Title
Modeling neural representation using statistical features of natural scenes

Permalink
https://escholarship.org/uc/item/9hm2x32g

Author
Stansbury, Dustin

Publication Date
2014

Peer reviewed|Thesis/dissertation
Modeling neural representation using statistical features of natural scenes

by

Dustin Edward Stansbury

A dissertation submitted in partial satisfaction of the requirements for the degree of
Doctor of Philosophy

in

Vision Science

in the

Graduate Division
of the
University of California, Berkeley

Committee in charge:
Jack L. Gallant, Chair
Bruno A. Olshausen
Thomas L. Griffiths

Spring 2014
Modeling neural representation using statistical features of natural scenes

Copyright 2014
by
Dustin Edward Stansbury
Contents

List of Acronyms iv

Notation vi

1 Introduction 1
  1.1 Motivation .................................................. 2
  1.2 Organization ................................................. 3
  1.3 Published appearances of dissertation work ............... 4

2 Background & Approach 5
  2.1 The primate visual hierarchy ................................. 6
      2.1.1 The retina and LGN ..................................... 6
      2.1.2 Primary visual cortex ................................. 7
      2.1.3 The ventral and dorsal streams ......................... 9
  2.2 Modeling neural representation in the visual pathway ... 11
      2.2.1 System identification .................................. 11
      2.2.2 Linearized models ..................................... 12
  2.3 Statistical structure of natural scenes ..................... 16
      2.3.1 Local pixel statistics .................................. 17
      2.3.2 Pairwise pixel statistics .............................. 17
      2.3.3 Higher-order spatial statistics ......................... 20
      2.3.4 Spatio-temporal statistics ............................. 23
      2.3.5 Other forms of structure in natural scenes .......... 24
  2.4 Coding natural scene structure and models of neural representation 24
      2.4.1 Non-redundant codes .................................. 25
      2.4.2 Sparse, distributed, and overcomplete codes ........ 26
      2.4.3 Alternative image coding strategies ................. 29
      2.4.4 Beyond image coding .................................. 30
      2.4.5 Compositional hierarchical models .................... 31
  2.5 A general framework combining linearized system identification and statistical feature learning ........ 32
3 Modeling hierarchical image representation

3.1 Introduction ................................................................. 36
3.2 Unsupervised feature learning with restricted Boltzmann machines ........ 37
  3.2.1 Boltzmann machines .................................................. 37
  3.2.2 Training Boltzmann machines ....................................... 40
  3.2.3 Restricted Boltzmann machines ..................................... 41
  3.2.4 Gibbs sampling for RBMs ............................................ 42
  3.2.5 Contrastive divergence learning .................................... 42
  3.2.6 Stochastic gradient descent optimization .......................... 42
  3.2.7 Restricted Boltzmann machines for continuous data .......... 43
  3.2.8 Targeted regularization for RBMs .................................. 44
     3.2.8.1 Inducing sparsely-distributed representations ............ 45
  3.2.9 Deep Belief Networks ............................................... 47
3.3 Experiments .................................................................. 47
  3.3.1 Deep belief network trained on natural image patches .......... 50
     3.3.1.1 Analysis of 1st layer features ............................... 50
     3.3.1.2 Analysis of 2nd layer features ............................... 52
     3.3.1.3 Analysis of 3rd layer features ............................... 57
     3.3.1.4 Properties of hierarchical representation .................. 59
  3.3.2 Modeling neural responses based on learned hierarchical features .... 61
     3.3.2.1 Encoding models based on 1st layer features .............. 66
     3.3.2.2 Interpreting 1st layer model weights ..................... 68
     3.3.2.3 Encoding models based on 2nd layer features .......... 70
     3.3.2.4 Interpreting 2nd layer encoding model weights .......... 70
     3.3.2.5 Encoding models based on 3rd layer features .......... 74
3.4 Discussion .................................................................. 74

4 Modeling abstract scene processing ...................................... 79

4.1 Introduction ................................................................. 80
4.2 Experiments ................................................................ 81
  4.2.1 Category feature learning with latent Dirichlet allocation .......... 81
  4.2.2 Estimating linearized voxel-wise encoding models ............... 84
  4.2.3 Categories learned from natural scenes explain selectivity in many
       anterior visual rois ...................................................... 88
  4.2.4 Decoding natural scene categories from evoked brain activity .... 91
  4.2.5 Predicting the objects that occur in decoded natural scenes .... 93
4.3 Discussion .................................................................. 94

5 Epilogue ................................................................. 96

5.1 Summary of contributions ............................................ 97
5.2 Open questions and future directions ............................... 97
5.3 Closing remarks .......................................................... 99
A RBM Training Details

A.1 Maximum likelihood estimation for restricted Boltzmann machines ........ 117
  A.1.1 General parameter estimation ............................................. 117
  A.1.2 Negative log likelihood gradients $\partial \ell(v; \theta)/\partial \theta$ for Gaussian-input RBM 120
A.2 Probability density functions for Gaussian-input RBMs ...................... 122
  A.2.1 Joint density function ...................................................... 122
  A.2.2 Marginal density functions ................................................. 122
  A.2.3 Conditional density functions .............................................. 124
  A.2.4 Derivation of $E_{p(h|v)}[v_i h_j] = p(h_j = 1|v) v_i$ ...................... 125
A.3 Target regularization for RBMs .................................................. 126
### List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>Receptive Field</td>
</tr>
<tr>
<td>SI</td>
<td>System Identification</td>
</tr>
<tr>
<td>EM</td>
<td>Encoding Model</td>
</tr>
<tr>
<td>PEM</td>
<td>Polynomial Encoding Model</td>
</tr>
<tr>
<td>LM</td>
<td>Linearized Model</td>
</tr>
<tr>
<td>LEM</td>
<td>Linearized Encoding Model</td>
</tr>
<tr>
<td>LDM</td>
<td>Linearized Decoding Model</td>
</tr>
<tr>
<td>STA</td>
<td>Spike-triggered Average</td>
</tr>
<tr>
<td>STC</td>
<td>Spike-triggered Covariance</td>
</tr>
<tr>
<td>2DFT</td>
<td>2-Dimensional Fourier Transform</td>
</tr>
<tr>
<td>ICA</td>
<td>Independent Components Analysis</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum Likelihood Estimation</td>
</tr>
<tr>
<td>PoE</td>
<td>Product of Experts</td>
</tr>
<tr>
<td>MRF</td>
<td>Markov Random Field</td>
</tr>
<tr>
<td>BM</td>
<td>Boltzmann Machine</td>
</tr>
<tr>
<td>RBM</td>
<td>Restricted Boltzmann Machine</td>
</tr>
<tr>
<td>MCMC</td>
<td>Markov Chain Monte Carlo</td>
</tr>
<tr>
<td>CD</td>
<td>Contrastive Divergence</td>
</tr>
<tr>
<td>BGD</td>
<td>Batch Gradient Descent</td>
</tr>
<tr>
<td>SGD</td>
<td>Stochastic Gradient Descent</td>
</tr>
<tr>
<td>TGD</td>
<td>Threshold Gradient Descent</td>
</tr>
<tr>
<td>RGC</td>
<td>Retinal Ganglion Cell</td>
</tr>
<tr>
<td>LGN</td>
<td>Lateral Geniculate Nucleus</td>
</tr>
<tr>
<td>V1-V4</td>
<td>Primary Visual Areas 1-4</td>
</tr>
<tr>
<td>MT</td>
<td>Medial Temporal Cortices</td>
</tr>
<tr>
<td>EBA</td>
<td>Extrastriate Body Area</td>
</tr>
<tr>
<td>FFA</td>
<td>Fusiform Face Area</td>
</tr>
<tr>
<td>OFA</td>
<td>Occipital Face Area</td>
</tr>
<tr>
<td>PPA</td>
<td>Parahippocampal Place Area</td>
</tr>
<tr>
<td>TOS</td>
<td>Transverse Occipital Sulcus</td>
</tr>
<tr>
<td>RSC</td>
<td>Retrosplenial Cortex</td>
</tr>
<tr>
<td>LOC</td>
<td>Lateral Occipital Complex</td>
</tr>
</tbody>
</table>
Notation

\( \begin{align*}
N & \quad \text{# of stimulus-response pairs} \\
M & \quad \text{# of stimulus dimensions} \\
D & \quad \text{# of data points in a modeled dataset} \\
V & \quad \text{# of visible units in an RBM} \\
H & \quad \text{# of hidden units in an RBM} \\
B & \quad \text{# of training points in a minibatch} \\
O_d & \quad \text{# of objects in scene } d \text{ for LDA model} \\
K & \quad \text{# of possible categories in LDA model} \\
W & \quad \text{# of objects in vocabulary for LDA model} \\
\hat{r}(s) & \quad \text{the predicted neural response to stimulus } s \\
\Phi_s(s) & \quad \text{a transformation of stimulus } s \text{ into a feature space} \\
\beta_s & \quad \text{a weight vector for encoding model of class} \\
B_s & \quad \text{a weight matrix for encoding model of class} \\
x & \in \mathbb{R}^{V \times 1} \quad \text{a vector of observed random variable states} \\
X & \in \mathbb{R}^{V \times D} \quad \text{a matrix of observed random variable states} \\
v & \in \mathbb{R}^{V \times 1} \quad \text{a vector of visible unit states} \\
h & \in \mathbb{R}^{H \times 1} \quad \text{a vector of hidden units states} \\
W & \in \mathbb{R}^{V \times H} \quad \text{a matrix of connection weights between visible and hidden units} \\
A & \in \mathbb{R}^{H \times H} \quad \text{a matrix of intralayer connection weights between hidden units} \\
a & \in \mathbb{R}^{H \times 1} \quad \text{a vector of hidden unit biases} \\
B & \in \mathbb{R}^{V \times V} \quad \text{a matrix of intralayer connection weights between visible units} \\
b & \in \mathbb{R}^{H \times 1} \quad \text{a vector of visible unit biases} \\
E_s(z; \theta) & \quad \text{a scalar energy function for model class} \ast \text{ parameterized by } \theta \\
E_z[z] & \quad \text{the expected value of } z \text{ with respect to distribution} \ast \\
L_s(z; \theta) & \quad \text{the likelihood function of data } z \text{ for model class} \ast \text{ parameterized by } \theta \\
\ell_s(z; \theta) & \quad \text{the negative log likelihood function of } z \text{ for model class} \ast \text{ parameterized by } \theta \\
z^{[s]} & \quad \text{the state of a random variable } z \text{ at step } s \text{ in a Markov Chain} \\
Z_s(\theta) & \quad \text{the partition function for model class} \ast \text{ parameterized by } \theta \\
\psi(z) & \quad \text{target distribution calculated from random variable } z \\
\Psi(z) & \quad \text{a regularization function based on the values of } z \\
M_\tau[z] & \quad \text{threshold masking function applied to } z \text{ based on threshold } \tau
\end{align*} \)
parameter update for $\theta$ based on algorithm of class $\ast$

an optimal set of parameters

regularization constant

gradient descent step size

the point-wise sigmoid activation function applied to the values of $z$
Acknowledgments

This dissertation would not have been possible without the support of the Vision Science Group and funding from the NIH and the NSF Center for the Science of Information. There are also many individuals who were pivotal to bringing this work into fruition: My advisor, Jack Gallant, who has always challenged me to pursue novel and exciting research, who has painstakingly guided me on how to implement excellent science, and who has provided me with a stimulating academic environment. Interacting with the current and past member’s of Jack’s laboratory has been crucial for developing my skills and thinking as a scientist. In particular, I would like to acknowledge Thomas Naselaris, who guided me from my first rotation project to my first peer-reviewed publication; Michael Oliver who convinced me to join the lab and, in recent years, has been a go-to for stimulating discussion and crazy ideas; also, James Gao, who has selflessly sacrificed his time and effort to ensure that the lab avoids technological meltdown.

I would also like to thank those who have given me feedback on numerous manuscript drafts: Tom Griffiths, Shinji Nishimoto, Tolga Cukur, Mark Lescoarte, Michael Oliver, Alex Huth, James Gao, Natalia Bilenko, Anwar Nunez, Ben Dichter, and Melanie Miller.

I would like to thank the members of my dissertation committee: Tom Griffiths and Bruno Olshausen, whose scientific pedigree and acumen have pushed me to do the best work possible. I would also like to acknowledge the members of my qualifying exam committee: Lynn Robertson, Stanley Klein, Frederic Theunissen, and Michael Silver; their insights and suggestions have assisted me greatly during my development as a young researcher.

Additionally, I would like to thank my close friends, Jamie and Alyson Vitt, who relocated from across the country just to help keep me in touch with the real world, and who have always managed to accept my rants of scientific jibberspeak with sincere interest.

This dissertation would not have been possible without the support and patience of Melanie Miller. She has been my bedrock and her family has been a steadfast source of love and encouragement for me here on the west coast.

Most importantly, I would like to thank my family, particularly my grandmother Carolyn, brother Patrick, and sister Amanda. Though none of us were dealt the best hand, we’ve all managed to do pretty well for ourselves. These three are an inspiration to me and a constant reminder of what hope, hard work, and brazen stubbornness can achieve.
Chapter 1

Introduction
1.1 Motivation

Seeing the world, despite seeming automatic and effortless, is a very complicated process. Visual perception results from random light signals bouncing off of objects and surfaces in the world, passing through the imperfect optics of the eye, and being conveyed by noisy neural channels into the cortex. Even still, our visual system reconstructs the environment with a stability and coherency that allows us to easily operate in complex, dynamic world. We as neuroscientists would like to understand the computational and organizational principles that allow the populations of neurons that comprise the visual system to encode information about the visual world, a phenomenon referred to as neural representation.

A classic approach to investigating neural representation in the visual system is to study how electrophysiological activity evoked in neurons changes when presented with various visual stimuli. In this approach, stimuli are selected to vary along some specific dimension (e.g. luminance, contrast, etc) so that straight-forward conclusions can be made about the neural representation of that dimension. This approach has been highly informative for characterizing representation in the early stages of the visual system, leading to standard computational models that objectively account for neural responses observed in experiments (Carandini et al. 2005).

However, this classic approach breaks down when our intuitions as to which stimulus dimensions are important for representation fail to be informative. This has been the case for experiments aimed at characterizing representation in intermediate and late stages of the visual system. In these areas, the represented stimulus dimensions are mostly unknown, leading experimenters to construct peculiar stimuli that test subjective, indiosyncratic hypotheses. This approach has been widely adopted in the neuroimaging community, where researchers have tried to identify the anatomical locus of the representation for various abstract visual features (Spiridon et al. 2006).

A key test for any model of neural representation is the ability to accurately predict neural function under natural operating conditions. The interpretations of classic studies, however, generally make inaccurate predictions of brain responses to stimuli that lie outside of the specific set tested. This makes directly and objectively comparing hypotheses tested by multiple experiments difficult. What is needed in order to effectively characterize representation in intermediate and later stages of visual processing is a framework a) that can inform new hypotheses without relying on the intuitions of the experimenter, b) can generate tractable and interpretable predictive models of neural function based on these hypotheses, and c) provides a protocol for assessing model accuracy at predicting neural function under natural conditions.

An alternative to the classic approach for studying neural representation is to make few assumptions regarding relevant dimensions. The experimenter thus presents a large array of randomly-sampled stimuli rather than stimuli that vary along a specific dimension. Then, by noting the stimuli that evoke brain responses, the stimulus dimensions that are important for representation can be reverse-engineered. This approach, referred to as
system identification (SI), reformulates neural characterization as a regression problem. The SI approach offers a number of benefits over classic methods. In particular, the regression-based focus is objective and provides a means to directly test and compare the predictions of multiple hypotheses on general stimuli.

A third approach to studying neural representation is to develop theoretical models that are rooted in a general computational principle. Theoretical models are important because they allow objective interpretation of previous experiments in terms of general functional properties that are made explicit in the model formulation. More importantly, theoretical models can also objectively identify stimulus dimensions that may be important for neural representation, thus guiding future experiments. A general computational principle that has provided a solid foundation for developing accurate theoretical models of neural representation is the concept of efficient coding. Efficient coding suggests that representation in a neural system should be matched, either through evolution or development, to statistical structure of the natural environment. It follows that characterizing the statistical structure of natural scenes should inform us about the coding principles underlying neural representation. Statistical models based on efficient coding have provided convincing characterizations of early visual processing in the retina and thalamus, as well as in the first stages of cortical processing. However, until recently, there has been little work applying efficient coding to develop models of representation in intermediate and late cortical stages of the visual pathway. This is likely because standard statistical methods fail to capture the high-order properties of natural scenes. However, recent developments in statistical feature learning provide powerful frameworks for modeling high-order structure of natural scenes. This opens up a promising avenue for generating objective hypothesis regarding neural function at intermediate and late stages of visual processing.

The goal of this dissertation is to combine system identification with statistical feature learning to investigate neural representation at various stages of the cortical visual hierarchy.

1.2 Organization

In Chapter 2, I frame the general problem of modeling neural representation in the visual system and propose a novel framework for doing so. I begin with a brief review of classic accounts of neural function along the visual pathway and highlight some of the standard computational models developed in light of these data. I then move on to describe system identification methods used in sensory neuroscience, focusing a specific class of methods known as linearized models. I follow with a description of natural scene structure and explain how theoretical models based on this structure provide accurate characterizations of neurons in the early stages of the visual system. Finally, I propose a general framework that combines linearized modeling with statistical feature learning to develop, test, and compare competing models of neural representation in the visual
Chapter 3 introduces a general statistical learning module that, through a simple and general regularization scheme, can be trained to learn a sparse and selective representation of natural images. I then show how the same learning module can easily be stacked to compose a hierarchical model of natural images. I then train a 3-layered model in this manner to learn a hierarchical feature representation of natural images. The features learned at each layer of the model are able to account for a various properties previously observed in visual areas V1 and V2. I then use the features learned at each layer of the hierarchical model to construct predictive models of neural responses at multiple stages of the primate visual system, namely, areas V1 and V2. Interestingly, the feature representations learned in higher layers of the network provide better characterizations V2 neurons than the lower layers. This suggests that a feedforward, hierarchical representation of natural images provides a valid account for the information encoded in V2 neurons.

In Chapter 4, I use statistical feature learning to develop a model of abstract scene representation in later stages of the human visual system. Specifically, I use a topic model developed to analyze large corpora of text to learn feature representations from lists of objects that occur in natural scenes. The resulting features are intuitively interpreted as discrete scene categories. I then build predictive models of brain activity based on the learned categorical features. These models provide an accurate predictions of fMRI responses evoked in late stages of the human visual hierarchy during natural vision. I also show that scene category identity, as well as information about individual objects can be accurately inferred from brain activity evoked when viewing previously unseen images.

1.3 Published appearances of dissertation work

The work in Chapter 4 originally appeared in the publication:

Chapter 2

Background & Approach
An overarching goal of visual neuroscience is to understand how the visual system processes natural scenes. Natural scenes possess non-trivial statistical structure and theoretical models based on this structure have provided numerous insights into the processing mechanisms implemented in the early visual system (Barlow 1981; Field 1987; Adelson and Bergen 1985). Despite the success of these models in the early visual system, there have been few attempts to use models based on natural scene statistics to characterize neural function at later stages in the visual hierarchy. In this chapter I motivate and propose a simple but powerful framework for developing, testing, and comparing models of neural representation that are based on natural scene statistics.

2.1 The primate visual hierarchy

Many of the functional properties of neurons in the visual system have been identified using receptive field mapping experiments. In these experiments various stimuli are presented to the visual system and evoked neural responses are measured. Stimuli include points of light (Enroth-Cugell and Robson 1966), luminance-defined bars (Hubel and Wiesel 1959), two-dimensional (2D) sinusoidal gratings (Movshon et al. 1978ab), or even whole objects (Tanaka 1996; Kanwisher et al. 1997; Epstein and Kanwisher 1998). The aim is to identify the visual features associated with these stimuli that best drive a neuron. The identified features are referred to as the neural receptive field.\(^1\) The results of these classic experiments suggest that the brain processes visual information through a stage-wise hierarchy (Felleman and Van Essen 1991). Progression along the hierarchy is accompanied by increasingly nonlinear stimulus-response functions (Carandini et al. 2005) and neural responses becomes more and more modulated by top down mechanisms such as attention and memory (Reynolds and Chelazzi 2004). Here I give an overview of the basic anatomy and functional properties of the visual hierarchy, as well some of the standard computational models that have been developed to explain visual representation at various stages. Note that I focus on feedforward processing mechanisms. These mechanisms pass information from the peripheral sensory areas toward central processing stages. It is also important to note that feed-forward connections are accompanied by numerous feedback and intra-area connections (Felleman and Van Essen 1991). These mechanisms are beyond the scope of this work, however.

2.1.1 The retina and LGN

The first stage of visual processing occurs is the retina, where light is sensed by nearly 125 million photoreceptor cells that line the back of the eye. This information is compiled

\(^1\)Classically, the visual receptive field was designated to spatially-localized features, but I extend the definition to any visual feature that may drive a neuron.
by 1.5 million retinal ganglion cells (RGC) and is passed to the lateral geniculate nucleus (LGN) of the thalamus via the optic nerve. Receptive field mapping studies based on stimulating the retina with localized points of light suggest that simple features, generally described as either luminance blobs or antagonistic center-surround patterns of luminance or color, are best at driving RGC and LGN neurons (Enroth-Cugell and Robson 1966; Carandini et al. 2005). The responses of these neurons are also approximately a linear function of the amplitude of the preferred feature (DeValois and DeValois 1990; Carandini et al. 2005). Standard computational models of RGC and LGN neural responses are based on filtering the stimulus image with a kernel function such as a 2D Gaussian (to approximate luminance blobs) or difference of two Gaussians (to approximate center-surround features), followed by a rectifying nonlinearity (to account for the fact that neurons do not fire negative spikes; Enroth-Cugell and Robson 1966). These models are quite accurate at explaining some of the simpler response mechanisms in the retina and LGN (Carandini et al. 2005) and have biological interpretability in terms of local neural interactions shown to occur in the retina (Srinivasan et al. 1982; Atick and Redlich 1992; van Hateren 1992).

2.1.2 Primary visual cortex

The LGN relays information to the visual cortex, which is comprised of a number of anatomically defined areas located near the back of the brain (Figure 2.1.1). The first cortical processing stage is the primary visual cortex (V1, or striate cortex) which is located
on the posterior pole of the brain. Area V1 exhibits an overcomplete neural representation of the signals received from the LGN: for every afferent connection arriving from the LGN, there are approximately 100 V1 neurons (Barlow 1981). Area V1 receptive fields were originally identified by presenting luminance-defined bars of various orientations (Hubel and Wiesel 1959; Hubel and Wiesel 1962). The spatial extent of the receptive field for V1 neurons is larger than that of LGN neurons (representing 0.3-5 degrees of visual angle; (Gattass et al. 1981)). Area V1 receptive fields capture simple visual features such as oriented edges. It was later found that the stimuli that optimally drive V1 cells are well-approximated by a Gabor function—a sinusoidal grating modulated by a Gaussian envelope (Daugman 1985; Jones and Palmer 1987; Ringach 2002; Figure 2.1.2). Neurons in V1 also exhibit velocity selectivity in the presence of time-varying stimuli (Orban et al. 1986).

Figure 2.1.2: **Gabor model fits to V1 receptive fields.** **Top row,** Each subpanel is a receptive fields for V1 simple cell, estimated using reverse correlation. **Middle row,** Each subpanel displays the Gabor functions fit to the receptive field above. **Bottom row,** the error of the fit. The Gabor function provides a good fit to V1 receptive fields. Adapted from (Ringach 2002)

Neurons in V1 are often grouped into two classes: simple and complex cells (Hubel and Wiesel 1968). Simple cells are linearly driven by edges and bars that have a specific color, orientation, width, and location. Thus the standard computational model for simple cell responses involves filtering a stimulus image with an 2D, oriented, and frequency bandpass filter (i.e. Gabor function), followed by a rectifying nonlinearity (Movshon et al. 1978b). Complex cells are generally invariant to the luminance polarity, as well as to small variations in the bar’s location (Hubel and Wiesel 1968; Movshon et al. 1978a; Heeger et al. 1996). Thus when complex cells are probed with sinusoidal grating stimuli, their responses are generally invariant to the phase of the contrast modulation (Movshon et al. 1978a). The standard computational model for complex cells—known as the “energy
model”—predicts responses by filtering a stimulus image with two Gabor functions of the same orientation and spatial frequency but different phases, then squaring the output of each filter response and taking their sum (Adelson and Bergen 1985; Heeger 1992ab; Heeger et al. 1996).

Though the standard models for V1 simple and complex cells can accurately predict basic responses to simple, parametric stimuli, they do not fully account for functional properties of these neurons. Even simple cells exhibit many nonlinear phenomena. For example, the response of a V1 cell is often affected by the stimulus structure just outside of the neuron’s receptive field (Cavanaugh et al. 2002; Knierim and van Essen 1992; Vinje and Gallant 2000). Thus, natural scenes drive cells differently than parametric stimuli (David et al. 2004). Additional mechanisms such as divisive normalization have been added to the standard models to account for these phenomena (Carandini et al. 1997; Heeger 1992b), but there are still a number of unaccounted-for phenomena that exist (Carandini et al. 2005). The simple-complex cell dichotomy is also likely flawed. Most complex cells are not completely phase invariant and thus the distinction between simple and complex cells is likely an oversimplification that reflects two extrema in a spectrum between linear nonlinear stimulus-response properties (Ringach 2002; Hegdé and Van Essen 2007).

2.1.3 The ventral and dorsal streams

Area V1 passes a majority of its feedforward connections to visual area V2. Area V2 neurons share many functional properties with V1 neurons, including selectivity to location, orientation, color, and spatial frequency (Gattass et al. 1981; Hegde and Van Essen 2003). However, V2 neurons have larger spatial receptive fields than V1 (Gattass et al. 1981) and can be selective to multiple orientations (Lennie 1998). V2 neurons are selective for more complex visual features such as curves (Hegde and Van Essen 2003), angles and junctions (Ito and Komatsu 2004), and have complex suppressive subfields (Willmore et al. 2010). V2 neurons have also been implicated in coding higher-order features (Baker and Mareschal 2001), such as edges defined by texture (von der Heydt and Peterhans 1989) and illusory contours (von der Heydt et al. 1984). Area V2 is also slightly modulated by attention (Luck et al. 1997; Reynolds et al. 1999), indicating a larger role in top-down processing than area V1.

A large amount of the information processed in V2 is passed to visual area V4. Area V4, represents yet more complex visual features. Like neurons in V1 and V2, neurons in V4 are sensitive to orientation, spatial frequency, and color information (Felleman and Van Essen 1991; Lennie 1998; Hegde and Van Essen 2007). Additionally, V4 neurons have even larger spatial receptive fields than V2 (Gattass et al. 1988), are selective for features of intermediate complexity (Lennie 1998), and are driven by non-Cartesian gratings (Gallant et al. 1993) and curved shapes (Pasupathy and Connor 2001). Area V4 is also moderately modulated by top-down mechanisms such as attention, learning, and memory (Luck et al. 1997; Reynolds and Chelazzi 2004; David et al. 2008).
Area V4 passes information onward to the inferior temporal lobe (IT) which is populated with numerous anatomical subregions that represent abstract visual features such as the presence of specific types of objects. For example, the extrastriate body area (EBA) (Downing et al. 2001), located anteriorly to V4 in occipito-temporal cortex, is selective for the presence of human and animal body parts. The fusiform face area (FFA; Kanwisher et al. 1997), located along the fusiform gyrus on the inferior portion of temporal lobe, represents the presence of complete faces. Neurons in these areas generally have large, bilateral, spatial receptive fields that cover a majority of the visual field (Tanaka 1996). They also exhibit a large degree of invariance to mean luminance, location, pose, and the spatial scale of their preferred stimulus (Tanaka 1996).

