according to angle selectivity. . . 98

List of Tables

Table 3.1 VisNet dimensions 22
Table 3.2 Sigmoid parameters 23
Table 3.3 Lateral inhibition parameters 24
Table 3.4 Layer 1 connectivity 26

Table 6.1 Table showing the percentage errors when a LCDBN is trained for a discriminative task on MNIST. Note that the 28 receptive field size is equivalent to the all-all connectivity 93

Acknowledgements

I would first like to thank the most important person in the world: my wife and best friend, Kelli. Her support over the course of this PhD has been nothing short of fantastic and every day that passes I am reminded how lucky I am to be going through life with her by my side. It absolutely goes without saying that I would not be the person I am today without her help and encouragement. Thanks, toots!

I would of course also like to thank my supervisors, Prof. Edmund Rolls and Dr. Ben Graham. Edmund, has consistently gone above and beyond in his duties as a supervisor and his exceptional work ethic will be an enduring inspiration. Ben was likewise always willing to entertain discussions that would run on for hours longer than scheduled, frequently leaving me reinvigorated for the project. So thank you Edmund. Thank you Ben. It was very much appreciated.

It would be remiss of me to not thank all the people within the Complexity Department who have supported my studies and the EPSRC for a generous stipend that made this journey financially viable.

Lastly, I would like to thank my family and friends who have tirelessly feigned interest in my work over these years! I promise I will talk (a bit) less about vision and machine learning in the future.

Publications

1. Robinson, L. & Rolls, E. T., 2015. *Invariant Visual Object Recognition: Biologically Plausible Approaches.* **Biological Cybernetics, 2015, 109(4-5):505-35**.

2. Robinson, L. & Graham, B., 2015. *Confusing Deep Convolutional Networks by Relabelling.* **arXiv Preprint, arXiv:1510.06925**.

3. Graham, B. Reizenstein, J & Robinson, L., 2015. *Efficient Batchwise Dropout Training Using Submatrices.* **arXiv Preprint, arXiv:1502.02478**.

Abstract

Understanding the processes that facilitate object recognition is a task that draws on a wide range of fields, integrating knowledge from neuroscience, psychology, computer science and mathematics. The substantial work done in these fields has lead to two major outcomes: Firstly, a rich interplay between computational models and biological experiments that seek to explain the biological processes that underpin object recognition. Secondly, engineered vision systems that on many tasks are approaching the performance of humans.

This work first highlights the importance of ensuring models which are aiming for biological relevance actually produce biologically plausible representations that are consistent with what has been measured within the primate visual cortex. To accomplish this two leading biologically plausible models, HMAX and VisNet are compared on a set of visual processing tasks.

The work then changes approach, focusing on models that do not explicitly seek to model any biological process, but rather solve a particular vision task with the goal being increased performance. This section explores the recently discovered problem convolution networks being susceptible to adversarial exemplars. An extension of previous work is shown that allows state-of-the-art networks to be fooled to classify any image as any label while leaving that original image visually unchanged. Secondly an efficient implementation of applying dropout in a batchwise fashion is introduced that approximately halves the computational cost, allowing models twice as large to be trained. Finally an extension to Deep Belief Networks is proposed that constrains the connectivity of the a given layer to that of a topologically local region of the previous one.

Introduction

Introduction

The research described in this thesis aims to investigate both biologically plausible and machine learning approaches to invariant visual object recognition.

This thesis first summarises the substantial work that has been done trying to understand the biological mechanisms behind invariant object recognition and the attempts to build both explanatory models and specific engineering solutions to solve vision tasks. Specific emphasis is placed upon the interdisciplinary nature of this work, with successful models integrating knowledge across neuroscience, psychology, computer science and mathematics.

The thesis is then split into two parts; the first highlighting the importance of biologically plausible models producing representations that are consistent with that measured from within the primate visual cortex. The aim of these models isn't to engineer high performance on a particular vision task, but rather investigate how the brain accomplishes the task. This interplay between the experimental evidence and computational modelling allows for the refinement of quantitative theories of vision that seek to explain how the cortical structures might be computing useful representations for vision. Specifically, the two leading biologically plausible models of invariant object recognition, VisNet (Rolls 2008b, 2012b) and HMAX (Riesenhuber and Poggio 1999b, 2000b; Serre et al. 2005). The aim of this comparison is to investigate which of these two approaches better account for what is found neurophysiologically in the primate brain areas involved in invariant visual object recognition.

The second part focuses on extending models that do not explicitly seek to model any of the biological processes, but rather solve a particular vision task with the guiding principle being that of increased performance rather than biological plausibility. This second half focuses on extending the Deep Belief Network model (Hinton 2009; Hinton et al. 2006) with a local connectivity constraint, exploring a new method for creating adversarial exemplars (Goodfellow et al. 2015; Szegedy et al. 2014) and an efficient way to apply dropout (Hinton et al. 2012) without losing model accuracy.

The contrast of these approaches at even a high level is both interesting and revealing. One fundamental difference is that the majority of biologically plausible approaches have strived to account for how an individual member of a class, for example a particular person, can be recognised invariantly with respect to various transforms such as view, pose, etc. In contrast machine learning approaches to vision have tended to attempt to solve a slightly different problem, that of the categorisation of individual exemplars into broad classes such as cars, dogs, cats, etc. The distinction between the identification and classification problem might on the surface seem trivial and perhaps unnecessary, but it is important to point out that the neurophysiology has consistently shown the representations built at the highest levels of the visual cortex are that of identities and not classes (Rolls 2008b).

Another difference is that the biologically plausible approaches have sometimes presented systematic training sets in which for example, a series of different views of each object or class to be

learned is provided VisNet (Rolls 2008b, 2012b). In contrast many machine learning approaches tend to use less well posed training sets in that they typically comprise very large numbers of exemplars of many categories of object, with no attempt to provide a systematic set of different views of each object to provide a basis for view-invariant object recognition.

Some of the biologically plausible approaches have used semi-supervised learning, which has been provided a mechanism to attempt to learn from the temporal continuity that reflects natural image statistics (Rolls 2008b, 2012b). This assumes that typically humans and other primates inspect an object for a short period while it undergoes transforms, for example rotation into different views, scale change, etc. The eyes then move to a different object in which transforms of the new object may then also be seen for a short time period. Some biologically plausible approaches such as VisNet explicitly take advantage of these statistics to help the system learn which images correspond to the transforms of an individual object (Rolls 2008b, 2012b). In contrast most machine learning approaches that are unsupervised, only identify categories based on similarities of image statistics of large numbers of exemplars of many categories of object.

Partly for these reasons comparing these two types of approach may enable strengths of each type of approach to be combined to enable new progress to be achieved.

Modelling Vision: Object Recognition

Introduction

In this chapter the basic results from neurophysiology that are thought to underpin the biological basis of vision, and specifically object recognition are reviewed. It shall be seen that this body of work strongly suggests that at least for object recognition, the visual cortex acts as a hierarchical series of feature extraction and combination stages. Before a detailed discussion of how these results are integrated into biologically inspired computational models of object recognition a brief discussion of other potential psychophysical and computational approaches will be undertaken. A comprehensive review of the neurophysiology and computational approaches can be found in (Rolls and Deco 2002; Ullman 1996).

Neurophysiology of Vision

The human visual system is a remarkably powerful system that is perhaps capable of discriminating between tens of thousands of distinct objects (Biederman 1987). Detailed and accurate knowledge of the architecture and circuitry involved in the visual cortex is required to inform biologically sound computational models. The invasive nature of traditional measurement techniques inevitably mean that the bulk of our knowledge has been gathered from non-human primates. The remainder of this section will describe some of the fundamental neurophysiology that has been uncovered about the visual cortex with particular attention to mechanisms that are implicated in object recognition.

Dorsal and ventral streams

The first place to begin when explaining the known neurophysiology, is with that of the high level cortical organisation involved with the processing of visual stimuli. The visual pathway has been described as being made up of two primary pathways, named the dorsal and ventral stream (see Fig. 2.1).

The dorsal stream starts in the primary visual cortex (V1), from where it proceeds to areas V2 and the middle temporal area (MT) before ending up in the posterior parietal cortex. The dorsal stream is often characterised as the 'where' or 'how' pathway and is strongly associated with the perception of motion, the positions of objects in the visual field and feedback control of the eyes

The ventral stream likewise starts in the primary visual cortex (V1), where is proceeds to areas V2 and V4 before ending up in the inferior temporal cortex (IT). The ventral stream is often characterised as the 'what' pathway and is strongly associated with object recognition, categorisation and representation.

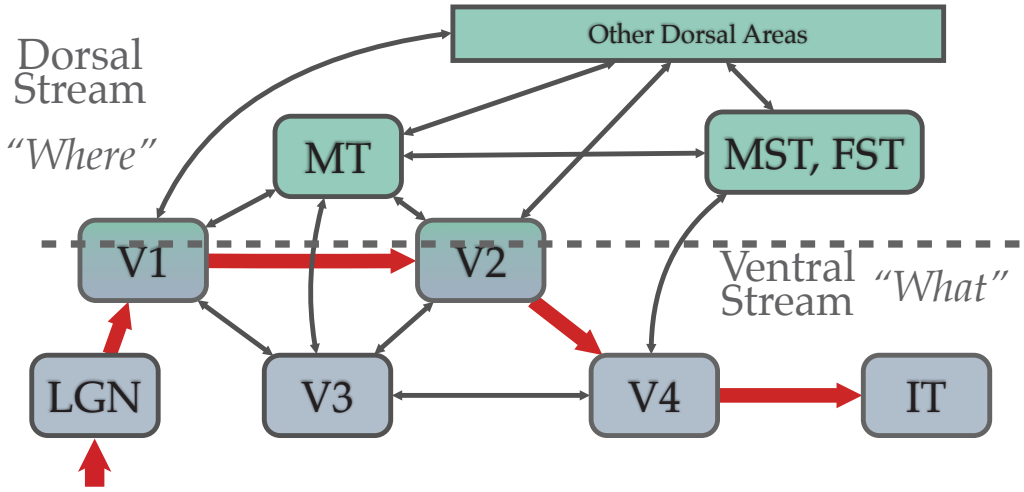


Figure 2.1: Caricature of the regions of the visual cortex and their major interconnections. The dorsal areas, MT (medial temporal), MST (medial superior temporal) and FST (fundus of superior temporal suculus) seem to be primarily concerned with motion and the location of objects. The red (bold) feedforward ventral pathway is where the majority of object recognition processes are thought to take place. Adapted from Gross et al. (1993)

The split of the visual pathway into this dichotomy is known as the 'two-streams hypothesis' and was first proposed by Ungerleider and Mishkin (1982). Significant research supports the idea of two functionally distinct processing streams within the primate visual system (Baizer et al. 1991; Felleman and Van Essen 1991; Livingstone and Hubel 1988, 1987; Maunsell and Newsome 1987; Van Essen et al. 1992a). While this split hierarchy is well established the two pathways are not completely independent as can be seen on Fig. 2.1. There are significant cross pathway connections (Ungerleider and Haxby 1994; Van Essen et al. 1992a). The remainder of this section will limit itself to the cortical structures that dominate the ventral stream since the focus of this work is object recognition processes.

When light enters the back of the eye, it stimulates light sensitive cells on the retina. These cells effectively transduce the incoming light into an electric form, that is carried by the nervous system. The retina itself comprises of quite complex neural circuitry, the retinal ganglion cells already do some processing of the visual information before relaying it via the optic nerve to the lateral geniculate nucleus (LGN) in the thalamus before again relaying it to the occipital lobe - the first area of the visual cortex, V1 (Callaway 2004).

Primary visual cortex - V1

The V1 area was first systematically studied with the ground breaking work of Hubel and Wiesel (1962, 1968a). Their work cemented the ideas that individual neurons in V1 have a region, or *receptive field* in the visual field that generates a maximum response and that these receptive fields vary somewhat continuously over space forming a retinotopic map (Talbot and Marshall 1941). This map is not an isometric mapping, there is significantly more cortical area devoted to that of the central portions of the visual field. Neurons within V1 can be classified into three groups: simple

cells, complex cells and hypercomplex cells. The simple cells have a centre surround receptive field with elongated excitatory and inhibitory areas at a specific angle - which makes them sensitive to oriented luminous edges (i.e are edge/bar detectors) in the visual field (Hubel and Wiesel 1962). The complex cells, much like the simple cells have receptive fields that act as edge detectors but they respond over a larger region of the visual field - they have a degree of invariance to position (but usually not orientation) (Hubel and Wiesel 1962). The hypercomplex cells exhibit even more complex receptive fields, seemingly only strongly responding to lines of a specific orientation moving in a specific direction and only if the stimulus is under a certain length (Wiesel and Hubel 1965). These properties have led to the idea that (at least in regard to static images) V1 functions in some ways analogous to a filter bank of oriented edge/bar detectors - the operation of which is commonly modelled by Gabor functions (Carandini et al. 2005; Daugman 1988b; Teich and Qian 2006). Other distinguishing properties of V1 are known, such as ocular dominance regions (LeVay et al. 1980) and color specific processing regions (blobs) (Livingstone and Hubel 1988).

Visual areas V2 & V4

V2, or the 'prestriate cortex' has been shown to build upon the responses of neurons in V1, with evidence of specific selectivity to combinations of orientations from V1 (Anzai et al. 2007). V2 physiology is characterised by dark bands of thick and thin stripes, with lighter regions between them called 'interstripes' (Livingstone and Hubel 1982). Neurons within the thick bands have been named 'form' cells and are strongly orientation and direction sensitive, much like complex cells of V1. Neurons within the thin bands show no orientation sensitivity, but are sensitive to particular colours and so are thought to be principally connected to 'blob' cells from V1 (Livingstone and Hubel 1988). V2 has been implicated in the perceptual phenomena of illusory contours, with Peterhans and von der Heydt (1989) reporting significant numbers of neurons in this region responding to contours extending across gaps.

V4 region is thought to be made up of at least receives input primarily from the thin and interstripe cells of V2 and as such are heavily implicated in colour (Dubner and Zeki 1971) and orientation (Desimone and Schein 1987). V4 is the first area of the ventral stream where strong attentional (top down) mechanisms can be measured - the receptive field sizes contract at the position being attended to (Moran and Desimone 1985; Reynolds et al. 1999; Schiller 1994). The receptive fields of neurons in V4 are significantly larger than that of V1 and V2, but smaller than those in the nex area, the inferior temporal cortex (IT) (Kobatake and Tanaka 1994).

Inferior temporal cortex - IT

The inferior temporal cortex (IT) is thought to be the final area of the brain that is dedicated to exclusively processing visual information and completes the hierarchy through the ventral stream - taking the majority of its input from V4 (Ungerleider and Haxby 1994; Ungerleider and Mishkin 1982).

Cells within IT tend to preferentially respond to complex visual stimuli (Gross et al. 1972, 1993; Tanaka 1996). Indeed many neurons show no response to simple oriented edges (Kobatake and Tanaka 1994). Furthermore the responses to complex stimuli are robust to transformations (position, pose, etc) across a wide region of the visual field (Aggelopoulos and Rolls 2005; Gross et al.

1993; Kobatake and Tanaka 1994; Perrett et al. 1982; Rolls 1991; Rolls et al. 1994). The building of these invariant responses (neurons that fire strongly for a given stimuli regardless of the transformation) is fundamental to the task of object recognition (Logothetis et al. 1995).

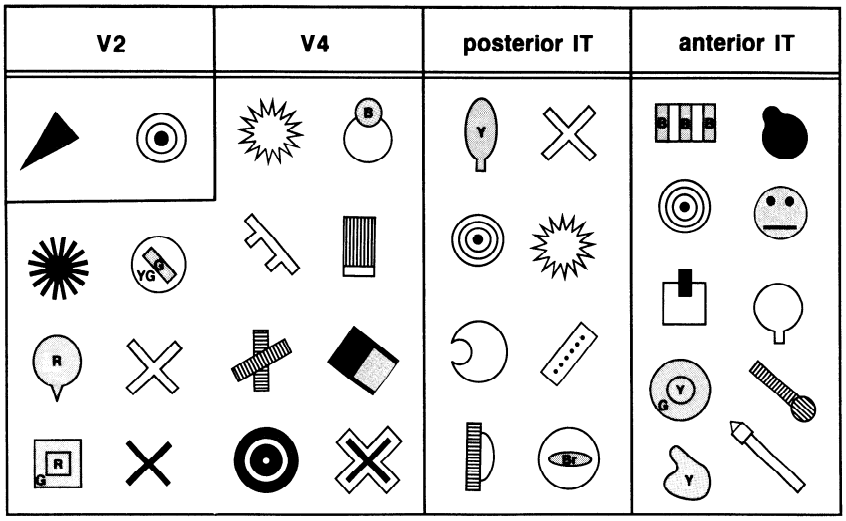


Figure 2.2: Examples of features identified by Kobatake and Tanaka (1994) grouped by the region of the visual hierarchy. Note that the stimuli become more visually complex as you traverse the ventral stream. (After Kobatake and Tanaka (1994)).

The neurophysiological evidence of the preceding section strongly suggests that the ventral processing stream tends to be organised to integrate information in a hierarchical manner. The preferred stimuli of neurons becomes more complex as you move from area to area (see Fig. 2.2) (Kobatake and Tanaka 1994), the receptive fields increase significantly until they encompass upwards of 50 deg or more (Boussaoud et al. 1991; Perrett and Oram 1993; Rolls 1992) and typically encompass the foveal region (Gross et al. 1972). Finally the degree of invariance to transformations that the preferred stimuli can undergo increases. These architectural features will play a significant part in the development of computational models of object recognition.

Computational Models of Vision

The approaches to constructing computational models of object recognition can be broadly classified into two groups, each with different motivations and goals. The first adopts the perspective of computational neuroscience and utilise computational modelling as a tool to refine the understanding of complex experimental neurological evidence towards that of a cohesive theory of vision. In keeping with this aim these types of models are thus heavily constrained to operate in a manner which is biologically plausible. In contrast the second group follow a more machine learning, engineering led philosophy that eschews the strict restriction of biology and instead embarks on an undertaking to simply engineer a vision system that for a given specific task can by various metrics perform well - i.e mimic the human visual system, rather than explain.

The variety of computational models that have arisen can be similarly grouped into two loose categories: structural models and view models.

Structural Models

One approach to modelling object recognition might be to assume that objects can be represented by a decomposition of the object into simple component parts and within such an object referenced representation a hierarchical spatial relationship between components can be established. Several models exhibiting such a decomposition can be thought to stem from the influential work of David Marr (Marr 1982; Marr and Nishihara 1978). The chief motivation of these models seems to be that invariant object recognition is an easier task if the object has first been reduced to a structural description. With Marr suggesting that this decomposition of objects not only results in an explicit 3D representation, but that recognition occurs in a fundamentally bottom-up hierarchical fashion (Marr 1982). Subsequent extensions of this idea were made by Binford (1981); Brady et al. (1985); Dane and Bajcsy (1982); Pentland (1986), with perhaps the most notable being Biederman and his 'Recognition by Components' (RBC) theory (Biederman 1985, 1987). Biederman sought to show that objects can be naturally decomposed into a finite set of abstract primitive shapes, or 'geons' - perhaps as few as 36 3D shapes comprised of transformed boxes, cylinders, spheres, etc. A computational implementation of the RBC theory was attempted by Hummel and Biederman (1992) by explicitly using a neural network model - though other such computational efforts at instantiating many of these ideas are notably lacking.

The precise details of how these family of models arrive at their ultimate representation differs but the common theme is that shape information in an object-centric form is accessible to the visual cortex. Structural models give a good account of how adapt humans seem to be at generalising across object categories - for instance consider the structural relationship between the constituent parts of a chair and then compare that to all possible instantiations of what we consider a chair. Though such ideas struggle to explain how specific object identification between similar objects of a given class can be achieved without increasingly fine grained structural descriptions - consider how structurally similar all dogs are to one another.

View Models

A different approach suggests that what constitutes an object can be thought of as a collection of views from which view-dependent features can be extracted. This essentially relegates the task of object recognition to that of matching the current stimuli to that of previously seen images. This idea is fundamentally different from that of the structural approach as there are no explicit requirements to represent objects as decomposed components, instead the view-dependent features uniquely define the objects spatial configuration.

Significant psychological evidence suggest that not all views of a known object are as easy to recognise (Edelman and Bülthoff 1992; Logothetis et al. 1994; Rock and Divita 1987; Rock et al. 1981; Tarr 1995; Tarr et al. 1998). Inverting the viewed object (and especially faces) negatively affects the ability of humans to recognise the object in question (Valentine 1988). Rotating objects both within and out (i.e in depth) of the viewing plane demonstrates that object recognition shows a strong orientation dependence (Edelman and Bülthoff 1992; Logothetis and Pauls 1995; Tarr et al. 1998). A review of the results specifically relating to the recognition of objects undergoing rotations can be found in the work of Biederman (2000). Furthermore neurophysiological recordings have also demonstrated view dependent firing of inferotemporal (IT) neurons (Desimone 1991; Logothetis et al. 1995).

With strong evidence to suggest that object recognition does not depend on an object centred representation, the issue becomes how can independent views of objects be tied together such that invariant object recognition can take place? Perhaps the most straightforward way to implement a view based approach to object recognition would be template matching, a process where by the image on the retina is directly compared to that of a stored picture of the object. Unfortunately just directly comparing the image formed on the retina to that of a stored image is very sensitive to transformations of the object that create a view of the object which is different to that of the stored image. In principle a solution to this problem may be to first transform the incoming image into a canonical form and so compensate for the difference between the stored image and the viewed object (Ullman 1996). The type of transformations required to perform this task are in general complex, especially in the case of non-affine deformations.

An alternative method is to consider each object to be made up of specific features and so each object resides in an N dimensional feature space (Tou and Gonzalez 1974). The set of features are chosen to be "large", such that even though there are likely to be significant overlap (the features can range from simple line segments, up to complex textured patches) there will exist unique subsets of features for each distinct object.

Simple feature space models, sometimes termed feature extraction pipelines, can be thought of as being based around the following sequence of processes:

1. Feature extraction - the input images are processed to calculate the set of features. In practice this usually involves convolving the image with a set of filters.

2. Invariance building - the set of initial features are further processed to increase the tolerance of the final set of features to small transformations. This is usually accomplished by a pooling function that combines the feature filter responses over a small spatial extent within the image.

3. Classification - the resulting collections of feature activations for each image are used to train a classifier for the specific task.

The features exhibited by simple and complex cells (Hubel and Wiesel 1968b) have been the inspiration for many feature extraction processing stages. The filters are chosen to mimic the receptive fields of simple cells in that they have compact support, and are non-isotropic - i.e edge detectors. They are commonly chosen to be a family of Gabor filters at various scales, orientations and symmetries (Daugman 1988b). A selection of such filters can be seen in Fig. 2.3 along with the output of such a set of filters in Fig. 2.4. In practice these filters compute a decomposition of the input image into a set of edge responses at different scales and orientations.

Coupling a very simple feature extraction method using a set of Gabor filters at various scales and orientations with a classifier such as a simple linear SVM it is possible to do significantly better than chance (and much better than using template matching, or a more complex classifier on the raw pixel information) when trying to identify the object class of a given image when using databases that contain natural images with hundreds of object classes (Pinto et al. 2008).

The basic premise outlined above is at the core of many computer vision systems (Dalal and Triggs 2005; Daugman 1988b; Lowe 2004; Reid et al. 1989; Viola and Jones 2002).

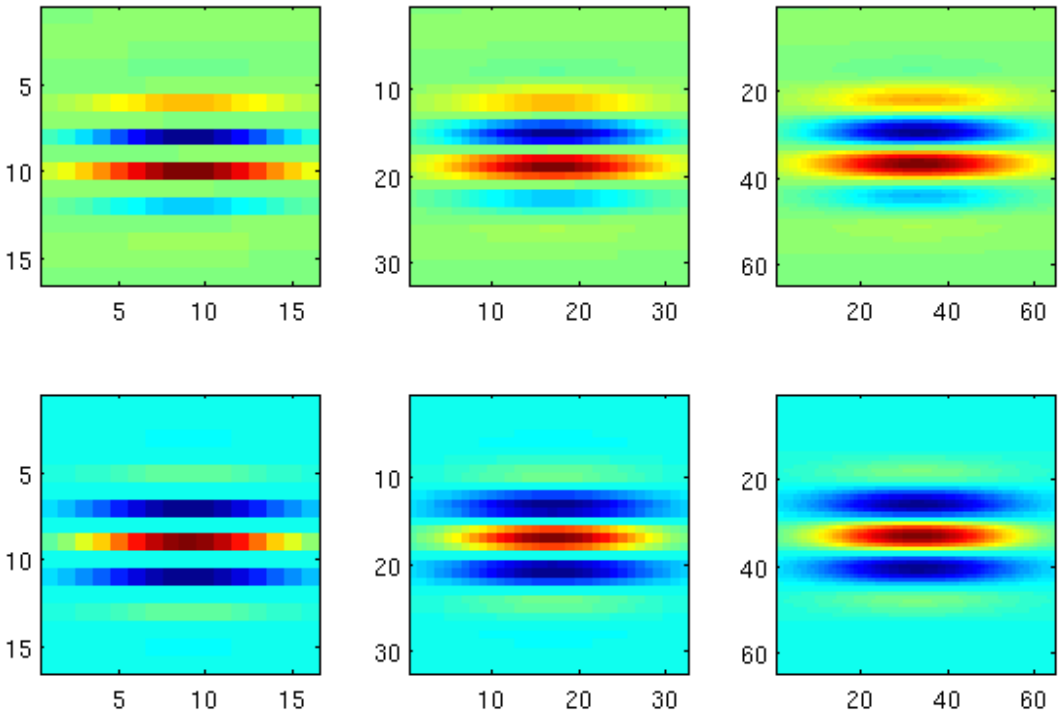
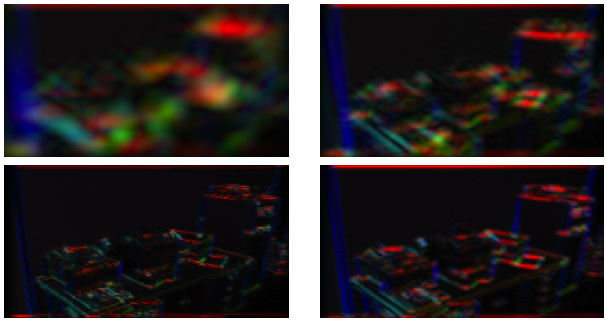


Figure 2.3: Examples of various symmetric (bottom) and anti-symmetric (top) Gabor filters at various scales (horizontal progression).

(a)



(b)

Figure 2.4: Example image (a), and its Gabor filtered output at four scales (scale decreases clockwise from top-left) and four orientations (b). Orientation of the filter is encoded in the colour.

Feature Hierarchies and Invariance

The natural extension of the feature extraction pipeline described above is that not only are the low level features extracted from the stimuli, but in the subsequent levels of processing these features are *combined* to produce ever more complex features. For example, a corner feature is a more complex feature than that of a simple oriented straight edge and furthermore it can be thought of as a composite feature that is built from the composition of multiple specifically oriented straight edge features from the preceding layer. In this way as you traverse a hierarchy of feature extraction and combination layers ever more complex features are constructed. These features can encode complex spatial relationships as the receptive field sizes of each unit begin to encompass significant

portions of the original stimuli via the composition. The end result of this process is a transition from very simple, spatially local feature detectors in the initial stages (blobs and lines) towards that of features that respond to different specific objects over a wide range of input transforms at the output layer.

One of the earliest computational models to implement this idea was the Neocognitron (Fukushima 1975, 1980). The Neocognitron is a hierarchical neural network, with an architecture of alternating layers of *simple* (S) and *complex* (C) cells (after Hubel and Wiesel (1968a)). The S cells can be thought of being a template defined at a particular location and orientation, the C cells then pool the responses of the S cells of the preceding layer and thus build feature combinations and increase the invariance. The Neocognitron also takes advantage of a convolution structure - the S cells of a specific type are tiled across the entire visual field of the model via explicit sharing of the synaptic weights between the preceding layer. The synapses of the S layers can be modified by an unsupervised learning process. The learning algorithm is essentially a winner-take-all process that directly modifies the synaptic weights of the most strongly responding S layer neuron, making their selectivity better match the incoming input. The Neocognitron successfully demonstrates that both the invariance and selectivity of features can be increased as you traverse up through a hierarchy, and moreover this increase can result as a consequence of the learning mechanisms of the model.

VisNet, initially developed by Rolls (2008b); Rolls and Milward (2000); Wallis and Rolls (1997a) attempts to model the entire visual cortex thought to be involved in object recognition, at the neuron level. The model consists of a series of competitive rate neurons organized in hierarchical layers encompassing short-ranged mutual inhibition within each layer. The connectivity between layers (and the input) is convergent, topologically consistent (spatially local receptive fields), feed-forward and probabilistic in nature with a distribution in accordance with the known receptive field size of neurons at each layer - see Wallis and Rolls (1997a) for further details. This structure explicitly allows neurons in the top layer (via the intermediary layers) to integrate information across the whole of the input visual field. The VisNet model is trained in an unsupervised way via a modified associative (Hebb-like) learning rule which incorporates a temporal trace of the neuron's previous activity - which has been shown key to enabling neurons learn transform invariances that are thought to be central to the problem of object recognition (Földiák 1991; Rolls 1992, 2012b; Wallis and Rolls 1997a; Wallis et al. 1993).

The trace rule of VisNet can be thought to be complimentary to that of another general approach called slow feature analysis (SFA). Slow feature analysis attempts to extract the slowly varying features from that of an base data stream which is in contrast varying quickly (Wiskott 2003; Wiskott and Sejnowski 2002). In the context of vision the slowly varying features will be the higher level representations of the viewed objects - the object class, identity, etc. while the quickly varying signal is the actual stimuli. The application of SFA to images by moving a small receptive field across natural images while undergoing translations, rotations and scaling results in recovering feature extractors that are quantitatively similar to that of Complex cells found within the early visual cortex (V1) (Berkes and Wiskott 2005; Wiskott 2006). Extending SFA to deal with large input dimensions results in a hierarchical formulation that has shown good abilities to generate invariant features useful for the classification of objects within complex images (Franzius et al. 2008).

HMAX was originally proposed by Riesenhuber and Poggio (1999a), though has seen numerous additions and extensions (Mutch and Lowe 2008; Serre et al. 2007b,c). HMAX is a hierarchical feature extraction pipeline, with parameters that are constrained to be biologically relevant from experimental data. The overall structure is much like the previously described Neocognitron, with HMAX being composed of alternating heterogeneous layers of S and C cells. Again, the C cells pool across previous S cell output though this time information is integrated via the standard maximum operator. Neurons in the S layers act as template matching, with the stimuli for which they maximally respond to artificially set to be random patches of the preceding C_{n-1} layer. This random sampling process is used as a crude unsupervised learning process, since features that are most often represented in the training images (or indeed unrelated natural images) have an increased probability to be sampled and thus have an S unit at a given scale selective to that feature.

An interesting approach taken by Pinto et al. (2009) is to appeal to optimisation and turn the problem of defining a particular model architecture into one of efficiently searching a parametrised model space for models that perform well on a given visual task. The underlying assumption being that many of the types of models described so far have similar properties, but the observed performance of any one of them is strongly dependent on the parameters that instantiate that particular model. The family of models explored are strictly feed-forward and have three layers that consist of numerous filtering, pooling and normalisation operations depending on the particular parameters. The search of model space is carried out by random sampling, with each model being subject to an unsupervised developmental phase of learning (that itself is parametrised and subject to selection), before the model is tested on a separate two-class object discrimination problem. Thousands of randomly instantiated models are tested in this fashion to find architectures that exhibit high performance. Extensions of this screening approach have led to models that build representations in the top most layers that are predictive of measured neural responses from non-human primates (Yamins et al. 2014).

Hinton, Osindero, and Teh (2006) showed that a hierarchical set of Restricted Boltzmann Machines (RMBs), each of which when trained independently in a greedy, unsupervised fashion with contrastive divergence (Hinton 2002) produce units that are sensitive to simple features in the lower layers, while more abstract and complex feature appear in the deeper layers (Hinton and Salakhutdinov 2006). Extensions of this model to the convolution setting by Lee et al. (2009a) has further demonstrated the ability of stacked RBMs to extract complex hierarchical compositions of natural images with the extra advantage of increased translation invariance due to the convolutional structure (Lee et al. 2011). Work by Norouzi et al. (2009) attempts to further increase the invariance exhibited by stacked RBMs by introducing explicit sets of linear transformations that approximate local rotations, translations and scaling. These transforms are interleaved between the layers, resulting in the output of each receptive field in the layer above pooling not only spatially over the input but also over the set of responses resulting from the transforms. Similar explicit transforms in the context of stacked RBMs have been explored by Kivinen and Williams (2011) and Sohn and Lee (2012).

Convolution neural networks (CNNs) (LeCun et al. 1989, 2010, 1990) are an extension of the traditional multiple layer perceptron neural network model, specifically designed to exploit the two dimensional structure of images. CNNs have provided state-of-the-art results when it comes

to object recognition on many image benchmark datasets (Graham 2014; He et al. 2015; Krizhevsky et al. 2012; Szegedy et al. 2014; Taigman et al. 2014).

The typical CNN architecture consists of many convolution and pooling layers interleaved which are then followed by a series of fully connected layers much like the traditional multiple layer perceptron. The convolution layers can be thought of as k neurons that are only connected to a local region of the input layer. The weights of a particular neuron then defines a filter, the output of which is naturally defined as a convolution operation. Applying this convolution to the whole input layer (with a specific stride) has the effect of tiling the particular neuron across the input layer. Doing this for all k neurons results in a $k \times n \times m$ output structure, where n and m are defined by the input layer size and the stride of the convolution (i.e the specific connectivity). Equivalently, this operation can be thought of as producing k sets of neurons that have local connectivity which are laterally propagated over the input layer and thus share the same weights. The value k is typically increased as you traverse the hierarchy from the input. The pooling layers are sub-sampling operations (typically the maximum operator, but could be an average or the L_2 norm) over small $p \times p$ regions that only operate over the spatial dimensions and not the over the k distinct output maps. Concretely, the result of a typical pooling layer (with $p = 2$) is to produce an output structure that has dimensions $k \times n/2 \times m/2$. The pooling layers are helpful in introducing a small amount of invariance to transformations, and reduce the computational load involved with computing the required convolutions as the number of features increase. Training CNNs is typically accomplished in a supervised manner by stochastic gradient descent with the required gradients computed end-to-end from the cost function via backpropagation of error (LeCun et al. 1989; Rumelhart et al. 1986; Werbos 1974).

While demonstrating excellent results on large scale datasets of natural images CNNs only have limited mechanisms for learning invariances. The convolution operation adds a degree of equivariance and the pooling layers add some local invariance but these mechanisms are not enough to provide features that are invariant to large transforms of the input image (Cohen and Welling 2015; Lenc and Vedaldi 2015). There have been numerous attempts to build explicit architectural or learning mechanisms that increase the ability of the models to produce invariant representations. One approach is to relax the strict weight sharing across the spatial dimension of the k neurons in any given convolution layer. Instead of the usual arrangement, only neurons that are spatially d steps away from each other are tied together with weight sharing (so $d = 1$ is a normal CNN). This has the effect of increasing the number of potential features that are then pooled over and has been shown to increase the rotational invariance (Le et al. 2010).

Another is to transform the input by a set of explicit rotation transformations, with the separate sub networks re-integrating this information at higher stages. This can be accomplished by learning sets of filters of transformed features (Dieleman et al. 2015; Kanazawa et al. 2014; Sohn and Lee 2012; Xu et al. 2014) or by creating ensembles of models with different initial transforms (Alvarez et al. 2012). In a similar vein to VisNet and SFA some approaches with convolutional networks have attempted to use the concept of slowness to leverage the extra information present in correlated input images to help build complex invariances (Goroshin et al. 2015; Mobahi et al. 2009; Zou et al. 2012).

Discussion

Hopefully this review has conveyed just how much has been uncovered about the biological mechanisms underpinning vision and the efforts to create integrative theories of object recognition via the interplay between computational modelling and biological experimentation. None of the models mentioned in this section completely explain object recognition, with each model having its own defined set of goals, advantages and limitations which underlines the fact that there is much work to be done in fully understanding even this narrowed aspect of visual processing.

