A UNIFIED APPROACH TO THE NONLINEARITIES OF VISUAL NEURONS: THE CURVED GEOMETRY OF NEURAL RESPONSE SURFACES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

James R. Golden

August 2015

A UNIFIED APPROACH TO THE NONLINEARITIES OF VISUAL NEURONS: THE CURVED GEOMETRY OF NEURAL RESPONSE SURFACES

James R. Golden, Ph.D.

Cornell University 2015

The responses of visual cortical neurons are highly nonlinear functions of image stimuli. I present a geometric view of these nonlinear responses and classify them as forms of selectivity or invariance, building on a body of established work. With the sparse coding network, a well-known network model of V1 computation, I attempt to quantify selectivity and invariance by measuring the curvature of neural response surfaces in both low-dimensional subspaces and image state space. I argue that this geometric view allows the precise quantification of feature selectivity and invariance in network models in a way that provides insight into the computations necessary for object recognition, and that this view may be a useful tool for future physiological experiments.

BIOGRAPHICAL SKETCH

James Golden grew up in Hatboro, Pennsylvania and attended Hatboro-Horsham High School. He graduated with a bachelor's degree in engineering from Swarthmore College in 2005. He worked as a patent examiner in Washington, DC until 2007 and then started in the electrical and computer engineering graduate program at Cornell University while residing at the Telluride House on campus. After receiving an M.Eng. in 2009, he started in the graduate program in psychology at Cornell and studied under Professor David J. Field.

This dissertation is dedicated to my wife Allison, who put up with me living in Ithaca, as well as my parents, who always had time to help me with my math homework. I would also like to dedicate this to my teachers and mentors: David Field, Shimon Edelman, James Cutting, Ron Hoy, Bruce Johnson, Bernie Hutchins, Carr Everbach, Joseph Carapucci, Jacqueline Anderson, Derek Fromal, Joseph Birzes, Claire Tuckman, Joe Turk, Margaret Deardorff, Nancy Pregler, Jim Kelly and Robert Ayton.

I would also like to dedicate this to the fellow students who inspired me during my graduate career: Jared Strait, Stephen Mahaffey, Bill Stubler, Nicole Baran, Jung Mee Park, Chris Garces, Kedar Vilankar, Gil Menda, Paul Shamble, Eyal Nitzany, Andrew Bielak, Ethan Warsh, Jared Mast, Anders Linderot, Angela Previdelli, Thea Walsh, Dan Ranweiler, Thea Whitman, Simona Subonj, Rayna Bell, Chelsea Morris, Derek Lockhardt and Ronald Ilma.

ACKNOWLEDGEMENTS

Research reported in this dissertation was supported by the Eunice Kennedy Shriver National Institute of Child Health and Human Development of the National Institutes of Health under award number T32HD551775. The content is solely the responsibility of the author and does not necessarily represent the official views of the National Institutes of Health.

Research reported in this dissertation was additionally supported by a Google Research grant to David J. Field.

The author was also supported by a scholarship from L.L. Nunn and the Cornell Branch of the Telluride Association while he lived at the Telluride House on campus from 2010-2013.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vi
List of Figures . . . viii
Preface . . . 1
0.1 Solving Vision . . . 1
0.2 Outline of Dissertation . . . 4

1 The nonlinearities of visual neurons 7
1.1 Background: Physiology and Computation . . . 9
1.2 Nonlinearities in V1 Responses . . . 12
1.3 Image State Space . . . 13

2 The sparse coding network 25
2.1 Redundancy and Efficient Coding of Natural Images . . . 26
2.2 Derivation of the sparse coding network equations . . . 33
2.3 The Karklin-Lewicki Network . . . 37

3 The curvature of the sparse coding network: 2D subspaces 43
3.1 The Sparse Coding Network in a 2D State Space . . . 44
3.2 The Sparse Coding Network in 2D Subspaces of Image State Space 48
3.2.1 Parabolic Fits to Subspace Curvature . . . 52
3.2.2 Averages of Subspace Isocontours . . . 55
3.2.3 Fan Fits to Subspace Isocontours . . . 58
3.2.4 The Effect of the Cost Function . . . 62

4 The curvature of the sparse coding network: image state space 69
4.1 Curvature in high dimensions: a primer . . . 71
4.2 Results of curvature measurements in high dimensions . . . 73
4.3 Summary Measures of Neural Response Curvature . . . 81
4.4 Conclusion . . . 89

5 Conclusion and future work 92

A Measuring Curvature in High Dimensions 95
A.1 History of Curvature . . . 96
A.2 The curvature of surfaces in high dimensions . . . 98
A.3 Deriving a Formula for Curvature . . . 101
A.4 The curvature of isosurfaces in high dimensions . . . 106
A.5 The meaning of curvature in high dimensions . . . 109
A.6 Conclusion . . . 118

B Additional Figures **121**

LIST OF FIGURES

1.1 Response of the optic nerve fiber of *Limulus* retina to a moving stimulus of arbitrary form 7
1.2 Illustrating neural responses in a toy low-dimensional state space . 16
1.3 Endstopping in V1 cells is explained by curved isocontours of the neuron's response 19
1.4 A model for tolerant and invariant responses due to phase shift of a grating 21

2.1 Fourier spectra of images. 28
2.2 The basis set Φ determined by the sparse coding algorithm for 8x8 natural scene patches 37
2.3 Second-order basis functions that have response properties similar to V2/V4 neurons 41
2.4 Detail of a second-order basis function 42

3.1 A scatter plot of 2D sparse data set with three causes. 46
3.2 Four sparse coding networks of varying degrees of overcompleteness 49
3.3 Histograms representing the angles between every pair of basis vectors for 8x8 sparse coding networks 50
3.4 The iso-response contours for a neuron from a 6.4X overcomplete sparse coding network 51
3.5 The iso-response contours from the above neuron fit with a parabola 53
3.6 The marginal distributions of the parabolic fit parameter to the isocontours 54
3.7 The parabolic fit parameter as a function of angle between basis vectors 56
3.8 Example isocontours and average isocontours for sparse coding neurons 57
3.9 Average isocontours for neurons from networks with different degrees of overcompleteness 58
3.10 The isocontours described by the equation of a folding fan for three vectors 60
3.11 Example sparse coding response surface with isocontours fit to the equation of the folding fan 62
3.12 Fan curvature as a function of angle between neurons in the sparse coding network 63
3.13 Solutions to an optimization problem are found when the at the intersection of the lowest possible isocontour of the cost function with the plane representing possible solutions 64
3.14 Effect of the cost function on isocontour shape 66
3.15 Average isocontours for the sparse coding network with the Laplace prior/absolute value cost function 68

4.1 The principal curvatures and principal directions of the hypersphere 72
4.2 The curvature on the response manifold of one sparse coding neuron at one point . . . 75
4.3 The curvature on the iso-response manifold of one neuron at one point . . . 77
4.4 Comparison of principal curvatures of full response surface and iso-response surface of a sparse coding neuron . . . 78
4.5 Isosurface curvature is affected by iso-level . . . 79
4.6 The curvature of a second-layer neuron from the Karklin & Lewicki network. . . . 80
4.7 The distribution of principal curvature magnitudes for points on the basis vectors on the iso-response surfaces of sparse coding neurons . 83
4.8 The distribution of principal curvature magnitudes for points near the basis vectors on the full response surfaces of neurons from sparse coding neurons . . . 84
4.9 The distribution of principal curvature magnitudes for points on the isosurfaces of neurons . . . 85
4.10 The distribution of the mean curvature (average of all principal curvatures) for response surfaces of sparse coding neurons . . . 86
4.11 A histogram of the number of principal curvatures/eigenvalues for isosurfaces which account for 95% of the total absolute curvature . 87
4.12 The absolute mean curvature as a function of the angle between the basis function of the response surface and the point at which the curvature is measured . . . 88
4.13 Plots of the mean curvature for isosreponse surfaces of Olshausen & Fieldneurons at natural scene points . . . 89
4.14 A summary of the effect of sparsity and overcompleteness on the curvature as a function of angle in the sparse coding network . . . 90

A.1 A toy model of a neural response surface . . . 99
A.2 An example surface σ with the normal vector, normal section, tangent vector and tangent plane. . . . 110
A.3 A demonstration that the curvature measure is exact in three dimensions . . . 111
A.4 Isosurfaces of a cylinder in a 3D state space . . . 112
A.5 Surface normals and principal curvatures on isosurfaces of the cylinder in 4D . . . 113
A.6 The normal sections of the 4D cylinder . . . 114
A.7 The isosurfaces for a quadratic surface defined by $z = \sqrt{[64 - (1 * x_1^2 + 0.33 * x_2^2 - 0.5 * x_3^2)]}$ . . . 115
A.8 The normal sections for the quadratic surface shown in Fig. A.7 . . 116
A.9 The normal sections for a 9-dimensional sphere with R = 4 . . . 117
A.10 A quadratic surface in 9D with simultaneous positive and negative curvature . . . 118

A.11 The principal curvatures and principal directions of the hypersphere with R = 2 in 64D state space. 119

B.1 Examples of isosurface curvature for the sparse coding network with OC = 1X 121
B.2 Examples of isosurface curvature for the sparse coding network with OC = 1.6X 122
B.3 Examples of isosurface curvature for the sparse coding network with OC = 3.2X 123
B.4 Examples of isosurface curvature for the sparse coding network with OC = 6.4X 124

PREFACE

0.1 Solving Vision

At rare moments of deep insight, the mathematician Erdős was known to announce to his colleagues, "This one's from the Book!" In Erdős' conception, the Book was a volume possessed only by God containing the most elegant proofs in number theory, and Erdős thought that the work of the mathematician was to come to know its contents (Aigner et al., 2010).

Vision scientists may already have been granted the introduction from their own Book in Marr's *Vision* (1982), but consider what would be found in a final, complete account of vision. There would certainly be a definitive philosophical argument about what it means to see; further, there would be a descriptive theory and an algorithmic implementation of the computations that are carried out after light interacts with photoreceptors; there would also be physiological implementations of the algorithms; and finally a repair guide for when the physiological systems break down. In other words, the backbone of Marr's book, but fleshed out with mathematics, code, physiology and medicine.

This dissertation is concerned with only one aspect of this endeavor: the theory and algorithms implemented by the mammalian visual system at the computational level. Here, I present a novel analysis of existing models of cortical responses that provides insight into how neural representations of image information are formed. There is still a great deal of room to improve upon existing low-level models for the responses of cortical neurons (Vinje and Gallant, 2000; Köster and Olshausen, 2013; when compared with, for example, retinal ganglion cell models Pillow et al., 2008). Simple computational models of visual neurons have a linear systems foun-

dation that is elaborated by output nonlinearities like inhibition, thresholding and randomized firing. Mammalian V1 neurons are described with complex models that attempt to capture an array of nonlinear effects like contrast gain, endstopping, and position invariance. The following investigation attempts to use a geometric language with nonlinearity as a fundamental principle to unify the descriptions of these different effects (Field and Wu, 2004) as well as quantify them in neural network models.

At some level, this is necessarily just a different description of what existing models do, because all mathematical models can be illustrated and quantified geometrically. One of the advances made in this investigation is a direct engagement with the high-dimensional image space in which visual neurons respond. The receptive field of the linear model is a single vector in image space and necessarily cannot capture effects like endstopping. Even nonlinear (quadratic) models that utilize a set of features vectors from spike-triggered covariance captures some but not all of the observed nonlinear responses of V1 neurons (Fitzgerald et al., 2011; Berkes and Wiskott, 2006). Here, we begin with the assumption that the neuron's response is a curved, continuously-differentiable surface in image state space (Edelman, 1999) and provide direct measurements of its nonlinearities (Field and Wu, 2004). To be sure, this approach has its limitations as well, but it is a new perspective that offers fundamental insight into how a network of model of neurons processes in a nonlinear manner, and leads to numerous predictions for what could be seen in actual neural responses.

Along with straightforward prediction of how a neuron will respond to an arbitrary stimulus, we are interested in the image features that affect a neuron's response. In addition to a general approach incorporating nonlinearities, we argue

that they can be classified into two distinct groups. Consider three neurons with the same classical receptive field. The first neuron's response is linear, the second shows position invariance, and the third shows endstopping. We will make a geometric argument that the second neuron with invariance will respond more strongly to a random stimulus of a given contrast on average than the linear neuron, while the third neuron with endstopping will respond less strongly on average than the linear neuron. (See Section 1.3 for details and figures.) Individual types of nonlinearities have been classified as selective or invariant, and here we provide evidence that selectivity and invariance can be seen as negative and positive curvature, respectively, of the neuron's response surface in state space (Berkes and Wiskott, 2006; Erhan et al., 2010; Tsai and Cox, 2015).

Moreover, we present a new way to quantify a neuron's selectivity and invariance from direct measurements of its response surface in high-dimensional state space. In addition to its potential usefulness for describing the responses of actual neurons measured physiologically, our approach has utility for understanding deep neural networks. These networks have been increasingly used to build models for object recognition and image labeling, and their selectivity and invariance properties are somewhat difficult to discern computationally (Erhan et al., 2010). Our geometric approach may provide a more intuitive understanding for what those networks are doing.

The most impressive experiments quantifying selectivity and invariance in high-order visual neurons (Rust and DiCarlo, 2012) use a protocol requiring the response to many different types of images, and similar approaches are used to describe neurons in deep networks (Goodfellow et al., 2009). Selectivity is measured by the change in response to texture-scrambled images of objects, and invariance

is measured by the responses to translated, scaled and rotated objects on the background. Our geometric method offers an alternative with a more direct measurement, where the magnitude and direction of the curvature of a neuron's response surface in image state space is calculated. We can use these curvatures as direct quantifications of selectivity and invariance in a new way. Thus not only does the geometric view show that selectivity and invariance are opposing manifestations of curvature, it also offers a new way to precisely quantify these aspects of a neuron's response.