The pathway that includes V2, V4, and IT make up what is referred to as the “ventral stream” (Figure 2.1.1) of the visual system (Mishkin and Ungerleider 1982). The ventral stream is believed to be essential for visual object recognition. It is involved in processing form and color, as well as performing object segmentation and identification. There are currently no standard computational models for the regions that comprise the ventral stream. The classic account is that visual features of increasing complexity are represented by nonlinearly pooling the activity of neurons from preceding stages in the visual hierarchy (Felleman and Van Essen 1991). Thus V2 pools local edge information from V1 to represent corners and junctions while V4 pools information from V2 and V1 to represent yet more complex visual features. From this cascade of nonlinear pooling operations emerges increasing receptive field size and invariance properties with each stage. This account seems reasonable, but there have been few computational models based on this principle that explicitly aim to predict neural responses (Hegdé and Van Essen 2007; Tanaka 1996).

A complementary visual pathway to the ventral stream, referred to as the “dorsal stream,” is organized along the dorsal and medial aspects of the visual system, terminating in the parietal lobe (Figure 2.1.1). The dorsal stream is necessary for processing spatial relationships and motion information. This is supported by lesion studies that demonstrate damage to the areas in the dorsal stream create deficits in spatial and motion processing (Mishkin and Ungerleider 1982). Within the dorsal stream, visual areas V1 and V2 project information to area V3, which is located just anterior to V2. Information is then passed from V3 to medial temporal cortex (MT/V5). Areas V3 and MT are selective for velocity of movement in the visual field (Albright 1992; O’Keefe and Movshon 1998). The standard computational model of neural responses in MT is based on decomposing a time-varying stimulus with oriented and bandpass space-time filters (space-time Gabor functions), followed by rectification and contrast normalization nonlinearities (Simoncelli and Heeger 1998; Nishimoto and Gallant 2011).

Late stages of the dorsal stream include the transverse occipital sulcus (TOS) (Nakamura et al. 2000; Grill-Spector 2003; Hasson et al. 2003), the parahippocampal place area (PPA; Epstein and Kanwisher 1998), and the retrosplenial cortex (RSC; Maguire 2001). These areas have been identified by the presence of increased activations when stimulated with images that contain buildings and landscapes. It is thus believed that these areas
are involved in representing information vital for identifying environments and performing spatial navigation. However, as in the intermediate- and late stages of the ventral stream, there are still no principled, predictive models of brain activity for the late stages of the dorsal stream.

2.2 Modeling neural representation in the visual pathway

The characterization of the visual system outlined above is rooted in experiments designed to probe how neurons represent specific dimensions of the visual input. This approach has proven effective for characterizing neurons in the early stages of the visual system where few transformations of the sensory input have occurred. However, this approach becomes less informative for characterizing neurons at later stages of visual processing. Another approach is to make few assumptions regarding neural representation and present a large array of randomly-sampled stimuli. The aim is to randomly probe the important, but unknown dimensions of the visual input. By noting the correspondence between stimulus and response, the experimenter can reverse-engineer the visual features represented by neuron. This approach, referred to as system identification (SI) reformulates neural characterization as a regression problem and offers a number of benefits over classic methods, which I develop in this section.

2.2.1 System identification

System identification is a general set of techniques used to characterize input-output systems (Marmarelis 2004; Wu et al. 2006). Under the system identification approach a (generally nonlinear) parametric function is used to describe the relationship between stimuli and evoked brain responses (Theunissen et al. 2001; David and Gallant 2005; Wu et al. 2006). I will refer to the this function as an encoding model (EM) because it describes how stimuli are encoded in evoked brain activity. The EM parameter values are fit by observing $N$ stimulus-response pairs; $\{s_n, r_n\}$ where $n = 1, 2, \ldots, N$, $s_n \in \mathbb{R}^{M \times 1}$ is the $n$-th stimulus represented in vector notation, and $M$ is the number of stimulus dimensions. The model parameters are adjusted to minimize some function that defines the error between the measured brain responses $\hat{r}$ and the responses predicted by the the model $\hat{f}$. Once the EM parameters are fit, the model is evaluated in terms of its ability to predict responses evoked by novel stimuli (i.e. those not used in parameter fitting).

The EM can take many functional forms (Marmarelis 2004; Wu et al. 2006). The most common class of encoding models used in sensory and motor neuroscience is the Polynomial Encoding Model (PEM). The functional form of the PEM is a linear combination polynomial functionals applied to the stimulus dimensions. For example, a PEM of order 2 generates predicted responses to a set of stimuli with the following stimulus-response function:
\[ \hat{r}_{PEM}(s) = \beta_0 + s^T \beta_1 + s^T B_2 s. \] (2.2.1)

The first term in Equation 2.2.1 is a scalar offset that can generally be ignored with proper scaling of the stimulus and responses. The second term in Equation 2.2.1 captures linear relationships between the stimulus dimensions and the responses (the parameter \( \beta_1 \) is a weight vector). This term is often referred to as the spike-triggered average (STA) in the neuroscience literature (de Boer and Kuyper 1968; Marmarelis and Naka 1972; Wu et al. 2006). The third term in Equation 2.2.1 captures pairwise interactions between the stimulus dimensions and the responses (the parameter \( B_2 \) is a weight matrix). This term is often referred to as the spike-triggered covariance (STC) (Brenner et al. 2000; Touryan et al. 2005).

The PEM is a general function estimator that makes few assumptions about the shape of the stimulus-response function. By adding higher-order terms a PEM can, in principle, capture more complicated, nonlinear relationships. Many studies have used the STA to describe the responses of simple cells in primary visual cortex (e.g. Jones and Palmer 1987; Alonso 2001; Touryan et al. 2002; Rust et al. 2005). The STC has been used to characterize the properties of complex cells in primary visual cortex (Brenner et al. 2000; Rust et al. 2005; Touryan et al. 2002 2005).

Though PEMs provide good models of low (i.e. 1st and 2nd) order stimulus-response functions, they are limited in their usefulness as general nonlinear models of visual processing. One limitation is that PEMs generally assume that the stimuli have spherical, Gaussian statistics (Chichilnisky 2001; Paninski 2003). (Though there are methods for transforming natural scenes to have spherical statistics, these methods often greatly amplify stimulus noise and must be regularized for effective application (Touryan et al. 2005).) This makes the PEM unsuitable for modeling how the brain processes natural scenes, which have non-Gaussian statistics (Field 1987). This is particularly problematic for characterizing neural representation in intermediate and later visual areas, which are driven poorly by Gaussian noise stimuli, but quite effectively by natural scenes. Another, perhaps greater limitation, is that PEMs do not scale well, requiring exponentially more data to fit each higher-order term (Marmarelis 2004). Thus, it is generally not feasible to fit a PEM beyond second order (but see Oliver et al. 2012). This makes characterizing nonlinear stimulus-response functions problematic for PEMs and for modeling higher-order visual processing, which involves highly nonlinear stimulus-response functions.

### 2.2.2 Linearized models

To overcome some of the limitations of PEMs, linearized models (LM) can be used to capture higher-order nonlinearities (Wu et al. 2006). Linearized modeling breaks the system identification problem into two steps. First, stimuli are transformed nonlinearly into an intermediate feature space \( \Phi(s) \). The feature space is chosen so that the relationship between the features and evoked brain response are as linear as possible. The feature
space can thus be interpreted as an instantiation of an explicit hypothesis regarding the information encoded in evoked brain activity (Wu et al. 2006; Naselaris et al. 2011). In the second step, linear regression is used to fit a set of weights $B_E$ that best map the intermediate features onto the measured brain activity (Figure 2.2.1). The resulting Linearized Encoding Model (LEM) describes how the intermediate features are explicitly (i.e. linearly) encoded in evoked brain activity (Naselaris et al. 2011). A fit LEM generates predicted responses to a set of stimuli with the following stimulus-response function:

$$\hat{r}_{LEM}(s) = \Phi(s)^T B_E.$$  \hspace{1cm} (2.2.2)

Though the LEM is less general than the PEM, it offers several important benefits. First, LEM parameters are fit by linear regression, a simple, well-established technique whose theoretical properties are firmly understood. For example, a variety of regularization techniques have been developed for linear regression (Bishop 2006; Friedman et al. 2010). Regularization is important for several reasons: it allows the experimenter to add prior knowledge to the parameter fitting procedure, it reduces the tendency to overfit the model to noise, and it ensures stable parameter estimates for ill-conditioned problems (i.e. when the number of stimulus-response pairs is fewer than the number of model parameters). Thus, a LEM can generally be fit with fewer data than are required to fit a PEM.

Figure 2.2.1: Diagram of Linearized Models. In linearized modeling, the stimulus (left) is first nonlinearly mapped into a feature representation. This feature representation is then mapped onto measured responses by a linear encoding model. In principle the measured responses can come from any modality, including neural spike rates, fMRI BOLD measurements, or behavioral reaction times (right, red). Linearized decoders can also be constructed to map measured responses onto the feature representations.
The LEM facilitates model interpretation as well. It is often difficult to interpret or visualize PEMs (Wu et al. 2006), but the linear weights $B_E$ can be viewed directly. One can interpret the weights of an encoding model as estimates of the tuning curve for the set of features under investigation. Tuning curves are a standard method used to interpret neural representation in classic neurophysiology. However the tuning curves can also offer important interpretations for functional magnetic resonance imaging (fMRI) studies (Huth et al. 2012; Stansbury et al. 2013).

Linearized encoding models also do not require stimuli to have spherical Gaussian statistics, so stimuli can be sampled from natural scenes. Using natural scene stimuli is important for a number of reasons. First, the visual system exhibits an array of nonlinear response properties that are absent in the presence of synthetic or noise stimuli (Dan et al. 1996; David et al. 2004). Second, natural scenes contain stimulus features that range from low-level pixel structure to high-level semantic concepts. Consequently, natural scenes elicit activity from neurons that represent stimuli at various levels of complexity. If it is possible to record multiple regions of the brain simultaneously (as is the case in fMRI experiments), then natural scene stimuli offer the opportunity to investigate multiple stages of visual processing with a single experiment.

Finally, LEMs can be used to directly compare multiple competing hypotheses on the same data. For example, a standard hypothesis is that neurons (or voxels) in primary visual cortex (area V1) represent spatially localized image structure, such as orientation and spatial frequency (Hubel and Wiesel 1968; Movshon et al. 1978ab; Adelson and Bergen 1985; Jones and Palmer 1987; Carandini et al. 2005). This hypothesis can be tested with an LEM by transforming stimuli into a feature space that represents local image structure. Such a transformation is provided by a Gabor wavelet decomposition (Daugman 1985). The class of LEMs based on Gabor wavelet decomposition provide remarkably accurate predictions of brain activity measured in single neurons (Willmore et al. 2008) and in single voxels (Kay et al. 2008; Nishimoto et al. 2011) in area V1.

A large body of recent work has demonstrated the effectiveness of using LEMs for studying natural scene processing. Neurophysiology studies have used the 2D Fourier power spectrum of images as a feature space to fit LEMs for single neurons in V1 (David et al. 2004) and area V4 (David et al. 2006). The Fourier power spectrum is a spatially global representation of the stimulus. Thus, this class of LEM captures position-invariant properties of these neurons. A neurophysiology study of area MT (Nishimoto and Gallant 2011) used a feature space consisting of a pyramid of motion-energy filters (i.e., spatio-temporal Gabor wavelets). The motion-energy pyramid describes how spatially-localized orientation, spatial frequency, temporal frequency, and direction information represented in area V1 are pooled and encoded in the activity of single MT neurons. This class of LEM accurately predicts responses of individual MT neurons to arbitrary stimuli, including natural movies.

Functional MRI studies have also used feature spaces provided by the Gabor wavelet pyramid (Kay et al. 2008) and motion-energy pyramid (Nishimoto et al. 2011) to fit LEMs to single voxels throughout early visual cortex. These LEMs provide can account of a
wide range of phenomena that have been reported in previous fMRI studies of early visual cortex. For example, Kay et al. 2008 and Nishimoto et al. 2011 showed that retinotopic maps (DeYoe et al. 1996) in the early visual system can be recovered by analyzing the weights of Gabor-based LEMs fit to fMRI responses evoked during natural vision.

Higher visual areas such as the FFA and PPA appear to represent more abstract, semantic content of natural scenes. Thus, voxels in these higher visual are not well characterized by models based on simple image features. An alternative approach is to model these areas in terms of semantic features. For example one option is to represent stimuli in terms of the objects (nouns) and actions (verbs) that appear in a scene (Cukur et al. 2013; Huth et al. 2012). These LEMs reveal that semantic information in natural scenes is widely distributed in broad gradients that extend across higher visual cortex. Results from these studies suggest that classically-defined regions of interest (ROIs) such as the FFA and PPA are nodal points within these gradients.

Yet another option is to represent stimuli in terms of abstract scene categories (Naselaris et al. 2009; Naselaris et al. 2012; Stansbury et al. 2013). Linearized models based on scene categories provide an accurate account of brain activity measured throughout anterior visual cortex. Results from these studies reveal that the correspondence between selectivity for objects, actions, and scene categories in the human brain likely reflect statistical relationships between objects, actions, and categories that exist in the natural world (Mitchell et al. 2008; Naselaris et al. 2012; Stansbury et al. 2013)

A complementary approach to the LEM is the linearized decoding model (LDM). As in the LEM approach, the LDM is constructed by mapping stimuli into an intermediate feature space. However, linear regression is performed in the opposite direction, fitting a set of weights $B_D$ that optimally map brain activity onto the intermediate feature representation. Thus, an LDM describes how stimulus features are recovered from brain activity (Naselaris et al. 2011). Given a set of responses, an LDM generates a predicted set of stimulus features $\hat{\Phi}_{LDM}(s; r)$ using the following response-stimulus function:

$$\hat{\Phi}_{LDM}(s; r) = r^T B_D.$$

It is also possible to transform a fit LEM into a LDM using a probabilistic formulation based on Bayes’ theorem (Thirion et al. 2006; Naselaris et al. 2009; Nishimoto et al. 2011). Specifically, one calculates the posterior distribution of the stimulus given the observed responses:

$$p(s|r) \propto p(r|s)p(s).$$

Here, the likelihood function of the responses $p(r|s)$ will depend on the feature space transformation and the assumed noise model (Wu et al. 2006). For example, assuming that response noise is Gaussian-distributed gives the following linearized likelihood $p(r|s) \sim N(\Phi(s)^T B_E, \Sigma_r)$. The noise covariance $\Sigma_r$ can be estimated from the residuals
of the encoding model. The prior over stimuli \( p(s) \) can be estimated empirically (Naselaris et al. 2009; Nishimoto et al. 2011), or approximated parametrically (Naselaris et al. 2009 2011). Our lab has shown that LDMs are able to accurately decode the structural content of natural images (Kay et al. 2008) and movies (Nishimoto et al. 2011), as well as the semantic content of natural scenes (Naselaris et al. 2009; Naselaris et al. 2012; Stansbury et al. 2013). Though decoding models provide no additional information about the stimulus-response relationship than encoding models, they are effective tools for characterizing the information content of brain activity. By diagnosing the dimensions that can be decoded accurately, the experimenter can determine the features in the linearized space that are represented in measured brain responses.

Applying linearized models requires a linearizing transformation that maps stimuli into an intermediate feature space (Naselaris et al. 2011). Thus, a primary goal when using linearized models is to develop new feature spaces (and the associated linearizing transformations) that capture novel hypotheses about neural representation. To date, most linearized models have used feature spaces developed by hand. Some of these feature spaces reflect documented filter-like computations (Daugman 1985; Jones and Palmer 1987; Ringach 2002) associated with the neural population under investigation (Kay et al. 2008; Nishimoto et al. 2011). Others capture intuitive linguistic or cognitive labels (Mitchell et al. 2008; Naselaris et al. 2009; Huth et al. 2012). However, these approaches suffer from the same drawback of classic studies that select stimuli that vary along a specific dimension in order to test a specific hypothesis: It is often unclear, especially in intermediate and late visual areas, what the functional form the feature space should take. It would be preferable to develop new feature spaces in an objective, principled manner. I propose that theoretical models based on natural scene structure can provide a principled means of objectively generating novel feature spaces, and thus new hypotheses regarding neural representation. In the next section, I support this proposition by reviewing some of the notable structural properties of natural scenes and showing that theoretical models based on these properties provide accurate characterizations of neural representation in the visual system.

2.3 Statistical structure of natural scenes

In order for an organism to survive, their brain must possess neural processes that extract relevant information from the environment. For example, it is likely that the brain will need to represent low-level image structure such as luminance edges in order to distinguish individual objects and surfaces. Indeed, neuroscience has found substantial evidence for the representations of luminance edges by neurons in the early visual system (Hubel and Wiesel 1959; Movshon et al. 1978ab). However, the brain must represent many more structural aspects of natural scenes, including image structure of intermediate and high level of complexity, as well as abstract and semantic structure. Thus, having a solid understanding of the structural properties of the natural world can provide insight into
neural representation.

A majority of the notable work on characterizing the structure of natural scenes has focused on building mathematical models based on images of natural scenes. This approach is motivated by the fact that the brain must learn the structure of the world form the sensory input it receives, i.e. the image projected on the retina. Additionally, the sources and events that create natural images (e.g. lighting, texture, motion, occlusion) are not deterministic, but probabilistic. Thus these mathematical models are generally formulated to capture the statistical structure of natural scenes. Here I review some of the previous work aimed at measuring and modeling the basic statistical properties of natural images.

2.3.1 Local pixel statistics

The simplest properties of natural images are at the scale of a single pixel. Such properties are luminance, local contrast, and color. It has been shown (Laughlin 1981; Brady and Field 2000) that pixel intensities in natural scenes vary by many orders of magnitude, but that the distribution of luminance values is positively skewed (approximately Gaussian-distributed on a log scale). This indicates that natural natural scenes are dominated by dark pixels with infrequent, but large luminance impulses.

Local contrast measures the luminance of a pixel relative to the luminance of surrounding pixels and is an indicator of local distinguishability. Laughlin 1981 and Brady and Field 2000 have shown that the distribution of contrasts in natural scenes is also positively skewed. These findings suggest that natural scenes are dominated by smooth, continuous surfaces with the occasional presence of large luminance discontinuities. These discontinuities likely result from object edges and shadows. Although both luminance and contrast are biased toward lower values in natural scenes, surprisingly, they are only very weakly correlated (Mante et al. 2005).

In addition to luminance and contrast, each pixel carries information about the wavelength of light at a location in the image. The primate retina represents wavelength information using three types of photoreceptors, each of which is sensitive to a specific range of wavelengths. Thus an informative way of analyzing the wavelength content of images is in terms of these sensitivities. (Ruderman et al. 1998) showed that the wavelengths of light in natural scenes vary primarily along 3 dimensions within a subspace spanned by (the log of) these cone sensitivities. Perceptually, these 3 dimensions correspond to the dimensions of red versus green, blue versus yellow, and high versus low luminance intensity.

2.3.2 Pairwise pixel statistics

Though natural scenes exhibit statistical regularities at the scale of individual pixels, far more information is contained in the spatial patterns of pixels across the image. Attneave 1954 was the first to point out that natural scenes exhibit spatial correlations
due to the presence of continuous surfaces and objects in the world. An example of such spatial correlations is displayed in Figure 2.3.1. The luminance values of two adjacent pixels display a high degree of correlation, whereas the luminances of distant pixels are essentially independent.

A common way of measuring the spatial correlation in an image is to calculate the spatial autocorrelation function, or analogously, the power spectrum of the image. The power spectrum of an image is obtained by squaring one of the two components of the 2D Fourier transform (2DFT), the magnitude spectrum (the remaining component of the 2DFT is the phase spectrum). The power spectrum captures the correlation information contained in an image at various spatial frequencies and orientations (Figure 2.3.2, middle row).

Figure 2.3.1: Pairwise distribution of pixel values in natural images for nearby and distant pixels. The dark subpanels display the location of two pixels, indicated in cyan and magenta. Each point in the main plots is the joint activation for the cyan (horizontal axis) and magenta (vertical axis) pixel for a single natural image. The marginal distributions for each pixel are indicated by the corresponding color on the boundary of the main plots. The red line indicates perfect correlation. A. Nearby pixels are much more correlated than distant pixels (B), suggesting that local smoothness is persistent in natural scenes.

(Field 1987) studied how the power spectrum of natural images varies with spatial frequency content (by averaging the power spectrum across all orientations; Figure 2.3.2). He found that (on a log scale) power decreases approximately linearly with the squared frequency. One explanation for the observed power spectrum is that natural scenes have frequency content that is invariant to scale changes, such as those created by viewing the same scene at different distances (Ruderman 1997).

The other component of the power spectrum of natural scenes—orientation—also exhibits consistencies. Coppola et al. 1998 demonstrated that natural scenes are dominated by orientations that lie along the cardinal (horizontal and vertical) axes (Figure 2.3.2).
Figure 2.3.2: Power spectra of three classes of images. The images sampled from natural scenes, $1/f$ noise, and white (Gaussian) noise, and are organized in the left, center, and right columns, respectively. **Top row**, grayscale sample of each class of image. **Middle row**, the power spectrum calculated for the sampled image above. **Bottom row**, the radial average about the center of the log power spectrum for each image (blue line), along with the (red) line indicating the $1/f^2$ trend. The $1/f$ noise image has a very similar power spectrum as the natural image, but does not resemble the structure of the natural scene. The Gaussian noise image has a flat power spectrum, indicating equal power at all frequencies.

This finding points out the dramatic effect that features such as the horizon, trees, and buildings have on the statistical properties of images.

Just as natural scenes have structure in their frequency and orientation information, Thomson 2001 also observed that there is considerable structure in the phase spectrum of natural scenes. Furthermore, he proposed that while the amplitude spectrum can explain correlations between one or two pixels only, higher-order spatial correlations that are required to describe extended features like edges and contours are contingent on the
phase spectra. Indeed when the phase information of a natural image has been removed, so too is the high-level structure in the image (Figure 2.3.3).

Figure 2.3.3: Demonstration of how the magnitude and phase spectra relate to the structure of natural images. A, Two natural scene images. B, The magnitude spectrum of each image. C, To demonstrate the role of the phase spectrum in the high-order structure in natural scenes, the phase spectra of the two images are interchanged while preserving the original magnitude spectra. D, The inverse Fourier transform of each image after switching their phase spectra. Phase carries much of the high-order structure in each image.

2.3.3 Higher-order spatial statistics

A common account of neural function in V1 is that it performs a filtering operation that signals the presence of spatially-correlated information such as an edge. Thus, an informative analysis for understanding how the brain forms visual representations of natural scenes is to analyze how such filters might behave in the presence of natural stimuli. Additionally, analyzing the structure in responses of oriented, bandpass filters (such as V1-like, Gabor filters) can directly highlight the higher-order statistical structure of natural scenes, as such filters already account for low-order structure (Daugman 1985).

Field 1987 and Daugman 1985 studied how V1-like cells in early visual cortex respond to natural scenes. They reported that such filters exhibit large but infrequent responses, resulting in a highly kurtotic distribution of response amplitudes (Figure 2.3.4). Field 1987 reasoned that the kurtotic distribution was due to the infrequent event that a local image feature overlapped with the spatial receptive field of the cell. He further proposed that having sparse response profiles are a highly informative strategy for encoding spatial
structure. An interesting finding by Huang and Mumford 1999 showed that any linear filter (with zero mean) exhibits a kurtotic distribution of responses in the presence of natural scenes. This suggests that all spatial features, not just oriented edges, exhibit a sparse distribution in natural scenes.

![Figure 2.3.4: Distribution of V1-like filter responses to a natural scene. The black line shows the distribution of a Gabor function activations in response to 100,000 image patches. The distribution is highly kurtotic. A heavy-tailed (Laplace) distribution (blue curve) provides a much better fit to the curve than a Gaussian distribution, which has no kurtosis (red curve). Here, Laplace and Gaussian distributions were fit via maximum likelihood estimation.](image)

Images are not represented by a single cell, but rather multiple cells. Thus, an another important analysis for understanding how neural representations may be affected by the structure of natural scenes is to investigate the dependencies of multiple V1-like cells during natural stimulation. Zetzsche et al. 1999 studied the joint distribution of responses exhibited by pairs of V1-like filters in the presence of natural scenes. They found that the joint distribution for pairs of filters with similar position, scale, and orientation exhibit a roughly circularly symmetric shape (Figure 2.3.5, B). A circular joint distribution indicates that the filters are uncorrelated, but not statistically independent. The joint distribution for independent variables should exhibit a “4-pointed star” shape (Figure 2.3.5, B).

Simoncelli and Buccigrossi 1997 further characterized the dependence of V1-like filters in the presence of natural scenes. They observed that the variance of the activation for one filter is dependent on the activation of a second non-overlapping filter. This effect is demonstrated by a “bow-tie” shape in the conditional distribution of the filter activations (Figure 2.3.6). Additional shapes of the conditional distribution have also been reported (including “diamonds” and “pillows”), indicating different interactions between pairs of V1-like cells in response to natural scenes (Pitkow 2010). In addition, Simoncelli and
Figure 2.3.5: *Joint distribution of two V1-like filters in response to natural images.* A, The predicted distribution if the filters were truly independent. The contour plot shows a “star” shaped distribution; marginal histograms for each filter are plotted along the boundary of the main plot. B, The empirical distribution measured in response to 100,000 image patches. The distribution is uncorrelated, but not independent. Probability mass has shifted from the tails along each axis in A toward the center, forming a circular-symmetric distribution.

Buccigrossi 1997 showed that the strength of conditional dependencies decreases with distance between the filters. One interpretation is that these dependencies arise because features tend to cluster in localized regions of natural scenes: when one cell exhibits a large activation it is likely that the feature driving that activation will also affect the activations of other cells. Note that these conditional dependencies are nonlinear and thus indicate the presence of high-order (i.e. beyond 1st and 2nd order) statistical structure.

To further investigate the role of specific features in natural scenes on the pairwise statistics of V1-like filters, Geisler et al. 2001 measured the co-occurrence statistics of filters that signaled the presence of specific contours. They found that when non-overlapping V1-filters signal the same contour they generally exhibit colinear position and orientations (Sigman et al. 2001). Supporting evidence for colinear structure in natural scenes was provided by (Kaschube et al. 2001). Geisler et al. 2001 also reported that colinear filters exhibit correlated phase, giving further support for notion that the phase structure of natural scenes is related to the presence of extended contours (Thomson 2001). Additionally, Geisler et al. 2001 observed that strong activation of one V1-like filter with a particular orientation is often accompanied by the activation of a parallel filter, indicat-
ing the prevalence of parallel structure in natural scenes. Supporting evidence for parallel structure was later provided by (Kayser et al. 2003).