Biologically Plausible Approaches

Biological plausibility of VisNet and HMAX

Introduction

The aim of this chapter is to assess the biological plausibility of two models that purport to be biologically plausible or at the very least biologically inspired. The work will consist of investigations probing just how biologically plausible they are, by comparing them to the expected responses of inferior temporal cortex neurons. Four key experiments are performed to measure the firing rate representations provided by neurons in the models; whether the neuronal representations are of individual objects or faces as well as classes; whether the neuronal representations are transform invariant; whether whole objects with the parts in the correct spatial configuration are represented; and whether the systems can correctly represent individual objects that undergo catastrophic view transforms. In all these cases, the performance of the models is compared to that of neurons in the inferior temporal visual cortex. The overall aim is to provide insight into what must be accounted for more generally by biologically plausible models of object recognition by the brain, and in this sense the research described here goes beyond these two models. Non-biologically plausible models are not considered here as the main aim is neuroscience, how the brain works, but we do consider in the Discussion some of the factors that make some other models not biologically plausible, in the context of guiding future investigations. We note that these biologically inspired models are intended to provide elucidation of some of the key properties of the cortical implementation of invariant visual object recognition, and of course as models the aim is to include some modelling simplifications, which are referred to below, in order to provide a useful and tractable model.

One of the major problems that is solved by the visual system in the primate including human cerebral cortex is the building of a representation of visual information that allows object and face recognition to occur relatively independently of size, contrast, spatial frequency, position on the retina, angle of view, lighting, etc. These invariant representations of objects, provided by the inferior temporal visual cortex (Rolls 2008a, 2012b), are extremely important for the operation of many other systems in the brain, for if there is an invariant representation, it is possible to learn on a single trial about reward/punishment associations of the object, the place where that object is located, and whether the object has been seen recently, and then to correctly generalize to other views etc. of the same object (Rolls 2008a, 2014). In order to understand how the invariant representations are built, computational models provide a fundamental approach, for they allow hypotheses to be developed, explored and tested, and are essential for understanding how the cerebral cortex solves this major computation.

The chapter is organised as follows: first a summary account is given of some of the fundamental properties of the responses of primate inferior temporal cortex (IT) neurons (Rolls 2008a, 2012b; Rolls and Treves 2011) that need to be addressed by any biologically plausible models of invariant

visual object recognition. Then a discussion of how models of invariant visual object recognition can be tested to reveal whether they account for these properties is undertaken. The two leading approaches to visual object recognition by the cerebral cortex, that are used to highlight whether these generic biological issues are addressed, are VisNet (Rolls 2008a, 2012b; Rolls and Webb 2014; Wallis and Rolls 1997a; Webb and Rolls 2014) and HMAX (Mutch and Lowe 2008; Serre, Kreiman, Kouh, Cadieu, Knoblich, and Poggio 2007a; Serre, Oliva, and Poggio 2007b; Serre, Wolf, Bileschi, Riesenhuber, and Poggio 2007c). In comparing these models, and how they perform on invariant visual object recognition, the aim is to make advances in the understanding of the cortical mechanisms underlying this key problem in the neuroscience of vision. The architecture and operation of these two classes of network are described below.

Some of the key properties of IT neurons that need to be addressed, and that are tested in this paper, include:

1. Inferior temporal visual cortex neurons show responses to objects that are typically translation, size, contrast, rotation, and in many cases view invariant, that is, they show transform invariance (Aggelopoulos and Rolls 2005; Booth and Rolls 1998; Hasselmo et al. 1989; Logothetis et al. 1995; Rolls 2012b; Rolls and Baylis 1986; Rolls et al. 1985, 1987, 2003; Tovee et al. 1994; Trappenberg et al. 2002).

2. Inferior temporal cortex neurons show sparse distributed representations, in which individual neurons have high firing rates to a few stimuli and lower firing rates to more stimuli, in which much information can be read from the responses of a single neuron from its firing rates (because they are high to relatively few stimuli), and in which neurons encode independent information about a set of stimuli, as least up to tens of neurons (Abbott, Rolls, and Tovee 1996; Baddeley, Abbott, Booth, Sengpiel, Freeman, Wakeman, and Rolls 1997; Rolls 2008a, 2012b; Rolls and Tovee 1995; Rolls and Treves 2011; Rolls, Treves, Tovee, and Panzeri 1997a; Rolls, Treves, and Tovee 1997b; Tovee, Rolls, Treves, and Bellis 1993).

3. Inferior temporal cortex neurons often respond to objects and not to low-level features, in that many respond to whole objects, but not to the parts presented individually nor to the parts presented with a scrambled configuration (Perrett et al. 1982; Rolls et al. 1994).

4. Inferior temporal cortex neurons convey information about the individual object or face, not just about a class such as face vs non-face, or animal vs non-animal (Abbott et al. 1996; Baddeley et al. 1997; Rolls 2008a, 2012b; Rolls and Tovee 1995; Rolls and Treves 2011; Rolls et al. 1997a,b). This key property is essential for recognising a particular person or object, and is frequently not addressed in models of invariant object recognition, which still focus on classification into e.g. animal vs non-animal, hats vs bears vs beer mugs etc (Mutch and Lowe 2008; Serre et al. 2007a,b,c; Yamins et al. 2014).

5. The learning mechanism needs to be physiologically plausible, and that is likely to include a local synaptic learning rule (Rolls 2008a). We note that lateral propagation of weights, as used in the neocognitron (Fukushima 1980), HMAX (Mutch and Lowe 2008; Riesenhuber

and Poggio 1999a; Serre et al. 2007b), and more generally in convolution nets LeCun et al. (2010, 1990), is not a biologically plausible mechanism.

Methods

Overview of the architecture of the ventral visual stream model, VisNet

In this section, the architecture of VisNet (Rolls 2008a, 2012b) is summarized briefly, with a full description provided afterwards.

Fundamental elements of Rolls' 1992 theory for how cortical networks might implement invariant object recognition are described in detail elsewhere (Rolls 2008a, 2012b). They provide the basis for the design of VisNet, which can be summarized as:

- A series of competitive networks, organized in hierarchical layers, exhibiting mutual inhibition over a short range within each layer. These networks allow combinations of features or inputs occurring in a given spatial arrangement to be learned by neurons using competitive learning (Rolls 2008a), ensuring that higher order spatial properties of the input stimuli are represented in the network. In VisNet, layer 1 corresponds to V2, layer 2 to V4, layer 3 to posterior inferior temporal visual cortex, and layer 4 to anterior inferior temporal cortex. Layer one is preceded by a simulation of the Gabor-like receptive fields of V1 neurons produced by each image presented to VisNet (Rolls 2012b).

- A convergent series of connections from a localized population of neurons in the preceding layer to each neuron of the following layer, thus allowing the receptive field size of neurons to increase through the visual processing areas or layers, as illustrated in Fig. 3.1.

- A modified associative (Hebb-like) learning rule incorporating a temporal trace of each neuron's previous activity, which, it has been shown (Földiák 1991; Rolls 1992, 2012b; Rolls and Milward 2000; Wallis and Rolls 1997a; Wallis et al. 1993), enables the neurons to learn transform invariances.

The learning rates for each of the four layers were 0.05, 0.03, 0.005, and 0.005, as these rates were shown to produce convergence of the synaptic weights after 15–50 training epochs. 50 training epochs were run.

The network implemented in VisNet

The network itself is designed as a series of hierarchical, convergent, competitive networks, in accordance with the hypotheses advanced above. The actual network consists of a series of four layers, constructed such that the convergence of information from the most disparate parts of the network's input layer can potentially influence firing in a single neuron in the final layer – see Fig. 3.1. This corresponds to the scheme described by many researchers (Rolls 1992, 2008a; Van Essen et al. 1992a, for example) as present in the primate visual system – see Fig. 3.1. The forward connections to a cell in one layer are derived from a topologically related and confined region of the preceding layer. The choice of whether a connection between neurons in adjacent layers exists or not is based upon a Gaussian distribution of connection probabilities which roll off radially from the focal point of connections for each neuron. (A minor extra constraint precludes the repeated connection of any pair of cells.) In particular, the forward connections to a cell in one layer come

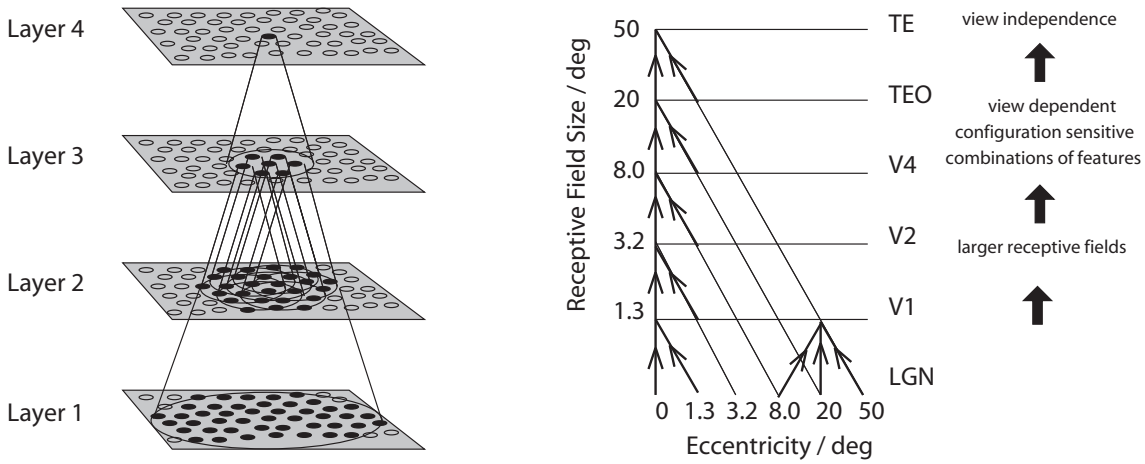


Figure 3.1: Convergence in the visual system. Right – as it occurs in the brain. V1, visual cortex area V1; TEO, posterior inferior temporal cortex; TE, inferior temporal cortex (IT). Left – as implemented in VisNet. Convergence through the network is designed to provide fourth layer neurons with information from across the entire input retina.

from a small region of the preceding layer defined by the radius in Table 3.1 which will contain approximately 67% of the connections from the preceding layer. Table 3.1 shows the dimensions for the research described here, a (16x) larger version than the version of VisNet used in most of our previous investigations, which utilized 32x32 neurons per layer. For the research in this chapter, the number of connections to layer 1 neurons was reduced to 100 (from 272), in order to increase the selectivity of the network between objects. The number of connections to each neuron in layers 2–4 to was increased to 400 (from 100), because this helped layer 4 neurons to reflect evidence from neurons in previous layers about the large number of transforms (typically 100 transforms, from 4 views of each object and 25 locations) each of which corresponded to a particular object.

Table 3.1: VisNet dimensions

	Dimensions	# Connections	Radius
Layer 4	128x128	400	48
Layer 3	128x128	400	36
Layer 2	128x128	400	24
Layer 1	128x128	100	24
Input layer	256x256x16	–	–

Figure 3.1 shows the general convergent network architecture used. Localization and limitation of connectivity in the network is intended to mimic cortical connectivity, partially because of the clear retention of retinal topology through regions of visual cortex. This architecture also encourages the gradual combination of features from layer to layer which has relevance to the binding problem, as described elsewhere Rolls (2008a, 2012b).

Competition and lateral inhibition in VisNet

In order to act as a competitive network some form of mutual inhibition is required within each layer, which should help to ensure that all stimuli presented are evenly represented by the neurons

Table 3.2: Sigmoid parameters

Layer	1	2	3	4
Percentile	99.2	98	88	95
Slope β	190	40	75	26

in each layer. This is implemented in VisNet by a form of lateral inhibition. The idea behind the lateral inhibition, apart from this being a property of cortical architecture in the brain, was to prevent too many neurons that received inputs from a similar part of the preceding layer responding to the same activity patterns. The purpose of the lateral inhibition was to ensure that different receiving neurons coded for different inputs. This is important in reducing redundancy Rolls (2008a). The lateral inhibition is conceived as operating within a radius that was similar to that of the region within which a neuron received converging inputs from the preceding layer (because activity in one zone of topologically organized processing within a layer should not inhibit processing in another zone in the same layer, concerned perhaps with another part of the image).

The lateral inhibition and contrast enhancement just described are actually implemented in VisNet2 Rolls and Milward (2000) and VisNet Perry, Rolls, and Stringer (2010) in two stages, to produce filtering of the type illustrated elsewhere Rolls (2008a, 2012b). The lateral inhibition was implemented by convolving the activation of the neurons in a layer with a spatial filter, I, where δ controls the contrast and σ controls the width, and a and b index the distance away from the centre of the filter

$$I_{a,b} = \begin{cases} -\delta e^{-\frac{a^2+b^2}{\sigma^2}} & \text{if } a \neq 0 \text{ or } b \neq 0, \\ 1 - \sum\limits_{a \neq 0, b \neq 0} I_{a,b} & \text{if } a = 0 \text{ and } b = 0. \end{cases} \tag{3.1}$$

The second stage involves contrast enhancement. A sigmoid activation function was used in the way described previously Rolls and Milward (2000):

$$y = \mathrm{f}^{\mathrm{sigmoid}}(r) = \frac{1}{1 + e^{-2\beta(r-\alpha)}} \tag{3.2}$$

where r is the activation (or firing rate) of the neuron after the lateral inhibition, y is the firing rate after the contrast enhancement produced by the activation function, and β is the slope or gain and α is the threshold or bias of the activation function. The sigmoid bounds the firing rate between 0 and 1 so global normalization is not required. The slope and threshold are held constant within each layer. The slope is constant throughout training, whereas the threshold is used to control the sparseness of firing rates within each layer. The (population) sparseness of the firing within a layer is defined Franco, Rolls, Aggelopoulos, and Jerez (2007); Rolls (2008a); Rolls and Treves (1998, 2011) as:

$$a = \frac{(\sum_i y_i/n)^2}{\sum_i y_i^2/n} \tag{3.3}$$

where n is the number of neurons in the layer. To set the sparseness to a given value, e.g. 5%, the threshold is set to the value of the 95th percentile point of the activations within the layer.

The sigmoid activation function was used with parameters (selected after a number of optimization runs) as shown in Table 3.2.

In addition, the lateral inhibition parameters are as shown in Table 3.3.

Table 3.3: Lateral inhibition parameters

Layer	1	2	3	4
Radius, σ	1.38	2.7	4.0	6.0
Contrast, δ	1.5	1.5	1.6	1.4

The VisNet trace learning rule

The learning rule implemented in the VisNet simulations utilizes the spatio-temporal constraints placed upon the behaviour of 'real-world' objects to learn about natural object transformations. By presenting consistent sequences of transforming objects the cells in the network can learn to respond to the same object through all of its naturally transformed states, as described by Földiák (1991), Rolls (1992), Wallis, Rolls, and Földiák (1993), Wallis and Rolls (1997a), and Rolls (2012b). The learning rule incorporates a decaying trace of previous cell activity and is henceforth referred to simply as the 'trace' learning rule. The learning paradigm described here is intended in principle to enable learning of any of the transforms tolerated by inferior temporal cortex neurons, including position, size, view, lighting, and spatial frequency (Rolls 1992, 2000, 2008a, 2012b; Rolls and Deco 2002).

Various biological bases for this temporal trace have been advanced as follows: The precise mechanisms involved may alter the precise form of the trace rule which should be used. Földiák 1992 describes an alternative trace rule which models individual NMDA channels. Equally, a trace implemented by temporally extended cell firing in a local cortical attractor could implement a short-term memory of previous neuronal firing (Rolls 2008a).

- The persistent firing of neurons for as long as 100–400 ms observed after presentations of stimuli for 16 ms (Rolls and Tovee 1994) could provide a time window within which to associate subsequent images. Maintained activity may potentially be implemented by recurrent connections between as well as within cortical areas (Rolls 2008a; Rolls and Deco 2002; Rolls and Treves 1998). The prolonged firing of inferior temporal cortex neurons during memory delay periods of several seconds, and associative links reported to develop between stimuli presented several seconds apart (Miyashita 1988) are on too long a time scale to be immediately relevant to the present theory. In fact, associations between visual events occurring several seconds apart would, under *normal* environmental conditions, be detrimental to the operation of a network of the type described here, because they would probably arise from different objects. In contrast, the system described benefits from associations between visual events which occur close in time (typically within 1 s), as they are likely to be from the same object.

- The binding period of glutamate in the NMDA channels, which may last for 100 ms or more, may implement a trace rule by producing a narrow time window over which the *average* activity at each pre-synaptic site affects learning (Földiák 1992; Hestrin, Sah, and Nicoll 1990; Rhodes 1992; Rolls 1992; Spruston, Jonas, and Sakmann 1995).

- Chemicals such as nitric oxide may be released during high neural activity and gradually decay in concentration over a short time window during which learning could be enhanced (Földiák 1992; Garthwaite 2008; Montague, Gally, and Edelman 1991).

The trace update rule used in the baseline simulations of VisNet (Wallis and Rolls 1997a) is equivalent to both Földiák's used in the context of translation invariance (Wallis, Rolls, and Földiák 1993) and to the earlier rule of Sutton and Barto (1981) explored in the context of modelling the temporal properties of classical conditioning, and can be summarized as follows:

$$\delta w_j = \alpha \overline{y}^{\tau} x_j \tag{3.4}$$

where

$$\overline{y}^{\tau} = (1 - \eta) y^{\tau} + \eta \overline{y}^{\tau - 1} \tag{3.5}$$

and

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

At the start of a series of investigations of different forms of the trace learning rule, Rolls and Milward (2000) demonstrated that VisNet's performance could be greatly enhanced with a modified Hebbian trace learning rule (equation 3.6) that incorporated a trace of activity from the preceding time steps, with no contribution from the activity being produced by the stimulus at the current time step. This rule took the form

$$\delta w_j = \alpha \overline{y}^{\tau - 1} x_j^{\tau}. \tag{3.6}$$

The trace shown in equation 3.6 is in the post-synaptic term. The crucial difference from the earlier rule (see equation 3.4) was that the trace should be calculated up to only the preceding time-step. This has the effect of updating the weights based on the preceding activity of the neuron, which is likely given the spatio-temporal statistics of the visual world to be from previous transforms of the same object (Rolls and Milward 2000; Rolls and Stringer 2001). This is biologically not at all implausible, as considered in more detail elsewhere (Rolls 2008a, 2012b), and this version of the trace rule was used in this investigation.

The optimal value of η in the trace rule is likely to be different for different layers of VisNet. For early layers with small receptive fields, few successive transforms are likely to contain similar information within the receptive field, so the value for η might be low to produce a short trace. In later layers of VisNet, successive transforms may be in the receptive field for longer, and invariance may be developing in earlier layers, so a longer trace may be beneficial. In practice, after exploration we used η values of 0.6 for layer 2, and 0.8 for layers 3 and 4. In addition, it is important to form feature combinations with high spatial precision before invariance learning supported by a temporal trace starts, in order that the feature combinations and not the individual features have invariant representations (Rolls 2008a, 2012b). For this reason, purely associative learning with no temporal trace was used in layer 1 of VisNet (Rolls and Milward 2000).

The following principled method was introduced to choose the value of the learning rate α for each layer. The mean weight change from all the neurons in that layer for each epoch of training was measured, and was set so that with slow learning over 15–50 trials, the weight changes per epoch

would gradually decrease and asymptote with that number of epochs, reflecting convergence. Slow learning rates are useful in competitive nets, for if the learning rates are too high, previous learning in the synaptic weights will be overwritten by large weight changes later within the same epoch produced if a neuron starts to respond to another stimulus (Rolls 2008a). If the learning rates are too low, then no useful learning or convergence will occur. It was found that the following learning rates enabled good operation with the 100 transforms of each of 4 stimuli used in each epoch in the present investigation: Layer 1 α=0.05; Layer 2 α=0.03 (this is relatively high to allow for the sparse representations in layer 1); Layer 3 α=0.005; Layer 4 α=0.005.

To bound the growth of each neuron's synaptic weight vector, $\mathbf{w}_i$ for the ith neuron, its length is explicitly normalized (a method similarly employed by von der Malsburg 1973 which is commonly used in competitive networks (Rolls 2008a)). An alternative, more biologically relevant implementation, using a local weight bounding operation which utilizes a form of heterosynaptic long-term depression (Rolls 2008a), has in part been explored using a version of the Oja 1982 rule (see Wallis and Rolls (1997a)).

The input to VisNet

VisNet is provided with a set of input filters which can be applied to an image to produce inputs to the network which correspond to those provided by simple cells in visual cortical area 1 (V1). The purpose of this is to enable within VisNet the more complicated response properties of cells between V1 and the inferior temporal cortex (IT) to be investigated, using as inputs natural stimuli such as those that could be applied to the retina of the real visual system. This is to facilitate comparisons between the activity of neurons in VisNet and those in the real visual system, to the same stimuli. In VisNet no attempt is made to train the response properties of simple cells, but instead uses a predefined series of filters to perform fixed feature extraction to a level equivalent to that of simple cells in V1, as have other researchers in the field (Buhmann, Lange, von der Malsburg, Vorbrüggen, and Würtz 1991; Fukushima 1980; Hummel and Biederman 1992). The aim is to simulate the more complicated response properties of cells between V1 and the inferior temporal cortex (IT). The elongated orientation-tuned input filters used accord with the general tuning profiles of simple cells in V1 (Hawken and Parker 1987) and were computed by Gabor filters. Each individual filter is tuned to spatial frequency (0.0626 to 0.5 cycles / pixel over four octaves); orientation (0^o to 135^o in steps of 45^o); and sign (± 1). Of the 100 layer 1 connections, the number to each group in VisNet is as shown in Table 3.4. Any zero D.C. filter can of course produce a negative as well as positive output, which would mean that this simulation of a simple cell would permit negative as well as positive firing. The response of each filter is zero thresholded and the negative results used to form a separate anti-phase input to the network. The filter outputs are also normalized across scales to compensate for the low frequency bias in the images of natural objects.

The Gabor filters used were similar to those used previously (Deco and Rolls 2004; Rolls 2012b; Rolls and Webb 2014; Webb and Rolls 2014). Following Daugman (1988a) the receptive fields of the simple cell-like input neurons are modelled by 2D-Gabor functions. The Gabor receptive

Table 3.4: VisNet layer 1 connectivity. The frequency is in cycles per pixel.

Frequency	0.5	0.25	0.125	0.0625
# Connections	74	19	5	2

fields have five degrees of freedom given essentially by the product of an elliptical Gaussian and a complex plane wave. The first two degrees of freedom are the 2D-locations of the receptive field's centre; the third is the size of the receptive field; the fourth is the orientation of the boundaries separating excitatory and inhibitory regions; and the fifth is the symmetry. This fifth degree of freedom is given in the standard Gabor transform by the real and imaginary part, i.e by the phase of the complex function representing it, whereas in a biological context this can be done by combining pairs of neurons with even and odd receptive fields. This design is supported by the experimental work of Pollen and Ronner (1981), who found simple cells in quadrature-phase pairs. Even more, Daugman (1988a) proposed that an ensemble of simple cells is best modelled as a family of 2D-Gabor wavelets sampling the frequency domain in a log-polar manner as a function of eccentricity. Experimental neurophysiological evidence constrains the relation between the free parameters that define a 2D-Gabor receptive field (De Valois and De Valois 1988). There are three constraints fixing the relation between the width, height, orientation, and spatial frequency (Lee 1996). The first constraint posits that the aspect ratio of the elliptical Gaussian envelope is 2:1. The second constraint postulates that the plane wave tends to have its propagating direction along the short axis of the elliptical Gaussian. The third constraint assumes that the half-amplitude bandwidth of the frequency response is about 1 to 1.5 octaves along the optimal orientation. Cells of layer 1 receive a topologically consistent, localized, random selection of the filter responses in the input layer, under the constraint that each cell samples every filter spatial frequency and receives a constant number of inputs. The mathematical details of the Gabor filtering are described elsewhere (Rolls 2012b; Rolls and Webb 2014; Webb and Rolls 2014).

Recent developments in VisNet implemented in the research described here

The version of VisNet used in this thesis differed from the versions used for most of the research published with VisNet before 2012 (Rolls 2012b) in the following ways. First, Gabor filtering was used here, with a full mathematical description provided here, as compared to the Difference of Gaussian filters used earlier. Second, the size of VisNet was increased from the previous 32x32 neurons per layer to the 128x128 neurons per layer described here. Third, the steps described in the Method to set the learning rates α to values for each layer that encouraged convergence in 20–50 learning epochs were utilized here. Fourth, the method of pattern association decoding described in Section 3.2.8 to provide a biologically plausible way of decoding the outputs of VisNet neurons was used in the research described here. Fuller descriptions of the rationale for the design of VisNet, and of alternative more powerful learning rules not used here, are provided elsewhere (Rolls 2008a, 2012b; Rolls and Stringer 2001).

The HMAX models used for comparison with VisNet

The performance of VisNet is to be compared against the popular HMAX model. HMAX was originally proposed by Riesenhuber and Poggio (1999a), though has seen numerous additions and extensions (Mutch and Lowe 2008; Serre et al. 2007b,c). Generically HMAX is a hierarchical feature extraction pipeline, with parameters that are constrained to be biologically relevant. Like other hierarchical feature models (Fukushima 1980), it uses lateral copying of filters (i.e convolutional structure), and an unsupervised learning process to build up the feature hierarchy parameters. This is in contrast to convolutional deep networks networks (LeCun et al. 2010) which typically leverage

supervised learning via backpropagation of errors - a process that does not aim for biologically plausibility (Rolls 2008a).

HMAX is composed of numerous heterogeneous layers of 'computational units', alternating between 'simple', S and 'complex', C units. Generically, the S units combine multiple preceeding layer outputs with a Gaussian shaped tuning function that modelling selectivity, while C units pool across previous outputs using the standard maximum operator to increase invariance. As you go up through the layers the filters used in each unit become larger in accordance with the receptive field size increases measured within the visual cortex. The first layer, S_1 is made up of units that collectively form a filter bank of Gabor functions (in much the same way as VisNet) over numerous scales and orientations. Units in the proceeding S_n layers have their maximally responsive stimuli artificially set to be selective to random patches of the preceding C_{n-1} layer. This random sampling process is used as a crude learning process - visual features that are most often represented in the training images have an increased probability to be sampled in this process and have a S_n unit specifically selective to that feature. The inspiration for this architecture Riesenhuber and Poggio (1999a) may have come from the simple and complex cells found in V1 by Hubel and Wiesel (1968a). A diagram of the model as described by Riesenhuber and Poggio (1999a) is shown in Fig. 3.2. The final complex cell layer, C_n is typically used as an input to a non-biologically plausible support vector machine or least squares computation to perform classification of the representations into object classes. One difference is that VisNet is normally trained on images generated by objects as they transform in the world, so that view, translation, size, rotation etc invariant representations of objects can be learned by the network. In contrast, HMAX is typically trained with large databases of pictures of different exemplars of for example hats and beer mugs as in the Caltech databases, which do not provide the basis for invariant representations of specific objects to be learned, but are aimed at object classification.

When assessing the biological plausibility of the output representations of HMAX the implementation of the HMAX model described by Mutch and Lowe (2008) was used. In this instantiation of HMAX with 2 layers of S-C units, the assessment of performance was typically made using a support vector machine applied to the top layer C neurons. However, that way of measuring performance is not biologically plausible. However, Serre et al. (2007b) took the C2 neurons as corresponding to V4, and following earlier work in which View Tuned Units were implemented Riesenhuber and Poggio (1999a), added a set of View Tuned Units (VTU) which might be termed an S3 layer which they suggest corresponds to the posterior inferior temporal visual cortex. These VTUs were implemented in the way described by Riesenhuber and Poggio (1999a) and Serre et al. (2007b) with an S3 VTU layer, by setting up a moderate number of view tuned units, each one of which is set to have connection weights to all neurons in the C2 layer that reflect the firing rate of each C2 unit to one exemplar of a class. (This will produce the firing for any VTU that would be produced by one of the training views or exemplars of a class.) The S3 units that were implemented can thus be thought of as representing posterior inferior temporal cortex neurons Serre et al. (2007b).

To ensure that the particular implementation of HMAX that is used for the experiments - that of Mutch and Lowe (2008) - were not different generically in the results obtained from other HMAX-type family of models, further investigations were performed with the version of HMAX described by Serre et al. (2007b), which has 3 S-C layers. The S3 layer is supposed to correspond to posterior inferior temporal visual cortex, and the C3 layer, which is followed by S4 View Tuned Units, to

anterior inferior temporal visual cortex. The results with this version of HMAX were found to be generically similar in our investigations to those of the version implemented by Mutch and Lowe (2008) described in this chapter and hence are omitted for brevity. Note that the code used for all the HMAX investigations is available[1] and furthermore that code defines the details of the architecture and the parameters, which were used unless otherwise stated, and for that reason the parameter details of the HMAX implementations are not considered in great detail here. Also, of note is that the HMAX family of models have on the order of 10 million computational units (Serre et al. 2007b), which is at least 100 times the number contained within the current implementation of VisNet (which uses 128x128 neurons in each of 4 layers, i.e. 65,536 neurons).

Measures for network performance

Information theory measures

The performance of VisNet has historically been measured by Shannon information-theoretic measures that are identical to those used to quantify the specificity and selectiveness of the representations provided by neurons in the brain (Rolls 2012b; Rolls and Milward 2000; Rolls and Treves 2011). Two such metrics have been used: A single cell information measure indicated how much information was conveyed by the firing rates of a single neuron about the most effective stimulus and a multiple cell information measure indicated how much information about every stimulus was conveyed by the firing rates of small populations of neurons.

[1] http://cbcl.mit.edu/jmutch/cns/index.html#hmax

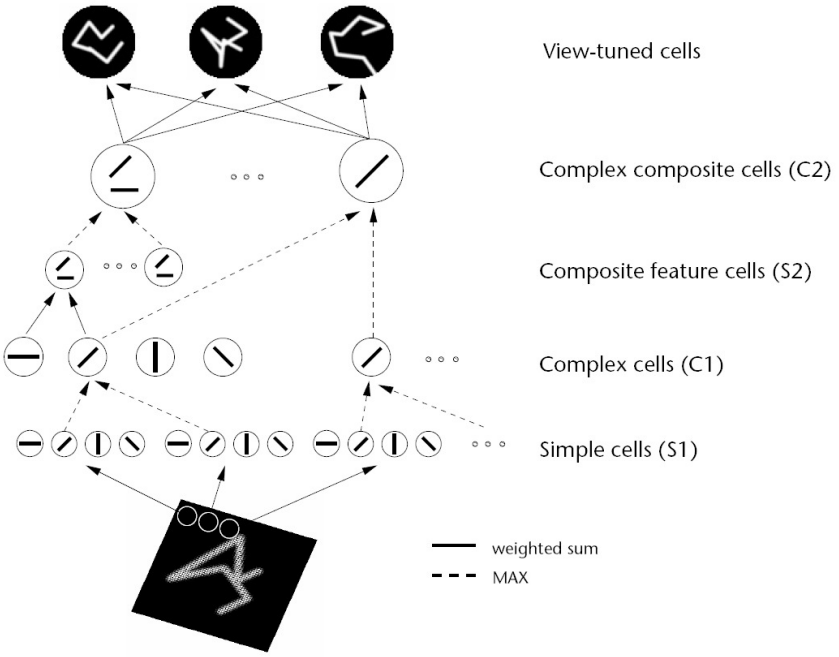


Figure 3.2: Sketch of the HMAX model of invariant object recognition (Riesenhuber and Poggio 1999a, 2000a). The model includes layers of 'S' cells which perform template matching (solid lines), and 'C' cells (solid lines) which pool information by a non-linear MAX function to achieve invariance (see text). (After (Riesenhuber and Poggio 1999a))

A neuron can be said to have learnt an invariant representation if it discriminates one set of stimuli from another set, across all transforms. For example, a neuron's response is translation invariant if its response to one set of stimuli irrespective of presentation is consistently higher than for all other stimuli irrespective of presentation location. Note that some care must be taken in referring to a 'set of stimuli' since neurons in the inferior temporal cortex are not generally selective for a single stimulus but rather a subpopulation of stimuli (Abbott, Rolls, and Tovee 1996; Baylis, Rolls, and Leonard 1985; Franco, Rolls, Aggelopoulos, and Jerez 2007; Rolls 2007, 2008a; Rolls and Deco 2002; Rolls and Treves 1998, 2011; Rolls, Treves, and Tovee 1997b).

For completeness, the two information theoretic measures will be briefly defined here - which are described in detail by Rolls and Milward (2000) (see Rolls (2008a) and Rolls and Treves (2011) for an introduction to the concepts). The measures assess the extent to which either a single cell, or a population of cells, responds to the same stimulus invariantly with respect to its location, yet responds differently to different stimuli. The measures effectively show what one learns about which stimulus was presented from a single presentation of the stimulus at any randomly chosen transform. Results for top (4th) layer cells are shown. High information measures thus show that cells fire similarly to the different transforms of a given stimulus (object), and differently to the other stimuli. The single cell stimulus-specific information, $I(s, R)$, is the amount of information the set of responses, R, has about a specific stimulus, s (see Rolls, Treves, Tovee, and Panzeri (1997a) and Rolls and Milward (2000)). $I(s, R)$ is given by

$$I(s, R) = \sum_{r \in R} P(r|s) \log_2 \frac{P(r|s)}{P(r)} \tag{3.7}$$

where r is an individual response from the set of responses R of the neuron. For each cell the performance measure used was the maximum amount of information a cell conveyed about any one stimulus. This (rather than the mutual information, $I(S, R)$ where S is the whole set of stimuli s), is appropriate for a competitive network in which the cells tend to become tuned to one stimulus. ($I(s, R)$ has more recently been called the stimulus-specific surprise (DeWeese and Meister 1999; Rolls and Treves 2011). Its average across stimuli is the mutual information $I(S, R)$.)

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

$$I(S, S') = \sum_{s,s'} P(s, s') \log_2 \frac{P(s, s')}{P(s)P(s')} \tag{3.8}$$

This was calculated for the subset of cells which had as single cells the most information about

which stimulus was shown. In particular, in Rolls and Milward (2000) and subsequent papers, the multiple cell information was calculated from the first five cells for each stimulus that had maximal single cell information about that stimulus, that is from a population of 35 cells if there were seven stimuli (each of which might have been shown in for example 9 or 25 positions on the retina).

Pattern association decoding

In addition, the performance was measured by a biologically plausible one-layer pattern association network using an associative synaptic modification rule. There was one output neuron for each class (which was set to a firing rate of 1.0 during training of that class but was otherwise 0.0), and 10 input neurons per class to the pattern associator. These 10 neurons for each class were the most selective neurons in the output layer of VisNet or HMAX to each object. The most selective output neurons of VisNet and HMAX were identified as those with the highest mean firing rate to all transforms of an object relative to the firing rates across all transforms of all objects, and a high corresponding stimulus-specific information value for that class. Performance was measured as the percent correct object classification measured across all views of all objects.

The output of the inferior temporal visual cortex reaches structures such as the orbitofrontal cortex and amygdala, where associations to other stimuli are learned by a pattern association network with an associative (Hebbian) learning rule (Rolls 2008a, 2014). Therefore a one-layer pattern association network (Rolls 2008a) is used to measure how well the output of VisNet could be classified into one of the objects. The pattern association network had one output neuron for each object or class. The inputs were the 10 neurons from layer 4 of VisNet for each of the objects with the best single cell information and high firing rates. For HMAX, the inputs were the 10 neurons from the C2 layer (or from 5 of the View Tuned Units) for each of the objects with the highest mean firing rate for the class when compared to the firing rates over all the classes. The network was trained with the Hebb rule:

$$\delta w_{ij} = \alpha y_i x_j \tag{3.9}$$

where δw_{ij} is the change of the synaptic weight w_{ij} that results from the simultaneous (or conjunctive) presence of pre-synaptic firing x_j and post-synaptic firing or activation y_i, and α is a learning rate constant that specifies how much the synapses alter on any one pairing. The pattern associator was trained for one trial on the output of VisNet produced by every transform of each object.

Performance on the training or test images was tested by presenting an image to VisNet, and then measuring the classification produced by the pattern associator. Performance was measured by the percentage of the correct classifications of an image as the correct object.

