Explicit Object Representation by Sparse Neural Codes

Thesis by

Stephen Waydo

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2008

(Defended September 21, 2007)

Acknowledgments

Despite having only a single name on the cover, this thesis represents the work of a great many people. Some have provided mentorship and guidance, some have directly contributed to the work and the writing, and many more have helped me to become who I am as a researcher and as a person. I have been exceedingly fortunate in the friends and colleagues I have amassed over the years, and without them none of this would have been possible.

Richard Murray has been my advisor since my first days at Caltech, and has been an invaluable mentor as I have made the transition from coursework to independent research. Richard is a continual fountain of enthusiasm no matter the subject area, and has encouraged all of my explorations into a wide variety of subject areas as I sought a suitable area for thesis research. Always available to help me make progress, not with an answer but by finding the right question to ask, Richard has been a great teacher and friend.

Christof Koch supervised the years of work that went into creating this thesis. I have never met someone with such a deep and abiding love of science; every meeting with him leaves me recharged and enthusiastic about my work despite any obstacles. He has been a close and active collaborator on every aspect of this work, and I am immensely grateful for his willingness to invite me into his lab and help me find where I could contribute, despite having no background in biology, let alone neuroscience. His ability to stay true to biology while understanding and appreciating the value of modeling and mathematics is unparalleled. I hope our interactions have been even a small fraction as fulfilling for him as they have been for me.

I have been fortunate to have a thesis committee that has gone beyond simply being available for an examination and has actively participated in shaping the work contained in this thesis. Along with Richard and Christof, these are Jerry Marsden, Pietro Perona, and Bruno Olshausen.

In the modern era of interdisciplinary and collaborative research, very little work is done in solitude. The work contained in this thesis was a collaborative effort that I could not have even attempted without excellent colleagues. In addition to Christof, Sasha Kraskov, Rodrigo Quian Quiroga, and Itzhak Fried directly contributed to the work presented here. Furthermore, the members of Richard's and Christof's research groups have all contributed greatly with their insight and questions as this work progressed.

Two professors from my undergraduate days at the University of Washington had a particularly significant impact on my academic career. Mark Campbell provided me with my first opportunities for research as an undergraduate and helped me to secure funding to spend time working on spacecraft rather than on a more ordinary job. Juris Vagners introduced me to dynamics and control in a way that instilled in me a lasting love of the subject area and drove my choice of graduate studies. Both Mark and Juris inspired and encouraged me to pursue a Ph.D. and were instrumental in helping me gain admission and funding for my graduate work. Together with Richard and Christof they set the standard for who I want to be as a teacher and mentor.

I have been lucky to work in a wide variety of research areas while at Caltech before settling in on the work presented in this thesis. All of my collaborators have influenced my growth as a researcher and thus left an imprint on this work as well. Particularly significant among these are Lars Cremean, Bill Dunbar, John Hauser, Eric Klavins, everyone who worked on any of the incarnations of the Multi-Vehicle Wireless Testbed, and everyone who TA'd Richard's classes with me.

The Fannie and John Hertz Foundation provided generous financial support throughout my graduate career, for which I will always be grateful. My Hertz Fellowship enabled me to explore numerous avenues of research without regard to what was funded or even fundable at the time, and was a decisive factor in my success at Caltech. The Hertz Foundation has also provided some wonderful opportunities to meet luminaries of science and technology from across the public and private sectors. It is truly a unique and special organization.

I could never have survived the long years of graduate school without the love and support of a tight-knit group of friends. Chris and Lydia Voorhees, Peter and Jeannette Illsley, Brea Dyk, Tim Chung, Steve Collins, and Tony Vanelli have been there all along the way. Their friendship and the good (and bad) times we have spent together mean more to me than I can say.

Finally, last but most of all, I thank my beautiful wife Jaime. Jaime has been at my side throughout the epic adventure that is graduate school, and she has always been my biggest fan and strongest supporter, as well as a constant source of love and encouragement. She didn't question my desire to spend a year flying spaceships rather than focusing on the "real" research that would get me closer to finishing school, and she wholeheartedly supported my rather questionable decision to pursue an entirely new area of research when I should have been thinking about wrapping up work that I had already started. This work is dedicated to her, and now I look forward to the next phase of our life together with tremendous excitement.

Abstract

Neurons have been identified in the human medial temporal lobe (MTL) that display a strong selectivity for only a few stimuli (such as familiar individuals or landmark buildings) out of perhaps 100 presented to the test subject. While highly selective for a particular object or category, these cells are remarkably insensitive to different presentations (i.e., different poses and views) of their preferred stimulus. This invariant, sparse, and explicit representation of the world may be crucial to the transformation of complex visual stimuli into more abstract memories. In this thesis I first discuss the issue of how best to quantify sparseness, particularly in very sparse systems where biases are significant, and show the results of this analysis applied to human MTL data. I also provide an overview of existing results from other investigators on measuring sparseness both elsewhere along the primate visual pathway and in selected other sensory processing systems. From there I move into the computational realm. Sparse coding as a computational constraint applied to the representation of natural images has been shown to produce receptive fields strikingly similar to those observed in mammalian primary visual cortex. I apply sparse coding as a model for processing further along the visual hierarchy: not directly to images but rather to an invariant feature-based representation of images analogous to that found in the inferotemporal cortex. This combination of sparseness and invariance naturally leads to explicit category representation. That is, by exposing the model to different images drawn from different categories, units develop that respond selectively to different categories. After extending an existing model of sparse coding and providing some mathematical analysis of its operation, I show results obtained by applying this method both to unsupervised category discovery in images and to differentiation between images of different individuals.

Contents

List of Figures

List of Tables

Chapter 1

Overview

1.1 Experimental Motivation

The fundamental motivation for the research culminating in this thesis was the results of Quian Quiroga, Reddy, Kreiman, Koch, and Fried (2005), who recorded the activity of single neurons in the human medial temporal lobe (MTL), a brain area linked to memory consolidation and cross-modal association. The recordings were carried out in the laboratory of neurosurgeon Itzhak Fried at UCLA, with his active participation in all stages of the experiments. Dr. Fried implants chronic electrodes in patients with pharmacologically intractable epilepsy for the purpose of localizing the seizure focus for later resection. In the experiments I describe here, microwires capable of measuring the activity of individual neurons were included at the electrode tips. During the roughly one week period of time that the electrodes were in place in each patient researchers were able to record neuronal activity while the patient—who volunteered for these studies—participated in various experiments.

Two complimentary experimental paradigms involving the patient viewing natural visual stimuli form the experimental motivation for my work. In the first, known as a "screening" session, the patient viewed multiple presentations of roughly 100 different images of individuals, animals, objects, and landmark buildings presented on a laptop computer. The goal of this session was to discover at least one stimulus that some neuron was selective for. In a subsequent "invariance" session, the patient again viewed numerous images, but several different images (with varying pose, lighting, background, etc.) of objects that elicited strong responses in the screening session were presented in addition to the standard images. Two important discoveries came out of these two experiments. First, in general, MTL neurons responded strongly (defined by a threshold above background firing rate) only very rarely—most neurons did not respond strongly to any image in the screening session, and those that did sometimes respond strongly only did so to a very small number of images. This was evidence that MTL employs what is known as a "sparse" code, as opposed to a "dense" or combinatoric code in which individual neurons would respond much more frequently. Second, several neurons were identified (and many more have been since) that responded strongly to many very different images of the same person or object, but not to images of different objects (even very similar ones), a property known as "invariance." The best known example from this study was a neuron that responded to seven different images of the actress Jennifer Aniston with an average of 4.85 spikes between 300 and 500 ms after stimulus onset, but was virtually silent otherwise (with a baseline rate of 0.03 spikes/s and very few spikes in response to other images). Further investigations have uncovered cells that are invariant not only to different images of the same object, but also to the name of the object both printed or spoken aloud (Quian Quiroga, *personal communication*), underscoring the fact that, while it receives input from visual areas, MTL itself is not limited to visual processing.

These results suggest a sparse and invariant encoding in MTL and seem to imply the existence of "grandmother cells" that respond to only a single category, individual, or object (Konorski, 1967; Barlow, 1972; Gross, 2002), though limitations on the number of images that can be presented and neurons that can be recorded from stop us short of making such a controversial claim. Further, these neurons seem to respond to the high-level "concept" of their preferred object rather than to any particular features of the input. The work in this thesis represents an effort to better understand the behavior of these remarkable cells from two perspectives—quantifying as precisely as possible the behavior of these cells, and building a computational model capable of reproducing some aspects of this behavior.

1.2 Outline and Contributions of Thesis

Chapters 2 and 3 of this thesis are devoted to developing a better understanding of the experimental results first reported by Quian Quiroga and colleagues. First, in Chapter 2, I discuss the various ways one might answer the fundamental question "How sparse is the code?" based on experimental data. Sparseness is an important parameter both for understanding the level of network activity and for quantifying network capacity (Tsodyks & Feigel'man, 1988; Treves & Rolls, 1991; Meunier, Yanai & Amari, 1991; Willmore & Tolhurst, 2001; Hahnloser, Kozhevnikov & Fee, 2002), but no single measure exists that serves these purposes well in all circumstances. I describe several commonly used sparseness measures and discuss the strengths and weaknesses of each. I then turn to the practical problem of how to estimate sparseness based on neural recordings. My primary contributions in this area are to show that the most direct method for approaching this task breaks down in very sparse regimes such as the human MTL due to extreme sensitivity to noise, and to propose a less direct but more robust method for estimating sparseness in this setting. Then, in Chapter 3, I apply this method to the human MTL data reported by Quian Quiroga et al., showing that very sparse, though likely not grandmother, coding is employed by MTL. This work has appeared in journal form as "Sparse Representation in the Human Medial Temporal Lobe" (Waydo, Kraskov, Quian Quiroga, Fried & Koch, 2006). I also place this data in the context of experimental results from other systems, both at different locations along the primate ventral visual stream and in selected other systems such as rat hippocampus and auditory cortex.

In Chapters 4 and 5, I present a computational model for the human MTL cells described above and how they can arise as a consequence of an unsupervised learning process. My central hypothesis is that the two distinct but complimentary computational principles of *sparseness* and *invariance* together naturally lead to the type of sparse, selective representation observed in MTL. I treat these two principles as separable, modeling the ventral visual stream as a system for producing invariant, but not necessarily sparse, representations, then modeling MTL as learning a sparse representation for the activity of the visual system (without the benefit of a teacher who labels each image). The process by which a sparse code is learned builds on work by Olshausen and Field (1996, 1997). In that work, Olshausen and Field developed a neurally implementable learning algorithm that seeks a sparse representation of its inputs (meaning one in which the individual coding elements are active rarely), and applied it directly to natural image patches, learning a set of basis functions for images much like that observed in mammalian early visual cortex. In Chapter 4 I describe this process in detail, then extend the model in several ways that improve both its computational efficiency and its relevance as a model for MTL. In Chapter 5 I apply this model to collections of images of different individuals and categories after first processing them through one of two different models for invariant feature extraction. That is, rather than applying the model directly to pixels, as Olshausen and Field did, I apply it to some invariant representation of image features obtained from established biologically-motivated machine vision algorithms. Through this learning process, model neurons emerge that respond selectively to images of particular individuals or categories, much like those observed in human MTL. Portions of this work have appeared as a conference paper (Waydo & Koch, 2007a), and a journal version is in press (Waydo & Koch, 2007b).

Finally, in Chapter 6, I summarize the results of the thesis and outline a number of potentially fruitful avenues of future research. Possible extensions include expanding the scope of the model to cover the entire visual hierarchy (rather than applying it only at the top and the bottom), implementing the model using more biophysically realistic neurons, and developing a method for cross-modal association to model the multi-modal effects briefly mentioned above.

Chapter 2

Quantifying the Sparseness of Neural Codes

A fundamental question confronting any examination of neural coding schemes is "How sparse is the code?" (Barlow, 1972; Olshausen & Field, 2004). Sparseness, either intuitively defined as how frequently a neuron responds above some threshold or according to various more general schemes, is an important quantity both for understanding the level of network activity and for quantifying network capacity (Tsodyks & Feigel'man, 1988; Treves & Rolls, 1991; Meunier et al., 1991; Willmore & Tolhurst, 2001; Hahnloser et al., 2002). In later chapters I will explore in detail the sparseness measured along the visual pathway and the implications for neural coding, and develop computational models of visual processing inspired by these findings. First, however, I turn to the task of quantifying sparseness. In Section 2.1 I will describe several candidate measures for sparseness and discuss the strengths and weaknesses of each. In Section 2.2 I discuss the practical problem of estimating sparseness from neural recordings. The work in Section 2.2 was performed in collaboration with Alexander Kraskov and Christof Koch, with additional contributions from Rodrigo Quian Quiroga and Itzhak Fried in portions that overlap with material published as "Sparse Representation in the Human Medial Temporal Lobe" (Waydo et al., 2006).

2.1 Sparseness Measures

While seemingly an intuitive concept, sparseness can be very difficult to rigorously define and quantify, and the appropriate choice of measure can vary depending on the nature of the questions being investigated. Several authors have discussed and compared different measures (Willmore & Tolhurst, 2001; Olshausen & Field, 2004); what follows is an expanded description of the most common measures, along with their strengths and weaknesses.

2.1.1 Notation

I denote random variables by capital letters, with corresponding samples in lower case, i.e., x is a sample of a random variable X. The probability of an event is written $P[event]$, so the probability that X takes on a value larger than a is written $P[X > a]$. The probability density function for X is denoted by $f_X(x)$. The expectation operator is denoted by $E[\cdot]$ or $\langle \cdot \rangle$, with the special cases of the mean $E[X] = \mu_X$ and the variance $E[(X - \mu_X)^2] = \sigma_X^2$.

2.1.2 Threshold

The intuitive notion we would like to capture with sparseness is the likelihood that a neuron will respond "significantly" to any particular stimulus. In the case of a truly binary neuron (such as in a Hopfield network), then, sparseness can be simply defined as the probability that a neuron will be in the "on" state. Real neurons, however, do not necessarily fire in a clean "on/off" fashion; rather a neuron responds with some rate R to a stimulus. In this more general case, we choose some reasonable threshold r_T and define the sparseness as

$$t = P[R \geq r_T]. \tag{2.1}$$

Provided the threshold r_T is chosen in a meanful way, this definition clearly captures the basic question of "how active is this neuron?" It is particularly relevant when attempting to quantify the behavior of neurons that have a strongly bimodal distribution, such as a neuron that fires with some high mean rate when a preferred stimulus is present and some low background rate otherwise. For this reason it has been useful when studying the responses of category- and individual-specific cells in the human medial temporal lobe (Waydo et al., 2006).

In the case where a neuron has a unimodal distribution of firing rates and significant information may be carried in the smoothly varying firing rate (as opposed to a binary present/not present judgement), this measure may fail to capture important subtleties in the rate distribution. I shall show below, however, that it may still provide a reasonable estimate of more sophisticated measures that is robust to noise. I turn now to two sparseness measures that directly address the issue of continuously variable firing rates.

2.1.3 Kurtosis

One common definition of sparseness is that a sparse distribution has more probability density concentrated both near the mean and far from it than a Gaussian of the same variance (Dayan & Abbott, 2001, p. 378), that is, it has a sharp peak and a heavy tail. This definition is related to a measure called *kurtosis*, which is the fourth central moment of a probability distribution. The kurtosis k of a probability distribution $f_R(r)$ is defined as

$$k \equiv E\left[\left(\frac{R-\mu_R}{\sigma_R}\right)^4\right]. \tag{2.2}$$

Occasionally an alternative definition $k^* = k - 3$ (sometimes called the "kurtosis excess") is used so that a Gaussian distribution has a kurtosis of $k^* = 0$, with less sparse distributions having negative kurtosis excess and sparse distributions having positive

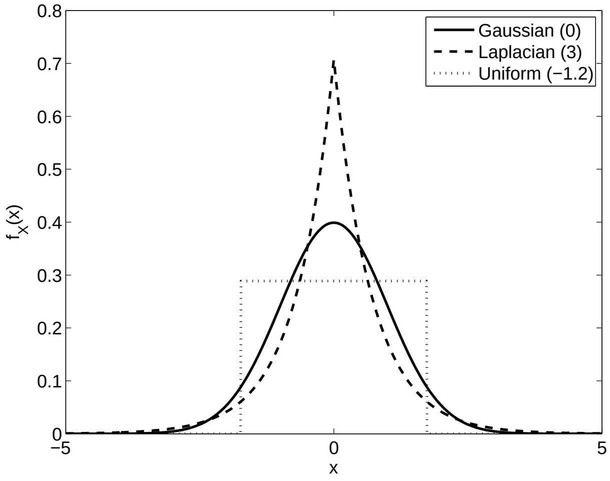


Figure 2.1: Kurtosis excess for several probability distributions, each with zero mean and unit variance. Shown are a Gaussian distribution (solid, $k^* = 0$), a Laplacian distribution (dashed, $k^* = 3$), and a uniform distribution (dotted, $k^* = -1.2$).

kurtosis excess, though this distinction has no bearing on the following discussion. Figure 2.1 gives a few examples of probability distributions with zero mean and unit variance but different kurtosis. Note that larger kurtosis corresponds to a taller peak and heavier tails, which corresponds well with the intuitive definition of sparseness described above.

Kurtosis is generally described as reflecting either the "peakedness" or the heaviness of the tails of f_R, and has the convenient property of being invariant both to shift and scale. In the neural coding literature large values of k are identified with sparse codes (Olshausen & Field, 1996; Bell & Sejnowski, 1997; Vinje & Gallant, 2000; Willmore & Tolhurst, 2001). This description, however, comes with the caveat that kurtosis is only appropriately applied to reasonably symmetric, unimodal distributions such as those obtained from linear filters or artificial neurons (Vinje & Gallant, 2000; Olshausen & Field, 2004), and not to the one-sided distributions necessarily obtained from real neurons. Vinje and Gallant (2000) alleviate this difficulty by reflecting their measured neural responses about zero before computing k (that is, for each response r they include an artificial response of $-r$), but in the case of bimodal distributions, such as those that may be obtained from neurons involved in recognition, the meaning of kurtosis remains unclear.

A further challenge confronting the application of kurtosis as a measure of neural sparseness is its invariance with respect to flipping a distribution about its mean (because it measures *only* shape). When evaluated with kurtosis, a neuron that is highly active, only rarely dropping its firing rate, would be considered just as sparse as a neuron that is highly inactive, only rarely firing strongly. From an information-theoretic point of view the two neurons may carry equal information, one conveying that information by a decrease in firing rate and the other by an increase, but in a biological context a reasonable sparseness measure should rate the mostly inactive neuron as much more sparse than the mostly active neuron.

Many of these difficulties stem from the fact that, as a high-order moment, several disparate factors influence kurtosis and it is difficult at best to capture it intuitively. Numerous papers in the statistics literature have lamented this difficulty, with comments such as "what do we even mean by kurtosis?" (Bickel & Lehmann, 1975), "there seems to be no universal agreement about the meaning and interpretation of kurtosis"(Moors, 1986), and "there is no agreement on what kurtosis measures"(Ruppert, 1987). Darlington first challenged the traditional interpretation of kurtosis, arguing that "kurtosis is best decribed not as a measure of peakedness versus flatness, as in most texts, but as a measure of unimodality versus bimodality" (Darlington, 1970). This view was later found to fall short of capturing the essence of kurtosis, and the most precise interpretation is the intuitively unsatisfying one that kurtosis measures the dispersion of the distribution about the two points $\mu \pm \sigma$ (Moors, 1986).

For the reasons outlined above I conclude that, while kurtosis may be a useful tool for interpreting the sparseness of filters and *artificial* neurons with symmetric response distributions, it may not be appropriate for interpreting real neural data. This is particularly true in the case of bimodal response distributions that may be obtained, for example, from neurons that fire strongly to some preferred stimulus and weakly or not at all to other stimuli.

2.1.4 Treves-Rolls Sparseness

Treves and Rolls (1991) present an alternative measure of sparseness more appropriate for application to neural data. Let $f_R(r)$ be the probability density function for the neuron's response rates. They defined

$$a \equiv \frac{E[R]^2}{E[R^2]}, \tag{2.3}$$

that is, the square of the mean response divided by the mean squared response. With this definition the (dimensionless) sparseness a varies between 0 and 1, and small values of a correspond to sparse representations. This definition has two convenient properties. First, in the case of a binary neuron that responds to a stimulus with probability t (and has zero response otherwise), $a = t$; so indeed, a is the probability that the neuron responds significantly. Second, from elementary properties of the mean and variance we can rewrite Equation 2.3 as

$$a = \frac{\mu_R^2}{\mu_R^2 + \sigma_r^2}, \tag{2.4}$$

Thus the sparseness is small if the variance is large compared to the mean (i.e., when the neuron has widely separated responses to different stimuli), and large if the variance is small compared to the mean (i.e., when most responses are very similar).

In addition to being a relatively intuitive generalization of our notion of sparseness to neurons with continuous rates, a is related to the theoretical storage capacity of an autoassociative neural network (Tsodyks & Feigel'man, 1988; Treves & Rolls, 1991; Meunier et al., 1991). Hence we take an interest in a not simply as a means of quantifying the question of how frequently neurons fire “strongly,” but also as a functionally relevant parameter.

As it is more appropriate than kurtosis when measuring the sparseness of real neural data, this measure has been used extensively in experimental work (Rolls & Tovee, 1995; Vinje & Gallant, 2000; Weliky, Fiser, Hunt & Wagner, 2003). In the remainder of this work I will take this as my primary definition of sparseness.

2.1.5 A Selectivity Index

Working in the context of sparse, invariant neurons in the human medial temporal lobe (Quian Quiroga et al., 2005), Quian Quiroga and colleagues (2007) propose a novel threshold-independent index for quantifying the selectivity of neurons. They first define a function describing the normalized number of responses above a threshold r_T

$$\tilde{S}_r(r_T) = \frac{1}{S}\sum_{i=1}^{S}\theta(r_i - r_T), \tag{2.5}$$

where $\theta(x) = 1$ for $x > 0$, $\theta(x) = 0$ for $x \leq 0$. Note that if $f_R(r)$ is the probability density function of the response distribution and $F_R(r)$ the corresponding cumulative density function, then for large S $\tilde{S}_r(r_T)$ approaches $1 - F_R(r_T)$. The area under this curve (as r_T is varied) is

$$A = \frac{1}{M}\sum_{j=1}^{M}\tilde{S}_r(r_{T,j}), \tag{2.6}$$

where $r_{T,j} = r_{min} + j\left(\frac{r_{max}-r_{min}}{M}\right)$ defines M equally spaced threshold values between the minimum and maximum responses r_{min} and r_{max} (equivalently one can simply rescale the responses to lie between 0 and 1). This area will be close to 0.5 for a uniform distribution of firing rates and much smaller when only a small fraction of responses are significant. Quian Quiroga and colleagues then define their *selectivity index* by

$$I = 1 - 2A. \tag{2.7}$$

Consider the case of a binary neuron. If all responses but one are significant, $I = 1 - 2\left(\frac{S-1}{S}\right)$, so for large S I approaches -1. If instead only a single response is significant, $I = 1 - 2\left(\frac{1}{S}\right)$, so for large S I approaches 1.

Assuming a large number of samples and noting the relationship between $\tilde{S}_r(r_T)$ and $F_R(r_T)$, some algebra yields the relationship

$$I = \frac{2}{r_{max} - r_{min}} \int_{r_{min}}^{r_{max}} F(r)dr - 1, \tag{2.8}$$

so the selectivity is (in the limit) defined by the cumulative density function of the response distribution. From this definition it can be seen that any symmetric response distribution (e.g., Gaussian or uniform) will, in the limit, have $I = 0$. Thus, like skewness (which is related to the 3^{rd} central moment of a distribution), this measure quantifies the asymmetry of the response distribution. Values close to the minimum of -1 indicate that most responses are clustered near the maximum (the neuron nearly always responds), values close to the maximum of 1 indicate that most responses are clustered near the minimum (the neuron rarely responds), and values close to zero indicate a symmetric response distribution.

As noted by Quian Quiroga et al., this index has a few convenient features. It is threshold-independent, and captures the selectivity of roughly binary neurons well and so conforms to our intuitive notion of sparseness. As with any threshold-independent measure, accurate results depend strongly on a given experiment finding enough responses to characterize the response distribution well (since no assumptions are placed on the form of the distribution).

2.1.6 Population versus Lifetime Sparseness

In most discussions of sparse coding, the quantity of interest would more precisely be defined as *lifetime sparseness*, which refers to the sparseness of an individual neuron's responses over time. This is the sense in which I defined sparseness above. It is also possible, however, to discuss the sparseness of the responses of a population of neurons, or the *population sparseness.* In this case, the relevant sparseness measures would be the same as defined above, except that the expectations would be taken across the population of neurons for an individual stimulus rather than across the universe of stimuli for an individual neuron (perhaps then averaging across all stimuli).

If the neurons' responses to stimuli are independent and identically distributed, it is clear from the definitions above that lifetime and population sparseness are exactly equivalent. Simply speaking, the fraction of stimuli an individual neuron responds to will be equal to the fraction of neurons that respond to a particular stimulus. If, however, some neurons participate in many more representations than others, the population sparseness may be very different than the lifetime sparseness. Willmore and Tolhurst (2001) investigated this issue by examining the representation of a set of natural images within several different filtering schemes such as Gabor, principal components, and independent components filters. They computed both the population and lifetime sparseness of the responses of each of the filters in each of these coding systems and found no direct relationship between the two. This should not be an unexpected result. For example, principle components analysis (PCA) specifically seeks filters such that a small number of filters code for a large portion of the input statistics (Hancock, Baddeley & Smith, 1992). A set of PCA filters would be expected to have a high population sparseness but low average lifetime sparseness, because a few of the filters have large output much of the time, while many of the filters are active only rarely.

From an experimental point of view, it is not possible to directly measure the population sparseness of a given representation—to do so would require recording from a large enough subset of the entire coding set of neurons to establish the response statistics at the population level. The below discussion will then be restricted to the lifetime sparseness, which can be estimated by recording a single neuron's responses to a large group of stimuli. Where I make inferences about the population sparseness, it is under the assumption that the population of neurons under consideration is homogenous (in the sense of their responses being i.i.d.).

It should be noted that sparseness should properly be defined with respect to a particular class of "relevant" stimuli. I assume in what follows that the stimulus set is relevant to the computation performed by the neurons from which we record. In other work describing experimental results obtained from the human medial temporal lobe my co-authors and I discuss the potential bias due to choice of stimulus set (Waydo et al., 2006, and see Chapter 3). Note also that the issues I discuss here are different than the extreme temporal sparseness observed, for example, in high vocal center neurons of the zebra finch (Hahnloser et al., 2002; Fiete, Hahnloser, Fee & Seung, 2004). There, neurons appear to encode a time-varying signal (the finch's song) using precise spike timing, and "sparseness" refers to the fact that individual neurons encode their portion of the signal using an extremely small number of spikes. In this work, I am instead concerned with encoding static signals, and sparseness refers to how rarely individual neurons will be active (in the case of lifetime sparseness) or how few neurons will be active simultaneously (in the case of population sparseness).

2.2 Estimation of Sparseness from Neural Recordings

As we have seen above, a great deal of work has been done to find an appropriate quantitative definition of sparseness and to understand how sparseness fits in models of network performance. Comparatively little attention has been paid to the practical challenge of how to accurately measure it, the problem to which I now turn. In a typical experimental paradigm, the activity of one or more neurons is recorded while a collection of stimuli is presented to the subject (Young & Yamane, 1992; Rolls & Tovee, 1995; Vinje & Gallant, 2000; Quian Quiroga et al., 2005; Kreiman et al., 2006). Due to constraints on experiment duration, the number of stimuli presented generally varies in the range of about 50–100. The rate of "spontaneous" background firing, usually of unclear significance (is it noise or signal?), can be significant and needs to be properly accounted for. In this section I examine two methods for estimating representational sparseness from spiking data, direct computation and a binary model-based approach. My primary contribution is to show that the direct computation is, in many cases, vulnerable to corruption by noise, particularly if the underlying code is sparse. I further show that this issue is likely to arise when the mean noise is large compared to the mean response, regardless of the peak response. In this regime it is more accurate to apply a binary model using a response threshold and compute the probability that a neuron fires above that threshold.

This work was performed in collaboration with Christof Koch and Alexander Kraskov (now at University College London); my contribution was the development of the probabilistic reasoning about sparseness and the quantification of the biases inherent in computing sparseness from limited, noisy datasets.

2.2.1 Direct Computation of Sparseness

Let S be the number of stimuli presented and r_i be the neuron's response to stimulus i. The obvious way to estimate a is to calculate the sample mean and the sample mean square, or, letting $\hat{a}$ be the estimate of a,

$$\hat{a} = \frac{\left(\frac{1}{S}\sum_{i=1}^{S} r_i\right)^2}{\frac{1}{S}\sum_{i=1}^{S} r_i^2} = \frac{\overline{r}^2}{\overline{r^2}}, \tag{2.9}$$

where the bar over a quantity denotes the sample average.

This method of calculating $\hat{a}$ has two clear strengths. First, it is a direct calculation of the quantity of interest, and so the result needs little interpretation. Secondly, no underlying assumptions about the neuron's behavior (i.e., a firing-rate model) are required—one simply collects the neuron's responses and plugs them in to Equation 2.9. This second strength, however, may also be a pitfall of this method. If one has a very sparse neuron for which large responses are rare, one may measure a large number of responses that are purely noise. Because these responses are all very similar to one another, Equation 2.9 will erroneously yield a large value. In what follows I will make this issue more precise.

A fundamental challenge confronting the application of Equation 2.9 is that a is sensitive to a uniform translation of responses, that is to adding a constant offset to all responses, such as when taking spontaneous firing into account. For example, consider a binary neuron with an "off" rate of 0 spikes/s, an "on" rate of 5 spikes/s, and a firing probability of 5%. The sparseness calculated from Equation 2.3 is $a = 5\%$. If instead it fires at 6 spikes/s with the same probability and 1 spike/s otherwise, $a = 57\%$. This is a very different result, but we would argue that the answer to our basic question ("how often does this neuron respond significantly") has not changed.

This feature in turn means that the calculation of a can be highly vulnerable to noise, particularly for very sparse systems. I will here examine the effect of this vulnerability for a simple model with additive noise. Consider a system in which a neuron's response to a stimulus s is the sum of two components, that is

$$R(s) = X(s) + Y, \tag{2.10}$$

where $X(s)$ is the deterministic portion of the neuron's response (that is, $X(s)$ is some parameterization of the neuron's *tuning curve*) and Y is a noise term that is independent of X. Assuming we choose stimuli randomly from the universe of all possible stimuli, X can be viewed as a random variable.

Let X and Y have means μ_x and μ_y and variances σ_x^2 and σ_y^2, respectively. Presumably in any discussion of sparseness what we are truly interested in is the sparseness of the distribution of X, which by Equation 2.4 is

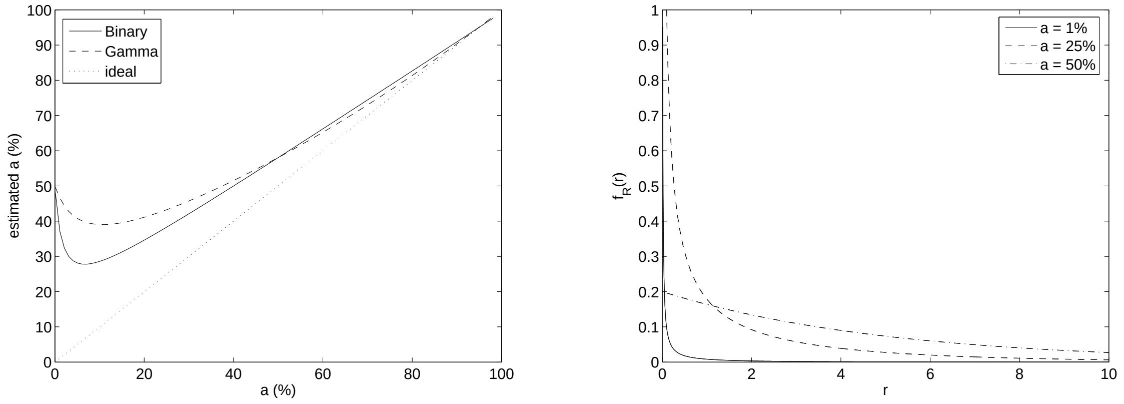
$$a = \frac{\mu_x^2}{\mu_x^2 + \sigma_x^2}. \tag{2.11}$$

However, we have only the noisy responses R with which to characterize it. Because X and Y are independent, the response distribution f_R has mean $\mu_x + \mu_y$ and variance $\sigma_x^2 + \sigma_y^2$. Applying Equation 2.4, the sparseness of the noisy distribution is then

$$\hat{a} = \frac{(\mu_x + \mu_y)^2}{(\mu_x + \mu_y)^2 + (\sigma_x^2 + \sigma_y^2)}. \tag{2.12}$$

Comparing Equations 2.11 and 2.12, we see that $\hat{a}$ approaches a if μ_x is large compared to μ_y and σ_y. If this is not the case, $\hat{a}$ may, in fact, be quite different from the underlying a that we wish to estimate. Roughly speaking, we have a signal-to-noise ratio characterized by μ_x/μ_y or μ_x/σ_y. Even for seemingly low levels of noise, this can present a significant difficulty. Although the significant responses may be quite large in comparison to the noise, for a very sparse system we expect the *mean* response to be small and so $\hat{a}$ will not accurately reflect a. Note also that only the mean and the variance of the noise affect Equation 2.12; apart from these parameters the error is independent of the details of the noise distribution.