![Figure 2.3.6: Conditional distribution of two V1-like filters in response to natural images.](image)

**A.** Plot is the same data as in Figure 2.3.5, but each column has been normalized to sum to one, defining the conditional distributions \( p(a_2|a_1) \). The conditional distribution has a “bow-tie” shape, indicating that the variance in activation of filter 2 is dependent on the value of filter 1. **B.** Two slices through the conditional distribution, at \( a_1 = [-1.79, 0.11] \), suggests when filter 1 has a large activation (magnitude), then filter 2 is more likely to be active.

### 2.3.4 Spatio-temporal statistics

Due to self motion, eye movements, and the movement of objects in the world, the retinal image varies in time. Neurons also exhibit specific spatio-temporal response properties (Orban et al. 1986; Nishimoto and Gallant 2011), and it is likely that these properties are related to the spatio-temporal structure of natural scenes. Thus the spatio-temporal statistics of natural scenes should also be considered.

Just as the autocorrelation function of natural images can be studied based on the 2D power spectrum, Dong and Atick 1995 studied the spatio-temporal autocorrelation of natural scenes by analyzing the 3D power spectrum of natural movie clips. They reported that spatio-temporal power in natural scenes is dependent on both temporal and spatial frequency. In particular, they found that the slope of the power spectrum is shallower for higher spatial and temporal frequencies. Additionally, they report that spatial and temporal frequency are dependent on one another. One explanation for these findings is that spatially correlated pixels created by contiguous objects progress across
the retinal image with a characteristic range of velocities. (Dong and Atick 1995) showed that empirical measurements are accurately characterized by a power law distribution, indicating that large but infrequent velocities accompany by more frequent, slowly-varying or static phenomena.

2.3.5 Other forms of structure in natural scenes

In addition to pixel-based image structure, there are likely other forms of scene structure that are important for how the brain forms neural representation. Included in these other forms of structure are depth (Yang and Purves 2003), binocular disparities (Cor-mack et al. 2005), and object-defining boundaries and grouping (Geisler and Perry 2006). Furthermore, natural scenes have statistical structure at relatively higher levels of abstraction. For example, natural scenes consist of collections of objects that interact in regular ways. Cars are usually found on streets and fish are usually found under water. Thus, when crossing the street it is more likely that one will encounter a car than a haliibut. Of course, these relationships are not absolute, but probabilistic. Cars can be found in showrooms, in museums, or in a garage. Fish sometimes jump out of the water, or are found on a dinner plate. It is likely that the brain represents these abstract statistical relationships, and that it exploits this information during natural vision. However, to date, there have been few attempts at measuring or characterizing this high-level scene structure. In Chapter 4, I present work on modeling such structure, and provide evidence that the human brain does indeed exploit this structure when representing natural scenes.

2.4 Coding natural scene structure and models of neural representation

The neural computations performed by an organism are constrained by a number of factors including the behavioral goals of the organism, as well as the metabolic and computational resources available. A successful organism should maximize the trade-off between these constraints and the amount of relevant information extracted from the world. Barlow 1961, borrowing principles from information theory (Shannon 1948), first formalized this concept with the efficient coding hypothesis. He proposed that an organism should make efficient use of its available computational resources by conveying as much information about the world with as few neural responses as possible. If the environment possesses structure—we have reviewed above that it indeed does—then the efficient coding hypothesis states that it is possible to formally derive the properties of the neural code as being those that maximize information about this structure. Below I present a series of theoretical models based on this premise that provide accurate predictions of neural response properties.
2.4.1 Non-redundant codes

One prediction of the efficient coding hypothesis is that the amount of information encoded by individual neurons should be maximized while the amount of information replicated amongst individual neurons should be minimized. One strategy to maximize information encoded by a single neuron is by performing response histogram equalization.

Due to biophysical constraints, a neuron operates within a limited range of responses. Information theory states that a maximally-informative strategy for a range-limited code is to distribute responses uniformly across the available range (Jaynes 1989). This implies that if a neuron encodes maximum information about natural signals, it will exhibit a response function that makes the response to any specific natural stimulus equally likely. Laughlin 1981 provided the first evidence that sensory neurons perform response histogram equalization when stimulated by natural scenes. He compared the histogram of local contrasts measured in natural scenes to the luminance response function in the eye of the blowfly. He then showed that the luminance response function is shaped such as to make the response to any local contrast in natural scenes equally probable.

![Figure 2.4.1: Example of decorrelation by whitening in the Fourier domain](image)

A, An original image of a natural scene. B, The radial average of the amplitude spectrum (square root of the power spectrum) for the original image (blue), a linear fit to this line (red), and the amplitude spectrum after applying a whitening filter calculated from the linear fit (black). The amplitude spectrum is dominated by low frequencies, indicated by the negative slope. The transformed amplitude spectrum has a flatter profile across all frequencies. C, The resulting whitened image (inverse Fourier transform of after flattening the amplitude spectrum). All local, pairwise correlations have been removed. D, The spatial representation (inverse Fourier transform) of the filter used to flatten the amplitude spectrum. Here, we zoom in on the central 5 × 5 pixels of the whitening filter. The filter has a center-surround spatial profile akin to that observed in the retina and LGN.

Just as individual neurons should maximize information by utilizing the entire range of responses, a population of neurons should use every possible combination of responses. In a strict sense, this implies that neural responses are statistically independent. Testing this hypothesis however requires characterizing a joint response histogram that grows exponentially with the number of neurons in the code. This task quickly becomes infeasible.
for realistically-sized neural populations. A simpler approach is to only consider pairwise interactions. Under these constraints, a maximally-informative population code will exhibit decorrelated responses.

In support of the decorrelation theory, Srinivasan et al. 1982 showed that lateral inhibition in the retina indeed decorrelates population responses in the presence of natural scenes. Ruderman et al. 1998 also used the decorrelation theory to predict the color opponent processes reported previously in physiology (Stockman et al. 1993) and psychophysics (Flanagan et al. 1990). Another related model proposed that the retina performs decorrelation by whitening the power spectrum of natural images. Atick and Redlich 1992 and van Hateren 1992 showed that devising a filter that flattens the power spectrum of an image in the Fourier domain resembles a center-surround filter when represented in the spatial domain (by way of inverse Fourier transform; Figure 2.4.1-D). As mentioned in Section 2.1.1, the retina and LGN exhibit similar center-surround receptive field profiles. An extension of the work of (Atick and Redlich 1992; van Hateren 1992; van Hateren 1992) to the space-time domain was provided by Dan et al. 1996. They showed that devising a filter that flattens the 3D power spectrum of natural movies provides an accurate characterization of the LGN space-time response profiles. These results suggest that subcortical processing stages of the visual system perform an operation akin to decorrelating population responses in response to natural images, thus providing solid support for the efficient coding hypothesis.

### 2.4.2 Sparse, distributed, and overcomplete codes

Though, the retina and LGN appear to decorrelate population responses in order to efficiently code for low-level (i.e. 1st and 2nd order) image structure, natural scenes contain a substantial amount of high-order structure. For example, Figure 2.4.1 demonstrates the results of decorrelating the pixels in a natural scene. Though most of the local structure has been removed, the long-range structure that defines a majority of the important features in the scene still remains. How does the primary visual cortex represent this high-order structure?

As mentioned earlier (Section 2.3.3), Daugman 1985 and Field 1987 observed that V1-like filters exhibit sparse but large activations in response to natural scenes. Having a sparse, distributed representation offers a number of theoretical benefits. First, having neurons primarily in their off state reduces metabolic consumption. Sparse codes are also helpful for downstream computation because large, infrequent activations increases the signal-to-noise ratio of relayed information.

Also, distributing representation across a small number of active neurons forces individual neurons to carry more explicit information (Barlow 1972). Note that sparse codes should not be confused with a winner-take-all (grandmother cell) representation. In fact, sparse, distributed codes are computationally more efficient than a winner-take-all...
Physiological evidence for sparse codes was documented in V1 by Vinje and Gallant 2000, who showed that responses elicited by natural stimuli are more sparse than responses evoked by synthetic stimuli. They proposed that the high-order spatial structure of natural scenes provides information from outside of the classical receptive field (CRF) to increase the specificity, and thus sparsity, of neural responses. (Weliky et al. 2003) also found that, under natural stimulation, responses in V1 exhibit properties that are consistent with a sparse, distributed code.

The first attempt to link sparse representations to neural function in the visual system was provided by Field 1987, who modeled the responses of V1-like filters to natural images. Specifically, he measured the response histograms of Gabor filters in the presence of to natural images while varying the spatial frequency of the sinusoidal grating and aspect ratio of the Guassian envelope that defined the filter. He found that the parameters that maximized the sparsity of activation histogram are closely matched to the properties of V1 neurons observed in physiology experiments (Daugman 1993). This provided the first quantitative evidence that neural representation in V1 supports a sparse code.

As noted in Section 2.1.2, V1 forms a highly overcomplete representation of the input received from the LGN, having nearly 100 V1 cells for every LGN afferent. Overcomplete representations offer a number of benefits from a signal processing perspective. For example, overcomplete codes increases invariance to small transformations (Simoncelli et al. 1992), and enables the ability to represent high resolution images with a low-resolution neural code (Chen et al. 1998). Overcomplete codes also facilitate a sparse representation (Olshausen and Field 1997).

Olshausen and Field 1996 extended the work of Field 1987 to develop a statistical model of how a population of theoretical neurons under the constraints of a sparse, distributed, and overcomplete code, respond to natural images. Rather than assuming that the theoretical neurons had a specific form of receptive field such as a Gabor filter, their model learned the shape of the receptive field. This was done by formulating the problem as a generative process: the pixel values of an image composed of $M$ pixels $x \in \mathbb{R}^{M \times 1}$ is assumed to be generated from a linear mixture of $H$ individual basis functions:

$$ x = Wh + \epsilon. $$

Each column of $W \in \mathbb{R}^{M \times H}$ can be interpreted as the receptive field for one of the model neurons and $h \in \mathbb{R}^{H \times 1}$ as the amount that each of the neurons contributes to the image description through their activation; $\epsilon$ is a residual error term of noise and nonlinear

\[ \text{For example, a code of } K \text{ binary units with } L \text{ units active at any time can represent } K!/((L!)(K - L)!)) \text{ distinct signals, whereas winner-take-all code can represent only } K. \]

\[ \text{This interpretation is not completely correct, as a RF is generally defined in terms of the stimulus features that evoke a response, not a basis set used to generate images. However, mapping the RF for the model neurons using reverse correlation gives spatial profiles that are very similar to the entries in } W \text{ (see Figure 2.4.2-B), thus we can interpret the features as receptive fields without loss of synthesis.} \]
Figure 2.4.2. *The sparse coding model trained on natural image patches.* **A**, Each square represents a column of $W$, reshaped into the size of the input patches (16 x 16 pixels). The learned features are oriented bandpass filters akin to those observed in V1. **B**, The spatial receptive fields for each of the model neurons in the bottom row (boxed in red), mapped using reverse correlation by probing each neuron with localized point light sources. The receptive fields closely resemble the learned features. Adapted from (Olshausen and Field 1996)

effects that the model cannot adequately describe. The model incorporates a sparseness constraint on the number of model neurons that are active at one time by placing a heavy-tailed prior on the distribution of $h$. The model parameters are trained by minimizing an energy function $E(x, h; W, \lambda)$ that is composed of the the squared error between an image and the model reconstruction, plus a weighted regularizer function $\Psi(h)$ that influences the vector $h$ to be sparse:

$$E(x, h; W, \lambda) = (x - Wh)^T(x - Wh) + \lambda \Psi(h)$$

The regularization parameter $\lambda$ determines the strength of the sparsity constraint. The model can also incorporate an over-complete representation by setting $H$ to be greater than $M$ (When $H = M$ and the system is assumed to be noiseless ($\epsilon = 0$), the sparse coding model is equivalent to the independent components analysis (ICA) algorithm (Bell and Sejnowski 1995; Olshausen and Field 1997; Hyvarinen and Oja 2000) When this “sparse coding” model is trained on thousands of (whitened) patches sampled from natural
images, the features that are learned (columns of $W$) exhibit properties that are strikingly similar to the spatial receptive fields of simple cells observed in V1. Specifically, the learned features are spatially-localized, oriented and bandpass filters (Figure 2.4.2).

### 2.4.3 Alternative image coding strategies

Theoretical models based on redundancy reduction and sparse coding provide compelling evidence for the correspondence between cortical representation and the statistical structure of natural scenes. In addition to these models, other theoretical models based on alternative coding strategies provide even more support for this notion.

For example, the visual system must be able to represent the world in the presence of noisy sensory inputs (i.e. optics, blur, etc.) and channel noise (i.e., limited precision in neural code, cell death, refractory periods, etc.). Thus, redundancy reduction cannot be the only coding strategy employed in the retina, as some redundancy is required to deal with this noise (Atick and Redlich 1990). Doi et al. 2006 applied this idea to formally derive the optimal neural codes for retinal cells in the presence of both input and channel noise. The model minimizes the squared error of image reconstruction while optimizing the trade off between efficiency and redundancy. They found that the optimal coding strategy decomposes into two terms. The first term is a Wiener filter that compensates for the noisy inputs. The second term turns out to be an optimal code for a noisy Gaussian channels, which accurately approximates a population of neurons with limited channel capacity (Borst and Theunissen 1999).

Neurons must also operate under biophysical constraints such as the availability of metabolic resources required for spiking. Karklin and Simoncelli 2011 proposed a nonlinear model of retinal coding that optimizes the tradeoff between metabolic cost of neural spiking and the amount of information transferred by the code. The model is also formulated to learn the response nonlinearity for each model neuron rather assuming a particular functional form such as rectification. When trained on natural image patches, the model neurons exhibit luminance blob and center-surround receptive fields like those observed RGC cells. The model also identifies half-wave rectification nonlinearities like those used in standard computational models of retinal processing (Carandini et al. 2005).

Another coding strategy is to have high-level representations that vary slowly in time. While the activity of neurons in the early stages of visual processing vary quickly in response to simple, localized features, the activity of neurons involved in higher-level image representation—such as the presence of an object—should vary more slowly in time (Hyvarinen et al. 2003). Wiskott and Sejnowski 2002 and Wiskott and Berkes 2003 used this notion to develop the “slow feature analysis” model. In this model the activities of model neurons are forced to vary smoothly in time by imposing a constraint on the second temporal derivative of their activities. The model is trained on sequences of image patches sampled by moving sliding and rotating windows over natural images (to mimic panning and rotation observed in natural movies). Once trained, the model yields theoretical neurons that demonstrate a number of documented response properties associated with both
simple and complex cells, including velocity and temporal frequency selectivity (Hubel and Wiesel 1962; De Valois et al. 1982), orthogonal and non-orthogonal orientation inhibition (De Valois et al. 1982), and invariance to phase (Hubel and Wiesel 1968; Movshon et al. 1978a).

In general, statistical feature learning models like sparse coding and slow feature analysis assume that the activities of the code units are independent of one another. However, as was discussed in Section 2.3.3, V1 simple cells display complex dependencies in response to natural scenes that cannot be accounted for by models that assume an independent code. This residual dependency can be implicitly modeled by grouping code units into local neighborhoods, where the units within a given neighborhood are influenced to have similar activations. Hyvarinen et al. 2001 used this approach to modify the ICA algorithm. The modification groups model neurons into neighborhoods that are influenced to have similar activations. When trained on natural images, this topographic ICA model learns features that, within a group, have similar orientations and/or scales, but vary in location, and phase. When the feature representation of the entire population is viewed at once, a functional organization of the features emerges. The functional organization shares many qualities with the cortical organization of representation in V1 (Wong-Riley 1979; Dow 2002). Specifically, the orientation of the features with relatively higher spatial frequency varies smoothly across the map while low frequency, non-oriented features form blob-like structures. Similar approaches have been applied to other generative models (Osindero et al. 2006; Garrigues and Olshausen 2008), as well as wavelet-based models (Simoncelli and Buccigrossi 1997; Wainwright et al. 2001) to learn functional topographies from natural images.

2.4.4 Beyond image coding

Though theoretical models based on efficiently coding natural image structure provide accurate characterizations of neural function and organization in early visual areas, they provide poor characterizations of neural properties at later stages of visual processing. This is likely because these later stages of processing are likely not representing the 2D image per se. Instead, these areas are likely using image information from lower processing stages to infer higher-order scene properties. Thus, a reasonable approach to modeling representation in later stages of visual processing is to develop hierarchical theoretical models where the features represented at one stage in the hierarchy are modeled as a nonlinear function of the features represented in the previous layer.\footnote{Note that nonlinearity at each level is necessary, as building a hierarchy of linear transformations can be represented equally well as a single layer of linear transformations.}

Working along these lines, Karklin and Lewicki 2005 developed a hierarchical generative model of images that includes a linear input layer akin to sparse coding model, but also includes a hierarchical prior distribution on the variances of the code unit activations. The prior is defined in terms of a second layer of representation that aims to capture de-
dependencies among the code units in the first layer. When the model is trained on natural images, the first layer learns features that resemble those identified by the sparse coding model, namely Gabor-like spatial functions. The second layer learns feature representations that exhibit properties associated with complex cells, including phase invariance (Hubel and Wiesel 1968; Movshon et al. 1978a), surround suppression (Jones et al. 2002), mask suppression (Bonds 1989), and 2nd order phenomena (Baker and Mareschal 2001). The second layer also identifies a number of visual features of intermediate complexity, including extended contours, junctions, and textures.

Due to retinal, object, and self-motion, the projection of the 3D environment on the 2D retina forms a time-varying image with complex statistical structure. In order to infer 3D properties of the world from the 2D image, it would be reasonable for the visual system try and disambiguate the information in the image that is related to form from the information related to the motion. Cadieu and Olshausen 2012 developed a hierarchical generative model that attempts to capture the separation of form and motion in time-varying images. Their model also incorporates sparse-coding-like first layer. However the units in the first layer are complex-valued. The magnitude components of the features capture spatial structure, while the phase components capture interpolation of that structure across the receptive field. The second layer of the model is factored into two sets of variables, one that captures structure in the form components of the features in the first layers, and a second that captures structure in the motion components. When this model is trained on clips of natural movies, the first layer learns direction-selective features akin to space-time Gabor functions. The second layer learns hierarchical spatial features with various properties of intermediate complexity including contours, textures and texture boundaries, and cross-orientation tuning. The second layer also learns hierarchical motion features that capture many forms of motion including localized and full-field translation, rotation, and shearing.

2.4.5 Compositional hierarchical models

Formal hierarchical models are an principled way of characterizing high-order, nonlinear statistical structure in natural images and thus could provide promising theories of neural representation in later stages of the visual system. However, due to complications related to explaining away effects (Pearl 1988) and the intractability of integrals needed for inference (Bishop 2006), developing and implementing formal hierarchical models with more than 2 layers has proven difficult in practice.

Recently a more tractable approach based on layer-wise model composition has proven successful for training hierarchical models of arbitrary depth (Hinton et al. 2006; Hinton and Salakhutdinov 2006; Bengio et al. 2007). Rather than estimating the parameters for a full hierarchical model all at once, the model is composed of multiple stages of

\(^5\) An extension of this model which captures the covariances of unit activations, is able to capture non-stationary image structure resulting from different contexts (Karklin and Lewicki 2009)

\(^6\) In a rough sense, the model can be interpreted as a theoretical approximation to the ventral and dorsal streams associated with the primate visual system.
Figure 2.4.3: Compositional model training. A hierarchical model is composed by training multiple layers of simpler models. A, In training stage I, the structure of some input signal (i.e. natural image patches) is modeled using a single-layer generative model with a latent feature code conveyed in the activations $a^{(1)}$. B, In training stage II the parameters for the first layer model $W^{(1)}$ are frozen and the activations $a^{(1)}$ are used as data $x^{(2)}$ for the second layer. C, What results is a hierarchical model that captures more complex input features at each stage in the hierarchy. Note that the connection structure presented in this cartoon is not meant to define a specific model architecture, but to convey general approach of modeling input structure with a set of feature codes.

simpler models. Each of the simpler models is trained sequentially (Figure 2.4.3): A first layer model is trained to learn the statistical structure of some input signal (i.e. natural images). Afterwards, the first layer parameters are frozen and the code unit activations for that model are calculated for the same input signals. These activations are used as inputs for a second layer model, which learns the structure in the code activations, and thus captures higher-order structure in the input. This process can be repeated for an arbitrary number of stages, resulting in a “deep,” hierarchical representation of the input signal. Compositional models have shown impressive performance on a number of computer vision (Hinton et al. 2006; Bengio et al. 2007; Le et al. 2012), natural language processing, and bioinformatics (Hinton and Salakhutdinov 2006) tasks. Although their formulation is generally motivated by the hierarchical organization of biological sensory systems, there have been few attempts to assess compositional models on their potential to provide theoretical accounts of neural representation (though see Lee et al. 2008).

2.5 A general framework combining linearized system identification and statistical feature learning

In Sections 2.2 and 2.4 I discussed two complimentary approaches for studying visual representation: linearized modeling and probabilistic models that learn statistical features of natural scenes. These two approaches can be combined in order to study natural scene processing across the entire visual system. To do this, a statistical model is first used to learn a feature representation of natural scenes. The learned features are then used to
implement the transformation required for linearized modeling. The resulting linearized model provides a stimulus-response function that is easily interpreted in terms of the learned features.

As an example of this framework in action, take the sparse coding model discussed earlier. The features identified by the sparse coding model are strikingly similar to V1 simple-cell spatial response profiles. A nonlinear stimulus transformation can be obtained by calculating the sparse coding feature activations evoked by the stimuli,\(^7\) and using the activations as a feature transformation (Figure 2.5.1). An LEM based on linear sparse coding features provides accurate predictions of simple cell activity in V1 (Vu et al. 2011).

While this framework is effective for modeling simple-cell-like representation in V1, it is unclear if the approach will be effective for modeling representation in later visual areas where computation becomes more complex. Hierarchical models provide a promising avenue for testing this approach. The features that emerge in the higher layers of hierarchical models can provide potential forms of intermediate- and high-order neural representation. These features can be validated by estimating and testing linearized models of responses evoked in neurons and voxels in later stages of the visual hierarchy. Yet another promising approach is to learn statistical features from more abstract visual primitives (i.e. compared to pixels), such as objects. In the remainder of this thesis I use the proposed framework to derive, test and compare models of representation at various stages of the visual hierarchy during natural vision.

---

\(^7\)This involves solving a nonlinear $L_1$ optimization problem to that maximize the stimulus likelihood with respect to the model activations.
Figure 2.5.1: General framework combining statistical feature learning and linearized modeling.  
A, The linearized system identification framework described in Figure 2.2.1. Here, the linearizing transformation comes from statistical feature learning. B, Natural scenes (left box) have a specific statistical structure that can be modeled using statistical feature learning techniques (center box). Statistical feature learning can identify latent features in natural scenes that are theoretically important for neural representation. For example, oriented bandpass filters emerge from analysis of the pixel intensity distributions of small image patches extracted from natural scenes (right box). When these features are used to perform the feature transformation in the linearized modeling framework, the resulting models accurately account for brain activity measured in early cortical stages of the primate visual system.
Chapter 3

Modeling hierarchical image representation
3.1 Introduction

The primate cortical visual system is composed of a hierarchy of processing stages. The first processing stage in the hierarchy, area V1, has been well studied using synthetic (Hubel and Wiesel 1968; Movshon et al. 1978b) and natural stimuli (David et al. 2004). These studies suggest that V1 represents simple image features akin to oriented edges. Similar features can be derived automatically by performing statistical feature learning on natural image patches (Olshausen and Field 1996; Bell and Sejnowski 1997). The outputs of V1 neurons project primarily to area V2. When probed with simple synthetic stimuli such as Cartesian gratings (Levitt et al. 1994), V2 exhibits similar tuning properties to V1. However, V2 neurons also appear to represent higher-order image structure. Evidence for this comes from studies that have probed V2 neurons with more complex stimuli such as second order gratings (Baker and Mareschal 2001), illusory contours (von der Heydt et al. 1984), texture-defined contours, (von der Heydt and Peterhans 1989), complex image features (Hegde and Van Essen 2003; Ito and Komatsu 2004), natural textures (Freeman et al. 2013), or natural images (Willmore et al. 2010). Currently it is unclear how higher-order image structure represented in V2 differs from the image structure that is represented in V1.

One possibility is that the representation of image structure is built up in a hierarchical fashion. This notion is supported by the hierarchical organization of the visual system (Felleman and Van Essen 1991). According to this coding strategy, V1 represents localized, edge-like features while V2 represents higher-order image structure as nonlinear combinations of these features. In principle, this type of representation could be learned from natural scenes using a hierarchical model. In this work we would like to test if different levels of a hierarchical representation learned from natural scenes can differentially account for the responses evoked in V1 and V2 neurons. We plan to do this using the general modeling framework proposed in Chapter 2. However, using the proposed framework requires a feature learning method that can learn a hierarchical representation of natural scenes.

Previous work by Karklin and Lewicki 2005 2009 developed a 2-layer generative model that to learn a hierarchical representation of natural scenes. The first layer learns dependencies amongst pixels while the second layer learns dependencies amongst the activations of the code units in the first layer. When trained on natural images, the first layer of the model learns V1-simple-cell-like features while the 2nd-layer features captures higher-order image structure such as extended contours and textures. The model also explains a number of nonlinear response properties of neurons observed in complex V1 cells and cells in V2. It is unclear, however, if two layers of representation are sufficient to account for all the response properties exhibited by V2 neurons. Extending the work of Karklin and Lewicki 2005 2009 to learn deeper hierarchical representations proves to be difficult, however, due of several complications that arise when learning joint probabilistic models with multiple stages of dependencies.

A more tractable approach to learning deeper hierarchical models is to compose them
using the layer-wise training process described in Section 2.4.5 (Hinton et al. 2006). However, to do so, one must make a decision as to which feature learning model should be trained at each layer of the hierarchy. In this chapter we use the restricted Boltzmann machine (RBM; Ackley et al. 1985) for training each layer. The RBM model has a number of properties that make it useful for composing hierarchical models. First, the RBM can be formulated to allow the input data and latent variable activations to follow any distribution from the exponential family (Welling et al. 2005). This enables the RBM to model a diversity of input domains. Furthermore, statistical inference for the RBM is efficient so the model can be trained very efficiently on large datasets. This makes the RBM useful for unsupervised feature learning which generally requires large amounts of unlabeled data. Finally, as we will develop later in the chapter, the RBM can easily incorporate domain knowledge in order to influence the learned code to have specific properties such as sparsity or specificity for particular features.

The roadmap for the remainder of the chapter is as follows. We begin by describing the formulation and training of general Boltzmann machines and extend the discussion to RBMs. We then go on to use RBMs to train a hierarchical model of color natural images. We then investigate the coding properties of the features learned at each stage of the model hierarchy. Then, to contextualize the properties of the learned features, we compare the features to previous findings in neurophysiology and to features learned using related theoretical models such as sparse coding (Olshausen and Field 1996) and (Karklin and Lewicki 2005)’s hierarchical model. Finally, we validate the learned features as potential forms of neural representation. We do this by constructing predictive models based on the learned features that are used to predict neurophysiological responses evoked in macaque V1 and V2 cells during natural vision.