This approach to measuring the performance is very biologically appropriate, for it models the type of learning thought to be implemented in structures that receive information from the inferior temporal visual cortex such as the orbitofrontal cortex and amygdala (Rolls 2008a, 2014). The small number of neurons selected from layer 4 of VisNet might correspond to the most selective for this stimulus set in a sparse distributed representation (Rolls 2008a; Rolls and Treves 2011). The method would measure whether neurons of the type recorded in the inferior temporal visual cortex with good view and position invariance are developed in VisNet. In fact, an appropriate neuron for an input to such a decoding mechanism might have high firing rates to all or most of

the view and position transforms of one of the stimuli, and smaller or no responses to any of the transforms of other objects, as found in the inferior temporal cortex for some neurons (Booth and Rolls 1998; Hasselmo et al. 1989; Perrett et al. 1991), and as found for VisNet layer 4 neurons (Rolls and Webb 2014). Moreover, it would be inappropriate to train a device such as a support vector machine or even an error correction perceptron on the outputs of all the neurons in layer 4 of VisNet to produce 4 classifications, for such learning procedures, not biologically plausible (Rolls 2008a), could map the responses produced by a multilayer network with untrained random weights to obtain good classifications.

Results

Categorization of objects from benchmark object image sets: Experiment 1

The performance of HMAX and VisNet was compared on a test that has been used to measure the performance of HMAX Mutch and Lowe (2008); Serre, Oliva, and Poggio (2007b); Serre, Wolf, Bileschi, Riesenhuber, and Poggio (2007c) and indeed typical of many approaches in computer vision, the use of standard datasets such as the CalTech-256 Griffin et al. (2007) in which sets of images from different object classes are to be classified into the correct object class.

Object benchmark database

The Caltech-256 dataset (Griffin et al. 2007) is comprised of 256 object classes made up of images that have many aspect ratios, sizes and differ quite significantly in quality (having being manually collated from web searches). The objects within the images show significant intra-class variation and have a variety of poses, illumination, scale and occlusion as expected from natural images (see examples in Fig. 3.3). In this sense, the Caltech-256 database has been considered to be a difficult challenge to object recognition systems (though in recent years it has been superseded by significantly larger datasets like ImageNet (Russakovsky et al. 2014)). It shall be seen, that one conclusion from the following experiments might be that the benchmarking approach with this type of dataset is not useful for training a system that must explicitly learn invariant object representations. The reason for this is that the exemplars of each object class in the CalTech-256 dataset are too discontinuous to provide a basis for learning transform invariant object representations. For example, the image exemplars within an object class in these datasets may be very different indeed.

Performance on a Caltech-256 test

VisNet and the HMAX model were trained to discriminate between two object classes from the Caltech-256 database, the *beer mugs* and *cowboy-hat* (see examples in Fig. 3.3). The images in each class were rescaled to 256×256 and converted to grayscale, so that shape recognition was being investigated. The images from each class were randomly partitioned into training and testing sets with performance measured in this cross-validation design over multiple random partitions. Figure 3.4 shows the performance of the VisNet and HMAX models when performing the task with these exemplars of the Caltech-256 dataset. Performance of HMAX and VisNet on the classification task was measured by the proportion of images classified correctly using a linear support vector

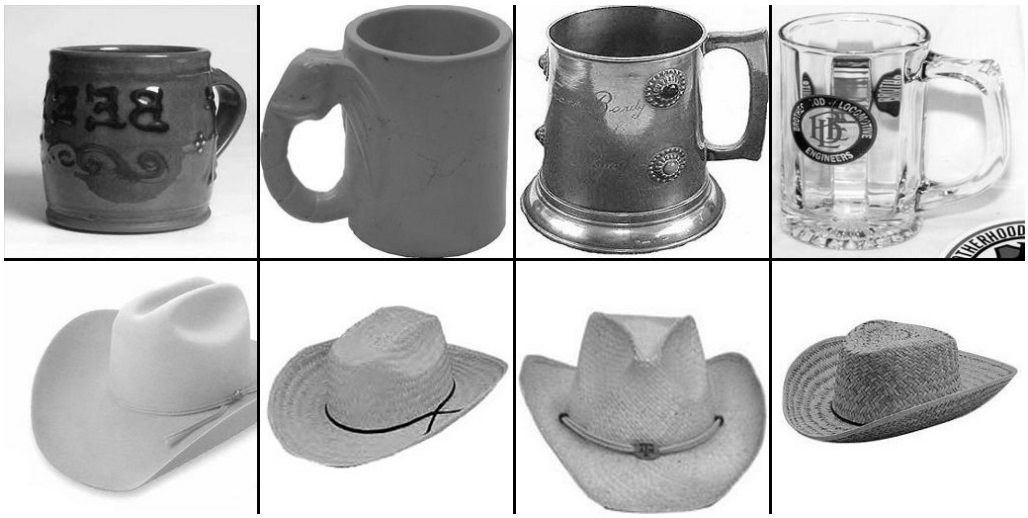


Figure 3.3: Example images from the Caltech256 database for two object classes, hats and beer mugs.

machine (SVM) on all the C2 cells in HMAX (chosen as the way often used to test the performance of HMAX Mutch and Lowe (2008); Serre et al. (2007b,c)) and on all the layer 4 (output layer) cells of VisNet. The error bars show the standard deviation of the means over 3 cross-validation trials with different images chosen at random for the training set and test set on each trial. The number of training exemplars is shown on the abscissa. There were 30 test examples of each object class. Chance performance at 50% is indicated. Performance of HMAX and VisNet was similar, but was poor, probably reflecting the fact that there is considerable variation of the images within each object class, making the cross-validation test quite difficult. The nature of the performance of HMAX and VisNet on this task is assessed in the next section.

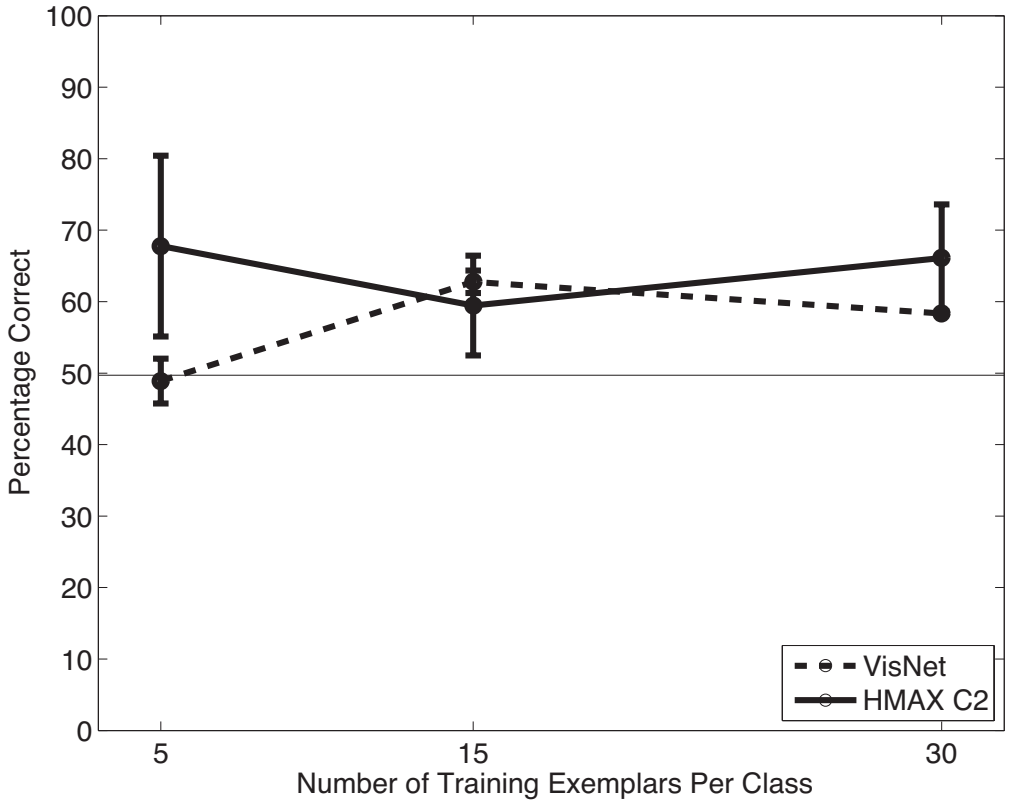


Figure 3.4: Performance of HMAX and VisNet on the classification task (measured by the proportion of images classified correctly) using the Caltech-256 dataset and linear support vector machine (SVM) classification. The error bars show the standard deviation of the means over 3 cross-validation trials with different images chosen at random for the training set on each trial. There were 2 object classes, hats and beer-mugs, with the number of training exemplars shown on the abscissa. There were 30 test examples of each object class. All cells in the C2 layer of HMAX and layer 4 of VisNet were used to measure the performance. Chance performance at 50% is indicated.

The biological plausibility of the neuronal representations of objects that are produced

In the temporal lobe visual cortical areas, neurons represent which object is present using a sparse distributed representation (Rolls and Treves 2011). Neurons typically have spontaneous firing rates of a few spikes/s, and increase their firing rates to 30–100 spikes/s for effective stimuli. Each neuron responds with a graded range of firing rates to a small proportion of the stimuli in what is therefore a sparse representation (Rolls and Tovee 1995; Rolls et al. 1997a). The information can be read from the firing of single neurons about which stimulus was shown, with often 2–3 bits of stimulus-specific information about the most effective stimulus (Rolls et al. 1997a; Tovee et al. 1993). The information from different neurons increases approximately linearly with the number of neurons recorded (up to approximately 20 neurons), indicating independent encoding by different neurons (Rolls et al. 1997b). The information from such groups of responsive neurons can be easily decoded (using for example dot product decoding utilizing the vector of firing rates of the neurons) by a pattern association network (Rolls 2008a, 2012b; Rolls and Treves 2011; Rolls et al. 1997b). This is very important for biological plausibility, for the next stage of processing, in brain regions such as the orbitofrontal and amygdala, contains pattern association networks that associate the outputs of the temporal cortex visual areas with stimuli such as taste (Rolls 2008a, 2014).

VisNet and HMAX are compared on the representations that they produce of objects, to analyze whether they produce these types of representation, which are needed for biological plausibility. It should be noted that the usual form of testing for VisNet does involve the identical measures used to measure the information present in the firing of temporal cortex neurons with visual responses (Rolls 2012b; Rolls and Milward 2000; Rolls et al. 1997a,b). On the other hand, the output of HMAX is typically read and classified by a powerful and artificial support vector machine (Mutch and Lowe 2008; Serre et al. 2007b,c), so it is necessary to test its output with the same type of biologically plausible neuronal firing rate decoding used by VisNet. Indeed, the results shown in section 3.3.1 were obtained with support vector machine decoding used for both HMAX and VisNet. In this section, the firing rate representations produced by VisNet and HMAX are analysed, to assess the biological plausibility of their output representations.

Figure 3.5 Upper shows the firing rates of two VisNet neurons for the test set, in the experiment with the Caltech-256 dataset using two object classes, beer mugs and hats, when trained on 30 exemplars of each class, and then tested in a cross-validation design with 10 test exemplars of each class that had not been seen during training. For the testing (untrained, cross-validation) set of exemplars, one of the neurons responded with a high rate to 8 of the 10 untrained exemplars of one class (hats), and to 1 of the exemplars of the other class (beer mugs). The single cell information was 0.38 bits. The other neuron responded to 5 exemplars of the beer mugs class, and to no exemplars of the hats class, and its single cell information was 0.21 bits. The mean stimulus-specific single cell information across the 5 most informative cells for each class was 0.28 bits. The results for the cross-validation testing mode shown in Fig. 3.5(upper) thus show that VisNet can learn about object classes, and can perform reasonable classification of untrained exemplars. Moreover, these results show that VisNet can do this using simple firing rate encoding of its outputs, which might potentially be decoded by a pattern associator. To test this, a pattern association network is trained on the output of VisNet to compare with the support vector machine results shown in Fig. 3.4. With 30 training exemplars, classification using the best 10 neurons for each class was 61.7% correct, compared to chance performance of 50% correct.

Figure 3.5 (middle) shows two neurons in the C2 and (bottom) two neurons in the View-Tuned Unit layer of HMAX on the test set of 10 exemplars of each class in the same task. It is clear that the C2 neurons both responded to all 10 untrained exemplars of both classes, with high firing rates to almost presented images. The normalized mean firing rate of one of the neurons was 0.905 to the beer mugs, and 0.900 to the hats. Again a pattern association network is used on the output of HMAX C2 neurons to compare with the support vector machine results shown in Fig. 3.4. With 30 training exemplars, classification using the best 10 neurons for each class was 63% correct, compared to chance performance of 50% correct. When biologically plausible decoding by an associative pattern association network is used, the performance of HMAX is poorer than when the performance of HMAX is measured with powerful least squares classification. The mean stimulus-specific single cell information across the 5 most informative cells for each class was 0.07 bits. This emphasizes that the output of HMAX is not in a biologically plausible form.

The relatively poor performance of VisNet (which produces a biologically plausible output), and of HMAX when its performance is measured in a biologically plausible way, raises the point that training with a diverse sets of exemplars of an object class as in the Caltech dataset is not a very useful way to test object recognition networks of the type found in the brain. Instead, the brain produces view invariant representations of objects, using information about view invariance simply not present in the Caltech type of dataset, because it does not provide training exemplars shown with different systematic transforms (position over up to 70°, size, rotation, and view) for transform invariance learning. In the next experiment, we therefore investigated the performance of HMAX and VisNet with a dataset in which different views of each object class are provided, to compare how HMAX and VisNet perform on this type of problem.

Figure 3.5 (bottom) shows the firing rates of two View Tuned layer Units of HMAX. It is clear that the View Tuned neurons had lower firing rates (and this is just a simple function of the value chosen for σ, which in this case was 1), but that again the firing rates differed little between the classes. (For example, the mean firing rate of one of the VTU neurons to the beer mugs was 0.3, and to the hats was 0.35. The single cell stimulus-specific information measures were 0.28 bits for the hats neuron, and 0.24 bits for the beer mugs neuron. The mean stimulus-specific single cell information across the 5 most informative VTUs for each class was 0.10 bits. Note that if the VTU layer was classified with a least squares classifier (i.e. a perceptron, which is not biologically plausible, but is how the VTU neurons were decoded by Serre et al. (2007b)), then performance was at 67%. (With a pattern associator, the performance was 66% correct.) Thus the performance of the VTU outputs (introduced to make the HMAX outputs of C neuron appear more biologically plausible) was poor on this type of CalTech-256 problem when measured both by a linear classifier and by a pattern association network.

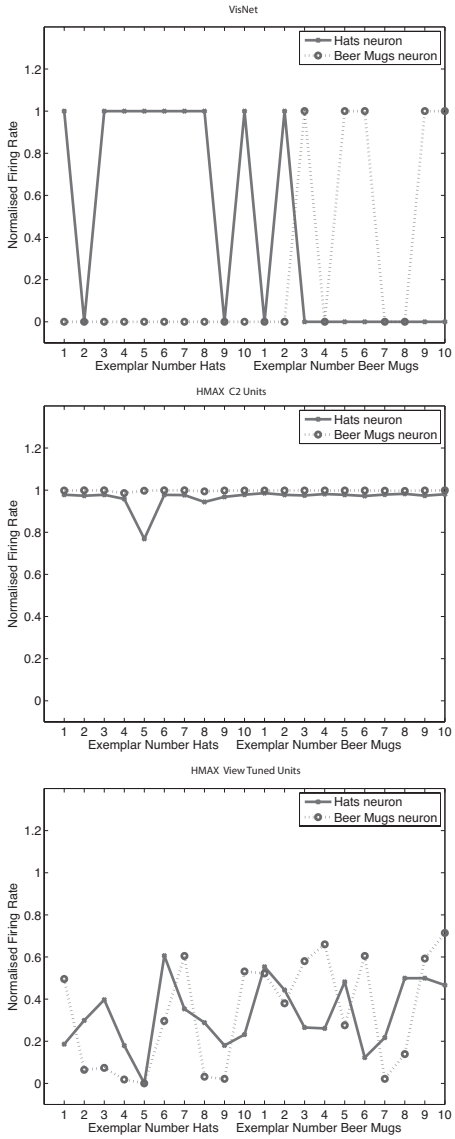


Figure 3.5: Top: Firing rate of two output layer neurons of VisNet, when tested on two of the classes, hats and beer mugs, from the Caltech 256. The firing rates to 10 untrained (i.e. testing) exemplars of each of the two classes are shown. One of the neurons responded more to hats than to beer mugs (solid line). The other neuron responded more to beer mugs than to hats (dashed line). Middle: Firing rate of two C2 Tuned Units of HMAX when tested on two of the classes, beer mugs and hats, from the Caltech 256. Bottom: Firing rate of a View Tuned Unit of HMAX when tested on two of the classes, hats (solid line) and beer mugs (dashed line), from the Caltech 256. The neurons chosen were those with the highest single cell information that could be decoded from the responses of a neuron to 10 exemplars of each of the 2 objects (as well as a high firing rate) in the cross-validation design.

Evaluation of categorisation when tested with large numbers of images presented randomly

The benchmark type of test using large numbers of images of different object classes presented in random sequence has limitations, in that an object can look quite different from different views. Catastrophic changes in the image properties of objects can occur as they are rotated through different views (Koenderink 1990). One example is that any view from above the cup into the cup that does not show the sides of the cup may look completely different from any view where some of the sides or bottom of the cup are shown. In this situation, training any network with images presented in a random sequence (i.e. without a classification label for each image) is doomed to failure in view-invariant object recognition. This applies to all such approaches that are unsupervised, and that attempt to categorize images into objects based on image statistics.

In contrast, the training of VisNet is based on the concept that the transforms of an object viewed from different angles in the natural world provide the information required about the different views of an object to build a view-invariant representation, and that this information can be linked together by the continuity of this process in time. Temporal continuity (Rolls 2012b), or even spatial continuity (Perry et al. 2010; Stringer et al. 2006), and typically both (Perry et al. 2006), provide the information that enables different images of an object to be associated together. Thus two factors, continuity of the image transforms as the object transforms through different views, and a principle of spatio-temporal closeness to provide a label of the object based on its property of spatio-temporal continuity, provides a principled way for VisNet, and it is proposed for the real visual system of primates including humans, to build invariant representations of objects (Rolls 1992, 2008a, 2012b). This led to Experiment 2.

Performance with the Amsterdam Library of Images: Experiment 2

Partly because of the limitations of the Caltech-256 database for training in invariant object recognition, a new set of experiments were undertaken with the Amsterdam Library of Images (ALOI) database Geusebroek et al. (2005)[2]. The ALOI database takes a different approach to the Caltech-256, and instead of focussing on a set of natural images within an object category or class, provides images of objects with a systematic variation of pose and illumination for 1000 small objects. Each object is placed onto a turntable and photographed in consistent conditions at 5 degree increments, resulting in a set of images that not only show the whole object (with regard to out of plane rotations), but does so with some continuity from one image to the next (see examples in Fig. 3.6).

Eight classes of object (with designations *156, 203, 234, 293, 299, 364, 674, 688*) from the dataset were chosen (see Fig. 3.6 for two examples). Each class or object comprises of 72 images taken at 5 degree increments through the full 360 degree horizontal plane of rotation. Three sets of training images were used as follows. The training set consisted of 4 views of each object spaced 90 degrees apart; or 9 views spaced 40 degrees apart; or 18 views spaced 20 degrees apart. The test set of images was in all cases a cross-validation set of 18 views of each object spaced 20 degrees apart and offset by 10 degrees from the training set with 18 views and not including any training view. The aim of using the different training sets was to investigate how close in viewing angle the training images need to be; and also to investigate the effects of using different numbers of training images.

[2]available at: http://staff.science.uva.nl/~aloi/

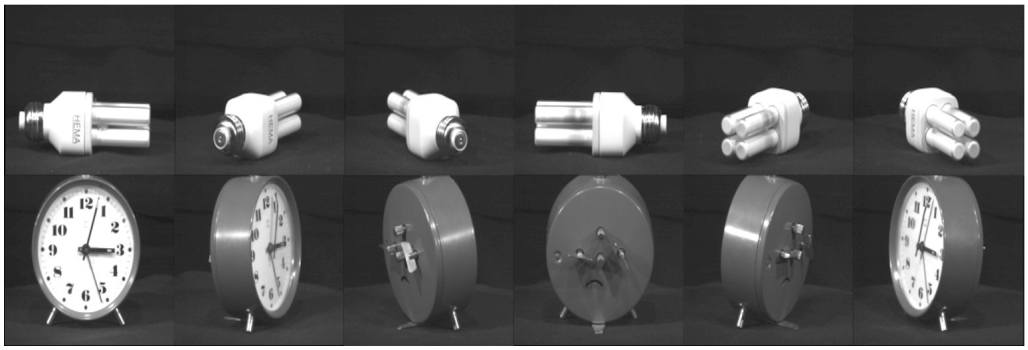


Figure 3.6: Example images from the two object classes within the ALOI database, (a) 293 (light bulb) and (b) 156 (clock). Only the 45 degree increments are shown.

The performance was measured with a pattern association network with one neuron per object and 10 inputs for each class that were the most selective neurons for an object in the output layer of VisNet or the C2 layer of HMAX. The best cells of VisNet or HMAX for a class were selected as those with the highest mean rate across views to the members of that class relative to the firing rate to all views of all objects; and with a high stimulus-specific information for that class. Figure 3.7 shows (measuring performance with a pattern associator trained on the 10 best cells for each of the 8 classes) that VisNet performed moderately well as soon as there were even a few training images, with the coding of its outputs thus shown to be suitable for learning by a pattern association network. In a statistical control it was found that an untrained VisNet performed at 18% correct when measured with the pattern association network compared with the 73% correct after training with 9 exemplars that is shown in Fig. 3.7. HMAX performed less well than VisNet. There was some information in the output of the HMAX C2 neurons, for if a powerful linear support vector machine (SVM) was used across all output layer neurons, the performance in particular for HMAX improved, with 78% correct for 4 training views and 93% correct for 9 training views and 92% correct for 18 training views (which in this case was also achieved by VisNet).

What VisNet can do here is to learn view invariant representations using its trace learning rule to build feature analysers that reflect the similarity across at least adjacent views of the training set. Very interestingly, with 9 training images, the view spacing of the training images was 40 degrees, and the test images in the cross-validation design were the intermediate views, 20 degrees away from the nearest trained view. This is promising, for it shows that enormous numbers of training images with many different closely spaced views are not necessary for VisNet. Even 9 training views spaced 40 degrees apart produced reasonable training.

We next compared the outputs produced by VisNet and HMAX, in order to assess their biological plausibility. Figure 3.8 Upper shows the firing rate of one output layer neuron of VisNet, when trained on 8 objects from the Amsterdam Library of Images, with 9 exemplars of each object with views spaced 40 degrees apart (set 2 described above). The firing rates on the training set are shown. The neuron responded to all 9 views of object 4 (a light bulb), and to no views of any other object. The neuron illustrated was chosen to have the highest single cell stimulus-specific information about object 4 that could be decoded from the responses of a neuron to the 9 exemplars of object 4 (as well as a high firing rate). That information was 3 bits. The mean stimulus-specific single cell information across the 5 most informative cells for each class was 2.2 bits. Figure 3.8

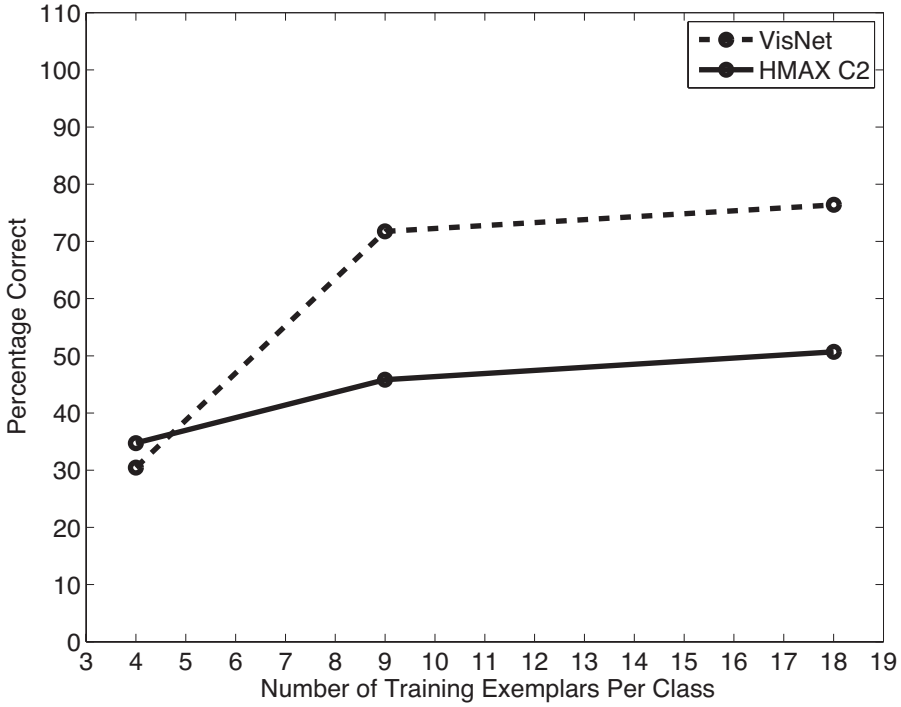


Figure 3.7: Performance of VisNet and HMAX C2 units measured by the percentage of images classified correctly on the classification task with 8 objects using the Amsterdam Library of Images dataset and measurement of performance using a pattern association network with one output neuron for each class. The training set consisted of 4 views of each object spaced 90 degrees apart; or 9 views spaced 40 degrees apart; or 18 views spaced 20 degrees apart. The test set of images was in all cases a cross-validation set of 18 views of each object spaced 20 degrees apart and offset by 10 degrees from the training set with 18 views and not including any training view. The 10 best cells from each class were used to measure the performance. Chance performance was 12.5% correct.

Middle shows the firing rate of one C2 unit of HMAX when trained on the same set of images. The unit illustrated was that with the highest mean firing rate across views to object 4 relative to the firing rates across all stimuli and views. The neuron responded mainly to one of the 9 views of object 4, with a small response to 2 nearby views. The neuron provided little information about object 4, even though it was the most selective unit for object 4. Indeed, the single cell stimulus-specific information for this C2 unit was 0.68 bits. The mean stimulus-specific single cell information across the 5 most informative C2 units for each class was 0.28 bits. Figure 3.8 Bottom shows the firing rate of one VTU of HMAX when trained on the same set of images. The unit illustrated was that with the highest firing rate to view 1 of object 4. Small responses can also be seen to view 2 of object 4, and to view 9 of object 4, but apart from this, most views of object 4 were not discriminated from the other objects. The single cell stimulus-specific information for this VTU was 0.28 bits. The mean stimulus-specific single cell information across the 5 most informative VTUs for each class was 0.67 bits.

The stimulus-specific single unit information measures show that the neurons of VisNet have much information in their firing rates about which object has been shown, whereas there is much less information in the firing rates of HMAX C2 units or View Tuned Units. The firing rates for

different views of an object are highly correlated for VisNet, but not for HMAX. This is further illustrated in Fig. 3.10, which shows the similarity between the outputs of the networks between the 9 different views of 8 objects produced by VisNet (top), HMAX C2 (middle), and HMAX VTUs (bottom) for the Amsterdam Library of Images test. Each panel shows a similarity matrix (based on the cosine of the angle between the vectors of firing rates produced by each object) between the 8 stimuli for all output neurons of each network. The maximum similarity is 1, and the minimal similarity is 0. The results are from the simulations with 9 views of each object spaced 40 degrees apart during training, with the testing results illustrated for the 9 intermediate views 20 degrees from the nearest trained view. For VisNet (top), it is shown that the correlations measured across the firing rates of all output neurons are very similar for all views of each object (apart from 2 views of object 1), and that the correlations with all views of every other object are close to 0.0. For HMAX C2 units, the situation is very different, with the outputs to all views of all objects being rather highly correlated, with a minimum correlation of 0.975. In addition, the similarity of the outputs produced by the different views of any given object are little more then the similarity with the views of other objects. This helps to emphasize the point that the firing within HMAX does not reflect well even a view of one object as being very different from the views of another object, let alone that different views of the same object produce similar outputs. This emphasises that for HMAX to produce measurably reasonable performance, most of the classification needs to be performed by a powerful classifier connected to the outputs of HMAX, not by HMAX itself. The HMAX VTU firing (bottom) was more sparse (σ was 1.0), but again the similarities between objects are frequently as great as the similarities within objects.

Experiment 2 thus shows that with the ALOI training set, VisNet can form separate neuronal representations that respond to all exemplars of each of 8 objects seen in different view transforms, and that single cells can provide perfect information from their firing rates to any exemplar about which object is being presented. The code can be read in a biologically plausible way with a pattern association network, which achieved 77% correct on the cross-validation set. Moreover, with training views spaced 40 degrees apart, VisNet performs moderately well (72% correct) on the intermediate views (20 degrees away from the nearest training view. In contrast, C2 output units of HMAX discriminate poorly between the object classes (Fig. 3.9 Middle), View Tuned Units of HMAX respond only to test views that are 20 degrees away from the training view, and the performance of HMAX tested with a pattern associator is correspondingly poor.

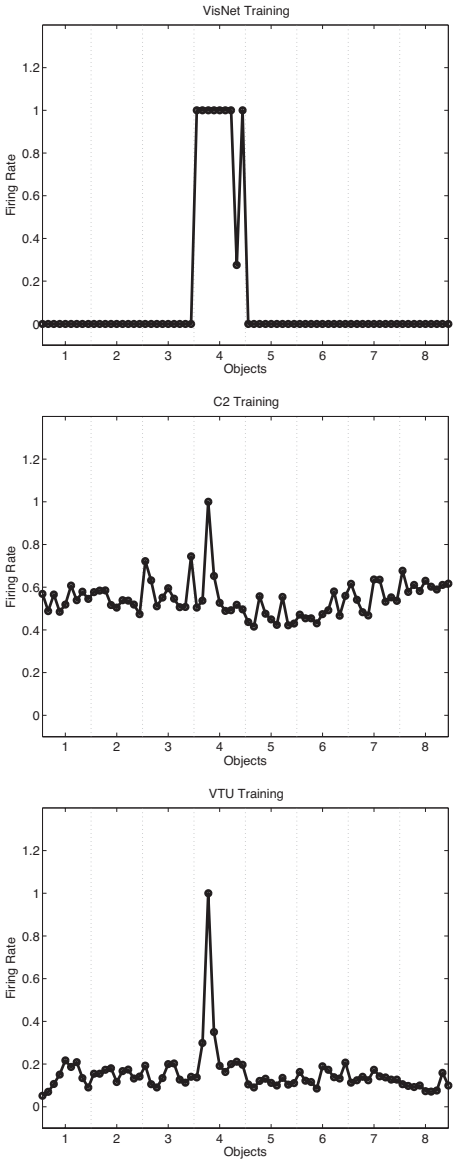


Figure 3.8: Top: Firing rate of one output layer neuron of VisNet, when trained on 8 objects from the Amsterdam Library of Images, with 9 views of each object spaced 40 degrees apart. The firing rates on the training set are shown. The neuron responded to all 9 views of object 4 (a light bulb), and to no views of any other object. The neuron illustrated was chosen to have the highest single cell stimulus-specific information about object 4 that could be decoded from the responses of the neurons to all 72 exemplars shown, as well as a high firing rate to object 4. Middle: Firing rate of one C2 Unit of HMAX when trained on the same set of images. The unit illustrated was that the highest mean firing rate across views to object 1 relative to the firing rates across all stimuli and views. Bottom: Firing rate of one View Tuned Unit (VTU) of HMAX when trained on the same set of images. The unit illustrated was that the highest firing rate to view 1 of object 1.

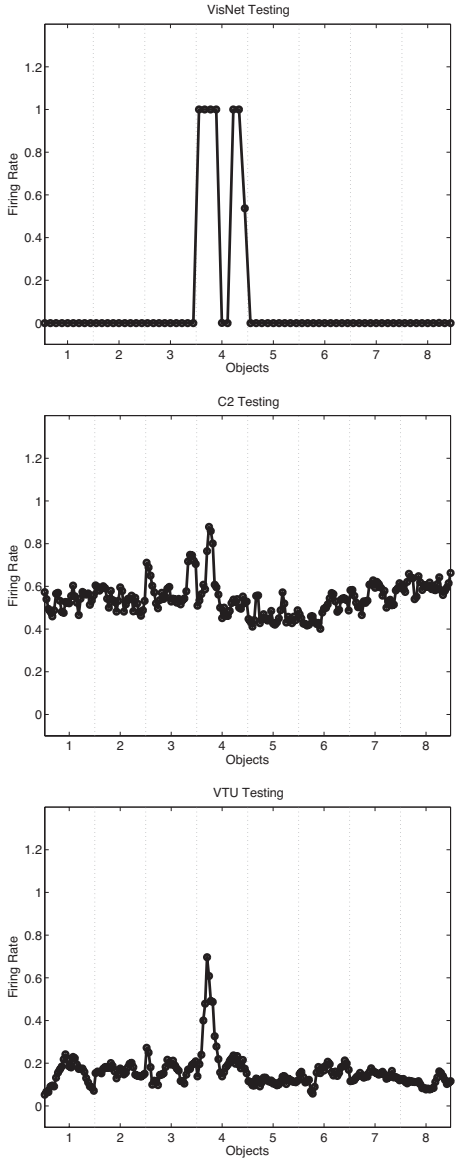


Figure 3.9: Top: Firing rate during cross-validation testing of one output layer neuron of VisNet, when trained on 8 objects from the Amsterdam Library of Images, with 9 exemplars of each object with views spaced 40 degrees apart. The firing rates on the cross-validation testing set are shown. The neuron was selected to respond to all views of object 4 of the training set, and as shown responded to 7 views of object 4 in the test set each of which was 20 degrees from the nearest training view, and to no views of any other object. Middle: Firing rate of one C2 Unit of HMAX when tested on the same set of images. The neuron illustrated was that the highest mean firing rate across training views to object 1 relative to the firing rates across all stimuli and views. The test images were 20 degrees away from the test images. Bottom: Firing rate of one View Tuned Unit (VTU) of HMAX when tested on the same set of images. The neuron illustrated was that the highest firing rate to view 1 of object 1 during training. It can be seen that the neuron responded with a rate of 0.8 to the two training images (1 and 9) of object 4 that were 20 degrees away from the image for which the VTU had been selected.

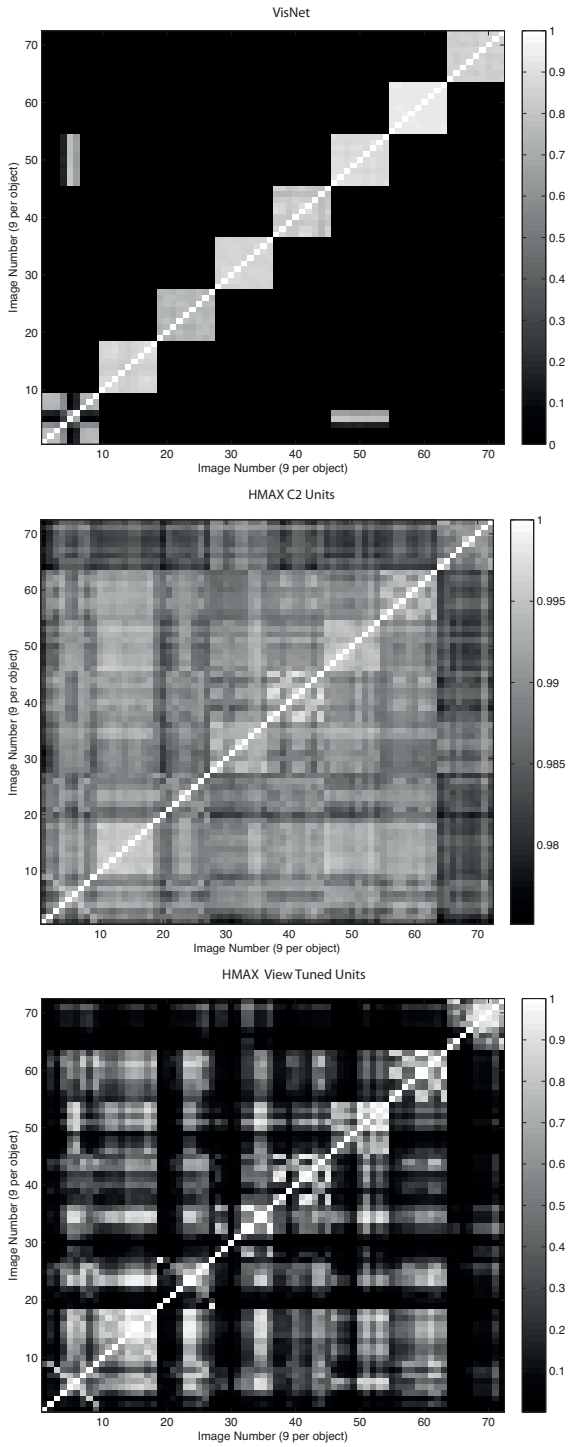


Figure 3.10: Similarity between the outputs of the networks between the 9 different views of 8 objects produced by VisNet (top), HMAX C2 (middle), and HMAX VTUs (bottom) for the Amsterdam Library of Images test. Each panel shows a similarity matrix (based on the cosine of the angle between the vectors of firing rates produced by each object) between the 8 stimuli for all output neurons of each type. The maximum similarity is 1, and the minimal similarity is 0.