Consider the case where X is a binary distribution, with an "off" rate of 0 spikes/s, and an "on" rate of 10 spikes/s. The model neuron fires to a random stimulus with probability a (as we pointed out above, the firing probability of such a neuron is exactly its sparseness a). Now add to each response an independent noise component with mean and standard deviation of 1 spike/s. It would seem in this case that the signal-to-noise ratio ought to be very favorable: the "on" response is 10 times greater than the mean noise level. However, the relevant comparison for our computation of



(a) Theoretical sensitivity of sparseness calculation (Equation 2.12) to noise

(b) Gamma probability density function for selected values of a

Figure 2.2: Sensitivity of response sparseness $\hat{a}$ to noise. Response sparseness is plotted as a function of the underlying distribution's sparseness for binary (solid) and gamma (dashed) distribution. High rate for the binary neuron is 10 spikes/s and scale parameter λ for the gamma neuron is 5, so that both neurons have mean firing rate 5 spikes/s at $a = 1/2$. Sparseness is varied by adjusting the firing probability a for the binary neuron and the shape parameter α for the gamma neuron. The noise is held constant at 1 spike/s mean and standard deviation.

$\hat{a}$ is between the *mean* response μ_x and the mean noise $\mu_y = 1$. In this case, the mean response is $10a$, and so if a is even as low as 10% we may run into trouble. Plugging these numbers into Equations 2.11 & 2.12 reveals that $\hat{a} = 29\%$ for $a = 10\%$, and $\hat{a} = 38\%$ for $a = 1\%$!

While this example used binary model neurons to illustrate noise sensitivity, the basic issue remains for any response model in which the mean response decreases with a. For example, consider a neuron whose noiseless firing rates follow a gamma distribution with shape parameter η and scale parameter λ,

$$f_R(r) = \frac{r^{\eta-1} e^{-\frac{r}{\lambda}}}{\lambda^{\eta}\Gamma(\eta)}, \tag{2.13}$$

where $\Gamma(\eta)$ is the gamma function. This distribution, depicted in Figure 2.2(b), is convenient because it has an exponential falloff in rates and an easily tunable sparseness. The sparseness of this distribution is $a = \frac{\eta}{1+\eta}$, while the mean rate is

$\eta\lambda$. If we fix the scale parameter λ, then the mean rate declines with η and we have the same bias problem as before. Figure 2.2(a) illustrates this issue for both the gamma and binary neuron. Plotted is the sparseness estimate $\hat{a}$ as a function of the underlying distribution's sparseness a. Both neurons are parameterized such that their mean firing rate is 5 spikes/s when $a = 1/2$. The sparseness of the binary neuron is adjusted using the response probability a, while that of the gamma neuron is adjusted using the shape parameter η while the scale parameter λ is held fixed. For all levels of sparseness the noise is fixed with mean and standard deviation 1 spike/s. We see from this figure that the bias due to noise can be substantial. Worse, the variation of $\hat{a}$ with a is not even monotonic—the estimated sparseness *increases* as the true sparseness becomes very small.

2.2.2 A Binary Model

If we make some *a priori* assumptions about the underlying rate distribution, we can generate a few alternative methods for estimating a. In contrast to the direct calculation, in which the signal-to-noise ratio was that of the means of the response and noise components, the relevant ratio will be that of the size of the "large" responses to the noise. We can achieve this by assuming that the neuron responds to *some* stimuli with at least some rate r_T, where we pick r_T to be our threshold for considering a response significant. With this threshold we can then treat our neurons as behaving in a binary fashion, with responses above r_T considered "on" and all others "off." It is then straightforward to estimate the sparseness simply from the fraction of stimuli a neuron responds to, or

$$\hat{a} = \frac{S_r}{S}, \tag{2.14}$$

where S is the total number of stimuli presented and S_r is the number of stimuli the neuron responded to with at least r_T. This approach has the advantage that it exactly answers the question of how frequently the neuron responds at a significant level, as long as we can define a reasonable value for significance. Note that if the underlying probability of achieving a significant response is a, then S_r follows a binomial distribution and $E[\hat{a}] = a$. This estimate is biased upward by the probability that an "off" response will be pushed above threshold by noise, so this bias is reduced by increasing r_T. Setting r_T too high may, of course, cause significant responses to be ignored as noise, so the separation between the significant responses and the background noise is the limiting factor for this method, which is much more favorable than the difference between the mean response and the background noise in the case of a sparse distribution.

Continuing with this binary model, I developed a method for determining a *range* of sparseness that is consistent with our data, or alternatively the probability distribution of the underlying response probability a. Let f_a be the probability density function of the response probability a. We want to determine $f_a(\alpha|S_r = s_r)$, the probability density function for a given the observed number of responses s_r. We place no *a priori* assumption on a, so we set $f_a(\alpha) = 1$ for $0 \leq \alpha \leq 1$, that is, a is equally likely to take on any value between 0 and 1.

At a particular value of a, the number of responses follows the binomial distribution

$$P[S_r = s_r|a = \alpha] = \left(\begin{array}{c} S \\ \\ s_r \end{array} \right) \alpha^{s_r}(1-\alpha)^{S-s_r}. \tag{2.15}$$

Bayes' rule applied to this system gives

$$P[S_r = s_r|a = \alpha]f_a(\alpha) = f_a(\alpha|S_r = s_r)P[S_r = s_r], \tag{2.16}$$

or, solving for the desired distribution,

$$\begin{aligned}
f_a(\alpha|S_r = s_r) &= \frac{P[S_r = s_r|a = \alpha]f_a(\alpha)}{P[S_r = s_r]} \\
&= \frac{\begin{pmatrix} S \\ s_r \end{pmatrix} \alpha^{s_r}(1-\alpha)^{S-s_r} f_a(\alpha)}{\int_0^1 P[S_r = s_r|a = \alpha]f_a(\alpha)d\alpha} \\
&= \begin{pmatrix} S \\ s_r \end{pmatrix} \alpha^{s_r}(1-\alpha)^{S-s_r}(S+1). \qquad (2.17)
\end{aligned}$$

For a given experiment we obtain a curve $f_a(\alpha)$ describing the range of plausible values for the underlying response probability a. Figure 2.3 gives examples of three such curves derived from three different response patterns with $S = 100$. It is easy to check that the peak of this distribution is at $\alpha = \frac{s_r}{S}$, so the most likely value from this distribution matches the intuitive definition of a for a binary model. We now have additional information, however, about just how close the true a is likely to be to our estimate $\hat{a}$.

While the imposition of the binary model eliminates the noise sensitivity of the direct method, it is of course not without challenges of its own. Primary among these is that the binary model is an approximation of the true behavior that varies in its accuracy depending on the details of the true firing rate distribution. In some cases we may have a neuron that responds robustly at a high rate to some stimuli and much less to others, and so the appropriate choice of threshold is clear. In other cases, though, a neuron may respond with a wide variety of rates and the resulting estimate of a based on a binary model will be sensitive to threshold. Obviously this method sacrifices some of the detail in the neuron's responses in favor of robustness to noise.

It is possible to quantify to some extent the relationship between the true sparse-

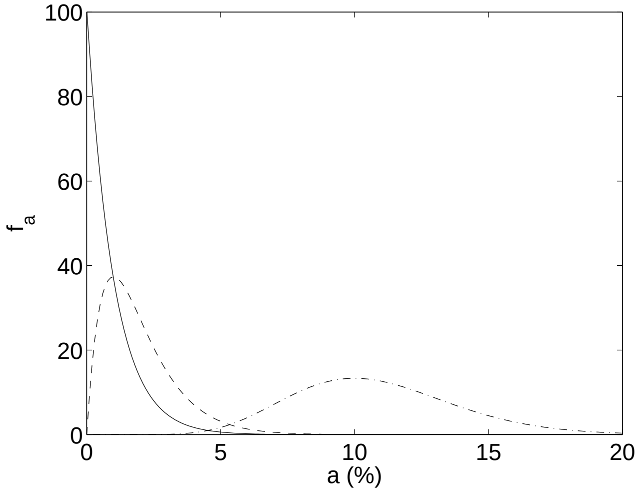


Figure 2.3: Example probability density functions for sparseness (expressed as a percentage of stimuli that evoke a response) computed using Equation 2.17 in three scenarios in which the number of stimuli presented was $S = 100$. The solid curve would be obtained from a neuron for which no significant responses were recorded, while the dashed and dash-dotted curves correspond to neurons for which 1 and 10 significant responses were recorded, respectively. As the number of stimuli shown to the cell approaches the total number of images stored by the network, the density function will converge to an ever-narrower curve centered at the true sparseness a.

ness of the underlying rate distribution and our estimate $\hat{a}$ derived from a binary model. Note that $\hat{a} = P[R \geq r_T]$, that is, our estimate of a is just the probability that a response is greater than the threshold r_T. The Markov inequality provides an upper bound for the probability that a positive random variable exceeds some threshold (Leon-Garcia, 1994). By this inequality,

$$\hat{a} = P[R \geq r_T] \leq \frac{E[R]}{r_T}. \tag{2.18}$$

Applying our additive noise model from above, we have

$$\hat{a} \leq \frac{\mu_x + \mu_y}{r_T}. \tag{2.19}$$

Now assume as above that $\mu_x = a\mu_0$ for some rate μ_0. This is true for a binary neuron with $\mu_0 = r_h$, and is approximately true for a gamma neuron with $\mu_0 = \lambda$ and small a. We now have

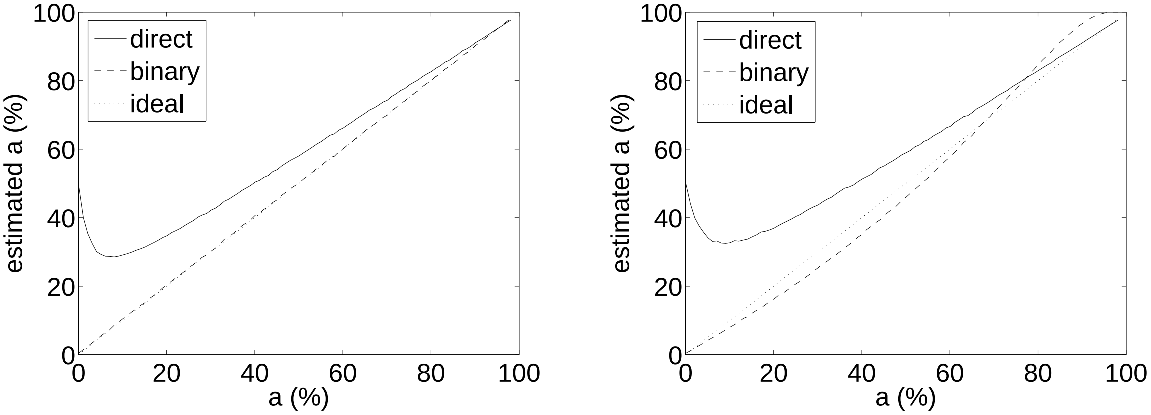
$$\hat{a} \leq a\frac{\mu_0}{r_T} + \frac{\mu_y}{r_T}. \tag{2.20}$$

Here we have an upper bound on $\hat{a}$ with two convenient features. First, the bias is equal to $\frac{\mu_y}{r_T}$, the mean noise relative to the threshold, so if our threshold is large compared to the mean noise the bias is small. Second, the bound decreases with a, so we are guaranteed that for any choice of threshold a small true a will give us a small estimate $\hat{a}$ (provided the mean noise μ_y is small compared to the threshold).

This bound can also inform our choice of r_T, but some caution is required. Clearly we would like r_T to be significantly larger than the mean noise μ_y to reduce the offset from the noise term. From the first term, the temptation would be to set r_T close to one's best guess for μ_0. The bound given by the Markov inequality may be quite loose, though, and this approach could result in a significant underestimate of a. A balance must be struck between making r_T as large as possible without getting too close to μ_0.

2.2.3 Simulation Results

Figures 2.4(a) and (b) depict Monte Carlo simulation results of sparseness estimation with binary and gamma underlying rate distributions, respectively. The binary neuron had an "on" response of 10 spikes/s, while the gamma neuron was tuned to have the same mean response at each a as the binary neuron. For each neuron I generated responses to $S = 100$ stimuli using the appropriate probability distribution plus Poisson noise with unit (i.e., 1 spike/s) mean and variance. I then estimated the sparseness of each neuron using both the direct calculation of Equation 2.9 and the binary model with $r_T = 5$ spikes/s (corresponding to $r_T = \mu_0/2$ in the above discussion). Plotted are the mean estimates over 1000 simulated neurons of each



(a) Binary rate distribution

(b) Gamma rate distribution

Figure 2.4: Monte Carlo simulation results for sparseness estimation for binary and gamma distribution rate models. Noise was Poisson with 1 spike mean and standard deviation. Solid line is the direct calculation (Equation 2.9); dashed line is the binary calculation with a threshold of 5 spikes/s (Equation 2.14). Dotted line is the ideal of $\hat{a} = a$ (which exactly overlaps the binary calculation in (a)). The number of stimuli was $S = 100$, and 1000 neurons were simulated.

type.

In both cases the binary model produced substantially more accurate estimates than the direct computation, particularly (as predicted) for very sparse distributions. Because the binary model matched the underlying rate distribution, performance in that case was nearly perfect. In the case of the gamma distribution, though, performance was still very good in the sparse regime despite the model mismatch.

Figure 2.5 shows the effect of the choice of response threshold. Thresholds of 3, 5, and 7 spikes/s are considered with a fixed mean noise level of 1 spike/s. In the case of the 3 spikes/s threshold, a significant number of "responses" were due to noise and an overestimate of a resulted, though this overestimate was still much more accurate than the direct computation in the sparse regime. The 5 and 7 spikes/s thresholds both provided estimates close to the true a, and, most importantly, varied monotonically with a. Figure 2.6 shows the effect of variations in mean noise level on both the binary (a) and direct (b) computations. In the binary case, as the noise

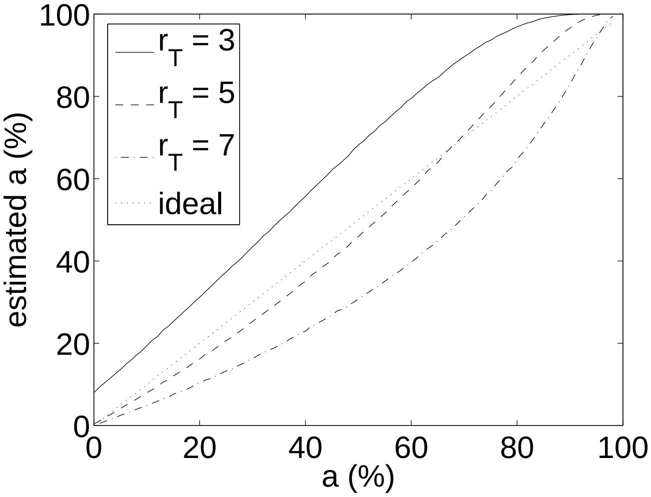
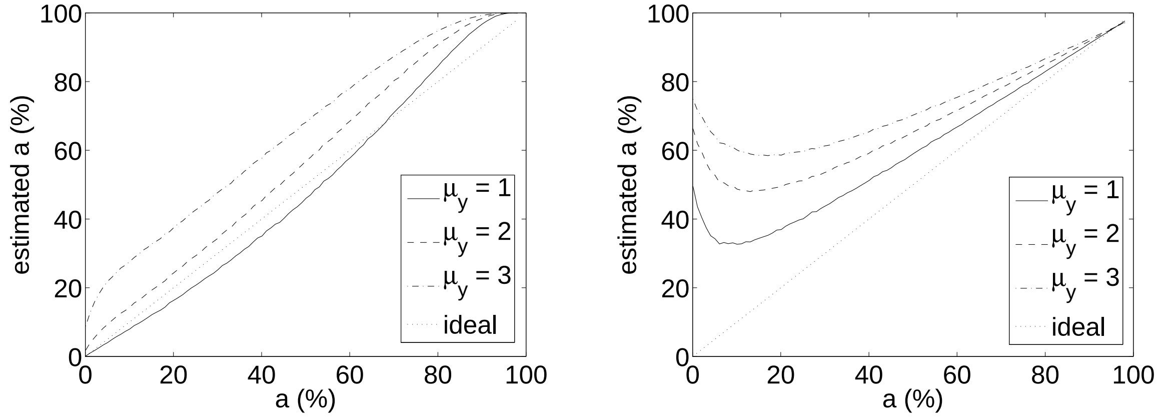


Figure 2.5: Variation of estimated sparseness with threshold. Response thresholds of 3, 5, and 7 spikes/s are considered, with noise level fixed at $\mu_y = 1$ spike/s. Simulated responses are drawn from the same gamma distribution as in Figure 2.4(b).

level drew close to the threshold (fixed at 5 spikes/s), many "responses" were due to noise, resulting in an overestimate of a, but not as severe an overestimate as in the direct computation case. At noise levels of 1 and 2 spikes/s, the estimates were close to the true a. The direct computation provided a much worse overestimate of a, particularly in the sparse regime.

2.2.4 Application to Data

I applied both the binary model and Equation 2.9 to the spiking responses obtained from 1425 human MTL units from 34 experimental sessions in 11 patients (Quian Quiroga et al., 2005; Waydo et al., 2006). This data will be discussed in more detail in the next chapter, but it serves to illustrate the issues I discuss here. Figures 2.7 and 2.8 depict histograms of the results. In Figure 2.7, I calculated for each unit the percentage of stimuli for which the median response was at least 3 standard deviations above its background firing rate (for lower thresholds, many "responses" are a result of random fluctuations; see (Waydo et al., 2006) for further discussion on



(a) Binary computation with threshold fixed at $r_T = 5$ spikes/s

(b) Direct computation

Figure 2.6: Variation of estimated sparseness with mean noise level. Mean noise levels of 1, 2, and 3 spikes/s are considered. Simulated responses are drawn from the same gamma distribution as in Figure 2.4(b).

the choice of threshold). The large majority of these units responded (according to the 3 standard deviation criterion) to less than 2% of presented stimuli, and the mean value of this distribution is 1.5%. This is consistent with the qualitative observation that these units respond in a highly selective manner (Quian Quiroga et al., 2005). By contrast, Figure 2.8 depicts the sparseness estimate $\hat{a}$ computed using Equation 2.9 for the same data. In Figure 2.8(a) I applied Equation 2.9 to the raw firing rates, and the estimate is fairly evenly distributed from 0–100%, with the spike at zero caused by considering an entirely silent unit to have a sparseness of zero. The mean value of $\hat{a}$ computed in this way is 37.8%. In Figure 2.8 I take a common approach to compensating for noise by subtracting each neuron's baseline firing rate from its responses (setting the response to zero in cases where the result is negative). This improves the results somewhat, with estimates now evenly distributed from 0–40%, and a mean of 16.6%.

Recalling the example from Section 2.2.1, in which the numbers were motivated by our experimental data, we see that a true sparseness of 1% can easily lead to a sparseness estimate of 38%. Thus the seemingly contradictory results from Figures

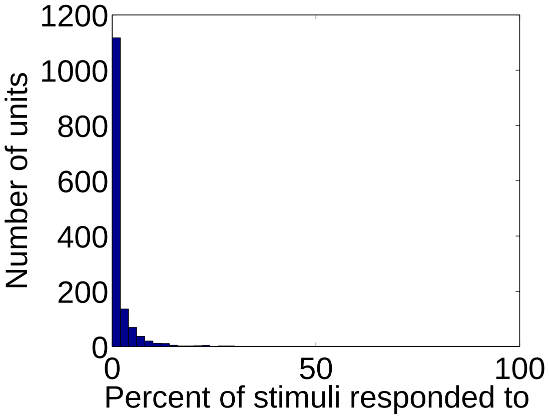
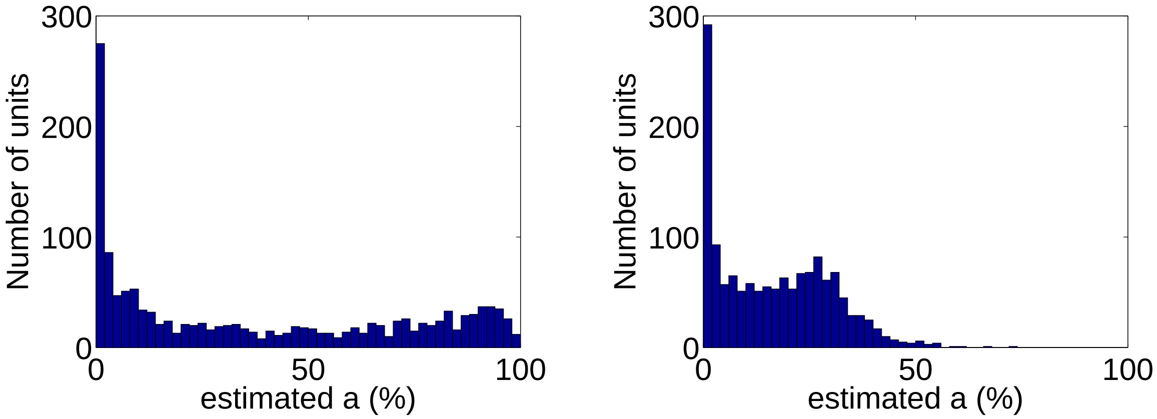


Figure 2.7: Histogram of the percentage of stimuli for which the median response was at least three standard deviations above the background rate computed from spiking responses of 1425 human MTL units. The mean is 1.5%.



(a) Raw firing rates. The mean is 37.8%.

(b) Firing rates with baseline subtracted. The mean is 16.6%.

Figure 2.8: Histograms of estimated sparseness calculated using the direct computation of Equation 2.9 applied to spiking responses of 1425 human MTL units

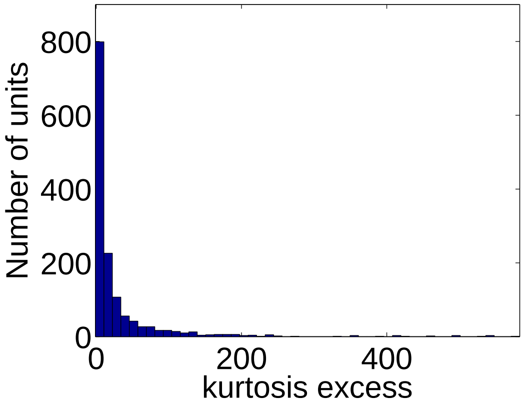


Figure 2.9: Histogram of the kurtosis excess for the responses of 1425 human MTL units. The mean is 29.5.

2.7 and 2.8 are reconciled by the noise sensitivity of Equation 2.9 as described by Equation 2.12. For this reason, and because it matches much better with the qualitative interpretation of the responses as highly sparse (Quian Quiroga et al., 2005) (that is, the observation of highly selective responses to very few stimuli), I believe the sparseness value of 1–2% implied by Figure 2.7 is a much more accurate description of the data.

Figure 2.9 is a histogram of the kurtosis excess for the same set of responses. The kurtosis excess is positive in all but 2% of neurons and has a mean of 29.5, indicating a sparse response distribution in nearly all cases. Beyond this statement, however, the kurtosis gives us little quantitative information about neuronal behavior for the reasons discussed above.

2.2.5 Multiple-Unit Recordings

A limitation of the binary model approach outlined in Section 2.2.2 is that if, for example, two neurons are presented with the same 100 stimuli and neither responds, the true sparseness is likely to be much smaller than that implied by the individual density curves (although the neurons may simply be unresponsive to any stimulus).

As in some recent experiments responses are collected simultaneously from up to several dozen neurons, I extend the binary approach to account for an experiment in which N neurons are recorded simultaneously while S stimuli are presented. Define N_r to be the number of neurons that respond above threshold to at least one stimulus, and S_r to be the number of stimuli that produce a response above threshold in at least one of these. The derivation of the closed-form joint probability distribution of N_r and S_r involves solving a recursive relation for the conditional distribution of S_r given N_r and is described in Appendix A. I simply state the result here:

$$\begin{aligned} P[N_r = n_r \wedge S_r = s_r | a = \alpha] &= \begin{pmatrix} S \\ s_r \end{pmatrix} \begin{pmatrix} N \\ n_r \end{pmatrix} (1-\alpha)^{NS} (-1)^{n_r} \\ & \sum_{k=1}^{n_r} \begin{pmatrix} n_r \\ k \end{pmatrix} (-1)^k \left[(1-\alpha)^{-k} - 1 \right]^{s_r} . \end{aligned} \tag{2.21}$$

As in the single-neuron case discussed above, we can invert this relationship using Bayes' rule to obtain the probability distribution of a given N_r and S_r:

$$f_a(\alpha | N_r = n_r \wedge S_r = s_r) = \frac{P[N_r = n_r \wedge S_r = s_r | a = \alpha] f_a(\alpha)}{\int_0^1 P[N_r = n_r \wedge S_r = s_r | a = \alpha] f_a(\alpha) d\alpha}. \tag{2.22}$$

This gives us the probability density function for a given the results of a full recording session: Rather than obtaining a single curve for each cell, we now obtain a single curve for each session that takes into account the presence of cells that did not respond to any stimulus or that responded to multiple stimuli.

2.2.6 Conclusions

I demonstrated a few of the pitfalls inherent in attempting to estimate sparseness from experimental spike recordings. In particular, I showed that the most direct way of calculating the sparseness can be very sensitive to noise, especially in the case where the true sparseness is small. From these developments I emerge with a few recommendations:

1. If the mean firing rate is on the order of the mean noise level or smaller, the direct computation will be very error-prone and applying the binary model will likely lead to much better results.

2. Noise can be problematic for both methods. Repeated exposure to each stimulus and response averaging should be used to reduce noise levels. Because the noise is likely to have nonzero mean, however (since these are spiking cells), it will produce a bias despite this averaging.

3. When applying the binary calculation, the response threshold should be varied over a wide range to examine how estimated sparseness varies with this threshold.

Chapter 3

Experimental Evidence for Sparseness

In this chapter I describe some of the evidence for sparse coding in biological systems obtained from electrophysiology experiments. In Section 3.1 I survey results from recordings taken throughout the visual system. In Section 3.2 I present my own results (generated in collaboration with Alexander Kraskov, Rodrigo Quian Quiroga, Itzhak Fried, and Christof Koch) from the human medial temporal lobe (MTL), which, though not a visual area, sits at the end of the ventral visual pathway and is linked to associating information across sensory modes and consolidating long-term memory. Finally, in Section 3.3 I discuss a few relevant findings from the sensory processing systems of other organisms.

3.1 The Visual System

Vinje and Gallant (2000) assessed the sparseness of the representation of natural scenes in V1 in awake macaque monkeys. The stimuli were extracted from natural scenes along simulated eye scan paths, and several patch sizes (1–4 times the classical receptive field size) were tested to explore the effect of nonclassical receptive field (nCRF) stimulation on sparseness. The representation became progressively more sparse as the size of the stimulus increased, with a mean of 38% (using the Treves-

Rolls definition of sparseness) for the largest stimuli. Furthermore, the responses of the 11 different neurons recorded became increasingly less correlated with one another as stimulus size increased, and so the increase in sparseness was linked to an increase in the independence of the information transmitted by different neurons. Weliky et al. (2003) obtained similar results in the primary visual cortex of the anesthetized ferret, measuring a mean lifetime sparseness (again using the Treves-Rolls definition) of about 50% in response to large natural images. As in the Vinje and Gallant study, the responses to large-field images were much more sparse than would be predicted by the classical receptive field behavior alone, indicating that a possible role of the nCRF is to increase the sparseness of the V1 visual representation.

Proceeding further along the ventral visual processing hierarchy, Rolls and Tovee (1995) recorded from single neurons in the superior temporal sulcus of the macaque temporal visual cortex, an area known to be selective for faces (Bruce, Desimone & Gross, 1981). Using a set of 23 face and 45 non-face stimuli, Rolls and Tovee measured an average response sparseness (that is, sparseness computed from responses with the mean firing rate subtracted) of 33%. However, they also note that these units were in general highly selective for faces, with many neurons responding to the "best" (most effective) face at a level at least 5 times that of the response to the best non-face. As about 33% of the images were of faces, this selectivity and the statistics of the input set could easily account for the result. The responses to faces were graded rather than binary in nature, and the mean response sparseness with respect to faces was 60%. These results suggest a non-sparse, distributed code for face identity in this area, with little information about non-faces represented. A separate analysis of similar data indicated that the representational capacity for faces in this area grows exponentially in the number of neurons, providing additional evidence for a distributed code (Abbott & Rolls, 1996). These results are consistent with those of Young and Yamane (1992), who also find evidence for a population code for face identity in macaque inferotemporal cortex. Gross (1992) provides evidence that cells in the inferior temporal cortex select for some aspect of shape, texture, or color rather than act as narrow filters for particular stimuli, also categorizing face-selective cells not primarily as face detectors but as part of a distributed code for facial identity. However, it is not clear from the published data whether the responses of these cells were invariant across different pictures of the same face, which would be a necessary feature of a code for facial identity. It could instead be the case that these cells simply make a face/non-face judgement, with the variations in firing rate due to differences in the images rather than due to the identity of each face.

3.2 Medial Temporal Lobe

Single unit recordings from the human MTL have revealed the existence of highly selective cells that may, for example, respond strongly to different images of a single celebrity, but not to 100 pictures of other people or objects (Quian Quiroga et al., 2005). These results suggest a sparse and invariant encoding in MTL and seem to imply the existence of "grandmother cells" that respond to only a single category, individual, or object (Konorski, 1967; Barlow, 1972; Gross, 2002). However, due to limitations on the sampling of MTL neurons and on the sampling of the stimulus space, it is unclear how many stimuli a given neuron will respond to on average and conversely, how many MTL neurons are involved in the representation of a given object. I here use the methods developed in Chapter 2 to explore these issues; this data was previously published in journal form (Waydo et al., 2006). This data was collected by Rodrigo Quian Quiroga (now at the University of Leicester) and Alexander Kraskov (now at University College London) in the lab of Itzhak Fried at UCLA, and this work was performed in collaboration with these individuals and Christof Koch at Caltech; much of it has appeared in journal form (Waydo et al., 2006). My primary contribution was the development of the probabilistic description of the data and the resulting numerical analysis.

Because the representations in this brain area are clearly very sparse, we use the binary model-based approach to estimate the sparseness. As the original data was acquired using 64 microelectrodes, we further make use of the extension to multiple-unit recordings discussed at the end of Chapter 2. This analysis rests on a few key assumptions. First, we assume the responses of all neurons can be treated in a binary fashion, that is, it is reasonable to define a threshold above which we consider a neuron to have responded (and we examine how the results vary with this threshold). Note however that the results of Chapter 2 tell us that even if the neurons truly respond in a more finely graded fashion this is still an accurate approach in the highly sparse regime. Second, we assume the stimulus presentations are independent, and further that the neuronal responses are independent of one another (aside from any stimulus-induced correlations). The independence assumptions are consistent with the observation of no significant correlations between neurons in the experimental data. Finally, we assume that all of our recorded neurons share the same underlying sparseness a. However, as our results are expressed as a probability density function over this value, the width of the density function can be interpreted as describing the range of sparseness present in the MTL.

The data set consisted of recordings of 1425 MTL units from 34 experimental sessions in 11 patients (Quian Quiroga et al., 2005). To fit the data against the binary model, we considered a response to be significant if it was larger than the mean plus a threshold number of standard deviations of the baseline rate and had at least two spikes in the post-stimulus time interval considered (0.3-1 sec) (as in previous work by Quian Quiroga et al. (2005)). The baseline rate was determined by averaging the number of spikes in the 1 second preceding stimulus onset across all trials. Figure 3.1 depicts the resulting probability distributions for thresholds of three and five

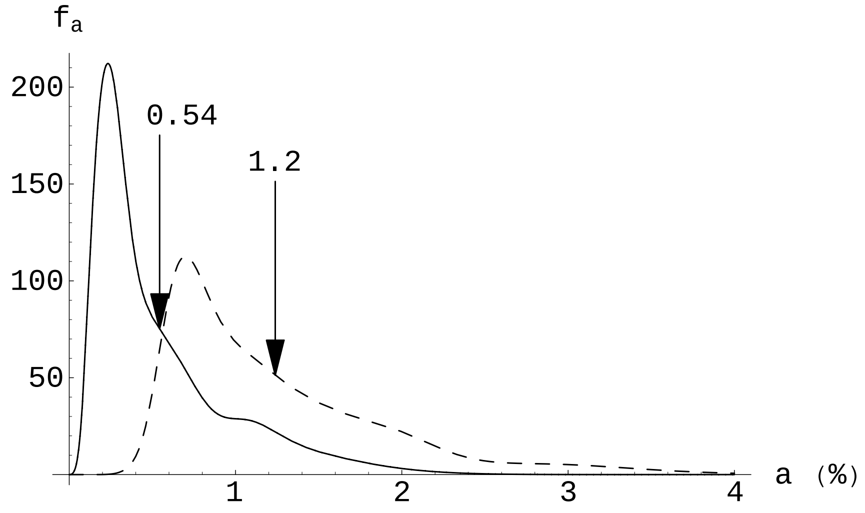


Figure 3.1: Probability density function for sparseness a averaged over 34 experimental sessions that yielded spiking responses from 1425 units. Two different thresholds for defining significant responses are considered: five (solid curve) and three (dashed) standard deviations above baseline. The means of the distributions, corresponding to the best estimates for a, are indicated by arrows, and the values below which a is likely to lie with 95% probability are $a = 1.4\%$ and 2.6%. The peaks of the distributions are at 0.23% and 0.70%. The average number of simultaneously recorded units per session, N, is 41.9 and the mean number of images shown to the subjects, S, is 88.4.

standard deviations; for lower thresholds many of the "responses" are due to random fluctuations in firing rate rather than genuine responses to stimuli. For a threshold of five standard deviations above baseline, the peaks of the 34 individual distributions lie in the range of 0.16–1.64%. For a threshold of three standard deviations above baseline, the individual curves peak in the range of 0.52–3.08%. The peaks of the average distributions shown in Figure 3.1 are at $a = 0.23\%$ and 0.70% for thresholds of five and three standard deviations, respectively, while the means are at $a = 0.54\%$ and 1.2%.