3.2 Unsupervised feature learning with restricted Boltzmann machines

Restricted Boltzmann machines are a subclass of models known as Boltzmann Machines (Ackley et al. 1985) (BM), which are stochastic neural networks and can be interpreted as a probabilistic version of Hopfield networks (Hopfield 1982). To get a deeper understanding of how RBMs are trained, we first formulate the BM and introduce the general training procedure for all BMs (including RBMs).

3.2.1 Boltzmann machines

Boltzmann machines are members of a general class of statistical models known as Markov random fields (MRFs). The goal of the MRF is to model a set of observed data $\mathbf{x}$ in terms of a set of visible random variables $\mathbf{v}$ and a set of latent random variables $\mathbf{h}$. The role of the visible units is to approximate the true data distribution. The goal of the latent variables it to extend the expressiveness of the model by capturing underlying causes of the
observed data. The latent variables are often referred to as hidden units, as they do not
directly observe the data. MRFs can be formulated as a product of experts (PoE; Hinton
2002), which defines a probability distribution over both \( \mathbf{v} \) and \( \mathbf{h} \) by taking the product
of \( C \) individual potential functions \( \phi_c(\mathbf{v}, \mathbf{h}; \theta) \) (referred to as “experts”). The functional
form of each potential is determined by a set of parameters \( \theta \). The joint probability of
the visible and hidden variables states under the PoE is as follows:

\[
p_{PoE}(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z_{PoE}} \prod_c \phi_c(\mathbf{v}, \mathbf{h}; \theta). \tag{3.2.1}
\]

Here the partition function \( Z_{PoE} \) is a normalizing constant calculated by integrating over
all possible variable states \( \mathbf{v}', \mathbf{h}' \in \mathbf{V}', \mathbf{H}' \):

\[
Z_{PoE}(\theta) = \int \int \prod_c \phi_c(\mathbf{v}', \mathbf{h}'; \theta) d\mathbf{v}' d\mathbf{h}'. \tag{3.2.2}
\]

In the case that either the visible or hidden units take on discrete values, the integrals
above are replaced accordingly by discrete summations.

Markov random fields (and thus BMs) are a special case of the PoE model where each
of the potential functions belongs to the exponential family:

\[
\phi_c(\mathbf{v}, \mathbf{h}; \theta) = e^{-\psi_c(\mathbf{v}, \mathbf{h}; \theta)}. \tag{3.2.3}
\]

Here, \( \psi_c(\mathbf{v}, \mathbf{h}; \theta) \) is a potential function operating on a clique of locally-connected units,
namely \( \mathbf{v} \) and \( \mathbf{h} \). The probability density defined by a MRF with \( C \) cliques is defined by
the Boltzmann distribution:

\[
p_{MRF}(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z_{MRF}} \prod_c e^{-\psi_c(\mathbf{v}, \mathbf{h}; \theta)} = \frac{1}{Z_{MRF}} e^{-\sum_c \psi_c(\mathbf{v}, \mathbf{h}; \theta)} = \frac{1}{Z_{MRF}} e^{-E(\mathbf{v}, \mathbf{h}; \theta)}, \tag{3.2.4}
\]

where the MRF partition function is:

\[
Z_{MRF}(\theta) = \int \int e^{-E(\mathbf{v}', \mathbf{h}'; \theta)} d\mathbf{v}' d\mathbf{h}'. \tag{3.2.5}
\]

The term \( E(\mathbf{v}, \mathbf{h}; \theta) \) is referred to as the energy function and sums over the local clique
potentials to quantify the degree of interaction between \( \mathbf{v} \) and \( \mathbf{h} \). Thus, \( E(\mathbf{v}, \mathbf{h}; \theta) \) is
Figure 3.2.1: *Graphical model of the Boltzmann machine.* A set of visible units $v$ and a set of hidden units $h$ are used to model observed data. The model is parameterized by a set of parameters $\theta = \{W, A, B, a, b\}$. The parameter $W$ is a matrix of undirected connections weights that captures interactions between the visible and hidden units. The parameters $A$ and $B$ are matrices of weights that capture intra-layer interactions within the hidden and visible units, respectively. No unit has a connection to itself. A set of biases, $a$ and $b$ (not shown) capture offsets for the hidden and visible units respectively.

defined by the graph structure of the model; an alternative interpretation is that one can formulate an energy function that defines an MRF with particular properties.

Boltzmann machines are fully-connected MRFs (excluding connections from any unit to itself), where all connections between units are symmetric. The energy function for a BM with binary visible and hidden units is defined as:

$$ E_{BM}(v, h; \theta) = -v^T W h - h^T A h - v B v^T - b^T v - a^T h, \quad (3.2.6) $$

where, again, $\theta = (W, A, B, a, b)$.

The first term in the BM energy function captures inter-layer interactions between $V$ individual visible and $H$ individual hidden units. The interactions between the visible and hidden units are defined by the weight matrix $W \in \mathbb{R}^{V \times H}$. The second and third terms capture intra-layer interactions within the visible and hidden layers. These interactions are defined by the weight matrices $A \in \mathbb{R}^{H \times H}$ and $B \in \mathbb{R}^{V \times V}$. The last two terms capture baseline offsets, defined by the bias vectors $a \in \mathbb{R}^{H \times 1}$ and $b \in \mathbb{R}^{V \times 1}$. The likelihood of the visible unit states for the Boltzmann machine is calculated by marginalizing over the hidden unit states:

1In general, the units can follow any distribution from the exponential family, but for this development we assume that all units are binary variables.
\[ p_{BM}(v; \theta) = \int p(v, h; \theta) dh \]
\[ = \int \frac{1}{Z_{MRF}} e^{-E_{BM}(v, h; \theta)} dh. \]  

3.2.2 Training Boltzmann machines

To train the BM model, we attempt to find the optimal parameters \( \theta^* \) that maximize the likelihood of the observed data \( \mathcal{L}(v; \theta) = p_{BM}(v; \theta) \):

\[ \theta^* = \arg \max_{\theta} \mathcal{L}(v; \theta). \]

For a number of reasons including numerical stability, a common strategy for solving this problem is to perform gradient descent on the negative log likelihood function \( \ell(v; \theta) = -\log \mathcal{L}(v; \theta) \) with respect to the model parameters \( \theta \) (see Appendix A.1.1 for details):

\[ \frac{\partial \ell_{BM}(v)}{\partial \theta} = \mathbb{E}_{p(h|v)} \left[ \frac{\partial}{\partial \theta} E_{BM}(v, h) \right] - \mathbb{E}_{p(v', h')} \left[ \frac{\partial}{\partial \theta} E_{BM}(v', h') \right]. \]  

(3.2.8)

The gradient in Equation 3.2.8 is composed of two expectations. The first term is the expected energy function gradient when the hidden unit states are conditioned on the states of the visible units. The second is the expected gradient of the energy function under the joint probability of the visible and hidden units. Calculating the exact gradients for BMs is difficult in practice because these expectations involve integrating over all possible model states. This task is generally infeasible for realistically-sized models. An alternative approach is to use Markov chain Monte Carlo (MCMC) methods to approximate these integrals (Andrieu et al. 2003). Using MCMC estimates gives the following approximation to the gradient (see Appendix A.1.1 for details):

\[ \frac{\partial \ell_{BM}(v)}{\partial \theta} \approx \left\langle \frac{\partial E(v, h)}{\partial \theta} \right\rangle_{x} - \left\langle \frac{\partial E(v, h)}{\partial \theta} \right\rangle_{\theta}, \]  

(3.2.9)

where \( \langle \cdot \rangle_{p(\cdot)} \) is the mean of samples drawn from the distribution \( p(\cdot) \). The first term is the average gradient of the energy function when the states of the current model are driven by observed data \( x \). The second term is the average energy function gradient when the states are sampled from the equilibrium distribution of the Markov chain defined by the current model parameters \( \theta \). Obtaining unbiased estimates of the first term is trivial if the observed data are sampled \( i.i.d. \) and we have in hand the expression for \( p(h|v) \). However, obtaining samples from the equilibrium distribution generally takes an infeasible amount of time for most BMs. For this work we instead employ an important subclass of BMs—restricted Boltzmann machines—that allow efficient sampling from the model distribution due to reduced connectivity.
3.2.3 Restricted Boltzmann machines

Removing all intra-layer connections in the BM energy function (i.e. setting $A$ and $B$ in Equation 3.2.6 to zero) gives the energy function for the restricted Boltzmann machine (RBM):

$$E_{RBM}(v, h; \theta) = -v^TWh - b^Tv - a^Th.$$  \hspace{1cm} (3.2.10)

The RBM energy function defines an MRF on a bipartite graph that is composed of a layer of visible units and a separate layer of hidden units (Figure 3.2.2). The bipartite structure of the RBM has important conditional independence properties that can be exploited to run an efficient Markov chain. Specifically, the states of units in the hidden layer are independent of one another given the states of the visible units (see Appendix for details A.2.3)

$$p(h|v) = \sigma(a + v^TWW).$$  \hspace{1cm} (3.2.11)

Here, $\sigma(z) = 1/(1 + e^{-z})$ is the point-wise logistic sigmoid function. In a similar fashion, the states of the units in the visible layer are independent of one another given states of the hidden units:

$$p(v|h) = \sigma(b + Wh).$$  \hspace{1cm} (3.2.12)

This structure allows the conditional probabilities for all units within a layer to be calculated in parallel. This enables blocked Gibbs sampling and thus provides an efficient means for running the Markov chain.
3.2.4 Gibbs sampling for RBMs

Figure 3.2.3 shows the Gibbs sampling routine for RBMs. Starting at step $s = 0$ the values of the hidden units $h^{[0]}$ are inferred from observed data $v^{[0]} = x$ via Equation 3.2.11. This is referred to as the “upward pass” (Figure 3.2.3, left) through the model. Following the upward pass, the values of $v^{[1]}$ are calculated conditioned on the states of $h^{[0]}$ via Equation 3.2.12. This is referred to as the “downward pass.” The states of $v^{[1]}$ are then used to calculate the states $h^{[1]}$. This procedure is repeated for $s = S$ steps until the corresponding Markov chain reaches the equilibrium distribution under the current model parameters.

After the Markov Chain reaches equilibrium, samples drawn from the equilibrium distribution are used to approximate the gradient. Specifically, the sample average $\left\langle \frac{\partial}{\partial \theta} E_{\text{RBM}}(v^{[0]} h^{[0]}) \right\rangle_{v}$ is used to approximate the expectation $E_{p(h|v)} \left[ \frac{\partial}{\partial \theta} E_{\text{RBM}}(v, h) \right]$ in Equation 3.2.8, and the sample average $\left\langle \frac{\partial}{\partial \theta} E_{\text{RBM}}(v^{[S]} h^{[S]}) \right\rangle_{\theta}$ is used to approximate the expectation $E_{p(v', h')} \left[ \frac{\partial}{\partial \theta} E_{\text{RBM}}(v', h') \right]$.\footnote{As $s \to \infty$, the approximation approaches that of the maximum likelihood gradient in 3.2.8}

3.2.5 Contrastive divergence learning

Though the RBM provides an efficient Markov chain, it is often still necessary for the chain to run for a very long time in order to mix properly. A learning rule known as contrastive divergence (CD; Hinton 2002) can be used to accelerate RBM training. In CD learning, rather than running the Markov chain until it reaches equilibrium, the chain is only run for a few steps $s \ll S$. This results in the following CD learning parameter updates (see Appendix A.1.2 for details):

$$
\Delta W = \epsilon \left( \left\langle v^{[0]} h^{[0]} \right\rangle - \left\langle v^{[S]} h^{[S]} \right\rangle \right),
\Delta a = \epsilon \left( \left\langle h^{[0]} \right\rangle - \left\langle h^{[S]} \right\rangle \right),
\Delta b = \epsilon \left( \left\langle v^{[0]} \right\rangle - \left\langle v^{[S]} \right\rangle \right),
\tag{3.2.13}
$$

where $(v^{[0]}, h^{[0]})$ are the states of the visible and hidden units when being driven by observed data, and $(v^{[S]}, h^{[S]})$ are the the states after $s$ Gibbs sampling steps. Though CD learning only coarsely approximates the true gradient (Hinton 2002), it works quite well in practice, even for $s = 1$.

3.2.6 Stochastic gradient descent optimization

To calculate the parameter updates in Equation 3.2.13 we use the stochastic gradient descent (SGD) method. In contrast to batch gradient descent (BGD), which estimates the average gradient based on all available training data, SGD uses random subsets of the data
Figure 3.2.3: Performing Gibbs sampling for the RBM. **Left.** At step $s = 0$, the hidden units states are driven by observed data, $v^{[0]} = x$. Because of the bipartite structure of the graph, all hidden unit states can be estimated in parallel. **Center.** The hidden states are then used to infer the states of the visible units, which is also performed in parallel. The inferred states of the visible units are then used to infer the states of the hidden units again. **Right.** This process is repeated until the Markov chain reaches the equilibrium distribution at step $s = S$.

to calculate many estimates of the gradient. This approach offers a number of benefits over BGD. First, BGD requires computation time on the order $O(NM)$ for each gradient calculation, where $N$ is the number of observations and $M$ is the dimensionality of each data point. This scaling is troublesome for problems that require large amounts of training data for parameter estimation. On the other hand, SGD only requires $O(BM)$ complexity, where $B \ll N$ is the size of the subset of the data used for gradient calculation (often referred to as a “mini-batch”). This makes SGD much better suited for big data problems like feature learning. Second, BGD does not take advantage of redundant structure in the training data. If data points are replicated across the training set, their contribution to the gradient will be replicated. However, by sampling random subsets of the data, SGD is potentially able to witness such redundancy before observing all the available data. Thus, SGD can estimate the true gradient much faster. Stochastic methods also provide random approximations to the true gradient that, on average, can be robust to noise in the training data and are helpful for escaping local minima in the likelihood surface. Combining the contribution of many noisy gradient estimates also compliments the approximate learning rule supplied by contrastive divergence.

### 3.2.7 Restricted Boltzmann machines for continuous data

The energy function in Equation 3.2.10 assumes that both the visible and hidden units are binary random variables. However, pixels from natural images take on continuous values. Thus, binary visible units are inappropriate for modeling natural images. However, the RBM energy function can be formulated to model visible and hidden units that follow any distribution from the exponential family, including continuous densities such as a the Normal distribution (Welling et al. 2005). The following energy function defines a more appropriate RBM for modeling natural images, having binary hidden units and Normally-distributed visible units, each with mean $b_i$ and independent variance $\sigma^2_i$ (Cho et al. 2011):
\[ E_{\text{NRBM}}(v, h; \theta) = -\left( \frac{v}{\sigma^2} \right)^T W h + \left\| \frac{v - b}{2\sigma} \right\|^2 - a^T h. \] (3.2.14)

Training this RBM requires estimating the parameters \( \theta = \{W, b, a, \sigma^2\} \). However, training can be simplified by preprocessing the observed data to have unit variance, which makes learning the values of \( \sigma^2 \) unnecessary. Rescaling the input variables thus gives the following simplified energy function:

\[ E_{\text{GRBM}}(v, h; \theta) = -v^T W h + \frac{1}{2}(v - b)^T(v - b) - a^T h, \] (3.2.15)

with reduced parameter set \( \theta = \{W, b, a\} \). Training RBMs with Normally-distributed visible units follows the same procedure as for binary visible units. The only differences are that that conditional distribution for the visible units follows a multivariate normal distribution (see Appendix A.2.3 for details):

\[ p(v|h) = \mathcal{N}(b + Wh, I^V), \] (3.2.16)

where \( I^V \) is \( V \)-dimensional identity covariance matrix. Additionally, the contrastive divergence weight updates for the visible biases take a slightly different than form than Equation 3.2.13:

\[ \Delta b = \epsilon \left( \langle v^{[0]} \rangle - b \right) - \left( \langle v^{[s]} \rangle - b \right). \]

Though binary visible units are inappropriate for modeling pixel values, we will see in Section 3.2.9 that binary visible units are necessary when training higher layers of a hierarchical model composed of multiple RBM.

### 3.2.8 Targeted regularization for RBMs

In general, RBMs are unaware of any specific task that the features learned by the model may be used to perform. Here we would like to use the features learned by the RBM not only as potential forms of neural representation, but also as building blocks for a hierarchical model. Thus, we would like to influence the RBM training procedure so that the learned feature representations have desired properties that are motivated by biology and theory. In this work we do so using a targeted regularization technique introduced by (Goh et al. 2010). Targeted regularization is implemented by solving the following regularized optimization problem.

\[ \theta_{\text{reg}}^* = \arg \min_{\theta} \ell(v; \theta) + \lambda \Psi(h_x). \] (3.2.17)
Here, $\ell(v; \theta)$ is the RBM negative log likelihood function and $\Psi(h_x)$ is a regularization term that penalizes the solution based on the states of the hidden units when being driven by data $h_x$ (i.e. $h^{[0]}$ for CD learning). The degree of penalty is determined by the parameter $\lambda$. The function $\Psi(h_x)$ compares the distribution of data-driven states to a target distribution $\psi$ that possesses the properties we would like the learned feature representation to exhibit. For binary hidden units we choose $\Psi(h_x)$ to be the cross entropy distance between $h_x$ and $\psi$ (Nair and Hinton 2009). This choice of regularizer gives the following CD weight updates (see Appendix A.3 for details):

$$\Delta_{\text{reg}} W = \epsilon \left( \langle v^T \tilde{h} \rangle - \langle v^T h^{(s)} \rangle \right)$$  \hspace{1cm} (3.2.18)

$$\Delta_{\text{reg}} a = \epsilon \left( \langle \tilde{h} \rangle - \langle h^{(s)} \rangle \right)$$  \hspace{1cm} (3.2.19)

where $\tilde{h} = (1 - \phi)h^{[0]} + \phi \psi$; $\phi = \frac{\lambda}{\epsilon}$ is a regularization hyperparameter that exists between zero and one and which defines the degree of interpolation between the data-driven activations and the target distribution. Note that the updates $\Delta_{\text{reg}} b$ are the same as for the unregularized optimization problem. This form of regularized optimization for RBMs is general in that $\psi$ can have any property, conditioned that it shares the support of the hidden unit states.

### 3.2.8.1 Inducing sparsely-distributed representations

As discussed in Section 2.4.2, the early stages of cortical visual processing (area V1) exhibit a sparse, distributed code. Thus if the task for the RBMs is to search for valid theoretical neural representations, we would like to influence an RBM to learn sparsely-distributed codes. Additionally, we plan to use RBMs as building blocks for constructing compositional hierarchical models. Sparsely-distributed codes contain more explicit information about the modeled input than do dense codes (Field 1987). Thus, sparse codes are potentially more useful signals for performing higher-order tasks. It is therefore advantageous for an RBM that is to be used as a building block for a compositional model to have a sparsely-distributed code in the hidden layer.

Several methods have been devised to influence RBMs to learn sparsely-distributed codes (Lee et al. 2008; Nair and Hinton 2009). However, these methods generally involve keeping a running average of the hidden unit activations across the entire training dataset. This running average is used to calculate a global offset that adjusts the hidden units to have low activation probability. However, these methods have a number of disadvantages. For one, sparsity is not guaranteed, as having many units in a low, but non-zero operating regime fulfills the regularization criterion. Also, having a low average activation does not ensure that the hidden units will exhibit specificity for a particular feature (Nair and Hinton 2009).

In this work we induce sparsity using a targeted regularization procedure described in (Goh et al. 2010). Specifically, we design a target distribution $\psi(h_x)$ that is calculated from the data-driven hidden unit activations such that $\psi(h_x)$ is sparsely-distributed
Figure 3.2.4: An example of calculating a sparse target distribution $\psi(h)$ from a set of hidden unit states $h$. Top row, a theoretical data-driven activation distribution $h$, sampled from $Beta(2,6)$ (left), along with the corresponding ranked probabilities (center) and histogram (right). Middle row, the equalized distribution $h_{EQ}$ (left) along with the corresponding ranked probabilities (center) and histogram (right). Bottom row, the target probability distribution $\psi(h)$ for $\mu = 0.2$ (left), along with the corresponding ranked probabilities (center) and histogram (right).

across the units (Figure 3.2.4). The transformation of the hidden units into the target distribution is implemented by first performing histogram equalization on the hidden unit activities such that the probability of response is distributed equally across all units. A simple form of histogram equalization is performed by replacing each hidden unit probability by its rescaled rank across the population (Figure 3.2.4, middle right). Rescaling is performed such that each rank exists between zero and one. This rescaling allows the resulting distribution $h_{EQ}$ to be transformed into a sparse distribution using following power law relationship:

$$\psi(h) = h_{EQ}^{\frac{1}{\mu}-1}.$$  \hspace{1cm} (3.2.20)

Here, $0 < \mu < 0.5$ is the mean of a positively-skewed target distribution. Thus, as $\mu \to 0$, the distribution becomes increasingly sparse.

When $H$ hidden units are driven by a mini-batch of $K$ samples, the resulting $H \times K$
matrix of activations $H_x$ is transformed along its columns to form the sparsely-distributed target distribution. Note that this distribution will influence a sparse code, but often one in which only a very few units are always active. In order to form a less redundant code we also induce specificity in hidden unit activations. This is done by performing an additional sparsification transformation along the rows of $H_x$. This additional transformation is justified in part by the fact that sparsity and specificity are generally uncorrelated (Willmore and Tolhurst 2001). Having units that code for specific features also has a number of theoretic benefits including improved signal-to-noise.

3.2.9 Deep Belief Networks

Multiple RBM models can be trained in a layer-wise fashion to form a hierarchical model known as a deep belief network (DBN; Figure 3.2.5; Hinton et al. 2006). During compositional training of DBNs, an initial RBM is first trained on an input signal (e.g. pixels). The visible units of the 1st layer RBM are chosen to match the distribution of the input signal, while the hidden units are assumed to be binary. Once trained, the parameters of the 1st layer RBM are frozen and the feed-forward activations of the hidden units are used as input to a second RBM with binary visible and hidden units. The 2nd layer RBM is trained to learn the dependencies between the hidden unit activities of the 1st layer RBM, thus forming a nonlinear, hierarchical feature representation of the input signal. This compositional training process can be repeated for an arbitrarily deep architecture.

Compositional training allows tractable unsupervised learning of hierarchical image features. These feature representations provide potential hypotheses regarding neural representation in the visual system. For the remainder of the chapter and we investigate this possibility by training a hierarchical model on natural images and assessing the features learned by the model in terms their theoretical properties and the ability to explain data from neurophysiology experiments.

3.3 Experiments

Here we use compositional training to learn a DBN model of a large collection of color natural image patches. Once trained, we analyze the features learned at each layer of the model by comparing them to results from previous neurophysiology. We also compare the learned features to the features learned using two related statistical learning models. The first related model is sparse coding (Olshausen and Field 1996). The second is the 2-layer hierarchical components model proposed by (Karklin and Lewicki 2005). We then assess the potential of the learned features for their ability at characterizing neural responses evoked in macaque V1 and V2 during natural vision.
Figure 3.2.5: Compositional training of a DBN hierarchical model. A, In the first stage, an RBM is trained on a set of inputs. If trained on pixel values, the RBM has Gaussian visible units $v^{(1)}$ and binary hidden unit $h^{(1)}$. B, Once the 1st layer RBM is trained, its parameters $W^{(1)}$ are frozen, and a 2nd layer RBM is trained on the binary feature representation of training images provided by the 1st layer RBM. The visible units $v^{(2)}$ and the hidden units $h^{(2)}$ in the second layer RBM are thus binary variables. C, The resulting deep belief network has undirected connections between the 1st and 2nd layer hidden units and directed connections between the 1st layer hidden units and the observed input variables.
Figure 3.3.1: Diagram of 3-layer hierarchical model of natural image patches. 300,000 20 × 20 color image patches were used to train the hierarchical model. Before training the patches were PC whitened, reducing the dimensionality from 1200 to 274. The model architecture was set to have 1024, 1600, and 2304 hidden units in layers 1, 2, and 3, respectively. The weights for each layer, $W^{(1)}$, $W^{(2)}$, and $W^{(3)}$, were optimized using compositional training to form a hierarchical representation of the image patches.
3.3.1 Deep belief network trained on natural image patches

We trained a DBN with 3 layers of hidden units on 300,000 $20 \times 20$ image patches that were randomly sampled from an in-house database of 4242 color photographs. The DBN model architecture had 1024 units in the first hidden layer, 1600 units in the second hidden layer, and 2034 units in the third hidden layer (Figure 3.3.1). Prior to training, the image patches were principle component (PC) whitened, retaining the 274 dimensions that accounted for 95% of the variance across the dataset (Hyvarinen and Oja 2000). Target regularization was used to induce sparsity and specificity amongst the hidden units and a target mean of $\mu = 0.05$ was used for both sparsity and specificity distributions. These target means were also used for all three layers. Stochastic gradient descent with minibatches of size 100 and contrastive divergence weight updates ($s = 1$) were used to train each RBM model layer with 5000 cycles through the training dataset.

3.3.1.1 Analysis of 1st layer features

Figure 3.3.2 shows the features learned in the first layer of the hierarchical model. To visualize the learned features, we mapped the model weights $W^{(1)}$ back into pixel space using the inverse of the PC whitening transform. A majority of the features are spatially-localized, oriented, frequency bandpass filters resembling 2-dimensional (2D) Gabor functions. The remaining features represent color information and generally exhibit lower spatial frequency and little orientation preference.

To summarize how the 1st layer features span spatial frequency, location, scale, and orientation in an image, we fit Gabor functions to the profiles of each feature (Figure 3.3.3). Using the parameters of the fit Gabor functions we can compare the features learned by the first layer of the DBN to related theoretical models and previous neurophysiology. The top row of Figure 3.3.4-(A-C) shows summary distributions of the Gabor parameters fit to the 1st layer features of the DBN. Spatial frequency and orientation cover a large proportion of available representation space, but there is a bias toward cardinal and diagonal (45 and 135 degrees) orientations. Similar orientation biases have been reported when analyzing the Fourier power spectrum of natural images (Coppola et al. 1998). Additionally, sensitivity toward these orientations have also been reported in perceptual (Appelle 1972) and neurophysiological (Leventhal and Schall 1983) studies. When compared to the features learned using sparse coding (3.3.4-D) or the features learned in the first layer of the hierarchical components model proposed by (Karklin and Lewicki 2005), spatial frequency is more clustered around the central region of the polar axis, indicating a bias toward lower spatial frequency.

The bias toward lower spatial frequency is more evident in Figure 3.3.4-B. This bias is likely due to the presence of both low frequency luminance and color features. The distribution of spatial frequencies more closely resembles that of physiology data for V1 neurons (De Valois et al. 1982; Figure 3.3.4-J) than does the distribution for the features learned with sparse coding (van Hateren and van der Schaaf 1998) or in the first layer of
Figure 3.3.2: Feature representation of natural image patches learned by the 1st layer of the DBN. Each square in the grid plots the learned feature for an individual model unit (1024 total). The features include spatially-localized, oriented, and frequency bandpass features. These features generally little color information. The luminance-only features are interspersed with low-frequency color features.
the hierarchical components model (Figure 3.3.4-E).