Figure 3.11: Examples of images used in the scrambled faces experiment. Top: Two of the 8 faces in 2 of the 5 views of each. Bottom: examples of the scrambled versions of the faces.

The effects of rearranging the parts of an object: Experiment 3

Rearranging parts of an object can disrupt identification of the object, while leaving low-level features still present. Some face-selective neurons in the inferior temporal visual cortex do not respond to a face if its parts (e.g. eyes, nose, and mouth) are rearranged, showing that these neurons encode the whole object, and do not respond just to the features or parts (Perrett, Rolls, and Caan 1982; Rolls, Tovee, Purcell, Stewart, and Azzopardi 1994). Moreover, these neurons encode identity in that they respond differently to different faces (Baylis, Rolls, and Leonard 1985; Rolls and Treves 2011; Rolls, Treves, Tovee, and Panzeri 1997a; Rolls, Treves, and Tovee 1997b). It should be clear that some other neurons in the inferior temporal visual cortex respond to parts of faces such as eyes or mouth (Issa and DiCarlo 2012; Perrett, Rolls, and Caan 1982), consistent with the hypothesis that the inferior temporal visual cortex builds configuration-specific whole face or object representations from their parts, helped by feature combination neurons learned at earlier stages of the ventral visual system hierarchy (Rolls 1992, 2008a, 2012b, 2016).

To investigate whether neurons in the output layers of VisNet and HMAX can encode the identity of whole objects and faces (as distinct from their parts, low-level features etc), VisNet and HMAX were tested with both normal faces and with faces that have had their parts scrambled. The dataset was 8 faces from the ORL database of faces[3] each with 5 exemplars of different views, as illustrated in Fig. 3.11. The scrambling was performed by taking quarters of each face, and making 5 random permutations of the positions of each quarter. The procedure was to train on the set of unscrambled faces, and then to test how the neurons that responded best to each face then responded when the scrambled versions of the faces were shown, using randomly scrambled versions of the same eight faces each with the same set of 5 view exemplars.

[3]available at: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

VisNet was trained for 20 epochs, and performed 100% correct on the training set. When tested with the scrambled faces, performance was at chance, 12.5% correct, with 0.0 bits of multiple cell information using the 5 best cells for each class. An example of a VisNet layer 4 neuron that responded to one of the faces after training is shown in Fig. 3.12 top. The neuron responded to all the different view exemplars of the unscrambled face (and to no other faces in the training set). When the same neuron was then tested with the randomly scrambled versions of the same face stimuli, the firing rate was zero. In contrast, HMAX neurons did not show a reduction in their activity when tested with the same scrambled versions of the stimuli. This is illustrated in Fig. 3.12 bottom, in which the responses of a View Tuned Neuron (selected as the neuron with most selectivity between faces, and a response to exemplar 1 of one of the non-scrambled faces) were with similarly high firing rates to the scrambled versions of the same set of exemplars.

This experiment provides evidence that VisNet learns shape-selective responses that do not occur when the shape information is disrupted by scrambling. In contrast, HMAX must have been performing its discrimination between the faces based not on the shape information about the face that was present in the images, but instead on some lower-level property such as texture or feature information that was still present in the scrambled images. Thus VisNet performs with scrambled images in a way analogous to that of neurons in the inferior temporal visual cortex (Rolls et al. 1994).

The present result with HMAX is a little different from that reported by Riesenhuber and Poggio (1999a) where some decrease in the responses of neurons in HMAX was found after scrambling. We suggest that the difference is that in the study by Riesenhuber and Poggio (1999a) the responses were not of natural objects or faces, but were simplified paper-clip types of image, in which the degree of scrambling used would (in contrast to scrambling natural objects) leave little feature or texture information that may normally have a considerable effect on the responses of neurons in HMAX.

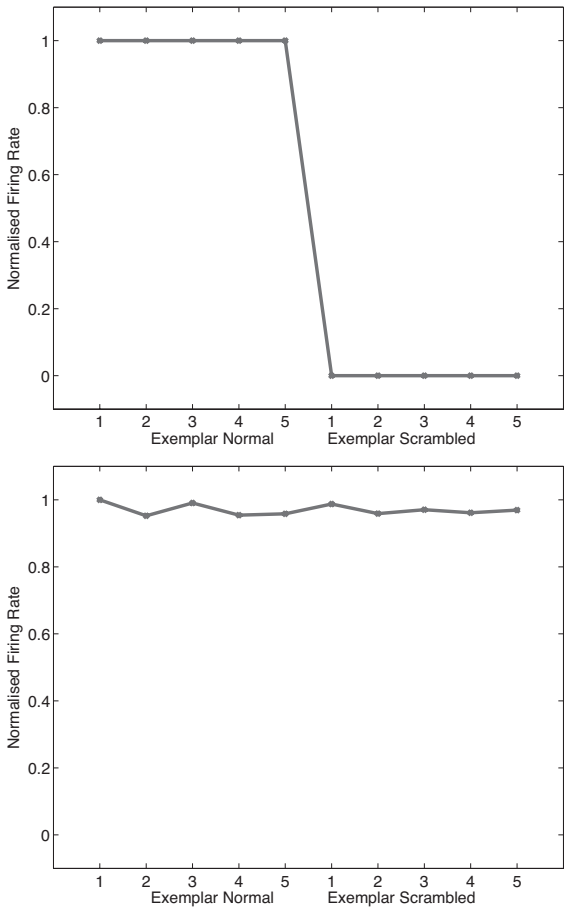


Figure 3.12: Top. Effect of scrambling on the responses of a neuron in VisNet. This VisNet layer 4 neuron responded to one of the faces after training, and to none of the other 7 faces. The neuron responded to all the different view exemplars 1–5 of the unscrambled face (exemplar normal). When the same neuron was then tested with the randomly scrambled versions of the same face stimuli (exemplar scrambled), the firing rate was zero. Bottom. Effect of scrambling on the responses of a neuron in HMAX. This View Tuned Neuron of HMAX was chosen to be as discriminating between the 8 face identities as possible. The neuron responded to all the different view exemplars 1–5 of the unscrambled face. When the same neuron was then tested with the randomly scrambled versions of the same face stimuli, the neuron responded with similarly high rates to the scrambled stimuli.

View invariant object recognition: Experiment 4

Some objects look different from different views (i.e. the images are quite different from different views), yet we can recognise the object as being the same from the different views. Further, some inferior temporal visual cortex neurons respond with view-invariant object representations, in that they respond selectively to some objects or faces independently of view using a sparse distributed representation (Booth and Rolls 1998; Hasselmo, Rolls, Baylis, and Nalwa 1989; Logothetis, Pauls, and Poggio 1995; Rolls 2012b; Rolls and Treves 2011). An experiment was designed to compare how VisNet and HMAX operate in view-invariant object recognition, by testing both on a problem in which objects had different image properties in different views. The prediction is that VisNet will be able to form by learning neurons in its output layer that respond to all the different views of one object, and to none of the different views of another object, whereas HMAX will not form neurons that encode objects, but instead will have its outputs dominated by the statistics of the individual images.

The objects used in the experiment are shown in Fig. 3.13. There were two objects, two cups, each with four views, constructed with Blender. VisNet was trained for 10 epochs, with all views of one object shown in random permuted sequence, then all views of the other object shown in random permuted sequence, to enable VisNet to use its temporal trace learning rule to learn about the different images that occurring together in time were likely to be different views of the same object. VisNet performed 100% correct in this task by forming neurons in its layer 4 that responded either to all views of one cup (labelled 'Bill') and to no views of the other cup (labelled 'Jane'), or vice versa, as illustrated in Fig. 3.14 top. Typical most highly discriminating C2 layer neurons of

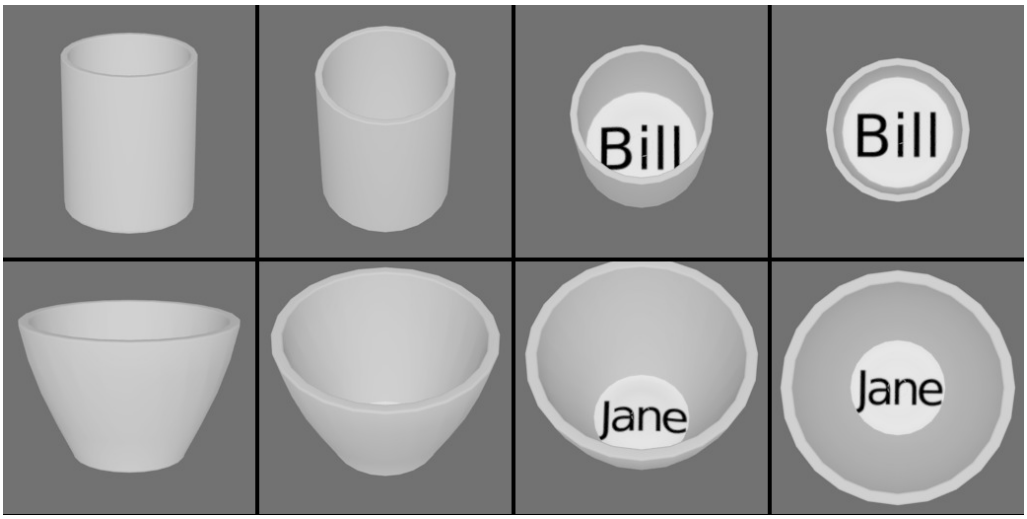


Figure 3.13: View invariant representations of cups. The two objects, each with four views.

HMAX are illustrated in Fig. 3.14 middle. The neurons did not discriminate between the objects, but instead responded more to the images of each object that contained text. This dominance by text is consistent with the fact that HMAX is up to this stage operating to a considerable extent as a set of image filters, the activity in which included much produced by the text. The performance of the C2 layer when decoded by the information analysis routines (using the 5 most object selective neurons for each class of object) was 50% correct (where chance was 50%), with 0.0 bits of information about which stimulus had been presented.

Typical most highly discriminating VTU (view-trained unit) layer neurons of HMAX are illustrated in Fig. 3.14 bottom. (Two VTUs were set up for each object during the analysis stage, one for a view of an object without text, and one for a view of an object with text. σ was set to 1.0.) The neurons did not discriminate between the objects, but instead responded much more to the images of an object that contained text. The performance of the VTU layer when decoded by the information analysis routines (using the 5 most object selective neurons for each class of object) was 50% correct (where chance was 50%), with 0.0 bits of information about which stimulus had been presented. A similarity matrix (based on the cosine of the angle between the vectors of firing rates produced by each stimulus) for the VTU neurons indicated that there were high correlations between the images that contained text, and high correlations between the images that did not contain text, but no correlations reflecting similar firing to all views of either object.

This experiment draws out a fundamental difference between VisNet and HMAX. The output layer neurons of VisNet can represent transform invariant properties of objects, and can form single neurons that respond to the different views of objects even when the images of the different views may be quite different, as is the case for many real-world objects when they transform in the world. Thus basing object recognition on image statistics, and categorisation based on these, is insufficient for transform-invariant object recognition. VisNet can learn to respond to the different transforms of objects using the trace learning rule to capture the properties of objects as they transform in the world. In contrast, HMAX up to the C2 layer sets some of its neurons to respond to exemplars in the set of images, but has no way of knowing which exemplars may be of the same object, and no way therefore to learn about the properties of objects as they transform in the real world, showing catastrophic changes in the image as they transform (Koenderink 1990), exemplified in the example in this experiment by the new views as the objects transform from not showing to showing writing in the base of the cup. Moreover, because the C2 neurons reflect mainly the way in which all the Gabor filters respond to image exemplars, the firing of C2 neurons is typically very similar and non-sparse to different images, though if the images have very different statistics in terms of for example text or not, it is these properties that dominate the firing of the C2 neurons.

Similarly, the VTU neurons of HMAX are set to have synaptic strengths proportional to the firing of the C2 neurons that provide inputs to the VTUs when one view of one object is shown (Serre, Oliva, and Poggio 2007b). Because there is little invariance in the C units, many different VTUs are needed, with one for each training view or exemplar. Because the VTUs are different to each other for the different views of the same object or class, a further stage of training is then needed to classify the VTUs into object classes, and the type of learning is least squares error minimization (Serre, Oliva, and Poggio 2007b), equivalent to a delta-rule one-layer perceptron which again is not biologically plausible for neocortex (Rolls 2008a). Thus HMAX does not generate invariant representations in its S–C hierarchy, and in the VTU approach uses two layers of learning after the S–C hierarchy, the second involving least squares learning, to produce classification.The representation can be more sparse than that of the C2 neurons depending on the value of σ, but nevertheless represents properties of an image, and not of objects. The output of HMAX thus does not provide in general transform invariant representations, but instead reflects statistical properties of images. Therefore the output of HMAX must be classified by a powerful classifier such as a support vector machine, which then has to learn the whole set of outputs of the visual processing that correspond to any one object in all its transforms and views. This is biologically implausible, with pattern associators being the most typical known classifier in the cerebral cortex

(Rolls 2008a). In any case, because the output of C2 is so difficult to interpret by a brain-like decoder such as a pattern associator, and because VTUs by definition respond to one of perhaps many views of an object, VTUs are not generally used in more recent work with HMAX, and instead the final C layer of firing is sent directly to a support vector machine classifier (Mutch and Lowe 2008; Serre, Kreiman, Kouh, Cadieu, Knoblich, and Poggio 2007a; Serre, Oliva, and Poggio 2007b; Serre, Wolf, Bileschi, Riesenhuber, and Poggio 2007c).

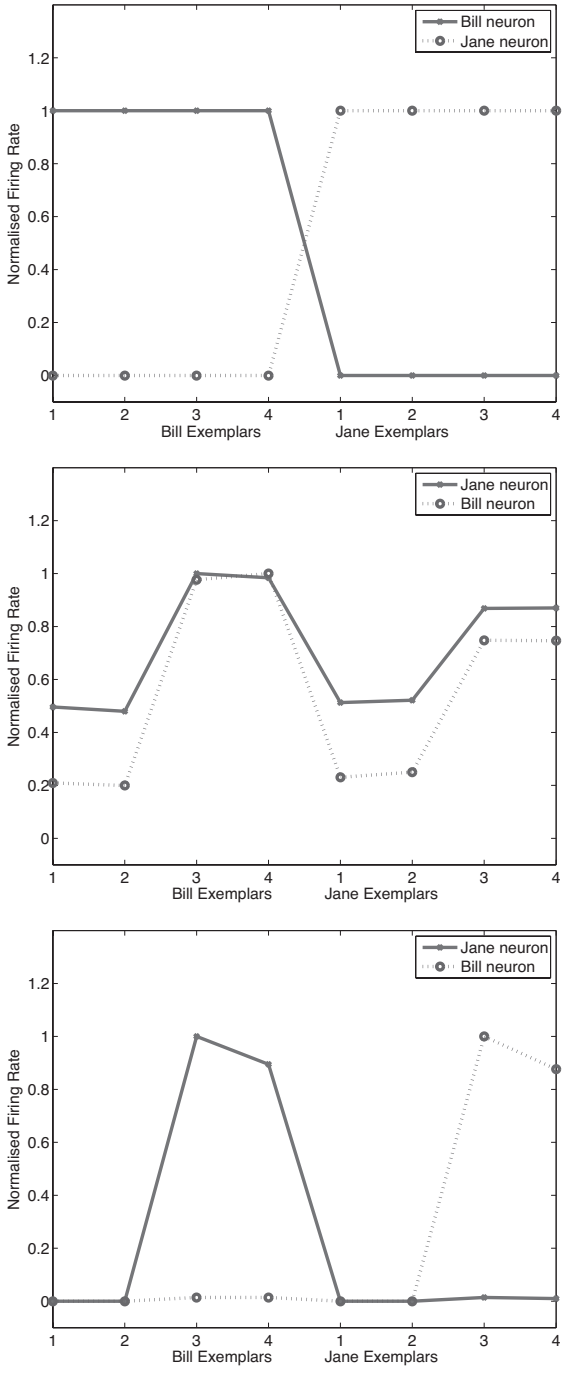


Figure 3.14: Top. View invariant representations of cups. Single cells in the output representation of VisNet. The two neurons illustrated responded either to all views of one cup (labelled 'Bill') and to no views of the other cup (labelled 'Jane'), or vice versa. Middle. Single cells in the C2 representation of HMAX. Bottom. Single cells in the View Tuned Unit output representation of HMAX.

Discussion

Overview of the findings on how well the properties of inferior temporal cortex neurons were met, and discussion of their significance

At the beginning of this chapter, some of the key properties of inferior temporal cortex (IT) neurons were listed that need to be addressed by models of invariant visual object recognition in the ventral visual stream of the cerebral cortex. This section will now assess to what extent these two models account for these fundamental properties of IT neurons. This assessment is provided for these two models provided as examples, and to illustrate how it may be useful for those who work with other models (e.g. Yamins et al. (2014)) to assess the performance of their models against the neurophysiological data. These comparisons are made in the interest of contributing to the further development of models of how the brain solves invariant visual object recognition.

The first property is that inferior temporal visual cortex neurons show responses to objects that are typically translation, size, contrast, rotation, and in a proportion of cases view invariant, that is, they show transform invariance (Booth and Rolls 1998; Hasselmo, Rolls, Baylis, and Nalwa 1989; Logothetis, Pauls, and Poggio 1995; Rolls 2012b; Tovee, Rolls, and Azzopardi 1994). Experiment 4 with the different views of different cups shows that VisNet can solve view invariant object recognition, and that HMAX does not. VisNet solves this object recognition problem because it has a learning mechanism involving associations across time to learn that quite different views may be of the same object. HMAX has no such learning mechanism, and indeed its performance on this task was dominated by the presence or absence of low-level image features such as whether text was visible or not. The remark might be made that HMAX is not intended to display view invariant object recognition. But that is perhaps an important point: by having no such mechanism, HMAX does not account for a key feature of the tuning of many neurons in the inferior temporal visual cortex. In fact, VisNet uses temporo-spatial continuity to learn about all the different types of invariance, and thus provides a generic approach to producing invariant representations.

The second property is that inferior temporal cortex neurons show sparse distributed representations, in which individual neurons have high firing rates to a few stimuli (e.g. objects or faces) and lower firing rates to more stimuli, in which much information can be read from the responses of a single neuron from its firing rates using for example dot product decoding (because the neuronal responses are high to relatively few stimuli), and in which neurons encode independent information about a set of stimuli, as least up to tens of neurons (Abbott, Rolls, and Tovee 1996; Baddeley, Abbott, Booth, Sengpiel, Freeman, Wakeman, and Rolls 1997; Rolls 2008a, 2012b; Rolls and Tovee 1995; Rolls and Treves 2011; Rolls, Treves, Tovee, and Panzeri 1997a; Rolls, Treves, and Tovee 1997b). Experiment 2 shows that VisNet produces single neurons with sparse representations, in which a single neuron can respond to many exemplars of one object, and to no exemplars of many other objects (Figs. 3.8 and 3.9). (Although these representations are relatively binary, with the most selective neurons for an object having high firing rates to only one object and low firing rates to all other objects, other neurons that are less selective have more graded firing rate representations, and the representations can be made more graded by reducing the value of the sigmoid activation function parameter β specified in Eqn 3.2) In contrast HMAX produces neurons in its final C layer that have highly distributed representations, with even the most selective single neurons having high firing rates to almost all the stimuli (see Figs. 3.8 and 3.9) for the C2 top layer neurons with the Mutch and Lowe (2008) implementation of HMAX. Consistent with this, the information

could not be read from these final C layers of HMAX by a biologically plausible pattern association network using dot product decoding, but required a much more powerful support vector machine or linear least squares regressor equivalent to a delta-rule perceptron to classify these outputs of HMAX. If View Tuned Units were used to read the outputs of HMAX, then these units did have a more sparse representation, but again had some responses to all the exemplars of all objects as shown in the same Figures, and as noted for the first property, did not have view invariant representations and so again required powerful decoding to read the VTUs to classify an image as a particular object. That is, most of the work for the classification was done by the external system reading the activity of the output neurons, rather than being present in the firing of the HMAX neurons themselves. Similar conclusions about the representations produced by HMAX follow from Experiment 1 with the CalTech stimuli, though as noted below under property 4, the use of such datasets and classification into a class of object such as animal vs non-animal does not capture the fundamental property 4 of encoding information about individual faces or objects, as contrasted with classes.

A third property is that inferior temporal cortex neurons often respond to objects and not to low-level features, in that many respond to whole objects, but not to the parts presented individually, nor to the parts presented with a scrambled configuration (Perrett, Rolls, and Caan 1982; Rolls, Tovee, Purcell, Stewart, and Azzopardi 1994). Experiment 3 showed that rearranging the parts of an object ('scrambling') led to no responses from VisNet neurons that responded to the whole object, showing that it implemented whole object recognition, rather than just having responses to features. This follows up with images of objects what was shown previously for VisNet with more abstract stimuli, combinations of up to four lines in different spatial arrangements to specify different shapes (Elliffe, Rolls, and Stringer 2002). In contrast, HMAX final layer neurons responded also to the scrambled images, providing an indication that HMAX does not implement shape representations of whole objects in which the parts are in the correct spatial arrangement, but instead allows features to pass from its Gabor filters to the output on the basis of which a powerful classifier might be able to specify because of the types of low level features what class of object may be present. This may be satisfactory for low-level feature identification that might then be used to classify objects into classes using e.g. a SVM, but is hardly a basis for shape recognition, which is a key property of IT neurons. VisNet solves the shape problem by implementing a feature hierarchy in which combinations of features in the correct spatial relationship become learned at every stage of the processing, resulting in shape not low level feature recognition (Elliffe, Rolls, and Stringer 2002; Rolls 1992, 2008a, 2012b).

A fourth property is that inferior temporal cortex neurons convey information about the individual object or face, not just about a class such as face vs non-face, or animal vs non-animal (Abbott, Rolls, and Tovee 1996; Baddeley, Abbott, Booth, Sengpiel, Freeman, Wakeman, and Rolls 1997; Rolls 2008a, 2011, 2012b; Rolls and Tovee 1995; Rolls and Treves 2011; Rolls, Treves, Tovee, and Panzeri 1997a; Rolls, Treves, and Tovee 1997b). This key property is essential for recognising a particular person or object, and is frequently not addressed in models of invariant object recognition, which still focus on classification into e.g. animal vs non-animal, or classes such as hats and bears from databases such as the CalTech (Mutch and Lowe 2008; Serre et al. 2007a,b,c; Yamins et al. 2014). It is clear that VisNet has this key property of representing individual objects, faces, etc., as is illustrated in Experiments 2, 3 and 4, and previously (Perry, Rolls, and Stringer 2006, 2010; Rolls 2012b; Rolls and Milward 2000; Rolls and Webb 2014; Stringer and Rolls 2000, 2002; Stringer, Perry, Rolls, and Proske 2006; Webb and Rolls 2014). VisNet achieves this by virtue of

its competitive learning, in combination with its trace learning rule to learn that different images are of the same specific object. It is unfortunate that we know little about this from previous publications with HMAX, but the results shown in Experiment 4 provide evidence that HMAX may categorise together images with similar low level feature properties (such as the presence of text), and not perform shape recognition relevant to the identification of an individual in which the spatial arrangements of the parts is important, as shown in Experiment 3.

A fifth property is that the learning mechanism involved in invariant object recognition needs to be biologically plausible, and that is likely to include a local synaptic learning rule (Rolls 2008a). This is implemented in VisNet, in that the information present to alter the strength of the synapse is present in the firing of the pre-synaptic and post-synaptic neuron, as is shown in Eqn. 3.4. We note that lateral propagation of weights, as used in the neocognitron Fukushima (1980), HMAX (Mutch and Lowe 2008; Riesenhuber and Poggio 1999a; Serre et al. 2007b), and convolution networks (LeCun et al. 2010), is not biologically plausible (Rolls 2008a).

The training method

One difference highlighted by these investigations is that VisNet is normally trained on images generated by objects as they transform in the world, so that view, translation, size, rotation etc invariant representations of objects can be learned by the network. In contrast, HMAX is typically trained with large databases of pictures of different exemplars of for example hats and beer-mugs as in the CalTech databases, which do not provide the basis for invariant representations of objects to be learned, but are aimed at object classification. However, it is shown in Experiment 1 that VisNet can perform this object classification in a way that is comparable to HMAX. In Experiment 1 it is also shown that the activity of the output of the last layer of HMAX C neurons is very non-sparse, provides very little information in the single neuron responses about the object class, cannot be read by biologically plausible decoding such as might be performed in the brain by a pattern association network, and requires a support vector machine (or View Tuned Neurons followed by least-squares learning) to learn the classification. In contrast, because of the invariance learning in VisNet, the single neurons in the sparse representation at the output provide information about which class of object was shown, and the population can be read by a biologically plausible pattern association network. VisNet thus provides a representation similar to that of neurons in the inferior temporal visual cortex (Rolls 2012b; Rolls and Treves 2011), and HMAX does not produce representations that are like those found in the inferior temporal visual cortex.

In Experiment 2 it is shown that VisNet performs well with sets of images (from the ALOI set) that provide exemplars that allow view invariant representations to be formed. HMAX performs poorly at this type of task when assessed with biologically plausible measures, in that the C2 neurons discriminate poorly between the classes, and in that the VTU neurons generalise only to adjacent views, as there is no mechanism in HMAX to enable it to learn that quite different images may be different views of the same object. HMAX thus has to rely on powerful pattern classification mechanisms such as a support vector machine to make sense of its output representation. The difference of the output representations is also marked. Single neurons in VisNet provide considerable stimulus-specific information about which object was shown (e.g. 3 bits depending on the set size, with the maximum information being $\log_2 S$ where S is the number of objects in the set), in a way that is similar to that of neurons in the inferior temporal visual cortex (Rolls et al. 1997a; Tovee et al. 1993). In contrast, individual neurons in the HMAX C2 layer are poorly tuned to

the stimuli, and contain little stimulus-specific information about views let alone about objects. The same point applies to many other computer-based object recognition systems, including deep convolutional neural networks, namely that they have no way of learning transform invariances from systematically transformed sets of training exemplars of objects.

Representations of the spatial configurations of the parts

In Experiment 3 it is shown that VisNet neurons do not respond to scrambled images of faces, providing evidence that they respond to the shape information about faces, and objects, and not to low-level features or parts. In contrast, HMAX neurons responded with similarly high rates to both the unscrambled and scrambled faces, indicating that low level features including texture may be very relevant to the performance and classification produced by HMAX.

Object representations invariant with respect to catastrophic view transformations

In Experiment 4, it is shown that VisNet can learn to recognise objects even when the view provided by the object changes catastrophically as it transforms, whereas HMAX has no learning mechanism in its S-C hierarchy that provides for this type of view-invariant learning.

Thus the approach taken by VisNet provides a model of ventral visual stream processing that produces neurons at its output layer that are very similar in their invariant representations to those found in the inferior temporal visual and that can be read by pattern association networks in brain regions such as the orbitofrontal cortex and amygdala. In contrast, the approach taken in HMAX does not lead to neurons in the output C layer that provide view invariant representations of objects, are very non-sparse and unlike those found in visual cortical areas, and needs the major part of any object classification required to be performed by an artificial neural network such as a support vector machine. These investigations of different approaches to how the ventral visual stream can produce firing like that of neurons in the inferior temporal visual cortex that can be easily read by biologically plausible networks such as pattern associators have implications for future research, and provide interesting contrasts of approaches used in biologically plausible object recognition networks with transform-invariant representations of objects and artificial neural networks required to perform pattern classification. Our main aim here of comparing these two networks is that the comparison helps to highlight what a biologically plausible model of the ventral visual system in invariant visual object recognition needs to account for.

How VisNet solves the computational problems of view invariant representations

We provide now an account of how VisNet is able to solve the type of invariant object recognition problem described here when an image is presented to it, with more detailed accounts available elsewhere (Rolls 2008a, 2012b; Wallis and Rolls 1997a). VisNet is a 4-layer network with feedforward convergence from stage to stage that enables the small receptive fields present in its V1-like Gabor filter inputs of approximately 1 degree to increase in size so that by the fourth layer a single neuron can potentially receive input from all parts of the input space (Fig. 3.1). The feedforward connections between layers are trained by competitive learning, which is an unsupervised form of learning (Rolls 2008a), that allows neurons to learn to respond to feature combinations. As

one proceeds up though the hierarchy, the feature combinations become combinations of feature combinations (see Rolls (2008a) Fig. 4.20 and Elliffe et al. (2002)). Local lateral inhibition within each layer allows each local area within a layer to respond to and learn whatever is present in that local region independently of how much information and contrast there may be in other parts of a layer, and this, together with the non-linear activation function of the neurons, enables a sparse distributed representation to be produced. In the sparse distributed representation, a small proportion of neurons is active at a high rate for the input being presented, and most of the neurons are close to their spontaneous rate, and this makes the neurons of VisNet (Rolls 2008a, 2012b) very similar to those recorded in the visual system (Abbott et al. 1996; Rolls 2008a; Rolls and Treves 2011; Rolls et al. 1997a,b; Tovee et al. 1993). A key property of VisNet is the way that it learns whatever can be learned at every stage of the network that is invariant as an image transforms in the natural world, using the temporal trace learning rule. This learning rule enables the firing from the preceding few items to be maintained, and given the temporal statistics of visual inputs, these inputs are likely to be from the same object. (Typically primates including humans look at one object for a short period during which it may transform by translation, size, isomorphic rotation, and/or view, and all these types of transform can therefore be learned by VisNet.) Effectively, VisNet uses as a teacher the temporal and spatial continuity of objects as they transform in the world to learn invariant representations. (An interesting example is that representations of individual people or objects invariant with respect to pose (e.g. standing, sitting, walking) can be learned by VisNet, or representations of pose invariant with respect to the individual person or object can be learned by VisNet depending on the order in which the identical images are presented during training (Webb and Rolls 2014).) Indeed, we developed these hypotheses (Rolls 1992, 1995, 2012b; Wallis, Rolls, and Földiák 1993) into a model of the ventral visual system that can account for translation, size, view, lighting, and rotation invariance (Elliffe et al. 2002; Perry et al. 2006, 2010; Rolls 2008a, 2012b; Rolls and Milward 2000; Rolls and Stringer 2001, 2006, 2007; Stringer and Rolls 2000, 2002, 2008; Stringer et al. 2006, 2007; Wallis and Rolls 1997a). Consistent with the hypothesis, we have demonstrated these types of invariance (and spatial frequency invariance) in the responses of neurons in the macaque inferior temporal visual cortex (Booth and Rolls 1998; Hasselmo et al. 1989; Rolls and Baylis 1986; Rolls et al. 1985, 1987, 2003; Tovee et al. 1994). Moreover, we have tested the hypothesis by placing small 3D objects in the macaque's home environment, and showing that in the absence of any specific rewards being delivered, this type of visual experience in which objects can be seen from different views as they transform continuously in time to reveal different views leads to single neurons in the inferior temporal visual cortex that respond to individual objects from any one of several different views, demonstrating the development of view-invariance learning (Booth and Rolls 1998). (In control experiments, view invariant representations were not found for objects that had not been viewed in this way.) The learning shown by neurons in the inferior temporal visual cortex can take just a small number of trials (Rolls et al. 1989). The finding that temporal contiguity in the absence of reward is sufficient to lead to view invariant object representations in the inferior temporal visual cortex has been confirmed (Li and DiCarlo 2008, 2010, 2012). The importance of temporal continuity in learning invariant representations has also been demonstrated in human psychophysics experiments (Perry, Rolls, and Stringer 2006; Wallis 2013). Some other simulation models are also adopting the use of temporal continuity as a guiding principle for developing invariant representations by learning (Einhauser et al. 2005; Franzius et al. 2007; Wiskott 2003; Wiskott and Sejnowski 2002; Wyss et al. 2006) (see review by Rolls (2012b)), and the temporal trace learning principle has also been applied recently Isik et al. (2012) to HMAX (Riesenhuber and Poggio 2000a; Serre et al. 2007c)

and to V1 (Lies et al. 2014).

VisNet is also well adapted to deal with real-world object recognition. If different backgrounds are present during testing, this does not disrupt the identification of particular objects previously trained, because the different backgrounds are not associated with the object to be recognised. This process is helped by the fact that the responses of inferior temporal cortex neurons shrink from approximately 78° in diameter in a scene with one object on a blank background, to approximately 22° in a complex natural scene (Rolls, Aggelopoulos, and Zheng 2003). This greatly facilitates processing in the ventral visual cortical stream object recognition system, for it means that it is much more likely that there is only one object or a few objects to be dealt with at the fovea that need to be recognised (Rolls, Aggelopoulos, and Zheng 2003). The mechanism for the shrinking of the receptive fields of inferior temporal cortex neurons in complex natural scenes is probably lateral inhibition from nearby visual features and objects, which effectively leave a neuron sensitive to only the peak of the receptive field, which typically includes the fovea because of its greater cortical magnification factor for inferior temporal cortex neurons (Trappenberg, Rolls, and Stringer 2002). Moreover, for similar reasons, VisNet can learn to recognise individual objects if they presented simultaneously with other objects chosen randomly (Stringer and Rolls 2008; Stringer, Rolls, and Tromans 2007).

The approach taken by HMAX

We now compare this VisNet approach to invariant object recognition to the approach of HMAX, another approach that seeks to be biologically plausible (Mutch and Lowe 2008; Riesenhuber and Poggio 2000a; Serre, Kreiman, Kouh, Cadieu, Knoblich, and Poggio 2007a; Serre, Oliva, and Poggio 2007b; Serre, Wolf, Bileschi, Riesenhuber, and Poggio 2007c), which is a hierarchical feedforward network with alternating simple cell-like (S) and complex cell-like (C) layers inspired by the architecture of the primary visual cortex, V1. The simple cell-like layers respond to a similarity function of the firing rates of the input neuron to the synaptic weights of the receiving neuron (used as an alternative to the more usual dot product), and the complex cells to the maximum input that they receive from a particular class of simple cell in the preceding layer. The classes of simple cell are set to respond maximally to a random patch of a training image (by presenting the image, and setting the synaptic weights of the S cells to be the firing rates of the cells from it receives), and are propagated laterally, that is there are exact copies throughout a layer, which is of course a non-local operation and not biologically plausible. The hierarchy receives inputs from Gabor-like filters (which is like VisNet). The result of this in HMAX is that in the hierarchy there is no learning of invariant representations of objects; and that the output firing in the final C layer (for example the second C layer in a four-layer S1-C1-S2-C2 hierarchy) is high for almost all neurons to most stimuli, with almost no invariance represented in the output layer of the hierarchy, in that two different views of the same object may be as different as a view of another object, measured using the responses of a single neuron or of all the neurons. The neurons in the output C layer are thus quite unlike those in VisNet or in the inferior temporal cortex, where there is a sparse distributed representation, and where single cells convey much information in their firing rates, and populations of single cells convey much information that can be decoded by biologically plausible dot product decoding (Abbott et al. 1996; Rolls 2008a; Rolls and Treves 2011; Rolls et al. 1997a,b; Tovee et al. 1993) such as might be performed by a pattern association network in the areas that receive from the inferior temporal visual cortex, such as the orbitofrontal cortex and amygdala (Rolls 2008a, 2012b, 2014; Rolls and Treves 2011). HMAX therefore must

resort to a powerful classification algorithm, in practice typically a Support Vector Machine (SVM), which is not biologically plausible, to learn to classify all the outputs of the final layer that are produced by the different transforms of one object to be of the same object, and different to those of other objects. Thus HMAX does not learn invariant representations by its output layer of the S–C hierarchy, but instead uses a SVM to perform the classification that the SVM is taught. This is completely unlike the output of VisNet and of inferior temporal cortex neuron firing, which by responding very similarly in terms of firing rate to the different transforms of an object show that the invariance has been learned in the hierarchy (Booth and Rolls 1998; Hasselmo et al. 1989; Rolls 2008a, 2012b).