0.2 Outline of Dissertation

The primary text is composed of four chapters. Chapter 1 provides background on the physiological and computational results that led to this work. The results of Hubel & Wiesel (1959, 1962, 1968) are described in a manner that connects their observations about V1 responses to our geometric argument about selectivity and invariance. Barlow's approach to a framework for efficient coding at the early stages of the visual system is also characterized. The history of the field is illustrated by Rosenblatt's neural network known as the Perceptron (Rosenblatt, 1958), the condemnations of the neural network approach by Minsky and Papert (Minsky and Papert, 1969), and the reinvention of the field with deep networks made possible by Rumelhardt and Hinton (1988). The history of the physiology and computation are connected to modern ideas about object image manifolds and neural response manifolds in high-dimensional image state space after Field and Wu (2004), Edelman (1999) and DiCarlo and Cox (2007) .

Chapter 2 focuses on one of the major computational results from the efficient

coding approach, the sparse coding network (Olshausen and Field, 1996), which captures a broad array of the nonlinear response properties of V1 neurons in a simple generative framework. In lieu of physiology, the sparse coding network is described as the basis for an investigation of the geometry of neural response surfaces in image state space. The equations that describe image representation in the network are derived and the general principles are described to set the stage for the novel observations of the remaining chapters. Additionally, a two-layer version of the sparse coding network formulated by Karklin & Lewicki (2003, 2005, 2009) is detailed. This network captures a class of response properties that are not present in the sparse coding network.

Chapter 3 presents a quantitative investigation of the response surfaces of the sparse coding network. Initially, the network is implemented in low-dimensional state spaces in order to completely visualize the response surfaces. We argue that the sparse coding network implements the right kind of curvature to produce some of the nonlinearities that have been observed in V1 neurons. Low-dimensional projections of the response surfaces from a network trained on natural images are visualized and quantified, and the curvature of the surfaces is measured as a function of various properties of the neurons and the network.

In Chapter 4, the neural response surfaces from the sparse coding network are measured in the high-dimensional image state space. These measurements allow not only the quantification of the magnitude of curvature of the manifolds, but the directions in which they are curved. Results are presented on the convexity of the surfaces, which we argue directly describes the selectivity and invariance of a neuron's response. Evidence is shown for the number dimensions of the state space in which the average surface shows nonzero curvature. These results are

compared to a similar measure of the curvature of iso-response surfaces. We also present similar results for neurons from the second layer of Karklin & Lewicki are known to have properties like V2 and V4 neurons and clearly demonstrate stronger curvature and more invariance than that of the neurons in the original sparse coding network.

Chapter 5 presents concluding thoughts on these results as well as possible futures directions. Several physiological predictions are offered, and the future of neural network modeling is considered.

Appendix A steps back to provide a theoretical overview of quantifying the curvature of a surface in a high-dimensional state space used in 4. The first step was made towards this measure in Chapter 3, but curvature was only measured in 2D subspaces of interest, while these methods allow for a measure of the curvature in the full high-dimensional state space. First, proven methods from differential geometry are described for the curvature of a surface in a two-dimensional state space and are expanded to surfaces in higher dimensions. A numerical method for measuring curvature within an optimization framework is validated and used to measure the curvature of simple quadratic surfaces in high dimensions. The intuition for curvature measures in high dimensions is built up with simple examples in order to be applied to neural response surfaces. Appendix B shows additional figures of the high-dimensional principal curvatures of sparse coding neurons, which are direct measures of selectivity and invariance.

CHAPTER 1

THE NONLINEARITIES OF VISUAL NEURONS

"The material employed in this study was the compound eye of young *Limulus polyphemus*. The action potential of this photoreceptor has been... seen to possess a simplicity of form which other arthropod eyes, as well as vertebrate retinas, do not show." - H.K. Hartline (1930)

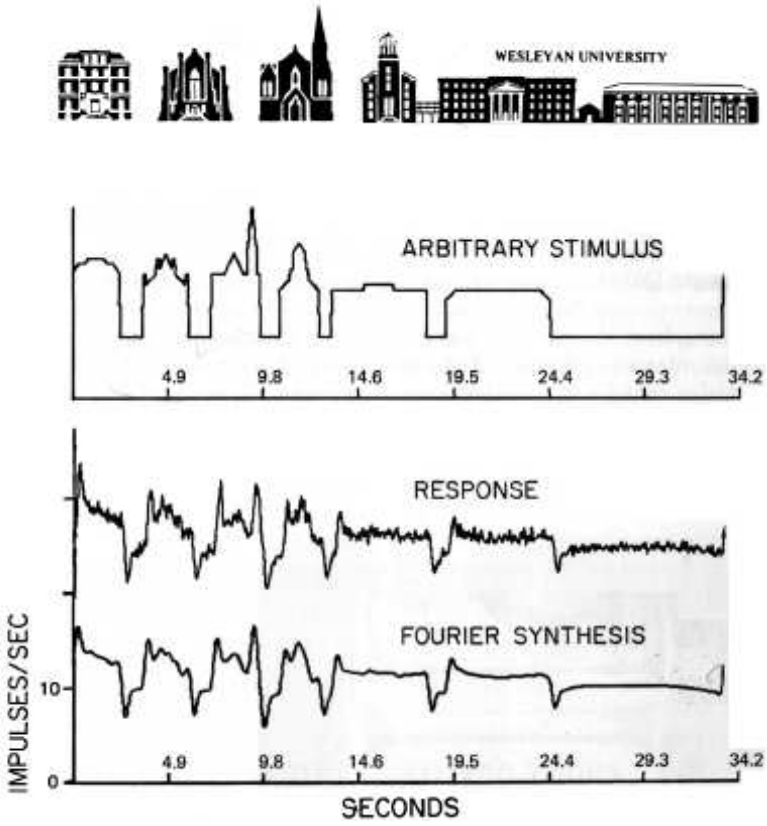

Figure 1.1: "Response of the optic nerve fiber of *Limulus* retina to a moving stimulus of arbitrary form. Upper trace: arbitrary stimulus. Middle trace: observed response (offset upward 15 impulses per second). Lower trace: predicted response" (Brodie et al., 1978).

The earliest investigations into the neurophysiology of vision began with the horseshoe crab *Limulus*. H.K. Hartline's work (1928), along with that of Adrian

(1928) in the frog, established the simple parameters that still guide the field of vision science. Some neurons exhibit responses to visual stimuli, and in order to understand how an animal sees, it is necessary to determine what causes these neurons to respond. A little over 50 years later, a team led by one of Hartline's students produced the remarkable prediction shown in Fig. 1.1. (Brodie et al., 1978). They had developed a mathematical model that described the response of *Limulus* photoreceptors. By observing the responses of a photoreceptor to an arbitrary set of stimuli, they were able to make a strikingly accurate prediction of the cell's response to a seemingly arbitrarily chosen light intensity signal that, when plotted as a function of time, traced out the shape of several buildings. They used linear systems analysis to build a model, and the accuracy of its prediction to such a highly unnatural stimulus is a testament to its soundness. The linear systems approach effectively solved the *Limulus* photoreceptor at the computational level. However, the accurate prediction of photoreceptor response is still far from a complete account of how a horseshoe crab sees the world.

This chapter provides a brief outline of the previous work on the physiological responses of visual neurons as well as the mathematical models used to understand them. Canonical physiology experiments are framed in such a way as to illustrate the complexities of modeling nonlinear V1 neurons. Early achievements in neural network modeling are also discussed in terms of their initial limitations due to their reliance on linearity. The goal of this history is to highlight the role that linear systems theory has played in both physiology and machine learning and to make clear the benefits of moving to nonlinear methods of analysis, specifically for visual cortical neurons. We bring in the idea of a state space and the geometric interpretation of nonlinear neural responses with toy examples. Two types of classical nonlinear V1 responses are interpreted in this geometric framework of

curved response surfaces. These examples are used to show the need for geometric analysis in a high-dimensional state space, which will be introduced in subsequent chapters.

1.1 Background: Physiology and Computation

Hartline's observation about the relative simplicity of this neuron when compared to those of other animals has indeed been borne out by the field over time. Lettvin et al. (1959) characterized retinal ganglion cells in the frog as firing in response to small spots, much like the flies that the frog preys upon, and called the neurons "fly detectors". Early efforts to understand visual cortical neurons in mammals were stymied by the inability to get any of the neurons to fire at all. Famously, Hubel & Wiesel (1959) determined that a cat V1 neuron fired strongly to the edge of a glass slide as it was removed from the projector, and discovered that sharp dark-light gradients maximized the firing of these neurons. Lettvin et al. (1959) as well as Hubel & Wiesel argued that visual neurons were feature detectors that only fired in response to specific images. Not long after, Campbell & Robson (1968) and Blakemore & Campbell (1969) began to use Fourier grating stimuli to probe responses. They argued against the feature detector idea and for the linear systems approach focused on spatial frequency analysis.

A linear system is one where the sum of the responses to a number of inputs is equal to the response to the sum of those inputs, or $f(x+y) = f(x) + f(y)$. This assumption is optimistic, because if it were true, the response of a neuron to any stimulus could be predicted based on its response to a small set of basis stimuli, and an understanding of the computations carried out would be straightforward.

The Fourier basis is a natural one because of its connection to periodic stimuli, and a great body of theory has been built around spectral/Fourier analysis as a component of linear systems theory. The approach has found extensive use in the design and analysis of basic electrical and mechanical systems and, as seen in Fig. 1.1, can be extremely accurate for simple neural systems. When attempting to characterize an unknown physical system, the initial attempt is usually a linear or spectrum-based approach to measure the filter/kernel (or, equivalently, receptive field, when the stimuli are images).

The naivety of the use of linear systems for vision was revealed in a debate over the theoretical limitations of such a system's complexity, focused on early artificial models for neural systems. The Perceptron (Rosenblatt, 1958) modeled sensory systems as linear classifiers, and was widely accepted as simulating some of the important properties of basic neural systems. However, a debate over its inability to carry out more complex computations for classification focused on its linearity (Minsky and Papert, 1969), which limited decision boundaries to planar surfaces. The critics were right, and because the layers could not be made nonlinear at the time, work on Perceptron-like sensory models stalled for decades. A linear model was not good enough to do the kinds of computations that brains do.

A linear classifier is not complex enough to perform difficult computations required for tasks like object recognition. A possible way around this for the Perceptron crowd was to apply a nonlinearity to the output and feed it into a second layer, but this would require training the first layer through "backpropagation", and the initial form of the equations with hard thresholds had undefined derivatives, preventing solutions for the equations for learning parameters (Olshausen, 2008). Rumelhart et al. (1988) made the fundamental insight that multilayer per-

ceptrons with soft threshold functions had solvable equations for backpropagation, and broke open the field of perceptron-like networks again. The machine learning community got around its linearity problem, but physiologists, faced with a different sort of struggle with linearity, had not found the ideal way forward.

While there is now a great deal of feedback between neuroscientists and researchers in machine learning, the neuroscience community has not developed a coherent, unified framework for describing what the visual system does at the computational level. Although the field has accumulated decades of work cataloging responses of visual cortical neurons to an endless array of stimuli, there are different names and models to describe every type of response. Here, I focus on computational models for visual neurons, and I propose a step towards a quantitative method for unifying these idiosyncratic responses.

First, I will describe the problem that has arisen by detailing a number of the nonlinear responses that have been observed in visual cortical neurons. I will discuss the sparse coding network as a model that captures some of these properties. We have developed a method for measuring the curvature, or degree of nonlinearity, of neurons in these networks, and I will argue that it can be used to quantify classes of nonlinearities. The method is applied to the original sparse coding network Olshausen and Field (1996), as well as the two-layer extension of the network (Karklin & Lewicki, 2003; 2003). Finally, I will argue for this approach as a method that could be applied to physiological experiments to test their validity.

1.2 Nonlinearities in V1 Responses

In the decades of investigation into the mammalian visual cortex, a number of nonlinear effects have been observed in responses. They include gain control (Schwartz and Simoncelli, 2001), endstopping (Hubel and Wiesel, 1968; Yazdanbakhsh and Livingstone, 2006), cross-orientation inhibition (Priebe and Ferster, 2006), changes in bandwidth with basis set (Oppenheim and Magnasco, 2013), and tolerance/invariance to position (Adelson and Bergen, 1985) and grating phase (Movshon et al., 1978), among others. Here we will provide a historical account of endstopping and position invariance in V1 neurons as they were first described Hubel & Wiesel. The purpose of this section is to show that these nonlinear responses can be described with a system of equations or with a geometric model in a low-dimensional state space.

Although it is not explicit, there is an element of the linear systems approach in how Hubel & Wiesel characterized neurons. In justifying why they designated some cat V1 neurons as complex cells, they noted "When separate ON and OFF regions could be discerned, the principles of summation and mutual antagonism, so helpful in interpreting simple fields, did not generally hold" (Hubel and Wiesel, 1962, p. 113). These are the principles of a liner system, and although many V1 cells were roughly linear and therefore designated as simple, most were not. The cells that were designated as "complex" exhibited a tolerance for the position of a light/dark edge: "Provided the slit was horizontal its exact positioning within the 3°-diameter receptive field was not critical" (Hubel and Wiesel, 1962, p. 114-115). This tolerance, or invariance, for position of the stimulus is a nonlinear effect, as it cannot be obtained by projecting a vector in image space representing the stimulus onto another vector representing the receptive field. Additionally, "the

orientation [of the slit] was critical, since a tilt of even a few degrees from the horizontal markedly reduced the response" (pg. 115); while tolerating variation in position, the cell was selective for a particular orientation. Even V1 neurons that exhibit position invariance are exhibiting simultaneous selectivity for spatial frequency; this could imply interesting geometric features, as will be described in the following section.