Investigating the phase of the Gabor fits can elucidate the degree of symmetry in the learned features: a phase offset of zero (defined relative to the center of the Gaussian modulation envelope) indicates even-symmetric features. A phase offset of $\pi/2$ indicates odd-symmetric features. Figure 3.3.4 demonstrates that most of the features learned in the first layer of the DBN are either even- or odd-symmetric. The remaining features are distributed uniformly between these two regimes. By comparison, the sparse coding model learns features that are more heavily biased toward being odd-symmetric, while the hierarchical components model learns features that have no bias toward symmetry whatsoever (3.3.4-(F,I)). The phase distribution observed in the first layer of the DBN are much closer to the distribution observed when fitting Gabors to real V1 simple cell RFs (Ringach 2002). The data from Ringach 2002 shows similar peaks at 0 and $\pi/2$ (3.3.4-K).

These results indicate that the 1st layer of the DBN learns valid spatio-chromatic features of natural images. The spatial properties of these features appear to be closely matched to the tuning of V1 simple cells in the macaque visual system (De Valois et al. 1982; Ringach 2002).

### 3.3.1.2 Analysis of 2nd layer features

The next question to ask is how the features learned in the 2nd layer of the DBN represent the image and whether or not they make any predictions regarding the tuning of complex V1 cells or cells in V2. The hidden units in the 2nd layer of the hierarchical
Figure 3.3.4: Comparing the properties of 1st layer features to other theoretical and physiological results. **Left column**, spatial frequency versus orientation on a polar plot; each dot represents a model unit for one of three theoretical models (indicated by text on far right). **Middle column**, histograms of peak spatial frequency in cycles/pixel for either Gabor functions fit to model unit features (rows 1-3, colored) or for tuned V1 simple cells (bottom row, black). **Right column**, histograms of phase parameters for Gabor functions fit to either model unit features (rows 1-3, colored) or to the receptive fields of real V1 simple cells (bottom row, black). Phase is determined from the center of the Gaussian modulation envelope fit; zero phase indicates even-symmetric features, while phase of π/2 indicates odd-symmetric features. **A-C**, Parameters for 1st layer of the DBN in the current study (green). **D-F**, Analogous results to A-C, but for features learned using the sparse coding algorithm (blue); figures adapted from Karklin 2007, *Fig 3.7*. **G-I**, Analogous results for A-C and D-F, but for the first layer of the hierarchical components model proposed by Karklin and Lewicki 2009 (red); figures also adapted from Karklin 2007, *Fig 3.7*. **J**, Spatial frequency tuning for V1 simple cells; adapted from De Valois et al. 1982, *Fig. 6*; data replotted in cycles/pixel while assuming 1 degree of visual angle per 20 x 20 image patch. **K**, Phase estimates for Gabor functions fit to real V1 cells; adapted from Ringach 2002, *Fig. 6B*. 
model learn dependencies amongst the units in the 1st layer. Figure 3.3.5 displays the features represented by 4 example units in the 2nd layer by plotting them in terms of the 1st layer features. In general, the 2nd layer units form the strongest connections to features in the 1st layer that have similar spatial location, orientation and scale. Figure 3.3.5-A (top, highlighted in orange) visualizes each of the example 2nd layer features as a linear combination of the subunits in the first layer with the largest connections to the corresponding 2nd layer unit (via $W^{(2)}$). Some of the 2nd layer units represent extended lines and contours that are formed as a superposition of the subunits in the 1st layer (3.3.5-A (iii-iv)). Others primarily represent simple color information (e.g. 3.3.5-A (ii)). However, many of the 2nd layer units represent nonlinear combinations of the 1st layer features. For example, the linear combinations of subunits depicted in 3.3.5-A(i) cancel out when viewed in the image space.

Units like that displayed in 3.3.5-A(i) can be better characterized by color-coding the visualization introduced in 3.3.3-C according to the connection strength to the corresponding 1st layer features. This visualization, shown in 3.3.5-B, shows that the unit in 3.3.5-A(i) has strong positive connections to those features in the 1st layer that are located in the lower right of the receptive field and that have multiple orientations. The same unit has strong negative connections to units with multiple orientations, located in the the upper left portion of the visual field. This visualization format also makes it more clear that spatial receptive field sizes are generally much larger in the 2nd layer units than in the 1st layer. Similar phenomenon are well known in neurophysiology when comparing the receptive field sizes of V1 and V2 neurons (Gattass et al. 1981).

Units in the 2nd layer of the DBN form a hierarchical image representation by nonlinearly combining 1st layer sub units. Thus the 2nd layer can potentially capture a diverse array of higher-order image structure, such as contours, curves, and textures. One could thus ask if there are any primary image dimensions that are represented by the 2nd layer. In order to summarize the primary dimensions represented in the 2nd layer, we perform nonlinear dimensionality reduction and clustering. Linear dimensionality reduction such as PC analysis proves to be inadequate for this problem, as a majority of the 2nd layer units tend to clump in the center of the subspace spanned by principle eigenvectors. This makes it difficult to separate more than a few distinct features difficult. Instead, we use t-distributed stochastic neighbor embedding (tSNE) (van der Maaten and Hinton 2008) to map the 2nd layer features nonlinearly into a 2D manifold. The results of the tSNE algorithm are shown in Figure 3.3.6-A. The features form many clusters in the low-dimensional manifold.

To identify specific sub classes within the 2nd layer, we performed hierarchical clustering on the embedded mapping of the 2nd layer features. To perform the clustering, we used Ward’s linkage function (Ward 1963), calculated on the Mahalanobis distance in the embedded space. The number of distinct clusters (18) was chosen using the variance ratio metric discussed in (Calinski and Harabasz 1974). Figure 3.3.6-B also shows the resulting cluster tree. Each point in 3.3.6-A is also color-coded with respect to these clusters. Figure 3.3.6-C shows the 2nd layer units in each of the clusters with the 5 largest $L_2$
Figure 3.3.5: Visualizing features represented by four example units in 2nd layer. A. Each of the four columns (i-iv) visualizes the features learned by an individual unit in the 2nd layer of the DBN hierarchical model. At the top row of each column, outlined in orange, is the linear combination of the 1st layer features that have the strongest positive (left, indicated by “+” at bottom) and negative (right, indicated by “-” at bottom) connections to the corresponding 2nd layer unit. The 1st layer subunits contributing to the sum are visualized below each orange outline. Each 2nd layer unit is connected to subunits in the 1st layer that represent similar spatial location, scale, orientation, and color information about the visual input. The example 2nd layer unit in the first column (i) represents a nonlinear combination of the 1st layer subunits. This is indicated by the fact that the linear sum of the 1st layer units cancels out in the image-level representation. B. An alternative visualization of the example units plots the parameters of Gabor functions fit to each unit in the 1st layer as a line segment. The line segments are colored with respect to their connection weight to the 2nd layer (columns of $W^{(2)}$); red is positive, blue negative, and gray zero weights (colormap at bottom left). This visualization gives a much richer characterization of how each 2nd layer unit represents spatial location, frequency, and orientation information. The 2nd layer units capture high-level image structure including coherent, spatially-localized orientation energy (e.g. iii and iv), spatially localized features of multiple orientations (e.g. i), and full-field color (e.g. ii).
Figure 3.3.6: *Summarizing the distribution of 2nd layer features.* **A,** the feature weights $W^{(2)}$ for the 2nd layer are embedded in a 2D space using t-SNE. Each dot represents an individual feature. The features tend to cluster in localized regions of the reduced space. **B,** Results from hierarchical clustering of features in the reduced space. The leaf nodes of the cluster tree are colored separately for each cluster. These colors are used to color the points in **A.** **C,** Each row within a subpanel plots the features with the 5 largest norms within an individual cluster identified in **B.** The cluster identity is indicated by the number to the left of each row and the color of the box surrounding each feature. The features for each cluster member are plotted in the same manner as Figure 3.3.5-B. Thus red lines indicate positive, blue lines negative, and gray lines zero weights onto the 1st layer features. The clusters tend to capture 3 general classes of information: 1) those that define extended borders and contours by way of coherent orientation energy (e.g. clusters 1 and 7); 2) those that represent full-field texture and texture boundaries (e.g. clusters 9 and 16); and 3) those that represent low-frequency color information (e.g. clusters 3 and 4).
norms using the same method as in 3.3.5-B. The clustering analysis suggests that there are indeed a set of primary dimensions represented in the 2nd layer. There appear to be multiple clusters (e.g., clusters 1, 7, and 8) that are excited by localized, coherent orientation energy. Others (e.g. cluster 16) appear to represent more full-field, coherent orientation at high spatial frequencies (indicated by short red lines). Other clusters (e.g. cluster 3) exhibit less coherency in spatial location, frequency, and orientation.

Though the visualization in Figure 3.3.6-C is useful for characterizing the general spatial distribution of frequency and orientation parameters, it is less helpful for elucidating how other information like color is represented by the 2nd layer units. Figure 3.3.7 shows some specific examples from 3.3.6-C, but also includes the units from the first layer with the largest excitatory and inhibitory connections. Including this additional information with Figure 3.3.5 provides a better intuition for the three general categories of units in the 2nd layer: 1) those that define extended borders and contours by way of coherent orientation energy (e.g. clusters 1 and 7); 2) those that represent full-field texture and texture boundaries (e.g. clusters 9 and 16); and 3) those that represent low-frequency color information (e.g. clusters 3 and 4).

For example, the excitatory units in cluster 7 combine multiple horizontally-oriented sub units across the upper portion of the receptive field in order to represent extended, horizontal luminance contours. While units in cluster 7 are excited by a horizontal contours in the upper receptive field they are suppressed by low frequency color. Thus the units in cluster 7 can be interpreted as performing a contour-color discrimination. The units in cluster 9 appear to represent texture-color discrimination, as they are excited by high-frequency, vertically-oriented texture and inhibited by low-frequency color contrast. Cluster 16 also performs texture-color discrimination, but is excited by orientation energy at both 45 and 135 degrees. Cluster 3 captures low-frequency color contrast information while cluster 4 appears to represent color opponency with little to no contrast information.

In principle, all of these mechanisms could be used to determine the boundaries of objects and surfaces using texture, contour, and/or color information. It is commonly believed that the intermediate stages of the visual system (i.e. areas V2 and V4) are involved in detecting illusory and texture-defined borders (von der Heydt et al. 1984; von der Heydt and Peterhans 1989) and border ownership (Zhou et al. 2000). If the features learned in the 2nd layer are able to account for neural function in V2, the border-processing hypothesis would be supported at least in part. More importantly, it would suggest that border processing in V2 can be accounted for by the high-order statistics of natural scenes. Later in this chapter we demonstrate that the features learned in the 2nd layer of the hierarchical do indeed account for neural responses V2.

### 3.3.1.3 Analysis of 3rd layer features

To get an idea of the basic image dimensions represented in the 3rd layer of the hierarchical model we can perform an analogous clustering analysis to that performed on the 2nd layer features. Specifically, we performed hierarchical clustering on the 3rd
Figure 3.3.7: *Information encoded in 6 example clusters defined in 3.3.6.* Each subpanel visualizes the information represented by the hidden units with the top 5 $L_2$ norms within an example cluster. The color bounding subpanel corresponds to the cluster colors defined in 3.3.6-B. Individual units with the displayed in 2 ways: first, along the top border of each subpanel, the unit weights onto all 1st layer subunits are plotted in the same line-based format as 3.3.5-B; second, the top five most excitatory and inhibitory subunits are displayed along the columns (left and right, respectively) directly under each line-based visualization. Clusters 1 and 7 perform contour-color discrimination, while clusters 3 and 4 represent color contrast and color opponency. Clusters 9 and 16 perform texture-color discrimination.
layer feature weights when embedded in a 2D subspace. We then analyzed the features within each cluster. Figure 3.3.8 shows the results of this analysis. 3.3.8-A displays the 2D representation of the features in the 3rd layer. The features form tight islands and manifolds in the 2D space. We identified 17 distinct clusters within this representation of the data. The units with the 5 largest norms in each of the 17 clusters are displayed in 3.3.8-C. In principle, we would like to interpret the information conveyed in the 3rd layer in terms of the visual input. To do so we visualize the projection of the 3rd layer features onto the 1st layer subunits by filtering the 3rd layer features through the connections between the 1st and 2nd layer. Specifically, in 3.3.8-C we visualize the following weighting on the 1st layer visual features:

\[
\tilde{W} = W^{(2)}W^{(3)},
\]

where \(\tilde{W} \in \mathbb{R}^{1024 \times 2304}\). This visualization suggests that, in terms of capturing dependencies between the 1st layer units, the 3rd layer features show substantial overlap with those learned in the 2nd layer. As in layer 2, there are clusters of units that represent orientation energy boundaries (e.g. clusters 1, 2, and 9), contour-color boundaries (e.g. clusters 11, 14), and texture-color boundaries (e.g. clusters 4, 5, and 6). However, in contrast to the 2nd layer, there few units that primarily code for color. Also, the features in the 3rd layer generally represent texture boundaries with greater spatial precision than in the 2nd layer. This suggests that the 3rd layer could be involved in more fine-tuned boundary detection, though there are other possible explanations (see Discussion).

It is unclear if visualizing the 3rd layer features in terms of the 1st layer features is the optimal way to characterize the information conveyed in the 3rd layer. The 1st layer features are only represented in the 3rd layer implicitly through computations performed in the 2nd layer. Also, by linearly projecting the 3rd layer features through the 2nd layer connections to the first layer, our interpretations may be heavily affected by the structure of the 2nd layer features. This could indeed be the case, given that our visualization of the 3rd layer features suggests a substantial overlap with the features learned in the 2nd layer.\(^3\) That said, we have found that interpreting the 3rd layer features is not straight-forward when visualization is performed in terms of 2nd layer features. Future work on visualizing nonlinear, hierarchical relationships is key to fully understanding the information conveyed in these deep models.

3.3.1.4 Properties of hierarchical representation

The hierarchical model used in this work learns a distributed representation of natural images by having each layer learn the dependencies of hidden units in the previous layer. Thus, we should expect that the units in higher layers of the hierarchy to represent more

\(^3\)Though later we show that the feature representations in the 2nd and 3rd layers provide similar predictions of responses in area V2, suggesting that both layers may indeed be representing very similar information about the visual input.
Figure 3.3.8: Summarizing the distribution of 3rd layer features. **A**, the feature weights $W^{(3)}$ for the 3rd layer are embedded in a 2D space using t-SNE, **B**, Hierarchical clustering of features in the reduced space. **C**, Features with the top 5 largest norms within each cluster are displayed. The color of the clusters in A and associated boxes around the features in C correspond to the clustering in B. Formatting follows that of Figure 3.3.6. Compared to features learned in the 2nd layer, the units in the 3rd layer tend to represent more defined contour and texture borders.
explicit information about the visual input than units in the lower layers (Barlow 1972). A more explicit code suggests that the features in the higher layers of the hierarchy will occupy more distinct regions in their respective feature spaces than do features in the lower layers. To test this hypothesis we first visualize the features learned in each layer of the hierarchical model by mapping them into an associated 2D embedding space. We then observe if the features in each layer occupy increasingly sparse regions of their respective embedded spaces. Figure 3.3.9 displays the features learned in each layer of the hierarchical model, plotted in their associated 2D embedding space calculated using tSNE. As predicted, the features in the 3rd layer occupy tight clusters and manifolds within their respective embedded space. In contrast, the features in the 1st layer are distributed diffusely throughout their respective embedded space. The features in the 2nd layer form both tight clusters, as well as large swaths of densely packed features. This indicates, at least qualitatively, that features represent more explicit information about the input with an ascent in the hierarchy.

Having a feature hierarchy that represents increasingly explicit information about the visual input would also suggest that it requires few units in the higher layers to represent rich patterns of activation in the lower layers. Figure 3.3.9 demonstrates this phenomena by plotting the weights from individual units in the 3rd layer, back-projected through the feature weights in the layers below. As a visual aid, the weight projections are plotted onto the 2D feature embeddings calculated for each layer. The visualization shows that individual units in the 3rd layer are associated with distinct patterns of connectivity to 2nd layer. In general, only a few, localized regions in the 2nd layer have positive connections to a 3rd layer unit. Plotting the projection of the 3rd layer weights into the 1st layer (i.e. $\tilde{W}$ defined in 3.3.1) shows that each 3rd layer unit is also associated with widespread, structured connections to the features in the 1st layer. This demonstrates how individual units in the 3rd layer capture complex structure in the activations of the 1st layer units, and thus the visual input.

### 3.3.2 Modeling neural responses based on learned hierarchical features

In the previous section we show that a compositionally-trained DBN learns a hierarchical representation of natural images. Motivated by the efficient coding principle, we propose that the features learned by the hierarchical model provide promising predictions for how neurons in visual system represent high-order image structure. If the learned features can accurately account for activity of real neurons, it suggests that hierarchical model has uncovered valid forms of neural representation. Accordingly, we test the accuracy of linearized models based on the learned hierarchical features to predict neural activity evoked in the primate visual system during natural vision.
Figure 3.3.9: Visualizing the global connection properties of the learned hierarchical representation. The left 3 columns plot the feature weights for layer 3 to layer 1. For each layer, the features are embedded in a 2D subspace calculated using tSNE. Each dot represents an individual feature. Each outlined row displays how an individual unit in the 3rd layer (indicated by the yellow dots in the left column) connects to the layers below. The colors of the dots in the second and third columns represent the linear back-projection of the connection weights from the corresponding 3rd layer unit highlighted at the left. Red indicates excitatory, blue inhibitory, and gray zero weights (colorbar in lower right). The far right column plots the projected weights onto the features in the 1st layer (3rd column from right) in the format used in 3.3.5-B. With an increase in the hierarchy, the learned features occupy increasingly distinct local regions of their respective embedded space. This suggests that the features learned at each layer become increasingly explicit in their representation of the input. Higher layers of the hierarchy also represent rich patterns of connectivity to the lower layers.
Neurophysiology Experiment

Color, full-field movies of natural environments were presented at a frame rate of 60 Hz to an awake behaving macaque trained to fixate at the center of the screen. Electrophysiological responses were measured using a $10 \times 10$ recording array implanted chronically over the V1/V2 boundary (data originally presented in Oliver et al. 2012). The cortical representation of the visual field at the array location was approximately one degree below the fovea. Data was obtained from 96 of the available 100 channels. The resulting signals were spike sorted to isolate individual neurons, giving data for 367 cells (281 from V1, 86 from V2). Spike rate data were calculated for each cell by binning spike trains at the stimulus frame rate (a resolution of 16.67ms). Multiple trials of stimuli were repeated and spike rates were averaged over repeats to obtain a mean firing rate for each neuron and stimulus frame. The number of stimulus repeats ranged from 1 to 8 and the number of stimulus-response pairs ranged from 20 thousand to 1.8 million, depending on the duration that a cell could be recorded.

Linearized Encoding Model Parameter Estimation

Linearized encoding models based on the features learned in each layer of the hierarchical model were fit individually for each of the recorded neurons (Figure 3.3.11).
Before fitting the linearized models, the center of the CRF was identified for each neuron by first estimating a reverse correlation model to the stimuli located within the central 48 × 48 pixels (2.58 degrees, at 18.6 pixels/degree) of fixation. The estimates of the CRF centers were then fine-tuned by analyzing the reverse correlation estimates. The stimuli within 20 × 20 pixels (1.08 degrees) surrounding each CRF center were then cropped and transformed into a feature space representing the features learned by each layer of the DBN.

The stimuli were whitened using the PC whitening transformation calculated from image patches used to train the DBN (see Section 3.3.1). Three classes of features based on each layer of the hierarchical model were calculated as the feed-forward hidden unit activations at each layer when driven by the whitened stimuli. To estimate the encoding model parameters, these activations were regressed onto the neural spike rates evoked by the stimuli (Figure 3.3.11). To account for the temporal response properties of each neuron, the feature activations within 8 frames (133 ms) preceding each response were included in the model regressors. Thus for models based on the 1st, 2nd, and 3rd layers of the DBN, the size of the estimated parameter space was 9,216, 14,400, and 20,736, respectively. Once the parameters were fit, the accuracy of the encoding models was evaluated on a novel testing set of stimulus-response pairs. Model accuracy was quantified as the correlation coefficient between the actual responses in the testing set and those predicted by the model.

To facilitate encoding model interpretation and to improve generalization to new data, it is generally useful to regularize the parameter estimation procedure. For this work we use the Threshold Gradient Descent (TGD) algorithm (Friedman and Popescu 2004) to perform regularization and variable selection. The TGD algorithm works in the similar manner to the standard gradient descent algorithm but it also includes a thresholding function that zeros out gradient contributions when they fall below a user-defined threshold. The parameter updates for the TGD algorithm are as follows:

\[
\Delta_{\text{TGD}} \theta = \epsilon \left( \frac{\partial \mathcal{L}(r; \theta)}{\partial \theta} \otimes M_{\tau} \left[ \frac{\partial \mathcal{L}(r; \theta)}{\partial \theta} \right] \right)
\]

where \( M_{\tau} [z_k] \) = \( \begin{cases} 1 & \text{if } |z_k| > \tau \max |z| \\ 0 & \text{otherwise} \end{cases} \) .

Here, \( \mathcal{L}(r; \theta) \) is the likelihood function for the modeled responses and \( \epsilon \) is a learning rate. The term \( M_{\tau} [z_k] \) is a masking function that indicates when the absolute value of the gradient for the \( k \)-th parameter exceeds a value proportional to largest parameter gradient; \( \otimes \) indicates element-wise product. For \( \tau = 0 \) the TGD algorithm is equivalent to standard gradient descent, providing a dense distribution of parameter values. Solutions for \( \tau = 1 \) are comparable to those obtained by \( L_1 \) regularization (Friedman et al. 2010) and provide a sparse distribution of parameter values. For \( 0 < \tau < 1 \) the algorithm gives solutions that are qualitatively similar to the elastic net (Friedman et al. 2010), providing a trade-off
Figure 3.3.11: Constructing linearized encoding models of V1 and V2 neural responses based on learned hierarchical image features. Using the general framework described in Section 2.5, we estimated models of neural responses evoked in visual areas V1 and V2 of the macaque. B, Before fitting the encoding models, natural image patches were modeled using a hierarchical model that learned three hierarchical layers of visual features. A, Given the trained hierarchical model, we then fit three separate predictive encoding models for each neuron. Each of the three models was based the features learned at each of the three layers of the hierarchical model. The models were estimated by mapping the hidden unit activation probabilities for a set of stimulus scenes onto the neural responses evoked by the same stimuli. The activation probabilities act as a nonlinear transformation of the stimulus into a feature space learned by the hierarchical model. The encoding models thus capture how neural responses are accounted for in terms of the hierarchical image features.
between variable selection (sparseness) and smoothness in parameter estimates. For this work we used the exponential distribution likelihood function to model the responses, a learning rate of $\epsilon = 0.02$, and a gradient threshold of $\tau = 0.7$. The value of $\tau$ was chosen to ensure relatively strong variable selection, while encouraging some variance to be shared among selected variables.

In order to improve generalization performance further, we also used 10-fold cross-validation with early stopping criterion and model averaging. Specifically, we partitioned the training data into 10 sections then fit the model parameters on 9 out the 10 sections, using the remaining section as an early stopping set. When prediction accuracy on the early stopping set (measured as correlation coefficient on actual and predicted responses) ceased to decrease, the TGD algorithm was stopped and the resulting parameters were used as a sample of potential solutions. This procedure was repeated by successively using each of the 10 sections as an early stopping set, resulting in 10 separate model parameter solutions. The average of these parameters was used as final model parameters. No cross validation was performed to select learning rates and gradient thresholds, but we found that the selected parameters provided an adequate trade-off between generalization and model interpretability.

### 3.3.2.1 Encoding models based on 1st layer features

Model performance for encoding models based on the first layer of features is shown in Figure 3.3.12. Figure 3.3.12-A plots model accuracy for each of the neurons as a function of the number of stimulus-response pairs used to estimate the EM parameters. There is a slight positive trend in the data (correlation: $r = 0.15$), indicating that increasing the training set size improves model performance, as expected. However, it also suggests that a large number of the neurons had sufficient signal-to-noise ratio to estimate accurate models with modest amounts of training data. On average, V1 neurons (Figure 3.3.12, magenta points) are predicted more accurately (mean correlation: $\bar{r}_{V2} = 0.24 \pm 0.01$) than V2 neurons (cyan points; mean correlation: $\bar{r}_{V1} = 0.14 \pm 0.01$). This suggests that the first layer features provide a more plausible form of representation for V1 neurons than for V2 neurons.

The results above support the notion that linearized models based on statistical features learned from natural scenes provide a sound method for predicting the responses of neurons in V1 and V2. However, we also wanted to compare the prediction performance of the linearized models against that of other standard modeling approaches. To do so, we also estimated the spike triggered average (STA) and spike triggered covariance (STC) models for all of the neurons as described in (Touryan et al. 2005).\(^4\)

The STA and STC models were fit using the same number of stimulus delays as was

\(^4\)However, in contrast to (Touryan et al. 2005), which presented random natural image sequences, the current experiment presented natural movies, which have substantial correlation in both space and time. Thus we whitened the stimulus in both the space and time using a Fourier method analogous to that described in (Van Hateren 1993).
Figure 3.3.12: Accuracy of linearized encoding model based on 1st layer features. Each point represents results for a single neuron. Magenta points represent neurons recorded from V1, cyan from V2. A, The horizontal axis gives the number of training points used to estimate the encoding model. The vertical axis gives the encoding model accuracy, calculated as the correlation coefficient between actual and predicted responses to a testing set of stimuli. Model accuracy is slightly correlated with the number of training data points \( r = 0.15 \). The dashed horizontal lines in the background plot the mean +/- one standard error from the mean of model accuracy for V1 and V2, respectively. On average, V1 neurons are better predicted by the 1st layer features than are V2 neurons. B, Comparison of encoding model accuracy for the linearized models with 1024 features versus the accuracy of an STA model. The dashed diagonal black line indicates equivalent prediction accuracy for both models; points above the black line indicate that the linearized model provides more accurate predictions for the corresponding neurons. The linearized models outperform the STA models for a majority of the recorded neurons. C, To demonstrate the efficacy of regularization, encoding model accuracy for linearized models with 1024 versus 784 features is compared. Encoding model accuracy is highly correlated (correlation of \( r = 0.97 \)), indicating that including extra parameters does not lead to overfitting.
used for estimating the linearized models (0-8 frames). For the STA, this resulted a model with 10,800 parameters. Even when collapsing the stimulus across color channels, the size of the parameter space for the STC model was much larger, requiring over 1.4 million parameters. For nearly all neurons, the STA model provided equal or better prediction of neural activity than the STC estimate (This could be due to both a large proportion of simple cells in the population and/or the large parameter space required to estimate the STC model). Therefore we present the STA results only. The accuracies for the two classes of models is compared in 3.3.12-B. The responses in most of the cells (86.1% of total, 86.8% of V1, 84.9% of V2) were predicted more accurately by the linearized models. These results provide convincing evidence that the proposed modeling framework based on linearized modeling and statistical feature learning is a potent alternative to standard methods for modeling neural representation.