Another way that the output of HMAX may be assessed is by the use of View-Tuned Units (VTUs), each of which is set to respond to one view of a class or object by setting its synaptic weights from each C unit to the value of the firing of the C unit to one view or exemplar of the object or class (Serre, Oliva, and Poggio 2007b). We note that this itself is not a biologically plausible operation, for it implies a teacher for each VTU to inform it how to respond, and then adjustment of the synaptic weights to the VTU to achieve this. Because there is little invariance in the C units, many different VTUs are needed, with one for each training view or exemplar. Because the VTUs are different to each other for the different views of the same object or class, a further stage of training is then needed to classify the VTUs into object classes, and the type of learning is least squares error minimization (Serre et al. 2007b), equivalent to a delta-rule one-layer perceptron which again is not biologically plausible for neocortex (Rolls 2008a). Thus HMAX does not generate invariant representations in its S–C hierarchy, and in the VTU approach uses two layers of learning after the S–C hierarchy, the second involving least squares learning, to produce classification. This is unlike VisNet, which learns invariant representations in its hierarchy by self-organization, and produces view invariant neurons (similar to those for faces (Hasselmo et al. 1989) and objects (Booth and Rolls 1998) in the inferior temporal visual cortex) that can be read by a biologically plausible pattern associator (Rolls 2008a, 2012b). In another approach, Biederman and colleagues have shown that HMAX does not show the advantage in psychophysical performance and in the activations of area LO that is related to viewpoint invariant or non-accidental properties (e.g., straight vs. curved), than metric properties (e.g., degree of curvature) of simple shapes.

Another difference of HMAX from VisNet is in the way that VisNet is trained, which is a fundamental aspect of the VisNet approach. HMAX has traditionally been tested with benchmarking databases such as the CalTech-101 and CalTech-256 (Griffin et al. 2007) in which sets of images from different categories are to be classified. The Caltech-256 dataset is comprised of 256 object classes made up of images that have many aspect ratios, sizes and differ quite significantly in quality (having being manually collated from web searches). The objects within the images show significant intra-class variation and have a variety of poses, illumination, scale and occlusion as expected from natural images. A network is supposed to classify these correctly into classes such as hats and beer mugs (Rolls 2012b). The problem is that examples of each class of object transforming continuously though different positions on the retina, size, isomorphic rotation, and view are not provided to help the system learn about how a given type of object transforms in the world. The system just has to try to classify based on a set of often quite different exemplars that are not transforms of each other. Thus a system trained in this way is greatly hindered in generating transform invariant representations by the end of the hierarchy, and such a system has to rely on a powerful classifier such as a SVM to perform a classification that is not based on transform invariance learned in the hierarchical network. In contrast, VisNet is provided during

training with systematic transforms of objects of the type that would be seen as objects transform in the world, and has a well-posed basis for learning invariant representations. It is important that with VisNet, the early layers may learn what types of transform can be produced in small parts of the visual field by different classes of object, so that when a new class of object is introduced, rapid learning in the last layer and generalization to untrained views can occur without the need for further training of the early layers (Stringer and Rolls 2002).

Some other approaches to invariant visual object recognition

Some other approaches to biologically plausible invariant object recognition are being developed with hierarchies that may be allowed unsupervised learning (DiCarlo et al. 2012; Pinto et al. 2009; Yamins et al. 2014). For example, a hierarchical network has been trained with unsupervised learning, and with many transforms of each object to help the system to learn invariant representations in an analogous way to that in which VisNet is trained, but the details of the network architecture are selected by finding parameter values for the specification of the network structure that produce good results on a benchmark classification task (Pinto et al. 2009). However, formally these are convolutional networks, so that the neuronal filters for one local region are replicated over the whole of visual space, which is computationally efficient but biologically implausible. Further, a general linear model is used to decode the firing in the output level of the model to assess performance, so it is not clear whether the firing rate representations of objects in the output layer of the model are very similar to that of the inferior temporal visual cortex. In contrast, with VisNet (Rolls 2012b; Rolls and Milward 2000) the information measurement procedures that we use (Rolls et al. 1997a,b) are the same as those used to measure the representation that is present in the inferior temporal visual cortex (Abbott, Rolls, and Tovee 1996; Aggelopoulos, Franco, and Rolls 2005; Baddeley, Abbott, Booth, Sengpiel, Freeman, Wakeman, and Rolls 1997; Franco, Rolls, Aggelopoulos, and Treves 2004; Franco, Rolls, Aggelopoulos, and Jerez 2007; Panzeri, Treves, Schultz, and Rolls 1999; Rolls and Tovee 1995; Rolls and Treves 2011; Rolls, Treves, Tovee, and Panzeri 1997a; Rolls, Treves, and Tovee 1997b; Rolls, Aggelopoulos, Franco, and Treves 2004; Rolls, Franco, Aggelopoulos, and Jerez 2006; Tovee and Rolls 1995; Tovee, Rolls, Treves, and Bellis 1993; Treves, Panzeri, Rolls, Booth, and Wakeman 1999).

The properties of inferior temporal cortex neurons that need to be addressed by models in visual invariant object recognition

One of the important points made here is that there are a number of crucial properties of inferior temporal cortex (IT) neurons that need to be accounted for by biologically plausible models. These properties include the sparse distributed coding in which individual neurons have high firing rates to a few objects, and gradually smaller responses to other stimuli. This allows much information to be read from the responses of a single neuron, or from several neurons with the information represented approximately independently for at least a limited number of neurons (Abbott, Rolls, and Tovee 1996; Rolls 2012b; Rolls and Tovee 1995; Rolls and Treves 2011; Rolls, Treves, Tovee, and Panzeri 1997a; Rolls, Treves, and Tovee 1997b; Tovee, Rolls, Treves, and Bellis 1993; Treves, Panzeri, Rolls, Booth, and Wakeman 1999). This is a general property of cortical encoding, and is important in the operation of associative neural networks that receive from structures such as the inferior temporal visual cortex (Rolls 2008a, 2016; Rolls and Treves 2011). This is shown here to be a property of VisNet, but not of HMAX. Another property is that some IT neurons respond

to parts of objects, and some only to the whole object (Perrett, Rolls, and Caan 1982). The latter was shown here to be a property of VisNet but not HMAX. Another property is view invariance, shown by some but not all neurons in IT (Booth and Rolls 1998; Hasselmo, Rolls, Baylis, and Nalwa 1989), which was shown to be a property of VisNet but not HMAX. Indeed, much more transform invariance than this must be shown by a model to account for the properties of IT neurons, including translation invariance (with 70° receptive fields shrinking to approximately 8° in complex scenes), size, contrast, and spatial frequency invariance, all properties of VisNet (Aggelopoulos and Rolls 2005; Booth and Rolls 1998; Logothetis, Pauls, and Poggio 1995; Rolls 2012b; Rolls and Baylis 1986; Rolls, Baylis, and Leonard 1985; Rolls, Baylis, and Hasselmo 1987; Rolls, Aggelopoulos, and Zheng 2003; Tovee, Rolls, and Azzopardi 1994; Trappenberg, Rolls, and Stringer 2002). An implication is that there is very much more to testing and assessing a good model of IT performance than measuring the similarity structure of the representations of images of objects, human faces, animal faces, body parts, etc produced by different including non-biologically plausible approaches to object recognition including deep neural networks (Cadieu et al. 2013, 2014; Khaligh-Razavi and Kriegeskorte 2014). Indeed these measures of similarity are likely to benefit from supervised training, as has been found (Khaligh-Razavi and Kriegeskorte 2014), whereas the similarity structure of models such as VisNet that utilize a temporal trace rule will depend on the exact similarity structure of the input across time, which needs to be taken into account in such assessments. Moreover, analyzing the similarity structure of model and IT representations for classes of object does not address fundamental issues of IT encoding, that IT neurons convey much information about which particular face is being shown (Abbott, Rolls, and Tovee 1996; Rolls 2012b; Rolls and Tovee 1995; Rolls and Treves 2011; Rolls, Treves, Tovee, and Panzeri 1997a; Rolls, Treves, and Tovee 1997b; Tovee, Rolls, Treves, and Bellis 1993; Treves, Panzeri, Rolls, Booth, and Wakeman 1999) not just about whether it is a human or animal face or another category (Cadieu et al. 2013, 2014; Khaligh-Razavi and Kriegeskorte 2014). The present research thus emphasizes that there are a number of key properties of IT neurons that need to be taken into account in assessing how well a model accounts for the properties of IT neurons.

Comparison with computer vision approaches to not only classification of objects but also to identification of the individual

We turn next to compare the operation of VisNet, as a model of cerebral cortical mechanisms involved in view-invariant object identification, with artificial, computer vision, approaches to object identification. However, we do emphasize that our aim in the present research is to investigate how the cerebral cortex operates in vision, not how computer vision attempts to solve similar problems. Within computer vision, we note that many approaches start with using independent component analysis (ICA) (Kanan 2013), principal component analysis (PCA) (Cottrell and Hsaio 2011), sparse coding (Kanan and Cottrell 2010), and other mathematical approaches (Larochelle and Hinton 2010) to derive what may be suitable 'feature analyzers', which are frequently compared to the responses of V1 neurons. Computer vision approaches to object identification then may take combinations of these feature analyzers, and perform statistical analyses using computer-based algorithms that are not biologically plausible such as Restricted Boltzmann Machines (RBMs) on these primitives to statistically discriminate different objects (Larochelle and Hinton 2010). Such a system does not learn view invariant object recognition, for the different views of an object may have completely different statistics of the visual primitives, yet are the different views of the same object. (Examples might include frontal and profile views of faces, which are well tolerated for in-

dividual recognition by some inferior temporal cortex neurons (Hasselmo, Rolls, Baylis, and Nalwa 1989); very different views of 3D object which are identified correctly as the same object by IT neurons after visual experience with the objects to allow for view-invariant learning (Booth and Rolls 1998); and many man-made tools and objects which may appear quite different in 2D image properties from different views.) Part of the difficulty of computer vision lay in attempts to parse a whole scene at one time (Marr 1982). However, the biological approach is to place the fovea on one part of a scene, perform image analysis / object identification there, and then move the eyes to fixate a different location in a scene (Rolls and Webb 2014; Rolls, Aggelopoulos, and Zheng 2003; Trappenberg, Rolls, and Stringer 2002). This is a divide-and-conquer strategy used by the real visual system, to simplify the computational problem into smaller parts performed successively, to simplify the representation of multiple objects in a scene, and to facilitate passing the coordinates of a target object for action by using the coordinates of the object being fixated (Aggelopoulos and Rolls 2005; Ballard 1990; Rolls 2008a, 2012b; Rolls and Deco 2002; Rolls et al. 2003). This approach has now been adopted by some computer vision approaches (Denil et al. 2012; Kanan 2013; Kanan and Cottrell 2010). We note that non-biologically plausible approaches to object vision are important in assessing how different types of system operate with large numbers of training and test images (Cadieu et al. 2014; Khaligh-Razavi and Kriegeskorte 2014), and that there are attempts to make multilayer error correction networks more biologically plausible (Balduzzi et al. 2014; O'Reilly and Munakata 2000), but that many of these systems are far from being biological plausible. Biologically plausible systems for object recognition need to have not only the properties described here, but also mechanisms that use a local learning rule, no separate teacher for each output neuron in a supervised learning scheme, and no lateral copying of weights (Rolls 2016). Moreover, understanding how the brain operates is important not only in its own right, but also for its implications for understanding disorders of brain function (Rolls 2008a, 2012a, 2016).

Outlook: some properties of inferior temporal cortex neurons that need to be addressed by models of ventral visual stream visual invariant object recognition

The analyses described in this chapter are intended to highlight some properties that models of visual object recognition in the brain in the ventral visual stream need to achieve if they are to provide an account of its functions in invariant visual object recognition, with the criteria being identified by the responses of neurons with transform invariant representations that are found in the inferior temporal visual cortex (Rolls 2008a, 2012b, 2016). First, the formation of single neurons with translation, view and rotation invariance needs to be accounted for. It is not sufficient to use a powerful decoder after the model network to achieve the required performance, instead of invariance being represented by the neurons themselves in the model of the ventral visual system. An important implication for future research is that the training set of stimuli needs to include different views of the same object, and not collections of images of objects in the same class. Indeed, an important distinction is that much of what is represented in the inferior temporal visual cortex is about the invariant representation of different objects, so that individual objects or faces can be recognised from different views (Booth and Rolls 1998; Hasselmo, Rolls, Baylis, and Nalwa 1989), rather than just knowing that the object is a face as in a classification task. Second, the neuronal representation should be in a sparse distributed form in which much information can be read from the responses of single neurons Rolls et al. (1997a). Third, the information should be represented approximately independently by different neurons, as least up to tens of neurons (Rolls et al. 1997b).

Fourth, the neuronal representation needs to be decodable by a biologically plausible network such as a pattern association network that uses dot product decoding, which is biologically plausible for neurons (Rolls 2008a; Rolls and Treves 2011; Rolls et al. 1997b). The reason why the representation is in this form in the inferior temporal visual cortex is, it is postulated, because the inferior temporal visual cortex projects directly to brain areas such as the orbitofrontal cortex and amygdala that perform pattern associations of these representations with for example reinforcers such as taste and touch (Rolls 2008a, 2014). Fifth, the network of ventral visual stream processing needs to implement learning, for different views of an object may look very different, yet single neurons do respond to these different views (Booth and Rolls 1998; Hasselmo, Rolls, Baylis, and Nalwa 1989), as is required for the appropriate associations to be output by the next stages of pattern association processing (Rolls 2008a, 2014). This paper has highlighted these properties. Further properties include how top-down selective attention can usefully bias the object recognition system (with a model of how this has been implemented for VisNet described previously by Deco and Rolls (2004)); how cortico-cortical backprojections implement recall (with models described previously (Kesner and Rolls 2015; Rolls 1989, 2008a, 2015; Treves and Rolls 1994)) (and this has implications for other possible functions that might be proposed in models of vision for backprojections (Rolls 2008a, 2016)); and how different systems scale up to deal with large numbers of objects.

Conclusions

In conclusion, in this chapter we have compared for the first time two leading approaches to object identification in the ventral visual system. We have shown how producing biologically plausible representations, that are similar to those of primate inferior temporal cortex neurons, is an important criterion for whether a model is successful as a model of the process undertaken in this cortical region. By this criterion, VisNet is biologically plausible, and HMAX is not (Experiment 1). The findings have important implications for future research, for this criterion will be important to bear in mind in developing models and theories of how the ventral visual system operates in invariant visual object recognition in future. Moreover, it is important to emphasise that neurons in the inferior temporal visual cortex provide representations suitable for the identification of individual objects, such as the face of a single individual seen from different views, and not just classification of objects as hats, beer-mugs, umbrellas, etc. We have also shown (Experiment 2) that there are advantages to training with training sets that provide the information for view invariant representations of objects to be learned, rather than trying to perform classification of images as certain types of object just by seeing random exemplars of the objects in random views, which invites pattern classification based on features relevant to a class, instead of facilitating invariant representations of objects to be learned. The latter, as implemented in VisNet, provides a foundation for objects to be recognised correctly when they are shown in what can be quite different views, which is a property reflected by the responses of some neurons in the primate ventral visual pathways, in regions that include the inferior temporal visual cortex Rolls (2008a, 2012b). Another important implication is that a theory and model of the ventral visual system must be able to account for object shape recognition, not just recognition based on features or parts, as tested by scrambling the parts of objects (Experiment 3). Finally, in Experiment 4 we showed that some objects that undergo catastrophic feature changes as they transform into different views cannot be correctly categorised by systems that depend on features in an image, such as HMAX, but can be correctly identified by systems such as VisNet that can learn associations across time as objects transform naturally in time by using a synaptic learning rule with a short-term temporal

trace. These findings and the conceptual points that we make have clear implications for what needs to be solved by future models of invariant visual object recognition in the ventral cortical visual stream. Moreover, the research described has clear implications for ways in which these computational problems may be solved in the ventral visual stream cortical areas.

Machine Learning Approaches

Confusing Convolutional Networks

Introduction

Deep convolutional neural networks are currently at the forefront of image classification tasks, reporting many state-of-the-art results, and indeed near-human performance on large scale natural image benchmarks in recent years (He et al. 2015; Krizhevsky et al. 2012; Schroff et al. 2015; Szegedy et al. 2014). One long-standing assumption was that since these networks exhibit strong generalisation qualities to unseen test images they should also be stable with regards to small perturbations of the input image - since such an image undoubtedly still contains an exemplar of the same class. This notion of stability has in fact been shown to be false.

Work by Szegedy et al. (2014) was first to show that it is indeed possible to find small perturbations of a given image (via a L-BFGS optimisation process) that leave them visually indistinguishable from the original, but nonetheless cause the network to misclassify the previously correctly classified exemplar. Moreover these adversarial exemplars (the perturbed images) were shown to generalise well across different classifiers that had different architectures. A similar result for generated images has been shown by Nguyen et al. (2014). In this instance instead of starting with an exemplar, the optimisation proceeds either from a random noise image or simply directly generates a synthetic image with geometric patterns that are strongly classified as a particular class. A plausible explanation for the ease at which these networks can be fooled is given by Goodfellow et al. (2015). With the central problem being identified as the move towards using somewhat linear functions at each layer due to the ease at which they can be trained. This results in the representation space being partitioned into half-spaces of increasingly confident but erroneous class predictions as you move away from the distribution of the training exemplars.

In this chapter an extension of previous work by Goodfellow et al. (2015) is presented that shows that small perturbations of a correctly classified image can be constructed in an efficient way that allows for the original image to be re-assigned to any of the other classes while remaining visually indistinguishable. Moreover this process is robust with respect to both initial and target class. In Section 5.4 the implementation details of the relabelling process is discussed. In Section 4.3 the results of relabelling experiments done on the ImageNet dataset are presented and some of the discovered limitations of the relabelling process are discussed. Section 4.4 adds some analysis of how the relabelling manifests as changes to the representations of the intermediate layers of the network.

Implementation

To explore the possibility that an input image (that on a trained network is correctly classified) can be re-assigned to another label experiments were performed with the Caffe framework (Jia et al.

2014). The Caffe framework is particularly convenient for two reasons: Firstly, the framework has pre-trained implementations of "AlexNet" (Krizhevsky et al. 2012) and "GoogLeNet" (Szegedy et al. 2014) on Imagenet (Russakovsky et al. 2014). Secondly, straightforward modifications of the model definitions allow for derivatives of the cost function to be backpropagated right up to the input image layer.

The algorithm for re-assigning the label of a given image to that of any other proceeds as follows. Given a network model with parameters θ, an input image X, a "confusing" target label y and a cost function, $L(\theta, X, y)$. In the networks considered in this paper, L is the standard softmax function. When computing the set of gradients, $\frac{\partial \mathrm{L}}{\partial \theta}$, we backpropagate one step further to that of the input image X, yielding $\frac{\partial \mathrm{L}}{\partial \mathrm{X}}$. The process then proceeds like traditional gradient descent, with N gradient updates occurring to the input image, X as follows

$$X_{i+1} \leftarrow X_i - \alpha f(\frac{\partial \mathrm{L}}{\partial \mathrm{X_i}}) \tag{4.1}$$

The only remaining definition is the nature of the function, f in the above equation. Initial testing showed that while using the raw gradient allowed for a successful relabelling of the input image, it was a destructive process leaving the image with noticeable blemishes unless very careful tuning of the update step, α and the number of total iterations was undertaken. The problem with the raw gradient (as seen in Figure 4.1) is that there are strong "hot-spots" where the gradient values have a much larger magnitude than the surrounding average value. This causes local regions of the image to be greatly modified at each update step. As mentioned above this problem can be somewhat ameliorated by scaling the gradient very small (i.e small α) while greatly increasing the number of updates. However this is not ideal as it not only results in much longer computation times but also makes the process quite brittle - as a key constraint is that the resulting relabelled images remain visually indistinguishable from the originals. Following the example of Goodfellow et al. (2015) rescaling the gradient so that the relative strengths of the differing spatial locations is much less severe is beneficial. This was accomplished by setting the function f to be the signum which is then scaled by the quantisation factor of standard 8-bit images, specifically

$$f(X) = \mathrm{sgn}(X)/255.0 \tag{4.2}$$

Image relabelling

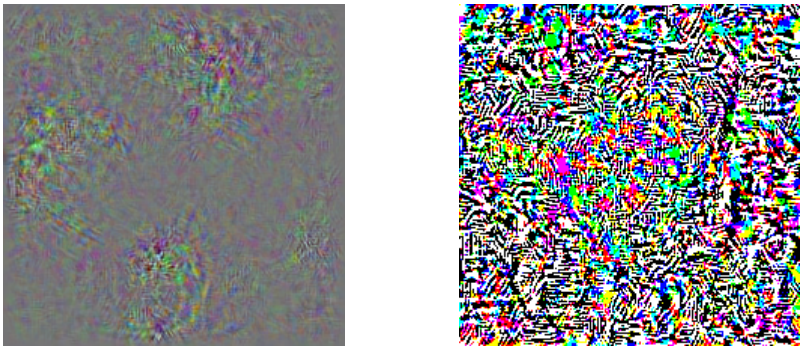
To perform the relabelling on a given image the optimization above is started with that initial image and the desired target class.

Similarly to Szegedy et al. (2014) the distortion measure that is used to defined as

$$distortion(x^{'}, x) = \sqrt{\frac{\sum_n (x_i^{'} - x_i)^2}{n}} \tag{4.3}$$

where x is the original image, $x^{'}$ the relabelled image and n is the number of pixels in the image (50176 when working with ImageNet images).

Figure 4.2 shows the results of the above image relabelling process on GoogLeNet, by choosing two exemplars at random from the ImageNet classes 'Shih-Tzu' and 'Half-Track' and swapping their labels. Average distortion measures from 30 such relabelling procedures was found to be 0.00814,



(a) $\frac{\partial L}{\partial X}$ (b) $\text{sgn}(\frac{\partial L}{\partial X})$

Figure 4.1: This figure shows the appearance of the gradients that are computed on the first iteration when relabelling an image from class 10, to class 348. Notice that the for the gradient passed through the sgn function (4.1(b)) the values are much more spread out spatially, and each colour channel are either -1/255, 0 or 1/255. This stops "hot spots" appearing in the image from the large clustered values apparent in 4.1(a) when used in the gradient descent.

which leaves the altered images visually indistinguishable from the originals. Figure 4.3 shows the associated class probabilities, confirming that the relabelling process has worked successfully. Furthermore the uncertainty of the relabelled image label is consistently very low when compared to that of true images of the class.

It should be noted that the classes were chosen to ensure that the initial experiments were on images that had significant qualitative features (consider the differences in texture, dominant features, shapes, etc exhibited by exemplars of these class) - so intuitively should be a 'hard' relabelling task and expressly not due to any specific pre-screening process.

How robust is this process?

To see how applicable this process is across the entire 1000 image classes of ImageNet, the following experiment was conducted. For each of the 1000 image classes 10 random exemplars were chosen and relabelled to each of the 999 other classes, with the caveat that the maximal distortion allowed for any relabelling attempt be capped at 0.01. This further restriction forces the relabelling to only be considered a success if the altered image is still visually indistinguishable and is particularly stringent in this regard as images that are corrupted with amounts of random noise resulting in distortion values magnitudes larger, while obviously altered, are still easily human-recognisable.

The results of this experiment were encouraging, with 98.7% of all class/target pairs were successful at being re-labelled below the 0.01 distortion threshold. In fact no class/target pairs were significantly harder to relabel at this distortion level. This implies that it would be perfectly possible to construct an augmented version of the ImageNet test set that would attain 0% accuracy, yet look visually indistinguishable from the original.

The only parameters in the relabelling process are the update stepsize, α and the number of iterations, N (which itself is a function of the maximal distortion you will accept since the $\text{sgn}(\frac{\partial L}{\partial X})$ is bounded at each step). In all the simulations described in this paper $\alpha = 50$ and the number of iterations were tuned accordingly if a target distortion was required, or the process was simply iterated until the relabelling process was successful. It is interesting to note that the gradient is

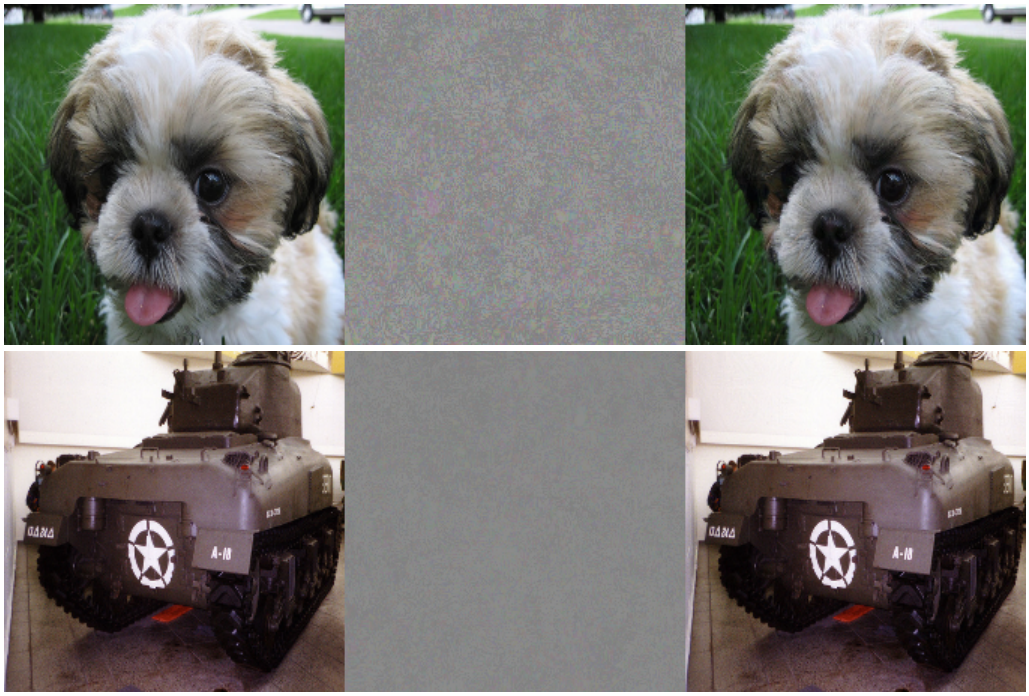


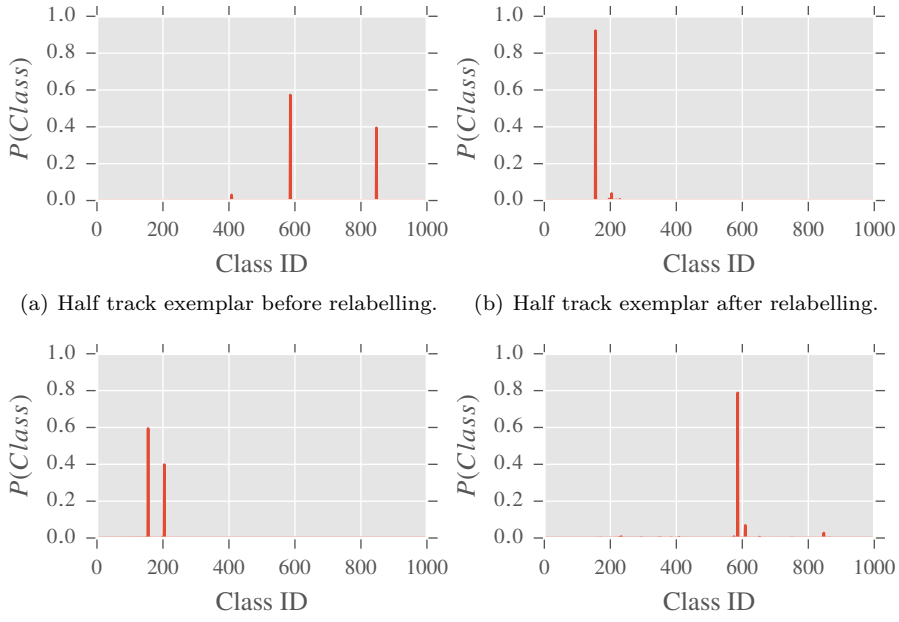
Figure 4.2: Examples of class relabelling on GoogLeNet. Left: Original images that are correctly classified as 'Shih-Tzu' and 'Half-track'. Right: relabelled images such that the classification output is reversed - i.e the 'Shih-Tzu' is now strongly classified as 'Half-track' and vice-versa. Centre: Pixel differences multiplied by 10 and scaled to mean-level for visibility. The distortion introduced in the top relableling was 0.00864 and the bottom 0.00712. These are a random pair of images taken from the above classes that were chosen to be qualitatively "maximally different". As we show in Section 4.3.1 relabelling is not sensitive to the initial or target class.

not *stationary*. Simply taking the first gradient and moving an equivalent distance in image space that would have arisen from many smaller steps does not successfully re-label the image (though can sometimes produce a mis-classification). This tends to suggest that there is not a simple linear relationship between the original image and the nearby perturbation that results in the new target label being acquired.

Unfortunately there are some limitations to this relabelling scheme. The changes to the image label are not robust to transformations of the image; cropping, translating and mirroring the image results in the label reverting to the correct one, which suggests that the perturbations are leveraging specific spatial location in the process of relabelling. Furthermore, while the relabelling process itself works with both GoogLeNet and AlexNet architectures (and probably many similarly architected networks), images that are relabelled on one do not transfer over to the other. All the experiments thus described have been carried out on the GoogLeNet architecture, but similar results were obtained with the AlexNet architecture and have been omitted for brevity.

Gaussian image synthesis

This relabelling process can also generate exemplars that are completely random images devoid of human-recognisable structure, that are nonetheless classified with a high confidence by simply



(a) Half track exemplar before relabelling.

(b) Half track exemplar after relabelling.

(c) ShihTzu exemplar before relabelling.

(d) ShihTzu exemplar after relabelling.

Figure 4.3: The class probabilities for each of the four images in Fig 4.2. Notice that after relabelling the uncertainty is also reduced.

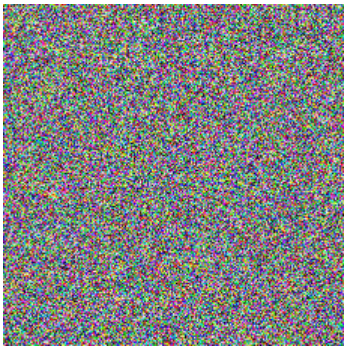
starting the process with a random image. Such a set of images can be shown in Figure 4.4. In fact to generate adversarial exemplars

Feature analysis

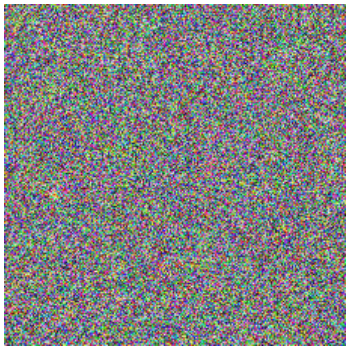
In an attempt to understand empirically what aspect of the network is failing we have done some cursory testing of the representations of the network

A natural question to ask might be, are there any differences in the feature space representation (i.e filter activations) between images that are relabelled and those that are "true" members of the target class? To try and investigate this at differing layers within GoogLeNet a SVM (with RBF kernel) was used on the raw features generated before the final fully connected layer. Specifically, 50 images from random classes were relabelled to a single target class. Twenty of these relabelled images, alongside 20 actual exemplars from the target class were passed through GoogLeNet and the intermediary layer activations were recorded. The features generated by these 40 images were used to train an SVM to see if the impostor relabelled images could be successfully identified from these feature representations alone. With 60 test features (30 of each type) the SVM was successfully able to identify the relabelled images with an average accuracy of 89.6% over 10 random sets of images and targets.

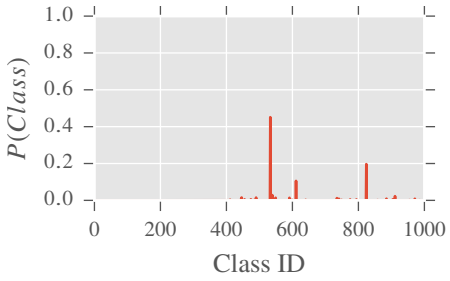
This is interesting because it supports the idea that the relabelled images are eliciting semantic "super responses" in the higher layer features. For example, if we were to relabel an image of a dog to that of a truck, then perhaps the best way to accomplish this is to trick the upper layers into detecting many, many wheel shaped objects - in fact many more than would be expected even in true truck images. In this way there should be quantitative differences in the representations produced at intermediary layers - even though both ultimately end up being classified as the same class.



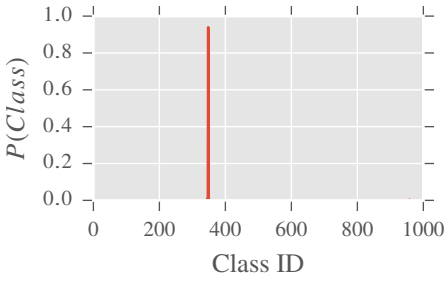
(a) Random gaussian image before relabelling.



(b) Random gaussian image after relabelling.



(c) The class probabilities of the image before relabelling.



(d) The class probabilities after relabelling.

Figure 4.4: These figures show a random gaussian image (4.4(a)) and the same image after being re-labeled to class 348 (4.4(b)). The bottom two graphs show the class probabilities before and after the relabelling process. The distortion introduced in this relabelling process was 0.03 - which is much higher than typically required to relabel natural images.

Discussion

This paper has described a method for confusing state-of-the-art deep convolutional neural networks, by perturbing images in such a way that they can be reassigned to any other class leaving the images visually indistinguishable. This has strong implications for what exactly the networks we have been training are actually learning from the data. The fact that such simple methods described here can fool these networks suggests that the information contained about image classes on even large datasets like ImageNet are insufficient to allow for comprehensive, continuous generalisation. Perhaps the current optimisation paradigms, network architectures or indeed both are fundamentally insufficient to resist these kinds of local instabilities.

While (Goodfellow et al. 2015) provide some evidence that adversarial exemplars themselves may be useful as a form of regularization to harden a system against the generation of such exemplars. It is unclear if a strategy along those lines will be sufficient given that the distribution of the adversarial exemplars dominate the feature space and don't exist in isolated pockets.

As a final aside it is interesting to note that while adversarial exemplars of the form in this chapter are highly unlikely to be perceptible in biological vision systems, the existence of numerous optical illusions and other perceptual tricks show the difficulty in creating a system that would be completely resistant to confusing exemplars of some sort.

Efficient Batchwise Dropout

Introduction

Dropout is a popular technique for regularizing artificial neural networks. Dropout networks are generally trained by minibatch gradient descent with a dropout mask turning off some of the units a different with a different pattern of dropout being applied to every sample in the minibatch (Hinton et al. 2012; Krizhevsky et al. 2012).

In this chapter, a simple alternative to the dropout mask is developed. Instead of masking dropped out units by setting them to zero, a matrix multiplication is performed using only a submatrix of the weight matrix. In this way unneeded hidden units are never calculated. Performing dropout *batchwise*, so that one pattern of dropout is used for each sample in a minibatch, substantially reduces training times. Batchwise dropout can be used with fully-connected and convolutional neural networks.

Independent dropout

Dropout is a technique to regularize artificial neural networks - it prevents overfitting (Srivastava et al. 2014). Experiments have shown that a fully connected network with two hidden layers of 80 units each can learn to classify the MNIST training set perfectly in as few as 20 training epochs. Unfortunately such a network grossly overfits, resulting in the test error being quite high, about 2%. Increasing the number of hidden units by a factor of 10 and using dropout to regularize the network results in a lower test error, about 1.1%.