From this figure we conclude that a most likely lies in the range of 0.2–1%. While this is a sparse coding scheme, considering the large number of MTL neurons and the large number of represented stimuli, it still results in a single unit responding to many stimuli, and many MTL units responding to each stimulus. We assume, however, that all cells we are listening to are involved in the representation of some stimulus, which may not be the case (i.e., some of them could serve a different function altogether) and which could cause a downward bias in our estimate. To quantify a plausible magnitude for this bias, we repeated the same analysis leaving out half of the unresponsive neurons, that is, we used

$$N' = N_r + \frac{N - N_r}{2}$$

in place of N. This analysis showed the potential bias to be small, as it yielded most likely values for a of 0.9% and 1.8% at thresholds of five and three standard deviations, respectively.

We can then estimate the probability of finding such highly selective cells in a given experiment. If the true sparseness is 0.54% (the mean of the distribution with a threshold of 5), in a typical session with $N = 42$ simultaneously recorded units and $S = 88$ test stimuli (the averages from our experiments), we would expect to find on average 15.9 units responding to 17.9 stimuli (with each responsive neuron responding on average to 1.3 images, and each evocative stimulus producing a response in an average of 1.1 neurons). In our experiments N ranged from 18 to 74 and S ranged from 57 to 114, and with a five-standard-deviation threshold we found on average 7.9 responsive units (range: 3 to 20) responding to 16.4 stimuli (range: 3 to 44). As a further check of our methods, we can examine how frequently two or more units responded to the same stimulus. At a five-standard-deviation threshold, on average 4.1% of stimuli produced a (simultaneous) response in at least two neurons (range: 0 to 17.9%; median: 1.6%), compared to a predicted value (at 0.54% sparseness) of 2.2%. Noting that we cannot expect perfect agreement between this prediction and the observed value because of the varying numbers of neurons and stimuli across recording sessions, we see that our model agrees very well with the observed statistics.

We developed a method for obtaining a probability distribution for sparseness based on multiple simultaneous neuronal recordings. This distribution allows us to not only examine the average sparseness observed in a given experiment, but also the range of sparseness consistent with the data. Averaging these distributions over 34 recording sessions in the human medial temporal lobe, we conclude that highly sparse (though not grandmother) coding is present in this brain region.

To animate this discussion with some numbers, consider 0.54% sparseness level. Assuming on the order of 10^9 neurons in both left and right human medial temporal lobes (Harding, Halliday & Kril, 1998; Henze et al., 2000), this corresponds to about 5 million neurons being activated by a typical stimulus, while a sparseness of 0.23% implies activity in a bit more than 2 million neurons. If we furthermore assume that a typical adult recognizes between $10,000$ and $30,000$ discrete objects (Biederman, 1987), $a = 0.54\%$ implies that each neuron fires in response to 50–150 distinct representations.

This interpretation relies on the assumption that the cells from which we record are part of an object representation system. Instead, it may be possible that these neurons signal recency or familiarity rather than the identity of a stimulus. Neurons responding to both novelty and familiarity have been identified in the human hippocampus (Fried, MacDonald & Wilson, 1997; Rutishauser, Mamelak & Schuman, 2006; Viskontas, Knowlton, Steinmetz & Fried, 2006) (and see Rolls, Perrett, Caan, and Wilson (1982) for related results in monkeys). Even if true, however, this view does not invalidate our conclusion that the true sparseness likely lies below 1%. Instead, it would imply that rather than a single neuron responding to dozens of stimuli out of a universe of tens of thousands, such a neuron might respond to only one or a few stimuli out of perhaps hundreds currently being tracked by this memory system, still with millions of neurons being activated by a typical stimulus. Further, Rolls, Xiang, and Franco (2005) identified neurons in macaque hippocampus and posterior perirhinal cortex responding to specific objects, places, and object-place combinations while the animal performed an object-place association task. Because only two objects and two places were used in each experiment it is impossible to assess the sparseness of this representation, but the results at least suggest units selective for specific stimuli as part of an episodic memory system.

These numbers are consistent with the results of Lennie (2003), who, building on earlier work in rats by Attwell and Laughlin (2001), used detailed estimates of cortical metabolism and the energy cost of spiking to calculate the maximum activity level possible in human cortex. Lennie concluded that the cortical metabolism can support an average spike rate of 0.80 spikes/s/neuron. Alternatively, if an "active" neuron fires at 50 spikes/s, then only about 1.6% of neurons could be active at the same time, and even fewer if the inactive neurons still maintained some small resting firing rate. These results imply that, aside from any computational considerations, the cortical metabolism can only support sparse codes in which only a small fraction of neurons are simultaneously active. This analysis may not be enough to justify sparse coding in and of itself, however—if it were significantly advantageous to utilize a distributed code in which more neurons were simultaneously active, then it is easy to imagine that the cortical metabolism would have evolved to support a higher energy consumption rate. It may be that the cortical metabolism can only support a sparse code because (as I will argue in the following chapters) a sparse code is computationally useful and so a more vigorous metabolism is unnecessary.

Two significant factors may bias our estimate upward. A large majority of neurons within the listening radius of an extracellular electrode are entirely silent during a recording session: there are as many as 120 to 140 neurons within the sampling region of a tetrode in the CA1 region of the hippocampus (Henze et al., 2000), but we typically only succeed in identifying 1–5 units per electrode. In rats as many as 2 out of 3 cells isolated in the hippocampus under anesthesia may be behaviorally silent

(Thompson & Best, 1989), though the reason for their silence is unclear. Thus, the true sparseness could be considerably lower. Furthermore, there is a sampling bias in that we present stimuli familiar to the patient (e.g., celebrities, landmarks, and family members) that may evoke more responses than less familiar stimuli. For these reasons these results should be interpreted as an upper bound on the true sparseness, and some neurons may provide an even sparser representation.

These results are consistent with Barlow's claim that "at the upper levels of the hierarchy a relatively small proportion [of neurons] are active, and each of these says a lot when it is active," and his further speculation that the "aim of information processing in higher sensory centres is to represent the input as completely as possible by activity in as few neurons as possible" (Barlow, 1972).

3.3 Sparseness Elsewhere

I here describe a few other interesting examples of sparse coding identified in different organisms.

3.3.1 Place Cells

O'Keefe and Dostrovsky (1971) discovered *place cells* in the rat hippocampus, which are silent most of the time but fire more vigorously when the rat is in a particular location (known as the cell's *place field*), a result verified by numerous investigators in the years since. Collectively, the place fields of hippocampal neurons form a map of the environment that could be used for various tasks such as navigation and association of places to memories. For example, place fields have been seen to be modulated by spatial cues (that is, stay tied to the cues rather than the physical environment) during a spatial memory task (O'Keefe & Speakman, 1997). As each place cell is highly selective for a specific location, these cells form a sparse representation for location much like the human MTL cells discussed above form a sparse representation for object category or identity. Extending these results to very different species, Ulanovsky and Moss (2007) recorded from hippocampal area CA1 of bats during a foraging task, finding well-defined place fields in a majority of active units. Remarkably, these place fields were generally stable between recording sessions even when different sessions allowed the use of different sensory modalities (vision and echolocation), a form of invariance perhaps related to that observed in human MTL.

Hafting and colleagues (2005) may have illuminated part of the computation that gives rise to place cells, recording earlier in the processing hierarchy in the dorsocaudal region of the medial entorhinal cortex (dMEC) of rats. Here they found units with multiple place fields, which each unit's place fields organized in a regular triangular grid. Spacing, orientation, and field size of these grids varied as a function of recording location. It is likely that a sparse coding strategy much like that discussed in the context of vision in later chapters could give rise to place cells when applied to inputs from such "grid cells."

3.3.2 Insect Olfaction

Sparsening of responses to stimuli as one proceeds up the processing hierarchy has also been observed in insect sensory systems. Perez-Orive and colleagues (2002) measured responses of neurons in locust antennal lobe (AL), which receives input directly from olfactory receptors, and the mushroom body (MB), the next stage in olfactory processing. In an experiment roughly analogous to our own human MTL experiments, they presented a panel of odors to the locusts and recorded spiking activity. Information about the odor presented appears to be coded across the population of AL neurons, with an average response probability (that is, the likelihood that a particular odor will elicit a strong response in a neuron) of multiglomerular projection neurons (which are the only pathway for olfactory input to MB) of 64%. At the next stage, however, the representation was considerably more sparse: the average response probability of Kenyon cells in MB was only 11%, and 58% of neurons recorded failed to respond to any odor tested. As only 5 to 24 odors were presented in a particular experiment, this means a typical MB neuron responded to only 1 to 3 of the presented odors. Clearly the information carried at the population level in AL has been made much more explicit in MB.

3.3.3 Temporal Sparseness

The issues of lifetime and population sparseness investigated above are distinct from the related notion of *temporal* sparseness, in which individual neurons may fire only a very small number of spikes in response to a stimulus or as part of a precise time sequence. In the extreme case, a neuron may send a signal using only a single spike. DeWeese, Wehn, and Zador (2003) recorded single neurons in the auditory cortex of ketamine-anesthetised rats as they responded to pure tones. They found neurons behaving in a nearly perfect binary fashion, responding to each stimulus with either zero or one spikes. The probability of spiking in response to a particular stimulus was a function of the tone frequency, approaching 1 at a specific preferred frequency and falling off rapidly away from it. These neurons display sparse selectivity in the sense defined above in that they respond selectively to a very specific stimulus (frequency), as well as extreme temporal sparseness in that they generally fire only a single spike in response to even an "optimal" stimulus. DeWeese and colleagues also suggest that these results demonstrate a more precise control of spike number (or equivalently, the presence of less noise) than is generally assumed possible. It may be that much of the "noise" observed in cortex (including in our own data) is in fact stimulus driven rather than a reflection of an inherently noisy computational system, and that the ability to stimulate auditory cortex with a precise signal exposes the true precision of sensory cortex.

Hahnloser, Kozhevnikov, and Fee (2002) recorded single units in the high vocal center (HVC) of zebra finches, finding units that burst just once during an approximately 1 second song motif. These bursts consisted of about 5 spikes and were time-locked to the song motif with better than 1 ms accuracy. The units were virtually silent at other times, with a background rate in awake, non-singing birds less than 0.001 spikes/s. The authors suggest that this precision is the temporal analogue of the grandmother cell. Fiete and colleagues (2004) created a neural network model of birdsong and found that this high level of sparsity speeded learning due to the decreased interference between different patterns (i.e., different points in time) when the patterns are sparse.

3.4 Conclusion

All of the results discussed in this chapter support the notion that a goal of sensory processing is to represent the sensory world in a compact, sparse manner. That is, sensory cortex transforms the behaviorally important information present only implicitly at the periphery into an explicit representation in central structures in which the activity of (relatively) small numbers of neurons carries a great deal of information about the outside world. Individual neurons are then *feature detectors* (Martin, 1994) that indicate the presence of different sensory features in the input stream, with feature complexity increasing as one progresses along the hierarchy. Again, Barlow (1972) expressed this idea most clearly:

> The central proposition is that our perceptions are caused by the activity of a rather small number of neurons selected from a very large population of predominantly silent cells. The activity of each single cell is thus an important perceptual event and it is thought to be related quite simply to our subjective experience. The subtlety and sensitivity of perception re-

> sults from the mechanisms determining when a single cell becomes active, rather than from complex combinatorial rules of usage of nerve cells.

In the next chapters I will examine computational methods by which sparse, explicit representation may be achieved in a neurally plausible manner, and the results obtained from applying such a model to visual information processing.

Chapter 4

Models for Sparse Coding

In this chapter I discuss the manner in which sparse representations can be generated from sensory information. In Section 4.1 I introduce the general idea of a *generative model* and how such models are computed. I then discuss how this framework has been used to develop a model of sparse coding that has successfully reproduced a V1-like code for natural images in Section 4.2. In Section 4.3 I describe the extensions I have made to this model for the particular type of coding I seek to reproduce. Computational results from the application of these methods to visual processing will be presented in Chapter 5. Finally, in Section 4.4 I discuss a few other unsupervised learning algorithms and their relationships to sparse coding.

4.1 Generative Models

The framework I will use to develop a computational model for sparse coding is that of a *generative model*. I provide here an overview of generative models sufficient to motivate my results; for a more extensive discussion see Dayan and Abbot (2001, Chapter 10), from which I derive much of the following notation.

Our model of the world is one in which a random process generates *causes* $V \in \mathbb{R}^m$ according to some distribution $f_V(v)$. These causes in turn generate *inputs* $U \in \mathbb{R}^n$ according to the marginal distribution $f_U(u|v)$. The inputs are the observable quantity, and in general are of much higher dimension than the causes, so $n \gg m$. Our eventual goal will be recognition, in which an estimate of the cause $\hat{v}(u)$ (or a distribution of causes $f_V(v|u)$) can be determined for any particular observed input u. We further desire to carry out this inference in an *unsupervised* manner—no information about the specific causes underlying the observed inputs will be provided to the model. Instead, the model must extract the cause estimates solely from the statistics of the inputs u, subject to a set of *heuristics*, or assumptions on the structure of the data and causes, that we place on the model. In general we view a particular cause as potentially giving rise to a great many different inputs (or, in recognition, a particular cause could be attributed to many different inputs). For example, the collection of objects in an image would be considered the cause of the image, but many different images (inputs) could be produced by the same set of objects.

I use the symbol $\mathcal{G}$ to stand for all of the (yet-to-be-specified) parameters and assumptions of our generative model. In general, the model $\mathcal{G}$ is characterized by two distributions: the *generative distribution* $f_U(u|v, \mathcal{G})$ by which causes generate inputs, and the *prior distribution* of causes $f_V(v|\mathcal{G})$, the distribution according to which the causes themselves occur. In other words, the prior distribution is a model of the statistical structure of the outside world, while the generative distribution is a model of the sensory process. The prior distribution and generative distribution together define the *marginal distribution* of inputs within the model,

$$f_U(u|\mathcal{G}) = \int_v f_U(u|v, \mathcal{G}) f_V(v|\mathcal{G}). \tag{4.1}$$

The goal is then to find a model $\mathcal{G}$ for which the distribution of inputs generated by the model closely matches the observed distribution of inputs, or

$$f_U(u|\mathcal{G}) \approx f_U(u). \tag{4.2}$$

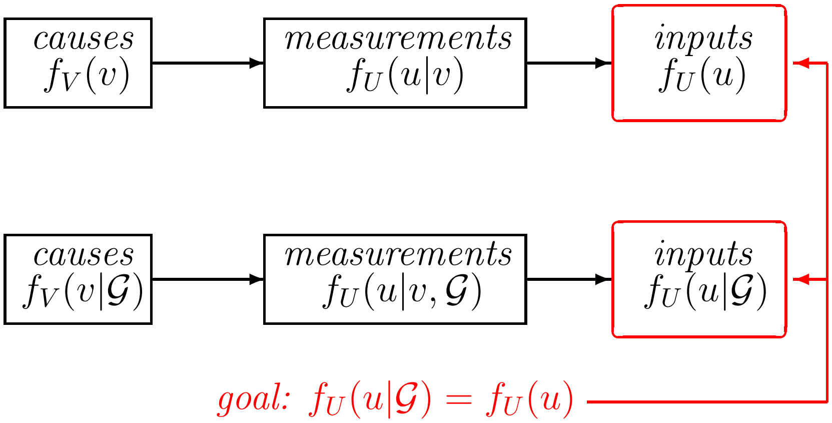


Figure 4.1: World model (top) and the generative model (bottom) that attempts to match its behavior

Figure 4.1 depicts this assumed structure of the world and the generative model designed to match the world's behavior.

Once such a model has been obtained, we can use it to estimate the causes underlying the individual data points u. Applying Bayes' rule we obtain the *recognition distribution*

$$f_V(v|u,\mathcal{G}) = \frac{f_U(u|v,\mathcal{G})f_V(v|\mathcal{G})}{f_U(u|\mathcal{G})}. \tag{4.3}$$

From this distribution we can compute the expected or most likely cause underlying a particular input u and use that as our estimate $\hat{v}$. In many cases, however, the integrals involved in evaluating Equation 4.3 are computationally intractable (that is, it is impractical to evaluate Equation 4.1) and the model $\mathcal{G}$ is called *noninvertible*. We then rely on an *approximate recognition distribution* $q_V(v|u,\mathcal{G})$ and include as part of our optimization of $\mathcal{G}$ improving the fit of our approximate recognition distribution to the true recognition distribution. Ultimately we would like

$$q_V(v|u,\mathcal{G}) \approx f_V(v|u,\mathcal{G}). \tag{4.4}$$

By placing various assumptions on the structure of $\mathcal{G}$, we can obtain different types of coding. The nature of the true underlying causes of the observed data determines what coding strategy will be most appropriate.

4.1.1 Expectation Maximization

The method we shall use to optimize our generative models (that is, to attempt to satisfy Equation 4.2) is known as *expectation maximization* (EM), introduced by Dempster, Laird, and Rubin (1977). In our setting, EM is based on maximizing the function

$$\mathcal{F}(q,\mathcal{G}) = \left\langle \int_v q_V(v|u,\mathcal{G}) \log \frac{f_{VU}(v,u|\mathcal{G})}{q_V(v|u,\mathcal{G})} dv \right\rangle, \tag{4.5}$$

where the expectation $\langle\cdot\rangle$ is taken over all observed inputs u. Noting that Bayes' rule states that $f_{VU}(v,u|\mathcal{G}) = f_V(v|u,\mathcal{G})f_U(u|\mathcal{G})$ and with some rearrangement of terms we can see why $\mathcal{F}$ is a useful quantity to consider:

$$\begin{aligned}
\mathcal{F}(q,\mathcal{G}) &= \left\langle \int_v q_V(v|u,\mathcal{G}) \log f_U(u|\mathcal{G}) dv - \int_v q_V(v|u,\mathcal{G}) \log \frac{q_V(v|u,\mathcal{G})}{f_V(v|u,\mathcal{G})} dv \right\rangle \\
&= \langle \log f_U(u|\mathcal{G})\rangle - \left\langle \int_v q_V(v|u,\mathcal{G}) \log \frac{q_V(v|u,\mathcal{G})}{f_V(v|u,\mathcal{G})} dv \right\rangle \\
&= \langle \log f_U(u|\mathcal{G})\rangle - \big\langle D_{KL}\big(q_V(v|u,\mathcal{G}), f_V(v|u,\mathcal{G})\big)\big\rangle, \qquad (4.6)
\end{aligned}$$

where D_{KL} is the *Kullback-Leibler divergence* (Kullback & Leibler, 1951), which measures how different two probability distributions are from one another. The simplification of the first term occurs because $f_U(u|\mathcal{G})$ does not depend on v, and $q_V(v|u,\mathcal{G})$ integrates to one since it is a probability distribution. Thus $\mathcal{F}$ has two very meaningful terms. The first term is the average log-likelihood that the model would generate the observed inputs, and so it rewards a model that generates the observed data with high probability. The second term penalizes the discrepancy between the approximate recognition distribution $q_V(v|u,\mathcal{G})$ and the true recognition distribution $f_V(v|u,\mathcal{G})$. In the case of an invertible model (so $f_V(v|u,\mathcal{G})$ is known), the second term vanishes and we are left simply maximizing the likelihood of the model generating the data.

Expectation maximization separately maximizes $\mathcal{F}$ with respect to its two arguments $\mathcal{G}$ and q. In the expectation (E) phase, $\mathcal{F}$ is increased with respect to q_V by improving the fit of q_V to the true recognition distribution, holding the other parameters of $\mathcal{G}$ constant. In the maximization (M) phase, $\mathcal{F}$ is increased with respect to $\mathcal{G}$ while holding q_V constant. In the cases we will consider, optimization proceeds in a series of alternating E and M phases.

When we consider sparse coding we will further restrict ourselves to *deterministic recognition*, in which for a given input u we compute a specific estimate of the underlying cause, $\hat{v}(u)$. In this case we consider the limit of Equation 4.5 as $q_V(v|\mathcal{G})$ approaches the Dirac δ-function $\delta(v - \hat{v}(u))$, obtaining

$$\mathcal{F}(\hat{v}(u), \mathcal{G}) = \langle \log f_{VU}(\hat{v}(u), u|\mathcal{G}) \rangle. \tag{4.7}$$

In the E phase we find the function $\hat{v}(u)$ that maximizes $\mathcal{F}$, while in the M phase we maximize with respect to $\mathcal{G}$ as before. This structure has the simple interpretation that we are trying to maximize the probability that our model $\mathcal{G}$ would have simultaneously produced the inputs u and causes $\hat{v}(u)$.

4.2 Sparse Coding with a Generative Model

With an appropriate structure on $\mathcal{G}$, the approach described above can be used to learn a sparse code for the inputs u, that is, a code in which the individual coding elements are active only rarely. This technique was first used by Olshausen and Field to generate a code for natural images resembling the oriented bar filters observed in V1 (Olshausen & Field, 1996, 1997). As my results build on an extension of their work, I will summarize it here (though we adopt the notation of Dayan and Abbott (2001) to map better to the extensions described later).

Olshausen and Field developed a model in which the inputs u were assumed to be a linear function of the unknown underlying causes v plus additive, zero-mean Gaussian noise, so

$$u = Gv + \xi, \tag{4.8}$$

where ξ is a zero-mean Gaussian random variable with covariance λI. The columns of $G \in \mathbb{R}^{n \times m}$ can be viewed as (non-orthogonal) basis functions for u. They further assume that the individual causes $v_i \in \mathbb{R}$ (that is, the individual elements of the cause vector) are sparse, independent, and identically distributed, defining the *sparse prior* distribution

$$f_V(v) \propto \prod_{i=1}^{m} \exp(S(v_i)), \tag{4.9}$$

where S is a function designed such that $f_V(v)$ is sparse. In this context, sparse means that both large responses and small ones are more likely than under a Gaussian distribution. The exponential form of the prior is chosen for mathematical convenience. For simplicity I omit the proportionality constant required to make this distribution integrate to 1 (this constant would drop out of the forthcoming optimization, so there is no loss of generality). In Olshausen and Field (1997), where this strategy was used to develop a visual cortex-like sparse code for natural images, the sparse prior S followed a Cauchy distribution with $S(v) = -\log(1 + v^2)$.

The problem of optimizing the model is now reduced to finding the weight matrix G that maximizes the average log likelihood of the observed inputs u. Ideally one would like to find the matrix G^* such that

$$G^* = \arg\max_G \left\langle \log \left(\int_v f_U(u|v, G) f_V(v) dv \right) \right\rangle, \tag{4.10}$$

where the expectation $\langle \cdot \rangle$ is taken over all inputs u. However, the integral within the optimization is in practical terms intractable (to perform for each of perhaps

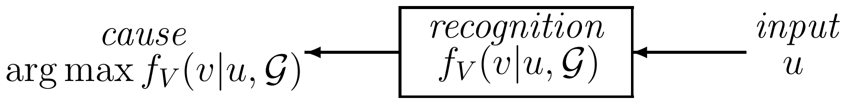


Figure 4.2: Recognition model derived from the generative model of Figure 4.1

thousands of input stimuli for each G). Olshausen and Field instead take the deterministic recognition approach, approximating the integral by its maximum value (a valid approach if it has a tightly localized peak), finding

$$G^* = \arg\max_G \left\langle \max_v \log(f_U(u|v,G)f_V(v)) \right\rangle . \tag{4.11}$$

Substituting in the assumed linear (plus noise) relationship between u and v and the sparse prior, we find the function to be maximized is

$$\begin{aligned} \mathcal{F}(\hat{v}(u),G) &= \langle \log(f_U(u|\hat{v}(u),G)f_V(\hat{v}(u))) \rangle \qquad (4.12) \\ &= \left\langle -\frac{1}{2\lambda}\|u - G\hat{v}(u)\|^2 + \sum_{i=1}^{m} S(\hat{v}_i(u)) \right\rangle + C, \end{aligned}$$

where $\hat{v}(u)$ is the most likely cause given the input u (and so is our estimate of the true cause v), or

$$\hat{v}(u) = \arg\max_v f_U(u|v,G)f_V(v). \tag{4.13}$$

This deterministic recognition process is illustrated in Figure 4.2.

Inspecting the cost function $\mathcal{F}$, we find that it has two meaningful parts. The first penalizes $\|u - G\hat{v}(u)\|$, the mismatch between the generated input $G\hat{v}(u)$ and the actual input u, and so expresses how well the current model represents the input set. The second penalizes causes $\hat{v}(u)$ that are unlikely under the sparse prior $S(v)$. Thus the optimization seeks to find a set of basis functions G that describe the data well, subject to the sparseness constraint. The constant C is the result of the proportionality factors required so that the various distributions integrate to one, and does not effect the optimization, and so will be dropped.

The optimization of $\mathcal{F}$ is carried out via the two-step expectation-maximization process outlined above. In the first (E) step, we compute the cause estimate $\hat{v}$ for a particular input u drawn from some set of inputs $\{u\}$ (solving Equation 4.13). Taking the derivative of $\mathcal{F}$ with respect to $\hat{v}$ and setting it equal to zero we find

$$0 = G^T(u - G\hat{v}) + \lambda S'(\hat{v}), \tag{4.14}$$

where $S'(\hat{v})$ is shorthand for the vector whose i^{th} component is $S'(\hat{v}_i)$ and the prime denotes the derivative of S with respect to its argument. This equation can be solved by simulating the differential equation

$$\dot{v} = G^T(u - Gv) + \lambda S'(v) \tag{4.15}$$

until it reaches some equilibrium point v_∞, and setting $\hat{v} = v_\infty$. In Appendix B I show that it is always the case that $\dot{v} \to 0$ as $t \to \infty$, and so in practical terms this process always converges. If one makes the additional assumption that all solutions of Equation 4.14 are isolated, then this also implies that Equation 4.15 converges to one such solution (and this has been the case in all test cases in the following chapter).

Equation 4.15 can be interpreted as the two-layer recurrent neural network pictured in Figure 4.3. The neurons in the lower layer compute the reconstruction error $u - Gv$, while those in the upper layer compute the most likely cause v. The neurons are of two different types: the "error" neurons in the lower layer simply output the sum of their inputs (with no internal dynamics), while those in the upper "output" layer have dynamics that integrate their inputs while incorporating a nonlinear self-inhibition term given by S'. The recurrent feedback ($-G^TG$) term introduces competition between output units that represent similar inputs, producing winner-take-all behavior, and this stage of the optimization can be viewed as computing the set of basis functions that best represent the input, subject to the sparseness

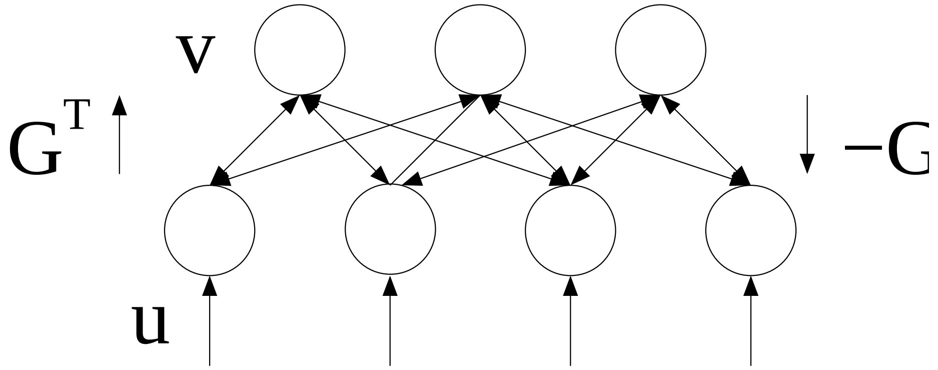


Figure 4.3: Neural network implementation of Equation 4.15. The bottom-layer neurons compute the reconstruction error $u - G\hat{v}$, while the upper layer outputs the causes v. The learning rule (Equation 4.16) is a Hebbian rule for this network.

constraint imposed by the self-inhibition term S'. Note that the sparse prior only enters the dynamics through the self-inhibition term, not through any interactions between neurons. If one wished to alter the response probability of a given neuron to reflect changing assumptions about the world (for example due to some top-down attentional effect), one would only need to change that neuron's inhibition term.

For a particular input u and cause estimate $\hat{v}(u)$ computed as above, Olshausen and Field performed gradient-ascent learning to improve G, which results in the update rule

$$G \to G + \frac{\delta}{\lambda}(u - G\hat{v}(u))\hat{v}(u)^T, \tag{4.16}$$

where δ is a small, positive learning constant. Expanding this learning rule to look only at the update of the single connection between error neuron i and output neuron j in Figure 4.3, we obtain

$$\Delta g_{ij} = \frac{\delta}{\lambda}(u_i - \sum_j G_{ij}\hat{v}_j)\hat{v}_j. \tag{4.17}$$

In the network of Figure 4.3 (which implements the dynamics of Equation 4.15), the

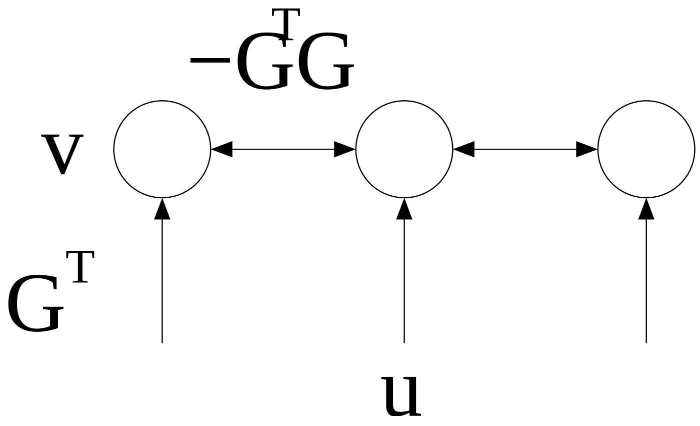


Figure 4.4: Alternative network implementation of Equation 4.15

output of all the error neurons is

$$e = u - G\hat{v}, \tag{4.18}$$

so the output of error neuron i is

$$e_i = u_i - \sum_j G_{ij}\hat{v}_j. \tag{4.19}$$

Thus the update of synaptic weight ij is proportional to the product of the presynaptic input and the post-synaptic response, and this is a standard Hebbian learning rule for the network in Figure 4.3.

The network depicted in Figure 4.3 is not the only possible network implementing the dynamics of Equation 4.15. These dynamics would also be implemented by the network depicted in Figure 4.4, in which the input u is passed through the weight matrix G^T directly into the output ($\hat{v}$) layer, which comprises neurons identical to those in the upper layer of Figure 4.3 connected with recurrent weights $-G^T G$. The learning rule given by Equation 4.16 is not Hebbian for this alternate network, however, and so this topology does not have as straightforward a biological interpretation.

One possible weakness of the network interpretation is that there is a constraint imposed that the feedforward weights must match the feedback weights (transposed). However, if the network were initialized with different feedforward weights G and feedback weights $-H$, Hebbian learning would give $\Delta h_{ij} = \Delta g_{ji}$. To verify this, first note that the output of error neuron i is

$$e_i = u_i - \sum_j H_{ij} \hat{v}_j,$$

while its pre-synaptic input from output neuron j is simply $\hat{v}_j$. The Hebbian weight update is then

$$\Delta h_{ij} = \frac{\delta}{\lambda}(u_i - \sum_j H_{ij} \hat{v}_j) \hat{v}_j.$$

Meanwhile, the output of output neuron j is $\hat{v}_j$, while its pre-synaptic input from input neuron i is $u_i - \sum_j H_{ij} \hat{v}_j$, so the Hebbian weight update is

$$\Delta g_{ji} = \frac{\delta}{\lambda}(u_i - \sum_j H_{ij} \hat{v}_j) \hat{v}_j.$$

Hence $\Delta H = (\Delta G)^T$. If a decay term were then included in the weight update (as described in Section 4.3.2 below) the two weight matrices would converge to the same solution over time (as their dynamics would be described by stable linear systems with the same dynamics and inputs, differing only in initial conditions).

Olshausen and Field point out that there is a penalty inherent in approximating the integral of Equation 4.10 with the maximum operation of Equation 4.11, namely that there will be a trivial solution for G, since the larger G is the smaller $\hat{v}(u)$ can be, and so the larger $f_V(v)$ can be (assuming that $f_V(v)$ increases as v approaches zero). Without further constraints, then, G could grow without bound while trending toward a good set of basis vectors. This problem was alleviated by adapting the length of the basis functions (in our notation, the columns of G) to maintain the variance of the individual cause estimates $\hat{v}_i$ at a desired level. In Section 4.3.3 I discuss another method of mitigating this issue that fits directly in the optimization and may have a cleaner biological interpretation.