To ensure that the TGD fitting routine was effective at preventing overfitting, we compare model accuracy of the 1024-feature model to that of a smaller model based on 784 features (roughly 25% fewer parameters). The 784-feature model was based on an RBM with 784 hidden units that was trained on the same 300,000 image patches. If the encoding models are inadequately regularized then the model with fewer parameters should generalize to the testing set better than the larger model. The comparison of prediction accuracy for the two models classes is shown in 3.3.12-C. The predictions are highly correlated ($r = 0.97$). This indicates that the TGD algorithm is indeed providing adequate regularization.

### 3.3.2.2 Interpreting 1st layer model weights

The linearized encoding models used in this work predict neural responses as linear combination of the features learned by the hierarchical model. It is thus straightforward to interpret the fit models in terms of these features. Figure 3.3.13 displays the encoding model weights on the 1st layer features fit for three example cells from V1 and V2 (6 total). Though neural responses are generally not well characterized by a single feature learned in the 1st layer, the learned features do provide an effective basis for capturing a wide range of simple receptive field properties.

Many of the cells primarily represent color information and show little selectivity for luminance contrast. For example, V1 cell 301 represents blue-yellow color opponency while V2 cell 46 represents red-green color opponency. Other cells represent a mixture of color and luminance tuning (e.g. cells 222 and 323 recorded from V2). Other cells, like 121 and 190 recorded from V1, primarily represent luminance contrast. Though these results do not extend beyond what is known about V1 simple cells, these findings are important mainly because the models were fit using features derived in a fully data-driven manner. An important next step in the current work is to analyze how features from the higher layers of the hierarchical model perform at modeling neurons in V1 and V2.
Figure 3.3.13: Visualization of 1st layer encoding model weights for six example neurons.  
A, Models estimated for V1 neurons. Each of the three columns visualizes the encoding model weights fit to the responses of a single V1 cell. The title of each column displays the cell number, the associated visual area in parentheses, and the model correlation coefficient (Pearson’s $r$) calculated on the testing set. The stem plots represent the fit weights to each of the first layer features. Thumbnails of the 1st layer features with the strongest weights are plotted at end of their corresponding stem lines. The two images to the right of each stem plot display the weighted linear sum of the top excitatory (top) and inhibitory (bottom) features for the neuron. B, Model estimates for V2 neurons. Formatting is the same as in A.
3.3.2.3 Encoding models based on 2nd layer features

Figure 3.3.14 displays the prediction accuracy for encoding models based on the features learned in the 2nd layer of the hierarchical model. Just as for encoding models based on the 1st layer features, the amount of training data positively affects model accuracy, but to an even larger degree (3.3.14-A; correlation between model accuracy and training set size is \( r = 0.25 \)). The 2nd layer features provide substantially more accurate predictions for V2 neural responses than the 1st layer features. This is indicated in Figure 3.3.14-A, B, which demonstrates a large shift in the mean prediction accuracy across the population of V2 neurons. The mean prediction accuracy shifts from \( \bar{r}_{V2} = 0.14 \pm 0.01 \) for the 1st layer features to \( \bar{r}_{V2} = 0.36 \pm 0.02 \) for the 2nd layer features. There is also a substantial improvement in population accuracy for many V1 cells, shifting the mean accuracy from \( \bar{r}_{V1} = 0.24 \pm 0.01 \) to \( \bar{r}_{V1} = 0.32 \pm 0.01 \). The large number of cyan points above the black dashed line indicate that the 2nd layer models are dramatically outperforming the 1st layer models for most of the V2 cells (92.8%). While many V1 cells are better predicted by the 1st layer model, a majority of the V1 cells (68.7%) are better predicted by the 2nd layer model. Presumably these V1 cells represent higher-order image structure than what can be captured by the linear features in the 1st layer. Taken in conjunction, these results suggest that the features learned in the 2nd layer of the hierarchical model provide a more valid form of representation for V2 neurons and a subset of nonlinear V1 neurons.

3.3.2.4 Interpreting 2nd layer encoding model weights

Encoding models based on the 2nd layer features provide more accurate responses of V2 neurons than models based on the 1st layer, but what is the relevant information represented in the 2nd layer that results in such an improvement? One can address this question by interpreting the 2nd layer encoding model weights fit for V2 neurons. Figures 3.3.15-3.3.16 display the encoding model weights fit for two example V2 neurons.

For the first example cell, shown in 3.3.15, the 2nd layer model gives a 4-fold improvement in explained response variance when compared to the model based on the 1st layer features (from 12.3% to 51.8%). Analyzing the fit weights, the cell appears to be tuned for luminance contrast at multiple orientations, particularly 0 and 90 degrees. The cell exhibits selectivity for a multiple spatial frequencies, ranging from approximately one to two cycles per spatial receptive field. Whereas excitatory features for the cell are located just to the right of the vertical meridian, inhibitory features are located in the upper left of the receptive field. These inhibitory features span a larger range of spatial frequencies (approximately 2-4 cycles per receptive field). The cell shows very little color selectivity.

Cell 130 also shows a dramatic improvement in the amount of response variance explained when moving from the 1st layer model to the 2nd layer model (3.6% to 41.0%). This cell is more narrowly tuned for both orientation and spatial frequency, preferring features oriented at approximately 135 degrees and spatial frequencies around 3 cycles
Figure 3.3.14: Accuracy of linearized encoding model based on 2nd layer features. Formatting is the same used in Figure 3.3.12. **A**, Plots the encoding model accuracy of 2nd layer models versus the number of training data points used to estimate the model parameters. The dashed horizontal lines in the background represent the mean prediction accuracy across the population of V1 and V2 neurons +/- one standard error from the mean. Model accuracy is slightly correlated with the number of training data points ($r = 0.25$). **B**, The results from 3.3.12-A are replotted to facilitate comparison of the population model accuracy for 1st and 2nd layer models. The 2nd layer models dramatically improve prediction for V2 cells. **C**, A direct comparison of encoding model accuracy for 1st (horizontal axis) and 2nd layer (vertical axis) models. The dashed diagonal black line indicates equivalent prediction accuracy. Points above the black dashed line indicate that the 2nd layer features provides a more accurate account of the responses of the corresponding neurons. The 2nd layer models outperform the linearized models for a majority of the neurons in V2.
Figure 3.3.15: Visualization of 2nd layer encoding model weights for an example V2 neuron. The title displays the cell number, visual area, and the correlation coefficient on the testing set (Pearson’s $r$). To demonstrate improvement of model accuracy based on the 2nd layer, the correlation for the same cell based on the 1st layer encoding models is indicated in parentheses. The plots in the top half of the figure display the 2nd layer features with the five largest positive encoding model weights. The top row of visualizations are plotted in the same format as 3.3.5-B. The row immediately below visualizes the top excitatory and inhibitory subunits in the 1st layer for the same 5 2nd layer hidden units. The color of the frames around each visualization represents the relative encoding model weight fit to neural responses. The magnitude and sign of the encoding model weights are indicated by the color bar in the center. The plots in the bottom half of the figure display the five 2nd layer features with the largest inhibitory encoding model weights. Responses are evoked in this cell by multiple orientations, particularly 0 and 90 degrees. The cell is selective for a range of spatial frequencies ranging from 1-2 cycles/receptive field.
Figure 3.3.16: Visualization of 2nd layer encoding model weights an example V2 neuron. Formatting is the same as in Figure 3.3.15. The cell is predicted with greater accuracy by the 2nd layer model than the 1st layer model. Responses are evoked in this cell by the presence of contours at the center of the receptive field, oriented at 135 degrees. Responses are suppressed by horizontal contours at the top of the visual field, and orientations at 45 degrees in the bottom right of the visual field.
per receptive field. The cell is inhibited by a number of features including horizontal features in the upper visual field and features oriented at 45 degrees in the upper left and lower right of the visual field. The inhibitory features also span a large range of spatial frequencies, from 1.5 to 6 cycles per receptive field. Generally cells are characterized primarily on the features that drive the cell, but this example shows that inhibitory subfields can be quite complex and are also important for predicting cell behavior. Similar results supporting the importance of inhibitory subfields are reported in (Willmore et al. 2010).

3.3.2.5 Encoding models based on 3rd layer features

Figure 3.3.17 displays the prediction performance for models based on the 3rd layer features. Similar to the 2nd layer models, the 3rd layer models predict the responses of V2 cells with a greater accuracy than V1 cells ($\bar{r}_{V1} = 0.29 \pm 0.01$; $\bar{r}_{V2} = 0.36 \pm 0.02$). When compared to the 2nd layer models, model performance for the 3rd layer models is nearly identical for V2 neurons (56.7% of V2 cells are better predicted by the 2nd layer model) but slightly worse for V1 neurons (74.9% of V1 cells are better predicted by the 2nd layer model). This suggests that the features learned in the 3rd layer of the DBN likely represents overlapping information to 2nd layer that is useful for characterizing V2 responses. Given the similarity between the 2nd and 3rd layers observed in Figures 3.3.6 and 3.3.8, this finding is not surprising. What is interesting is the discrepancy between predictions for V1 cells. The features learned in the 2nd layer that are useful for characterizing the responses of many V1 cells appear to be absent from the representation learned in the 3rd layer. This suggests that the hierarchical representation learned by successive layers of the model is gradually abstracting away from the pixel-level representations. An interesting prediction then is that the 3rd layer may provide a better proxy for the next stage of visual processing, V4, than does the 2nd layer. However, we leave this prediction to be tested in future work.

3.4 Discussion

In this chapter we have shown that a simple, objective framework based on hierarchical feature learning and linearized system identification can be used to study how image structure is represented in visual areas V1 and V2. The feature learning component of the framework recovers features that share a number of linear and nonlinear properties with V1 and V2 neurons. The learned features further suggest that the stimulus dimensions that V2 neurons might be encoding, namely extended contours, texture boundaries, and color information. The linearized modeling component of the framework is capable of constructing from these learned features accurate and interpretable predictive models of V1 and V2 neural responses evoked during natural vision. Most importantly, this approach is fully data-driven. Outside of enforcing a few biologically- and theoretically-motivated constraints on the feature learning process (i.e. sparsity and specificity and
Figure 3.3.17: Accuracy of linearized encoding model based on 3rd layer features. Formatting is analogous to 3.3.12. A, Plots the encoding model accuracy of 3rd layer models versus the number of training data points used to estimate the model parameters. The dashed horizontal lines in the background represent the mean prediction accuracy across the population of V1 and V2 neurons +/- one standard error from the mean. Model accuracy is slightly correlated with the number of training data points ($r = 0.21$). B, The results from 3.3.14-A replotted to make the comparison in population model accuracy more clear. There is very little difference in population accuracy between the two classes of models for area V2 neurons. The 2nd layer models provide slightly more accurate models for V1. C, A direct comparison of encoding model accuracy for 3rd (horizontal axis) and 2nd layer (vertical axis) models. The dashed diagonal black line indicates equivalent prediction accuracy; points above the black line indicate that the 2nd layer provides more accurate predictions for the corresponding neurons. The prediction performance is highly correlated, particularly for V2 neurons. The 2nd layer models slightly outperforms the 3rd layer models for a majority of the neurons in V1.
model network size), the process relies on little input from the experimenter.

This work shares a number of similarities with previous theoretical and modeling endeavors. As in other feature learning approaches like sparse coding and ICA, we train a statistical model to learn a set of sparsely-distributed latent variables that capture important features in natural images. Accordingly, the first layer of the DBN learns similar visual features as reported in these previous works. Namely the features are localized, oriented, and frequency-bandpass filters. However, the features learned in the current work show biases toward low spatial frequency as well as cardinal and diagonal orientations. These biases are more indicative of neurophysiology findings than the biases observed in sparse coding. A potential explanation for the low spatial frequency bias is the inclusion of color information, which tends to vary slowly across space in natural scenes. Also, the RBM models used here to construct the DBN incorporate a specificity constraint that influences each hidden unit to have sparsely-distributed activations across many image patches. In principle, this additional constraint could be responsible for the biases in the features learned by the RBM, but it is still unclear. A more systematic investigation of the specificity parameter is necessary to answer this question.

The hierarchical model in the current work is closely related to the variance and covariance components models introduced by Karklin and Lewicki 2005 2009. Like those models, we attempt to capture the dependencies of latent variable activities in the 1st layer using a second layer of representation. However, rather than modeling high-order dependencies implicitly by assuming that the 2nd layer places a generative prior on the variances (or covariances) of the latent activities in the 1st layer, we attempt to model the dependencies directly.

The features learned in the 2nd layer of our model share a number of properties with the hierarchical features learned with Karklin’s models: increased receptive field size, and the representation of contours, curves, and textures. However there are also some key differences between the hierarchical features learned by the two models. Karklin reports a number of 2nd layer features that have multiple sub-bands of modulation across the visual field, akin to Gabor-like modulations of the 1st layer features activations across space. Our results fail to replicate this phenomena. In general the 2nd order features learned by the DBN show at most a single excitatory sub band across the 1st layer features, and at most 2 inhibitory sub bands (for even symmetric units).

Another difference between the current work and that of Karklin and Lewicki 2005 2009 is that we learn even higher-order dependencies through a 3rd layer of representation. The model units in the 3rd layer appear to be more precise at defining texture boundaries than in the 2nd layer. Like in Karklin 2007, we use the activities of the latent variables as features for predictive models of V2 neurons. Similar to their results, we find that the hidden units in our model provide an accurate account of neural responses in V2. However, this work models data from both V1 and V2 simultaneously based on multiple

---

5It turns out that the hierarchical model that results from the compositional training procedure does have directed, generative connections in the bottom two layers and undirected connections between the 2nd and 3rd layers. However, the parameters are not estimated assuming this model structure.
stages of hierarchical image representation. This allowed us to directly show how the model accuracy for these two successive stages of the visual system correspond with successive stages in a hierarchical representation learned from natural visual inputs.

In terms of feature learning, the current work is most similar to that of Lee et al. 2008. In that work, they learned a 2-layer hierarchical model of natural scenes via compositional training of RBM model layers. In a similar fashion, the hidden units in each RBM layer were influenced to have a sparse distribution. However, a much coarser regularization method was used that adds a global bias aimed at making the average probability of activation across the population low. Lee et al. 2008 report that the hidden units in the 2nd layer represent contours, corners, and junctions, and establish tuning for these features by adapting a protocol previously used in neurophysiology to study V2 neurons (Ito and Komatsu 2004). However, these features were never tested on their ability to predict real neural responses. Lee et al. 2008 also do not report the presence of features in the 2nd layer that represent global texture or texture boundaries. This could be due in part to the fact that they learned features from grayscale rather than color images, or that their regularization scheme did not enforce specificity in individual units.

Recent work based on presenting natural textures (Freeman et al. 2013) has shown that statistical dependencies beyond 2nd order are represented in V2 but not V1. Specifically, two textures with the same power spectra elicit similar responses in V1 despite their phase spectra. However, V2 appears to be sensitive to the extended coherent structure observed in natural images, and is thus sensitive to the phase spectra of natural scenes. The results of the current work supports these findings. The features learned in the first layer can be interpreted as a local Fourier representation of the image and thus cannot capture extended coherent structure. Accordingly, these features lead to accurate models of V1 neurons, but not cells in V2. Instead, the features learned in the 2nd layer, which can capture extended, coherent structure provide much more accurate models of V2 neurons. A number of the features learned in the 2nd layer of the current hierarchical model are full-field, coherent textures. This suggests that by presenting textures, the work of Freeman et al. 2013 could be probing one many potential forms of high-order image structure that are represented in V2.

An interesting finding of the current work is that features learned in the 3rd layer of the hierarchical model provide accurate encoding models of V2 responses, but do not improve performance beyond the 2nd layer encoding models. Comparing the features learned in the 2nd and 3rd layers reveals a high degree overlap in general representation (i.e. full field texture, texture boundaries, contour boundaries), though the 3rd layer units appear to be more finely-tuned for spatial location and orientation coherency. This overlap could be due to a number of underlying factors. For example, the spatial extent of the $20 \times 20$ receptive field may not be large not enough to allow the third layer to learn longer-range dependencies. The 3rd layer RBM, in turn, simply fine-tunes the representations learned in the 2nd layer. Another explanation is that the fine-tuning of the 2nd layer features is an important information processing mechanism for representing high-order structure in natural images. Working along these lines, an important future study will be to devise
feature learning models that can capture progressively larger areas of the visual input. Another test would be to assess the accuracy of linearized models based on the 3rd layer at predicting the responses of the visual area V4, which receives a majority of the output connections from V2.
Chapter 4

Modeling abstract scene processing
In Chapter 3 we investigated how the features learned by a hierarchical model of pixel values can be used to account for the responses of neurons in V1 and V2. However, statistical regularities exist not only at the level of pixels, but at more abstract levels as well. For example, particular types of objects tend to occur together specific types of scenes. In this chapter we apply feature learning methods to higher-level visual primitives—objects—to model how late stages of the human visual system represent abstract information about natural scenes.

4.1 Introduction

During natural vision humans categorize the scenes that they encounter. A scene category can often be inferred from the objects present in the scene. For example, a person can infer that she is at the beach by seeing water, sand, and sun bathers. Inferences can also be made in the opposite direction: the category “beach” is sufficient to elicit the recall of these objects plus many others such as towels, umbrellas, sandcastles, and so on. These objects are very different from those that would be recalled for another scene category such as an office. These observations suggest that humans use knowledge about how objects co-occur in the natural world to categorize natural scenes.

There is substantial behavioral evidence to show that humans exploit the co-occurrence statistics of objects during natural vision. For example, object recognition is faster when objects in a scene are contextually consistent (Biederman 1972; Palmer 1975). When a scene contains objects that are contextually inconsistent then scene categorization is more difficult (Davenport and Potter 2004; Joubert et al. 2007). Despite the likely importance of object co-occurrence statistics for visual scene perception, few functional magnetic resonance imaging (fMRI) studies have investigated this issue systematically. Most previous fMRI studies have investigated isolated and decontextualized objects (Kanwisher et al. 1997; Downing et al. 2001), or a few, very broad scene categories (Epstein and Kanwisher 1998; Peelen et al. 2009). However, two recent fMRI studies (Walther et al. 2009; MacEvoy and Epstein 2011) provide some evidence that the human visual system represents information about individual objects during scene perception.

Here we test the hypothesis that the human visual system represents scene categories that capture the statistical relationships between objects in the natural world. To investigate this issue, we used a statistical learning algorithm originally developed to model large text corpora to learn scene categories that capture the co-occurrence statistics of objects found in a large collection of natural scenes. We then used fMRI to record blood-oxygenation-level-dependent (BOLD) activity evoked in the human brain when viewing natural scenes. Finally, we used the learned scene categories to model the tuning of individual voxels and we compared predictions of these models to alternative models based on object co-occurrence statistics that lack the statistical structure inherent in natural scenes. For details on experimental methods and analyses, we refer the reader to the original publication of this work (Stansbury et al. 2013).
We report three main results that are consistent with our hypothesis. First, much of anterior visual cortex represents scene categories that reflect the co-occurrence statistics of objects in natural scenes. Second, voxels located within and beyond the boundaries of many well-established functional ROIs in anterior visual cortex are tuned to mixtures of these scene categories. Third, scene categories and the specific objects that occur in novel scenes can be accurately decoded from evoked brain activity alone. Taken together, these results suggest that scene categories represented in the human brain capture the statistical relationships between objects in the natural world.

4.2 Experiments

4.2.1 Category feature learning with latent Dirichlet allocation

To test whether the brain represents scene categories that reflect the co-occurrence statistics of objects in natural scenes, we first had to obtain such a set of categories. We used statistical feature learning methods to solve this problem. First, we created a learning database by labeling the individual objects in a large collection of natural scenes. The frequency counts of the objects that appeared in each scene in the learning database were then used as input to the Latent Dirichlet Allocation (LDA) learning algorithm (Blei et al. 2003). Latent Dirichlet allocation is a generative probabilistic model, originally developed to model underlying topics in a collection of documents based on the co-occurrence statistics of the words in the documents (Figure 4.2.1). The model assumes that the objects in a scene are drawn from a mixture model of $K$ distinct categories. A scene is represented by a length $O$ indicator vector $o$ whose entries are the number of times that the $w$-th object from an available vocabulary ($w = 1...W$) occurs in the scene. The likelihood of observing a scene under LDA is defined as follows:

$$p(o; \alpha, \beta) = \int \prod_{j=1}^{O} \sum_{c_j=1}^{K} p(o_j, c_j; \theta; \beta, \alpha) d\theta$$

$$= \int \left( \prod_{j=1}^{O} \sum_{c_j=1}^{K} p(o_j|c_j; \beta)p(c_j|\theta) \right) p(\theta; \alpha) d\theta,$$

(4.2.1)

where

$$p(\theta; \alpha) \sim Dirichlet(\alpha)$$

(4.2.2)

$$p(c|\theta) \sim Multinomial(\theta)$$

(4.2.3)

$$p(o|c = k; \beta) \sim Multinomial(\beta^{(k)}).$$

(4.2.4)

The model assumes that the objects in a scene are generated by the following process: first, a parameter vector $\theta$ is first drawn (via Equation 4.2.2). The parameter $\theta$ defines
a multinomial probability distribution \( p(c = k|\theta) \) over the \( K \) possible scene categories (Equation 4.2.3). Given the parameter \( \theta \), the \( j \)-th object in a scene \((j = 1,...O)\) is associated with a category index \( c_j = k \) that is drawn independently from \( p(c = k|\theta) \). Each of the possible scene category indices \((k = 1...K)\) are associated a column \( \beta^{(k)} \) of the parameter matrix \( \beta \in \mathbb{R}^{W \times K} \). Each entry of \( \beta \) is the average likelihood that the \( w \)-th object (row) occurs in (or more exactly, is drawn from) the \( k \)-th category (columns). Given \( c_j \) and \( \beta \), the index to an object label is drawn from the conditional multinomial distribution in 4.2.4. For a database of many scenes, the parameter \( \alpha \) is a length \( K \) vector that characterizes the overall statistical distribution of the scene categories observed across the entire database. Specifically, \( \alpha \) is a list of parameters for a \( K \)-dimensional Dirichlet distribution from which category mixtures for each scene are drawn.

Due to the intractability of the integral with respect \( \theta \) to in Equation 4.2.1, the probability of a given document cannot be evaluated exactly. Instead, a variational approximation of the log-probability is used (see Blei et al. 2003 for details). The model parameters \( \alpha \) and \( \beta \) are optimized for a learning database \( \mathcal{D} \) of \( D \) labeled natural scenes \((\mathcal{D} = \{o_1, o_2, ... o_D\})\) by maximizing the probability over all observations:

\[
p(\mathcal{D}; \alpha, \beta) = \prod_{d=1}^{D} p(\mathcal{o}_d; \alpha, \beta).
\]

(4.2.5)

The variational expectation maximization algorithm used to optimize the parameters is beyond the scope of this work, but we refer the reviewer to the original paper (Blei et al. 2003) for details. During training, the LDA model learns a set of latent features that capture the co-occurrence statistics of the objects in the learning database. The features are learned without supervision. However, the number of distinct features learned and the object vocabulary must be specified by the experimenter. The vocabulary used for our study consisted of the most frequent objects in the learning database.

When applied to the learning database of object labels, LDA learns a set of features that are intuitively interpreted as scene categories (Figure 4.2.2-B (right)). Each of the learned categories can be named intuitively by inspecting the objects they are most likely to contain. For example, the first category in Figure 4.2.2-B is aptly named 'Roadway' because it is most likely to contain the objects “car,” “vehicle,” “highway,” “crash barrier,” and “street lamp.” The other examples shown in Figure 4.2.2-B can also be assigned intuitive names that describe typical natural scenes.

A key feature of LDA is that once the parameters of the model are optimized, the posterior distribution over the latent categorical features can be calculated for a new
Figure 4.2.1: Graphical representation of the latent Dirichlet allocation model. For each of the $D$ scenes that compose the learning database, it is assumed that the observed objects in the $d$-th scene are generated as follows: First, a parameter vector $\theta_d$, $(d = 1 ... D)$ is drawn independently from a Dirichlet distribution $p(\theta; \alpha)$. Each $\theta_d$ defines a scene-level multinomial probability distribution over the indices to $K$ possible scene categories $p(c = k|\theta_d)$, $(k = 1 ... K)$. Further, LDA assumes that each of $O_d$ objects $o_{dj}$, $(j = 1 ... O_d)$ in the $d$-th scene are associated with the scene category index $c_{dj}$. The category indices are drawn from $p(c|\theta_d)$. For each of the $K$ possible values that $c$ can take, there is an associated object-level parameter vector $\beta^{(k)}$ that defines a multinomial probability distribution $p(o = w; \beta^{(k)})$, $(w = 1 ... W)$. Here $W$ is the size of the available object vocabulary. Each of the objects in the $d$-th scene are then assumed to be generated by drawing an object index from the conditional distribution $p(o_{dj} = v|c_{dj}; \beta)$. During category learning, the goal is to estimate the parameters $\beta$ and $\alpha$ that would have generated the object labels observed in the learning database.
scene, conditioned on the objects occurring in that scene:\(^1\)

\[
p(c|o; \alpha, \beta) = \int p(c, \theta|o; \alpha, \beta) d\theta
\]

\[
= \int \frac{p(o, c, \theta; \alpha, \beta)}{p(o; \alpha, \beta)} d\theta.
\]

(4.2.6)

In the remainder of this work we use these inferred posterior probabilities as the features transformations to construct linearized encoding models of voxel activity evoked in the human brain during natural visions (Figure 4.2.2).

4.2.2 Estimating linearized voxel-wise encoding models

To determine whether the brain represents the scene categories learned by LDA we recorded BOLD brain activity evoked when human subjects viewed 1260 individual natural scene images. We used the LDA posterior inference procedure to estimate the probability that each of the presented stimulus scenes belonged to each of a learned set of categories. For instance, if a scene contained the objects “plate,” “table,” “fish,” and “beverage,’ LDA would assign the scene a high probability of belonging to the “Dining” category in Figure 4.2.2-B, a lower probability to the “Aquatic” category, and near zero probability to the remaining categories (Figure 4.2.2-A, left).

The category probabilities inferred for each stimulus scene were used to construct voxel-wise encoding models. The encoding model for each voxel consisted of a set of weights that best mapped the inferred category probabilities of the stimulus scenes onto the BOLD responses evoked by the scenes (Figure 4.2.2-A, right). Model weights were estimated using regularized linear regression applied independently for each subject and voxel. The prediction accuracy for each voxel-wise encoding model was defined to be the correlation coefficient (Pearson’s \(r\) score) between the responses evoked by a novel set of stimulus scenes and the responses to those scenes predicted by the model.