Endstopping is a type of nonlinear selectivity that has also been observed (see Fig. 1.4 below for a geometric explanation). Hubel & Wiesel (1968) discovered a response in macaque V1 neurons that they designated "hypercomplex". "For these, extending the line (slit, edge or dark bar) beyond the activating part of the receptive field in one or both directions caused a marked fall-off in the response" (pg. 220). This type of response is called "endstopping", and is another example of a response that cannot be modeled by a linear projection of the stimulus vector onto a receptive field vector in image state space. We argue below that neurons that exhibit endstopping are selective nonlinearities because on average a random stimulus will cause the cell to respond less than if it were linear with the same receptive field.

1.3 Image State Space

In order to examine the geometry of a neuron's response, the first step is to describe an image stimulus as a point in a high-dimensional state space. Each pixel is a dimension, and each pixel's intensity is the image's coordinate in that dimension of the state space. For example, consider an image that is just two pixels (x, y). Images consist of pairs of coordinates that represent the intensity of each pixel:

$(0,1)$, $(1,0)$ or $(1,1)$, and they can each be plotted as a point in a 2D state space. A 10x10-pixel grayscale image is represented by the 100 intensity values of its pixels, and it too can be conceptualized as a point in a 100-dimensional space at $(x_1, x_2, x_3, ..., x_{99}, x_{100})$. An image in state space can also be considered a vector from the origin that ends at $(x_1, x_2, x_3, ..., x_{99}, x_{100})$.

The principles of linear systems analysis can be applied to generate an estimated neural response using image state space. The neuron must first be characterized in terms of its receptive field. The receptive field is an image that represents the optimal stimulus that will maximize the neuron's firing. In practice, the receptive field can be found by calculating the spike-triggered average of a neuron's responses to an ensemble of noise images, which is the average of the images weighted by the number of spikes each evoked. The receptive field is an image and therefore also a point in image state space. The linear prediction of a response from this neuron is the inner product of the receptive field vector with the stimulus vector (or, depending on whether the receptive field is defined in the spatial or frequency domain, the response may be determined by a 2D convolution with the stimulus vector).

With this definition of a linear neuron and its receptive field in image state space, we may now also give a geometric definition to selectivity and invariance. For this definition, we must restrict the set of possible images in state space to be constrained to equal RMS contrast or energy, such that the set lies on the N-dimensional sphere at a radius of 1. The image on the sphere of radius 1 that maximizes the response of a linear neuron is its receptive field; the response falls off with the cosine of the angle between the receptive field vector and the stimulus. Consider another neuron that has a nonlinear response. We can still find a stimulus

that generates a maximum response from the set of possible images, and suppose it is the same as the linear neuron. The response of this neuron is measured in every possible direction moving away from the optimal stimulus, and in every direction it falls off at least as fast as that of the linear neuron. We can define this as a purely selective response (Berkes and Wiskott, 2006; Tsai and Cox, 2015). A neuron with a response that follows these criteria will respond less on average to random stimuli than a linear neuron with the same response, as was mentioned above, although this definition alone is much less rigorous.

This restriction to fall off more quickly than a linear neuron can also be translated to the language of curvature of the iso-response contours of the neuron, which are the lines (or surfaces) in the state space where the response has the same value. In order to discuss selectivity in terms of iso-response contours, we must relax the restriction to images of equal contrast that lie on the surface of a sphere in state space. If the definition of pure selectivity holds, then the iso-response contour at the optimal stimulus will curve toward the optimal stimulus vector, while the linear iso-response contour necessarily has no curvature. The selective isocontours appear hyperbolic. A selective neuron will have a lower response on average than a linear neuron with the same receptive field to a random stimulus.

Suppose for a third neuron, we find the same optimal stimulus. As we measure its response in every possible direction, we find that it falls off more slowly than a linear neuron. This is defined as a purely tolerant response (Berkes and Wiskott, 2006; Tsai and Cox, 2015). (A purely invariant response would remain constant moving in any direction. Tolerance is a less restrictive term than invariance, so even though "invariance" is described throughout this text, the intended meaning is actually that of a tolerant response.) The way that the purely tolerant neurons

have been defined, they will have a stronger response on average to a random stimulus than a linear neuron with the same receptive field. The tolerant response can also be viewed as due to iso-response contours that curve away from the optimal stimulus vector and tend to be spherical. We will focus on the curvature of iso-response contours as measures of selectivity or invariance.

With selectivity and invariance defined as the curvature of iso-response contours from the point of maximum response for a given contrast, it is possible to have a neuron that exhibits selectivity in some dimensions and tolerance/invariance in other dimensions. It is also possible to measure the curvature of the iso-response surfaces at points away from the optimal stimulus for given RMS contrast, which will serve as a gauge of the nonlinearity of the response surface.

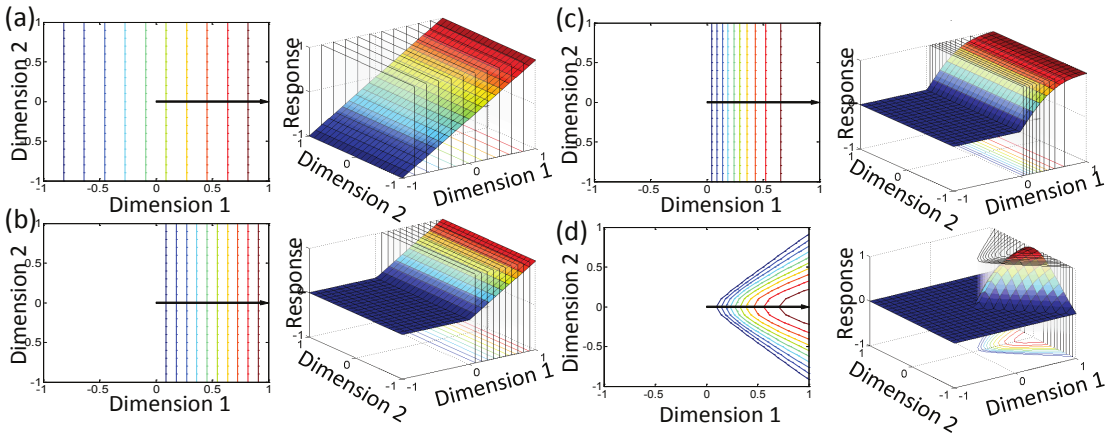

Figure 1.2: Illustrating neural responses in a toy low-dimensional state space (Golden et al., 2015). The state space consists of dimensions 1 and 2, while the response is plotted in the third dimension. Iso-response contours are drawn in the state space on the left. a) A linear neuron; b) a linear neuron with a threshold at 0; c) a neuron with a threshold at 0 and a compressive nonlinearity; d) a neuron with a selective nonlinearity.

With these definitions of selectivity and invariance based on geometry, it is informative to visualize a toy 2D state space, where the x- and y-axes represent two pixels, and the z-axis represents the response of the neuron. First, a linear neuron's response is shown in Fig. 1.2a; note that the value of the response can be projected onto the image state space by the colored lines that represent the iso-response contours. Fig. 1.2b shows a simple threshold nonlinearity, where the response is 0 when pixel 1 has a value below 0. Fig. 1.2c shows a compressive nonlinearity, where the isocontours are not spaced evenly. This is described as a "planar" nonlinearity, as its isocontours are straight lines, and it does not show selectivity or invariance compared with the linear neuron in Fig. 1.2a. Fig. 1.2d shows a simple form of a selective nonlinearity according to the above definition: compared to 1.2b and 1.2c, the neuron described by 1.2d has iso-response contours that are curved toward the optimal stimulus vector.

The particular selective nonlinearity of Fig. 1.2d is representative of endstopping, the phenomenon observed by Hubel and Wiesel in the cells that they termed hypercomplex. They described an example for which the receptive field consisted of a bright bar on a dark background with a particular orientation. An extension of the bright bar caused the firing rate to decrease. The receptive field can be represented as a point (or a vector from the origin to that point) in image state space. A toy example of this is presented in Fig. 1.3d using a 2D state space, as in Fig. 1.2d, with the x- and y-axes representing pixel intensities. Consider the images that represent the bright bar as well as the extension of the bar. Fig. 1.3a shows the optimal bar, or the receptive field; Fig. 1.3b shows an image of two bright spots that can be added to Fig. 1.3a in order to extend the bar; Fig. 1.3c shows the extended bar resulting from the sum of Figs. 1.3a and 1.3b. Although these images are points in a high-dimensional state space, it is apparent that they

lie in a 2D subspace that is illustrated in Fig. 1.3d. Figs. 1.3a and 1.3b are orthogonal, because their inner product is zero. The 2D subspace of image state space is shown in Fig. 1.3d, where the receptive field of Fig. 1.3a is represented by the green point, the two bright spots of Fig. 1.3b by the gray point, and the extended bar of Fig. 1.3c by the red point.

The neuron's response is represented by isocontours in the state space in Fig. 1.3d. If the neuron is linear, the isocontours are necessarily straight, evenly spaced and orthogonal to the receptive field vector. As shown by the dotted lines representing the isocontours in Fig. 1.3d, for a linear neuron, the response to the optimal bar (Fig. 1.3a) and extended bar (Fig. 1.3c) images should be equal at 8 spikes/second. However, as has been observed in physiological recordings, this is not the case: the response for the extended bar images in a hypercomplex V1 cell is less than the response to the optimal bar image (in this example, less than 8 spikes/second). In other words, $f(x+y) \neq f(x)+f(y)$, so the neuron's responses violate the assumption of linearity. Linearity implies straight isocontours; therefore, a simple way to represent the experimentally-observed nonlinear behavior of these neurons is to curve the isocontours. The solid curves represent these isocontours in Fig. 1.3d, and the yellow arrow shows that for a neuron with these isocontours, the firing rate to the extended bar image will only be 2 spikes/second, less than the 8 spikes/second of the optimal bar, which is a match for the observed physiological nonlinear response. The use of curved isocontours in state space offers a simple quantitative description for the endstopping response, and I will argue that this can be done more rigorously for the observed nonlinear responses in cortical visual neurons.

The effects of tolerant/invariant nonlinearities can also be described by curved

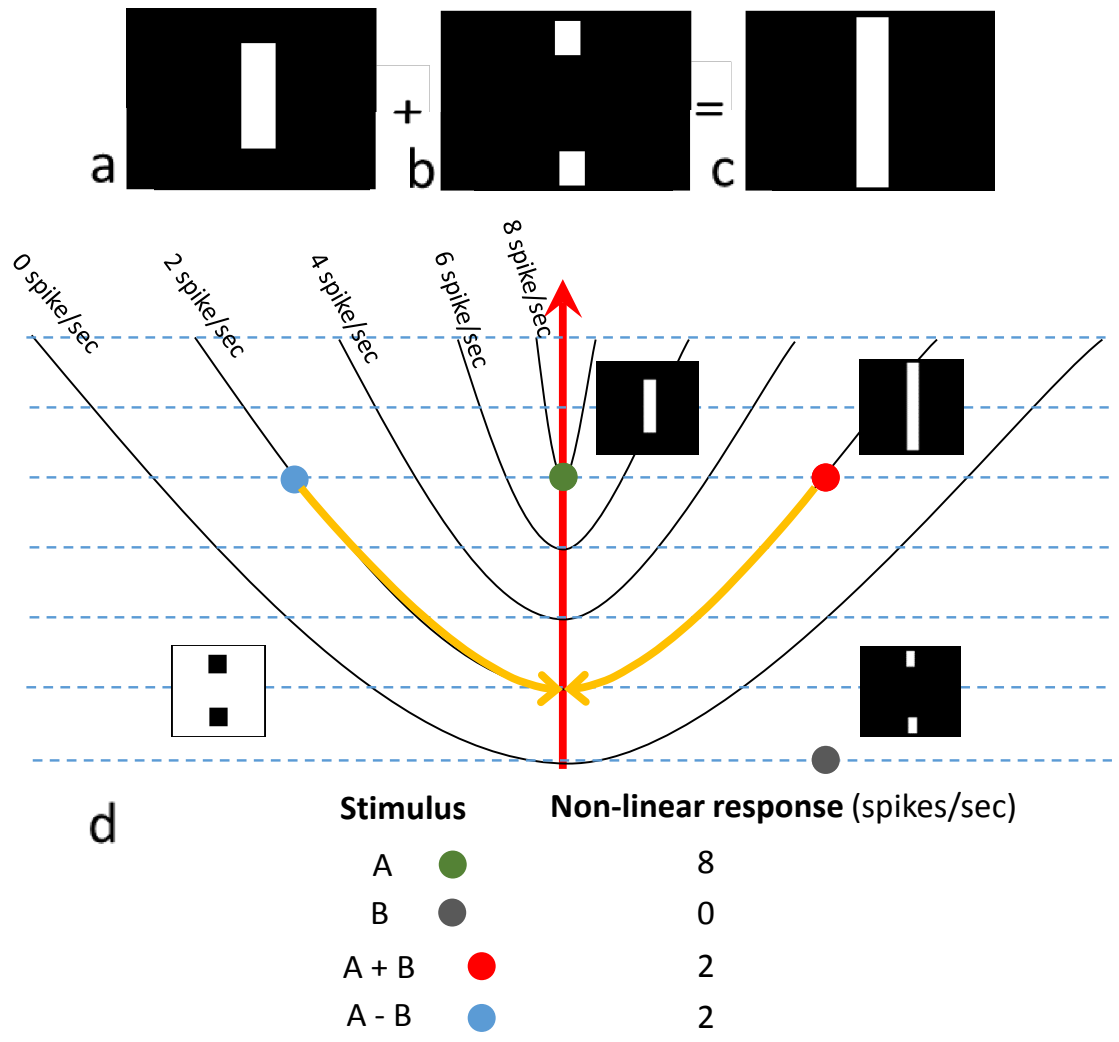

Figure 1.3: Endstopping in V1 cells is explained by curved isocontours of the neuron's response (Golden et al., 2015). a), b) and c) Example images used to illustrate endstopping. d) A state space description of endstopping. Curved isocontours describe the neuron's nonlinear selective response much more accurately than straight isocontours for a linear neuron.