4.3 Extensions

I here describe several extensions to the original Olshausen and Field sparse coding model besides the application of the model to a much higher level in the visual hierarchy described in the next chapter; these extensions compose my main theoretical contribution in this area. This section expands on work that has recently been published elsewhere (Waydo & Koch, 2007a, 2007b).

4.3.1 Bimodal Sparse Prior

I would like to adapt the approach of Olshausen and Field to generate sparse, invariant representations of objects in the visual world like those observed in human MTL (Quian Quiroga et al., 2005; Waydo et al., 2006, and see Chapter 3). Olshausen and Field used a Cauchy prior distribution for v, which is sparse in the sense that it will generate more responses close to zero and far from it than a Gaussian. Because the neuronal behavior I would like to replicate is more binary in nature (i.e., responses are either “off,” near zero, or “on,” near some large firing rate), I employ a different sparse prior that reflects this desire. The prior distribution I choose is a weighted average of Gaussians centered at zero and some higher “on” rate r_h. Denoting the probability that the neuron responds strongly by a and the desired variance as σ^2, my sparse prior is

$$\begin{aligned} f_V(v) &= \frac{1-a}{\sqrt{2\pi}\sigma} e^{\frac{-v^2}{2\sigma^2}} + \frac{a}{\sqrt{2\pi}\sigma} e^{\frac{-(v-r_h)^2}{2\sigma^2}} \\ &= \alpha e^{\frac{-v^2}{2\sigma^2}} + \beta e^{\frac{-(v-r_h)^2}{2\sigma^2}}. \end{aligned} \tag{4.20}$$

The constants α and β are introduced to simplify notation. In principle the variance σ^2 could be different for the two Gaussians, though this modification has not been necessary. It would also be desirable to impose the constraint that neuronal responses are always greater than zero (both for biological realism and to ease interpretation of the results), which corresponds to $f_V(v) = 0$ for $v < 0$. To avoid numerical difficulties stemming from the resulting discontinuity, I instead impose a very narrow Gaussian prior distribution for $v < 0$, which will result in negative responses being strongly pushed toward zero. In the following development I assume $v \geq 0$ and omit this detail.

To use this formulation within the framework described above, I define $S(v)$ such that $\exp(S(v)) = f_V(v)$, or

$$S(v) = \log\left(\alpha e^{\frac{-v^2}{2\sigma^2}} + \beta e^{\frac{-(v-r_h)^2}{2\sigma^2}}\right). \tag{4.21}$$

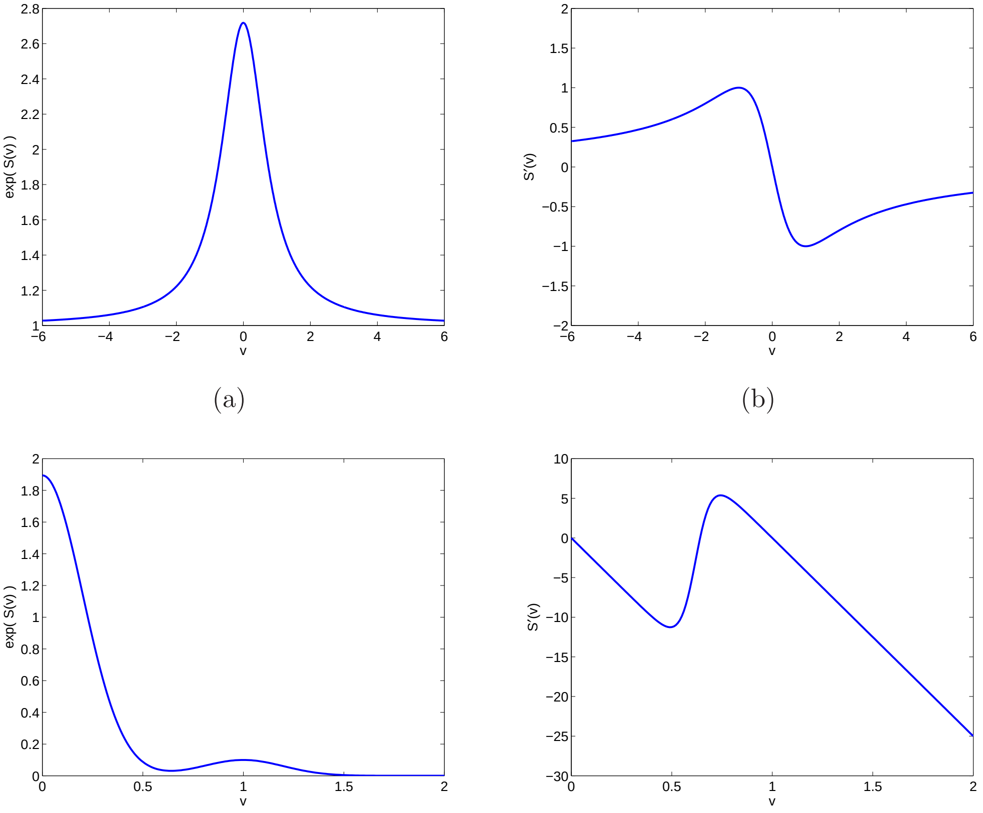
Taking the derivative of S with respect to v gives us the needed function describing the neuronal dynamics,

$$S'(v) = -\frac{v}{\sigma^2} + \frac{\beta r_h}{\sigma^2}\frac{1}{\alpha + \beta e^{\frac{2r_h v - r_h^2}{2\sigma^2}}} e^{\frac{2r_h v - r_h^2}{2\sigma^2}}. \tag{4.22}$$

Figure 4.5 provides a comparison between this form of $S'(v)$ and that used by Olshausen and Field (as well as the corresponding sparse priors $\exp(S)$), where

$$S'(v) = -\frac{2v}{1+v^2}. \tag{4.23}$$

The differences in scale between the two approaches are not important, as the two models are driven by different inputs, but the difference in the overall shape is crucial. In the Olshausen and Field approach (Equation 4.23), small responses are linearly suppressed (i.e., $S'(v) \propto v$ for small v), while the self-inhibition becomes small for large v. In our case (Equation 4.22), small responses are linearly suppressed toward zero, while larger responses are linearly suppressed toward r_h (in this figure, $r_h = 1$), giving rise to a bimodal response distribution.



(a)

(b)

(c)

(d)

Figure 4.5: Comparison of sparse prior $\exp(S)$ (a, c) and the derivative S' (b, d) between the approach of Olshausen and Field and that taken here. The differences in scale are related to the differences in scale of our input sets and are unimportant; the overall shape of the curve determines the resulting distribution of responses. (a, b): Olshausen and Field approach (Equation 4.23). (c, d): Our approach (Equation 4.22). Only the portion for $v \geq 0$ is shown, as we will later restrict ourselves to this regime.

4.3.2 Weight Penalty

I also extend the approach of Olshausen and Field to incorporate a prior probability distribution on the elements g_{ij} of the weight matrix G to express the constraint that the weights not be too large. I do so by placing a zero-mean Gaussian prior distribution (with variance γ) on g_{ij},

$$f(g_{ij}) = \frac{1}{\sqrt{2\pi\gamma}} e^{\frac{-g_{ij}^2}{2\gamma}}, \tag{4.24}$$

and further assuming that the distributions $f(g_{ij})$ are independent, so

$$f(G) = \prod_{i,j} f(g_{ij}). \tag{4.25}$$

The function to be maximized is now the average log likelihood of the inputs u, the cause estimates $\hat{v}_V(u)$, and the weights G, or

$$\mathcal{F}(\hat{v}(u), G) = \langle \log(f_U(u|\hat{v}(u), G) f_V(\hat{v}(u)) f(G)) \rangle. \tag{4.26}$$

This strategy for introducing additional structure on G is closely related to the method of "hyperparameter estimation" introduced in the original work describing EM (Dempster et al., 1977).

Plugging in the expressions for the various distributions and neglecting the constant terms we find

$$\mathcal{F}(\hat{v}(u), G) = \left\langle -\frac{1}{2\lambda} \|u - G\hat{v}(u)\|^2 + \sum_{j=1}^{m} S(\hat{v}_j) - \frac{1}{2\gamma} \sum_{j=1}^{m} \sum_{i=1}^{n} g_{ij}^2 \right\rangle. \tag{4.27}$$

Taking the derivative with respect to $\hat{v}$ and setting equal to zero gives us the same result as before (Equation 4.14), and so we compute $\hat{v}$ via the same differential equation as before (Equation 4.15). The Gaussian distribution on g_{ij} introduces a decay term to the update rule for G, however, and we have

$$G \rightarrow \left(1 - \frac{\delta}{\gamma}\right) G + \frac{\delta}{\lambda}(u - G\hat{v}(u))\hat{v}(u)^T \tag{4.28}$$

for learning rate δ. This decay term keeps the size of the weights under control, eliminating the need for the explicit constraint employed by Olshausen and Field, and reflects the plausible biological condition that rarely active synapses become weaker over time (alternatively, “forgetting” is built into the model through the decay term).

4.3.3 Batch Learning

The quadratic penalty on the weights g_{ij} of G also allows us to explicitly solve for the optimal G for a set of inputs $\{u\}$ and cause estimates $\{\hat{v}(u)\}$. To do so, we first carry out the E-step computation for *all* inputs u for fixed G, obtaining an estimate $\hat{v}(u)$ for each u. We then take the derivative of $\mathcal{F}$ with respect to G, set it equal to zero, and solve for G to obtain the batch update rule

$$G \longrightarrow \langle u\hat{v}(u)^T \rangle \left(\frac{\lambda}{\gamma} I + \langle \hat{v}(u)\hat{v}(u)^T \rangle \right)^{-1} . \tag{4.29}$$

Because $\frac{\lambda}{\gamma}I$ is positive definite and $\langle vv^T \rangle$ is positive semidefinite, their sum is positive definite and thus nonsingular, so this learning rule is always well defined and yields the globally optimal G for the current $\hat{v}(u)$. This rule is a significant extension of the method, as the large M step results in much faster convergence of the EM algorithm than the incremental rule presented in Olshausen and Field (1997). In the applications discussed in the next chapter, a typical experiment was sped up by easily an order of magnitude or more by implementing the batch process.

The batch algorithm is then as follows. We denote the k^{th} iteration of G and $\hat{v}$ by $G^{(k)}$ and $\hat{v}^{(k)}$, respectively.

Initially: $\hat{v}^{(0)}(u) = \mathbf{0}$ for all $u \in \{u\}$, $G^{(0)} = rand(n, m)$

E step: For each $u \in \{u\}$, compute $\hat{v}^{(k+1)}$ by gradient ascent on $\mathcal{F}$ starting at $\hat{v}^{(k)}$ with

$G = G^{(k)}$. That is, simulate the differential equation

$$\dot{v} = \nabla_v \mathcal{F} = G^T(u - Gv) + \lambda S'(v) \tag{4.30}$$

until $\|\dot{v}\|$ falls below some convergence threshold $\dot{v}_T$.

M step: Set $G^{(k+1)}$ according to the update rule

$$G^{(k+1)} = \langle u\hat{v}^T \rangle \left(\frac{\lambda}{\gamma} I + \langle \hat{v}\hat{v}^T \rangle \right)^{-1} \tag{4.31}$$

with $v = v^{(k+1)}$.

Iteration: Alternate E and M steps until the average change in the weights g_{ij} falls below some threshold δg_T.

One advantage of the batch learning rule is that it renders the algorithm more amenable to analysis. In Appendix B I show that this algorithm converges to some set of local maximizers of $\mathcal{F}$; in practice the algorithm has always converged by the average change in weight criterion.

In summary, the algorithm I use for the remainder of this work implements sparse coding with a bimodal prior via EM. The model requires 6 parameters to be specified:

1. m, the number of output (v) neurons,

2. λ, the noise variance,

3. γ, the target weight variance,

4. σ^2, the variance of the Gaussians in the sparse prior,

5. a, the probability of a large response in the sparse prior, and

6. r_h, the high response rate (though this is simply a rescaling and is always set to 1.

4.4 Related Models

The sparse coding algorithm discussed here is a special case of a very general class of linear Gaussian generative models, in which the observed inputs are a noisy linear function of unobserved states ("causes" in our terminology). Roweis and Ghahramani (1999) provide an extensive discussion of such models, detailing how such disparate learning techniques as factor analysis, principal component analysis, mixtures of Gaussians, and Kalman filters can be described in this framework. I here highlight a few important cases that are particularly relevant to the learning problems discussed in this work.

Sparse coding is closely related to *factor analysis*, discussed by Dempster, Laird, and Rubin (1977) alongside the EM algorithm. As with sparse coding algorithm discussed here, factor analysis assumes the observed variables (inputs) depend in an affine way on a lower-dimensional set of unobserved "factors" (causes), and the EM algorithm is used to estimate the parameters of the mapping from factors to observed variables as well as the factor scores themselves. In Dempster and colleagues' discussion, however, the assumed prior distribution on factors ($\exp(S)$ in my notation) is a zero-mean, unit-variance Gaussian (though he does not include this assumption as part of the definition of factor analysis).

Taking the limit of factor analysis as the noise goes to zero (with the usual zero-mean Gaussian assumption on the factors, so $\exp(S)$ becomes a zero-mean Gaussian with increasingly small variance) yields *principal component analysis* (PCA), first introduced (though not given this name) by Pearson (Pearson, 1901) as an approach to the problem of optimally fitting lines and planes to systems of points. PCA finds the directions of maximum variance in a set of input points, thereby finding the most efficient set of basis vectors for representing the inputs (in terms of minimizing reconstruction error for any fixed number of basis vectors). From an information-theoretic point of view, PCA maximizes the information about the inputs that can be carried by a limited number of basis vectors (Dayan & Abbott, 2001). Though PCA can be solved exactly via singular value decomposition (SVD), an EM algorithm has also been found (Roweis, 1997). This algorithm, found by viewing PCA as the zero-noise limit of factor analysis and applying the same learning techniques, offers two distinct advantages over the SVD approach: its complexity grows only linearly in number of data points, input dimension, and number of components to be learned, and it can deal gracefully with incomplete data points by estimating maximum likelihood values for any missing information. PCA has been applied to natural images with the aim of describing the behavior of V1 cells, but only the first few principal components were found to bear significant resemblance to known V1 responses (Hancock et al., 1992).

If the number of causes and inputs is the same (so $m = n$) and no noise is included in the model, the problem of estimating the causes and their mapping to inputs is known as *independent components analysis* (ICA), first introduced by Herault and Jutten (1986) in the context of extracting source signals from sensors sensitive to an unknown linear combination of the sources. Bell and Sejnowski (1995) generalized this problem and cast it in an information-theoretic framework, with applications to blind source separation and blind deconvolution problems. In later work they apply it to natural scenes, finding that Gabor-like filters develop with more sparsely distributed outputs than other decorrelating filters such as principal components (Bell & Sejnowski, 1997). ICA has also been used to learn efficient codes for natural sounds, with the resulting code bearing a great resemblance to that observed in cochlear nerve cells (Lewicki, 2002).

Chapter 5

Application to Visual Information

The sparse coding model described in the last chapter was originally applied directly to natural images. It developed receptive fields strikingly similar to those of simple cells in the mammalian primary visual cortex (Olshausen & Field, 1996, 1997). In this work I am interested instead in building a model capable of reproducing the selective, invarient behavior observed further along the visual pathway and in the MTL as described in Chapter 3 (Quian Quiroga et al., 2005; Waydo et al., 2006). My central hypothesis is that the machinery of the ventral visual pathway is largely concerned with building an invarient feature-based description of visual inputs, transforming the input data but not necessarily increasing the sparseness of representation. The MTL, then, builds a sparse model for these invarient features. These two simple computational principles, sparseness and invariance, naturally lead to explicit representation as observed in MTL. In the first half of this thesis, I described the representation at various stages along the ventral visual pathway, culminating in the highly sparse, selective, and invariant representation observed in MTL. In the previous chapter I described one method for learning a sparse representation for sample data, and now in this chapter I will show that this method, when combined with a separate system for invariant feature extraction, is sufficient to reproduce these response patterns. To do so I apply the sparse coding model of the previous chapter to the outputs of different models for invariant feature extraction. Through training, units develop displaying sparse, invariant selectivity for particular object categories (such as faces or cars) or even for particular individuals, much like that observed in the MTL data. Portions of these results are currently in press for publication elsewhere (Waydo & Koch, 2007a, 2007b).

To achieve the high-level invariance observed in the human MTL data, it is first necessary to develop an invariant feature-based (rather than pixel-based) description of images such as may exist in inferotemporal cortex (IT) as input to MTL. Aside from mimicking the observed data from electrophysiology, this process projects images from the space of pixels (or patterns of retinal activity) in which different images of the same object may be wildly different to a space of features in which different images of the same object will lie close to one another (and hopefully images of different objects are far apart). That is, the features are robust to "unimportant" (from the standpoint of recognition) transformations such as lighting, pose, and scale. Cells in monkey IT have been found to be selective for "moderately complex" features—that is, features more complex than orientation, size, color, and texture, but in general not complex enough to represent natural objects (Tanaka, 1997) (with the exception of faces, for which specialized machinery appears to exist (Bruce et al., 1981; Perrett, Rolls & Caan, 1982, and see Chapter 3)). I investigate three methods of generating such a representation here and show results of applying the sparse coding network to this representation for different input sets. The primary method (upon which most of my results are based) is the feedforward neural network model of visual processing of Serre et al. (2005, 2007), which has the advantage of employing biologically plausible computations throughout the hierarchy; its practical drawback is that it is very computationally complex. This model is also highly non-invertible, in the sense that it is impossible to determine what portions of each image contribute most to any given response. I describe this model in more detail and present results in Section 5.2. The second model, discussed in Section 5.3, is based on the Scale-Invariant

Feature Transform (SIFT) algorithm of Lowe (1999). While the SIFT computations are less biologically plausible, the outputs at least are still analogous to IT responses and can be computed much faster. Furthermore, it is possible to determine what image features drive the responses. In Section 5.4 I present some preliminary results from applying this model to a feature extraction system tailored specifically to face recognition (Holub & Moreels, 2007). In Section 5.5 I discuss the statistics of the response distribution in more detail. Finally, in Section 5.6 I investigate the structure of the G matrix after training and examine the effects of quantizing and truncating G on recognition performance. Robustness of G to these disturbances, which model synaptic noise and pruning, is crucial to establishing the biological plausibility of my results.

5.1 Classification Accuracy Metrics

In the following sections I apply the sparse coding model of Chapter 4 to features extracted from various collections of images of objects drawn from different categories (or in come cases, images of different people). My goal is for the sparse coding network to develop units that respond selectively to the different categories present in the input set, without being given information about which images belong to which categories, or even the number or type of categories present. Given that I use a purely unsupervised training process, and that the model is free to identify fewer or more categories than are present in the training set, there are several possibilities for evaluating the classification accuracy of this system. I consider three metrics here, two of which are weakly supervised as they require us to decide what category each unit is selective for, and one of which is fully unsupervised:

Metric 1: Single-category classifier. I consider each unit individually as a classifier for its most preferred category. The accuracy figure I use is the receiver-operating characteristic (ROC) equal error rate (i.e., p(true positive) = 1-p(false positive)) testing against the other categories. Chance level in this case is 50%. The metric is the average accuracy of the best classifier for each category.

Metric 2: Weakly supervised classifier. I use all selective units together to classify each input image into one of the input categories. To do so, I first manually assign to each unit a category for which it is most selective, as before (so multiple units could be assigned the same category). I then classify each image according to which unit responded the most strongly. The accuracy is then the percentage of testing images correctly classified, and the chance level is one over the number of categories.

Metric 3: Unsupervised classifier. In the fully unsupervised setting I rely on the output units to both define the categories and assign images to them. Each image is assigned to a putative category based on which output unit responded the most strongly. I then form a confusion matrix in which element (i, j) is the percentage of images from input category j assigned to output category i and rearrange this matrix to maximize the average of the diagonal elements, thereby picking the output categories that best correspond to the input categories. This average is then the classification accuracy, and chance level is one over the number of output units.

Note that each of these metrics says something different about the behavior of the network, and none of them by themselves describe exactly the sparse, invariant selectivity that is our goal. Metric 1 quantifies how selective individual units are for particular categories, but disregards the separation between “on” and “off” responses. Metric 3 quantifies how precisely the categories discovered by the network correspond to those we defined, but a network that divides one or more categories into subcategories would score poorly here despite qualitatively good performance. Metric 2 alleviates this issue, but could disregard excessive subcategorization. Hence, sparse, invariant representation of the input categories is only captured by good scores according to all three metrics.

It is important to note that I label metrics 1 and 2 "weakly supervised" purely because evaluating them requires information about which images are in which categories. In all cases the model is trained in a completely unsupervised manner: no information is supplied about which images are in which categories, or even how many categories are present in the input set. The model simply receives a collection of inputs and learns a representation for them.

5.2 A Feedforward Model of Visual Processing

5.2.1 Overview of the Model

The first model I use to generate an invariant feature-based description of images is the feedforward model of Serre et al. (2005, 2007), which is an extension of the HMAX model of Riesenhuber and Poggio (1999). This model processes images via a series of alternating layers of S (simple) and C (complex) units in an extension of the Hubel and Wiesel (1962) simple-to-complex cell hierarchy. The S units provide Gaussian-like tuning around template features, while the C units provide scale and position invariance by pooling S units with the same feature selectivity across nearby positions and scales. The initial S layer, called S_1, consists of units which, like V1 simple cells, are tuned to oriented bars and edges at a variety of scales and orientations. In the next layer, C_1, each unit pools the responses of S_1 units with the same preferred orientation but with small variations in position and scale, increasing the receptive field size and the invariance to transformations and modeling complex cell behavior. Continuing up the hierarchy, each S_2 unit is tuned to the activity of nearby C_1 units with different feature selectivity, increasing the complexity of the unit's preferred feature, and each C_2 unit pools the responses of similar S_2 units over position and scale. In this way both feature complexity and receptive field size increase as one progresses up the hierarchy, until at the output layers of the model each unit responds to the presence of a particular complex feature located anywhere in the input image, in a manner analogous to IT cells. The most recent version of this model (which I use here) incorporates two parallel processing paths with somewhat different parameters for the selectivity and pooling range, a "standard" route with three simple-to-complex stages terminating with layer C_3 ($S_1 \to C_1 \to S_2 \to C_2 \to S_3 \to C_3$) and a "bypass" route with two stages terminating with layer C_{2b} ($S_1 \to C_1 \to S_{2b} \to C_{2b}$). This model normally terminates with a layer S_4 (receiving inputs from C_{2b} and C_3) that is task-specific in that its feature templates are learned from images from the set to be classified. I instead rely only on the task-independent C_{2b} and C_3 outputs. That is, in the version of the model I use, the feature templates for the S layers are learned from images unrelated to the specific tasks at hand. Despite being designed primarily to model biological vision, this model has been shown to perform on par with the state of the art in image classification tasks in a supervised setting (Serre, Wolf, Bileschi, Riesenhuber & Poggio, 2007) and even to match human performance in a rapid categorization task (Serre, Oliva & Poggio, 2007). The software is available from `http://cbcl.mit.edu/software-datasets/`.

5.2.2 Inputs to the Model

All images used in this investigation were taken from the Caltech-256 database of images from 256 categories (Griffin, Holub & Perona, 2006). Images were resized (using MATLAB's `imresize` with nearest-neighbor interpolation) so that the smaller dimension was 128 pixels while preserving the aspect ratio. The outputs of the C_{2b}

and C_3 layers of the visual processing model were computed using a feature set derived from training on 500 natural images. The feature set I used for the S layer templates was the "universal" set included with the software distribution. There were 1000 units in each of these layers, for a total of $n = 2000$ outputs. In some cases an input image was large enough to have multiple C units for the same feature in the top layer, in which case I performed an additional max operation over these units to preserve input dimension. After computing the outputs for all input images, I renormalized them to have zero mean and unit variance (they were initially values between 0 and 1). While it may be possible to find parameters of the sparse coding network that work well on the unnormalized data, this rescaling makes it possible to apply the network to different input sets (such as this set and the SIFT features described below) without adjusting the various network parameters for optimum performance.

5.2.3 Results: Categorization

I performed several object categorization experiments with this model. In all cases the number of output units was $m = 10$ and the network parameters were $\lambda = 10$, $t = 0.05$, and $\sigma^2 = 0.04$. The weight penalty was $\gamma = 100$. The matrix G was initialized with uniformly distributed random weights between -0.5 and 0.5. In each experiment I used the batch update rule and terminated the optimization when the average change in the weights g_{ij} was less than 2% for 5 consecutive iterations. I used 40 random images from each category for training and reserved 40 different images for testing. After training, I ran the recognition model on the novel testing images; these are the responses depicted below.

I performed the following three experiments:

(A) Three object categories. I trained and tested the model on images of motorbikes, airplanes, and faces. This is directly comparable to experiment (C) of Sivic et al. (2005).

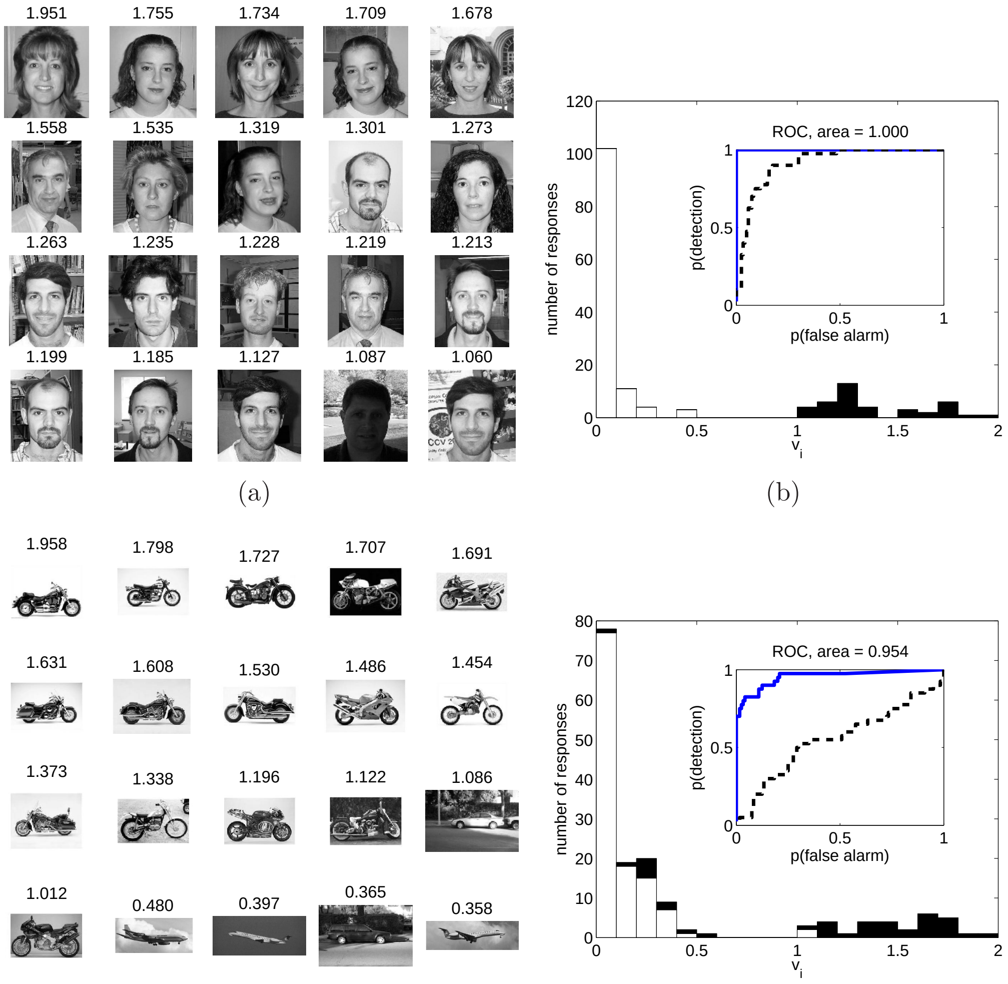
(B) Four object categories. I added a fourth category (cars) to the training set from experiment (A). This is similar to experiment (D) of Sivic et al. (2005), except that I used side- rather than rear-views of cars.

(C) Four object categories. As the images from experiment (B) are relatively easy to classify (a supervised classifier operating on the same inputs can perform this task at near 100% accuracy), I performed the same experiment with four more difficult categories: blimps, elephants, ketches (a type of sailboat), and leopards.

I ran each experiment 10 times with different random initial conditions for G. All model parameters were identical between the three experiments—no adjustment was required to account for different number or type of input categories between experiments.

I here focus on describing the response profiles of the output units from a typical run of experiment (B); results from the other trials and experiments were qualitatively similar. Figure 5.1 depicts the responses of two of the selective units (from the same session) that emerged in training. For each unit this figure shows 20 of the 40 images that evoked the strongest responses (every other response is omitted for clarity) as well as a histogram of all responses. The ROC curve for each unit treated as a classifier for its preferred category is inset in the histogram, along with the ROC curve for the best principal component for that category for comparison. We see from these figures that category tuning has spontaneously emerged from the learning process.

The quantitative results of each experiment, as measured by the three metrics described above, averaged over 10 trials (one “trial” refers to a complete training/testing run with random initial conditions), are summarized in Table 5.1. As a baseline for



(a)

(b)

(c)

(d)

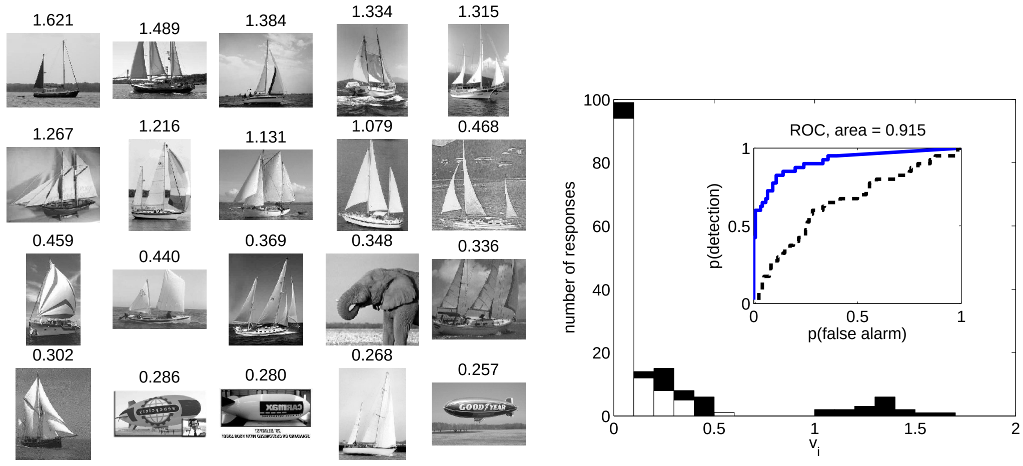
Figure 5.1: Responses of two selective units (out of 10) after the unsupervised category learning. (a, c): images that evoked the top responses, with the activation level above each image. Every 2^{nd} image omitted for clarity. (b, d): response histograms. x-axis is the activation level; y-axis is the number of test images (160 total) evoking a response at that level. Responses to preferred category in black; responses to all other images in white. Insets: ROC curves. Solid line is ROC curve for selected unit, dashed line is ROC curve for best principal component. ROC equal-error accuracies were 100% and 88%.

Ex	Metric 1				Metric 2					Metric 3		
	SN	PCA	SVM	ch	SN	PCA	k-means	SVM	ch	SN	PCA	ch
A	91.7	69.2	98.1	50.0	90.6	55.0	95.8	96.7	33.3	64.0	37.5	10.0
B	89.8	71.9	97.4	50.0	82.6	46.9	91.9	96.9	25.0	66.1	40.6	10.0
C	77.0	69.2	88.1	50.0	63.8	47.5	57.5	81.9	25.0	41.4	36.3	10.0

Table 5.1: Classification accuracy computed using different metrics averaged over 10 trials with random initial conditions. In all cases unseen images were used for testing. For each metric I report the classification accuracy (as a percentage) for the sparse network (SN) and for PCA applied to the same inputs, as well as chance level. For metrics 1 and 2 I also provide the accuracy of a supervised SVM classifier applied to the same inputs, and for metric 2 I further include the accuracy of k-means with k equal to the true number of categories.

comparison, I also evaluated the performance of PCA applied to the same inputs as the sparse coding network against these three metrics. As there were 10 units in the output layer of the sparse coding network, I used the top 10 principal components for this comparison. I also found the best performance I could achieve using a supervised SVM classifier applied to the same inputs, which provides a reasonable upper bound on achievable performance and an objective measure of task difficulty. For metric 1 I report the average accuracy of a binary SVM classifier for each category versus the others, while for metric 2 I report the accuracy of a multi-way SVM. Finally, I applied a k-means algorithm with k equal to the true number of categories. As in this case the number of categories is a given, this performance metric is most comparable to the semi-supervised performance of the sparse coding network.

The seemingly poor results from experiment (C) still occur in the context of units that show very clean selectivity for each category. However, in each case the units responded strongly only to a *subset* of the category in question. Figure 5.2 gives an example of such a unit which responded selectively to some but not all of the ketch images. Note also that this task is considerably more difficult than the others, as quantified by the large drop in supervised SVM accuracy (also listed in Table 5.1).