Introspection suggests that humans can conceive of a vast number of distinct objects and scene categories. However, because the spatial and temporal resolution of fMRI data are fairly coarse (Buxton 2002), it is unlikely that all these objects or scene categories can be recovered from BOLD signals. BOLD signal-to-noise ratios (SNR) also vary dramatically across individuals, so the amount of information that can recovered from individual fMRI data also varies. Therefore, before proceeding with further analysis of the voxel-wise models, we first identified the single set of scene categories that provided the best predictions of brain activity recorded from all subjects. To do so, we examined how the

---

\(^1\)This posterior also intractable to compute. Instead a factorized variational distribution \(q(c, \theta|o) \approx q(c)q(\theta)\) is used (see Blei et al., 2003). We then only keep the category term \(q(c)\) and thus do not need to perform the integration over \(\theta\).
Figure 4.2.2: Modeling the representation of scene categories in the human brain. A, The general framework proposed in Chapter 2 is used to estimate encoding and decoding models of fMRI BOLD responses. The feature mapping is supplied by posterior inference from LDA. The feature representation captures how each stimulus scene is described as a probabilistic mixture of the scene category features learned by LDA. B, The LDA algorithm learned the high-level scene features from the objects that occur natural scenes. (left). The objects in a large database of natural scenes were hand-labeled and provided as input to the LDA algorithm (center). LDA learns a set of latent features that can be interpreted as probabilistic scene categories (right). Each list on the right depicts one of the features learned by LDA. The darkness of each object label within a list gives the relative probability of the object belonging to corresponding feature. The labels above each column are interpretations of the object probabilities and intuitively interpreted as natural scenes categories.
amount of accurately-predicted cortical territory across subjects varied with specific settings of the number of individual scene categories and object vocabulary size assumed by the LDA algorithm during category learning. Specifically, we incremented the number of individual categories learned from 2 to 40 while also varying the size of the object label vocabulary from the 25 most frequent to 950 most frequent objects in the learning database. Figure 4.2.3-A shows the relative amount of accurately-predicted cortical territory across subjects based on each setting. Accurate predictions are stable across a wide range of settings.

Across subjects, the encoding models perform best when based on 20 individual categories and composed of a vocabulary 850 objects (Figure 4.2.3-A, indicated by red dot). Examples of these categories are displayed in Figure 4.2.3-B. Previous fMRI studies have only used 2-8 distinct categories and 2-200 individual objects (see Walther et al. 2009; MacEvoy and Epstein 2011). Thus, our results show there is more information in BOLD signals related to encoding scene categories than has been previously appreciated.

We next tested whether natural scene categories were necessary to accurately model the measured fMRI data. We derived a set of null scene categories by training LDA on artificial scenes. The artificial scenes were created by scrambling the objects in the learning database across scenes, thus removing the natural statistical structure of object co-occurrences inherent in the original learning database. If the brain incorporates information about the co-occurrence statistics of objects in natural scenes, then the prediction accuracy of encoding models based upon these null scene categories should be much poorer than encoding models based on scene categories learned from natural scenes.

Indeed, we find that encoding models based on the categories learned from natural scenes provide significantly better predictions of brain activity than do encoding models based on the null categories, for all subjects ($p < 1 \times 10^{-10}$ for all subjects, Wilcoxon rank-sum test for differences in median prediction accuracy across all cortical voxels and candidate scene category settings; subject S1: $W(15025164) = 9.96 \times 10^{13}$; subject S2: $W(24440399) = 3.04 \times 10^{14}$; subject S3: $W(15778360) = 9.93 \times 10^{13}$; subject S4: $W(14705625) = 1.09 \times 10^{14}$).

In a set of supplemental analyses, we also compared the LDA-based models to several other plausible models of scene category representation. The alternative models were based on various encodings including diagnostic objects from the LDA categories, linear encodings of object co-occurrences (Deerwester et al. 1990), spectral signatures (Oliva and Torralba 2001), and wavelet decomposition (Kay et al. 2008). Compared to these alternative models, we find that the LDA-based models provide superior prediction accuracy of voxel responses. These results support our central hypothesis that the human brain encodes categories that reflect the co-occurrence statistics of objects in natural scenes.
Figure 4.2.3: Identifying the best scene categories for modeling data across subjects. **A**, Encoding model performance across a range of settings for the specified number of distinct categories learned using LDA (y-axis) and vocabulary size (x-axis). Each pixel corresponds to one of the candidate scene categories learned by LDA when applied to the learning database. The color of each pixel represents the relative amount of cortical territory across subjects that is accurately predicted by encoding models based on a specific setting for the number of individual categories and vocabulary size. The number of individual categories was incremented from 2 to 40. The object vocabulary was varied from the 25 most frequent to the 950 most frequent objects in the learning database. The red dot identifies the number of individual categories and vocabulary size that produce accurate predictions for the largest amount of cortical territory across subjects. **B**, Ten examples taken from the 20 best scene categories identified across subjects (corresponding to the red dot in A). The seven most probable objects for each category are shown as individual lists. The probability of each object occurring each category is indicated by saturation.
4.2.3 Categories learned from natural scenes explain selectivity in many anterior visual rois

Previous fMRI studies have identified functional regions of interest (ROIs) tuned to very broad scene categories, such as places (Epstein and Kanwisher 1998), as well as to narrow object categories such as faces (Kanwisher et al. 1997) or body parts (Downing et al. 2001). Can selectivity in these regions be explained in terms of the categories learned from natural scene object statistics?

We evaluated scene category tuning for voxels located within the boundaries of several conventional functional ROIs: the fusiform face area (FFA; Kanwisher et al. 1997), the occipital face area (OFA; Gauthier et al. 2000), the extrastriate body area (EBA; Downing et al. 2001), the parahippocampal place area (PPA; Epstein and Kanwisher 1998), the transverse occipital sulcus (TOS; Nakamura et al. 2000; Grill-Spector 2003; Hasson et al. 2003), retrosplenial cortex (RSC; Maguire 2001) and lateral occipital cortex (LO; Malach et al. 1995).

Figure 4.2.4-A shows the boundaries of these ROIs, identified using separate functional localizer experiments, and projected on the cortical flat map of one representative subject. The color of each location on the cortical map indicates the prediction accuracy of the corresponding encoding model. All encoding models were based on the 20 best scene categories identified across subjects. These data show that the encoding models accurately predict responses of voxels located in many ROIs within anterior visual cortex. To quantify this effect, we calculated the proportion of response variance explained by the encoding models, averaged across all voxels within each ROI. We find that the average proportion of variance explained to be significantly greater than chance for every anterior visual cortex ROI, and for all subjects ($p < 0.01$; Figure 4.2.4-B). Thus, selectivity in many previously identified ROIs can be explained in terms of tuning to scene categories learned from natural scene statistics.

To determine whether scene category tuning is consistent with tuning reported in earlier localizer studies, we visualize the weights of encoding models fit to voxels within each ROI. Figure 4.2.5 shows encoding model weights averaged across all voxels located within each function ROI. Scene category selectivity is broadly consistent with the results of previous functional localizer experiments. For example, previous studies have suggested that PPA is selective for presence of buildings (Epstein and Kanwisher 1998). The LDA algorithm suggests that images containing buildings are most likely to belong to the “Urban/Street” category (see Figure 4.2.3-B), and we find that voxels within PPA have large weights for the “Urban/Street” category. To take another example, previous studies have suggested that OFA is selective for the presence of human faces (Gauthier et al. 2000). Under the trained LDA model, images containing faces are most likely to belong to the “Portrait” category, and we find that voxels within OFA have large weights for the “Portrait” category.
Figure 4.2.4: Encoding model performance. A, Encoding model prediction accuracies are mapped onto the left (LH) and right (RH) cortical surfaces of one representative subject. Gray indicates areas outside of the scan boundary. Bright locations indicate voxels that are accurately predicted by the corresponding encoding model (prediction accuracy at two levels of statistical significance—$p < 0.01 (r = 0.21)$ and $p < 0.001 (r = 0.28)$—are highlighted on the colorbar). ROIs identified in separate retinotopy and functional localizer experiments are outlined in white. The bright regions overlap with a number of the ROIs in anterior visual cortex. These ROIs are associated with representing various high-level visual features. However, the activity of voxels in retinotopic visual areas (V1, V2, V3, V4, V3a, V3b) are not predicted accurately by the encoding models. Prediction accuracy was calculated on responses from a separate validation set of stimuli not used to estimate the model. ROI Abbreviations: Retinotopic Visual Areas 1-4 (V1-V4); Para-hippocampal Place Area (PPA); FFA, Fusiform Face Area (PPA); Extrastriate Body Area (EBA); Occipital Face Area (OFA); Retrosplenial Cortex (RSC); Transverse Occipital Sulcus (TOS). Center key: A=anterior; P=posterior, S=superior; I=inferior. B, Each bar indicates the average proportion of voxel response variance in an ROI that is explained by voxel-wise encoding models estimated for a single subject. Bar colors distinguish individual subjects. Error bars are standard error from the mean. For all anterior visual ROIs and for all subjects, a significant proportion of voxel response variance ($p < 0.01$, indicated by red lines) is explained by encoding models based on scene categories learned from natural scenes.
Figure 4.2.5: *Average encoding model weights across subjects, calculated for various functional ROIs.* The average encoding model weights for voxels within distinct functional ROIs. Averages are calculated across all voxels located within the boundaries of an ROI, and across subjects. Each row displays the average weights for the scene category listed on the left margin. Each column distinguishes average weights for individual ROIs. The color of each pixel represents the positive (red) or negative (blue) average ROI weight for the corresponding category. The size of each pixel is inversely proportional to the magnitude of the standard error of the mean estimate; larger pixels indicate selectivity estimates with greater confidence. Standard error scaling is according to the data within an ROI (column). ROI tuning is generally consistent with previous findings. However, tuning also appears to be more complex than indicated by conventional ROI-based analyses. ROIs definitions follow the same conventions as in Figure 4.2.4.
Although category tuning within functional ROIs is generally consistent with previous reports, Figure 4.2.5 demonstrates that tuning is clearly more complicated than assumed previously. In particular, many functional ROIs are tuned for more than one scene category. For example, both FFA and OFA are thought to be selective for human faces, but voxels in both these areas also have large weights for the “Plants” category. Additionally, area TOS, an ROI generally associated with encoding information important for navigation, has relatively large weights for the “Portrait” and “People Moving” categories. Thus, our results suggest that tuning in conventional ROIs may be more diverse than generally believed (for additional evidence, see (Huth et al. 2012; Naselaris et al. 2012).

### 4.2.4 Decoding natural scene categories from evoked brain activity

The results presented thus far suggest that information about natural scene categories is encoded in the activity of many voxels located in anterior visual cortex. It should therefore be possible to decode these scene categories from brain activity evoked by viewing a scene. To investigate this possibility, we constructed a decoder for each subject that uses voxel activity evoked in anterior visual cortex to predict the probability that a viewed scene belongs to each of 20 best scene categories identified across subjects. To maximize performance, the decoder used only those voxels for which the encoding models produced accurate predictions on a held-out portion of the model estimation data.

We used the decoder to predict the 20 category probabilities for 126 novel scenes that had not been used to construct the decoder. Figure 4.2.6-A shows several examples of the category probabilities predicted by the decoder. The scene in the upper right of Figure 4.2.6-A depicts a harbor in front of a city skyline. The predicted category probabilities indicate that the scene is most likely a mixture of the categories “Urban” and “Boatway”, which is an accurate description of the scene. Inspection of the other examples in the Figure suggests that the predicted scene category probabilities accurately describe many different types of natural scenes.

To quantify the accuracy of each decoder, we calculated the correlation (Pearson’s $r$) between the scene category probabilities predicted by the decoder and the probabilities inferred using the LDA algorithm (conditioned on the labeled objects in each scene). Figure 4.2.6-B shows the distribution of decoding accuracies across all decoded scenes, for each subject. The median accuracies and 95% confidence interval on median estimates are indicated by the black cross-hairs. Most of the novel scenes are decoded significantly for all subjects. Prediction accuracy across all scenes exhibited systematically greater-than-chance performance for all subjects ($p < 0.02$ for all subjects, Wilcox rank-sum test; subject S1: $W(126) = 18585$; subject S2: $W(126) = 17274$; subject S3: $W(126) = 17018$; subject S4: $W(126) = 19214$. The voxels selected for the decoding analysis summarized in Figure 4.2.6 were located throughout the visual cortex. However, we also find that
Figure 4.2.6: Decoding scene categories and objects in novel natural scenes. A, Examples of scene category and object probabilities decoded from evoked BOLD activity. Blue boxes display novel stimulus scenes observed by subjects S1 (top row) through S4 (bottom row). Each red box encloses the top category probabilities predicted by the decoder for the corresponding scene to the left. The saturation of each label represents the predicted probability that the observed scene belongs to the corresponding category. Black boxes enclose the objects with the highest estimated probability of occurring in the observed scene to the left. The saturation of each label represents the estimated probability of the corresponding object occurring in the scene. B, Decoding accuracy for predicted category probabilities. Category decoding accuracy for a scene is the correlation coefficient between the category probabilities predicted by the decoder and the category probabilities inferred directly using LDA. Each plot shows the (horizontally mirrored) histogram of decoding accuracies for a single subject. Median decoding accuracy and 95% confidence interval (CI) calculated across all decoded scenes is represented by black cross-hairs overlaid on each plot. Decoding accuracy greater than 0.58 was considered statistically significant ($p < 0.01$) (indicated by the red line). A large majority of the decoded scenes are statistically significant, including all examples shown in A. C, Decoding accuracy for predicted object probabilities. Object decoding accuracy is the ratio of the likelihood of the objects labeled in each scene given the decoded category probabilities, to the likelihood of the labeled objects in each scene if all were selected with equal probability (chance). A likelihood ratio greater than one (red line) indicates that the objects in a scene are better predicted by the decoded object probabilities than by selecting objects randomly. Each plot shows the (horizontally mirrored) histogram of likelihood ratios for a single subject.
accurate decoding can be obtained using the responses of subsets of voxels located within specific ROIs.

### 4.2.5 Predicting the objects that occur in decoded natural scenes

Our results suggest that the visual system represents scene categories that capture the co-occurrence statistics of objects in the natural world. This suggests that we should be able to accurately predict the likely objects in a scene based on the scene category probabilities decoded from evoked brain activity.

To investigate this issue, we estimated the probability that each of the 850 objects in the vocabulary for the single best set of scene categories identified across subjects occurred in each of the 126 decoded validation set scenes. The probabilities were estimated by combining the decoded category probabilities with the probabilistic relationship between categories and objects established by LDA learning algorithm during category learning. The resulting probabilities give an estimate of the likelihood that each of the 850 objects occurs in each of the 126 decoded scenes.

In Figure 4.2.6-A, labels in the black boxes indicate the most likely objects estimated for the corresponding decoded scene. For the harbor and skyline scene at upper right, the most probable objects predicted for the scene are “building,” “sky,” “tree,” “water,” “car,” “road,” and “boat.” All of these objects either occur in the scene or are consistent with the scene context. Inspection of the other examples in the Figure suggests that the most probable objects are generally consistent with the scene category.

To quantify how accurately the objects were decoded, we used the distribution of object probabilities estimated for each scene to calculate the likelihood of labeled objects in the scene. We then calculated the likelihood of the labeled objects from a naive distribution that assumes all 850 objects are equally likely to occur. The ratio of these likelihoods provides a measure of accuracy for the estimated object probabilities. Likelihood ratios greater than one indicate that the estimated object probabilities better predict the labeled objects in the scene than by picking objects at random.

Figure 4.2.6-C shows the distribution of likelihood ratios for each subject, calculated for all 126 decoded scenes. The medians and 95% confidence intervals of the median estimates are indicated by the black cross-hairs. Object prediction accuracy across all scenes indicates systematically greater-than-chance performance for all subjects ($p < 1 \times 10^{-15}$ for all subjects, Wilcoxon rank-sum test; subject S1: $W(126) = 9983$; subject S2: $W(126) = 11375$; subject S3: $W(126) = 11103$; subject S4: $W(126) = 10715$).

The estimated object probabilities and the likelihood ratio analysis both show that the objects that are likely to occur in a scene can be predicted probabilistically from natural scene categories that are encoded in human brain activity. This suggests that humans might use a probabilistic strategy to help infer the likely objects in a scene from fragmentary information available at any point in time.
4.3 Discussion

This study provides compelling evidence that the human visual system encodes scene categories that reflect the co-occurrence statistics of objects in the natural world. First, categories that capture co-occurrence statistics are consistent with our intuitive interpretations of natural scenes. Second, voxel-wise encoding models based on these categories accurately predict visually evoked BOLD activity across much of anterior visual cortex, including within several conventional functional ROIs. Finally, the category of a scene and its constituent objects can be decoded from BOLD activity evoked by viewing the scene.

Previous studies of scene representation in the human brain used subjective categories that were selected by the experimenters. In contrast, our study used a data-driven, statistical algorithm (LDA) to learn the intrinsic categorical structure of natural scenes from object labels. These learned, intrinsic scene categories provide a more objective foundation for scene perception research than is possible using subjective categories.

One previous study of scene perception used a similar statistical learning approach to investigate the intrinsic category structure of natural scenes (Fei-Fei and Perona 2005). In that study the input to the learning algorithm was visual features of intermediate spatial complexity. Because our goal was to determine whether the brain represents the object co-occurrence statistics of natural scenes, we used object labels of natural scenes as input to the learning algorithm rather than intermediate visual features.

The voxel-wise modeling and decoding framework employed here (Thirion et al. 2006; Kay et al. 2008; Mitchell et al. 2008; Naselaris et al. 2009; Nishimoto et al. 2011; Naselaris et al. 2012) provides a powerful alternative to conventional methods based on statistical parametric mapping (Friston et al. 1996) or multivariate pattern analysis (MVPA; Norman et al. 2006). Studies based on statistical mapping or MVPA do not aim to produce explicit predictive models of voxel tuning, so it is difficult to generalize their results beyond the specific stimuli or task conditions used in each study. In contrast, the goal of voxel-wise modeling is to produce models that can accurately predict responses to arbitrary, novel stimuli or task conditions. A key strategy for developing theoretical models of natural systems has been to validate model predictions under novel conditions (Hastie et al. 2008). We believe that this strategy is also critically important for developing theories of representation in the human brain.

Our results generally corroborate the many previous reports of object selectivity in anterior visual cortex. However, we find that tuning properties in this part of visual cortex are more complex than reported in previous studies. This difference likely reflects the sensitivity afforded by the voxel-wise modeling and decoding framework. Still, much work remains before we can claim a complete understanding of what and how information is represented in anterior visual cortex (Huth et al. 2012; Naselaris et al. 2012).

Several recent studies (Kim and Biederman 2009; Peelen et al. 2009; MacEvoy and Epstein 2011) have suggested that the lateral occipital complex (LO) represents, in part, the identity of scene categories based on the objects therein. Taken together these studies
suggest that some sub-regions within LO should be accurately predicted by models that link objects with scene categories. Our study employs one such model. We find that the encoding models based on natural scene categories provide accurate predictions of activity in anterior portions of LO (Figure 4.2.4A-B). Note however, that our results do not necessarily imply that LO represents scene categories explicitly.

Functional MRI provides only a coarse proxy of neural activity and has a low SNR. In order to correctly interpret the results of fMRI experiments it is important to quantify how much information can be recovered from these data. Here we addressed this problem by testing many candidate models in order to determine a single set of scene categories that can be recovered reliably from the BOLD activity measured across all of our subjects (Figure 4.2.3-A). This test places a clear empirical limit on the number of scene categories and objects that can be recovered from our data. These numbers are larger than what has typically been assumed in previous fMRI studies of scene perception (Epstein and Kanwisher 1998; Peelen et al. 2009; Walther et al. 2009; MacEvoy and Epstein 2011), but they are still far smaller than the likely representational capacity of the human visual system. Theoreticians have argued that the simple statistical properties of natural scenes explain selectivity to low-level features in peripheral sensory areas (Olshausen and Field 1996; Lewicki 2002; Smith and Lewicki 2006). Behavioral data suggest that low-level natural scene statistics also influence the perception of scene categories (Greene and Oliva 2009). Though several qualitative theories have been proposed that link the object statistics of natural scenes with human scene perception (Biederman 1972; Palmer 1975), none have provided an objective, quantitative framework to support this link. The current study provides such a framework. Our data-driven, model-based approach shows that scene categories encoded in the human brain can be derived from the co-occurrence statistics of objects in natural scenes. This further suggests that the brain exploits natural scene statistics at multiple levels of abstraction. If this is true, then natural scene statistics might be used as a principled means to develop quantitative models of representation throughout the visual hierarchy.
Chapter 5

Epilogue
5.1 Summary of contributions

The main contributions of this dissertation are as follows:

- I propose and evaluate a general framework to objectively develop, test, and compare theoretical models of neural representation in the brain. This framework combines system identification with the statistical feature learning to generate predictive, interpretable models of neural responses. This framework proves to be a powerful data-driven alternative to classic methods used to model the response properties of neurons.

- Using simple compositional rules I formulate a hierarchical model and apply the model to learn a hierarchical representation of natural images. I show that the features learned in the lowest layer of the model accurately characterize the responses of V1 cell, but not V2 cells. I then demonstrate that the features represented in the higher levels of the hierarchical model provide a better characterization of V2 neural responses. This supports using hierarchical features learned from natural images to characterize visual representation outside of V1. Analysis of the features in higher layers also points to potential stimulus dimensions that could be encoded in V2, suggesting directions for future research.

- I show that abstract visual features are learned by modeling the distribution of higher-order visual primitives in natural scenes. Specifically, I show that intuitive scene categories emerge when performing feature learning on the labels of objects that occur in natural scenes. I then go on to show that these features accurately characterize evoked cortical activity in late stages of visual processing in the human brain.

- During the course of this work, I have developed an object-oriented MATLAB software toolbox that implements statistical learning algorithms such as the restricted Boltzmann machine described in Chapter 3, as well as many others. This software is general and can be applied to many input domains. The code is publicly available online.

5.2 Open questions and future directions

The work presented in this dissertation can be extended and/or improved in a number of ways. One obvious extension is to apply the hierarchical feature learning described in Chapter 3 to model even later stages of the visual hierarchy, such as area V4. Though the features learned in layer 3 of the hierarchical model provide a good feature representation for characterizing V2 neurons, they do not dramatically improve model predictions when compared to the features learned in the 2nd layer. Could the 3rd layer provide a better characterization of neural responses in V4? If the 3rd layer features do provide
accurate models of V4, it would then be necessary to determine the key differences in image representation between the 2nd and 3rd layer features.

The last point suggests another important direction for future research: developing better methods for visualizing hierarchical data. The current work attempts to visualize the features in the higher layers of the hierarchical model in terms of the input. This approach has obvious intuitive merit, as the aim is to better understand how neurons represent the visual input. However, we must also address that most neurons in the visual system only represent the visual input implicitly through the activities of other neurons. Thus, in order to better understand the actual information processing principles implemented by neurons, we may need to step away from interpreting neural function in terms of the stimulus and instead in terms of the input signals from other neurons.

This raises the related question: how much information about brain function can we infer from the activities of single neurons? Neurons do not operate in isolation and it is just as likely that measured neural responses are due to the activity of locally-connected neurons than by the stimulus per se. It will thus be increasingly important to measure and characterize the activity of connected networks of neurons. Chronic array recordings like the one used in Section 3.3.2 offer one possible method for measuring population responses. However, the spatial sampling resolution offered by electrophysiological recording arrays may still be too coarse to capture functional connectivity of local networks. Recent neuroimaging techniques such as 2-photon microscopy and calcium imaging are thus likely to become evermore important for studying how information is encoded in neural populations.

Another obvious extension of the current work would be to assess the findings of classic neurophysiology experiments using the hierarchical representations learned from natural scenes. When presented with stimuli such as Cartesian gratings (Hubel and Wiesel 1968; Movshon et al. 1978ab), polar and non-Cartesian gratings (Gallant et al. 1993), curved shapes (Pasupathy and Connor 2001), and 2nd order stimuli (Baker and Mareshal 2001), do the higher-order features learned in the hierarchical model exhibit response properties akin to those measured in V2 and V4 neurophysiology? If so, it would suggest that a majority of the current hypotheses regarding the properties of V2 and V4 neurons can be explained in terms of high-order structure of natural images.

Chapter 3 suggest that the features learned in the 2nd layer of the hierarchical model primarily capture border information defined by contours, colors and textures. The fact that these features also provide an accurate characterization neural responses of V2 neurons (in terms of linear combinations of the features) suggests that stimuli derived by modulating these dimensions should drive V2 exceptionally well. Future neurophysiology experiments based on such a stimulus set would be informative for identifying specific functional and anatomical organization of V2 neurons.

In Chapter 4, rather than learning hierarchical image features from pixel values, high-level scene features were learned from more abstract visual primitives, namely objects. Though choosing to analyze objects is somewhat subjective, few would argue that scenes are not composed of individual objects. Modeling higher-order primitives offers a number
of benefits. First, modeling higher-order primitives offers statistical power. Though recent research in AI has shown that abstract semantic features can be learned directly from pixels (Le et al. 2012), these models require an extremely large amount of training data and extensive computational resources. Second, statistical features learned from objects are fairly intuitive, facilitating the interpretation of the learned features. Future work on identifying and analyzing other high-order visual primitives could provide promising directions in cognitive science.

Another way of improving the current work is to devise more biologically plausible feature learning algorithms. The DBN employed in Chapter 3 has a number of benefits including a simple, robust regularization scheme and tractable learning. However, this feature learning methods is far from biologically accurate. For example, it is known that the spatial extent of the receptive field increases with a progression in the visual hierarchy (Gattass et al. 1981). Interestingly, the 2nd layer of the hierarchical model is able to capture the increase in receptive field size observed between V1 and V2. However, the spatial extent of the features learned in 3rd layer of the model do not appear to increase beyond that of the second layer. This is likely due to the fact that the spatial scale of the visual input is equivalent at all stages of the DBN (encompassing only 20 pixels, or approximately 1 degree of visual angle). An interesting extension of the DBN based on convolution and spatial pooling operations at each layer was introduced by Lee et al. 2009 to learn features from whole images. When trained on natural images, the model learns hierarchical features that represent progressively larger regions of the input. It would be interesting to incorporate similar convolutional feature learning methods into the modeling framework proposed in this thesis to investigate neural representation in V4 and IT. However, such models may also need to incorporate mechanisms to account for top-down processing in order to provide accurate predictions of neural responses in the high-level visual processing areas (Luck et al. 1997; Reynolds and Chelazzi 2004).

The LDA feature learning model used in Chapter 4 is also likely not optimal for modeling the true distribution of objects in natural scenes. Latent Dirichlet allocation assumes that a scene is composed of a bag of words and only captures the co-occurrence dependencies among objects. In the real world however, objects not only have co-occurrence dependencies but a number of other dependencies such as spatial location, orientation, and saliency. Future feature learning methods will need to be able to capture this joint information while still being tractable to estimate.