iso-response contours. A neuron's response is described as phase invariant if it does not change with the phase of a grating stimulus (Movshon et al., 1978). If the neuron fires strongly to a white bar in between two black bars on a gray background, phase invariance means that it will also fire strongly to positional shifts

of the white and dark areas. More precisely, these stimuli may be parameterized as Gabor functions, which are exponentially modulated sine waves. As the phase of the sine wave is changed, the white and black areas of the image will shift in position. Smoothly changing the phase of an image of a Gabor function from $0 - 2\pi$ rad will trace out a circle in the image state space. In Fig. 1.4, the phase is plotted in two dimensions of the image state space and the neuron's response is plotted as a surface on the z-axis as well as isocontours in the state space. In Fig. 1.4a, the response at a number of different phases is plotted, and it shows complete phase invariance and its characteristic circular isocontours, where every phase of a particular grating at a given RMS contrast is the optimal stimulus. Fig. 1.4c shows elliptical isocontours, indicative of tolerance. These elliptical or circular isocontours are the signature of tolerance/invariance, and their quantification in high dimensions will be described below in Ch. 4. A linear neuron with a threshold shown in Fig. 1.4d has no invariance to phase. Fig. 1.4c shows a neuron with isocontours that curve away from the optimal stimulus vector, which is the different from the curvature seen in Figs. 1.2 and 1.3. This type of curvature causes the response of a neuron to fall of more slowly than the linear neuron of Fig. 1.4d, which means the response is therefore only somewhat tolerant.

What is the origin of these selective and invariant nonlinear responses? (It may be more appropriate to ask why they should be expected to be linear.) Given the assumption of a rate code, a lower threshold is already there, and considering the biochemistry of action potentials and a neuron's refractory period, there is an upper threshold as well. Neurons generate action potentials based on a nonlinear summation of the electrical signals received on their dendritic tree. The engineer's universal idealization of linearity is never valid when applied to a real biological system. However, these are mechanistic causes of general nonlinearities in neurons.

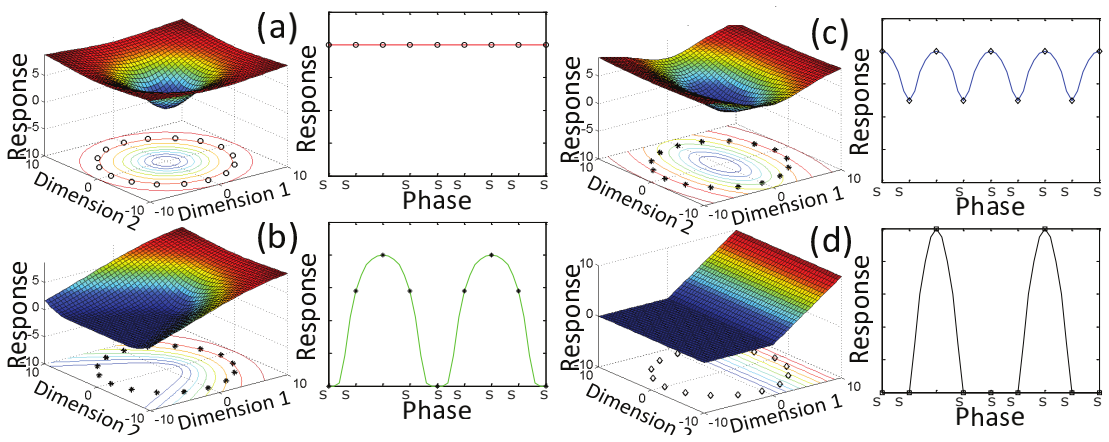

Figure 1.4: A model for tolerant and invariant responses due to phase shift of a grating in a 2D subspace of image state space (Golden et al., 2015). a) Perfect phase invariance. b) Weak phase tolerance. c) Strong phase tolerance. d) Phase response of a linear neuron with threshold.

As Rumelhart et al. (1988) found, nonlinear responses allow classifiers to have curved decision boundaries, and therefore make much more interesting and complex computations. We will argue that the specific selective and invariant nonlinearities of visual cortical neurons allow the brain to accomplish the computational feat of object recognition (Golden et al., 2015).

In order to claim that we understand what V1 neurons are doing, it is necessary to build a computational model that predicts the responses of real neurons to visual stimuli. The general approach has been to tailor computational models to specific types of stimuli, such that each model accounts for one type of nonlinearity with reasonable accuracy. This has been described as something of a "bag of tricks" approach, and models of even V1 neurons fail rather miserably when the stimuli are as unconstrained as natural movies (Vinje and Gallant, 2000; Köster and Olshausen, 2013). We aim to provide a conceptual unification of these disparate nonlinearities

by quantifying their geometry in high dimensions, and although this approach does not yet engage with the experimental data at the level of time series prediction of a single neuron's activity, it offers a quantitative method for describing many of these nonlinearities.

We will argue that the sparse coding network (Olshausen and Field, 1996) is one such model. The principle underlying this model was that a neural representation ought to have the same statistical structure as the stimuli it encodes. Evidence that natural images were composed of sparse sets of edges to a first approximation (Field, 1987, 1994) was used by Olshausen & Field (1996) to build a neural network that captured this sparse structure, and as a result learned V1-like receptive fields. This network finds an overcomplete representation through lateral inhibition between units, a process that is fundamentally nonlinear, and it has been found to reproduce non-classical effects like endstopping (Zhu and Rozell, 2013). The lateral inhibition of the network ensures that units will exhibit nonlinear selectivity in a way that closely resembles what happens in V1 representations. Thus, it seems selective nonlinearities emerge from a simple network that captures important statistical structure in natural scenes.

In the same way, V4 and IT responses capture high-order statistical structure that forms a representation which can be used to recognize objects (Rust and DiCarlo, 2012), and we argue that this requires simultaneous selective and invariant nonlinearities due to stronger curvature than what is found in V1 responses. Following Edelman (1999), Rao & Ruderman (1999), Field & Wu (2004) and Cox & DiCarlo (2007), it is evident that for a particular object (e.g., a particular face), all possible images of it (translations, rotations, scaling, etc.) lie on a smooth manifold in image state space. Another object will have its own manifold,

and the manifold of object A may come close to the manifold object B in state space. In order for the brain to recognize an object A as distinct from object B, it must determine which manifold the image in the field of view lies on. As the manifold become more similar, the task of discriminating between them becomes more difficult. At the same time, the brain must be tolerant enough to changes in the image to recognize novel views as lying on the same manifold as other images of the object that it has represented before.

The isocontours of a neuron's response in image space also describe a manifold. Although only 2D toy examples of selective and invariant nonlinearities are shown in Figs. 1.2-1.4, the response manifolds of real visual cortical neurons are high-dimensional. Most of these neurons are selective in certain dimensions of the state space while simultaneously invariant in others. We believe that the selective nonlinearities that have been observed are those that allow the brain to discriminate between image manifolds that are close in state space, and the invariant nonlinearities ensure that novel transforms of an object image are represented on the same manifold as previously represented images of that object. The network of these high-dimensional response manifolds in image state space allows the human brain to carry out object recognition with a level of accuracy far beyond what has been achieved with artificial systems (Girshick et al., 2014).

Here, I present a method for probing and quantifying this curvature in high dimensional state space. I utilize methods from differential geometry to measure the curvature of neurons in the original sparse coding network, which exhibits primarily selectivity, as well as the Karklin & Lewicki network (2003, 2005), which has neurons that exhibit strong degrees of simultaneous selectivity and invariance. I identify the sign of principal curvatures in high dimensional space as an indicator

of selectivity or invariance, the magnitude as a measure of how selective or invariant a neuron is in that dimension and the direction as the feature to which the neuron is selective or invariant. I argue that these methods could be applied to experimental measures of responses of visual cortical neurons, and that this approach could unify the field of scattered models for each observed nonlinearity.

CHAPTER 2

THE SPARSE CODING NETWORK

The ultimate goal of the investigation of the mammalian visual system is to understand how light impinging upon the retina is transformed to behavior. A more limited goal is to understand how photoreceptor activity is transformed to representations that are useful for executing behavior. In practical terms, an immediate project the field has taken on is that of accurately predicting the responses of neurons in visual cortex to image stimuli. When the model successfully captures the important computations of the neuron or population, this is evidence that the optimizations embodied in the equations of the model are the same principles that give rise to the computations in physiology. The "efficient coding hypothesis" emerged from this approach and has been a fruitful source of predictions for both modeling and physiology (Barlow, 1972; Field, 1994; Simoncelli and Olshausen, 2001; Olshausen and Lewicki, 2014).

The field at present is primarily engaged in two endeavors: modeling computational and physiological properties of neurons to predict the output of single units or populations over time (Pillow et al., 2008; Zhu and Rozell, 2013; Köster et al., 2014; Pagan et al., 2013), or engineering neural networks to automatically label objects in an image (Le and Ng, 2013; Krizhevsky et al., 2012; Girshick et al., 2014). The sparse coding network has influenced work in both of these areas, and as a simple model for a network of nonlinear visual neurons it matches many experimentally observed properties of V1 with a parsimonious set of assumptions (although it falls short of accurate biophysical predictions for single units).

We chose the sparse coding network as a model system in order to explore the curvature properties of the response surfaces of neurons in image state space.

Before we present that analysis, in this chapter we will provide some history behind the development of the model and show the derivation of its equations for the efficient representation of natural image data. It is important to understand the basic mathematics of the sparse coding representation in order to appreciate the curvature analysis in later chapters. We also introduce a more complex version of the sparse coding network which is thought to capture subtle nonlinearities found in V2 and V4 neurons. This network will also be explored in terms of its curvature in image state space.

2.1 Redundancy and Efficient Coding of Natural Images

The efficient coding hypothesis is the idea that sensory systems have evolved to efficiently encode the statistics of the world in which they exist. It has its roots in the work of Attneave (1954), who first applied Shannon's (1949) results about quantifying information in signals to how the brain may encode sensory data. Kuffler (1953), Hubel & Wiesel (1959, 1962, 1968), and other physiologists first measured the responses of V1 neurons to visual stimuli. Marcelja (1980) connected the work of Gabor (1946) on the wavelet transform to a hypothesis about the information captured by V1 receptive fields. Barlow (1972) presented a quantitative argument that sensory systems ought to have evolved in order to reduce redundancy in neural representations in line with Shannon's efficient communication systems.

Sherrington (1941) and Konorsky (1967) made important advances about how neurons in principle ought to optimally represent high-level information, like the identity of an object in the visual field. Given an average stimulus to be represented by a population of neurons, the stimulus could be encoded by activity in

anywhere from only one neuron to half of all the neurons in the system. A single "grandmother cell" is not robust to damage to the system, and a completely distributed representation can be wasteful in terms of the energy required for each representation state. Representation in the brain ought not to follow either the true grandmother cell scheme or the fully distributed scheme: the sparsity of the representation was an open experimental question (Barlow, 1972).

Field (1987, 1994) found experimental evidence for an aspect of Barlow's redundancy argument from a statistical perspective. Field (1987) found a striking signature of redundancy in a class of images that he termed "natural scenes". On average, two images of the natural world are far more statistically similar to each other than any pair of random, white-noise images. Natural scenes are typically composed of objects and edges. There is redundancy in the pixel-by-pixel intensity data as a result, because given a particular pixel, a nearby pixel ought to have roughly the same intensity due to the likelihood that that second pixel is part of the same object. Attneave (1954) made the point that this was a visual analogue of Shannon's "guessing game" which demonstrated the redundancy in written language, where it is quite easy to guess the letter following "TH" on average.

Field quantified this redundancy in natural scenes using the Fourier power spectrum of each image, which is the two-point, two-dimensional autocorrelation function for an image transformed to the Fourier domain. The autocorrelation function falls off as a function of distance, and this is reflected in the amplitude of the Fourier spectrum as a falloff with the inverse of the value of the spatial frequency, or $1/f$, as it has been denoted in work on scale-invariant signals (Pentland, 1984). The intensity of pixels in white noise images is generated randomly and independently for each pixel, so the autocorrelation is zero for any distance, and the

Fourier amplitude spectrum is flat (equal energy at all spatial frequencies). The surprising result was that every natural image, whether a forest scene, a beach view or a mountain valley, had an approximately $1/f$ Fourier spectrum. Natural images and their spectra, as well as a white noise image filtered to match the $1/f$ amplitude, are shown in Fig 2.1. The deviation from a flat spectrum is a sign of the redundancy or predictability of the image signal. The $1/f$ spectrum of natural images is a fundamental result about the statistical structure of the visual world, and it reveals that there is a great deal of redundancy in natural scene images. The visual system can form an efficient code by taking advantage of this redundancy.