(a) (b)

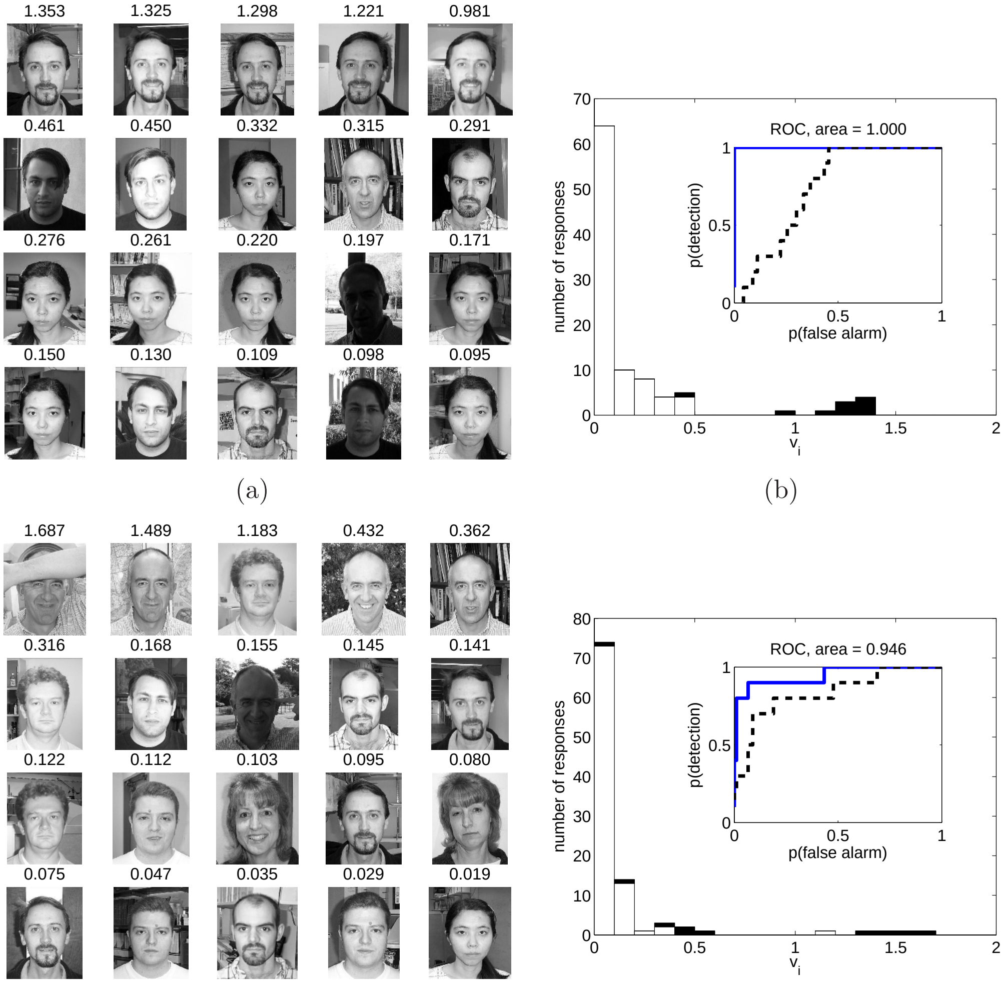
Figure 5.2: Responses of a ketch unit from experiment (C). (a): images that evoked the top responses, with the activation level above each image. Every 2^{nd} image omitted for clarity. (b): response histogram. x-axis is the activation level; y-axis is the number of test images (160 total) evoking a response at that level. Responses to ketches in black; responses to all other images in white. Inset: ROC curve. Solid line is ROC curve for this unit, dashed line is ROC curve for best principal component. ROC equal error accuracy with respect to all ketches was 85%.

5.2.4 Results: Face Discrimination

To evaluate performance in a finer discrimination (as opposed to categorization) task, I tested the algorithm on a dataset consisting of gray-scale frontal facial images of different individuals obtained from the Caltech-256 dataset (Griffin et al., 2006). Though the backgrounds vary slightly from image to image, these images are fairly well structured and could be viewed as the output of an attentional selection and segmentation process. Training was performed using 10 different images of each individual, with 10 different images of the same individuals reserved for testing. I performed experiments with 4 to 10 different individuals in the input set. All network parameters were identical to the categorization task described above, except that the number of output units was $m = 15$.

Figure 5.3 depicts the responses of two selective units (the best and a more typical unit) from a single training session with 10 different individuals in the input set. The mean ROC accuracy (that is, the average ROC accuracy of the best unit for each category) for this run was 91%, and the ROC accuracies for the two units shown were 100% (Figure 5.3 (a, b)) and 90% (Figure 5.3 (c, d)). The semi-supervised 10-way classification accuracy of the sparse network was 56%. PCA yielded a mean ROC accuracy of 78% and a semi-supervised classification accuracy of 37%. Additionally, in contrast to the responses of the sparse units depicted in Figure 5.3, the responses of the principal components were unimodal and so did not clearly indicate the presence of a category in the same way as the sparse units (which is reflected in the poor semi-supervised accuracy). Figure 5.4 depicts the response of the best principal component for *any* category from the same dataset as in Figure 5.3 and gives an example of this issue: while the ROC equal-error accuracy of this principal component for its "preferred" category is 90%, there is no clean separation between in-category and out-category responses.

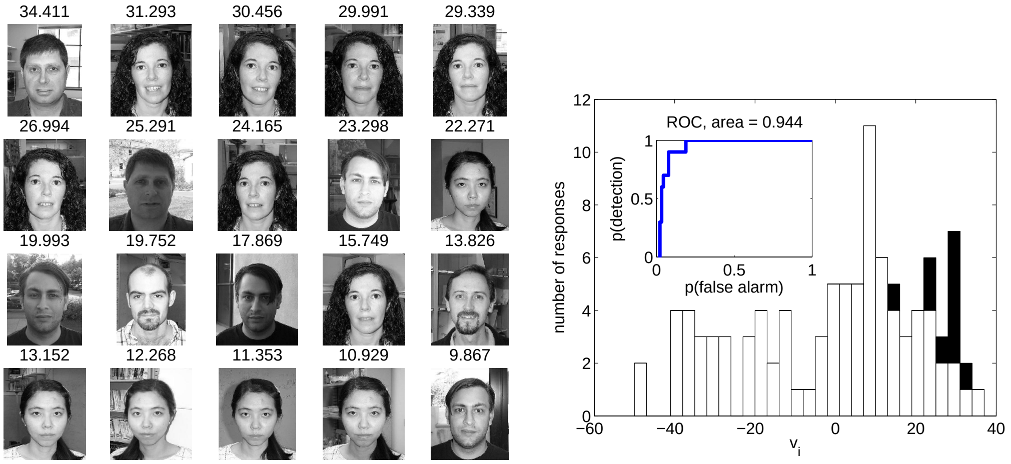
I repeated this experiment 50 times for each number of different individuals, each time starting with different random initial conditions (initial synaptic weights), using a different random subset of the 17 individuals for which the dataset contains at least 20 pictures, and using different random subsets for training and testing. Figure 5.5 summarizes the results for metrics 1 and 2 (ROC and semi-supervised) and compares them to those obtained from the top 15 principal components and the performance achieved by a supervised SVM; the complete numerical results are listed in Table 5.2 with the addition of the performance of a k-means algorithm with k equal to the true number of categories. Performance according to the ROC metric did not vary significantly with the number of people presented, indicating that in all cases units emerged that responded selectively to each individual. The mean ROC accuracy across all 350 trials was 91.3%, compared to 96.6% for a binary SVM and 80.4% for PCA. Performance according to the semi-supervised metric did decline as the number of people in the input set increased, dropping from a mean of 85.5% to 64.2% as the



(a) (b)

(c) (d)

Figure 5.3: Responses of two selective units (out of 15) after the unsupervised category learning. (a, c): images that evoked the top responses, with the activation level above each image. Every 2^{nd} image omitted for clarity. (b, d): response histograms. x-axis is the activation level; y-axis is the number of test images (100 total) evoking a response at that level. Responses to preferred person in black; responses to all other images in white. Insets: ROC curves. Solid line is ROC curve for selected unit (exactly along the vertical and horizontal axes in (b)), dashed line is ROC curve for best principal component.

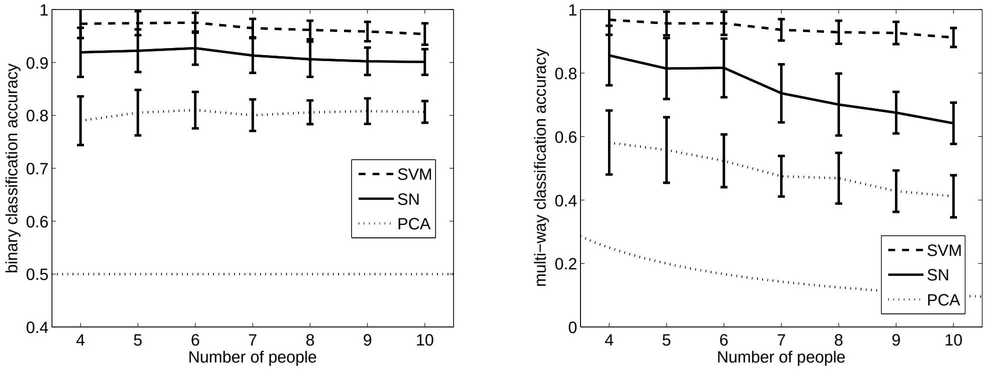


(a)

(b)

Figure 5.4: Responses of best principal component for a particular category for same inputs as in Figure 5.3. (a): images that evoked the top responses, with the component loading above each image. Every 2^{nd} image omitted for clarity. (b): response histogram. x-axis is the component loading; y-axis is the number of test images (100 total) evoking a response at that level. Responses to preferred person in black; responses to all other images in white. Inset: ROC curve.

number of people increased from 4 to 10. This is in all cases significantly better than the PCA performance, which decreased from 58.1% to 41.1%, and the performance of k-means, which decreased from 70.7% to 63.7%. This decline is not unexpected, because as more categories are presented it becomes more likely that, in addition to the "correct" unit responding to a given image, some other unit will spuriously respond strongly (which is also reflected in the decreasing chance performance). In the purely unsupervised case, performance increases slightly as the number of people's faces to be recognized rises from 4 to 6 before dropping off gradually with more people, likely because with 15 output units significant subcategorization may be taking place when there are few people in the input set. PCA sees a similar increase in accuracy in this regime. Figure 5.6 depicts an example of such subcategorization occurring when only 4 different faces were present in the input set. Shown are two units that, after training, responded to different images of the same individual. The top unit had an ROC equal-error accuracy of 80%, while the bottom unit had an accuracy of 90%.



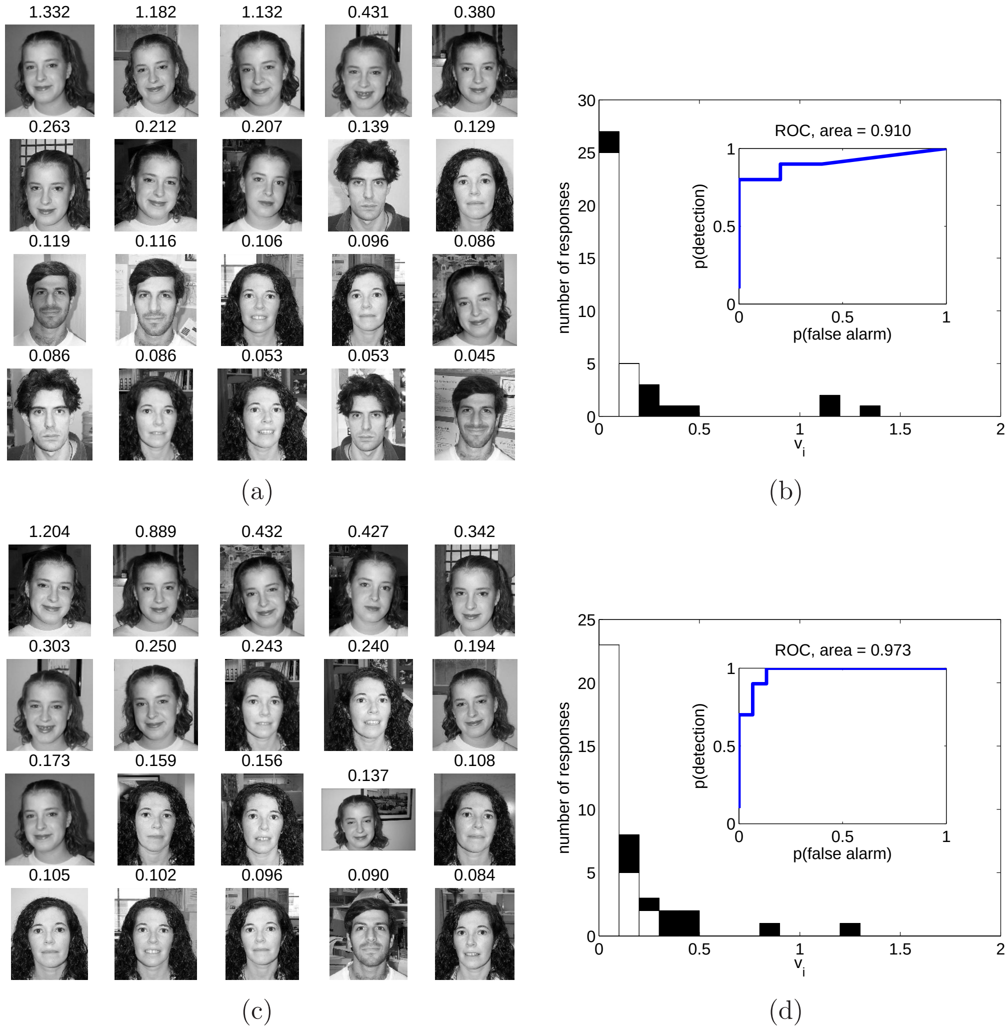
(a)

(b)

Figure 5.5: Face discrimination accuracy (mean ± s.d.) as a function of number of people in the input set. Solid line: sparse coding network, dashed line: SVM (supervised) classifier, dotted line: PCA. Dotted line without error bars depicts chance performance. (a): ROC equal-error accuracy for binary classification. (b): semi-supervised multi-way classification accuracy

The overall ROC accuracy of this run (which only the more accurate bottom unit contributed to) was 92.5%. However, since both units responded strongly to several images of the same person, either one was liable to have the strongest response to any particular image of that person, hurting the unsupervised accuracy (metric 3), which was 67% overall. In fact, each of these two units provided the strongest response to 40% of the testing images of this individual, so they evenly divide the category in two. The only clear difference between the two subsets of images is that the bottom unit's preferred images appear brighter; though the normalization steps in the HMAX model should provide invariance to brightness it may be that the top unit is responding to details in the darker images that are washed out in the bright images.

It is interesting to note that, even with 10 faces in the input set, performance on this task was essentially as good as in the categorization tasks described above. While the distinction between different faces is clearly more subtle than the distinction between categories, there is also less within-category variation in the face images than in the images from other categories, so different images of the same individual are likely to be tightly clustered in feature space. From this we see that the within-



(a) (b)

(c) (d)

Figure 5.6: Responses of two selective units (out of 15) after the unsupervised category learning. (a,c): images that evoked the top responses, with the activation level above each image. (b,d): response histograms. x-axis is the activation level; y-axis is the number of test images (40 total) evoking a response at that level. Responses to preferred person in black; responses to all other images in white. Insets: ROC curves.

#	Metric 1				Metric 2					Metric 3		
people	SN	PCA	SVM	ch	SN	PCA	k-means	SVM	ch	SN	PCA	ch
4	91.9	79.0	97.3	50.0	85.6	58.1	70.7	96.8	25.0	67.0	49.7	6.7
5	92.2	80.5	97.4	50.0	81.4	55.8	71.2	95.6	20.0	70.0	53.2	6.7
6	92.7	81.0	97.5	50.0	81.6	52.3	70.6	95.7	16.7	72.2	50.7	6.7
7	91.3	80.0	96.5	50.0	73.6	47.5	65.5	93.6	14.3	68.7	47.5	6.7
8	90.6	80.6	96.2	50.0	70.0	46.9	64.0	92.9	12.5	65.7	47.2	6.7
9	90.2	80.8	95.9	50.0	67.5	42.8	63.4	92.6	11.1	63.5	44.0	6.7
10	90.1	80.7	95.4	50.0	64.1	41.2	63.7	91.2	10.0	63.3	43.9	6.7

Table 5.2: Face discrimination accuracy computed using different metrics averaged over 10 trials with random initial conditions using HMAX features. In all cases, unseen images were used for testing. For each metric, I report the classification accuracy (as a percentage) for the sparse network (SN) and for PCA applied to the same inputs, as well as chance level. For metrics 1 and 2, I also provide the accuracy of a supervised SVM classifier applied to the same inputs, and for metric 2 I further include the accuracy of k-means, where k equals the true number of categories.

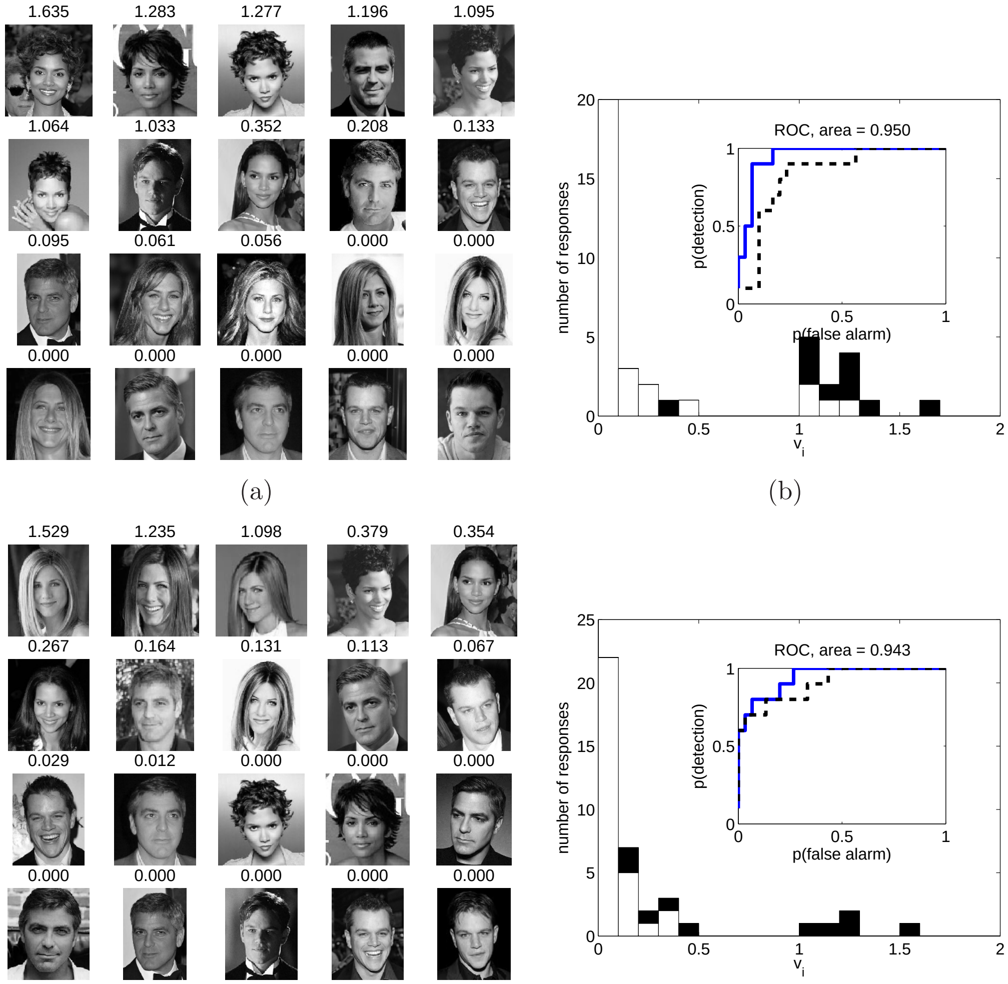
class homogeneity drives classification accuracy as much as the inter-class separation. These experiments also highlight the importance of the statistics of the input set to the representation learned. In experiments (A) and (B) in the previous section, faces were present often in the inputs, but no particular individual was present often. In this case we obtained a representation for “face,” but no individuation within that class. Roughly speaking, there was a cluster of inputs in feature space distinct from the inputs from the other categories, but no smaller clusters corresponding to specific individuals within it. In these experiments, particular individuals were present often (and each individual was present equally often), so it became apparent that there were well defined clusters within the “face” cluster, giving the network enough information to identify multiple individuals and represent them separately.

To evaluate robustness of this method to more widely varied facial images (and to improve the analogy with the results from human MTL), I also applied the model to images of four celebrities collected from the web (Jennifer Aniston, Halle Berry, George Clooney, and Matt Damon). I selected images that contained reasonably frontal views of faces and cropped them to include only the face, so the overall composition was similar to the images used above. However, these images contained substantially more variation in pose, facial expression, hairstyle, and background than the images used above, and were much more difficult to classify even using supervised methods. I again performed 50 trials (full training and testing runs) using 10 images for training and 10 different images for testing, randomizing over initial weights and which subsets of images were used for training and testing. The resulting average ROC accuracy was 77.4%, compared to an average supervised SVM accuracy of 84.3%. Relative to the benchmark of supervised classification (which expresses how well the underlying vision model separates the categories into distinct groups), then, performance was essentially the same as before.

Figure 5.7 depicts the responses of two selective units from a typical run on the celebrity images. The Halle Berry (upper) unit had an ROC equal-error accuracy of 90%, while the Jennifer Aniston (lower) unit had an accuracy of 80%. It is particularly interesting to note that the images “missed” by the Halle Berry unit all depicted her with long hair, while the images that evoked strong responses depicted her with short hair, so the unit in fact selects for a subcategory of Halle Berry images.

5.2.5 Results: Morphed Faces

A further investigation of the human MTL responses currently in progress involves presenting the patient with “morphed” images of familiar people, that is, images that are created by blending images of two different people (A. Kraskov, *personal communication*). This experiment serves two purposes: to investigate how the response of a neuron changes as an image is continuously transformed from a person the neuron responds strongly to into some other person (and back), and to see how the neuron’s activity correlates with the subject’s perception of the image’s identity. The first question can also be investigated within this computational framework, with the



(a) (b)

(c) (d)

Figure 5.7: Responses of two selective units (out of 15) after the unsupervised category learning. (a, c): images that evoked the top responses, with the activation level above each image. Every 2^{nd} image omitted for clarity. (b, d): response histograms. x-axis is the activation level; y-axis is the number of test images (40 total) evoking a response at that level. Responses to preferred person—Halle Berry in (b), Jennifer Aniston in (d)—in black; responses to all other images in white. Insets: ROC curves. Solid line is ROC curve for selected unit, dashed line is ROC curve for best principal component.

additional advantage that we can look at the responses of neurons that ordinarily represent both ends of the morph (while in the human studies the investigators generally only have access to a neuron representing one of the endpoints due to the small number of simultaneously recorded selective neurons in any one session).

To explore the question of how the model responds to morphed images, I picked two individuals with some similarities in appearance from the same training session. For this example I used the same trained network for which I presented results in Section 5.2.4 above, for which there were 10 different individuals in the input set. Both individuals used for morphing were very well represented by the trained network, with a unit providing 100% ROC accuracy and well separated in-category and out-category responses for each. I generated 9 morphed images between each of 5 different pairs of images using the commercially available photo morphing software "Morpheus" (available at `http://www.morpheussoftware.net/`). To ensure a smooth morph between the two images I manually matched keypoints such as eyes, ears, and mouths in the two images, so the resulting morph was a combination of distortion and grey-level interpolation between the starting and ending images. I then computed the response of the trained network to the morphed images. There is no effect of hysteresis in these results, as the state of the network (initial condition of v) was reset for each image presentation.

Figure 5.8 summarizes the results for all 5 morphings and gives an example morphing. Response strength to each morphed image is shown for the two neurons representing the two indivduals. Each curve is the response of one neuron to one set of morphed images; the curves are individually normalized by the strength of the neuron's response to the unmorphed image of its preferred person. As expected, the response curves are sigmoidal, with a sharp transition between on and off responses as some threshold is crossed. This sigmoidal transition is a feature of the sparse coding network and is due to a combination of the bimodal prior and the winner-take-all

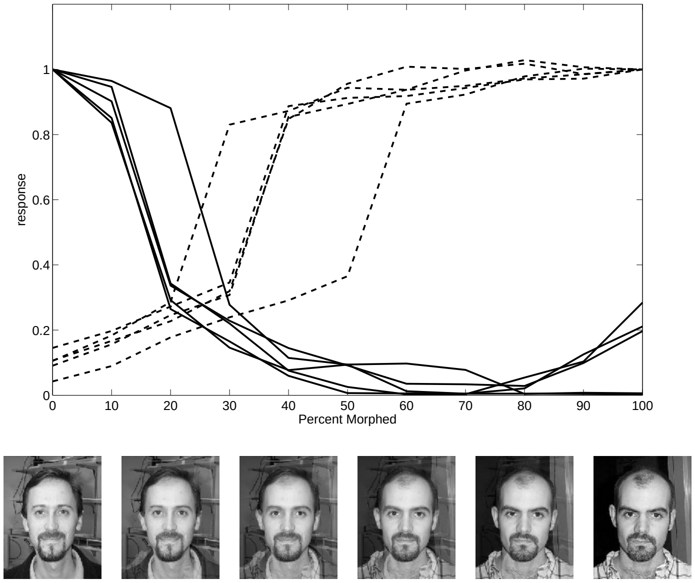


Figure 5.8: Responses of trained network to 5 different morphings between the same two individuals (top) and an example morphing (bottom). Solid lines are the responses of the neuron that prefers the person on the left; dashed lines are the responses of the neuron that prefers the person on the right. All responses are normalized by the response to the unmorphed preferred image.

like network topology; it is much different from the gradual transition that would be expected from linear filters. In a distributed population code in which individual neurons responded to, for example, different types of facial features, individual neurons may still switch on or off in the same sigmoidal fashion as their preferred features became more or less clear, though just as plausibly their activity could vary smoothly if they functioned as linear feature templates.

Much like in the human data from both electrophysiology and psychophysics, different morphings result in different transition thresholds, reflecting the difference between the qualitative similarity of a morphed image to one individual or the other and the distance along the continuum of morphings (Kraskov, *personal communication*). The average point at which the response was 50% of the response to the unmorphed preferred image was 18.6% morphed ($\sigma = 4.6$) for the neuron that preferred the individual on the left, and 35.3% morphed ($\sigma = 10.5$) for the neuron that preferred the individual on the right. Hence in most cases there is a range of morphings that produce only weak responses in both neurons—the network essentially decides that the image resembles neither individual. Because (as noted above) in the human experiments neurons representing both endpoints are only rarely available it is very difficult at this time to compare this particular aspect of the responses to real data, but this suggests one interesting question that could be asked if it is ever possible to perform the morphing experiment between images of two people that are represented by two different recorded neurons: is one or the other of two such neurons always active, or, like in the model, is there some range of morphed images that elicit no strong response? Further, how does this activity correlate with the subject's identification of the image as being one person or the other (or neither)?

5.3 Scale-Invariant Feature Transform

5.3.1 Overview of the Model

Another algorithm that can be used to produce invariant feature detectors is the Scale-Invariant Feature Transform (SIFT), first introduced by Lowe (1999) and later refined to the form applied here (Lowe, 2004). The first step in the SIFT algorithm is to identify a set of *keypoints* corresponding to features with a high interest level and high likelihood of invariance to scale and affine transformations. In this work I use Harris-Affine interest point detection (Mikolajczyk & Schmid, 2004), which combines the Harris edge and corner detector (Harris & Stephens, 1988) with an automatic scale selection algorithm (Lindeberg, 1998) to obtain a scale invariant detector. On the order of 500 (though at times as few as 100) features are identified for each image. Each keypoint is assigned an image location, scale, and orientation, which together naturally define a local 2D coordinate system that provides invariance to these parameters. The next step is to compute a *descriptor* for the image region around each of these keypoints that is both highly distinctive and invariant to other unimportant transformations such as changes in illumination and viewpoint. The descriptor is based on local image gradients. First, the gradient magnitude and orientation is computed at an array of sample points surrounding the keypoint, with the extent of the array defined by the scale of the keypoint. The gradient samples are weighted by a Gaussian centered at the keypoint (to avoid discontinuities in the descriptor with small changes in keypoint location) and accumulated into histograms over subregions. These histograms are smoothed to avoid discontinuities in the descriptor with small changes in keypoint orientation. Finally, the descriptor is formed by concatenating all of the subregion histograms into a single vector and normalizing to unit length. The resulting descriptor is invariant to changes in location, scale, and orientation (because these are explicitly accounted for in the keypoint identification), brightness (because it is based on gradients), and contrast (because it is normalized). In the SIFT implementation I use here, there are 16 subregions (in a 4×4 array about the keypoint) and 8 bins in each subregion histogram, so the resulting descriptor is 128 dimensional. The software I use to implement both keypoint detection and descriptor computation can be obtained from `http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html`.

The intent in the original SIFT work was to compute descriptors that could be matched between images, allowing test images to be classified with respect to template images. Here we need to convert the collection of SIFT descriptors extracted from each image into an n-dimensional vector describing in some sense the presence of various features in the input image. In other words, we need to compute a feature vector analogous to the outputs of the neural network model discussed above from the SIFT descriptors. To do so, I first extract n descriptors at random from some image set (I will describe exactly what image set I used in the results sections below) to be feature templates. I denote the i^{th} such template by τ_i. Denoting the descriptors in an image by d_j, the input u was computed by

$$u_i = \max_j(\tau_i \cdot d_j). \tag{5.1}$$

That is, for each template, each descriptor is assigned a score based on the dot product of the descriptor with the template, then the maximum of these scores is taken to be the response to that feature. In this way u_i will be large if there is some feature in the image very similar to τ_i.

While the computations used to generate these inputs have no simple biological implementation, the inputs themselves are not altogether unreasonable from a biological standpoint. They are essentially complex feature detectors with receptive fields of the entire field of view, and Lowe points out that "SIFT features share a number of properties in common with the responses of neurons in inferior temporal (IT) cortex in primate vision" (Lowe, 1999).

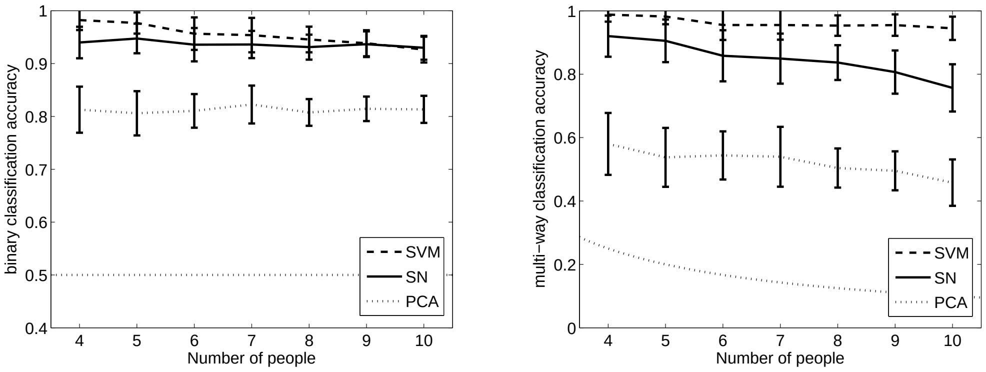
As with the first model, I normalize these inputs to have zero mean and unit variance prior to feeding them into the sparse coding network.

5.3.2 Results: Face Discrimination

As SIFT is geared more toward object recognition rather than the broader categorization task, I repeated the face discrimination experiments from Section 5.2.4 using SIFT inputs. All network parameters were the same as before, and I used $n = 2000$ SIFT features to match the number of inputs I used in the HMAX experiments. The feature templates were extracted at random from images from the Caltech-256 dataset. That is, I ran the SIFT algorithm on all of the images in the Caltech-256 dataset, then picked 2000 random descriptors from random images to be my feature templates. Repeating this analysis using only features from the face dataset did not significantly affect the results. Figure 5.9 summarizes the results for metrics 1 and

2 (ROC and semi-supervised) and compares them to those obtained from the top 15 principal components and the performance achieved by a supervised SVM; the complete numerical results are listed in Table 5.3 with the addition of the performance of a k-means algorithm with k equal to the true number of categories. Performance according to the ROC metric did not vary significantly with the number of people presented, indicating that in all cases units emerged that responded selectively to each individual. The mean ROC accuracy across all 350 trials was 93.7%, compared to 95.5% for a binary SVM and 81.3% for PCA. It is interesting to note that the performance of the unsupervised sparse coding network was just as good as the supervised SVM, especially as the number of people in the input set increased, though it could be that the choice of SVM parameters was not optimal. Performance according to the semi-supervised metric again declined as the number of people in the input set increased, dropping from a mean of 92.0% to 75.7% as the number of people increased from 4 to 10. This is in all cases significantly better than the PCA performance, which decreased from 58.0% to 45.8%, though it is about the same as the performance of k-means, which decreased from 94.1% to 82.6%. The fact that k-means performed much more on par with the sparse coding network in this case implies that the underlying clusters were more nearly spherical than in the case of HMAX features. In the purely unsupervised case, performance increased significantly as the number of people rose from 4 to 9 before dropping off slightly with 10 people, likely because with 15 output units significant subcategorization may be taking place when there are few people in the input set. This difference is more dramatic than with the HMAX algorithm used in Section 5.2.4, likely because the SIFT descriptors distinguish finer differences between images and so subcategorization is a bigger problem when excess output neurons are available. PCA sees a smaller increase in accuracy in this regime.