Consider a very simple ℓ-layer fully connected neural network with dropout. To train it with a minibatch of b samples, the forward pass is described by the following equation:

$$x_{k+1} = [x_k \cdot d_k] \times W_k \qquad k = 0, \ldots, \ell - 1. \tag{5.1}$$

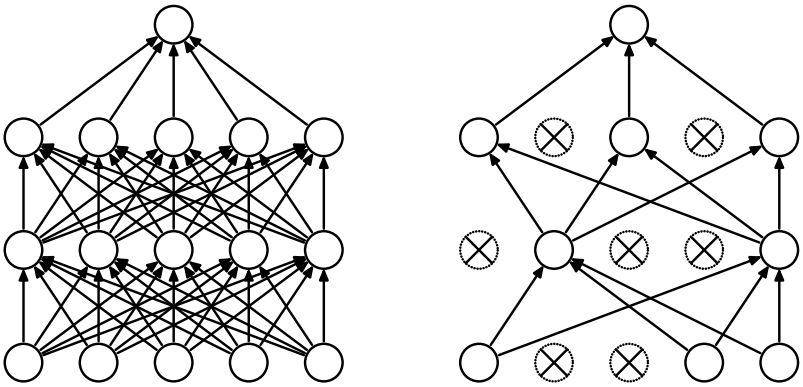
Here x_k is a $b \times n_k$ matrix of input/hidden/output units, d_k is a $b \times n_k$ dropout-mask matrix of independent Bernoulli$(1 - p_k)$ random variables, p_k denotes the probability of dropping out units in level k, and W_k is an $n_k \times n_{k+1}$ matrix of weights connecting level k with level $k + 1$. The $\cdot$ is used to signify element-wise (Hadamard) multiplication and $\times$ for usual matrix multiplication. To keep this example as simple as possible, I have omitted any non-linear functions (e.g. the rectifier function for the hidden units, and softmax for the output units). The result of applying such a dropout mask, d_k for one sample of the minibatch can be seen in Fig. 5.1.

The network can be trained using the backpropagation algorithm to calculate the gradients of a cost function (e.g. negative log-likelihood) with respect to the W_k:

$$\frac{\partial \text{cost}}{\partial W_k} = [x_k \cdot d_k]^\mathsf{T} \times \frac{\partial \text{cost}}{\partial x_{k+1}} \tag{5.2}$$

$$\frac{\partial \text{cost}}{\partial x_k} = \left(\frac{\partial \text{cost}}{\partial x_{k+1}} \times W_k^\mathsf{T} \right) \cdot d_k. \tag{5.3}$$

With dropout training, the goal is to minimize the cost function averaged over an ensemble of closely related networks - the ones created by the random removal of units. However, networks typically contain thousands of hidden units, so the size of the ensemble is *much* larger than the number of training samples that can possibly be sampled during training. This suggests that the independence of the rows of the dropout mask matrices d_k might not be important; the success of dropout simply cannot depend on exploring a large fraction of the available dropout masks. In the context of this chapter this definition, usually referred to as simply dropout will be referred to as *independent* dropout.



(a) Standard network. (b) Network with dropout.

Figure 5.1: Applying dropout to a fully connected neural network with two layers. The standard network 5.1(a) has been thinned by dropping out the crossed units (with a probability p_k) resulting in the network having a topology shown in 5.1(b). After (Srivastava et al. 2014).

Batchwise dropout

A network utilising dropout takes longer to train in two senses: each training epoch takes several times longer, and the number of training epochs needed increases too. This remainder of this chapter will present a technique for speeding up training with dropout, resulting in a substantial reduction in the time needed per epoch.

Some machine learning libraries such as Pylearn2 allow dropout to be applied in a batchwise manner instead of independently[1]. This is done in a naive fashion by replacing d_k with a $1 \times n_k$ row matrix of independent Bernoulli$(1 - p_k)$ random variables, and then copying it vertically b times to get the right shape. I now present an efficient version of batchwise dropout that takes

[1] Pylearn2 source: apply_dropout() in mlp.py - https://github.com/lisa-lab/pylearn2/blob/master/pylearn2/models/mlp.py

advantage of the redundancy inherent in the dropout mask once the constraint of independence is relaxed.

To be practical, it is important that each training minibatch can be processed quickly. A crude way of estimating the processing time involved is to count the number of floating point multiplication operations needed (naively) to evaluate the $\times$ matrix multiplications specified in Eqns 5.1, 5.2 and 5.3 above, which results in the following expression:

$$\sum_{k=0}^{\ell-1} \underbrace{b \times n_k \times n_{k+1}}_{\text{forwards}} + \underbrace{n_k \times b \times n_{k+1}}_{\partial\text{cost}/\partial W} + \underbrace{b \times n_{k+1} \times n_k}_{\text{backwards}}. \tag{5.4}$$

However, when taking into account the effect of the dropout mask, it can be seen that many of these multiplications are unnecessary. The (i, j)-th element of the W_k weight matrix effectively 'drops-out' of the calculations if unit i is dropped in level k, or if unit j is dropped in level $k+1$. Applying 50% dropout in levels k and $k+1$ renders 75% of the multiplications unnecessary.

If independent dropout is applied to a network, then the parts of W_k that disappear are different for each sample. This makes it effectively impossible to take advantage of the redundancy as it is typically slower to check if a multiplication is necessary than to just do the multiplication. However, if batchwise dropout is used, then it becomes easy to take advantage of the redundancy by simply removing the redundant parts of the calculations.

The binary $1 \times n_k$ batchwise dropout matrices d_k naturally define submatrices of the weight and hidden-unit matrices. Let $x_k^{\text{dropout}} := x_k[:, d_k]$ denote the submatrix of x_k consisting of the level-k hidden units that survive dropout. Let $W_k^{\text{dropout}} := W_k[d_k, d_{k+1}]$ denote the submatrix of W_k consisting of weights that connect active units in level k to active units in level $k+1$. The network can then be trained using the following equations:

$$x_{k+1}^{\text{dropout}} = x_k^{\text{dropout}} \times W_k^{\text{dropout}} \tag{5.5}$$

$$\frac{\partial \text{cost}}{\partial W_k^{\text{dropout}}} = (x_k^{\text{dropout}})^\mathsf{T} \times \frac{\partial \text{cost}}{\partial x_{k+1}^{\text{dropout}}} \tag{5.6}$$

$$\frac{\partial \text{cost}}{\partial x_k^{\text{dropout}}} = \frac{\partial \text{cost}}{\partial x_{k+1}^{\text{dropout}}} \times (W_k^{\text{dropout}})^\mathsf{T} \tag{5.7}$$

The redundant multiplications have been eliminated. There is an additional benefit in terms of memory needed to store the hidden units: x_k^{dropout} needs less space than x_k.

It should be emphasized that batchwise dropout only improves performance during training; during testing the full W_k matrix is used as normal, scaled by a factor of $1-p_k$. However, machine learning research is often constrained by long training times and high costs of equipment. In Section 5.5 the results will show that all other things being equal, batchwise dropout is similar to independent dropout, but faster. Moreover, with the increase in speed, all other things do not have to be equal. For instance, with the same resources, batchwise dropout can be used to

- increase the number of training epochs,
- increase the number of hidden units,

- increase the number of validation runs used to optimize "hyper-parameters", or
- train a number of independent copies of the network to form a committee.

These possibilities will often be useful as ways of improving generalization/reducing test error.

The rest of this chapter will attempt to characterise the implications of replacing independent dropout with that of batchwise dropout. In Section 5.6 batchwise dropout will be extended for use with convolutional networks. Dropout for convolutional networks is somewhat more complicated as weights are shared across spatial locations. A minibatch passing up through a convolutional network might be represented at an intermediate hidden layer by an array of size $100 \times 32 \times 12 \times 12$: 100 samples, the output of 32 convolutional filters, at each of 12×12 spatial locations. It is conventional to use a dropout mask with shape $100 \times 32 \times 12 \times 12$; again this will called independent dropout. In contrast, if the goal is to to apply batchwise dropout efficiently by adapting the submatrix trick, then the dropout mask will take on the shape $1 \times 32 \times 1 \times 1$. This looks like a significant change: the ensemble over which the average cost is optimised is different. During training, the error rates are higher. However, testing the networks gives very similar error rates.

Implementation

To see how much of a performance gain that should be expected by being able to remove 50% of the rows and columns of W_k (assuming a p_k of 0.5) a consideration of the time complexity of the operations involved must be undertaken. In theory, for $n \times n$ matrices, addition is an $O(n^2)$ operation, and multiplication is $O(n^{2.37\cdots})$ by the Coppersmith–Winograd algorithm. This suggests that the bulk of the processing time should be spent doing matrix multiplication, and that a performance improvement of about 60% should be possible compared to networks using independent dropout, or no dropout at all. In practice, SGEMM functions use Strassen's algorithm or naive matrix multiplication, so performance improvements of up to 75% should be possible[2].

Batchwise dropout was implemented for fully-connected and convolutional neural networks using CUDA/CUBLAS. It was clear that using the highly optimized cublasSgemm function to do the bulk of the work, with CUDA kernels used to form the submatrices W_k^{dropout} and to update the W_k using $\partial\text{cost}/\partial W_k^{\text{dropout}}$, worked well. Better performance may well be obtained by writing a SGEMM-like matrix multiplication function that natively understands submatrices, though this was considered beyond the scope of the current work. For large networks and minibatches, batchwise dropout is substantially faster, see Fig. 5.2. The approximate overlap of some of the lines on the left indicates that 50% batchwise dropout reduces the training time in a similar manner to halving the number of hidden units.

The graph on the right show the time saving obtained by using submatrices to implement dropout. Note that for consistency with the left hand side, the graph compares batchwise dropout with dropout-free networks, *not* with networks using independent dropout. The need to implement dropout masks for independent dropout means that Fig. 5.2 slightly undersells the performance benefits of batchwise dropout as an alternative to independent dropout.

[2]Coppersmith-Winograd would give $2(n/2)^{2.37} \approx 0.4n^{2.37}$, while the naive algorithm would give $2(n/2)^3 = 0.25n^3$.

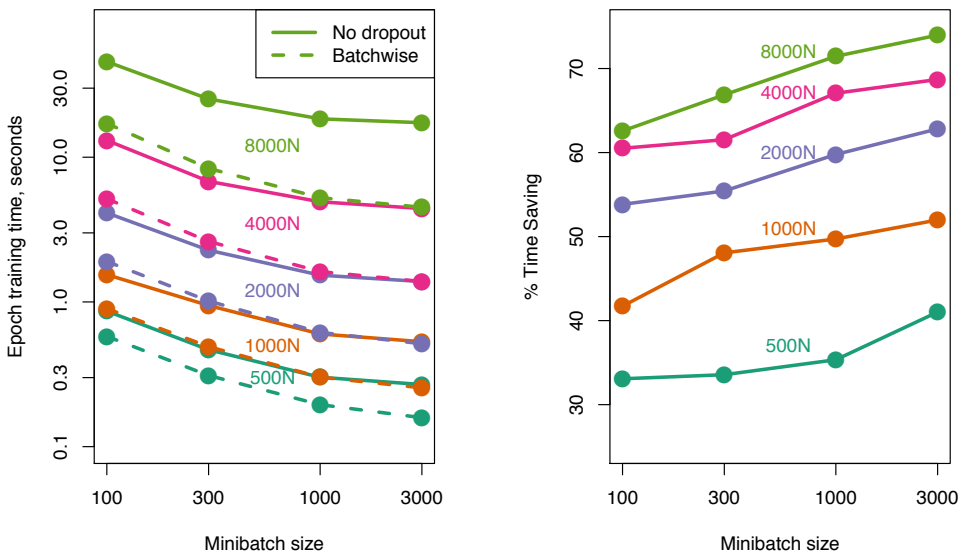
Efficiency considerations

If you have $n = 2000$ hidden units and you drop out $p = 50\%$ of them, then the expected number of dropped units is $np = 1000$, but with some small variation as you are really dealing with a Binomial(n, p) random variable - its standard deviation is $\sqrt{np(1-p)} = 22.4$. The sizes of the submatrices W_k^{dropout} and x_k^{dropout} are therefore slightly random. In the interests of both efficiency and simplicity, it is convenient to remove this randomness. An alternative to dropping each unit independently with probability p is to drop a subset of exactly np of the hidden units, uniformly at random from the set of all $\binom{n}{np}$ such subsets. It is still the case that each unit is dropped out with probability p. However, within a hidden layer there is no longer strict independence regarding which units are dropped out. Thus the probability of dropping out the first two hidden units changes very slightly, from

$$p^2 = 0.25 \qquad \text{to} \qquad \frac{np}{n} \cdot \frac{np-1}{n-1} = 0.24987.... \tag{5.8}$$

Also, in the experiments within this chapter a modified form of NAG-momentum minibatch gradient descent (Sutskever et al. 2013) was used. After each minibatch, only the elements of W_k^{dropout} are updated, not the entire W_k. With v_k and v_k^{dropout} denoting the momentum matrix/submatrix corresponding to W_k and W_k^{dropout}, the update becomes

$$\begin{aligned} v_k^{\text{dropout}} &\leftarrow \mu v_k^{\text{dropout}} - \varepsilon(1-\mu)\partial \text{cost}/\partial W_k^{\text{dropout}} \\ W_k^{\text{dropout}} &\leftarrow W_k^{\text{dropout}} + v_k^{\text{dropout}}. \end{aligned}$$



(a) Training time in seconds.

(b) Percentage time saving.

Figure 5.2: Left: MNIST training time for three layer networks (log scales) on an NVIDIA GeForce GTX 780 graphics card. Right: Percentage reduction in training times moving from no dropout to batchwise dropout. The time saving for the 500N network with minibatches of size 100 increases from 33% to 42% if you instead compare batchwise dropout with independent dropout.

The momentum still functions as an autoregressive process, smoothing out the gradients, just with a rate of decay μ reduced by a factor of $(1 - p_k)(1 - p_{k+1})$.

Results for fully-connected networks

The fact that batchwise dropout takes less time per training epoch would count for nothing if a much larger number of epochs was needed to train the network, or if a large number of validation runs were needed to optimize the training process. The following set of experiments compare independent and batchwise dropout to characterise these potential tradeoffs.

In many cases demonstrated in this section better results could be obtained by increasing the training time, annealing the learning rate, using validation runs to adjust the learning process, etc. These techniques were not explored as the primary motivation for batchwise dropout is efficiency, and excessive use of fine-tuning is not efficient.

For datasets, the following were used:

- The MNIST[3] set of 28×28 pixel handwritten digits.
- The CIFAR-10[4]dataset of 32x32 pixel color pictures (Krizhevsky 2009).
- An artificial dataset designed to be easy to overfit.

Following Srivastava et al. (2014), for MNIST and CIFAR-10 the networks were trained with 20% dropout in the input layer, and 50% dropout in the hidden layers. For the artificial dataset the input-layer dropout value was increased to 50% as this reduced the test error. In some cases, relatively small networks were used so that the time to train a number of independent copies of the networks was manageable. This was useful in order to see if the apparent differences between batchwise and independent dropout are significant or just noise.

MNIST

The first experiment explores the effect of dramatically restricting the number of dropout patterns seen during training. Consider a network with three hidden layers of size 1000, trained for 1000 epochs using minibatches of size 100. The number of distinct dropout patterns, 2^{3784}, is so large that it can be assumed that the same dropout mask will not be generated twice. During independent dropout training around 60 million different dropout patterns will be used in comparison to that during batchwise dropout training where there will be 100 times fewer dropout patterns.

For both types of dropout, 12 independent networks were trained for 1000 epochs, with batches of size 100. For batchwise dropout the mean test error was 1.04% [range (0.92%,1.1%), s.d. 0.057%] and for independent dropout the mean test error of 1.03% [range (0.98%,1.08%), s.d. 0.033%]. The difference in these mean test errors is not statistically significant.

To further explore the reduction in the number of dropout patterns seen, modifications to the code for (pseudo)randomly generating batchwise dropout patterns were undertaken to restrict the number of distinct dropout patterns used. Specifically, it was modified to have period n minibatches, with $n = 1, 2, 4, 8, \ldots$; see Fig. 5.3. For $n = 1$ this corresponds to only ever using one dropout mask, so that 50% of the network's 3000 hidden weights are never actually trained (and

[3]available: http://yann.lecun.com/exdb/mnist/
[4]available: http://www.cs.toronto.edu/~kriz/cifar.html

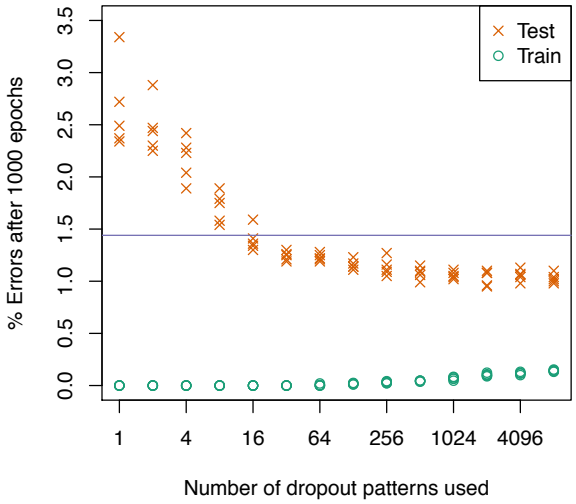


Figure 5.3: Dropout networks trained using a restricted the number of dropout patterns (each × is from an independent experiment). The blue line marks the test error for a network with half as many hidden units trained without dropout.

20% of the 784 input features are ignored). During training this corresponds to training a dropout-free network with half as many hidden units - the test error for such a network is marked by a blue line in Fig. 5.3. The error during testing is higher than the blue line because the untrained weights add noise to the network.

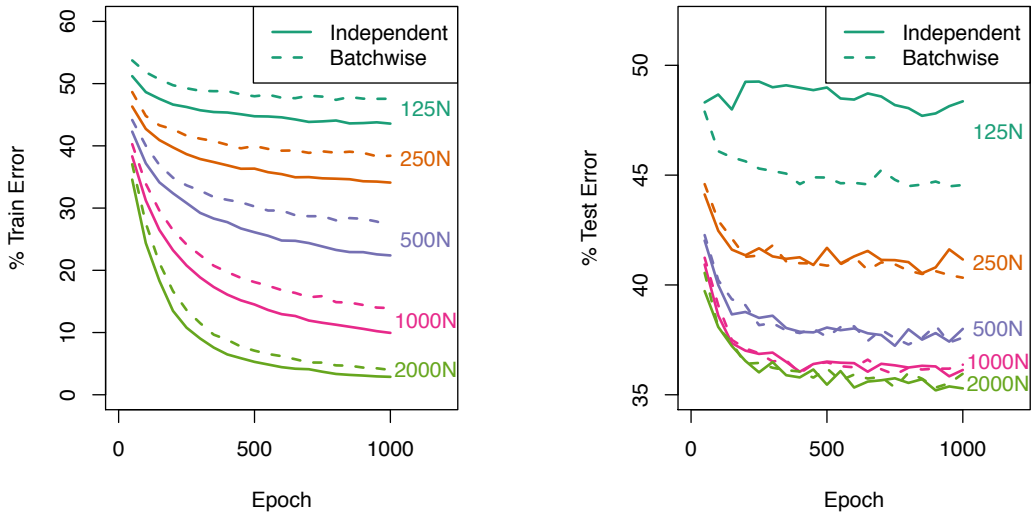
If n is less than thirteen, is it likely that some of the networks 3000 hidden units are dropped out every time and so receive no training. If n is in the range thirteen to fifty, then it is likely that every hidden unit receives some training, but some pairs of hidden units in adjacent layers will not get the chance to interact during training, so the corresponding connection weight is untrained. As the number of dropout masks increases into the hundreds, it is clear that you enter a regime of diminishing returns.

CIFAR-10 fully-connected

Learning CIFAR-10 using a fully connected network is rather difficult. Three layer networks were trained that had $n \in \{125, 250, 500, 1000, 2000\}$ hidden units per layer with minibatches of size 1000. The training dataset was augmented with horizontal flips. See Fig. 5.4.

Artificial dataset

To test the effect of changing network size, an artificial dataset was created. It consists of 100 classes, each containing 1000 training samples and 100 test samples. Each class is defined using an independent random walk of length 1000 in the discrete cube $\{0, 1\}^{1000}$. For each class a random walk is generated and then used to produce the training and test samples by randomly picking points along the length of walk (giving binary sequences of length 1000) and then randomly flipping 40% of the bits. Three layer networks were trained that had $n \in \{250, 500, 1000, 2000\}$ hidden units per layer with minibatches of size 100. See Fig. 5.5.



(a) Training error.

(b) Testing error.

Figure 5.4: Results comparing the performance of independent and batchwise dropout on the CIFAR-10 dataset using fully-connected networks of different sizes.

Looking at the training error against training epochs, independent dropout seems to learn slightly faster. However, looking at the test errors over time, there does not seem to be much difference between the two forms of dropout. Note that the x-axis is the number of training epochs, not the training time. The batchwise dropout networks are learning much faster in terms of 'wall-clock' time.

Results for convolutional networks

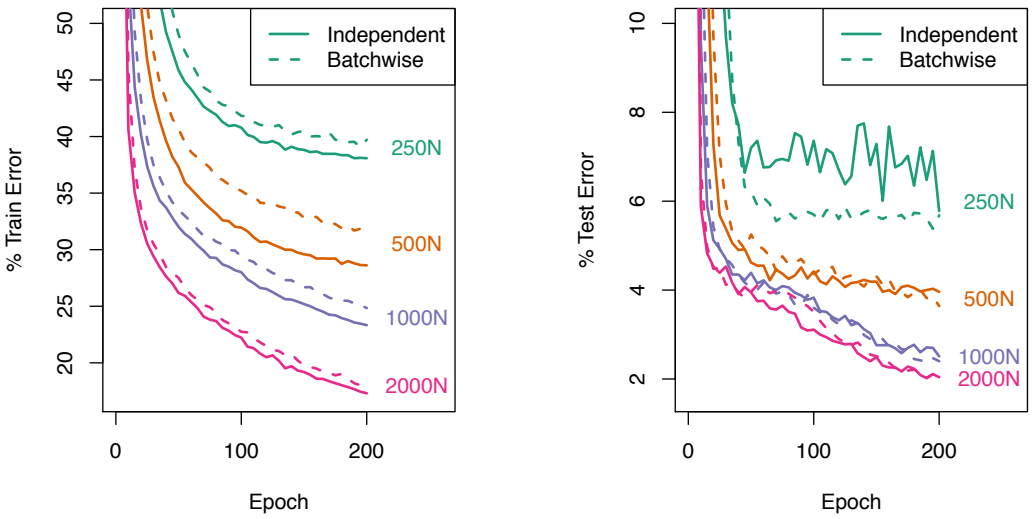
Dropout for convolutional networks is more complicated as weights are shared across spatial locations. Suppose layer k has spatial size $s_k \times s_k$ with n_k features per spatial location, and if the k-th operation is a convolution with $f \times f$ filters. For a minibatch of size b, the convolution involves arrays with sizes:

$$\text{layer } k : b \times n_k \times s_k \times s_k \tag{5.9}$$

$$\text{weights } W_k : n_{k+1} \times n_k \times f \times f \tag{5.10}$$

As with the fully connected case, dropout is normally applied using dropout masks with the same size as the layers. This will be called independent dropout - independent decisions are made at every spatial location. In contrast, batchwise dropout is now defined as a dropout mask with shape $1 \times n_k \times 1 \times 1$. Each minibatch, each convolutional filter is either on or off across all spatial locations.

These two forms of regularization seem to be doing quite different things. Consider a filter that detects the color red, and a picture with a red truck in it. If dropout is applied independently, then by the law of averages the message "red" will be transmitted with very high probability, but



(a) Training error.

(b) Testing error.

Figure 5.5: Results comparing the performance of independent and batchwise dropout on the artificial dataset using networks of different sizes. 100 classes each corresponding to noisy observations of a one dimensional manifold in $\{0,1\}^{1000}$.

with some loss of spatial information. In contrast, with batchwise dropout there is a 50% chance that the entire filter output is deleted. Experimentally, the only substantial difference between the methods is that batchwise dropout results in larger errors during training.

To implement batchwise dropout efficiently, notice that the $1\times n_k\times 1\times 1$ dropout masks corresponds to forming subarrays W_k^{dropout} of the weight arrays W_k with size

$$(1-p_{k+1})n_{k+1}\times(1-p_k)n_k\times f\times f. \tag{5.11}$$

The forward-pass is then simply a regular convolutional operation using W_k^{dropout}; that makes it possible, for example, to take advantage of the highly optimized cudnnConvolutionForward function from the NVIDIA cuDNN[5] package.

MNIST

For MNIST, a LeNet-5 type CNN was trained with two layers of 5×5 filters, two layers of 2×2 max-pooling, and a fully connected layer (LeCun et al. 1990; Lecun et al. 1998). There are three places for applying 50% dropout:

$$32\text{C5}-\text{MP2}\overset{50\%}{-}64\text{C5}-\text{MP2}\overset{50\%}{-}512\text{N}\overset{50\%}{-}10\text{N}.$$

The test errors for the two dropout methods are similar, see Fig. 5.6.

[5]available: https://developer.nvidia.com/cuDNN

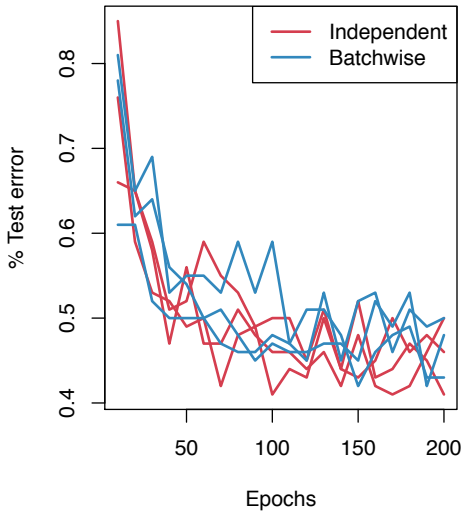


Figure 5.6: MNIST test errors, training repeated three times for both dropout methods.

CIFAR-10 with varying dropout intensity

For an initial experiment with CIFAR-10 a relatively small convolutional network with small filters was defined. The network is a scaled down version of the network from (Ciresan et al. 2012); there are four places to apply dropout:

$$128\text{C3} - \text{MP2} \overset{p}{-} 256\text{C2} - \text{MP2} \overset{p}{-} 384\text{C2} - \text{MP2} \overset{p}{-} 512\text{N} \overset{p}{-} 10\text{N}.$$

The input layer is 24×24. The network was trained for 1000 epochs using randomly chosen subsets of the training images, while also reflecting each image horizontally with probability of 0.5. For testing the centres of the images were used.

In Fig. 5.7 the effect of varying the dropout probability p is demonstrated. The training errors are increasing with p, and the training errors are higher for batchwise dropout. The test-error curves both seem to have local minima around $p = 0.2$. The batchwise test error curve seems to be shifted slightly to the left of the independent one, suggesting that for any given value of p, batchwise dropout is a slightly stronger form of regularization.

CIFAR-10 with many convolutional layers

A deep convolutional network was trained on CIFAR-10, this time *without* specific data augmentation. Using the notation of Graham (2014), the network has the form

$$(64nC2 - FMP\sqrt[3]{2})_{12} - 832C2 - 896C1 - \text{output},$$

i.e. it consists of 12 2×2 convolutions with $64n$ filters in the n-th layer ($64nC2$), 12 layers of fractional max-pooling (Graham 2014), followed by two fully connected layers; the network has approximately 12.6 million parameters. An increasing amount of dropout per layer was applied,

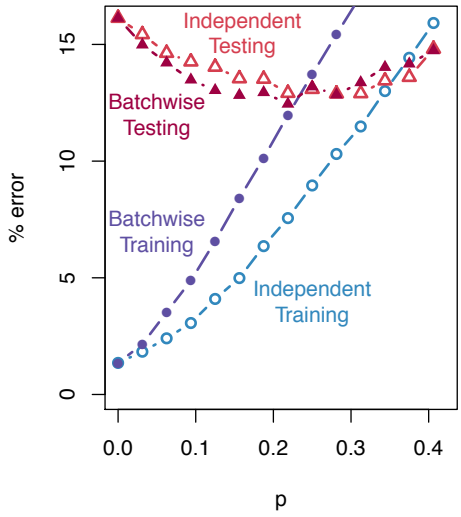


Figure 5.7: CIFAR-10 results using a convolutional network with dropout probability $p \in (0, 0.4)$. Batchwise dropout produces a slightly lower minimum test error.

rising linearly from 0% dropout after the third layer to 50% dropout after the 14th. Even though the amount of dropout used in the intermediary layers is comparatively small, batchwise dropout took less than half as long per epoch as independent dropout; this is because applying small amounts of independent dropout in large hidden-layers creates a bandwidth performance-bottleneck.

As the network's max-pooling operation is stochastic, the test errors can be reduced by repetition. Batchwise dropout resulted in a average test error of 7.70% (down to 5.78% with 12-fold testing). Independent dropout resulted in an average test error of 7.63% (reduced to 5.67% with 12-fold testing). Again this is further confirmation that batchwise dropout does not seem to incur any particular performance penalties.

Discussion

This chapter presented an implementation of an efficient form of batchwise dropout. All other things being equal, it learns at roughly the same speed in terms of numbers of epochs as independent dropout, but each epoch is faster in terms of 'wall-clock' time. Given a fixed computational budget, it will often allow you to train better networks.

The gains in training speed were network size dependent. For large networks and minibatches, batchwise dropout is substantially faster. In somewhat smaller networks, the performance improvement is lower since bandwidth issues result in the GPU being under utilized (see Fig. 5.2). If you were implementing batchwise dropout for CPUs, you would expect to see greater performance gains for smaller networks as CPUs have a lower processing-power to bandwidth ratio.

Fast dropout

A natural choice might have been to call batchwise dropout *fast dropout* but that name is already taken by Wang and Manning (2013). Fast dropout is an alternative form of regularization that uses a probabilistic modeling technique to imitate the effect of dropout; each hidden unit is replaced with a Gaussian probability distribution. The *fast* relates to reducing the number of training epochs needed when compared to regular independent dropout (with reference to results in a preprint[6] of (Srivastava et al. 2014)). Training a network 784-800-800-10 on the MNIST dataset with 20% input dropout and 50% hidden-layer dropout, fast dropout converges to a test error of 1.29% after 100 epochs of L-BFGS. This appears to be substantially better than the test error obtained in the preprint after 100 epochs of regular dropout training.

However, this is a potentially dangerous comparison to make. The authors of (Srivastava et al. 2014) used a learning-rate scheme designed to produce optimal accuracy eventually, *not* after just one hundred epochs. Batchwise dropout with minibatches of size 100 and an annealed learning rate of $0.01e^{-0.01\times\text{epoch}}$ was used on a similar network with two hidden layers of 800 rectified linear units each. In this case, training for 100 epochs resulted in a test error of 1.22% (s.d. 0.03%). After 200 epochs the test error has reduced further to 1.12% (s.d. 0.04%). Moreover, per epoch, batchwise dropout is faster than regular independent dropout while fast dropout is actually slower. Assuming comparisons across different programs are meaningful[7], the 200 epochs of batchwise dropout training take less time than the 100 epoch of fast dropout training.

Future Work

There are other potential uses for batchwise dropout that have not explored yet:

- Restricted Boltzmann Machines can be trained by contrastive divergence (Hinton et al. 2006) with dropout (Srivastava et al. 2014). Batchwise dropout could be used to increase the speed of training.
- When a fully connected network sits on top of a convolutional network, training the top and bottom of the network can be separated over different computational nodes (Krizhevsky 2014). The fully connected top-parts of the network typically contains 95% of the parameters - keeping the nodes synchronized is difficult due to the large size of the matrices. With batchwise dropout, nodes could communicate $\partial\text{cost}/\partial W_k^{\text{dropout}}$ instead of $\partial\text{cost}/\partial W_k$ and so reducing the bandwidth needed.
- Using independent dropout with recurrent neural networks can be too disruptive to allow effective learning; one solution is to only apply dropout to some parts of the network (Zaremba et al. 2014). Batchwise dropout may provide a less damaging form of dropout, as each unit will either be on or off for the whole time period.
- Dropout is normally only used during training. It is generally more accurate to use the whole network for testing purposes; this is equivalent to averaging over the ensemble of dropout patterns. However, in a "real-time" setting, such as analyzing successive frames from a video

[6] http://arxiv.org/abs/1207.0580

[7] In the software used for the simulations in this chapter, each batchwise dropout training epoch take 0.67 times as long as independent dropout. In (Wang and Manning 2013) a figure of 1.5 is given for the ratio between their fast and independent-dropout when using minibatch SGD; when using L-BFGS to train fast-dropout networks the training time per epoch will presumably be even more than 1.5 times longer, as L-BFGS use line-searches requiring additional forward passes through the neural network.

camera, it may be more efficient to use batchwise dropout during testing, and then to average the output of the network over time.

- Nested dropout (Rippel et al. 2014) is a variant of regular dropout that extends some of the properties of PCA to deep networks. Batchwise nested dropout is particularly easy to implement as the submatrices are regular enough to qualify as matrices in the context of the SGEMM function (using the LDA argument).
- DropConnect is an alternative form of regularization to dropout (Wan et al. 2013). Instead of dropping hidden units, individual elements of the weight matrix are dropped out. Using a modification similar to the one in Section 5.4.1, there are opportunities for speeding up DropConnect training by approximately a factor of two.

Locally Connected Deep Belief Networks

Introduction

Over the last few years there has been great interest in the machine learning community regarding the development of 'deep' generative models. These models are capable of building rich hierarchical representations of input data in an unsupervised fashion. Deep usually refers to the model having many stages or layers where typically non-linear processing of input from the proceeding layer occurs. These are perhaps best exemplified by multiple layer neural network-like architectures. This is contrasted with 'shallow' models, for example kernel methods such as support vector machines (SVM). These models typically involve one stage of template matching via inner products of the input data with the set of learned kernels, followed by some weighted linear combination that seeks to associate the input vector with some target output. Theoretically, deep networks seem to have an advantage over 'shallow' methods, having been shown to be able to represent a given function with potentially exponentially fewer units than an equivalent network that is one layer shallower (Bengio and LeCun 2007).

Until recently, training deep models had proven to be a difficult task using methods that work well with shallow architectures. Naively adding extra layers fails to produce the anticipated gains in model performance (Bengio et al. 2007). This is excluding notable exceptions like convolutional networks (LeCun et al. 1990) and VisNet (Rolls 2008c; Wallis and Rolls 1997b). The problem of training deep models became much more tractable with the work of Hinton et al. (2006) and the introduction of Deep Belief Networks (DBNs). The DBN is a deep model created by composing together a series of layers made up from Restricted Boltzmann Machines (RBMs), each of which were trained using a greedy unsupervised methods. The activation probabilities or *mean field* approximations of one layer are simply used as the inputs to the subsequent one for all N layers of the model. To sidestep the issue of not being able to analytically evaluate the partition function of such models, an efficient training algorithm based upon estimating the maximum likelihood gradient was constructed. This algorithm, *contrastive divergence* estimates the gradient by sampling from the model with a truncated (in practice usually one step) block Gibbs sampling initialised at an exemplar from the training set (Hinton 2002).

DBNs have recently been successfully applied to many classic machine learning tasks, ranging from image (Huang et al. 2012; Lee et al. 2009a) and audio (Lee et al. 2009b) classification, dimensionality reduction (Hinton and Salakhutdinov 2006), and semantic hashing (Salakhutdinov and Hinton 2008).

Given that the visual cortex is by its very nature a deep hierarchical network (Knierim and Van Essen 1992; Van Essen et al. 1992b) it seems promising that these developments may allow for the construction of biologically plausible deep models. Thus the goal of the following chapter is to

introduce the Deep Belief Network model, and propose an extension that restricts the connectivity of the neurons to that of a local topological region of the preceding layer - which has a connectivity closer to that understood to be found within the visual cortex (Rolls 2012b). The result of imposing such a topology on the DBN model will be explored by considering how well the model can then

Deep Belief Network Architecture

Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) is a complete, bipartite, undirected graphical model consisting of a set of binary, visible $\mathbf{v}$ and hidden $\mathbf{h}$ units (see Fig. 6.1).

A key property of RBMs (in contrast to traditional Boltzmann Machines) is that in removing the lateral connections within layers and relaxing the connectivity to be undirected, the units in a particular layer are conditionally independent given a set of observed activations in the other. This is a key difference that allows for the efficient learning and inference within these models. A natural way to analyse the dynamics of a RBM is to cast it as an energy based model and

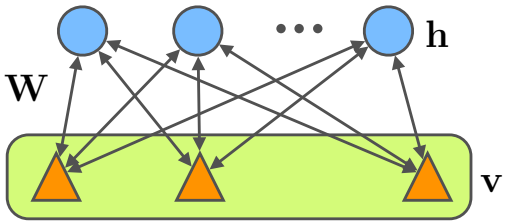


Figure 6.1: The general structure of a Restricted Boltzmann Machine (RBM) is comprised of a bipartite graph of nodes which are labelled visible, $\mathbf{v}$ and hidden $\mathbf{h}$. The pair-wise interactions of nodes between layers is defined by a connectivity matrix, $\mathbf{W}$.

utilise many results from statistical mechanics. Within this framework, each state of the RBM is associated with a particular energy value, $E(\mathbf{v}, \mathbf{h}; \theta)$. The parameters, θ are $\{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$, where $\mathbf{W}$ is the weight matrix defining the interactions between pairs of units between layers and $\mathbf{b}$ and $\mathbf{c}$ are bias vectors for the visible and hidden layers respectively. This allows for the probability of a particular state to be defined by the Boltzmann distribution,

$$P(\mathbf{v}; \theta) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}; \theta)} \tag{6.1}$$

with $Z(\theta)$ being the partition or normalising function, $\sum_{\mathbf{v},\mathbf{h}} P(\mathbf{v}, \mathbf{h}; \theta)$. Notice that this function is a combinatorial sum over all the possible states of the model so for all but trivial models sizes will not be computable.