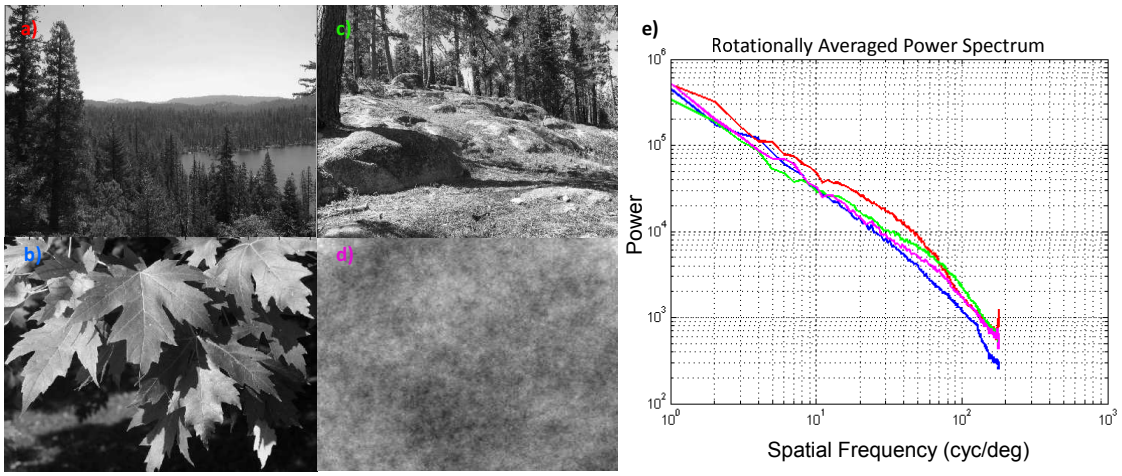

Figure 2.1: a-c) Natural scene photographs. d) A white noise image filtered to have a $1/f^2$ power spectrum. This $1/f$ noise image has the same two-point correlations as natural scenes, but lacks the higher-order dependencies that are due to extended contours and whole objects. e) The power spectrum of each photograph, averaged over all orientations, corresponds closely to $1/f^2$, which has a slope of about -2 on this log-log plot. There is surprising statistical similarity in the power spectrum given the visual differences in the natural scenes.

Field (1987) continued with an argument based on Barlow's hypothesis that

the mammalian visual system must take advantage of this redundancy in encoding the visual world. He argued not for redundancy reduction, but for accurately encoding the information with a small subset of neurons. The receptive fields of V1 neurons were modeled as a series of Gabor functions of varying bandwidth, spatial extent and phase. Marcelja (1980) argued that Gabor functions provide the optimal tradeoff between localizing a visual signal in terms of its position in the visual field as well as its spatial frequency spectrum. These Gabor sensors could further be tuned to represent most of the information of a particular image in a small set of active neurons compared to the total number in the model system. The philosophical point of the Field's argument was that there is statistical structure in natural scene images, and there is a straightforward way for neurons encoding visual information to represent it efficiently.

The statistical structure that was revealed by the $1/f$ amplitude spectrum of natural images is a finding about correlations between pairs of pixels in images. There is also structure in higher-order dependencies in natural images - for example, the intensity of three pixels along an edge between two objects are likely to be statistically dependent on one another. Theoretically, in order to recognize objects, a system must compare a set of arbitrarily high n^{th}-order dependencies present in an image to the high-order dependencies for objects it has seen before. These dependencies are not as easy to measure as the Fourier spectrum, but because of their role in object recognition they are indirectly the subject of much investigation (Zetzsche and Krieger, 1999). Olshausen & Field (1996) bypassed this somewhat intractable measurement step and developed a neural network to learn representations of higher-order dependencies present in natural scenes. The learned higher-order structure of the network would be present in both the receptive fields that the network learned as well as the n^{th}-order dependencies between

subsets of neurons in the network. The argument here has been refined to include the point that this is equivalent to the idea that high-order dependencies in the images are represented in the curvatures of the response manifolds of each unit in the network, as I will detail below.

Field (1994) argued that the Gabor-like structure of V1 receptive fields was a solution to efficiently encoding high-order structure in natural images. He demonstrated that the statistical signature of linear responses of Gabor filters to natural images was a heavy-tailed or highly kurtotic, non-Gaussian distribution of response values. In other words, the responses were "sparse", in that neurons usually did not fire, but infrequently fired with a large magnitude, and each neuron had an equal probability of firing on average. More precisely, the fourth moment of these distributions was much greater than that of a Gaussian distribution with the same variance. This sparse, distributed code, with many neurons that were mostly silent, was contrasted with a compact code with a smaller number of neurons based on the Fourier or principal components that had Gaussian activation distributions and were therefore more active on average. This was an effective argument against the idea that V1 neurons were merely computing two-point correlations via Fourier decomposition, which fit with the idea that regions further up the cortical hierarchy will be learning higher-order dependencies. This was also an empirical and computational effort to address the grandmother neuron question: Field's results came down strongly on the sparser side of the spectrum, at least for V1 neurons.

The theory for a sparse, distributed code was presented by Olshausen & Field (1996) as a model that captures more of the high-order structure in natural images than the Fourier power spectrum. The Fourier/PCA code finds the orthogonal directions of the highest variance (two-point correlations) in the data. Algorithms

that found the principal components of natural images did look somewhat like receptive fields in that they were selective for spatial frequencies, but did not have a distribution of orientations and were not spatially localized (Hyvärinen et al., 2009). To illustrate the shortcomings of the principal components analysis (PCA) approach, Field 1994 used an example of a toy "sparse" dataset where PCA will find those directions of highest variance, but they will ultimately be misleading about the true causes of the data, which are the sparse components. In contrast, a code that restricted the number of active units to represent any one data point would find these causes.

(Olshausen and Field, 1996) created a neural network that learned a sparse code on natural images and produced V1-like receptive fields localized in position and spatial frequency with a distribution of preferred orientations. The first insight that allowed the network to learn plausible receptive fields was the sparse constraint on the distribution of neuron activations. The second was that the network could be "overcomplete", with more sensors than the dimensionality of the input data. This was modeled on the human visual system (HVS), where the information in a visual scene is represented in 1.5 million axons from retinal ganglion cells and expanded into 100 million V1 neurons (Barlow, 1972).

The network begins with random receptive fields that look like noise, and gradually adapts the receptive fields to minimize the reconstruction error of the image representation balanced against the cost of the firing of neurons (Olshausen and Field, 1997).

$$E = [reconstruct\ image] + [penalty\ on\ activations] \tag{2.1}$$

A series of images are represented with random receptive fields, the energy of the representations are calculated, and the receptive fields are changed to decrease the energy. This is done iteratively over many sets of input images.

The receptive fields (technically "basis functions", although usually quite similar to the receptive fields determined by the spike-triggered average response) are stored in the columns of a matrix Φ. The input natural scene image is reconstructed from a weighted sum of all the basis functions, Φ*x, where x is a vector with each entry corresponding to a weight on a basis function. Better representations have a sparser array of activations x, which is imposed by the cost function. An optimal representation minimizes this energy function, so the reconstructed image Φ*x is nearly the same as the original image, and has a low cost for the vector of activations x. λ is a parameter that balances the relative importance of reconstruction error and sparsity.

$$E = [I - \Phi * x]^2 + \lambda * \sum cost(x) \tag{2.2}$$

A "critically-sampled" network, where the number of basis functions (columns in the matrix Φ) is equal to the data dimensionality, forms a linear representation of the data. The number of variables is equal to the number of equations, so there is only one representation that minimizes the squared error including the sparse cost function. This is the algorithm for independent components analysis (Bell and Sejnowski, 1997). Sparse coding, as an overcomplete system, has more equations than variables, and therefore has an infinite number of solutions; the network settles on a particular representation vector x through an iterative gradient descent procedure, although this is not guaranteed to be the globally optimal solution, and will be different on each trial due to randomized initial conditions.

The sparse coding algorithm iteratively finds the best representation through a nonlinear process, whereas the ICA algorithm determines how the network will represent a given image using a single linear matrix inverse. The sparse coding network is a better model for natural images than ICA because it is more general and therefore better captures the structure of the data. When the sparse coding network is built with the number of basis functions equal to the data dimensionality (as well as the assumption that the system is noiseless), it is exactly equivalent to ICA (Olshausen and Field, 1997). Overcomplete versions of ICA have since been developed (Lewicki and Sejnowski, 2000).

The other primary advantage of an overcomplete network is that the unit activations are a nonlinear function of the input image. A nonlinear representation is achieved through lateral inhibition between neurons in the network, and, as argued in Ch. 1, nonlinearity is an essential feature of real visual neurons. Zhu and Rozell (2013) demonstrate that neurons in the sparse coding network of Olshausen and Field (1996) exhibit endstopping and surround suppression, two nonlinear effects that have been observed in cortical neurons and deemed extra- or non-classical. We will argue below that the geometric approach to measuring curvature in the responses of neurons in the high-dimensional state space gives an explanation for why these nonlinear responses arise (Golden et al., 2015).

2.2 Derivation of the sparse coding network equations

The energy equation from the sparse coding model can be derived from first principles using a Bayesian formulation (Olshausen and Field, 1997). An image I is represented by a linear combination of basis functions Φ weighted by x:

$$I = \Phi * x + noise \tag{2.3}$$

I is originally an $\sqrt{N}$x$\sqrt{N}$ matrix, but is converted to a vector with N entries (that is, a 10x10-pixel image patch will be converted into a vector with 100 components). Φ is a matrix where each column is a vector with length N representing a receptive field (Φ will be 100x($100 * OC$), where each column can be reshaped to 10x10 pixels and viewed as an image, and OC is the degree of overcompleteness). x is a vector of length N, where each entry represents the weighting on a corresponding receptive field or column in Φ. The product $\Phi * x$ results in a vector of length N; this is the network's reconstruction of the image. If we know receptive fields Φ, we need to find an equation that results in the activations x for one image patch; and, ultimately, we want to find another equation that learns the receptive fields Φ over thousands of image patches.

In order to devise learning rules for x and Φ, we formulate the representation as a generative Bayesian model. The probability of generating an image from the model can be found by integrating the product of the likelihood for the image given a set of coefficients x and the probability of selecting those coefficients over all possible coefficients.

$$p(I) = \int p(I|x) * p(x)dx \tag{2.4}$$

If we assume that the likelihood $p(I|x)$ for representing an image I with coefficients x is noisy around the best reconstruction, it will be Gaussian. λ is an estimate of the standard deviation. Note that technically equation 2.4 and the equations below are proportionality relations, as the integral of the PDF does not

need to be normalized to have area 1 for our purposes, but here an equal sign is used for simplicity.

$$p(I|x) = \exp\left(\frac{-[I - \Phi * x]^2}{2\lambda^2}\right) \tag{2.5}$$

In terms of the prior, as has been argued, we will assume a sparse distribution for the coefficients on the basis functions. There are a number of functions that can be used for sparse probability distributions (for L^1 norm minimization), and here we will use the Laplace distribution.

$$p(x) = \exp\left(-\sum |x|\right) \tag{2.6}$$

Integrating over the distribution is not practical as above in 2.4, so instead we take its maximum likelihood value as an approximation, which is equivalent to minimizing the negative of the log of the posterior probability.

$$p(I) \approx arg\ max_{x,\Phi}[p(I|x) * p(x)] \tag{2.7}$$

$$p(I) \approx arg\ max_{x,\Phi}[\exp\frac{-[I - \Phi * x]^2}{2\lambda} * \exp\left(-\sum |x|\right)] \tag{2.8}$$

$$\log p(I) \approx arg\ max_{x,\Phi}[(\frac{-[I - \Phi * x]^2}{\lambda}) - (\sum |x|)] \tag{2.9}$$

This is the energy equation 2.2 presented above. The learning rules for Φ and x result from taking the derivative of this, setting it equal to zero and solving.

$$\log p(I) = [I - \Phi * x]^2 + \lambda * \sum cost(x) \tag{2.10}$$

This is the gradient descent equation for finding the vector x for each image iteratively. x is found by evaluating this equation, which results in a vector x_{update}, and $x_{new} = x + x_{update}$. This update is made some repeatedly (perhaps 100 iterations) and the final value is used as the representation. Note that if we assume

a square matrix Φ and no noise ($\lambda = 0$), the learning rule becomes the matrix inverse $x = inv(\Phi) * I$. This is called the "inner loop" step. The representation x is adapted using the vector determined by the error in the reconstruction and the derivative of the sparse cost on the representation.

$$\frac{d}{dx}[\log p(I)] = \Phi^T[I - \Phi * x] + \lambda * \sum \frac{d}{dx}(cost(x)) \tag{2.11}$$

A similar procedure results in the equation for Φ:

$$\frac{d}{d\Phi}[\log p(I)] = [I - \Phi * x]x^T \tag{2.12}$$

This results in a matrix Φ_{update} where $\Phi_{new} = \Phi + \Phi_{update}$. Φ is updated over thousands of generations in the "outer loop" step. When the algorithm is run, the matrix of basis vectors Φ is initialized randomly. The outer loop begins, and within it the inner loop is iterated through a number of times, where an image is presented to the network, and, given Φ, the components of x are learned to most sparsely represent the image with an accurate reconstruction, minimizing the energy of the representation. This is done for a large number of images, and then the outer loop update occurs, where errors in representation are used to update Φ. The outer loop is then run through again until the basis functions in Φ no longer change.

The basis set Φ was initially a square matrix; if the images were 10x10 pixels, or vectors with 100 elements, Φ was 100x100, or had 100 10x10-pixel basis functions. An overcomplete basis has more basis functions than dimensions in the data, so when Φ has more than 100 columns (although still 100 rows) it is overcomplete. The differences in the basis functions learned by overcomplete sparse coding and ICA are subtle. The real advantage of the overcomplete sparse coding representation is that the inner loop is nonlinear and reproduces effects like endstopping

(Zhu and Rozell, 2013). This is clear evidence that the sparse coding network better captures the response properties of V1 neurons.