An advantage of the SIFT approach is that it is possible to determine which image



(a)

(b)

Figure 5.9: Face discrimination accuracy (mean ± s.d.) as a function of number of people in the input set, using SIFT for invariant feature extraction. Solid line: sparse coding network, dashed line: SVM (supervised) classifier, dotted line: PCA. Dotted line without error bars depicts chance performance. (a): ROC equal-error accuracy for binary classification. (b): semi-supervised multi-way classification accuracy

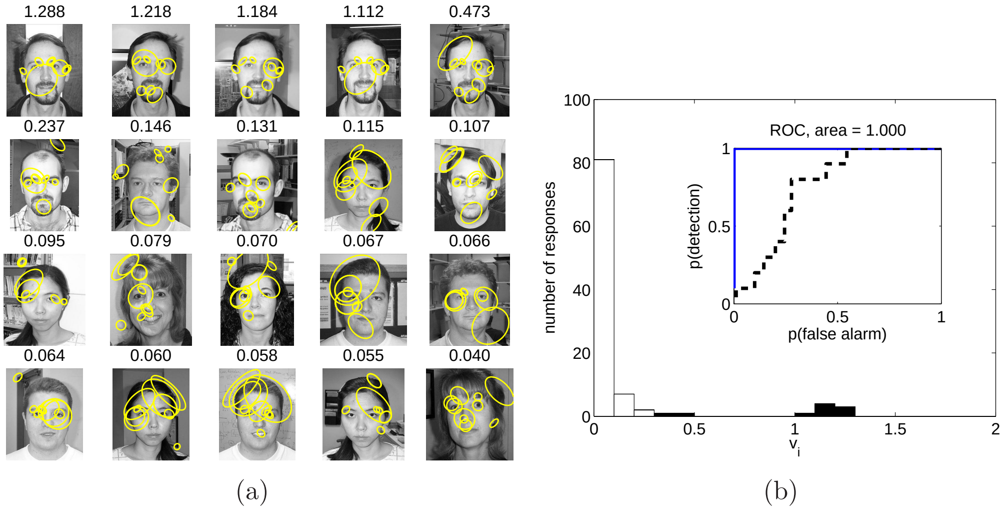
#	Metric 1				Metric 2					Metric 3		
people	SN	PCA	SVM	ch	SN	PCA	*k*-means	SVM	ch	SN	PCA	ch
4	94.0	81.3	98.2	50.0	92.0	58.0	94.1	98.8	25.0	60.2	49.3	6.7
5	94.8	80.6	97.7	50.0	90.5	53.8	87.7	98.2	20.0	67.4	48.2	6.7
6	93.6	81.1	95.7	50.0	85.8	54.4	86.7	95.5	16.7	69.7	50.4	6.7
7	93.6	82.3	95.4	50.0	84.9	53.9	85.9	95.5	14.3	73.8	51.1	6.7
8	93.1	80.8	94.6	50.0	83.7	50.4	82.5	95.3	12.5	76.6	48.1	6.7
9	93.7	81.4	93.9	50.0	80.7	49.5	83.8	95.5	11.1	77.5	49.2	6.7
10	93.0	81.3	92.7	50.0	75.7	45.8	82.6	94.5	10.0	74.1	46.0	6.7

Table 5.3: Face discrimination accuracy computed using different metrics averaged over 10 trials with random initial conditions using SIFT features. In all cases unseen images were used for testing. For each metric I report the classification accuracy (as a percentage) for the sparse network (SN) and for PCA applied to the same inputs, as well as chance level. For metrics 1 and 2 I also provide the accuracy of a supervised SVM classifier applied to the same inputs, and for metric 2 I further include the accuracy of *k*-means, where *k* equals the true number of categories.

features (SIFT descriptors) contributed the most to the observed response. To do so, I first determined which of the 2000 features drove the output unit most strongly (i.e., for neuron j which inputs i had the largest $u_i g_{ij}$). I then found which SIFT descriptors "won" the max operation in Equation 5.1 for these features. These descriptors then provided the largest input to the unit in question. Figure 5.10 depicts the responses of an example unit from a network trained on the same image set as in Figure 5.3 above, with the 10 most important descriptors highlighted. This unit formed a very clean, sparse representation for its preferred individual. From the figure we see that the most important features were those that we may expect to be most discriminatory between individuals: eyes, eyebrows, and other distinctive features such as hairline or goatee. It is also interesting to note that even for the distractor images very few of the driving features are from the image background, indicating that the model has successfully interpreted the variable background as noise and learned the more consistent features of the various faces.

5.3.3 Comparison with HMAX

Table 5.4 summarizes the results of the face discrimination task for both HMAX and SIFT inputs. Performance is fairly close between the two underlying models, indicating that they do a similar job of projecting images into a feature space good for discriminating between different faces while generalizing between different images of the same face. A few more subtle details expose differences between the two models, however. SIFT was designed to take advantage of features that should be very similar between different images of the same object (as opposed to images of different exemplars of the same class), so the features are generally very specific to that object. In fact, just three descriptors or so are often good enough to match an object between two images (Lowe, 1999). Performance as measured by the ROC or semi-supervised metrics, then, is somewhat better when the model is applied to



(a)

(b)

Figure 5.10: Responses of one selective unit (out of 15) after the unsupervised category learning on the same image set as in Figure 5.3 using SIFT features. (a): images that evoked the top responses with the 10 most important SIFT descriptors outlined and the activation level above each image. Every 2^{nd} image omitted for clarity. (b): response histograms. x-axis is the activation level; y-axis is the number of test images (100 total) evoking a response at that level. Responses to preferred person in black; responses to all other images in white. Insets: ROC curves. Solid line is ROC curve for selected unit, dashed line is ROC curve for best principal component.

SIFT features rather than HMAX features. This is because a unit is less likely to be excited by a non-preferred person because such an image is likely to be well separated from images of the preferred person in feature space. Furthermore, with fewer people in the input set than available coding units, the SIFT features make available finer distinctions between images than the HMAX features, so categories are more likely to be split into subcategories. Using the semi-supervised metric 2, this results in better performance, as multiple units representing different subsets of the same category are taken into account. Using the unsupervised metric 3, however, this results in worse performance for small numbers of input categories.

I also tested the SIFT approach on the multi-class categorization task (airplanes-cars-motorbikes-faces) described in Section 5.2.3 above, with very different results. In that case, the images from a single category are much more widely separated, so the generalization capabilities of the model need to be correspondingly better. This is

#	Metric 1			Metric 2			Metric 3		
people	HMAX	SIFT	ch	HMAX	SIFT	ch	HMAX	SIFT	ch
4	91.9	94.0	50.0	85.6	92.0	25.0	67.0	60.2	6.7
5	92.2	94.8	50.0	81.4	90.5	20.0	70.0	67.4	6.7
6	92.7	93.6	50.0	81.6	85.8	16.7	72.2	69.7	6.7
7	91.3	93.6	50.0	73.6	84.9	14.3	68.7	73.8	6.7
8	90.6	93.1	50.0	70.0	83.7	12.5	65.7	76.6	6.7
9	90.2	93.7	50.0	67.5	80.7	11.1	63.5	77.5	6.7
10	90.1	93.0	50.0	64.1	75.7	10.0	63.3	74.1	6.7

Table 5.4: Comparison of performance of sparse coding network applied to HMAX and SIFT features on face discrimination task.

where SIFT performs much worse than HMAX; in fact, performance is barely better than chance in this setting (and so the details are omitted). It turns out that, for example, images of two different motorcycles may be as widely separated in SIFT feature space as an image of a motorcycle and one of an airplane, so they are not likely to be clustered together.

These distinctions between HMAX and SIFT suggest a possible hierarchy for object and category representation in the brain. At one stage, neurons may operate on HMAX-like inputs to become selective to broad categories such as motorcycles and faces. Such neurons would explicitly represent their preferred categories, but within each category the identity of a particular exemplar would be carried only across the population. As discussed in Chapter 3, the "face cells" of the macaque inferior temporal cortex are an example of such neurons: individual cells respond much more strongly to faces than non-faces, but facial identity is carried across the population (Young & Yamane, 1992; Rolls & Tovee, 1995). These neurons may then be making explicit image features best suited for making fine distinctions between objects within their preferred category, but perhaps not suited for making broader category judgement; these features would be more akin to the SIFT features used here. A large population of these neurons with the same category selectivity, then, may form the input to a second sparse coding stage that makes identity within the category explicit. The sparse, invariant human MTL neurons are the clear example here (Quian Quiroga et al., 2005).

A second possibility also comes to mind, however. With only roughly $10,000$ afferents on average, cortical neurons receive input from only a tiny fraction of cells in the preceding region. Simply by chance, then, some neurons may receive input from neurons representing a subset of features well-suited to broad categorization (HMAX-like features) while others receive input from neurons that respond to features better adapted to making fine distinctions within some category (SIFT-like features). The emergence of category- and exemplar-selective cells would then happen in parallel rather than as a two-stage process. With the available data it is unclear which of these two architectures is more likely (or if there is a third possibility), though the clear existence of face-selective cells in macaque IT and individual-selective cells in human MTL makes the hierarchical architecture attractive.

5.4 A Specialized Facial Recognition Model

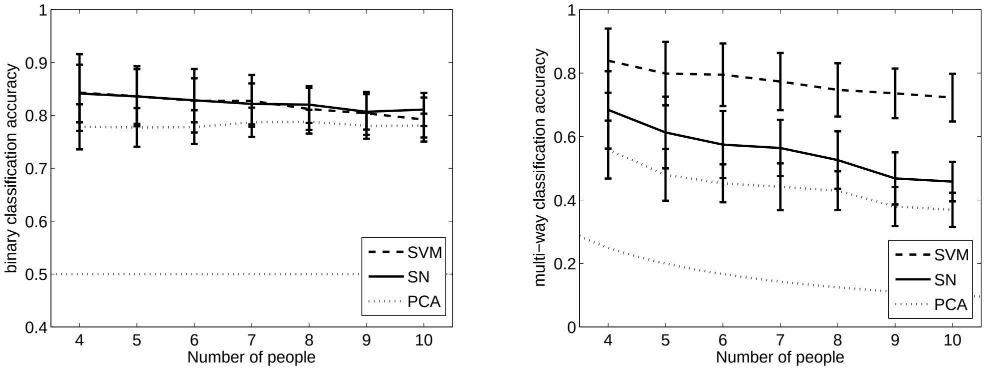
5.4.1 Overview of the Model

To generalize face discrimination results of Sections 5.2 and 5.3 to more natural images (that is, with more variation in lighting, pose, etc.) I applied the model to a machine vision model specifically tailored to face recognition, in which faces of the same individual are likely to produce similar feature vectors in a manner somewhat robust to common transformations (Holub & Moreels, 2007). Their model first detects and segments faces within an image using the Viola and Jones face detector (Viola & Jones, 2001). The segmented region is then passed to an Everingham facial feature detector (Everingham, Sivic & Zisserman, 2006), which identifies the position of 19 facial features such as eyebrows, eyes, nose, and mouth (and parts thereof). These features can be characterized in a number of ways, such as raw pixel intensity or intensity gradients. For this investigation I obtained features characterized by raw pixel intensity in a 9×9 patch at each feature, for $9 \times 9 \times 19 = 1539$ features per image.

5.4.2 Results: Celebrity Faces

I tested the algorithm applied to responses obtained from the face recognition model applied to facial images of 99 different celebrities collected from the web (note that this is a *different* celebrity dataset than that used in Section 5.2.4 above). These responses were provided by the Caltech Vision Lab; at the time of this writing I had access only to the model outputs, not the original images. These images contain significantly more variation in pose, lighting, etc. than the facial images used above. As before, I performed experiments with 4 to 10 different individuals in the input set with exactly the same network parameters. As fewer images of each individual were available, I used just 5 images for training and 5 for testing; the data set included 92 individuals for which I had responses to at least 10 images.

Figure 5.11 summarizes the results for metrics 1 and 2 (ROC and semi-supervised) and compares them to those obtained from the top 15 principal components and the performance achieved by a supervised SVM; the complete numerical results are listed in Table 5.5. Again, performance according to the ROC metric did not vary significantly with the number of people presented, indicating that in all cases units emerged that responded selectively to each individual. The mean ROC accuracy across all 350 trials was 82.4%, compared to 82.0% for a binary SVM and 78.1% for PCA. As in the SIFT case above, the sparse network matched the performance of supervised SVM in the binary identification task quantified by the ROC accuracy. Performance according to the semi-supervised metric again declined as the number of people in the input set increased, dropping from a mean of 68.4% to 45.8% as the number of people increased from 4 to 10. This is in all cases significantly better



(a)

(b)

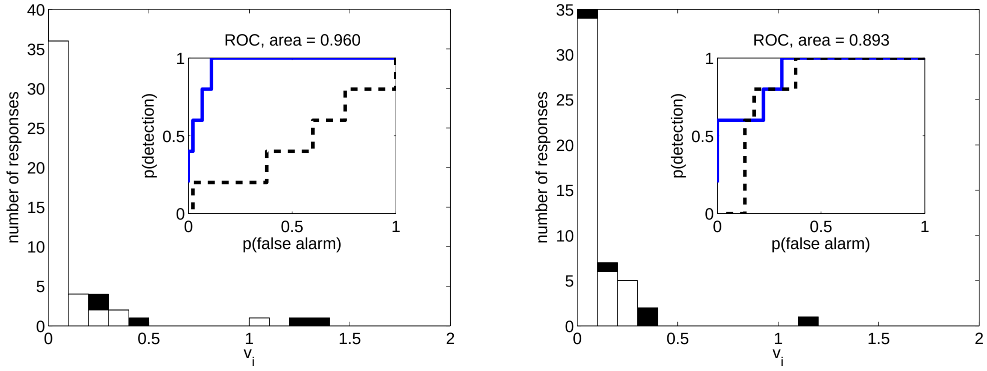
Figure 5.11: Face discrimination accuracy (mean ± s.d.) as a function of number of people in the input set using celebrity images and the face representation of Holub and Moreels (2007). Solid line: sparse coding network, dashed line: SVM (supervised) classifier, dotted line: PCA. Dotted line without error bars depicts chance performance. (a): ROC equal-error accuracy for binary classification. (b): semi-supervised multi-way classification accuracy

than the PCA performance, which decreased from 55.9% to 36.9%. In the purely unsupervised case, performance did not change significantly as the number of people rose from 4 to 10.

Figure 5.12 depicts the responses of two units (the best and a typical unit) from the same network after training on 10 individuals. It was frequently the case that even the most selective units would have very few (if any) large responses, as can be seen in this figure. This may help to explain why the accuracy of the sparse coding approach was no better than PCA for this dataset: if the network is never excited to large responses, the sparse prior is essentially a zero-mean Gaussian, and so the result approaches PCA as the variance becomes small. The reason behind the network being rarely excited to large responses is unclear; it may be that the very small size of the training set (just 5 training images of each individual) was insufficient for the network to extract the underlying sparse structure.

#			Metric 1				Metric 2			Metric 3	
people	SN	PCA	SVM	ch	SN	PCA	SVM	ch	SN	PCA	ch
4	84.1	77.8	84.3	50.0	68.4	55.9	83.9	25.0	50.8	44.3	6.7
5	83.6	77.7	83.6	50.0	61.3	47.9	79.8	20.0	48.5	41.0	6.7
6	82.9	77.8	82.8	50.0	57.5	45.3	79.5	16.7	52.3	41.7	6.7
7	82.2	78.7	82.7	50.0	56.4	44.2	77.3	14.3	51.9	43.4	6.7
8	82.0	78.8	81.2	50.0	52.6	43.0	74.7	12.5	51.0	43.3	6.7
9	80.7	78.0	80.4	50.0	46.8	38.0	73.6	11.1	48.9	39.6	6.7
10	81.1	78.2	79.2	50.0	45.8	36.9	72.3	10.0	47.5	38.8	6.7

Table 5.5: Face discrimination accuracy computed using different metrics averaged over 10 trials with random initial conditions using features from Holub and Moreels (2007). In all cases unseen images were used for testing. For each metric I report the classification accuracy (as a percentage) for the sparse network (SN) and for PCA applied to the same inputs, as well as chance level. For metrics 1 and 2 I also provide the accuracy of a supervised SVM classifier applied to the same inputs.



(a)

(b)

Figure 5.12: Response histograms for two units (the best and a typical unit) from the same training run on celebrity faces using the face representation of Holub and Moreels (2007). x-axis is the activation level; y-axis is the number of test images (100 total) evoking a response at that level. Responses to preferred person in black; responses to all other images in white. Insets: ROC curves. Solid line is ROC curve for selected unit, dashed line is ROC curve for best principal component. ROC equal-error accuracy of the left unit was 89%, of the right unit was 78%.

5.5 Statistics of the Response Distribution

In this section I discuss the statistics of the responses obtained from the sparse coding network after training. As an example I use the responses from the face discrimination network discussed in Section 5.2.4 (which was trained on 10 images of each of 10 different individuals); these results are typical of those obtained from other runs. For comparison I will look at two cases that strip key features from the sparse coding network. First, I cut the feedback connections but leave the dynamics of the individual neurons intact, to the network dynamics become

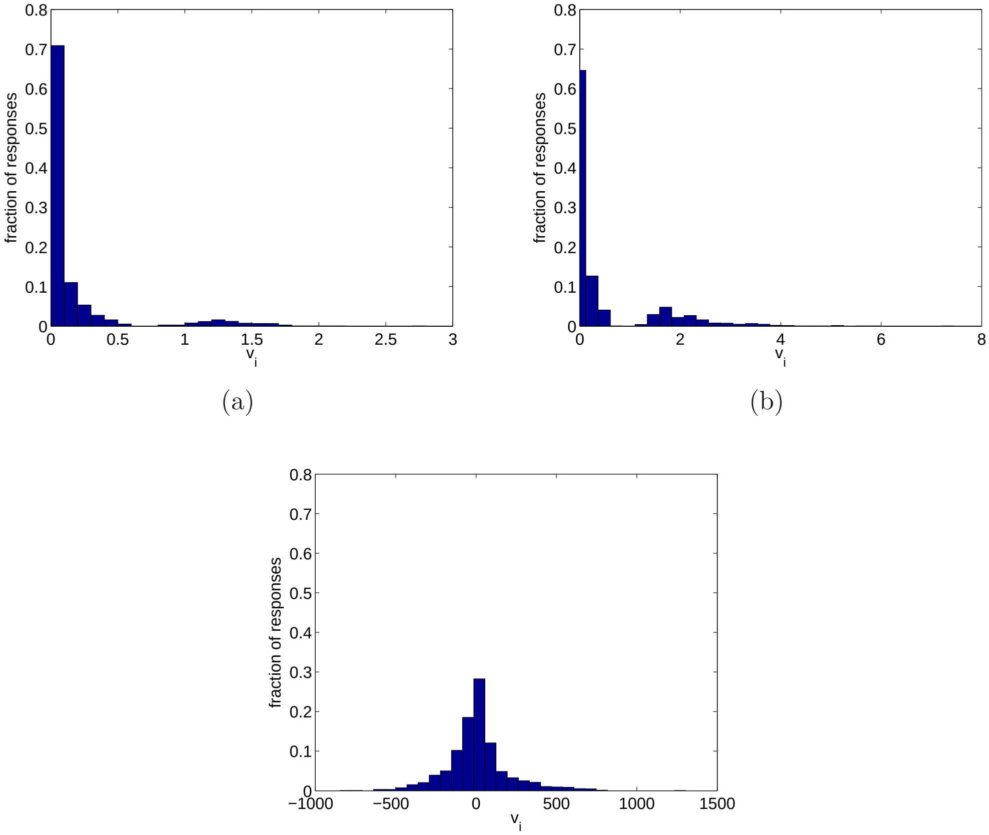
$$\dot{v} = G^T u + \lambda S'(v). \tag{5.2}$$

This will illuminate the role feedback plays in recognition performance and sparsening of responses, and provide a prediction of how recognition would suffer in the event that feedback connections were cut in the real biological system. Second, I simply treat the trained G matrix as a feed forward linear filter, that is, I set $v = G^T u$. This shows how similar each input u is to each learned basis function in the absence of the feedback inhibition that produces winner-take-all like behavior in the network. This linear feedforward network will allow us to see how much of the sparseness of the responses is due to the form of the learned basis functions and how much is due to the sparsening nature of the dynamics.

Both the dynamic and linear feedforward networks still perform well according to our classification metrics, with an average ROC accuracy of 89% in both cases (compared to 91% for the feedback network). However, a more detailed look at the responses reveals that true recognition performance would likely suffer somewhat more that the optimal ROC result suggests. The purely linear feedforward model lacks the bimodal response distribution that cleanly separates "on" responses from "off" responses and makes readout particularly easy. The response distribution of the dynamic feedforward network is still bimodal, but while the largest responses of an individual neuron tend to be to its preferred person, many significant responses are to other people due to the lack of inhibitory feedback from other neurons in the network. Hence our model predicts that, if feedback connections in the visual pathway were somehow cut, recognition performance would suffer but not be eliminated entirely—instead we would expect increased confusion between similar people or objects. Feedback is crucial for learning, however—we would expect a person with such an injury to be unable to learn to recognize new people or categories.

Figure 5.13(a) is a histogram of the strength of all responses to all images in the testing data set (100 images times 15 neurons for 1500 total responses). The response distribution is bimodal, as specified by the sparse prior, with most responses near zero. The "large" responses are centered around roughly 1.25, somewhat larger than the second peak location of 1 in the prior as the inputs bias all responses to be larger than the unstimulated equilibrium points of 0 and 1. The kurtosis excess of this distribution is 8.7, reflecting its sparse and bimodal nature. The responses of the dynamic feedforward network, depicted in Figure 5.13(b), are still bimodal, and are in general larger due to the lack of inhibitory feedback. These responses are still sparse, with a kurtosis excess of 6.6. The responses of the feedforward network are unimodal and widely varied, but due to the nature of the sparse basis functions are still sparse (but less so), with a kurtosis excess of 3.5. Figure 5.13(c) is the same histogram for the feedforward network; the distribution is clearly unimodal.

One often-suggested role for sparseness is the reduction of redundancy by decorrelating neuronal responses (Vinje & Gallant, 2000). Figure 5.14(a) is a histogram of the correlation coefficient between all neuron pairs (15 choose 2, or 105 pairs). Most correlation coefficients are negative, reflecting the inhibitory effect neurons have on one another. Overall correlations are weak, with a mean absolute value of the correlation coefficient of 0.18. Figure 5.14(b) is the same histogram for the network with

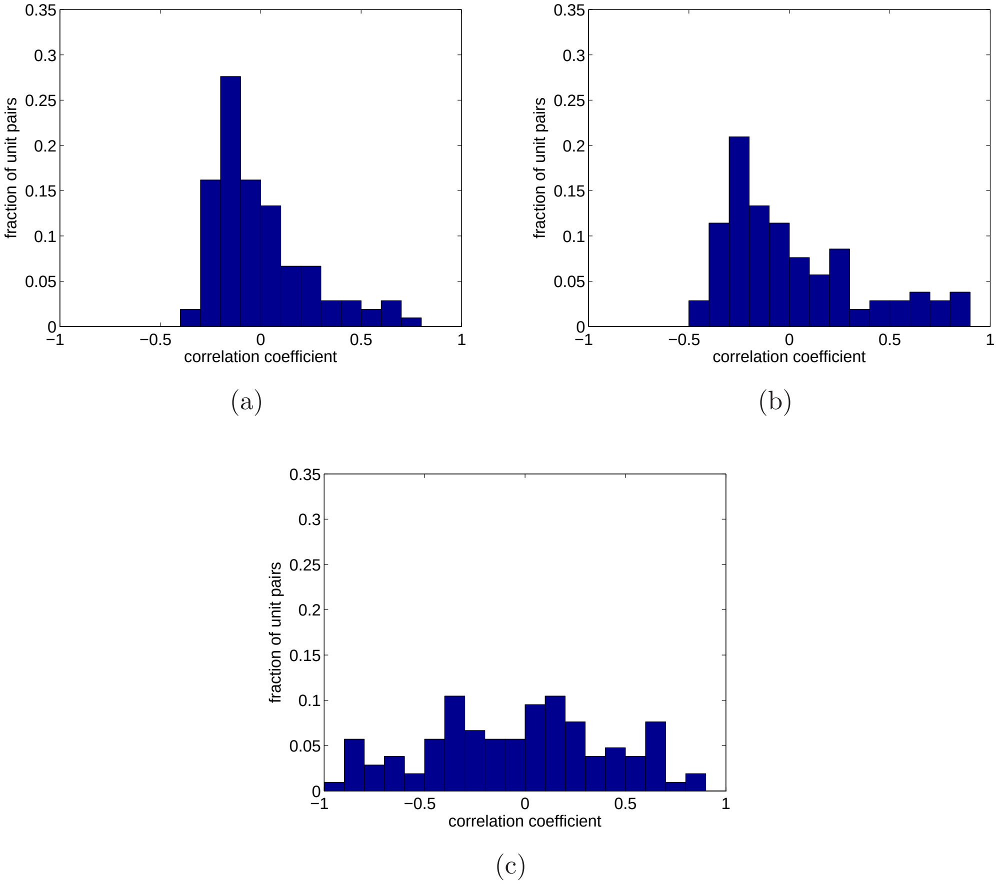


(a)

(b)

(c)

Figure 5.13: Histogram of the strength of all responses to all images in the testing data set (1500 responses total). (a): feedback network depicted in Figure 5.3. (b): the same network with the feedback connections cut. (c): linear feedforward network with the same G matrix.



(a)

(b)

(c)

Figure 5.14: Histogram of the correlation coefficient between all neuron pairs (105 neuron pairs total). (a): feedback network depicted in Figure 5.3. (b): the same network with the feedback connections cut. (c): linear feedforward network with the same G matrix.

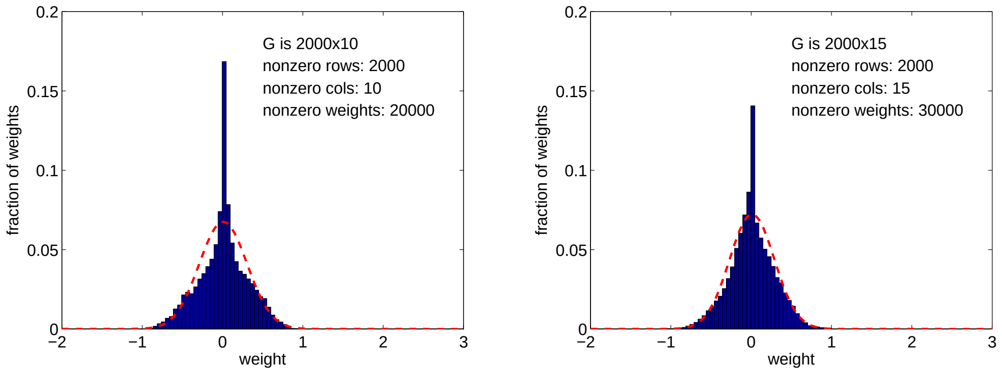
the feedback connections cut; correlations in this setting are somewhat higher, with an mean absolute value of 0.28. Finally, Figure 5.14(c) is the same histogram for the linear feedforward network. Neuronal responses are more strongly correlated in this case, with a mean absolute value of the correlation coefficient of 0.37. From this we see that both the dynamics induced by the sparse prior and the recurrent feedback play a role in decorrelating neural responses. Note that we are not considering temporal correlations here (as our network considers each image separately rather than an image sequence), but correlation in the amplitude of neural responses.

From these results we see that the structure of the recurrent sparse coding network serves to both enhance the sparseness of the responses (through the sparse prior distribution encoded in each neuron's dynamics) and to reduce the correlation between the responses of different neurons. This decorrelation reduces the redundancy of information carried in the firing rates of different neurons.

5.6 Structure and Robustness of the G Matrix

In this section I examine the structure of the synaptic weight matrix G after training, particularly with regard to the robustness of the network performance to perturbations in synaptic weights. Given that real neural networks do not enjoy full connectivity, and individual synapses most likely cannot have precisely controlled strength, this robustness is crucial to establishing the biological relevance of the model.

Figure 5.15 depicts histograms of the weights g_{ij} after training for the category classification example given in Section 5.2.3 and the face discrimination example given in Section 5.2.4 (both with HMAX features as inputs). Both sets of weights are essentially zero mean. The standard deviations are very similar, at 0.30 for the categorization network and 0.28 for the face discrimination network. The only significant difference between the two distributions is that the face discrimination network has more weights near zero and (though it cannot be seen in the figure) far from it, which is reflected in a higher kurtosis value of 4.5 compared to 3.2 for the categorization network. The dashed line overlaid on each histogram depicts a Gaussian distribution of the same mean and variance. It is clear from the figure that many more weights are near zero than would be expected from a Gaussian distribution. Though it is difficult to see from the figure, it is also the case that more weights are large than would be expected from a Gaussian: 5.4% and 5.3% of weights are more than 2 standard deviations from the mean for the category and face examples, respectively, compared to 4.6% for a Gaussian distribution. Thus the



(a) (b)

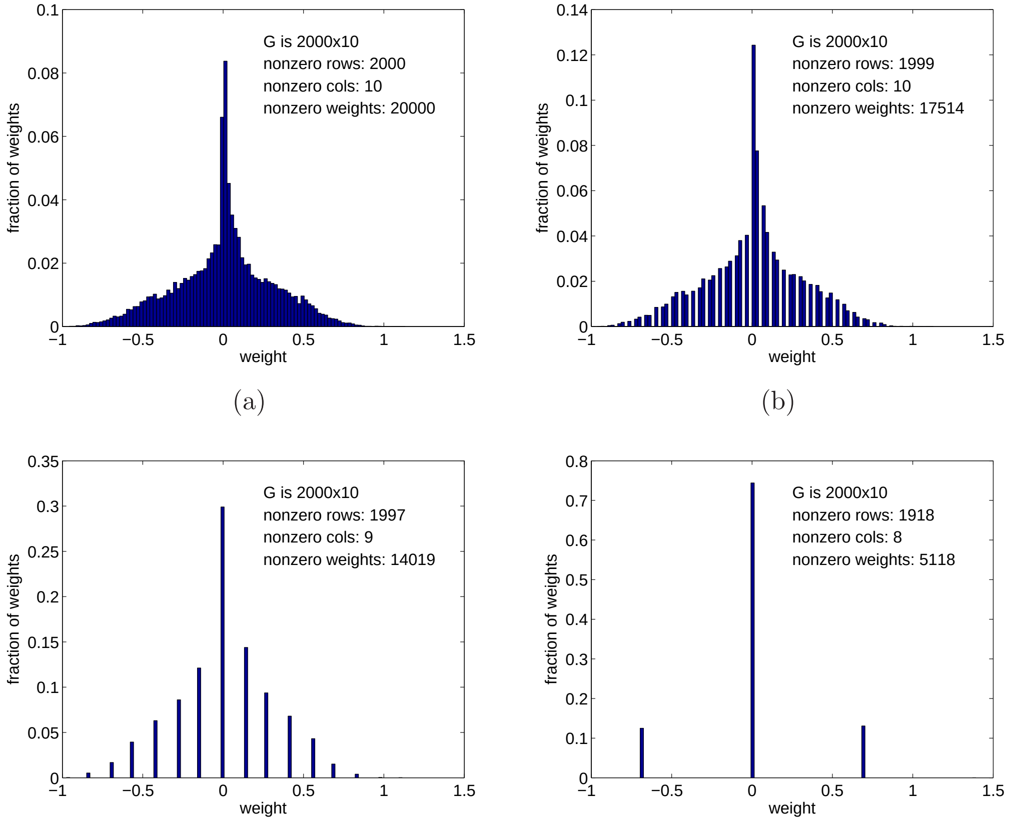
Figure 5.15: Histogram of synaptic weights g_{ij} after training. Bin locations and sizes are the same for both figures. Dashed line depicts a Gaussian distribution of the same mean and variance. (a): Category classification network of Figure 5.1. (b): Face discrimination network of Figure 5.3

weight distribution is sparse in the sense that more weights are both very small and very large than would be expected from a Gaussian distribution.

An important issue pertaining to the biological plausibility of all of these results is how much fidelity is required in the G matrix for successful recognition. It is unlikely that individual synaptic weights are precisely controlled to nearly the degree of accuracy used in the computational experiments described here. In Section 5.6.1 I investigate the effect of modeling the biological imprecision by quantizing the individual weights. Furthermore, the sparseness model assumes full connectivity of the neural network, again a biologically implausible constraint. In Section 5.6.2 I examine what happens when connectivity is reduced by truncating weights smaller than some threshold to zero. In both cases the network performance proves to be very robust to perturbations in G, bolstering the biological realism of the results.

5.6.1 Quantization

To determine the effect of noise or limited fidelity in synaptic weights, I quantized the G matrix to a fixed number of weights between the minimum and maximum trained



(a)

(b)

(c)

(d)

Figure 5.16: Histogram of synaptic weights g_{ij} after quantization for the category classification network. (a): original weights. (b): 6 bits quantization. (c): 4 bits quantization. (d): 2 bits quantization

values. This modification was carried out after training and the recognition model then run on previously unseen images. Figure 5.16 depicts histograms of the synaptic weights of the category classification network after quantization at different levels of fidelity.