In general the nature of the energy function, E depends upon the underlying distributions of the individual units, though in the case of binary units can be written as

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i,j} \mathbf{v}_i \mathbf{W}_{ij} \mathbf{h}_j - \sum_i \mathbf{b}_i \mathbf{v}_i - \sum_j \mathbf{c}_j \mathbf{v}_j \tag{6.2}$$

Inference is straightforward within this model since we have guaranteed conditional independence. Sampling from the conditionals, $P(\mathbf{v}|\mathbf{h})$ and $P(\mathbf{h}|\mathbf{v})$ can be done exactly, taking the form:

$$P(\mathbf{h}_j = 1|\mathbf{v}) = sigm\left(\mathbf{c}_j + \sum_i \mathbf{v}_i \mathbf{W}_{ij}\right) \tag{6.3a}$$

$$P(\mathbf{v}_i = 1|\mathbf{h}) = sigm\left(\mathbf{b}_i + \sum_j \mathbf{W}_{ij}\mathbf{j}_j\right) \tag{6.3b}$$

where $sigm(x) = 1/(1 + e^{-x})$, the standard logistic sigmoid function.

RBMs can also be extended to model more complex types of data (continuous, multimodal, etc) by being composed of units with a distribution from any of the exponential family (Welling et al. 2005). For simplicity the derivations in this section have focused on the simplest case, where all units are stochastic binary in nature.

When attempting to model natural images an input layer that is restricted to binary variables is not expressive enough to represent the data - natural images have many pixels that have intermediary values which are poorly modelled by a binary unit. To solve this problem units in the visible layer are redefined to have Gaussian distribution, resulting in the Gaussian RBM (GRBM). This modification is advantageous as it leaves the training process with contrastive divergence unchanged. The only modifications needed are to the energy function (Eqn 6.2) and associated conditional distributions (Eqn 6.3) as follows,

$$E(\mathbf{v}, \mathbf{h}; \theta) = \sum_i \frac{(\mathbf{v}_i - \mathbf{b}_i)^2}{2\sigma_i^2} - \sum_{i,j} \frac{\mathbf{v}_i}{\sigma_i^2} \mathbf{W}_{ij} \mathbf{h}_j - \sum_j \mathbf{c}_j \mathbf{v}_j \tag{6.4}$$

for the energy, and

$$P(\mathbf{h}_j = 1|\mathbf{v}) = sigm\left(\frac{\mathbf{c}_j}{\sigma_i^2} + \sum_i \mathbf{v}_i \mathbf{W}_{ij}\right) \tag{6.5a}$$

$$P(\mathbf{v}_i = v|\mathbf{h}) = \mathcal{N}\left(v \mid \mathbf{b}_i + \sum_j \mathbf{W}_{ij}\mathbf{h}_j, \sigma_i^2\right) \tag{6.5b}$$

for the conditional distributions.

These expressions depend on another set of parameters, namely the variances of each of the Gaussian units, σ. While these variances can be one of the learnt parameters (Tang and Mohamed 2012) in practice the Gaussian input neurons are typically assumed to have unit variance and the data is preprocessed by scaling.

Contrastive Divergence Learning

Having defined a graphical model, we now want to be able to fit the parameters of the model in such a way that the statistical features and underlying distribution within the dataset are reflected by the model. A natural way to do this is to consider maximising the likelihood of the dataset using gradient descent. That is, if we have a training set, D consisting of k n-dimensional data

vectors, we want to maximise the log-likelihood,

$$\log L = \log P(D) \tag{6.6}$$
$$= \sum_{\mathbf{v} \in D} \log P(\mathbf{v}; \theta) \tag{6.7}$$
$$= \sum_{\mathbf{v} \in D} (\log P^*(\mathbf{v}; \theta) - \log Z(\theta)) \tag{6.8}$$

where $P^*(\mathbf{v})$ is the un-normalised probability.

This allows us to compute the following expression for the gradient,

$$\frac{\partial}{\partial w} \log L \propto \underbrace{\frac{1}{N} \sum_{\mathbf{v} \in D}}_{\text{data}} \underbrace{\sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{v})}_{\text{posterior}} \frac{\partial}{\partial w} \log P^*(\mathbf{v}, \mathbf{h}) - \underbrace{\sum_{\mathbf{v}, \mathbf{h}} P(\mathbf{v}, \mathbf{h})}_{\text{joint}} \frac{\partial}{\partial w} \log P^*(\mathbf{v}, \mathbf{h}) \tag{6.9}$$

Notice that this gradient is made up of only two summations and that each one is an average over the expression $\frac{\partial}{\partial w} \log P^*(\mathbf{v}, \mathbf{h}; \theta)$. In the case of an RBM, this expression is straightforward to calculate, and turns out to be the simple product $\mathbf{v}_i \mathbf{h} j$. The first term is the expectation when the visible variables are set to an exemplar from the dataset and the hidden variables are sampled from the posterior, conditional $P(\mathbf{h}|\mathbf{v})$. The second term is the expectation when $\mathbf{v}$ and $\mathbf{h}$ are sampled from the joint, $P(\mathbf{v}, \mathbf{h})$. These terms are sometimes referred to as the 'clamped' (as in clamped to an exemplar from the dataset) and 'unclamped' terms. This compact expression is valid for any such bipartite graphical model and not just in the specific family of models that this chapter is concerned with.

While the gradient defined in Equation 6.9 is very appealing, it disguises the problem of the intractability of the partition function. We are unable to draw samples directly from the joint distribution and so will have to turn to estimation methods to compute the gradient. A naive approach would be to run a Markov Chain Monte Carlo algorithm such as Gibbs sampling, but this also requires that we generate sufficient samples such that the Gibbs sampling reaches the equilibrium distribution, which is a computationally expensive operation - something we wish to avoid as we will be computing the gradient many times while training the model.

Maximising the log-likelihood of the data is equivalent to minimising the Kullback-Leibler (KL) divergence between the visible input distribution Q_0 and the equilibrium visible data distribution Q_∞ (Hinton et al. 2006). That is,

$$\frac{\delta(Q_0 \parallel Q_\infty)}{\delta w_{ij}} = \langle \mathbf{v}_i \mathbf{h}_j \rangle^0 - \langle \mathbf{v}_i \mathbf{h}_j \rangle^\infty \tag{6.10}$$

where the angle brackets $\langle . \rangle$ denote the expectations from Equation 6.9. Specifically $\langle \mathbf{v}_i \mathbf{h}_j \rangle^0$ is the expectation of the product initially and $\langle \mathbf{v}_i \mathbf{h}_j \rangle^\infty$ is the expectation once we reach equilibrium. As alluded to above, it is not generally possible to actually reach this equilibrium state as it would require us to perform many blocked Gibbs iteration cycles (sampling back and forth between visible and hidden states). However we can forgo waiting until we reach this equilibrium state and simply take one step. That is instead of minimising the KL divergence between the data and the equilibrium distribution, $(Q_0 \parallel Q_\infty)$ we can minimise the difference between $(Q_0 \parallel Q_\infty)$ and $(Q_1 \parallel Q_\infty)$. This is the idea behind Contrastive Divergence (CD) learning (Hinton 2002; Hinton

et al. 2006). In taking this difference we observe that the $\langle \mathbf{v}_i \mathbf{h}_j \rangle^\infty$ term cancels out leaving us with,

$$\frac{\delta}{\delta w_{ij}}(Q_0 \parallel Q_\infty - Q_0 \parallel Q_\infty) \approx \langle \mathbf{v}_i \mathbf{h}_j \rangle^0 - \langle \mathbf{v}_i \mathbf{h}_j \rangle^1 \tag{6.11}$$

This is only an approximation as there is a missing term, though it is generally small and in practice doesn't seem to dominate the above difference (Hinton et al. 2006).

To be clear, a single step of CD will be described demonstrating the way that the above gradients are computed: First the visible nodes are clamped to that of an exemplar from the training set, $\mathbf{v}^0$. The states $\mathbf{h}^0$ of the hidden nodes are evaluated by computing and then sampling from the conditional $P(\mathbf{h}^0|\mathbf{v}^0)$. This in turn allows us to sample $\mathbf{v}^1$ and subsequently sample $\mathbf{h}^1$. These values can then be used directly to compute an estimate of the gradient. Fig. 6.2 shows pictorially this 'up' and 'down' sampling process. It should be noted that contrastive divergence can be extended to give better estimates by simply performing more iterations; doing n such iterations is usually called nCD learning.

Truncating the chain to n steps, implies that the function that is being maximised is no longer the true log-likelihood from Eqn 6.8. Empirical results have shown that in practice this isn't an issue and the likelihood is still improved with each update (Hinton 2002).

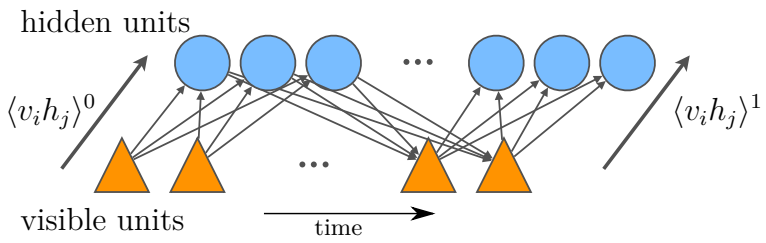


Figure 6.2: This diagram shows one step contrastive divergence (CD) learning. At time $t = 0$ we sample the states of the hidden units having clamped the visible units to an exemplar in the dataset. At time $t = 1$ we then sample the visible units given the previous hidden node samples which in turn allows a re-sampling of the hidden units. This iterative process can be computed more times to get better estimates, but commonly one is sufficient for gradient estimates good enough to learn with.

Coupled with a gradient descent family optimisation algorithm we arrive at the following set of parameter update rules using 1CD

$$\Delta \mathbf{W}_{ij} = \eta \left[\langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_1 \right] \tag{6.12a}$$

$$\Delta \mathbf{b}_j = \eta \left[\langle h_j \rangle_0 - \langle h_j \rangle_1 \right] \tag{6.12b}$$

$$\Delta \mathbf{c}_i = \eta \left[\langle v_i \rangle_0 - \langle v_i \rangle_1 \right] \tag{6.12c}$$

With these update equations we now are able to train a RBM on a given dataset. An important point to note in the context of this report is that CD learning is a local learning process. While on the surface it might appear to require individual units to have access to remote information (something that is difficult to imagine happening in biological neural network) during optimisation I believe this is incorrect. The CD learning process passes non-local information to individual units via the recurrent connectivity during training. The sampling steps only ever themselves only ever involve local information.

Deep Belief Networks

So far RBMs only have a single hidden layer. They are however the 'modules' that are combined together layer by layer into deeper models, the most common of which is the Deep Belief Network (DBN).

A DBN is simply a probabilistic model composed of many RBMs stacked on top of each other with each layer having been trained on the output of the previous one. Therefore each subsequent layer acts like constraints on the activities of the one below. Notice that this is a greedy training process and will likely not yield an optimum representation for the entire hierarchy - primarily because the individual layers were trained in isolation; once a layer is trained the weights are frozen. This inefficiency however can be remedied by a relatively small amount of fine tuning (generative or discriminative) after the stack is complete (Hinton and Salakhutdinov 2006).

It should be noted that a DBN itself is not a deep RBM. The lower layers do not themselves define an undirected model, only the top most two layers keep their undirected nature. However there has been work on modifications that restore the undirected nature of the lower layers allowing inference within the lower layers to make use of top-down connectivity as well as the usual bottom-up (Lee et al. 2009a; Salakhutdinov and Larochelle 2010).

Extending DBNs with local connectivity

So far DBNs lack many necessary features to be classed as a biologically plausible model of the visual cortex. The aspect of the model that will be considered in this section is the connectivity. The connectivity of the visual cortex is certainly not the all-to-all connectivity found in the DBN models presented thus far (see Fig. 6.1). Instead, within the visual cortex we find that neurons exhibit strong responses from topologically localized areas of the visual field (Hubel and Wiesel 1962, 1968b). A strong theory to account for this is a convergent local connectivity through the hierarchy of layers within the visual system (Knierim and Van Essen 1992; Rolls 2008b). This connectivity is a core requirement of a biological model of the visual cortex and is incorporated in successful models like VisNet (Rolls 2012b; Wallis and Rolls 1997a) and HMAX (Riesenhuber and Poggio 1999b; Serre et al. 2007d).

Imposing such a network topology upon a DBN acts as a strong prior on the model (since a training algorithm could in principle learn a local topology) at the cost of greatly reducing the number of units in the upper layer that can form representations of the image patch.

To introduce receptive fields into the DBN models is straightforward. A weighted adjacency matrix, $\mathbf{W}^*$ is defined, where $\mathbf{W}^*_{ij} = k_{ij}$ if the ith visible unit is within the jth's hidden unit's receptive field and zero everywhere else. This matrix is then multiplied element-wise with the interlayer weight matrix $\mathbf{W}$ after each gradient update step. fe to define a distance metric that defines k_{ij}, the strength of the connectivity. In the experiments described here the metric function was a simple Gaussian envelope of a given pixel radius, though similar results were obtained with a simple step function. DBN models with local connectivity will be referred to as a locally connected deep belief network (LCDBN).

Conceptually similar to local connectivity is the convolutional approach pioneered by Lecun et al. (1998) for neural networks. Within a convolution network many locally learned weights are convolved across the whole input image to produce the responses of neurons. This is accomplished

via explicit weight sharing, which while computationally elegant is difficult to imagine occurring in the cortex and thus not biologically plausible. A convolutional extension to the standard DBN is presented by Lee et al. (2009a). In contrast the receptive fields in the LCDBN only learn features that are useful at a specific visual field location

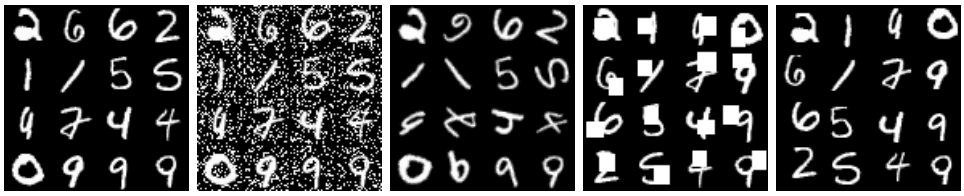
Results

MNIST classification experiments

A three hidden layer LCDBN model with binary input and hidden units was trained on the MNIST dataset[1] of handwritten digits. The network architecture was defined to have a structure of: 768-512-512-1024-10 - where the 768 layer is the input layer, three hidden layers of 512,512, and 1024; and the final output layer of 10 logistic regression units. Three sizes of receptive field were tested: 7, 11 and 14 pixels and was kept the same for each layer of the model. The receptive fields of successive units were shifted by one pixel, so adjacent units in the upper layer share a large overlap in input stimuli. Once the LCDBN was trained for 30 epochs with contrastive divergence and then fine tuned using back propagation for a further 30 epochs. The results can be seen in Table 6.1. It is clear that imposing such a topology hurts the discriminative performance of the model. The 7x7 LCDBN does more than twice as bad as the fully connected model (2.28% vs 5.12%) but has a sixteenth of the number of parameters.

RF Size	LCDBN	Fine tuned
28x28	3.1%	2.28%
14x14	5.6%	3.35%
11x11	5.8%	3.21%
7x7	6.0%	5.12%

Table 6.1: Table showing the percentage errors when a LCDBN is trained for a discriminative task on MNIST. Note that the 28 receptive field size is equivalent to the all-all connectivity



(a) (b) (c) (d) (e)

Figure 6.3: The MNIST handwritten digit database. The clean images (a) and the four corrupted versions: random noise (b), rotations (c), occlusions (d) and translations (e).

Given that local connectivity does not improve performance it is reasonable wonder if it has any other beneficial properties. A central problem that our visual cortex seems to have solved is how to create representations that are invariant to a host of common transformations. Perhaps the local connectivity helps with this task? To probe this question the models trained on the clean MNIST images were used to classify four corrupted sets of MNIST images. The corruptions tested were noise, rotations, translations and occlusions. For each type multiple datasets were created varying

[1] available at http://yann.lecun.com/exdb/mnist/

the amounts of corruption. For example, the rotation datasets have varying maximum rotation ranges and the occlusions dataset have varying sized occluders. The created MNIST variants with these corruptions can be seen in Fig. 6.3.

The results of such testing revealed that LCDBNs were much more invariant to random noise than their fully connected counterpart, though faired no better on the other corruption types - see Figures 6.4 and 6.5. It is not immediately clear why local connectivity should have this beneficial

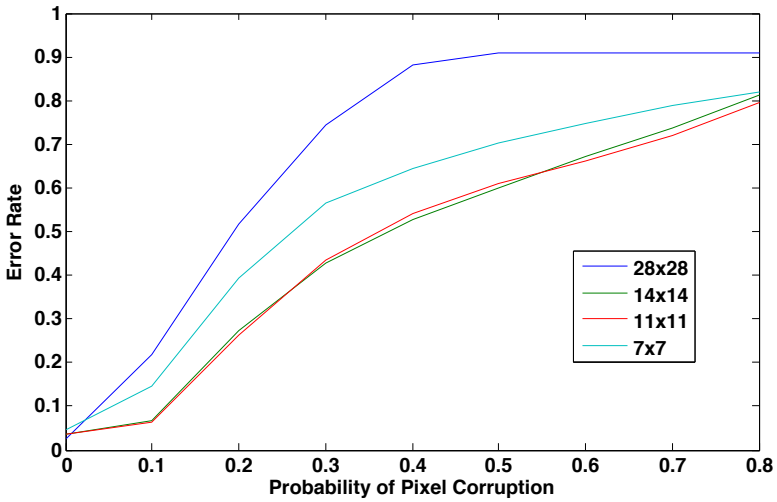


Figure 6.4: Error rates of DBN models with various receptive field sizes when tasked to classify MNIST images corrupted with increasing amounts of noise. Notice that while the fully connected model (image sized receptive fields - 28x28) very quickly degrades to random chance, the others do not.

effect on noisy inputs. Though a potential explanation may be a simple one: a local connection to the input layer limits the influence of the noise. In a standard DBN noise in the input area will modulate the response of all the units in the layer above. However in a LCDBN this harmful influence is contained within a small area.

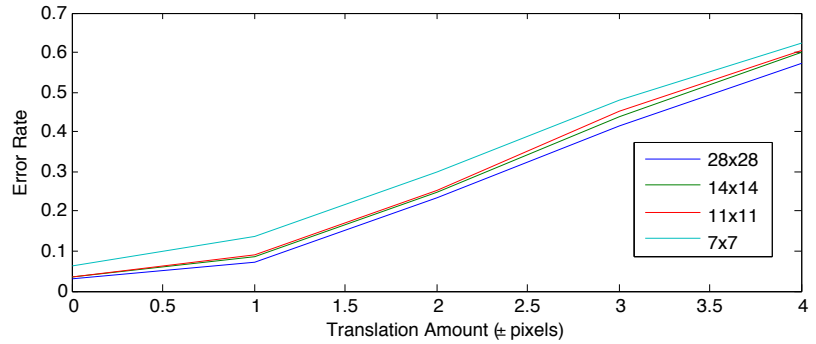
CIFAR-10 Orientation Maps

A two hidden layer LCDBN model with Gaussian input and binary hidden units was trained on the CIFAR-10[2] natural image patch dataset (Krizhevsky 2009).

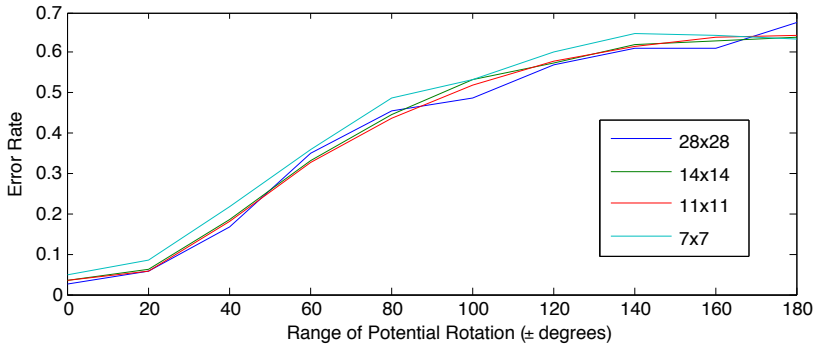
The network was defined to have a structure of 1024-4096-4096-10 (notation explained above in Sec. 6.4.1). While three sizes of receptive field sizes were tested (7, 11, and 14) only results for receptive field sizes of 11 are included here as the others were qualitatively similar. The discriminative ability of these models were after 100 epochs of contrastive divergence and a further 100 epochs of fine tuning by back propagation was relatively poor, only achieving a final error rate of 42% which far from comparable with the state-of-the-art (19.51% (Ciresan et al. 2011))[3], though in line with other DBN results (Krizhevsky 2009). The qualitative structure of the receptive fields however were interesting.

[2]available: http://www.cs.toronto.edu/~kriz/cifar.html

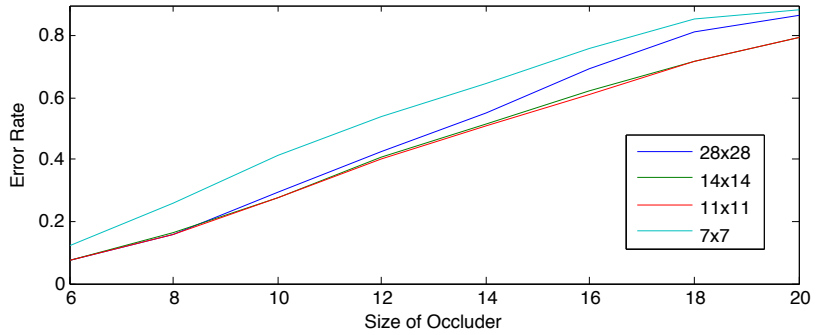
[3]at the time of this research this was the best reported result on CIFAR-10 and used traditional convolution networks.



(a)



(b)



(c)

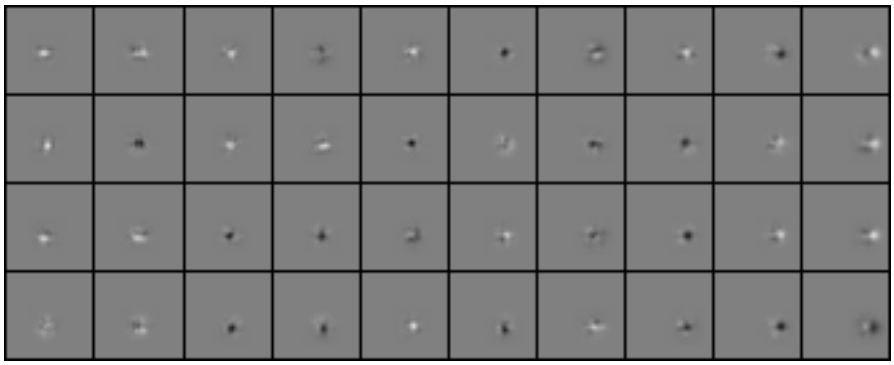
Figure 6.5: Error rates of LCDBN models with various receptive field sizes when tasked to classify MNIST images corrupted with translations (a), rotations (b) and occlusions (c). Unlike with noise corruption (Fig 6.4) in these cases local connectivity does not seem to offer any benefits.

The receptive fields of the first layer units resemble the classic on/off centre surround shape that characterises the responses of neurons in the LGN; the early layers of the visual system (Fig 6.7a). Interestingly, the second layer seems to be combining these into oriented edge detectors which is qualitatively similar to responses above the LGN, in visual area V1 (6.7b).

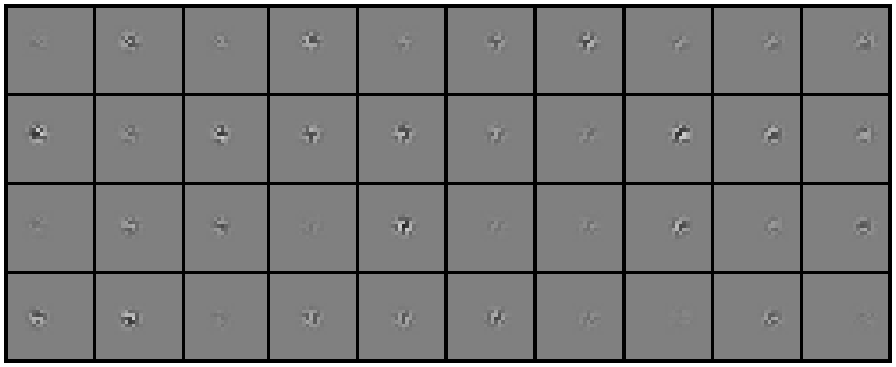
To attempt to characterise the extent of the orientation selectivity of the units in each layer a synthetic set of images comprising of grayscale two dimensional sine waves of varying orientation, frequency and phase was constructed. These images were presented to the LCDBN trained on the CIFAR-10 dataset and the responses of each layer are recorded. Of these recordings the maximal response across all frequencies and phases is calculated for each orientation which then allows the

calculation of the preferred orientation. This builds up a simple orientation map that after a small amount of blurring, we can use to visualise the differences in orientation selectivity in a natural way. Performing this procedure results in Fig. 6.8, where it is evident that the second layer exhibits a higher orientation selectivity than the first layer.

Figure 6.6: Example images drawn from four of the classes from the CIFAR-10 dataset. The rows correspond to the classes (from the top) *airplane*, *automobile*, *bird*, and *cat*. Each image is 32×32 in size.

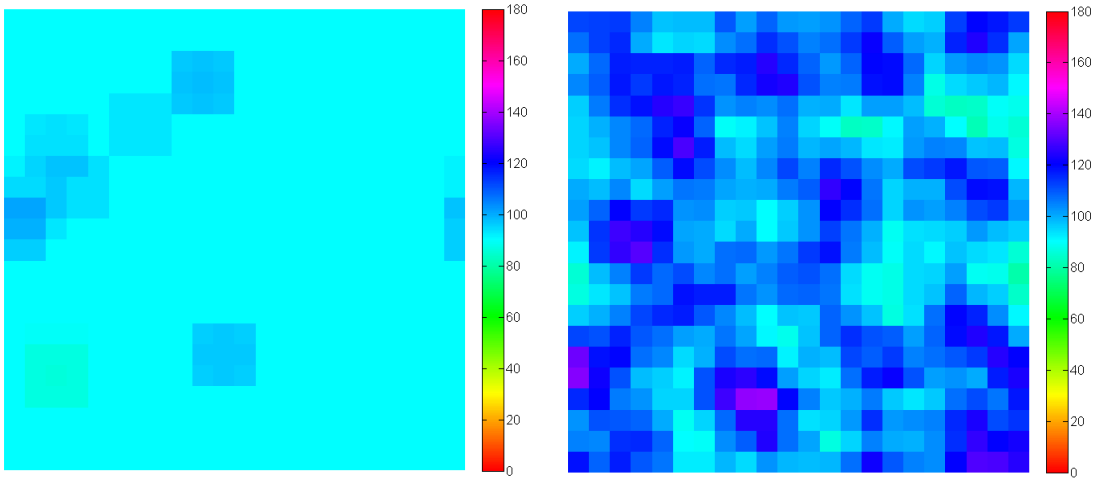


(a)



(b)

Figure 6.7: Some exemplar receptive fields of units in the first (a) and second layer (b) of a LCDBN trained on the CIFAR-10 dataset with 11x11 Gaussian receptive fields.



(a) (b)

Figure 6.8: Orientation maps computed for the first (a) and second (b) layers of a 11x11 sized receptive field LCDBN. Notice that the second layer is more selective for orientation as suggested by the receptive fields. This image is coloured according to angle selectivity.

Discussion

In summary, this work has shown that modifying the connectivity of DBNs to more closely resemble the topology of the visual cortex does not result in an improvement in discrimination ability both when using the features directly or after subsequent backpropagation fine tuning. This is confusing as Krizhevsky (2009) found that filters learnt directly from CIFAR-10 naturally tended to be very localised feature detectors - the model was converging to that of one with the apparent topology that is being imposed by the modifications in this very chapter. Furthermore the imposition of a local connectivity does not allow the model to generalise to unseen transformations of the input any better than the original DBN model. With the notable exception of corruption of the input data by random noise, where the local connectivity of the hidden layers isolates the effect of the noise.

When trained on natural images LCDBNs seem to learn receptive fields that are qualitatively similar to those found in the early layers of the visual cortex. While this was previously shown by Lee and Ekanadham (2007), that work only reported orientation selective receptive fields while using sparsity regularisation - which in their implementation was a process requiring global knowledge for each unit and hence is less biologically plausible.

Future Work

There are potential extensions of the work in this chapter to model further aspects of the visual cortex within the DBN framework.

- One of the most attractive reasons for attempting to create a more biologically plausible variant of a DBN is to utilise the natural recurrent connectivity afforded by the undirected nature of the model. This is slightly inaccurate since as stated earlier a DBN isn't actually a completely undirected model, but extending the local connectivity to a true undirected Deep Boltzmann Machine (Salakhutdinov and Hinton 2009) should be possible. Such a model

would be able to integrate top-down information (like attention modulation) when inferring the states of the intermediary layers. There is a potential issue here in that DBMs require doubling of the number of parameters during the CD learning phase which could pose a problem when working on high dimensional images (Salakhutdinov and Hinton 2009).

- In this work the receptive fields of each neuron is defined by a fixed grid of equal spacing. This is not the only way in which the local mapping between layers could be defined. An alternate configuration might be to simply randomly position the receptive fields of each neuron across the preceding layer, this would non-uniformly sample the preceding layer. Perhaps more interesting would be to have the positions of the receptive fields be a parameter of the network and learn an appropriate sampling based on the data. A straightforward way to accomplish this may be to set the position of the mask based upon the previous iteration unmasked activations.

- Even though RBMs and DBMs are all generative models this work does not make use of this fact and instead focuses on their use as a feature extraction hierarchy. Intuitively, generating samples from a locally connected DBNs poses a problem as the partitioned layer seemingly has no way to interact, given that explicit lateral interactivity is ruled out by the model structure. However this problem might be mitigated by the ability of the network to learn local descriptors

- Sparse neural coding has become a established theory as to one possible constraint upon the neural code. While there are already regularisation methods for RBMs that force the average activations of the hidden units across the dataset to be sparse (Lee and Ekanadham 2007) this is not guaranteed to induce sparse activations across a layer population. Rather this regularisation only affects the specificity of a given hidden unit to some input vector - by forcing the average activation to be low you are only allowing the neuron to be active for a small number of inputs. This is an important property of a neural code, but seemingly distinct from a sparse code for each exemplar. Furthermore the methods of Lee and Ekanadham (2007) utilise global averaging, which is a process that is beyond the capabilities of the visual cortex.

- It can be seen from the results in Fig. 6.5 that standard and locally connected DBNs are pretty poor at generalising to variations not found within the dataset. For example random translations of the MNIST images by only four pixels resulted in an error rate almost ten times higher. This is somewhat understandable as it is the goal of CD learning to try and best represent the underlying distributions present in the data. However a key feature of biological vision systems is that they are remarkably robust to simple input transforms such as rotation and translation. It seems that the features learned are somehow able to generalise to cover these states. How can something like this be built into the DBN framework? A simple idea might be to incorporate a set of transforms directly into the weight matrix itself. Though this is hard to justify from a biological standpoint. A potentially better idea might be to take a cue from VisNet (Rolls 2008c). This model accomplishes good invariance by utilising the local convergent connectivity (so neurons at the top of the hierarchy can gain information from the whole visual field) and a temporal trace learning rule. This rule seeks to correlate neurons that activate on sequential presentations of correlated data. This is an elegant way to make use of the temporal continuity exhibited in natural vision and in

principle could be adopted by a DBN network in a similar manner to VisNet by directly modulating the activities of units based on some weighted average of previous presentations.

Discussion & Conclusions

Discussion & Conclusions

Introduction

This thesis consists of two major parts. In the first part (Chapter 3) a comparison of two leading biologically relevant models of object recognition was undertaken to gauge how biologically plausible the representations that each model formed. The second part (Chapters 4, 5 and 6) focuses on approaches that do not attempt to explicitly model the biology and so touch upon numerous topics within machine vision.

Thesis Contributions

The major contributions of this thesis can be summarised as the following:

- When considering computational models of object recognition that are meant to be biologically relevant special care should be given to the nature of the representations that they produce. To emphasise this point a comparison of HMAX (Riesenhuber and Poggio 1999a) and VisNet (Wallis and Rolls 1997a) was undertaken. This work made numerous useful observations:

    1. HMAX fails to solve the core challenge of invariant object recognition - the construction of explicit invariant representations. This is in contrast to VisNet which does produce a representation that exhibits strong invariant properties (Experiment 4).

    2. HMAX does not exhibit sparse distributed representations; rather the representation is dense. (Experiment 2).

    3. VisNet is sensitive to the specific spatial locations of the features of a given object and does not respond to scrambled images. This is in contrast to HMAX which still correctly identifies the object after scrambling (Experiment 3)

- Extension of the Deep Belief Network (Hinton 2009; Hinton et al. 2006) model with a locally connected topology which results in individual neurons in the model having only a limited field of view, or receptive field akin to neurons found in the visual cortex. It is shown that the impact of imposing such a topology is minimal, with no effect on the ability of the model to generalise across transformations of the input stimuli. The tolerance of the model to noise is noted to increase.

- Extension of work by Goodfellow et al. (2015) demonstrating a simple gradient based optimisation process that creates adversarial exemplars that while visually indistinguisible from the original allows for the re-labelling of that exemplar to that of any ImageNet class. This is process is demonstrated to work on pre-trained instances of the popular AlexNet (Krizhevsky et al. 2012) and GoogLeNet (Szegedy et al. 2014) architectures.

- Method to apply the neural network regularization procedure dropout (Hinton et al. 2012; Srivastava et al. 2014) in a batchwise fashion that reduces the computational cost by approximately half without reducing the performance of the model on classification tasks with MNIST and CIFAR-10.

Discussion

Each section throughout this thesis has been somewhat distinct and self contained, so this final discussion section will briefly leave the reader with some closing thoughts regarding the future of computer vision models.

The interplay between computational models and neurophysiological evidence have made significant progress in highlighting the processes involved in how brains solve the complex task of recognising objects under unconstrained viewing conditions.

State-of-the-art object recognition utilising convolution networks on the ImageNet dataset (Russakovsky et al. 2014) (which has many thousands of object categories) has reached such a level that it can be realistically said to be approaching human (Krizhevsky et al. 2012; Szegedy et al. 2014) or perhaps even surpassing that of human performance (He et al. 2015) - at least in this limited setting. This is an impressive accomplishment, but it absolutely does not mean that vision has been "solved". Many algorithms which while not biologically plausible most certainly borrow heavily from the established body of neuroscience work have shown to work with complex, natural images. The best computer vision algorithms that have resulted in the increased performance of vision systems in the recent years have been ones that have pulled from computational neuroscience, machine learning and computer science.

Have these models really solved a fundamental aspect of vision? I propose that there are two fundamental reasons why we have seen a large surge in performance on standard datasets:

1. GPUs - Graphics processing units allow numerical algorithms that are highly data parallel to run substantially faster than has been possible previously, even with supercomputer scale computing. Convolution networks are an extremely good fit for such an architecture and so have exhibited substantial gains. It should be noted, that the current network architectures are not hugely different to that of 1990's era convolution networks (LeCun et al. 1990).

2. Image Datasets - huge datasets of labelled images have become available that are able to leverage supervised learning. Ever larger models can be constructed and effectively trained without model fitting problems as the datasets continue to grow. This is good for getting increased results on image dataset challenges, though it is less clear if this is helping solve the core vision problem.

The recent trend of quick performance gains will be difficult to continue as there has been relatively little progress on the core challenge of object recognition; which is the ability to build strongly invariant representations.