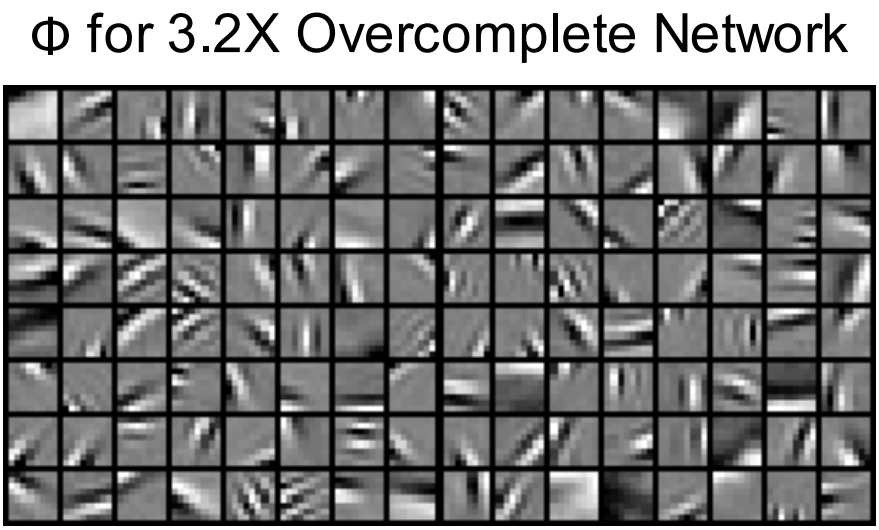

Figure 2.2: The basis set Φ determined by the sparse coding algorithm for 8x8 natural scene patches. This network had 200 basis vectors and was therefore 3.2X overcopmlete.

2.3 The Karklin-Lewicki Network

The principles of efficient coding led to the development of the sparse coding network, and the array of V1 response properties that sparse coding neurons reproduce is a testament to the soundness of those principles. What about the response properties of neurons beyond V1? Recent evidence points toward texture-like processing in V2 (Freeman et al., 2013), contour/figure selectivity in V4 neurons (Carlson et al., 2011), and object selectivity and tolerance in IT representations (Rust and DiCarlo, 2012). There has been a concurrent proliferation of neural network models that attempt to capture these response properties with varying degrees of success (Serre et al., 2007; Le and Ng, 2013; Yamins et al., 2014). Karklin

& Lewicki (2003, 2005) developed a straightforward extension of the sparse coding model that was able to capture some of the response properties of neurons in V2 and V4. Their idea was to simply employ the same framework as a second layer to learn the dependencies of the first layer representations. It was a straightforward application of the principles of efficient coding, but it produced second-layer neurons with interesting selective and invariant response properties. Below in Ch. 4, we describe this network in terms of curved geometry, which is worthy of investigation due to the increased complexity of the response surfaces far beyond what is observed in sparse coding neurons.

The Karklin-Lewicki network is based on the observation that the representations of natural scenes learned by ICA do not live up to the algorithm's name: Olshausen noted that "one is left with the awkward task of modeling the 'dependencies among the independent components' " (Olshausen, 2008). Since the system is linear and because the algorithm imposes a sparsity constraint on the representation, statistical independence is always sacrificed. Karklin & Lewicki (2003, 2005) devised a two-layer network where the second layer would learn scaling factors for the first-layer activations in order to make them more independent. Increased independence of activations was accomplished by forcing a second layer of neurons to learn a vector λ for each image, instead of the single user-chosen value as in the sparse coding network. Karklin & Lewicki called this the "variance components network", as the vector λ^2 encodes the components of variance of the sparse prior. The learned λ values by the second layer result in a more independent set of scaled activations for the first-layer neurons, so in a sense the second layer is carrying out a gain control operation (Schwartz and Simoncelli, 2001); note that this is distinct from the type of observed gain control where the saturating level of contrast response varies with grating frequency, as in (**?**), and is instead learned in

a generative framework. They found that the learned second layer neurons show a host of simultaneously selective and invariant responses, so the response surfaces of these neurons in high dimensions are even more complex than those of the single-layer sparse network.

The variance components network was designed in such a way as work with an overcomplete first layer like that of the sparse coding network. The sparse coding network is formulated as a matrix Φ with basis functions in the columns, and an image I is represented by the linear combination $\Phi * x$, where x is a vector of activations for each basis function. The activations of units in the sparse coding network over natural image data also show an analogous shortcoming to ICA's activation dependence (Karklin and Lewicki, 2005). The sparsity of the representation for a particular image can also be sub-optimal, because the distribution of coefficients in x is fixed by the chosen sparse distribution, like the Laplacian, $p(x) = \exp(-\frac{x}{\lambda})$. A choice of λ fixes the sparsity of x on average, and Karklin and Lewicki showed that individual image patches deviate from this degree of sparsity. In order to make the first layer more flexible, they designed the second layer to determine a vector of optimal λ parameters, one for each basis function. The second layer is governed by the linear system $\lambda^2 = \Psi * v$, where λ is the vector that determines the sparsity of the distribution, Ψ is a matrix of the "variance basis functions" and v is the vector of activations on the second-layer basis functions. During the "inner loop" step, both sets of activations x and v are learned iteratively and alternately in small batches. The outputs x are then divided by the square root of the outputs $\Psi * v$. For the "outer loop", the basis functions are also learned iteratively and alternately.

The learned first-layer basis functions turn out to be similar to those of the

original ICA and sparse coding networks (with some differences in the distributions of properties like size, bandwidth, orientation, etc.) (Karklin and Lewicki, 2003, 2005). The second-layer basis functions successfully make the first-layer activations sparser. The second-layer basis functions are more difficult to interpret than those of the first layer, because each is a vector of weights on first-layer basis functions, and therefore not image features themselves. Karklin & Lewicki found that they frequently consist of large weights on first-layer basis functions with similar properties, such as position, orientation or spatial frequency bandwidth.

They tested the activations of the second-layer basis functions over different types of natural images and found that their responses exhibited both selectivity and invariance. For example, given a large image, the most active basis function from the first layer changes with every pixel; however, for the second layer, the most active basis function is clustered spatially, as in Fig. 2.4.

The second-layer neurons perform a type of gain control on the responses of the first layer. This is somewhat analogous to the idea of divisive normalization, where the output of a neuron is divided by the outputs of a set of other nearby neurons (Schwartz and Simoncelli, 2001). The variance components network solves multiple problems at the same time: it proposes that gain control may occur in order to increase independence or sparsity in V1, and it extracts a new set of statistical properties from natural scene data in order to enforce greater independence in a way that replicates how higher-order neurons respond to natural scenes. An improved version of the network has also recently been created to take into account spatiotemporal variation (Cadieu and Olshausen, 2012).

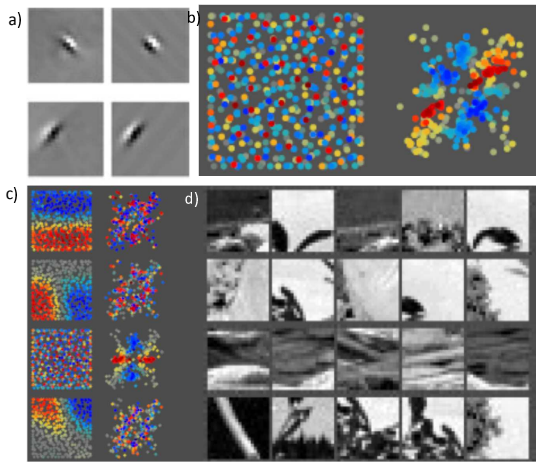

Figure 2.3: From Karklin & Lewicki, (2003; 2005). a) First-layer basis functions from the variance components network resemble V1-like filters from the sparse coding network. b) A visualization of a learned second-order basis function (a column of Ψ) that has response properties similar to a V2 or V4 neuron. Each dot on the left represents the center of a first-layer basis function (in image space), and the dot's color represents the weight of that first-layer basis function by the second-layer basis function. On the right is the same representation, but each dot represents the center of the first-layer basis function in Fourier space. Therefore this basis function is selective for high-frequency structure oriented at 45° over the whole image. c) Four other basis functions and d) the images that cause them to fire the most. Basis 1 is activated by spatial detail in the bottom half of the figure and inhibited by detail in the top half; basis 2 prefers spatial detail in the lower left and is inhibited by detail in the lower right; basis 3 is activated by horizontal spatial frequencies and inhibited by vertical frequencies. Basis 4 is activated by detail in the top left and inhibited by detail in the top right.

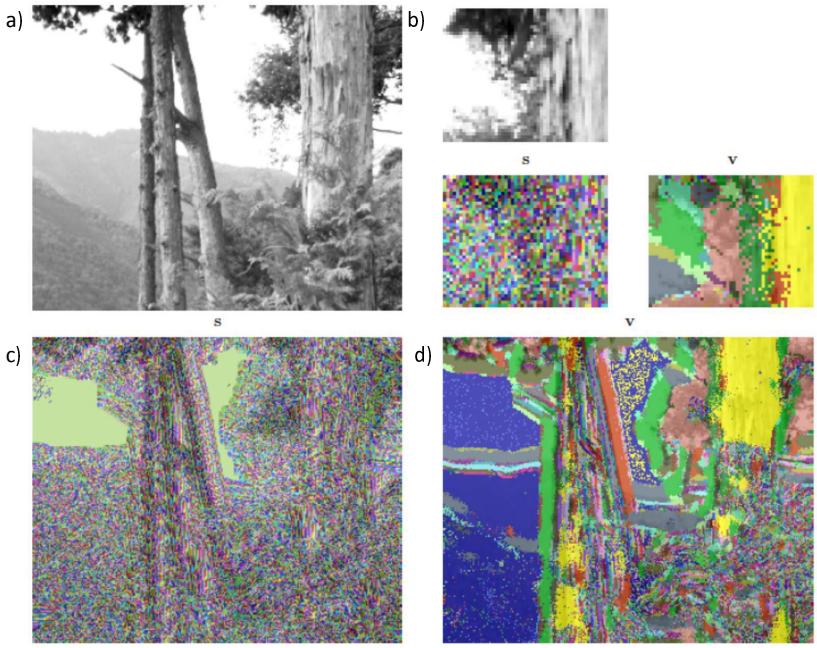

Figure 2.4: Karklin, (2007). a) a natural scene image. b) a small natural scene image colored by the basis function that has the maximum activation for that pixel. The multicolored map shows the basis function with the maximum activation in the first layer; the more solid colors show the basis function with the maximum activation from the second layer. c) the full max activation map for the first layer; d) the full max activation map for the second layer. Note the increased invariance of response from neurons in the second layer.

CHAPTER 3

THE CURVATURE OF THE SPARSE CODING NETWORK: 2D SUBSPACES

The sparse coding network was designed in order to demonstrate that the types of receptive fields that are observed in V1 could be learned by a neural network trained on natural images. The network found an efficient representation based on the particular statistics of the training set. The Gabor-like basis functions are evidence that natural scenes are composed of sparse linear combinations of Gabors; to a first-order approximation, these are localized, oriented edges of different bandwidths. Olshausen & Field (1996) showed that if the training set was composed of sparse linear combinations of gratings or dots, the network learned basis functions of gratings or dots. The match to V1 receptive fields is a result of the optimization of the right choice of statistic: Field Field (1994) had argued that sparseness was the key, and the sparse coding network verified this was correct. It was a victory for the efficient coding framework, because once efficiency was defined as sparseness, and the most efficient representation of a natural scene dataset was found, V1-like receptive fields emerged.

As described in Section 1.2, the receptive field is an important statistical summary of a V1 neuron's response. However, we have also seen that the majority of V1 neurons have interesting nonlinear responses that are not captured by the classical receptive field measure. The receptive field is a vector in the image state space, but the nonlinearities are evident in the curved isocontours throughout the state space. Although the most important results of the sparse coding network are its receptive fields, the representation is a nonlinear function of the input when the basis set is overcomplete. Once the basis functions have been learned, the responses

of the network can be found by iterating through the inner loop. The network can be treated like a physiological system, and its responses can be probed with different sets of stimuli. This was the approach taken by Zhu and Rozell (2013), who found that the sparse coding network exhibited several nonlinear responses like endstopping that are well known from physiological experiments. Their experiment is useful as a measurement of the nonlinear responses, but they do not provide a clear argument for why endstopping occurs.

In this chapter, the nonlinear aspects of sparse coding neurons are quantified by applying the theoretical ideas about the curvature of response surfaces from Ch. 1. Here, we probe the nonlinearities generated by the sparse coding network, first in a toy 2D state space, and then extend that approach to 2D subspaces of image state space. We measure the curvature of response surfaces of neurons in these networks in order to quantify their nonlinearity. We measure curvature as a function of three parameters: the angle between basis functions, the degree of sparsity set by λ and the choice of the sparse prior/cost function. In later chapters, we will expand this same analysis to high-dimensional image space. The sparse coding network learns a rich, nonlinear representation in this state space that has been largely unexplored due to its complexity. The network's representation is usually summarized by an image of its receptive fields, but its full range of nonlinear behavior requires a more nuanced investigation of response surfaces in state space.

3.1 The Sparse Coding Network in a 2D State Space

The sparse coding network was described in Ch. 2 as a model of image representation where an image I is represented by a set of basis functions Φ with corresponding activations x. When the sparse coding network is trained on whitened natural

images, it starts with a random basis set Φ. For the first batch of images, the coefficients x are learned using that random Φ in the "inner loop" step. Then, Φ is altered slightly according to the inefficiencies in the representation in the "outer loop" step. Over many iterations the basis vectors in Φ are altered and come out to be Gabor-like functions, localized in space and frequency and oriented over a distribution of angles.