Tables 5.6 and 5.7 give the resulting performance according to each metric, including the number and percentage of nonzero weights remaining, for the 4-category classification network used to generate Figure 5.1 and the trained face-discrimination network used to generate Figure 5.3 above. In both cases, performance according to all three metrics was preserved even at just 2 bits, or 4 quantization levels.

Looking at the actual responses depicted in Figures 5.17 and 5.18 tells a more

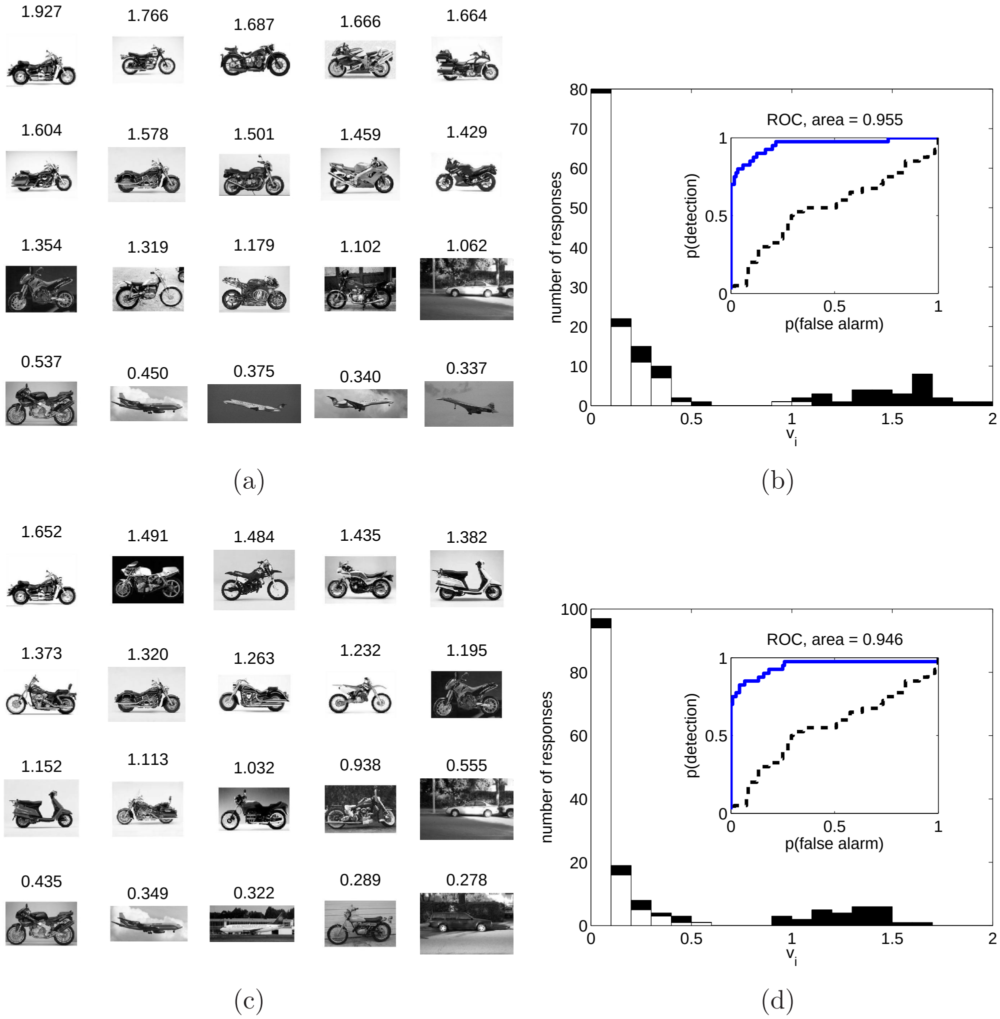
Table 5.6: Results from quantizing G, 4 category classification task

# bits	Metric 1	Metric 2	Metric 3
∞	91.9	82.5	75.6
8	91.9	81.3	75.6
6	91.9	81.3	75.6
4	92.1	80.6	75.6
2	91.7	85.0	76.3

Table 5.7: Results from quantizing G, 10 face discrimination task

# bits	Metric 1	Metric 2	Metric 3
∞	91.1	56.0	56.0
8	91.2	58.0	58.0
6	91.3	58.0	58.0
4	91.1	57.0	57.0
2	90.1	52.0	58.0

complete story, however. While it is true that the responses remain well-ordered with respect to the categories (thus producing a good ROC score) even at 2 bits quantization, the magnitude of the responses drops off as the quantization is decreased from 4 to 2 bits. In the case of the face recognition network, no responses are in the "high" regime at 2 bits quantization, though the ROC score remains at 99% accuracy. It appears that, by quantizing the G matrix, the overall input to each output (v) unit is decreased (because many weights are set to zero), resulting in the smaller responses. There may then be a strategy for rescaling G after quantization that could alleviate this issue, though simply rescaling G to preserve the average weight magnitude was not fruitful. It may also be possible to quantize G during training rather than just at the end, which would better reflect the biological reality of limited precision in synaptic weights and also allow the appropriate rescaling to happen as part of learning.



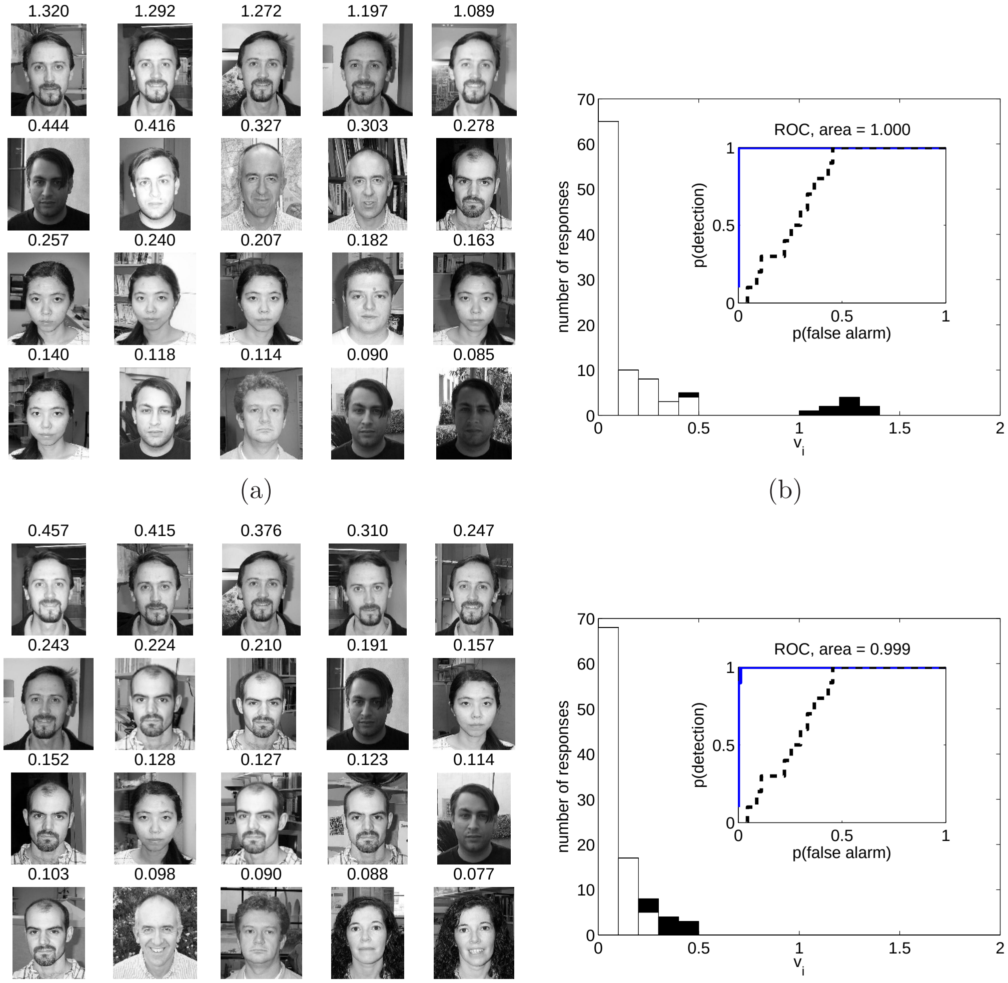
(a)

(b)

(c)

(d)

Figure 5.17: Responses of the motorbike unit of Figure 5.1(c, d) after quantizing G matrix. (a, b): 4 bits quantization. (c, d): 2 bits quantization. ROC equal-error accuracy was 87% at 4 bits and 88% at 2 bits.



(a) (b)

(c) (d)

Figure 5.18: Responses of the face unit of Figure 5.3(a, b) after quantizing G matrix. (a, b): 4 bits quantization. (c, d): 2 bits quantization. ROC equal-error accuracy was 100% at 4 bits and 99% at 2 bits.

5.6.2 Truncation

To determine the effect of reducing network connectivity, I truncated all weights of the G matrix smaller than a threshold number of standard deviations from zero to zero. Again, this modification was done after training, then the recognition model was run on previously unseen images. In contrast to the quantization results, in which it would be more realistic to quantize the network weights all along the way during training, truncation after training is somewhat more reasonable as it could reflect pruning of weak synapses.

Tables 5.8 and 5.9 give the resulting performance according to each metric, including the number and percentage of nonzero weights remaining, for the 4-category classification network used to generate Figure 5.1 and the trained face-discrimination network used to generate Figure 5.3 above. In both cases, performance according to all three metrics was preserved up to about 2 standard deviations truncation (at which point only just over 5% of weights remain nonzero), and drops off slowly after that.

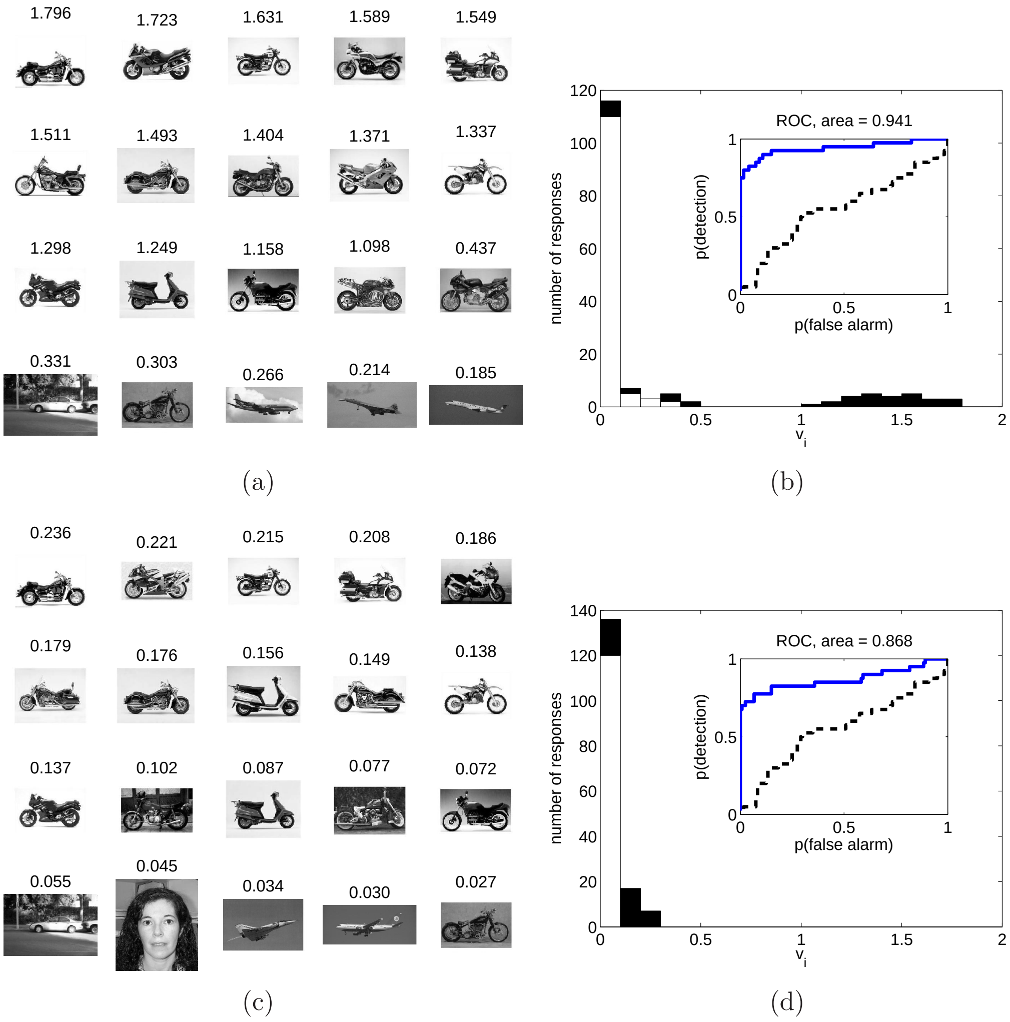
Table 5.8: Results from truncating G, 4 category classification task

thresh (σ)	# nonzero	% nonzero	Metric 1	Metric 2	Metric 3
0.0	20000	100.0	91.9	82.5	75.6
0.5	10643	53.2	91.9	80.6	76.3
1.0	6294	31.5	91.3	85.0	76.3
1.5	3179	15.9	92.1	81.9	77.5
2.0	1078	5.4	89.8	83.1	83.1
2.5	246	1.2	90.8	82.5	82.5
3.0	23	0.5	73.8	50.0	53.8

As with the quantization case, the actual responses depicted in Figures 5.19 and 5.20 show that, though the ROC score is maintained, truncating G too aggressively results in many responses dropping dramatically. The same comments about there perhaps being a way to rescale G to alleviate this issue apply.

Table 5.9: Results from truncating G, 10 face discrimination task

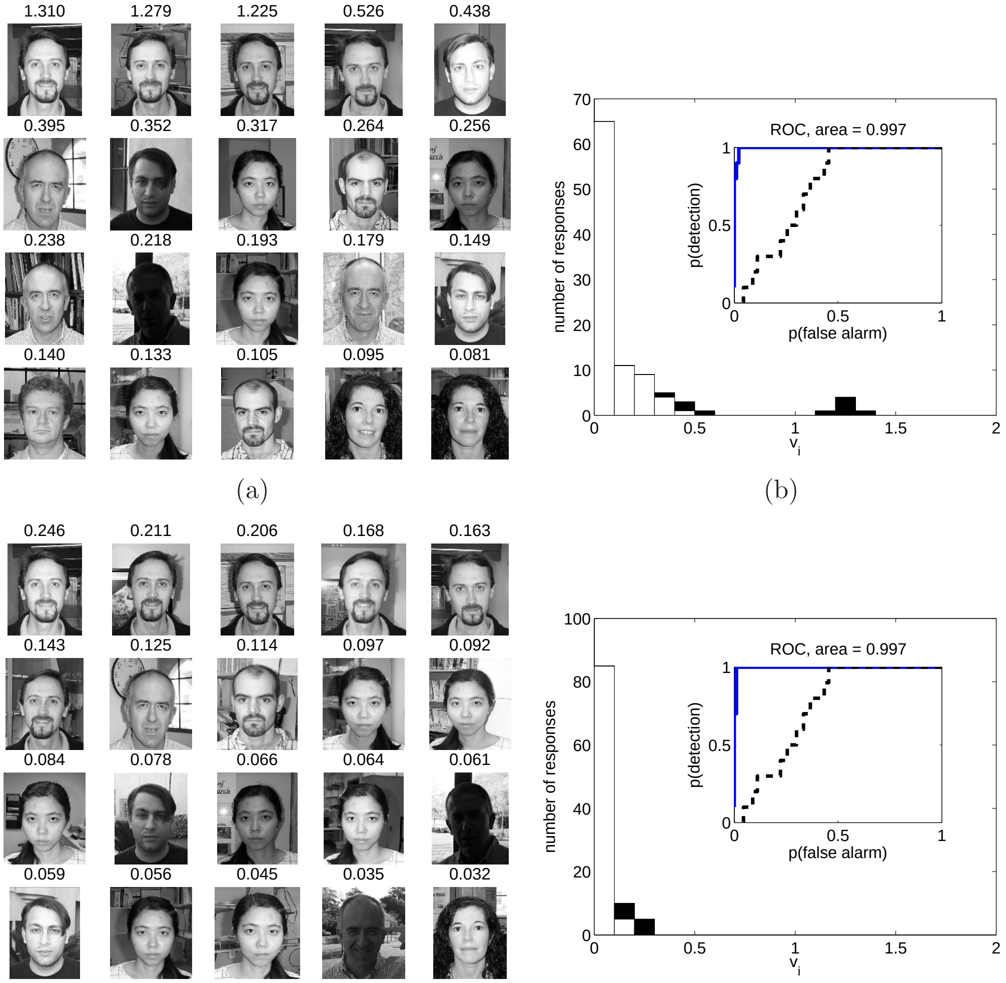
thresh (σ)	# nonzero	% nonzero	Metric 1	Metric 2	Metric 3
0.0	30000	100.0	91.1	56.0	56.0
0.5	16301	54.3	91.1	57.0	57.0
1.0	8777	29.3	91.1	56.0	57.0
1.5	4209	14.0	91.3	48.0	56.0
2.0	1592	5.3	91.2	49.0	56.0
2.5	515	1.7	89.3	45.0	50.0
3.0	163	0.5	89.0	48.0	49.0



(a) (b)

(c) (d)

Figure 5.19: Responses of the motorbike unit of Figure 5.1(c, d) after truncating G matrix. (a, b): 1.5σ truncation. (c, d): 2.5σ truncation. ROC equal-error accuracy was 89% at 1.5σ truncation and 82% at 2.5σ truncation.



(a) (b)

(c) (d)

Figure 5.20: Responses of the face unit of Figure 5.3(a, b) after truncating G matrix. (a, b): 1.5σ truncation. (c, d): 2.5σ truncation. ROC equal-error accuracy was 98% at 1.5σ truncation and 99% at 2.5σ truncation.

5.6.3 Summary

While preliminary, the results of this section are highly encouraging in that any reasonable model of real neural learning must be robust in the face of both limited connectivity and significant noise in synaptic weights. These two features of real networks are nicely modeled by truncation and quantization, respectively, and I have shown here that the trained network is very robust to these disturbances. The next step along these lines would be to implement some form of these errors during the learning process, as in the real brain these issues are present all the time. One challenge confronting this process is that some small weights may be necessary to give learning a place to take hold: if some column j of G (which, recall, can be seen as a basis function) becomes entirely zero due to truncation or quantization, inspection of the equations describing the algorithm reveals that the corresponding unit v_j will no longer respond, and further that the learning process will never cause the column to become nonzero again. If this proves to be an issue it may be necessary to either quantize not to zero but to some small nonzero number, or to include a noise source that occasionally randomly perturbs zero weights to nonzero values (possibly realistically reflecting new synaptic growth). On the plus side, however, implementing these disturbances during training may allow us to truncate (or quantize) aggressively and still maintain large responses, as G should automatically be rescaled by the learning process to keep some responses large.

5.7 Related Work

The problem I discussed in this chapter is clearly distinct from the more common approach to object recognition or classification in which a labeled training set is used to learn features common to the category. These features are then extracted from unlabeled images to classify them (Barnard et al., 2003). From a pure engineering standpoint, in many settings such as recognition of objects from previously learned templates, this supervised approach is likely to be the best one. However, the MTL data suggests that the brain is capable of forming internal representations of objects in the absense of explicit supervisory signals, the issue I explore here. Further, problems such as clustering and classification of large image databases will likely benefit from at least a partially unsupervised approach, as human labeling of all images may not always be feasible. Only a few examples of truly unsupervised image classification exist in the literature. The only directly comparable work is that of Sivic et al. (2005), who address much the same computational task using very different techniques. As in my work, they first compute a feature-based (as opposed to pixel-based) representation of images, but they do so using vector quantized SIFT descriptors (Lowe, 1999) where the quantized features are obtained from a k-means algorithm applied to descriptors from sample images from their input set. I instead obtain a feature-based representation using a more biologically plausible model of visual processing, the most recent extension of the HMAX model (Riesenhuber & Poggio, 1999; Serre, Oliva & Poggio, 2007). Sivic et al. then also apply a generative statistical model to the image features, using techniques developed for unsupervised topic discovery in text applied to the "words" (features) extracted from each image. An important distinction is that they found it important to restrict the number of categories (topics) searched for to the number truly present in their datasets, while my method is robust to varying numbers of input categories (and objects could in principle belong to multiple categories, though preliminary experiments along these lines have met mixed results). Nonetheless, the essential computational approach of first building a feature-based representation of images and then learning a generative statistical model for these features is the same.

Between the extremes of fully supervised and unsupervised classification lie a number of different approaches that can be described as "weakly" or "partially" supervised, in which at least some information about the stimulus set is provided to the algorithm. Fergus, Perona, and Zisserman (2003) use an unsupervised generative learning algorithm to build representations of particular image categories, but only images from a single category are presented to the model, which is then tested in a category-versus-background setting. Their model thus attempts to find the features common to *all* the images in the input set because it is known that they all come from the same category. In contrast, my model simultaneously learns representations for multiple image categories without *a priori* specification of the labels (or even the number of categories present). Weber, Welling, and Perona (2000) also cast the unsupervised categorization problem as emergent population coding, but again only present images from a single category at a time. The different "components" of their representation then correspond to different views or sub-categories of the input category, and each image is "explained" by a single component. In principal their method could be applied to an input set consisting of images from multiple categories and it should distinguish between them. As with Sivic et al., however, it would be important to specify the number of categories to be identified. Dong and Bhanu (2003) present a method for image search in which the user can specify whether or not returned images were relevant to the search. As in this work, image features are modeled as a Gaussian mixture dependent on the components (causes) present in the image, and the components of this model are estimated using unsupervised expectation maximization. Over time, a subset of images in the database are labeled though user feedback, and the system makes use of these labels to refine the category clustering.

Sparse coding as a tool for efficient representation and classification has attracted a great deal of attention in recent years, both in the context of vision and elsewhere. Olshausen and Field developed the algorithm I extend here and showed that, when applied to natural image patches, it generates feature selectivity much like that observed in simple cells in primary visual cortex (1996, 1997). Hinton and Ghahramani (1997) also cast sparse representation in a generative modeling framework, but, as with Olshausen and Field, they work directly at the image level. Mutch and Lowe (2006) improve the performance of one of the underlying vision system models I use here (HMAX), in part using sparsification to enhance selectivity lower in the hierarchy. They evaluate performance in a supervised setting by training a support-vector machine (SVM) for category selection. Ranzato et al. (2006) take an energy-based approach to the unsupervised learning of sparse representations of natural images and briefly discuss its extension to a hierarchical model, though their results are at a much lower level of the visual hierarchy and so do not address categorization. Their approach, if applied to a higher level of the feature hierarchy, may produce results similar to my own. The categorization task I discuss here can be viewed as a blind source separation problem. Li et al. (2004) discuss the utility of sparse coding applied to this problem, including the aspect that the number of sources is unknown. They consider several applications, including separating speech signals and separating mixed (superimposed) drawings of faces, but not the image categorization task I discuss here.

Chapter 6

Conclusions and Future Directions

In this final chapter I will revisit and tie together the topics discussed so far, then discuss a handful of ideas for carrying this work forward. In Section 6.1 I summarize the main results of this thesis and how they can be viewed as a whole. In the following sections I describe a few possible avenues of future research that could grow out of the work presented here. These can be broadly characterized as possible computational enhancements to the sparse coding model (Section 6.2) and as furthering the links to biological data (Section 6.3), though of course there will be significant cross-fertilization between these areas (as I hope has been the case throughout this work). Finally, in Section 6.4 I offer a few closing thoughts.

6.1 Summary of Thesis

The unifying theme of this thesis has of course been sparse coding and its computational utility. The primary motivating examples from biology are the extraordinary human MTL cells reported by Quian Quiroga and colleagues (2005). In my view the two most important questions to ask about such neurons are “How selective are they?” and “How did they come to be this way?” In Chapters 2 and 3 I answered the first question, first describing various methods by which it can be answered (and the strengths and limitations of different approaches) and then applying these methods to the MTL data. The result is that these cells are indeed highly selective, responding to perhaps 1% (or even much less) of all stimuli. These cells appear to me to sit at the top of a hierarchy devoted to extracting the sparse structure underlying the vastly complex bombardment of sensory information entering our brains from the world around us. Making this sparse structure explicit serves a variety of purposes: it makes readout of signals elsewhere in the brain (for example, to drive behavior in response to stimuli) very easy, it is metabolically efficient, and, if the hippocampus serves as some sort of "pointer table" indexing detailed memories stored in sensory cortex, it maximizes the number of patterns that can be stored therein and thus the number of memories indexed.

In Chapters 4 and 5 I moved on to the second question, investigating a neurally plausible scheme for learning a sparse code for sensory inputs. I first provided a theoretical discussion of sparse coding, building upon the work of Olshausen and Field and extending their model both to increase learning efficiency and to better serve as a model of the MTL data. Finally, I presented results from applying these ideas to the top of the visual processing hierarchy, showing that sparse coding as a computational constraint can naturally lead to the type of sparse, selective behavior observed in MTL. I believe these results compellingly illustrate the power of such computational models to better understand the biological data and, perhaps, to begin to explain it.

6.2 Computational Extensions

6.2.1 Extending the Scope of the Model

I have shown here that the same sparse coding model successfully employed by Olshausen and Field to model V1 can also be fruitfully applied to a much higher level of the visual hierarchy. That is not to say that the top and bottom of the hierarchy are the only places where sparse coding may be advantageous. In Chapter 3 I provided evidence that sparse codes may be advantageous from a metabolic point of view, and in Chapters 4 and 5 I argued for their computational utility. It is reasonable to expect that the principles of sparse coding described in this work could be fruitfully applied throughout the visual hierarchy to enhance the coding efficiency of visual inputs. In its current state, the feature selectivity of the HMAX network used to generate most of the results of Chapter 5 is simply memorized from a random selection of images. That is, in each S layer (aside from S_1, in which V1-like oriented bar filters are used), each neuron is given weights by propagating some image patch through the network up to that point and memorizing the resulting pattern of activity on its afferents as a template feature. While this method should capture the statistics of natural images, it makes no effort to build a particularly efficient representation as a sparse coding network does, and it must make use of millions of neurons in the intermediate layer in order to capture enough image features to support recognition. It is therefore reasonable to expect that applying the coding strategy described in this thesis throughout the hierarchy of a simple-complex processing network like HMAX (that is, at each simple cell stage) could provide a performance improvement both in fidelity of representation and in number of coding units required.

The primary obstacle to this approach is one of available computational resources—the intermediate layers of the vision model used here consist of millions of simulated neurons, and so the model is only tractable because these neurons operate in a purely feedforward fashion. By contrast, interactions between neurons in the same layer are crucially important to our sparse coding scheme, and so a more efficient means for computing the equilibrium of the network (and thus computing the representation) would be required. A few ideas may be of use here. First, if we assume the sparse coding network will truly learn a more efficient code for image features, we may be able to reduce the number of representing units. Second, we can exploit the sparse structure of the trained interconnection (G) matrix (as described in Chapter 5) to speed computation at the recognition stage, though this will not improve training speed. Finally, it may be possible to impose a sparse set of interconnections between neurons even during training rather than the full connectivity used in this work.

6.2.2 Multi-Modal Perception

One of the most striking results from human MTL reported by Quian Quiroga and colleagues was a neuron that, in addition to its robust invariant response to various images of the actress Halle Berry, responded vigorously to the letter string "Halle Berry" (Quian Quiroga et al., 2005). Furthermore, pilot data from continuing studies in this area reveal MTL cells that respond strongly to the name of their preferred stimulus spoken aloud by a computer (R. Quian Quiroga, *personal communication*). One intriguing area of future work is therefore to extend the computational work described in Chapters 4 and 5 to other forms of sensory input. In principal the same machinery for sparse coding should be sufficient—at the level of abstraction of the inputs to the sparse coding model, namely image features, nothing is specifically designed or tuned to the visual mode. Furthermore, evidence of neural plasticity across brain areas suggests that it may be worthwhile to seek general computational structures that apply across sensory modes (Pascual-Leone, Amedi, Fregni & Merabet, 2005, and references therein). The same methodology applied to an invariant representation of written or spoken words may be successful in extracting the sparse structure present therein.

Though written words (text) enter the brain through the visual system, evidence from fMRI experiments suggests that specialized machinery for the holistic processing of words develops in the *Visual Word Form Area* as reading skills are acquired (Gaillard et al., 2006; McCandliss, Cohen & Dehaene, 2003, and references therein) (but see criticisms of this view in Price and Devlin (2003)). For this reason it may be appropriate to treat written words as a distinct sensory mode and investigate the application of our coding methodology to it. To generate a representation of text invariant to transformations such as changes in scale and font one may either use one of many sophisticated systems for optical character recognition (OCR) currently available on any input images containing words, or simply apply the model directly to text represented as such. A minor distinction between this mode of input and the vision system model described above is the crucial importance of the spatial relationship between letters—in the vision model excellent performance is possible even when most spatial information is discarded, while in text rearranging letters generally destroys the meaning (though some robustness to this is present, as long as the first and last letters of a word are preserved).

In the auditory domain, one could use existing models of auditory language processing to project auditory signals into a space wherein the same word spoken by different individuals or otherwise manipulated will produce a similar representation. Smith and Lewicki (2005) discuss methods for developing efficient, shift-invariant representations for natural sounds using spiking models, among them a sparse generative model much like that employed by Olshausen and Field in the visual domain. Such a representation can then be fed into the sparse coding network, which could extract structure, such as commonly used words, from the input data stream.

The goal of this line of inquiry would be to replicate the multi-modal response characteristics observed in the human MTL recordings. Further, significant evidence from fMRI and psychophysical experiments indicate that this type of cross-modal interaction plays an important role in perception (Shimojo & Shams, 2001, and references therein). A likely strategy here would be to feed into the sparse coding network not inputs from a single sensory mode, but simultaneous inputs from multiple modes. For example, one would present the image of Halle Berry together with her name spoken aloud, each processed through the appropriate invariance model. The model will then be able to associate the inputs across modes, in essence allowing each mode to act as a supervisory signal for the other(s). Computational studies have already shown the utility of such multi-modal "self-supervision" in performing the unsupervised clustering task of learning vowels from spoken English using both auditory and visual information (Coen, 2006).

6.3 Enhancing the Link to Biology

A final important area of future research is to enhance the quantitative link of the model to real biological systems. I here describe a few areas in which either the link to biology could be strengthened, or in which it has not yet been fully investigated.

6.3.1 Neuronal Dynamics

The sparse coding model depends on neurons with a specific type of nonlinear self-inhibition (the $S'(v)$ term in the network dynamics), which enforces the constraint that the responses follow the sparse prior distribution. The physical meaning of this inhibition term has not been well explored either in this work or elsewhere. In the case of the prior distribution used here (the mixture of Gaussians), the input-output behavior of a single neuron with this nonlinearity is similar to that of a threshold-linear unit, and the dynamics can be approximated by a continuous firing-rate model of a leaky integrate-and-fire neuron. I have carried out some preliminary investigations in which I replaced the ideal sparse coding neurons with such a model with encouraging results (similar performance in the 4-category classification task). In fact, it may be possible to formalize this approximation as still constituting a gradient descent, though not a *steepest* descent, of the sparse coding cost function (B. Olshausen, *personal communication*). However, much more work is still needed to implement the sparse coding model with truly biophysically plausible neurons.

All of the computational results presented here were based on the steady-state responses of neurons to constant inputs. Another potential avenue for future work is to investigate the time-course of the neural responses and compare to the electrophysiological data. This work must be closely tied to the effort to implement the model using biophysically realistic neurons described above in order to have a hope of illuminating the data at a quantitative level.

6.3.2 Learning Dynamics

The learning model used here required numerous presentations of images from each category to be learned in order for the weights to converge to a point where sparse, selective behavior emerges. In contrast, it appears that the selective neurons observed in MTL may develop their representations in a relatively short time—cells have been observed that begin to respond to the clinical personnel and attending scientists after a number of hours of interactions with these people (R. Quian Quiroga, *personal communication*). While there is some evidence from studies in rat hippocampus that memories are “replayed” during sleep, effectively increasing the number of presentations (Hoffman & McNaughton, 2002), there still may be a gap between the relatively slow learning rate of the algorithms presented here and the speedy learning observed in experimental settings (and in our qualitative personal experience).

The evolution of the representation as learning progresses has not been carefully studied, though preliminary investigations have been interesting. For example, in the face discrimination task, the network first learns to distinguish between subsets of the individuals in the input set, then, as more information becomes available, it further differentiates between them. This bears a qualitative resemblance to how humans categorize data, with people only broadly categorizing rarely encountered classes but making fine distinctions between members of a more commonly seen (or more personally important) category. For example, many people may only recognize rough categories of automobiles such as “car,” “truck,” and “SUV,” while automotive enthusiasts make fine distinctions between different makes, models, and years of car.

6.4 Conclusion

I would like to close by repeating the quote from Barlow (1972) about sparse coding and perception:

> The central proposition is that our perceptions are caused by the activity of a rather small number of neurons selected from a very large population of predominantly silent cells. The activity of each single cell is thus an important perceptual event and it is thought to be related quite simply to our subjective experience. The subtlety and sensitivity of perception results from the mechanisms determining when a single cell becomes active, rather than from complex combinatorial rules of usage of nerve cells.