The current paradigm rests upon large scale supervised training with faster than before computational hardware. This is in complete contrast with the brain, where complex invariant representations seem to be built in a primarily unsupervised fashion with significantly less labelled exemplars.

Bibliography

Bibliography

L. F. Abbott, E. T. Rolls, and M. J. Tovee. Representational capacity of face coding in monkeys. *Cerebral Cortex*, 6:498–505, 1996.

N. C. Aggelopoulos and E. T. Rolls. Natural scene perception: inferior temporal cortex neurons encode the positions of different objects in the scene. *European Journal of Neuroscience*, 22: 2903–2916, 2005.

N. C. Aggelopoulos, L. Franco, and E. T. Rolls. Object perception in natural scenes: encoding by inferior temporal cortex simultaneously recorded neurons. *Journal of Neurophysiology*, 93: 1342–1357, 2005.

J. Alvarez, Y. LeCun, T. Gevers, and A. Lopez. Semantic road segmentation via multi-scale ensembles of learned features. *In ECCV Workshop on Computer Vision in Vehicle Technology: From Earth to Mars*, 2012.

A. Anzai, X. Peng, and D. C. van Essen. Neurons in monkey visual area v2 encode combinations of orientations. *Nature Neuroscience*, 10(10):1313–1321, 2007.

R. J. Baddeley, L. F. Abbott, M. J. A. Booth, F. Sengpiel, T. Freeman, E. A. Wakeman, and E. T. Rolls. Responses of neurons in primary and inferior temporal visual cortices to natural scenes. *Proceedings of the Royal Society B*, 264:1775–1783, 1997.

J.S. Baizer, L.G. Ungerleider, and R. Desimone. Organization of inputs to the inferior temporal and posterior parietal cortex in macaques. *Journal of Neuroscience*, 11:168–190, 1991.

D. Balduzzi, H. Vanchinathan, and J. Buhmann. Kickback cuts backprop's red-tape: Biologically plausible credit assignment in neural networks. *arXiv preprint arXiv:1411.6191*, 2014.

D. H. Ballard. Animate vision uses object-centred reference frames. In R. Eckmiller, editor, *Advanced Neural Computers*, pages 229–236. North-Holland, Elsevier, Amsterdam, 1990.

G. C. Baylis, E. T. Rolls, and C. M. Leonard. Selectivity between faces in the responses of a population of neurons in the cortex in the superior temporal sulcus of the monkey. *Brain Research*, 342:91–102, 1985.

Y Bengio and Yann LeCun. Scaling learning algorithms towards AI. *Large-Scale Kernel Machines*, 2007.

Y Bengio, P Lamblin, D Popovici, and H Larochelle. Greedy Layer Wise Training of Deep Networks. *Advances in Neural Information Processing Systems*, 2007.

P Berkes and L Wiskott. Slow Feature Analysis Yields a Rich Repertoire of Complex Cell Properties. *Journal of vision*, 2005.

I. Biederman. Human image understanding - recent research and a theory. *Computer Vision Graphics and Image Processing*, 32(1):29–73, 1985.

I Biederman. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 1987.

I Biederman. Recognizing Depth-Rotated Objects: A Review of Recent Research and Theory. *Spatial Vision*, 2000.

T. Binford. Inferring surfaces from images. *Artificial Intelligence*, 17(13):205–224, 1981.

M. C. A. Booth and E. T. Rolls. View-invariant representations of familiar objects by neurons in the inferior temporal visual cortex. *Cerebral Cortex*, 8:510–523, 1998.

D. Boussaoud, R. Desimone, and L. G. Ungerleider. Visual topography of area teo in the macaque. *Journal of Comparative Neurology*, 306(4), 1991.

M. Brady, J. Ponce, A. Yuille, and H. Asada. Describing surfaces. *Computer Vision, Graphics, and Image Processing*, 21(1):1–28, 1985.

J. Buhmann, J. Lange, C. von der Malsburg, J. C. Vorbrüggen, and R. P. Würtz. Object recognition in the dynamic link architecture: Parallel implementation of a transputer network. In B. Kosko, editor, *Neural Networks for Signal Processing*, pages 121–159. Prentice Hall, Englewood Cliffs, NJ, 1991.

C. F. Cadieu, H. Hong, D. Yamins, N. Pinto, N. J. Majaj, and J. J. DiCarlo. The neural representation benchmark and its evaluation on brain and machine. *arXiv preprint arXiv:1301.3530*, 2013.

C. F. Cadieu, H. Hong, D. L. K. Yamins, N. Pinto, D. Ardila, E. A. Solomon, N. J. Majaj, and J. J. DiCarlo. Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS Computational Biology*, 10(12):e1003963, 2014.

Edward M Callaway. Feedforward, Feedback and Inhibitory Connections in Primate Visual Cortex. *Neural Networks*, 2004.

Matteo Carandini, Jonathan B Demb, Valerio Mante, David J Tolhurst, Yang Dan, B A Olshausen, Jack L Gallant, and Nicole C Rust. Do We Know What the Early Visual System Does? *The Journal of Neuroscience*, 2005.

D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012. URL www.idsia.ch/~juergen/cvpr2012.pd.

Dan C. Ciresan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, and JüÃijrgen Schmidhuber. High-Performance Neural Networks for Visual Object Classification. *CoRR*, abs/1102.0183, 2011. URL http://arxiv.org/abs/1102.0183.

T. S. Cohen and M. Welling. Transformation properties of learned visual representations. *ICLR*, 2015.

G. W. Cottrell and J. H. Hsaio. Neurocomputational models of face processing. In A. J. Calder, G. Rhodes, M. H. Johnson, and J. V. Haxby, editors, *The Oxford Handbook of Face Perception*, chapter 21, pages 402–423. Oxford University Press, Oxford, 2011.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.177. URL http://dx.doi.org/10.1109/CVPR.2005.177.

C. Dane and R. Bajcsy. An object-centered three-dimensional model builder. *Proceedings of the 6th International Conference on Pattern Recognition*, pages 348–350, 1982.

J. Daugman. Complete discrete 2D-Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, 36:1169–1179, 1988a.

J G Daugman. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1988b.

R. L. De Valois and K. K De Valois. *Spatial Vision*. Oxford University Press, New York, 1988.

G. Deco and E. T. Rolls. A neurodynamical cortical model of visual attention and invariant object recognition. *Vision Research*, 44:621–644, 2004.

M. Denil, L. Bazzani, H. Larochelle, and N. de Freitas. Learning where to attend with deep architectures for image tracking. *Neural Computation*, 24:2151–2184, 2012.

R Desimone. Face-Selective Cells in the Temporal Cortex of Monkeys. *Journal of Cognitive Neuroscience*, 1991.

R. Desimone and S. J. Schein. Visual properties of neurons in area V4 of the macaque: Sensitivity to stimulus form. *Journal of Neuroscience*, 57:835–868, 1987.

M. R. DeWeese and M. Meister. How to measure the information gained from one symbol. *Network*, 10:325–340, 1999.

J. J. DiCarlo, D. Zoccolan, and N. C. Rust. How does the brain solve visual object recognition? *Neuron*, 73:415–434, 2012.

S. Dieleman, K. W. Willett, and J. Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *CoRR*, abs/1503.07077, 2015. URL http://arxiv.org/abs/1503.07077.

R. Dubner and S. Zeki. Response properties and receptive fields of cells in an anatomically defined region of superior temporal sulcus in monkey. *Brain Research*, 35(2), 1971.

S. Edelman and H. Bülthoff. Orientation dependence in the recognition of familiar and novel views of 3-dimensional objects. *Vision Research*, 32(12):2385âĂŞ2400, 1992.

W. Einhauser, J. Eggert, E. Korner, and P. Konig. Learning viewpoint invariant object representations using a temporal coherence principle. *Biological Cybernetics*, 93:79–90, 2005.

M. C. M. Elliffe, E. T. Rolls, and S. M. Stringer. Invariant recognition of feature combinations in the visual system. *Biological Cybernetics*, 86:59–71, 2002.

Daniel J Felleman and David C Van Essen. Distributed Hierarchical Processing in the Primate Cerebral Cortex. *Cerebral Cortex*, 1991.

P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3:193–199, 1991.

P. Földiák. Models of sensory coding. Technical Report CUED/F–INFENG/TR 91, University of Cambridge, Department of Engineering, Cambridge, 1992.

L. Franco, E. T. Rolls, N. C. Aggelopoulos, and A. Treves. The use of decoding to analyze the contribution to the information of the correlations between the firing of simultaneously recorded neurons. *Experimental Brain Research*, 155:370–384, 2004.

L. Franco, E. T. Rolls, N. C. Aggelopoulos, and J. M. Jerez. Neuronal selectivity, population sparseness, and ergodicity in the inferior temporal visual cortex. *Biological Cybernetics*, 96: 547–560, 2007.

M. Franzius, H. Sprekeler, and L. Wiskott. Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS Computational Biology*, 3(8):e166, 2007.

M. Franzius, N. Wilbert, and L. Wiskott. Invariant object recognition with slow feature analysis. *Artificial Neural Networks*, 5163:961–970, 2008.

K Fukushima. Cognitron: A Self-Organizing Multilayered Neural Network. *Biological Cybernetics*, 1975.

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

J. Garthwaite. Concepts of neural nitric oxide-mediated transmission. *European Journal of Neuroscience*, 27:2783–3802, 2008.

J.-M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The Amsterdam Library of Object Images. *International Journal of Computer Vision*, 61:103–112, 2005.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

R. Goroshin, M. Mathieu, and Y. Lecun. Learning to linearize under uncertainty. *Advances in Neural Information Processing Systems*, 2015.

Ben Graham. Fractional max-pooling. *CoRR*, abs/1412.6071, 2014. URL http://arxiv.org/abs/1412.6071.

G. Griffin, A. Holub, and P. Perona. The Caltech-256. *Caltech Technical Report*, pages 1–20, 2007.

C Gross, C Rocha-Miranda, and D Bender. Visual properties of neurons in inferotemporal cortex of the Macaque. *Journal of Neurophysiology*, 1972.

C Gross, H Rodman, P Gochin, and M Colombo. *Inferior Temporal Cortex as a Pattern Recognition Device*. 1993.

M. E. Hasselmo, E. T. Rolls, G. C. Baylis, and V. Nalwa. Object-centered encoding by face-selective neurons in the cortex in the superior temporal sulcus of the monkey. *Experimental Brain Research*, 75:417–429, 1989.

M. J. Hawken and A. J. Parker. Spatial properties of the monkey striate cortex. *Proceedings of the Royal Society, London B*, 231:251–288, 1987.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *CoRR*, abs/1502.01852, 2015. URL http://arxiv.org/abs/1502.01852.

S. Hestrin, P. Sah, and R. Nicoll. Mechanisms generating the time course of dual component excitatory synaptic currents recorded in hippocampal slices. *Neuron*, 5:247–253, 1990.

G E Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002.

G E Hinton. Deep Belief Nets. In *Machine Learning Summer School*, 2009.

G E Hinton and R Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science (New York, N.Y.)*, 2006.

G E Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 2006.

G E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. URL http://arxiv.org/abs/1207.0580.

GB Huang, MA Amherst, and H. Lee. Learning Hierarchical Representations for Face Verification with Convolutional Deep Belief Networks. *Journal of Machine Learning Research*, 2012.

D H Hubel and T N Wiesel. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. *The Journal of Physiology*, 1962.

D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology, London*, 195:215–243, 1968a.

D H Hubel and T N Wiesel. Receptive Fields and Functional Architecture of Monkey Striate Cortex. *The Journal of Physiology*, 1968b.

J. E. Hummel and I. Biederman. Dynamic binding in a neural network for shape recognition. *Psychological Review*, 99:480–517, 1992.

L. Isik, J. Z. Leibo, and T. Poggio. Learning and disrupting invariance in visual recognition with a temporal association rule. *Frontiers in Computational Neuroscience*, 6:37, 2012.

E. B. Issa and J. J. DiCarlo. Precedence of the eye region in neural processing of faces. *The Journal of Neuroscience*, 32:16666–16682, 2012.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093, 2014. URL http://arxiv.org/abs/1408.5093.

C. Kanan. Active object recognition with a space-variant retina. *International Scholarly Research Notices Machine Vision*, page 138057, 2013.

C. Kanan and G. W. Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2472–2479. IEEE, 2010.

A. Kanazawa, A. Sharman, and D. Jacobs. Locally scale-invariant convolutional neural networks. *Advances in Neural Information Processing Systems*, 2014.

R. P. Kesner and E. T. Rolls. A computational theory of hippocampal function, and tests of the theory. new developments. *Neuroscience and Biobehavioral Reviews*, 48:92–147, 2015.

S-M. Khaligh-Razavi and N. Kriegeskorte. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS Computational Biology*, 10(11):e1003915, 2014.

J. Kivinen and C. Williams. Transformation equivariant boltzmann machines. In *Artificial Neural Networks and Machine Learning*, pages 1–9. Springer, 2011.

J Knierim and David C Van Essen. ï£ijï£ijVisual Cortex: Cartography, Connectivity, and Concurrent Processing. *Current Opinion in Neurobiology*, 1992.

E Kobatake and K Tanaka. Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex. *Journal of Neurophysiology*, 1994.

J. J. Koenderink. *Solid Shape*. MIT Press, Cambridge, MA, 1990.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *CoRR*, abs/1404.5997, 2014. URL http://arxiv.org/abs/1404.5997.

H. Larochelle and G. E. Hinton. Learning to combine foveal glimpses with a third-order Boltzmann machine. *Advances in Neural Information Processing Systems (NIPS)*, 1:1243–1251, 2010.

Q. Le, J. Ngiam, Z. Chen, D. Chia, P Koh, and A. Ng. Tiled convolutional neural networks. *Advances in Neural Information Processing Systems*, 2010.

Y. LeCun, B Boser, J. S. Denker, D. Henderson, R. E Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 1989.

Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. *2010 IEEE International Symposium on Circuits and Systems*, pages 253–256, 2010.

Yann LeCun, B Boser, J Denker, D Henderson, R Howard, W Hubbard, and L Jackel. Handwritten Digit Recognition with a Back-Propagation Network. *Advances in Neural Information Processing Systems*, 1990.

Yann Lecun, LÃľon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

H. Lee and C Ekanadham. Sparse deep belief net model for visual area V2. *Advances in Neural Information Processing Systems*, 2007.

H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009a.

Honglak Lee, Y Largman, P Pham, and A.Y. Ng. ï£ijUnsupervised Feature Learning for Audio Classification Using Convolutional Deep Belief Networks. *Advances in Neural Information Processing Systems*, 2009b.

Honglak Lee, Roger Grosse, R. Ranganath, and Andrew Y Ng. Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks. *Communications of the ACM*, 2011.

T. S. Lee. Image representation using 2D Gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18,10:959–971, 1996.

K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. *CVPR*, 2015.

S. LeVay, T. N. Wiesel, and D. H. Hubel. The development of ocular dominance columns in normal and visually deprived monkeys. *Journal of Comparative Neurology*, 191:1–51, 1980.

N. Li and J. J. DiCarlo. Unsupervised natural experience rapidly alters invariant object representation in visual cortex. *Science*, 321:1502–1507, 2008.

N. Li and J. J. DiCarlo. Unsupervised natural visual experience rapidly reshapes size-invariant object representation in inferior temporal cortex. *Neuron*, 67:1062–1075, 2010.

N. Li and J. J. DiCarlo. Neuronal learning of invariant object representation in the ventral visual stream is not dependent on reward. *Journal of Neuroscience*, 32:6611–6620, 2012.

J-P. Lies, R. M. Häfner, and M. Bethge. Slowness and sparseness have diverging effects on complex cell learning. *PLoS Computational Biology*, 10(3):e1003468, 2014.

M. Livingstone and D. Hubel. Segregation of form, colour, movement, and depth: Anatomy, physiology, and perception. *Science*, 240:740–749, 1988.

M. S. Livingstone and D. H. Hubel. Thalamic inputs to cytochrome oxidase-rich regions in monkey visual-cortex. *Proceedings of the National Academy of Sciences USA*, 79:6098–6101, 1982.

M. S. Livingstone and D. H. Hubel. Connections between layer 4b of area-17 and the thick cytochrome-oxidase stripes of area-18 in the squirrel-monkey. *Journal of Neuroscience*, 7(11): 3371–3377, 1987.

N. K. Logothetis and J. Pauls. Psychophysical and physiological evidence for viewer-centered object representations in the primate. *Cerebral Cortex*, 5(3):270–88, 1995.

N. K. Logothetis, J. Pauls, H. Bulthoff, , and T. Poggio. View-dependent object recognition by monkeys. *Current Biology*, 4(5):401–414, 1994.

N. K. Logothetis, J. Pauls, and T. Poggio. Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, 5:552–563, 1995.

D G Lowe. Distinctive Image Features From Scale-Invariant Keypoints. *International Journal of Computer Vision*, 2004.

C. von der Malsburg. Self-organization of orientation-sensitive columns in the striate cortex. *Kybernetik*, 14:85–100, 1973.

D. Marr. *Vision*. Freeman, San Francisco, 1982.

D Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. In *Proc. of the Royal Society of London, series B*, volume 200, pages 269–294, February 1978.

J. H. R. Maunsell and W. T. Newsome. Visual processing in monkey extrastriate cortex. *Annual Review of Neuroscience*, 10:363–401, 1987.

Y. Miyashita. Neuronal correlate of visual associative long-term memory in the primate temporal cortex. *Nature*, 335:817–820, 1988.

Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep Learning from Temporal Coherence in Video. *Proceedings of the 26th International Conference on Machine Learning*, 2009.

P. R. Montague, J. A. Gally, and G. M. Edelman. Spatial signalling in the development and function of neural connections. *Cerebral Cortex*, 1:199–220, 1991.

J. Moran and R. Desimone. Selective attention gates visual processing in the extrastriate cortex. *Science*, 229:782–784, 1985.

J. Mutch and D. G. Lowe. Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision*, 80:45–57, 2008.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. *arXiv.org*, 2014.

M Norouzi, M Ranjbar, and G Mori. Stacks of Convolutional Restricted Boltzmann Machines for Shift-Invariant Feature Learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2009.

E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273, 1982.

J. O'Reilly and Y. Munakata. *Computational Explorations in Cognitive Neuroscience*. MIT Press, Cambridge, MA, 2000.

S. Panzeri, A. Treves, S. Schultz, and E. T. Rolls. On decoding the responses of a population of neurons from short time epochs. *Neural Computation*, 11:1553–1577, 1999.

A. P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331, 1986.

D I Perrett and M W Oram. Neurophysiology of shape processing. *Image and Vision Computing*, 1993.

D. I. Perrett, E. T. Rolls, and W. Caan. Visual neurons responsive to faces in the monkey temporal cortex. *Experimental Brain Research*, 47:329–342, 1982.

D. I. Perrett, M. W. Oram, M. H. Harries, R. Bevan, J. K. Hietanen, and P. J. Benson. Viewer–centered and object centered coding of heads in the macaque temporal cortex. *Experimental Brain Research*, 86:159–173, 1991.

G. Perry, E. T. Rolls, and S. M. Stringer. Spatial vs temporal continuity in view invariant visual object recognition learning. *Vision Research*, 46:3994–4006, 2006.

G. Perry, E. T. Rolls, and S. M. Stringer. Continuous transformation learning of translation invariant representations. *Experimental Brain Research*, 204:255–270, 2010.

E. Peterhans and R. von der Heydt. Mechanisms of contour perception in monkey visual cortex ii: contours bridging gaps. *Journal of Neuroscience*, 9:1749–1763, 1989.

N Pinto, David D Cox, and James J DiCarlo. Why is Real-World Visual Object Recognition Hard? *PLoS Computational Biology*, 2008.

N. Pinto, D. Doukhan, J. J. DiCarlo, and D. D. Cox. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS Computational Biology*, 5:e1000579, 2009.

D. Pollen and S. Ronner. Phase relationship between adjacent simple cells in the visual cortex. *Science*, 212:1409–1411, 1981.

M. Reid, L. Spirkovska, and E. Ochoa. Simultaneous position, scale, and rotation invariant pattern classification using third-order neural networks. *International Journal of Neural Networks âĂŞ Research & Applications*, 1(3):154–159, 1989.

J. Reynolds, L. Chelazzi, and R. Desimone. Competitive mechanisms subserve attention in macaque areas V2 and V4. *Journal of Neuroscience*, 19:1736–1753, 1999.

P. Rhodes. The open time of the NMDA channel facilitates the self-organisation of invariant object responses in cortex. *Society for Neuroscience Abstracts*, 18:740, 1992.

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

M. Riesenhuber and T. Poggio. Models of object recognition. *Nature Neuroscience Supplement*, 3: 1199–1204, 2000a.

Maximilian Riesenhuber and Tomaso Poggio. Hierarchical Models of Object Recognition in Cortex. *Nature Neuroscience*, 1999b.

Maximilian Riesenhuber and Tomaso Poggio. Models of Object Recognition. *Nature Neuroscience*, 2000b.

Oren Rippel, Michael A. Gelbart, and Ryan P. Adams. Recurrent neural network regularization. *CoRR*, abs/1402.0915, 2014. URL http://arxiv.org/abs/1402.0915.

I. Rock and J. Divita. A case of viewer-centered object perception. *Cognitive Psychology*, 19(2): 280–293, 1987.

I. Rock, J. DiVita, and Barbeitom R. The effect on form perception of change of orientation in the third dimension. *Journal of Experimental Psychology*, 19, 1981.

E. T. Rolls. Functions of neuronal networks in the hippocampus and neocortex in memory. In J. H. Byrne and W. O. Berry, editors, *Neural Models of Plasticity: Experimental and Theoretical Approaches*, chapter 13, pages 240–265. Academic Press, San Diego, CA, 1989.

E T Rolls. Neural Organization of Higher Visual Functions. *Current Opinion in Neurobiology*, 1991.

E. T. Rolls. Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical visual areas. *Philosophical Transactions of the Royal Society*, 335:11–21, 1992.

E. T. Rolls. Learning mechanisms in the temporal lobe visual cortex. *Behavioural Brain Research*, 66:177–185, 1995.

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

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

E. T. Rolls. *Memory, Attention, and Decision-Making. A Unifying Computational Neuroscience Approach*. Oxford University Press, Oxford, 2008a.

E T Rolls. *Memory, Attention, and Decision-Making*. Oxford University Press, 2008b.

E T Rolls. Invariant Visual Object Recognition Learning. In *Memory, Attention, and Decision-Making*. 2008c.

E. T. Rolls. Face neurons. In A. J. Calder, G. Rhodes, M. H. Johnson, and J. V. Haxby, editors, *The Oxford Handbook of Face Perception*, chapter 4, pages 51–75. Oxford University Press, Oxford, 2011.

E. T. Rolls. *Neuroculture: On the Implications of Brain Science*. Oxford University Press, Oxford, 2012a.

E. T. Rolls. Invariant visual object and face recognition: neural and computational bases, and a model, VisNet. *Frontiers in Computational Neuroscience*, 6(35):1–70, 2012b.

E. T. Rolls. *Emotion and Decision-Making Explained*. Oxford University Press, Oxford, 2014.

E. T. Rolls. Diluted connectivity in pattern association networks facilitates the recall of information from the hippocampus to the neocortex. *Progress in Brain Research*, 219:21–43, 2015.

E. T. Rolls. *Cerebral Cortex: Principles of Operation*. Oxford University Press, Oxford, 2016.

E. T. Rolls and G. C. Baylis. Size and contrast have only small effects on the responses to faces of neurons in the cortex of the superior temporal sulcus of the monkey. *Experimental Brain Research*, 65:38–48, 1986.

E. T. Rolls and G. Deco. *Computational Neuroscience of Vision*. Oxford University Press, Oxford, 2002.

E. T. Rolls and T. Milward. A model of invariant object recognition in the visual system: learning rules, activation functions, lateral inhibition, and information-based performance measures. *Neural Computation*, 12:2547–2572, 2000.

E. T. Rolls and S. M. Stringer. Invariant object recognition in the visual system with error correction and temporal difference learning. *Network: Computation in Neural Systems*, 12: 111–129, 2001.

E. T. Rolls and S. M. Stringer. Invariant visual object recognition: a model, with lighting invariance. *Journal of Physiology – Paris*, 100:43–62, 2006.

E. T. Rolls and S. M. Stringer. Invariant global motion recognition in the dorsal visual system: a unifying theory. *Neural Computation*, 19:139–169, 2007.

E. T. Rolls and M. J. Tovee. Processing speed in the cerebral cortex and the neurophysiology of visual masking. *Proceedings of the Royal Society, B*, 257:9–15, 1994.

E. T. Rolls and M. J. Tovee. Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *Journal of Neurophysiology*, 73:713–726, 1995.

E. T. Rolls and A. Treves. *Neural Networks and Brain Function*. Oxford University Press, Oxford, 1998.

E. T. Rolls and A. Treves. The neuronal encoding of information in the brain. *Progress in Neurobiology*, 95:448–490, 2011.

E. T. Rolls and T. J. Webb. Finding and recognising objects in natural scenes: complementary computations in the dorsal and ventral visual systems. *Frontiers in Computational Neuroscience*, 8:85, 2014.

E. T. Rolls, G. C. Baylis, and C. M. Leonard. Role of low and high spatial frequencies in the face-selective responses of neurons in the cortex in the superior temporal sulcus. *Vision Research*, 25:1021–1035, 1985.

E. T. Rolls, G. C. Baylis, and M. E. Hasselmo. The responses of neurons in the cortex in the superior temporal sulcus of the monkey to band-pass spatial frequency filtered faces. *Vision Research*, 27:311–326, 1987.

E. T. Rolls, G. C. Baylis, M. Hasselmo, and V. Nalwa. The representation of information in the temporal lobe visual cortical areas of macaque monkeys. In J.J. Kulikowski, C.M. Dickinson, and I.J. Murray, editors, *Seeing Contour and Colour*. Pergamon, Oxford, 1989.

E. T. Rolls, M. J. Tovee, D. G. Purcell, A. L. Stewart, and P. Azzopardi. The responses of neurons in the temporal cortex of primates, and face identification and detection. *Experimental Brain Research*, 101:474–484, 1994.

E. T. Rolls, A. Treves, M. Tovee, and S. Panzeri. Information in the neuronal representation of individual stimuli in the primate temporal visual cortex. *Journal of Computational Neuroscience*, 4:309–333, 1997a.

E. T. Rolls, A. Treves, and M. J. Tovee. The representational capacity of the distributed encoding of information provided by populations of neurons in the primate temporal visual cortex. *Experimental Brain Research*, 114:149–162, 1997b.

E. T. Rolls, N. C. Aggelopoulos, and F. Zheng. The receptive fields of inferior temporal cortex neurons in natural scenes. *Journal of Neuroscience*, 23:339–348, 2003.

E. T. Rolls, N. C. Aggelopoulos, L. Franco, and A. Treves. Information encoding in the inferior temporal visual cortex: contributions of the firing rates and the correlations between the firing of neurons. *Biological Cybernetics*, 90:19–32, 2004.

E. T. Rolls, L. Franco, N. C. Aggelopoulos, and J. M. Jerez. Information in the first spike, the order of spikes, and the number of spikes provided by neurons in the inferior temporal visual cortex. *Vision Research*, 46:4193–4205, 2006.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

O. Russakovsky, J. Deng, Su H., J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. URL http://arxiv.org/abs/1409.0575.

R Salakhutdinov and G E Hinton. Semantic Hashing. *International Journal of Approximate Reasoning*, 2008.

R Salakhutdinov and G E Hinton. Deep Boltzmann Machines. In *Proceedings of the 9th International Conference on Artificial Intelligence and Statistics*, 2009.

R Salakhutdinov and H Larochelle. Efficient Learning of Deep Boltzmann Machines. *Proceedings of the 9th International Conference on Artificial Intelligence and Statistics*, 2010.

P. Schiller. Area-v4 of the primate visual cortex. *Current Directions in Psychological Science*, 3 (3):89–92, 1994.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. *CoRR*, abs/1503.03832, 2015. URL http://arxiv.org/abs/1503.03832.

T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, and T. Poggio. A quantitative theory of immediate visual recognition. *Progress in Brain Research*, 165:33–56, 2007a.

T. Serre, A. Oliva, and T. Poggio. A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences*, 104:6424–6429, 2007b.

T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29: 411–426, 2007c.

Thomas Serre, M Kouh, C Cadieu, U Knoblich, G Kreiman, and Tomaso Poggio. A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex. *MIT: Computer Science and Artificial Intelligence Laboratory Technical Report*, 2005.

Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust Object Recognition with Cortex-Like Mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007d.

K. Sohn and H. Lee. Learning invariant representations with local transformations. *CoRR*, abs/1206.6418, 2012. URL http://arxiv.org/abs/1206.6418.

N. Spruston, P. Jonas, and B. Sakmann. Dendritic glutamate receptor channel in rat hippocampal CA3 and CA1 pyramidal neurons. *Journal of Physiology*, 482:325–352, 1995.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

S. M. Stringer and E. T. Rolls. Position invariant recognition in the visual system with cluttered environments. *Neural Networks*, 13:305–315, 2000.

S. M. Stringer and E. T. Rolls. Invariant object recognition in the visual system with novel views of 3D objects. *Neural Computation*, 14:2585–2596, 2002.

S. M. Stringer and E. T. Rolls. Learning transform invariant object recognition in the visual system with multiple stimuli present during training. *Neural Networks*, 21:888–903, 2008.

S. M. Stringer, G. Perry, E. T. Rolls, and J. H. Proske. Learning invariant object recognition in the visual system with continuous transformations. *Biological Cybernetics*, 94:128–142, 2006.

S. M. Stringer, E. T. Rolls, and J. M. Tromans. Invariant object recognition with trace learning and multiple stimuli present during training. *Network: Computation in Neural Systems*, 18: 161–187, 2007.

Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning.*, volume 28 of *JMLR Proceedings*, pages 1139–1147. JMLR.org, 2013. URL http://dblp.uni-trier.de/db/conf/icml/icml2013.html#SutskeverMDH13.

R. S. Sutton and A. G. Barto. Towards a modern theory of adaptive networks: expectation and prediction. *Psychological Review*, 88:135–170, 1981.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intruguing properties of neural networks. *CoRR*, abs/1312.6199:1–10, 2014. URL http://arxiv.org/abs/1412.6071.

Y Taigman, M Yang, Marc'Aurelio Ranzato, and L Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014.

S.A. Talbot and W.H. Marshall. Physiological studies on neurophysiological mechanisms of visual localization and discrimination. *American Journal of Opthalmology*, 24:1225–1264, 1941.

K Tanaka. Inferotemporal Cortex and Object Vision. *Annual Review of Neuroscience*, 1996.

Yichuan Tang and Adbel-rahman Mohamed. Multiresolution deep belief networks. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, 2012.

M. Tarr. Rotating objects to recognize them - a case study on the role of viewpoint dependency in the recognition of 3-dimensional objects. *Psychonomic Bulletin & Review*, 2(1):55âĂŞ82, 1995.

M. Tarr, P. Williams, W. Hayward, , and I. Gauthier. Three-dimensional object recognition is viewpoint dependent. *Nature Neuroscience*, 1(4), 1998.

Andrew F Teich and Ning Qian. Comparison Among Some Models of Orientation Selectivity. *Journal of Neurophysiology*, 2006.

J. Tou and A. Gonzalez. *Pattern Recognition Principles.* Addison-Wesley., 1974.

M. J. Tovee and E. T. Rolls. Information encoding in short firing rate epochs by single neurons in the primate temporal visual cortex. *Visual Cognition*, 2:35–58, 1995.

M. J. Tovee, E. T. Rolls, A. Treves, and R. P. Bellis. Information encoding and the responses of single neurons in the primate temporal visual cortex. *Journal of Neurophysiology*, 70:640–654, 1993.

M. J. Tovee, E. T. Rolls, and P. Azzopardi. Translation invariance and the responses of neurons in the temporal visual cortical areas of primates. *Journal of Neurophysiology*, 72:1049–1060, 1994.

T. P. Trappenberg, E. T. Rolls, and S. M. Stringer. Effective size of receptive fields of inferior temporal visual cortex neurons in natural scenes. In T. G. Dietterich, S. Becker, and Z. Gharamani, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 293–300. MIT Press, Cambridge, MA, 2002.

A. Treves and E. T. Rolls. A computational analysis of the role of the hippocampus in memory. *Hippocampus*, 4:374–391, 1994.

A. Treves, S. Panzeri, E. T. Rolls, M. Booth, and E. A. Wakeman. Firing rate distributions and efficiency of information transmission of inferior temporal cortex neurons to natural visual stimuli. *Neural Computation*, 11:601–631, 1999.

S. Ullman. *High-Level Vision.* Cambridge, MA: The MIT Press, 1996.

L G Ungerleider and J V Haxby. 'What' and 'Where' in the Human Brain. *Current Opinion in Neurobiology*, 1994.

L. G. Ungerleider and M. Mishkin. Two cortical visual systems. In D. J. Ingle, editor, *Analysis of Visual Behavior*, pages 549–586. MIT Press, Cambridge, 1982.

T. Valentine. Upside-down faces: a review of the effect of inversion upon face recognition. *British Journal of Psychology*, 79:471–491, 1988.

D. Van Essen, C. H. Anderson, and D. J. Felleman. Information processing in the primate visual system: an integrated systems perspective. *Science*, 255:419–423, 1992a.

David C Van Essen, Charles H Anderson, and Daniel J Felleman. Information Processing in the Primate Visual System: An Integrated Systems Perspective. *Science, New Series*, 1992b.

P Viola and M Jones. Robust Real-Time Object Detection. *International Journal of Computer Vision*, 2002.

G. Wallis. Toward a unified model of face and object recognition in the human visual system. *Frontiers in Psychology*, 4:497, 2013.

G. Wallis and E. T. Rolls. Invariant face and object recognition in the visual system. *Progress in Neurobiology*, 51:167–194, 1997a.

G Wallis and E T Rolls. Invariant Face and Object Recognition in the Visual System. *Progress in Neurobiology*, 1997b.

G. Wallis, E. T. Rolls, and P. Földiák. Learning invariant responses to the natural transformations of objects. *International Joint Conference on Neural Networks*, 2:1087–1090, 1993.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Lecun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *JMLR Proceedings*, pages 1058–1066. JMLR.org, 2013. URL http://jmlr.org/proceedings/papers/v28/wan13.html.

Sida Wang and Christopher Manning. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *JMLR Proceedings*, pages 118–126. JMLR.org, 2013. URL http://jmlr.csail.mit.edu/proceedings/papers/v28/wang13a.html.

T. J. Webb and E. T. Rolls. Deformation-specific and deformation-invariant visual object recognition: pose vs identity recognition of people and deforming objects. *Frontiers in Computational Neuroscience*, 8:37, 2014.

M Welling, M Rosen-Zvi, and G E Hinton. Exponential Family Harmoniums with an Application to Information Retrieval. *Advances in Neural Information Processing Systems*, 2005.

P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* PhD thesis, Harvard University, 1974.

T. N. Wiesel and D. H. Hubel. Receptive fields and functional archichitecture in 2 nonstriate visual areas (18 and 19) of cat. *Journal of Neurophysiology*, 28:229–289, 1965.

L. Wiskott. Slow feature analysis: A theoretical analysis of optimal free responses. *Neural Computation*, 15:2147–2177, 2003.

L. Wiskott. Is slowness a learning principle of visual cortex? In *Proc. Japan-Germany Symposium on Computational Neuroscience, Wako, Saitama, Japan, February 1-4*, page 25. RIKEN Brain Science Institute, 2006.

L. Wiskott and T. J. Sejnowski. Slow feature analysis: unsupervised learning of invariances. *Neural Computation*, 14:715–770, 2002.

R. Wyss, P. Konig, and P. F. Verschure. A model of the ventral visual system based on temporal stability and local memory. *PLoS Biology*, 4:e120, 2006.

Y. Xu, T Xiao, J. Zhang, K Yang, and Z. Zhang. Scale-invariant convolutional neural networks. *CoRR*, abs/1411.6369, 2014. URL http://arxiv.org/abs/1411.6369.

D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences of the U S A*, 111:8619–8624, 2014.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014. URL http://arxiv.org/abs/1409.2329.

W. Y. Zou, S. Zhu, A. Ng, and K. Yu. Deep learning of invariant features via simulated fixations in video. *Advances in Neural Information Processing Systems*, 2012. URL http://ai.stanford.edu/~wzou/nips_ZouZhuNgYu12.pdf.