We are interested in what the network does after it has learned Φ. In order to measure how the sparse coding network represents an image, it is necessary to iterate through the inner loop of the algorithm to find the coefficients x for the learned Φ. The efficient representation is found by minimizing the energy (eqn 2.2), which applies lateral inhibition between the receptive fields, so that similar basis functions inhibit each other (eqn 2.11). Since the network exhibits nonlinear responses, we applied the techniques from Ch. 1 in order to investigate the curvature of the response surfaces.

We created a sparse coding network in 2D using a toy data set to visualize the activity of the neurons over the whole state space. In Fig. 3.1, the x- and y-axes represent a subspace of image space, and the activation of the neuron is plotted on the z-axis as a surface, but visualized only as iso-response contours. We created a 2D sparse dataset with three causes (directions in which data points lie) and used a sparse coding network with three basis functions (3 basis vectors / 2 state space dimensions = 1.5X overcomplete) to encode the data (Fig. 3.1a). A two-vector description of these data will not result in an efficient representation according to the energy equation above (2.2), because the mismatch between the number of causes of the data and the number of encoding vectors prevents the representation of most data points from being sparse. A three-vector description, however, allows

the encoding vectors (basis functions) to be aligned with the data. Figs. 3.1b and 3.1c show the results of the sparse coding network when two values for λ (sparsity) are used. The λ parameter determines the sparsity of the representation at the expense of reconstruction error. A low lambda leads to a non-sparse representation, while a higher lambda leads to a sparser representation with more reconstruction error.

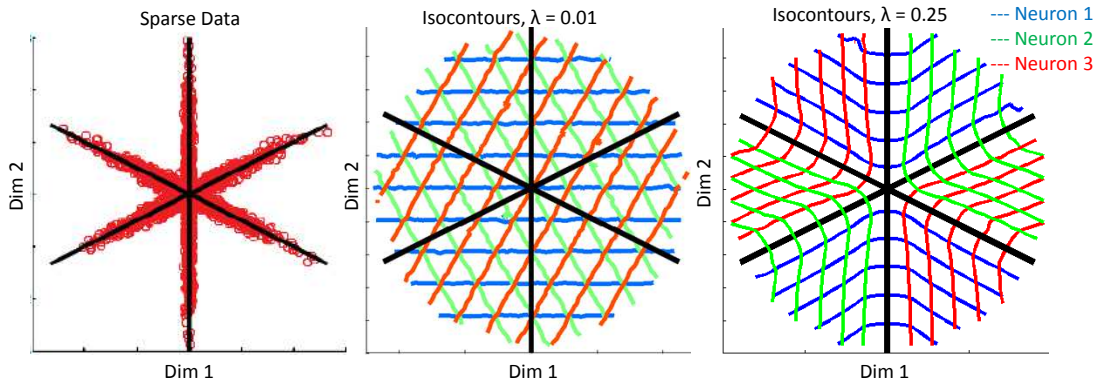

Figure 3.1: a) A scatter plot of 2D sparse data set with three causes. b) Iso-response contours for each neuron (vector) (red = neuron 1, green = neuron 2 and blue = neuron 3), where the response was determined using the inner loop with $\lambda = 0.01$. c) Iso-response contours for each neuron for $\lambda = 0.25$. Note the strong curvature resembling the theoretical curvature for a selective response in Fig. 1.2.

To characterize the network, we determined the iso-response contours of the encoding vectors by measuring the response of the network for points uniformly distributed throughout the 2D space (even though the efficiency of a sparse network is only possible because the data is not distributed normally, and the encoding vectors would not be likely to encode data from a uniform distribution; the response to uniformly-distributed data allows full characterization of the curvature in 2D). They are plotted for low and high sparseness (λ = 0.01, 0.25) for this 1.5X overcomplete network. The isocontours are not curved for the low λ condition; however, they are clearly curved for the high λ condition. This response of an overcomplete network (resulting from the sparsifying inner loop) produces the non-linear concave responses that were illustrated in Chapter 1. The response for an overcomplete network where λ is very low, however, results in a nearly linear output, because the inner loop is prioritizing reconstruction over sparsity (2.11), and the best reconstruction will be a linear projection of the stimulus onto each encoding vector. The sparse coding network alters the iso-response surfaces as lambda is increased in order to minimize the energy of the representation, and the minimal energy state depends on the choice of λ, so the two minimal energy states for the two values of λ result in different iso-response contours. In Figure 3.1c, when λ is high, no point in the state space is represented by more than two active neurons.

3.2 The Sparse Coding Network in 2D Subspaces of Image State Space

In order to examine the curvature of neural responses from the sparse coding network trained on natural images, we generalized the method used to create Fig. 3.1. Since the network trained on images operates in image space, the neural response surfaces are high-dimensional objects, and the easiest way to begin to quantify them is to examine relevant low-dimensional subspaces. We determined relevant subspaces based on hypotheses about the curvature of the neural response surfaces. We reasoned that curvature is likely to be higher in 2D subspaces between pairs of neurons, and therefore these are the subspaces we examined. We quantified their curvature as a function of the angle between basis vectors, the degree of overcompleteness and the chosen sparse prior/cost function.

In each case, measurements were made from four networks with varying degrees of overcompleteness that were trained on 8x8-pixel natural image patches, shown in Fig. 3.2. The networks that are more overcomplete have a more diverse set of basis functions than those that are less overcomplete. To examine the relationship connecting angles between encoding vectors and curvature, a first step is to measure the distribution of angles between all possible pairs of encoding vectors, as in Fig. 3.3.

By considering each basis function as a vector in image state space, the angle between two vectors is found by taking the inverse cosine of their inner product. Since the vectors in the sparse coding network can have negative activations, any two vectors will be at most 90° apart, because if the positive ends of two vectors are at 100°, then the positive end of one and the negative end of the other are

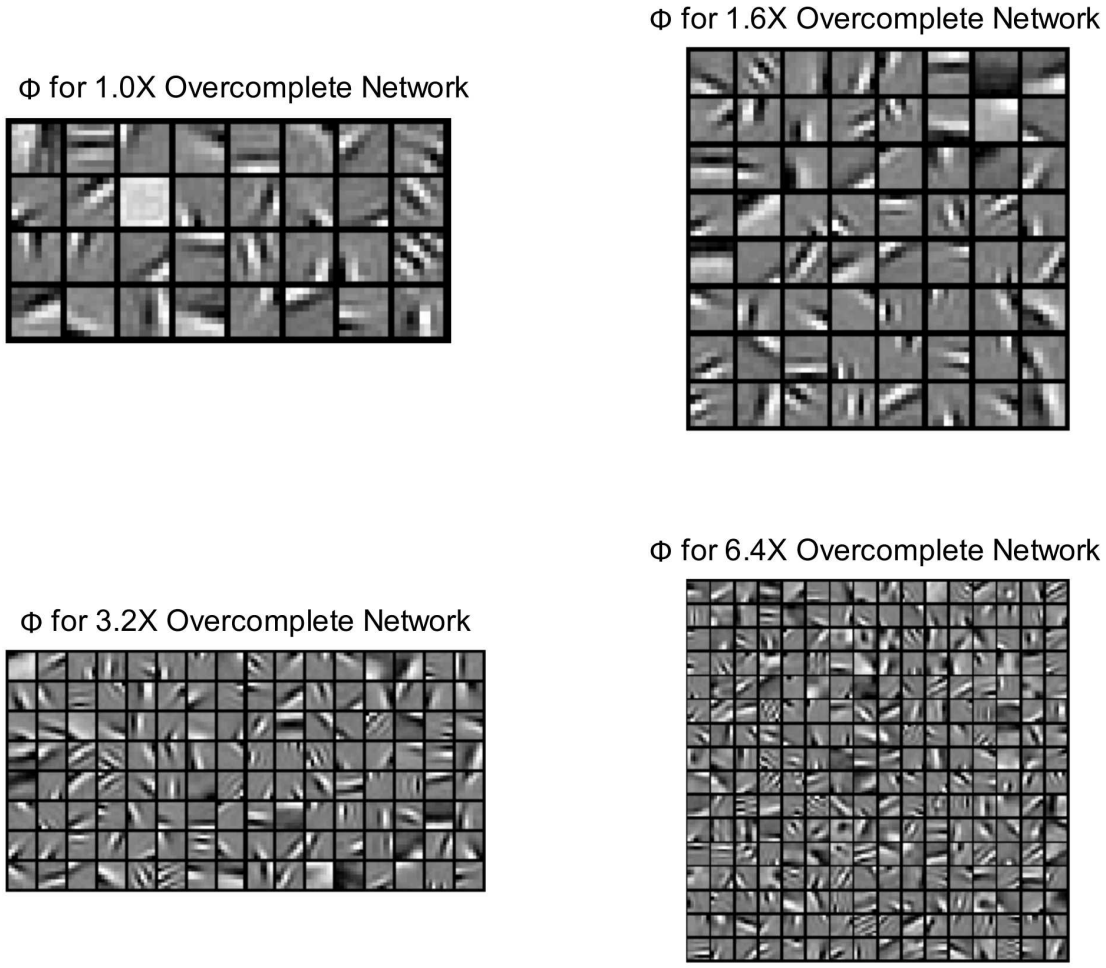

Figure 3.2: The four networks of varying degrees of overcompleteness used in the following measurements. Note that more overcomplete networks have different types of basis functions than less overcomplete networks, sensitive to higher spatial frequencies and curved contours.

at 80°. For these statistics, we have mapped angles from $0 - 180°$ into $0 - 90°$, so that any angle θ greater than 90° is transformed to $\theta_{new} = [90° - |90° - \theta|]$. Two randomly chosen vectors in a state space will be close to 90° on average, but depending on the dimensionality of the state space and the number of basis vectors, the mean will be closer or further from 90°. For the sparse coding network, Fig.

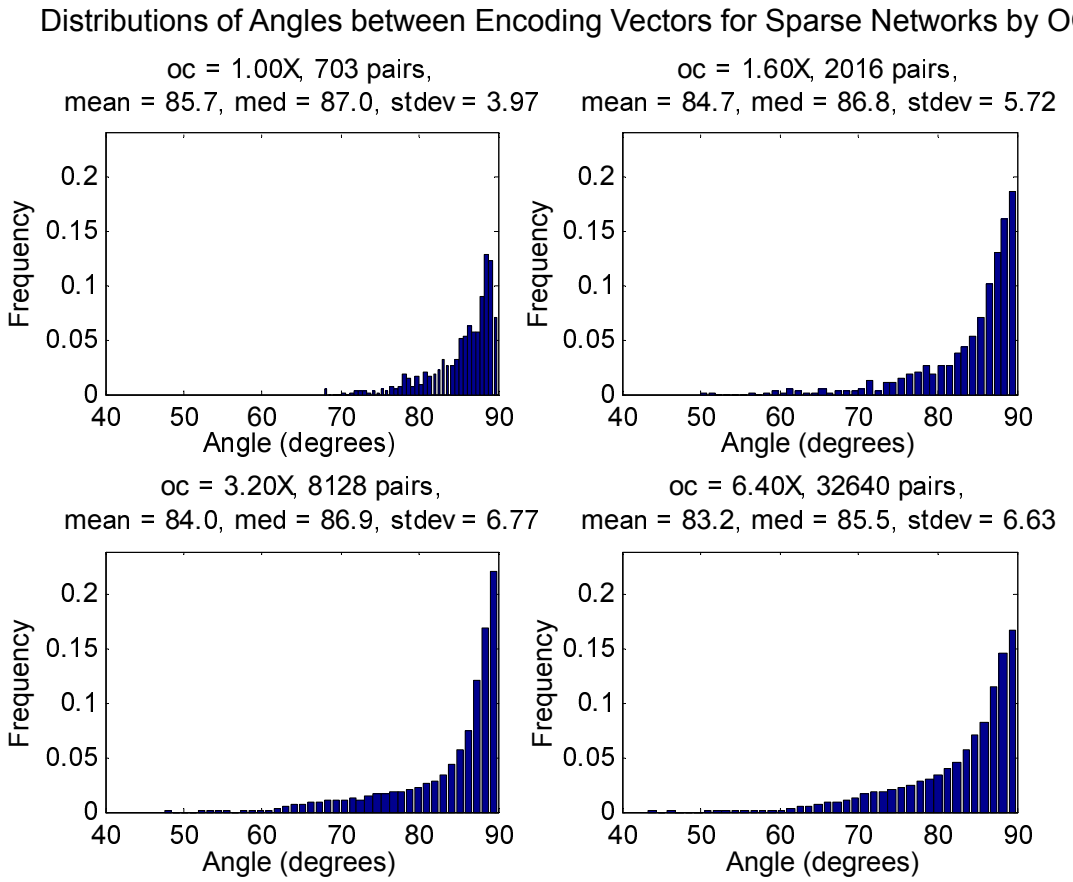

Figure 3.3: The histograms representing the angles between every pair of basis vectors for 8x8 sparse coding networks that are a) 1X overcomplete (703 pairs of basis vectors), b) 1.6X overcomplete (2016 pairs), c) 3.2X overcomplete (8128 pairs) and d) 6.4X overcomplete (32640 pairs). The mean and median angles between pairs for every network decreases with the degree of overcompleteness.

3.3 shows the mean and median of the distributions both decrease with the degree of overcompleteness. This makes intuitive sense, since packing more vectors into the same state space will necessarily result in more angles smaller than 90°. With an idea of the marginal distribution of angles between vectors, we now need to measure the curvature of the response surfaces in order to test for a relationship.