The MTL data strongly supports this “central proposition,” and my goals in this thesis have been to argue in favor of this point and to put forth a model for the “mechanisms determining when a single cell becomes active,” furthering at least a bit our understanding of the neural computations underlying our perception of the world around us.

Appendix A

Derivation of the Joint Probability of N_r and S_r

Here I describe how to calculate the joint probability of measuring N_r responsive neurons and S_r evocative stimuli given that we are recording from N neurons and presenting S stimuli. I will first derive a recursive relation for the conditional distribution of S_r given N_r, then solve the recurrence in closed form and apply Bayes' rule to obtain the joint distribution.

In what follows I assume the sparseness a is known, that is, all probability distributions are conditioned on a. First, let M be the number of stimuli among the S presented that a particular neuron responds to. The value of M follows a binomial distribution,

$$P[M = m] = \begin{pmatrix} S \\ m \end{pmatrix} a^m (1-a)^{S-m}.$$

If we assume that the neuron in question is responsive (i.e., $M \geq 1$), this distribution becomes (using Bayes' rule)

$$P[M = m | M \geq 1] = \frac{P[M = m]}{P[M \geq 1]} = \begin{pmatrix} S \\ m \end{pmatrix} \frac{a^m (1-a)^{S-m}}{1-(1-a)^S}. \qquad \text{(A.1)}$$

To begin our recursive definition, note that $P[S_r = s_r | N_r = 1] = P[M = s_r | M \geq 1]$.

Now assume the first $n_r - 1$ responsive neurons are excited by R stimuli. Let Q be the number of stimuli *not* in the set of size R that excite the next neuron (neuron n_r). The distribution of Q is given by

$$P[Q = q|R = r] = \sum_{m=q}^{r+q} P[M = m|M \geq 1] \frac{\begin{pmatrix} r \\ m-q \end{pmatrix} \begin{pmatrix} S-r \\ q \end{pmatrix}}{\begin{pmatrix} S \\ m \end{pmatrix}}, \qquad \text{(A.2)}$$

where $q \in \{0, \ldots, S - m\}$. The first term in the sum is simply the probability that the neuron in question responds to m stimuli total. The second term is the number of ways these m stimuli could be split such that q of them are not in the set of size r divided by the total number of ways these m stimuli could be chosen, so it is the probability that the m stimuli include exactly q stimuli not already in the set responded to by the first $n_r - 1$ neurons.

With a little effort we can find a closed-form expression for Equation A.2. If $q > 0$, we can combine equations A.1 and A.2 and pull out terms unrelated to the sum to obtain

$$P[Q = q|R = r] = \begin{pmatrix} S-r \\ q \end{pmatrix} \frac{(1-a)^S}{1-(1-a)^S} \sum_{m=q}^{r+q} \begin{pmatrix} r \\ m-q \end{pmatrix} \left(\frac{a}{1-a}\right)^m.$$

Substituting $m' = m - q$ this becomes

$$P[Q = q|R = r] = \begin{pmatrix} S-r \\ q \end{pmatrix} \frac{(1-a)^S}{1-(1-a)^S} \left(\frac{a}{1-a}\right)^q \sum_{m'=0}^{r} \begin{pmatrix} r \\ m' \end{pmatrix} \left(\frac{a}{1-a}\right)^{m'}.$$

Applying the binomial theorem to the sum we have

$$P[Q=q|R=r] = \begin{pmatrix} S-r \\ q \end{pmatrix} \frac{(1-a)^S}{1-(1-a)^S} \left(\frac{a}{1-a}\right)^q \left(\frac{1}{1-a}\right)^r \qquad (q>0). \quad \text{(A.3)}$$

If $q=0$, the first term in the sum vanishes and we have instead

$$P[Q=0|R=r] = \frac{(1-a)^S}{1-(1-a)^S} \sum_{m=1}^{r} \begin{pmatrix} r \\ m \end{pmatrix} \left(\frac{a}{1-a}\right)^m .$$

Again applying the binomial theorem this becomes

$$P[Q=0|R=r] = \frac{(1-a)^S}{1-(1-a)^S} \left[\left(\frac{1}{1-a}\right)^r - 1\right]. \qquad \text{(A.4)}$$

We can combine Equations A.3 and A.4 to obtain the final result,

$$P[Q=q|R=r] = \begin{pmatrix} S-r \\ q \end{pmatrix} \frac{(1-a)^S}{1-(1-a)^S} \left(\frac{a}{1-a}\right)^q \left[\left(\frac{1}{1-a}\right)^r - \delta(q)\right], \quad \text{(A.5)}$$

where $\delta(q)=1$ if $q=0$, and $\delta(q)=0$ otherwise.

The relationship in Equation A.5 now lets us complete the recursive definition of the conditional distribution of S_r given N_r:

$$P[S_r=s_r|N_r=n_r] = \sum_{y=1}^{s_r} P[S_r=y|N_r=n_r-1]P[Q=s_r-y|M=y]. \qquad \text{(A.6)}$$

Simply put, this is the probability that neuron n_r adds just enough new stimuli to the set responded to by the first n_r-1 neurons to total s_r. Since we calculated the base case $P[S_r=s_r|N_r=1]$ above, this probability can be calculated for any n_r and s_r by starting at $N_r=1$ and working upward.

Next I solve the recurrence to find a simpler expression for this probability and eliminate the need to calculate the needed probability recursively.

Proposition A.1. *The recurrence given in Equation A.6 has solution*

$$\begin{aligned} P[S_r = s_r | N_r = n_r] &= \begin{pmatrix} S \\ s_r \end{pmatrix} \left[\frac{(1-a)^S}{1-(1-a)^S} \right]^{n_r} (-1)^{n_r} \\ & \sum_{k=1}^{n_r} \begin{pmatrix} n_r \\ k \end{pmatrix} (-1)^k \left[(1-a)^{-k} - 1 \right]^{s_r}. \end{aligned} \tag{A.7}$$

Proof. For convenience, define

$$\begin{aligned} G &\equiv \frac{(1-a)^S}{1-(1-a)^S}, \\ H &\equiv \frac{a}{1-a}. \end{aligned}$$

Then the recurrence we are trying to solve is

$$\begin{aligned} P[S_r = s_r | N_r = n_r] &= \sum_{y=1}^{s_r} P[S_r = y | N_r = n_r - 1] P[Q = s_r - y | R = y] \qquad (A.8) \\ &= G \Bigg\{ H^{s_r} \sum_{y=1}^{s_r} P[S_r = y | N_r = n_r - 1] \begin{pmatrix} S - y \\ s_r - y \end{pmatrix} a^{-y} \\ & -P[S_r = y | N_r = n_r - 1] \Bigg\} \\ P[S_r = s_r | N_r = 1] &= \begin{pmatrix} S \\ s_r \end{pmatrix} H^{s_r} G, \end{aligned}$$

and the proposed solution is

$$P[S_r = s_r | N_r = n_r] = \begin{pmatrix} S \\ s_r \end{pmatrix} G^{n_r} (-1)^{n_r} \sum_{k=1}^{n_r} \begin{pmatrix} n_r \\ k \end{pmatrix} (-1)^k \left[\left(\frac{H}{a} \right)^k - 1 \right]^{s_r}. \tag{A.9}$$

We will prove the result using induction. For the base case, let $n_r = 1$. Then

$$\begin{aligned} P[S_r = s_r | N_r = 1] &= \left(\begin{array}{c} S \\ s_r \end{array} \right) G(-1)(-1) \left[\frac{H}{a} - 1 \right]^{s_r} \\ &= \left(\begin{array}{c} S \\ s_r \end{array} \right) G H^{s_r}, \end{aligned}$$

the desired result. For the inductive step, assume the result holds for $n_r - 1$. Then substituting into Equation A.8 we have

$$\begin{aligned} P[S_r = s_r | N_r = n_r] &= G \Bigg\{ H^{s_r} \sum_{y=1}^{s_r} \left(\begin{array}{c} S \\ y \end{array} \right) G^{n_r - 1} (-1)^{n_r - 1} \\ & \sum_{k=1}^{n_r - 1} \left(\begin{array}{c} n_r - 1 \\ k \end{array} \right) (-1)^k \left[\left(\frac{H}{a} \right)^k - 1 \right]^y \\ & \left(\begin{array}{c} S - y \\ s_r - y \end{array} \right) a^{-y} - \left(\begin{array}{c} S \\ s_r \end{array} \right) G^{n_r - 1} (-1)^{n_r - 1} \\ & \sum_{k=1}^{n_r - 1} \left(\begin{array}{c} n_r - 1 \\ k \end{array} \right) (-1)^k \left[\left(\frac{H}{a} \right)^k - 1 \right]^{s_r} \Bigg\} \\ &= G^{n_r} (-1)^{n_r} \left(\begin{array}{c} S \\ s_r \end{array} \right) \sum_{k=1}^{n_r} \left(\begin{array}{c} n_r \\ k \end{array} \right) (-1)^k \left[\left(\frac{H}{a} \right)^k - 1 \right]^{s_r}, \end{aligned}$$

where the simplification results from extensive algebraic manipulation and applications of the binomial theorem. This is exactly the proposed solution from Equation A.9 and the proof is complete. □

To obtain the joint probability of measuring N_r responsive neurons *and* S_r stimuli to which they respond from Bayes' rule, we will need the probability of measuring

N_r responsive neurons independent of S_r. The probability that a neuron responds to *some* stimulus, which we denote p_r, is 1 minus the probability that it responds to no stimuli, or

$$p_r = 1 - (1-a)^S.$$

The number of responsive neurons then follows a binomial distribution,

$$\begin{aligned} P[N_r = n_r] &= \begin{pmatrix} N \\ n_r \end{pmatrix} p_r^{n_r} (1-p_r)^{N-n_r} \\ &= \begin{pmatrix} N \\ n_r \end{pmatrix} \left[1 - (1-a)^S\right]^{n_r} (1-a)^{S(N-n_r)}. \end{aligned}$$

We are now ready to apply Bayes' rule to obtain the desired probability,

$$\begin{aligned} P[N_r = n_r \wedge S_r = s_r] &= P[S_r = s_r | N_r = n_r] P[N_r = n_r] \\ &= \begin{pmatrix} S \\ s_r \end{pmatrix} \left[\frac{(1-a)^S}{1-(1-a)^S}\right]^{n_r} (-1)^{n_r} \\ &\quad \sum_{k=1}^{n_r} \begin{pmatrix} n_r \\ k \end{pmatrix} (-1)^k \left[(1-a)^{-k} - 1\right]^{s_r} \\ &\quad \begin{pmatrix} N \\ n_r \end{pmatrix} \left(1-(1-a)^S\right)^{n_r} (1-a)^{S(N-n_r)} \\ &= \begin{pmatrix} S \\ s_r \end{pmatrix} \begin{pmatrix} N \\ n_r \end{pmatrix} (1-a)^{NS} (-1)^{n_r} \\ &\quad \sum_{k=1}^{n_r} \begin{pmatrix} n_r \\ k \end{pmatrix} (-1)^k \left[(1-a)^{-k} - 1\right]^{s_r}. \end{aligned} \tag{A.10}$$

Equation A.10 has been verified to match Monte Carlo results within 5% for all cases in which the number of trials (10^8 simulated sessions) was statistically significant (10^8 trials is sufficient to measure a probability of 0.01 to within 5% with 99% confidence) for select values of a. Note also that it can be shown that if the roles of N and S and those of n_r and s_r are reversed Equation A.9 does not change, an expected result due to the symmetry of the problem. Furthermore, summing Equation A.9 over all s_r or n_r yields the expected marginal distributions. We should also note that Equation A.9 is numerically very poorly conditioned, as the binomial coefficients can easily produce numbers much larger than machine precision allows. Hence care is needed when evaluating these probabilities numerically. In some cases wildly inaccurate results were obtained using MATLAB, and it was necessary to make use of Mathematica's arbitrary-precision capabilities to generate meaningful results.

Note that all of the above assumed a was known, so replacing a by α in the derived distribution we obtain the conditional distribution $P[N_r = n_r \wedge S_r = s_r | a = \alpha]$.

Appendix B

Convergence of EM for Sparse Coding

I here show that the EM algorithm for sparse coding described in Chapter 3 converges. The cost function to be maximized is:

$$\mathcal{F}(\hat{v}, G) = \left\langle -\frac{1}{2\lambda}\|u - G\hat{v}\|^2 + \sum_{j=1}^{m} S(\hat{v}_j) \right\rangle - \frac{1}{2\gamma} \sum_{j=1}^{m} \sum_{i=1}^{n} g_{ij}^2 \tag{B.1}$$

where to simplify the notation we denote $\hat{v}(u)$ by $\hat{v}$.

The algorithm is as follows:

Initially: $\hat{v}^{(0)}(u) = \mathbf{0}$ for all $u \in \{u\}$, $G^{(0)} = rand(n, m)$.

E step: For each $u \in \{u\}$, compute $\hat{v}^{(k+1)}$ by gradient ascent on $\mathcal{F}$ starting at $\hat{v}^{(k)}$ with $G = G^{(k)}$. That is, simulate the differential equation

$$\dot{v} = \nabla_v \mathcal{F} = G^T(u - Gv) + \lambda S'(v) \tag{B.2}$$

until $\|\dot{v}\|$ falls below some convergence threshold $\dot{v}_T$.

M step: Set $G^{(k+1)}$ according to the update rule

$$G^{(k+1)} = \langle u\hat{v}^T \rangle \left(\frac{\lambda}{\gamma} I + \langle \hat{v}\hat{v}^T \rangle \right)^{-1} \tag{B.3}$$

with $v = v^{(k+1)}$.

Lemma B.1. *The cost function $\mathcal{F}(\hat{v}, G)$ is bounded from above.*

Proof. Clearly

$$\mathcal{F}(\hat{v}, G) \leq \left\langle \sum_{i=1}^{m} S(\hat{v}_i) \right\rangle .$$

Recall $\exp(S)$ is a continuous probability distribution. Hence $\exp(S)$ has a finite maximum and so S has a finite maximum as well, which I denote by S_{max}. Then $\mathcal{F}(\hat{v}, G) \leq mS_{max}$. □

Proposition B.2. *The update rule given by Equation B.3 yields the G that* globally *maximizes $\mathcal{F}$ (for fixed $\hat{v}$).*

Proof. Let $G^*(\hat{v})$ be the value of G given by B.3, and let $\mathcal{H}(G) = \mathcal{F}(\hat{v}, G)$. Let $\Omega_c = \{G | \mathcal{H}(G) \geq c\}$, and choose $c < \mathcal{H}(G^*)$ so G^* lies in the interior of Ω_c. Note that $\mathcal{H}(G) \to -\infty$ as $\|G\| \to \infty$ (where $\|G\|$ denotes the norm of G taken as a vector), so Ω_c is compact. $\mathcal{H}$ is a continuous function on Ω_c, so it must have a maximum value in Ω_c. This maximum is not on the boundary of Ω_c because on the boundary $\mathcal{H} = c < \mathcal{H}(G^*)$, and so it must be the case that $\frac{\partial \mathcal{H}}{\partial G} = 0$ at the maximum. G^* is the only point at which this is the case, and so it must be the maximum of $\mathcal{H}$ on Ω_c. $\mathcal{H}(G) < c < \mathcal{H}(G^*)$ for G outside Ω_c and so G^* is the global maximum of $\mathcal{H}$. □

Lemma B.3. *The cause estimate $\hat{v}$ is bounded, that is, there exists some constant $c < \infty$ such that $\|\hat{v}\| < c$.*

Proof. The E step is a gradient ascent with respect to $\hat{v}$ on $\mathcal{F}$, and by Proposition B.2 the M step yields the global maximum of $\mathcal{F}$ with respect to G, so $\mathcal{F}$ is bounded from below by its initial value $\mathcal{F}_0$. Recall $\exp(S(\hat{v}))$ is a continuous probability distribution, so $\exp(S(\hat{v})) \to 0$ as $\|\hat{v}\| \to \infty$ and $S(\hat{v}) \to -\infty$ as $\|\hat{v}\| \to \infty$. The other term of $\mathcal{F}$ involving $\hat{v}$ is $-\frac{1}{2\lambda}\|u - G\hat{v}\|^2$, which also goes to $-\infty$ as $\|\hat{v}\| \to \infty$. Hence $\mathcal{F} \to -\infty$ as $\|\hat{v}\| \to \infty$, but $\mathcal{F}$ is bounded from below, so $\|\hat{v}\|$ must be bounded. □

Proposition B.4. *The E step converges to locally optimal cause estimates $\hat{v}(u)$.*

Proof. For fixed G, define the *Lyapunov function*

$$\mathcal{V}(\hat{v}) = -\mathcal{F}(\hat{v}, G). \tag{B.4}$$

The derivative of $\mathcal{V}$ is

$$\dot{\mathcal{V}} = -\nabla_v \mathcal{F} \dot{v} = -\|\dot{v}\|^2 \leq 0, \tag{B.5}$$

with equality if and only if $\|\dot{v}\| = 0$. By Lemma B.3 $\|v\|$ is bounded, so LaSalle's invariance principle states that the dynamics converge to the largest invariant set M such that $\dot{v} = 0$ for $v \in M$ (LaSalle, 1976). Further, Theorem 2.7.8 of LaSalle (1976) states that M is (locally) asymptotically stable, so it consists of local minimizers (with respect to $\hat{v}$) of $\mathcal{V}$, or maximizers of $\mathcal{F}$. □

Note that I have *not* shown that v converges to a particular unique equilibrium point, only that its derivative $\dot{v}$ goes to zero. This is sufficient to show that this step of the algorithm will converge. However, if we make the additional (very reasonable) assumption that the equilibrium points of the dynamics of v are isolated then it follows that v converges to such a point.

Theorem B.5. *The cause estimate $\hat{v}(u)$ and weight matrix G converge to a closed set of local maximizers of $\mathcal{F}$.*

Proof. Let $G^*(\hat{v})$ be the value of G given by B.3, and define the Lyapunov function $\mathcal{V}(\hat{v}) = -\mathcal{F}(\hat{v}, G^*(\hat{v}))$, that is, $\mathcal{V}$ is the negative of the cost function after the M step. Let $T(\hat{v})$ be the mapping of $\hat{v}$ through the E step. Define

$$\Delta\mathcal{V}(\hat{v}) = \mathcal{V}(T(\hat{v})) - \mathcal{V}(\hat{v}),$$

that is, $\Delta\mathcal{V}$ is the change in $\mathcal{V}$ across a full EM iteration. By the fact that the E step is gradient ascent on $\mathcal{F}$ and Proposition B.2, it is the case that $\Delta\mathcal{V} \leq 0$. By Lemma B.3 $\|\hat{v}\|$ is bounded, so by a discrete version of LaSalle's invariance principle (LaSalle, 1976) the limit set Ω_v is contained in the largest invariant set M such that $\Delta\mathcal{V}(\hat{v}) = 0$ in M. Thus $\hat{v}$ converges to a closed set of local minima of $\mathcal{V}$. In the M step G is a continuous function of $\hat{v}$, so this implies that G converges to the closed set

$$\Omega_G = \{G|G = G(\hat{v})\, for\, some\, \hat{v} \in \Omega_v\}.$$

Define the limit set of the EM algorithm,

$$\Omega = \{(\hat{v}, G)|\hat{v} \in \Omega_v,\, G = G(\hat{v})\}.$$

This is a closed set because both Ω_v and Ω_G are. By Proposition B.4 $\hat{v}$ is locally optimal at points in Ω_v, and G is globally optimal at points in Ω, so points in Ω are local maximizers of $\mathcal{F}$. □

In practice, not only does the algorithm converge to some set of maximizers, it always converges to a particular fixed point $(\hat{v}^*, G^*)$.

References

Abbott, L. & Rolls, E. (1996). Representational Capacity of Face Coding in Monkeys. *Cerebral Cortex*, *6*, 498–505.

Attwell, D. & Laughlin, S. (2001). An Energy Budget for Signaling in the Grey Matter of the Brain. *Journal of Cerebral Blood Flow and Metabolism*, *21*, 1133–1145.

Barlow, H. (1972). Single units and sensation: A neuron doctrine for perceptual psychology? *Perception*, *1*, 371–394.

Barnard, K., Duygulu, P., Freitas, N. de, Forsyth, D., Blei, D. & Jordan, M. (2003). Matching words and pictures. *Journal of Machine Learning Research*, *3*, 1107–1135.

Bell, A. & Sejnowski, T. (1995). An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, *7*, 1129–1159.

Bell, A. & Sejnowski, T. (1997). The 'Independent Components' of Natural Scenes are Edge Filters. *Vision Research*, *37*(23), 3327–3338.

Bickel, P. & Lehmann, E. (1975). Descriptive Statistics for Nonparametric Models I. Introduction. *The Annals of Statistics*, *33*(5), 1038–1044.

Biederman, I. (1987). Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, *94*(2), 115–147.

Bruce, C., Desimone, R. & Gross, C. (1981). Visual Properties of Neurons in a Polysensory Area in Superior Temporal Sulcus of the Macaque. *Journal of Neurophysiology*, *46*(2), 369–384.

Coen, M. (2006). Self-Supervised Acquisition of Vowels in American English. In

Proceedings of the 21st *National Conference on Artificial Intelligence (AAAI 2006).*

Darlington, R. (1970). Is Kurtosis Really 'Peakedness?' . *The American Statistician*, *24*(2), 19–22.

Dayan, P. & Abbott, L. (2001). *Theoretical Neuroscience*. The MIT Press.

Dempster, A., Laird, N. & Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, *39*(1), 1–38.

DeWeese, M., Wehn, M. & Zador, A. (2003). Binary Spiking in Auditory Cortex. *Journal of Neuroscience*, *23*(21), 7940–7949.

Dong, A. & Bhanu, B. (2003). A New Semi-Supervised EM Algorithm for Image Retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003).*

Everingham, M., Sivic, J. & Zisserman, A. (2006). Hello! My name is... Buffy—Automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference (BMVC 2006).*

Fergus, R., Perona, P. & Zisserman, A. (2003). Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003).*

Fiete, I., Hahnloser, R., Fee, M. & Seung, H. (2004). Temporal Sparseness of the Premotor Drive Is Important for Rapid Learning in a Neural Network Model of Birdsong. *Journal of Neurophysiology*, *92*, 2274–2282.

Fried, I., MacDonald, K. & Wilson, C. (1997). Single Neuron Activity in Human Hippocampus and Amygdala during Recognition of Faces and Objects. *Neuron*, *18*, 753–765.

Gaillard, R., Naccache, L., Pinel, P., Clemenceau, S., Volle, E., Hasboun, D. et al. (2006). Direct Intracranial, fMRI, and Lesion Evidence for the Causal Role of

Left Inferotemporal Cortex in Reading. *Neuron*, *50*, 191–204.

Griffin, G., Holub, A. & Perona, P. (2006). *The Caltech 256* (Technical Report). Caltech Computation and Neural Systems.

Gross, C. (1992). Representation of visual stimuli in inferior temporal cortex. *Philosophical Transactions of the Royal Society of London B*, *335*, 3–10.

Gross, C. (2002). Genealogy of the Grandmother Cell. *Neuroscientist*, *8*(5), 512–518.

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B. & Moser, E. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, *436*, 801–806.

Hahnloser, R., Kozhevnikov, A. & Fee, M. (2002). An ultra-sparse code underlies the generation of neural sequences in a songbird. *Nature*, *419*, 65–70.

Hancock, P., Baddeley, R. & Smith, L. (1992). The principal components of natural images. *Network*, *3*, 61–70.

Harding, A., Halliday, G. & Kril, J. (1998). Variation in Hippocampal Neuron Number with Age and Brain Volume. *Cerebral Cortex*, *8*(8), 710–718.

Harris, C. & Stephens, M. (1988). A Combined Corner and Edge Detector. In *Alvey Vision Conference.*

Henze, D., Borhegyi, Z., Csicsvari, J., Mamiya, A., Harris, K. & Buzsaki, G. (2000). Intracellular Features Predicted by Extracellular Recordings in the Hippocampus In Vivo. *Journal of Neurophysiology*, *84*, 390–400.

Herault, J. & Jutten, C. (1986). Space or time adaptive signal processing by neural network models. In J. Denker (Ed.), *Neural Networks for Computing.* American Institute for Physics.

Hinton, G. & Ghahramani, Z. (1997). Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society of London B*, *352*, 1177–1190.

Hoffman, K. & McNaughton, B. (2002). Sleep on it: cortical reorganization after-the-fact. *Trends in Neurosciences*, *23*(1), 1–2.

Holub, A. & Moreels, P. (2007). *Searching for Pictures of Grandma: Look at the Eyes!* (Technical Report). Caltech Computation and Neural Systems.

Hubel, D. & Wiesel, T. (1962). Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat's Visual Cortex. *Journal of Physiology*, *160*, 106–154.

Konorski, J. (1967). *Integrative activity of the brain; an interdisciplinary approach.* University of Chicago Press.

Kreiman, G., Hung, C., Kraskov, A., Quian Quiroga, R., Poggio, T. & DiCarlo, J. (2006). Object Selectivity of Local Field Potentials and Spikes in the Macaque Inferior Temporal Cortex. *Neuron*, *49*, 433–445.

Kullback, S. & Leibler, R. (1951). On Information and Sufficiency. *Annals of Mathematical Statistics*, *22*(1), 79–86.

LaSalle, J. (1976). *The Stability of Dynamical Systems.* Society of Industrial and Applied Mathematics.

Lennie, P. (2003). The Cost of Cortical Computation. *Current Biology*, *13*, 493–497.

Leon-Garcia, A. (1994). *Probability and Random Processes for Electrical Engineers* (2^{nd} ed.). Addison-Wesley.

Lewicki, M. (2002). Efficient coding of natural sounds. *Nature Neuroscience*, *5*(4), 356–363.

Li, Y., Cichocki, A. & Amari, S. (2004). Analysis of Sparse Representation and Blind Source Separation. *Neural Computation*, *16*, 1193–1234.

Lindeberg, T. (1998). Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, *30*(2), 79–116.

Lowe, D. (1999). Object Recognition from Local Scale-Invariant Features. In *International Conference on Computer Vision (ICCV 1999).*

Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Martin, K. (1994). A Brief History of the Feature Detector. *Cerebral Cortex*, *4*, 1–7.

McCandliss, B., Cohen, L. & Dehaene, S. (2003). The visual word form area: expertise for reading in the fusiform gyrus. *Trends in Cognitive Sciences*, *7*(7), 293–299.

Meunier, C., Yanai, H. & Amari, S. (1991). Sparsely coded associative memories: capacity and dynamical properties. *Network*, *2*, 469–487.

Mikolajczyk, K. & Schmid, C. (2004). Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, *60*(1), 63–86.

Moors, J. (1986). The Meaning of Kurtosis: Darlington Reexamined. *The American Statistician*, *40*(4), 283–284.

Mutch, J. & Lowe, D. (2006). Multiclass Object Recognition with Sparse, Localized Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006).*

O'Keefe, J. & Dostrovsky, J. (1971). The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, *34*(1), 171–175.

O'Keefe, J. & Speakman, A. (1997). Single unit activity in the rat hippocampus during a spatial memory task. *Experimental Brain Research*, *68*(1), 1–27.

Olshausen, B. & Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*, 607–609.

Olshausen, B. & Field, D. (1997). Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vision Research*, *37*(23), 3311–3325.

Olshausen, B. & Field, D. (2004). Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, *14*, 481–487.

Pascual-Leone, A., Amedi, A., Fregni, F. & Merabet, L. (2005). The Plastic Human Brain Cortex. *Annual Review of Neuroscience*, *28*, 377–401.

Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, *2*, 559–572.

Perez-Orive, J., Mazor, O., Turner, G., Cassenaer, S., Wilson, R. & Laurent, G. (2002). Oscillations and Sparsening of Odor Representations in the Mushroom Body. *Science*, *297*, 359–365.

Perrett, D., Rolls, E. & Caan, W. (1982). Visual neurones responsive to faces in the monkey temporal cortex. *Experimental Brain Research*, *47*(3), 329–342.

Price, C. & Devlin, J. (2003). The myth of the visual word form area. *NeuroImage*, *19*, 472–481.

Quian Quiroga, R., Reddy, L., Koch, C. & Fried, I. (2007). Decoding visual inputs from multiple neurons in the human temporal lobe. *Journal of Neurophysiology.* (to appear)

Quian Quiroga, R., Reddy, L., Kreiman, G., Koch, C. & Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, *435*, 1102–1107.

Ranzato, M., Poultney, C., Chopra, S. & LeCun, Y. (2006). Efficient Learning of Sparse Representations with an Energy-Based Model. In *Advances in Neural Information Processing (NIPS 2006).*

Riesenhuber, M. & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, *2*(11), 1019–1025.

Rolls, E., Perrett, D., Caan, A. & Wilson, F. (1982). Neuronal Responses Related to Visual Recognition. *Brain*, *105*(4), 611–646.

Rolls, E. & Tovee, M. (1995). Sparseness of the Neuronal Representation of Stimuli in the Primate Temporal Visual Cortex. *Journal of Neurophysiology*, *73*(2), 713–726.

Rolls, E., Xiang, J. & Franco, L. (2005). Object, Space, and Object-Space Representations in the Primate Hippocampus. *Journal of Neurophysiology*, *94*, 833–844.

Roweis, S. (1997). EM Algorithms for PCA and SPCA. In *Neural Information Processing Systems 10.*

Roweis, S. & Ghahramani, Z. (1999). A Unifying Review of Linear Gaussian Models. *Neural Computation*, *11*, 305–345.

Ruppert, D. (1987). What Is Kurtosis? An Influence Function Approach. *The American Statistician*, *41*(1), 1–5.

Rutishauser, U., Mamelak, A. & Schuman, E. (2006). Single-Trial Learning of Novel Stimuli by Individual Neurons of the Human Hippocampus-Amygdala Complex. *Neuron*, *49*, 805–813.

Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G. & Poggio, T. (2005). *A Theory of Object Recognition: computations and circuits in the feedforward path of the ventral stream in primate visual cortex* (Technical Report No. MIT-CSAIL-TR-2005-036). MIT Computer Science and Artificial Intelligence Laboratory.

Serre, T., Oliva, A. & Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Science*, *104*(15), 6424–6429.

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M. & Poggio, T. (2007). Object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(3), 411–426.

Shimojo, S. & Shams, L. (2001). Sensory modalities are not separate modalities: plasticity and interactions. *Current Opinion in Neurobiology*, *11*, 505–509.

Sivic, J., Russell, B., Efros, A., Zisserman, A. & Freeman, W. (2005). *Discovering Object Categories in Image Collections* (Technical Report No. MIT-CSAIL-TR-2005-012). MIT Computer Science and Artificial Intelligence Laboratory.

Smith, E. & Lewicki, M. (2005). Efficient Coding of Time-Relative Structure Using Spikes. *Neural Computation*, *17*, 19–45.

Tanaka, K. (1997). Mechanisms of visual object recognition: monkey and human studies. *Current Opinion in Neurobiology*, *7*, 523–529.

Thompson, L. & Best, P. (1989). Place Cells and Silent Cells in the Hippocampus of Freely-Behaving Rats. *Journal of Neuroscience*, *9*(7), 2832–2390.

Treves, A. & Rolls, E. (1991). What determines the capacity of autoassociative memories in the brain? *Network*, *2*, 371–397.

Tsodyks, M. & Feigel'man, M. (1988). The Enhanced Storage Capacity in Neural Networks with Low Activity Level. *Europhysics Letters*, *6*(2), 101–105.

Ulanovsky, N. & Moss, C. (2007). Hippocampal cellular and network activity in freely moving echolocating bats. *Nature Neuroscience*, *10*(2), 224–233.

Vinje, W. & Gallant, J. (2000). Sparse Coding and Decorrelation in Primary Visual Cortex During Natural Vision. *Science*, *287*, 1273–1276.

Viola, P. & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001).*

Viskontas, I., Knowlton, B., Steinmetz, P. & Fried, I. (2006). Differences in Mnemonic Processing by Neurons in the Human Hippocampus and Parahippocampal Regions. *Journal of Cognitive Neuroscience*, *18*, 1654–1662.

Waydo, S. & Koch, C. (2007a). Unsupervised Category Discovery in Images Using Sparse Neural Coding. In *Proceedings of the British Machine Vision Conference (BMVC 2007).*

Waydo, S. & Koch, C. (2007b). Unsupervised Learning of Individuals and Categories from Images. *Neural Computation.* (in press)

Waydo, S., Kraskov, A., Quian Quiroga, R., Fried, I. & Koch, C. (2006). Sparse Representation in the Human Medial Temporal Lobe. *Journal of Neuroscience*, *26*(40), 10232–10234.

Weber, M., Welling, M. & Perona, P. (2000). Towards Automatic Discovery of Object Categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000).*

Weliky, M., Fiser, J., Hunt, R. & Wagner, D. (2003). Coding of Natural Scenes in Primary Visual Cortex. *Neuron*, *37*, 703–718.

Willmore, B. & Tolhurst, D. (2001). Characterizing the sparseness of neural codes. *Network: Computation in Neural Systems*, *12*, 255–270.

Young, M. & Yamane, S. (1992). Sparse Population Coding of Faces in the Inferotemporal Cortex. *Science*, *256*(5061), 1327–1331.