5.3 Closing remarks

The human brain is the most magnificent computational system known. Ironically, our brains are quite poor reconciling their own function. As research on brain function continues, I believe that we will need to rely less on introspection and more on objective, data-driven methods to guide theory and experiments. This dissertation is one of the first attempts to show that a fully data-driven approach is not only a feasible but also a
valid means for studying neural representation in the brain. As neuroimaging, statistical learning, and numerical optimization methods continue to improve, so to will the relevance of this work.
Bibliography


Bell, A. J. and Sejnowski, T. J. (1997). The independent components of natural scenes are edge filters. *Vision research, 37*(23):3327–3338. 3.1


Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America. A, Optics and image science, 2*(7):1160–1169. 2.1.2, 2.2.2, 2.3.3, 2.4.2


Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47. 2.1, 2.1.3, 3.1

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394. 2, 2.2.1, 2.3.2, 2.3.3, 2.4.2, 2.4.2, 3.2.8.1


Marmarelis, P. V. Z. (2004). Nonlinear Dynamic Modeling of Physiological Systems. Wiley. 2.2.1, 2.2.1


Movshon, J. A., Thompson, I. D., and Tolhurst, D. J. (1978a). Receptive field organization of complex cells in the cat’s striate cortex. The Journal of Physiology, 283:79–99. 2.1, 2.1.2, 2.2.2, 2.3, 2.4.3, 2.4.4, 5.2


Naselaris, T., Kay, K. N., Nishimoto, S., and Gallant, J. L. (2011). Encoding and decoding in fMRI. NeuroImage, 56(2):400–410. 2.2.2, 2.2.2
Naselaris, T., Prenger, R. J., Kay, K. N., Oliver, M., and Gallant, J. L. (2009). Bayesian reconstruction of natural images from human brain activity. *Neuron*, 63:902–915. \[2.2.2, 4.3\]


Oliver, M. D., Nishimoto, S., Naselaris, T. N., and Gallant, J. L. (2012). Using natural movies to map foveal V1 spatial-temporal-chromatic receptive fields in awake animals. In *Society for Neuroscience Annual Meeting (poster), 2012*, New Orleans, LA. \[2.2.1, 3.3.2\]


Wu, M. C.-K., David, S. V., and Gallant, J. L. (2006). Complete functional characterization of sensory neurons by system identification. *Annual Review of Neuroscience*, 29:477–505. 2.2.1, 2.2.1, 2.2.2, 2.2.2


Appendix A

RBM Training Details
A.1 Maximum likelihood estimation for restricted Boltzmann machines

Restricted Boltzmann machines (RBMs) are a subclass of the more general Markov Random Field (MRF), and thus follow general parameter estimation procedures used for MRFs. Thus we start off by deriving parameter estimation for MRFs, then move on to calculate the gradient descent learning algorithm for RBMs.

A.1.1 General parameter estimation

Given a MRF model we would like to optimize the parameters to account for some observed data. A common, well-studied technique for estimating model parameters in a probabilistic model is Maximum Likelihood Estimation (MLE). In MLE we assume that we observe $D$ datapoints $X = (x_1, x_2, \ldots, x_D)$ that are sampled i.i.d. from some underlying function $f(x)$. We choose to model these data with a function $p(x; \theta)$ that is parameterized by the values in $\theta$ and defines a probability distribution over the data. The function $p(x; \theta)$ is referred to as the likelihood function, as it defines how likely the data are for given model parameters. Assuming the data are sampled $i.i.d$. allows the likelihood for all observed data to be defined as the product of probabilities for each individual data point:

$$p(X; \theta) = \prod_d p(x_d; \theta).$$  \hfill (A.1.1)

The goal is to estimate the optimal parameters $\theta^*$ that maximize the likelihood of the data. It is customary to instead find the parameters that minimize the negative log of the likelihood $\ell(X; \theta) = -\log p(X; \theta)$, giving the following optimization problem:

$$\theta^* = \arg \min_{\theta} \ell(X; \theta)$$

$$= \arg \min_{\theta} -\log \prod_d p(x_d; \theta)$$

$$= \arg \min_{\theta} -\sum_d \log p(x_d; \theta).$$  \hfill (A.1.2)

In this work, we use an MRF model with $x = \{v, h\}$ that includes a set of visible units $v$ and a set of hidden units $h$. The negative log likelihood over the visible units for this model is defined directly from Equation 3.2.4:
\[ \ell_{\text{MRF}}(v; \theta) = -\log p_{\text{MRF}}(v; \theta) \]
\[ = -\log \int p_{\text{MRF}}(v, h; \theta) \, dh \]
\[ = -\log \int \frac{1}{Z_{\text{MRF}}} e^{-E_{\text{MRF}}(v, h; \theta)} \, dh \]
\[ = \log Z_{\text{MRF}} + \log \int e^{-E_{\text{MRF}}(v, h; \theta)} \, dh \]
\[ = \log \int \int e^{-E_{\text{MRF}}(v', h'; \theta)} \, dv' \, dh' - \log \int e^{-E_{\text{MRF}}(v, h; \theta)} \, dh. \quad (A.1.3) \]

The standard approach for solving optimization problems like Equation A.1.2 is to use gradient descent. Thus we must find the gradient of \( \ell(v; \theta) \) in Equation A.1.3 with respect to \( \theta \). To make notation more concise, we will omit the explicit parameterization by \( \theta \). Thus, \( p(v; \theta) \) will be written as \( p(v) \), etc. Additionally we define \( p(v, h) = p_{\text{MRF}}(v, h) \), \( E(v, h) = E_{\text{MRF}}(v, h; \theta) \), and \( Z = Z_{\text{MRF}}(\theta) \). Using these conventions, the gradient of the negative log likelihood is given by the following calculations:

\[ \frac{\partial \ell_{\text{MRF}}(v)}{\partial \theta} = \frac{\partial}{\partial \theta} \left( \log \int \int e^{-E(v', h')} \, dv' \, dh' \right) - \frac{\partial}{\partial \theta} \left( \log \int e^{-E(v, h)} \, dh \right) \]
\[ = \frac{1}{\int \int e^{-E(v', h')} \, dv' \, dh'} \int \int \frac{\partial}{\partial \theta} e^{-E(v', h')} \, dv' \, dh' \]
\[ - \frac{1}{\int e^{-E(v, h)} \, dh} \int \frac{\partial}{\partial \theta} e^{-E(v, h)} \, dh \]
\[ = \frac{1}{\int e^{-E(v, h)} \, dh} \int e^{-E(v, h)} \frac{\partial}{\partial \theta} E(v, h) \, dh \]
\[ - \frac{1}{\int \int e^{-E(v', h')} \, dv' \, dh'} \int \int e^{-E(v', h')} \frac{\partial}{\partial \theta} E(v', h') \, dv' \, dh'. \]

Applying Equations 3.2.4 and 3.2.5 and the notion that \( Z \) is a constant, we obtain

\[ \frac{\partial \ell_{\text{MRF}}(v)}{\partial \theta} = \frac{1}{Z \int P(v, h) \, dh} \int p(v, h) \frac{\partial}{\partial \theta} E(v, h) \, dh \]
\[ - \frac{1}{Z} \int \int p(v', h') \frac{\partial}{\partial \theta} E(v', h') \, dv' \, dh'. \]

Integrating out the hidden units in the denominator of the first term and then bringing the resulting term inside the remaining integral with respect to \( h \), applying the definition of conditional probability, it follows that:
\[ \frac{\partial \ell_{MRF}(v)}{\partial \theta} = \int \frac{p(v, h)}{p(v)} \frac{\partial}{\partial \theta} E(v, h) dh - \int \int p(v', h') \frac{\partial}{\partial \theta} E(v', h') dv' dh' \]
\[ = \int p(h|v) \frac{\partial}{\partial \theta} E(v, h) dh - \int \int p(v', h') \frac{\partial}{\partial \theta} E(v', h') dv' dh' \]
\[ = \mathbb{E}_{p(h|v)} \left[ \frac{\partial}{\partial \theta} E(v, h) \right] - \mathbb{E}_{p(v', h')} \left[ \frac{\partial}{\partial \theta} E(v', h') \right]. \quad (A.1.4) \]

Thus \( \frac{\partial \ell_{MRF}(x)}{\partial \theta} \) is composed of two expectations under the current model parameters. The first term is the expected value of the hidden unit probabilities conditioned on the observed data points. The second is the expected value of the joint probability of the visible and hidden units. This gradient holds for all MRFs and we use this gradient when training RBMs.

In general, calculating the expectations in Equation A.1.4 is intractable due to integration. However, if the data samples and model distributions can be sampled \( i.i.d. \), then Monte Carlo methods can be used to approximate these expectations. Specifically, the exact expectations are replaced sample means:

\[ \frac{\partial \ell_{MRF}(v)}{\partial \theta} = \mathbb{E}_{p(h|x)} \left[ \frac{\partial}{\partial \theta} E(v, h) \right] - \mathbb{E}_{p(v', h')} \left[ \frac{\partial}{\partial \theta} E(v', h') \right] \]
\[ \approx \left\langle \frac{\partial E(v_x, h_x)}{\partial \theta} \right\rangle_{p(h_x|v_x)} - \left\langle \frac{\partial E(v_\theta, h_\theta)}{\partial \theta} \right\rangle_{p(h_\theta|v_\theta)} \]
\[ = \left\langle \frac{\partial E(v, h)}{\partial \theta} \right\rangle_x - \left\langle \frac{\partial E(v, h)}{\partial \theta} \right\rangle_\theta, \quad (A.1.5) \]

where \( \langle \cdot \rangle_{p(\cdot)} \) is the sample mean under the distribution \( p(\cdot) \). Here \( p(h_x|v_x) \) is the distribution of the hidden units states when conditioned on observed data \( v = x \), and \( p(h_\theta|v_\theta) \) is the distribution of the hidden unit states when conditioned on visible states that are sampled from the current model.

We assume that the observed data are sampled independently, thus the sample mean in the first term is trivial to calculate if we have an expression for \( p(h|v) \). However, the sample mean in the second term involves drawing independent samples from the model. Drawing these samples is difficult in most cases. However RBMs, which inherit a bipartite graph structure, allows blocked Gibbs sampling, and thus these samples can be generated efficiently.
A.1.2 Negative log likelihood gradients $\partial \ell (v; \theta) / \partial \theta$ for Gaussian-input RBM

To perform the gradient descent algorithm, we derive here the gradients $\partial \ell (v; \theta) / \partial \theta$ with respect to each of the RBM model parameters and the corresponding weight updates based on Monte Carlo approximation. We perform the calculations for RBMs with Gaussian visible units, these calculations are easily extended to RBMs with binary visible units.

Calculating $\partial \ell (v; \theta) / \partial W$

Noting that $\partial E(v, h) / \partial W = -v^T h$ (Equation 3.2.14), the partial derivatives of log likelihood with respect to $W$ are given by:

$$
\frac{\partial \ell (v; \theta)}{\partial w_{ij}} = \mathbb{E}_{p(h|v)} \left[ \frac{\partial}{\partial w_{ij}} E(v, h) \right] - \mathbb{E}_{p(v', h')} \left[ \frac{\partial}{\partial w_{ij}} E(v', h') \right]
$$

$$
= \int \sum_{h'} p(v', h') v'_i h'_j dv' - \sum_{h} p(h|v) v_i h_j
$$

$$
= \int p(v') \sum_{h'} p(h'|v') v'_i h'_j dv' - \sum_{h} p(h|v) v_i h_j
$$

$$
= \int p(v') p(h'=1|v') v'_i dv' - p(h_j = 1|v) v_i
$$

$$
= \mathbb{E}_{p(v')} \left[ p(h'_j = 1|v') v'_i \right] - p(h_j = 1|v) v_i. \quad (A.1.6)
$$

where we have used the notion that $\mathbb{E}_{p(h|v)} [v_i h_j] = p(h_j = 1|v) v_i$ (see Appendix A.2.4).

The Monte Carlo approximation to the gradient follows as:

$$
\frac{\partial \ell (v; \theta)}{\partial W} \approx \langle v^T h \rangle_\theta - \langle v^T h \rangle_x. \quad (A.1.7)
$$

and the corresponding gradient descent weight updates for learning rate $\epsilon$:

$$
\Delta W = \epsilon \left( \langle v^T h \rangle_x - \langle v^T h \rangle_\theta \right). \quad (A.1.8)
$$

Calculating $\partial L(v; \theta) / \partial a$

Noting that $\partial E(v, h) / \partial a = -h$ (Equation 3.2.14), the partial derivatives of the log likelihood with respect to $a$ are given by:
\[
\frac{\partial \ell(v; \theta)}{\partial a_j} = \mathbb{E}_{p(h|v)} \left[ \frac{\partial}{\partial a_j} E(v, h) \right] - \mathbb{E}_{p(v', h')} \left[ \frac{\partial}{\partial a_j} E(v', h') \right]
\]

\[
= \int \sum_{h'} p(v', h') h'_j d v' - \sum_h p(h|v) h_j
\]

\[
= \int p(v') \sum_{h'} p(h'|v') h'_j d v' - \sum_h p(h|v) h_j 
\]

\[
= \int p(v') p(h = 1|v') d v' - p(h = 1|v)
\]

\[
= \mathbb{E}_{p(v')} \left[ p(h'_j = 1|v') \right] - p(h = 1|v). \tag{A.1.9}
\]

with the Monte Carlo approximation given by:

\[
\frac{\partial \ell(v; \theta)}{\partial a} \approx \langle h \rangle_\theta - \langle h \rangle_x. \tag{A.1.10}
\]

and the corresponding gradient descent weight updates for learning rate \( \epsilon \)

\[
\Delta a = \epsilon (\langle h \rangle_x - \langle h \rangle_\theta) \tag{A.1.11}
\]

**Calculating \( \partial \ell(v; \theta)/\partial b \)**

Noting that \( \frac{\partial E(v, h)}{\partial b} = -(v - b) \) (Equation 3.2.14), the partial derivatives of the negative log likelihood with respect to \( b \) are given by:

\[
\frac{\partial \ell(v; \theta)}{\partial b_i} = \mathbb{E}_{p(h|v)} \left[ \frac{\partial}{\partial b_i} E(v, h) \right] - \mathbb{E}_{p(v', h')} \left[ \frac{\partial}{\partial b_i} E(v', h') \right]
\]

\[
= \int \sum_{h'} p(v', h') (v'_i - b_i) d v' - \sum_h p(h|v) (v_i - b_i)
\]

\[
= \int p(v') (v'_i - b_i) \sum_{h'} p(h'|v') d v' - (v_i - b_i) \sum_h p(h|v).
\]

Using the notion that \( \sum_h p(h|v) = 1 \), simplifies the expression:

\[
\frac{\partial \ell(v; \theta)}{\partial b_i} = \int p(v') (v'_i - b_i) d v' - (v_i - b_i)
\]

\[
= \mathbb{E}_{p(v')} [v'_i - b_i] - (v_i - b_i), \tag{A.1.12}
\]

with the Monte Carlo approximation of the gradient as follows:
\[
\frac{\partial \ell(v; \theta)}{\partial b} \approx \langle v - b \rangle_\theta - \langle v - b \rangle_x \tag{A.1.13}
\]
and the corresponding gradient descent weight updates for learning rate \( \epsilon \)
\[
\Delta b = \epsilon (\langle v - b \rangle_x - \langle v - b \rangle_\theta). \tag{A.1.14}
\]

A.2 Probability density functions for Gaussian-input RBMs

In order to calculate gradients of \( \ell_{RBM}(v) \) with respect to the model parameters, we
need to be able to sample from the conditional distributions \( p_{RBM}(h|v) \) and \( p_{RBM}(v|h) \). Here we derive these expressions for an RBM with Gaussian visible units, but the calculations are easily extended for RBMs with binary visible units.

A.2.1 Joint density function

The joint probability for an RBM with binary hidden units and Gaussian inputs and
assumed unit variance is as follows:

\[
p_{RBM}(v, h) = \frac{1}{Z_{RBM}} e^{-E_{GRBM}(v, h)} = \frac{1}{Z_{RBM}} \frac{\sum_{ij} v_i w_{ij} h_j}{\sigma_i^2} - \frac{\sum_i (v_i - b_i)^2}{2 \sigma_i^2} + \sum_j a_j h_j
\]

where the partition function for the RBM is:

\[
Z_{RBM} = \int \frac{\sum_{ij} \prod \left( e^{-v_i w_{ij} h'_j + \frac{1}{2} (v'_i - b_i)^2 - a_j h'_j} \right) dv' }{\int \sum_{ij} \prod \left( e^{-v_i w_{ij} h'_j + \frac{1}{2} (v'_i - b_i)^2 - a_j h'_j} \right) dv'}.
\]

A.2.2 Marginal density functions

The marginal probability \( p(v) \) for the Gaussian RBM is as follows:
\[ p(\mathbf{v}) = \sum_{\mathbf{h}'} p_{\text{RBM}}(\mathbf{v}, \mathbf{h}') \]
\[ = \sum_{h'} \frac{1}{Z_{\text{RBM}}} \prod_{ij} e^{v_i w_{ij} h'_j - \frac{1}{2}(v_i - b_i)^2 + a_j h_j} \]
\[ = \frac{1}{Z_{\text{RBM}}} e^{-\frac{1}{2} \sum_i (v_i - b_i)^2} \sum_{h'} \prod_{j} e^{(a_j + \sum_i v_i w_{ij})h'_j} \]
\[ = \frac{1}{Z_{\text{RBM}}} e^{-\frac{1}{2} \sum_i (v_i - b_i)^2} \prod_{j} \sum_{h'} e^{(a_j + \sum_i v_i w_{ij})h'_j} \]
\[ = \frac{1}{Z_{\text{RBM}}} e^{-\frac{1}{2} \sum_i (v_i - b_i)^2} \prod_{j} (1 + e^{(a_j + \sum_i v_i w_{ij})}). \quad (A.2.1) \]

The marginal probability \( p(\mathbf{h}) \) is as follows:

\[ p(\mathbf{h}) = \int p(\mathbf{v}', \mathbf{h}) d\mathbf{v}' \]
\[ = \int \frac{1}{Z_{\text{RBM}}} e^{\sum_{ij} v_i w_{ij} h_j - \frac{1}{2} \sum_i (v_i - b_i)^2 + \sum_j a_j h_j} d\mathbf{v}' \]
\[ = \frac{1}{Z_{\text{RBM}}} e^{\sum_j a_j h_j} \int \prod_{i} e^{\sum_j v'_i w_{ij} h_j - \frac{1}{2}(v'_i - b_i)^2} d\mathbf{v}' \]
\[ = \frac{1}{Z_{\text{RBM}}} e^{\sum_j a_j h_j} \prod_{i} \int e^{\sum_j v'_i w_{ij} h_j - \frac{1}{2}(v'_i - b_i)^2} d\mathbf{v}' \]
\[ = \frac{1}{Z_{\text{RBM}}} e^{\sum_j a_j h_j} \prod_{i} \int e^{\frac{2}{2} \sum_j v'_i w_{ij} h_j - (v'_i - b_i)^2} e^{-b_i h_j} d\mathbf{v}' \]
\[ = \frac{1}{Z_{\text{RBM}}} e^{\sum_j a_j h_j} \prod_{i} \int e^{-\frac{(v'_i - b_i - \sum_j w_{ij} h_j)^2}{2}} d\mathbf{v}'. \quad (A.2.2) \]

Adding a zero to the numerator of the exponent in Equation A.2.2 and completing the square gives the following:
Applying the fact that the integral in Equation A.2.3 is the normalizing constant for a Gaussian with mean equal to \( b_i + \sum_i w_{ij} h_j \) and dimensions having equal unit variance (via z-scoring), we obtain

\[
p(h) = \frac{1}{Z_{RBM}} e^{\sum_j a_j h_j} \prod_i e^{-\frac{1}{2} \sum_i (v_i^2 - 2v_i (b_i + \sum_i w_{ij} h_j) + (b_i + \sum_i w_{ij} h_j)^2)} \int d\mathbf{v}'
\]

(A.2.3)

A.2.3 Conditional density functions

Having expressions for the marginal probability densities, we now determine the expressions for conditional density functions. The probability of the hidden units conditioned on the visible units is given by:

\[
p(h|v) = \frac{p(v,h)}{p(v)} = \frac{1}{Z_{RBM}} \prod_j e^{a_j h_j} \prod_i e^{\frac{1}{2} \sum_i (v_i - b_i)^2 + a_j h_j} \prod_j (1 + e^{(a_j + \sum_i v_i w_{ij}) h_j})
\]

(A.2.4)
\[ p(h_j = 1|v) = \frac{e^{(a_j + v_i w_{ij})}}{1 + e^{(a_j + \sum_i v_i w_{ij})}} = \frac{1}{1 + e^{-(a_j + \sum_i v_i w_{ij})}} = \sigma(a_j + \sum_i v_i w_{ij}), \quad (A.2.5) \]

where \( \sigma(\cdot) \) indicates the logistic sigmoid function. In matrix form, this conditional probability is \( p(h = 1|v) = \sigma(a + W^T v) \). The conditional probability \( p(v = 1|h) = \sigma(b + Wh) \) for binary visible units is derived in a similar fashion.

The conditional probability for Gaussian visible units is given by:

\[
p(v|h) = \frac{p(v, h)}{p(h)} = \frac{1}{Z_{RBM}} \prod_i e^{\sum_j v_i w_{ij} h_j - \frac{1}{2}(v_i - b_i)^2 + \sum a_{ij} h_j} = \frac{1}{Z_{RBM}} e^{\sum_j a_{ij} h_j} \prod_i \sqrt{2\pi} e^{-2b_i \sum_j w_{ij} h_j + (\sum_j w_{ij} h_j)^2} \prod_i e^{\sum_j v_i w_{ij} h_j - \frac{1}{2}(v_i - b_i)^2} \prod_i \sqrt{2\pi} e^{-2v_i b_i + (\sum_j w_{ij} h_j)^2} = \prod_i \frac{1}{\sqrt{2\pi}} e^{-\frac{(v_i - b_i)^2}{2} - \sum_j v_i w_{ij} h_j} \prod_i \frac{1}{\sqrt{2\pi}} e^{\frac{v_i^2 - 2v_i b_i + b_i^2}{2} - \sum_j v_i w_{ij} h_j + 2b_i \sum j w_{ij} h_j + (\sum_j w_{ij} h_j)^2} = \prod_i \frac{1}{\sqrt{2\pi}} e^{-(v_i - b_i - \sum_j w_{ij} h_j)^2} = \prod_i \mathcal{N}(b_i + \sum_j w_{ij} h_j, 1). \quad (A.2.6)\]

Thus, the conditional density \( p(v|h) \) is a Normal distribution with mean \( b + Wh \) and identity covariance.

A.2.4 Derivation of \( \mathbb{E}_{p(h|v)}[v_i h_j] = p(h_j = 1|v) v_i \)

When calculating \( \partial \ell(v; \theta) / \partial W \) we use the notion that \( \mathbb{E}_{p(h|v)}[v_i h_j] = p(h_j = 1|v) v_i \), which is demonstrated here:
\[ E_{p(h|v)}[v; h_j] = \sum_h \prod_{k=1}^K p(h_k|v)v_i h_j \]
\[ = v_i \sum_h h_j \prod_k p(h_k|v) \]
\[ = v_i \sum_{h_1} p(h_1|v) \times \cdots \sum_{h_{j-1}} p(h_{j-1}|v) \times \ldots \sum_{h_j} p(h_j|v) \times \sum_{h_{j+1}} p(h_{j+1}|v) \times \cdots \sum_{h_M} p(h_M|v) \]
\[ = v_i (1 \times \cdots \times 1 \times \sum_{h_j} p(h_j|v)h_j \times 1 \times \cdots \times 1) \]
\[ = v_i \sum_{h_j} p(h_j|v)h_j \]
\[ = v_i (p(h_j = 0|v)0 + p(h_j = 1|v)1) \]
\[ = p(h_j = 1|v)v_i. \]

### A.3 Target regularization for RBMs

For this work we assume that the RBM hidden units take on binary states. For regularized parameter estimation, we aim to solve the following optimization problem:

\[ \theta_{reg}^* = \arg \min_{\theta} \ell(v; \theta) + \nu \Psi(h_x) \]
\[ = \arg \min_{\theta} \ell(v; \theta) + \nu \sum_{j=1}^H \sum_{d=1}^D \Psi(h_{jd}) \]

where \( \ell(v; \theta) \) is the unregularized negative log likelihood function and

\[ \Psi(h_{jd}) = -\psi_{jd} \log h_{jd} - (1 - \psi_{jd}) \log (1 - h_{jd}) \]

is a cross entropy regularizer. The regularizer measures the difference between the hidden unit states when being driven by the \( d \)-th data point, and a target binary distribution \( \psi \). The hyperparameter \( \nu \) is a regularization constant that determines the degree of regularization. If we define the term \( z = a + v_x^TW \), where \( v_x \) is the data-driven states of the visible units, then \( p(h_x|v_x) = \sigma(z) \), given by the point-wise logistic sigmoid function. The chain rule can be used to calculate the gradient of the regularization term with respect to \( \theta \):

\[ \frac{\partial \Psi(h_x)}{\partial \theta} = \frac{\partial \Psi(h_x)}{\partial h_x} \frac{\partial h_x}{\partial z} \frac{\partial z}{\partial \theta}. \]
Here the first term in the product is the derivative of the cross entropy function, the second term is the derivative of the sigmoid function, and the third term depends on the parameter with which to calculate the gradient. The gradients with respect to $W$ are then as follows:

\[
\frac{\partial \Psi(h_{jd})}{\partial w_{ij}} = \frac{\partial \psi(h_{jd})}{\partial h_{jd}} \frac{\partial h_{jd}}{\partial z_{jd}} \frac{\partial z_{jd}}{\partial w_{ij}} \\
= \left(\frac{(1 - \psi_{jd})}{(1 - h_{jd})} - \frac{\psi_{jd}}{h_{jd}}\right) ((h_{jd})(1 - h_{jd})) v_{id} \\
= v_{id} ((h_{jd} - h_{jd}\psi_{jd}) - (\psi_{jd} + h_{jd}\psi_{jd})) \\
= v_{id}(h_{jd} - \psi_{jd}) \tag{A.3.2}
\]

This term is now included to regularize the gradient descent parameter updates for $W$, with $\nu = -\lambda$ in order to perform descent:

\[
\Delta_{\text{reg}} W = \Delta W - \lambda \frac{\partial \Psi(h_{x})}{\partial W} \\
= \epsilon \left(\langle v^T h \rangle_x - \langle v^T h \rangle_\theta\right) - \lambda v^T (h_{x} - \psi) \\
= \langle v^T [h(\epsilon - \lambda) + \lambda \psi] \rangle_x - \epsilon \langle v^T h \rangle_\theta \\
= \epsilon \left(\langle v^T h(1 - \phi) + \phi \psi \rangle_x - \epsilon \langle v^T h \rangle_\theta\right) \\
= \epsilon \left(\langle v^T \tilde{h} \rangle_x - \langle v^T h \rangle_\theta\right),
\]

where $\Delta W$ is the weight update for unregularized gradient descent. The term $\tilde{h} = (1 - \phi) h_{x} + \phi \psi$ is an augmented version of the hidden unit states that is an interpolation between the data-driven states $h_{x}$ and the target distribution $\psi$. The amount of interpolation is defined by the hyperparameter $\phi = \lambda / \epsilon$, which is constrained to exist between zero and one, thus defining a convex combination. In principle, the target distribution can take on any form that supports binary variables. For example, $\psi$ is used in this work to induce both sparsity and specificity among hidden units. Substituting $\partial z / \partial a$ into Equation A.3.1 gives the updates for the hidden unit biases $a$:

\[
\Delta_{\text{reg}} a = \epsilon \left(\langle \tilde{h} \rangle_x - \langle h \rangle_\theta\right).
\]