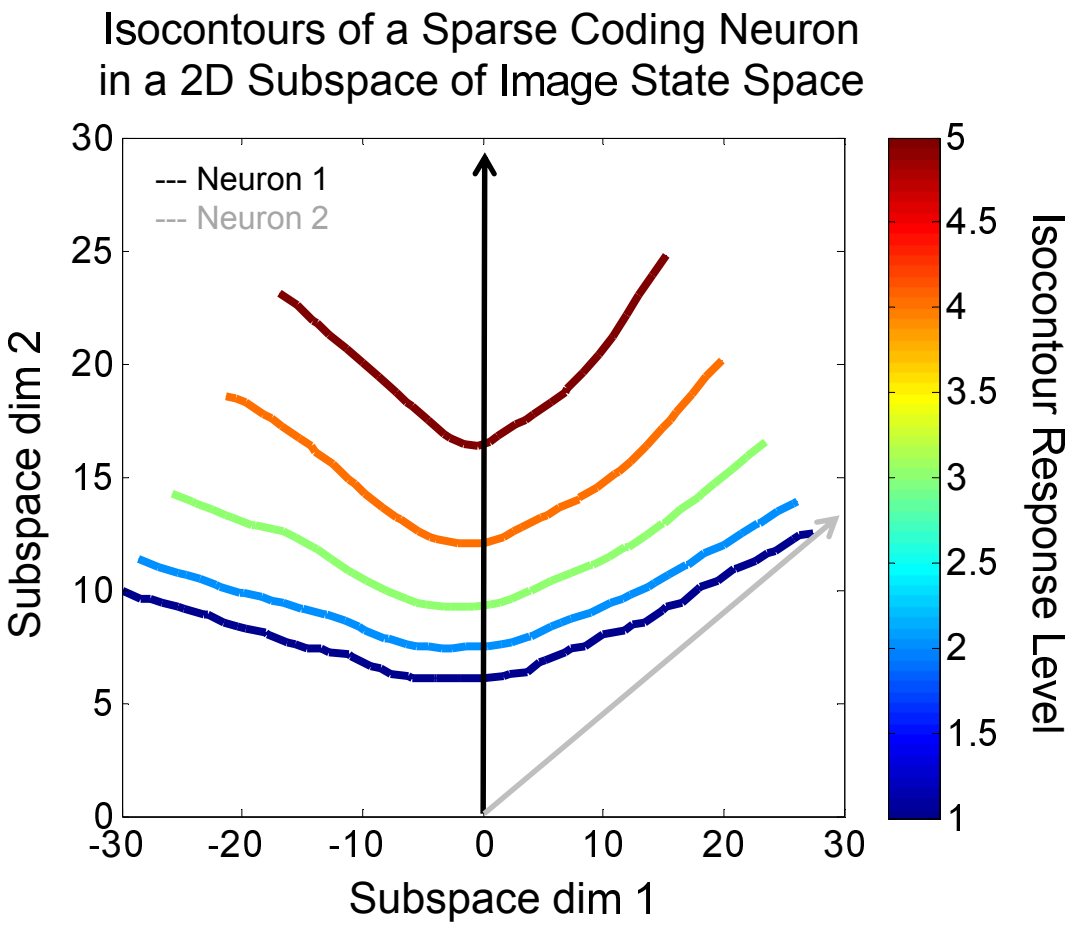

Figure 3.4: The iso-response contours at responses of 1, 2, 3, 4 and 5 for a neuron from a 6.4X overcomplete sparse coding network using a Gaussian cost function in a 2D subspace determined by this neuron's basis vector (blue) and its closest neighbor (red) at 60°. The basis vector points in the direction (0, 1), while the neighboring basis vector points at (0.9, 0.45). Note the curvature of the iso-response contours away from the neighboring basis vector.

To get a first approximation of the curvature of a neuron's response surface, we probed 2D subspaces of the neuron's response in high dimensions. In order to pick out a 2D subspace from the 64D state space, two vectors are needed, and we used pairs of basis vectors. (We wanted an orthogonal pair of vectors to find the

subspace, but since the basis vectors were not always at 90°, we used the Gram-Schmidt procedure to pick out an orthogonal 2D basis containing both of the basis vectors). The response of a neuron could then be plotted in a third dimension as a function of position in the 2D subspace. An example of this response is shown in Fig. 3.4. It is quite similar to Fig. 3.1, which was used to visualize a neural response surface for only a 2D state space. We represent the surface with its iso-response contours, and in this particular example the isocontours exhibit the type of selective curvature hypothesized in Fig. 1.2d. This surface is for a neuron from the 6.4X overcomplete network with a neighboring basis vector at an angle of 60°, so it is one of the strongest examples of curvature among these four networks.

3.2.1 Parabolic Fits to Subspace Curvature

In order to quantify the curvature of the response surface, we examined the iso-response contours of the surfaces in the 2D subspace. For a linear neuron, the isocontours are straight lines, while for a surface with curvature, the isocontours are similar to parabolas (at least for a network using the Gaussian cost function). We fit the each of the iso-response contours with a simple parabolic equation $y = a * x^2 + b * x + c$. The a parameter serves as a measure of the curvature of the response surface, as a linear neuron will be fit perfectly by a = 0 and nonzero values for b and c, while a curved surface will have a nonzero value for a. An example of curve fits to the surface from Fig. 3.4 is shown in Fig. 3.5, which demonstrates that the parabolic equation allows for an accurate fit (high R^2). The isocontours were calculated for the surfaces of each neuron in the 2D subspace determined by its basis function paired with every other basis function for each network. The isocontours were fit by the parabolic equation, and the distributions

of the a parameter for each of the networks are shown below in Fig. 3.6.

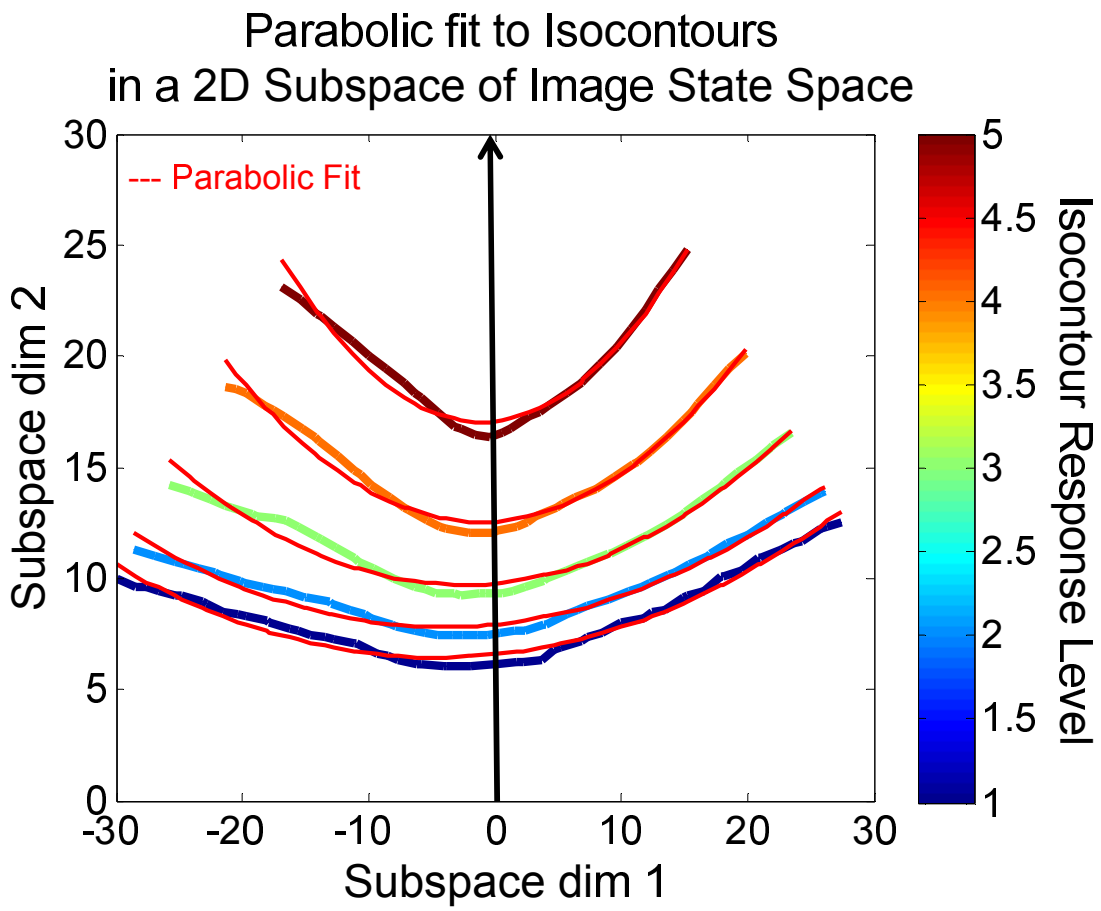

Figure 3.5: The iso-response contours from the above neuron fit with a parabola ($y = a * x^2 + b * x + c$). Note that each fit results in different a values for each iso-response level, as the parabolas for the higher responses are more curved.

The a parameter distributions below in Fig. 3.6 are quite similar to the distributions of angles between basis vectors shown in Fig. 3.3. There is clearly almost no curvature in the 1.0X and 1.6X overcomplete networks in comparison with the 3.2X and 6.4X overcomplete networks. The mean and median as well as the standard deviation of the a distribution grow with overcompleteness.

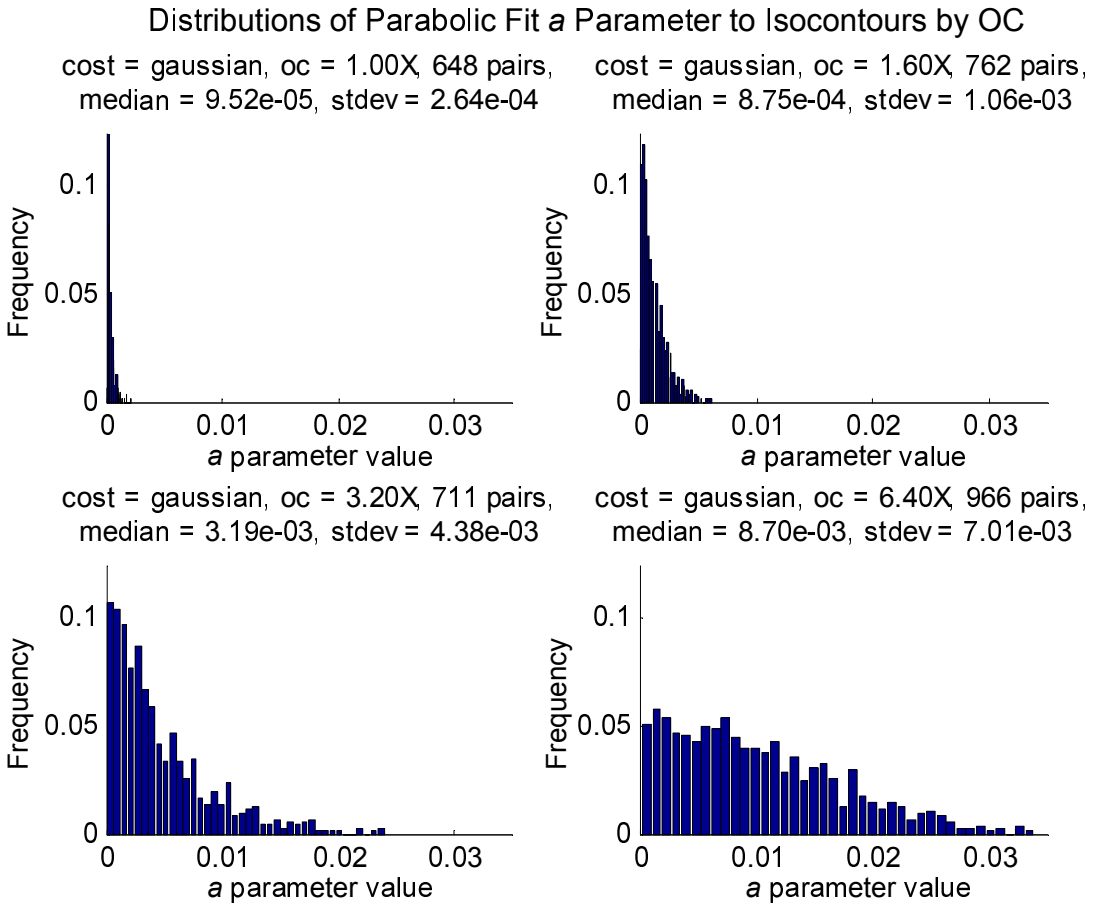

Figure 3.6: The marginal distributions of the parabolic fit parameter a to the isocontours for a subset of the 2D subspaces between basis vectors. The isocontours are becoming more curved (higher a) as the network becomes more overcomplete.

The value of a for iso-level = 5 of each surface is plotted as a function of the angle between basis vectors for each of the four networks in Fig. 3.7. The curvature does not change as a function of the angle for the critically sampled network in Fig. 3.7a, indicating that even neurons at less than 90° will not show curvature if the number of encoding vectors does not exceed the dimensionality of the (whitened) state space. However, this plot generally confirms the hypothesis that curvature changes as a function of the angle between basis vectors: the slope of a linear

fit to the parameter a as a function of the angle between basis vectors increases with the degree of overcompleteness. For a pair of basis vectors at a given angle in two networks with different degrees of overcompleteness, the curvature is likely to be higher in the more overcomplete network. Conversely, it is interesting that the curvature is not completely determined by the angle between basis vectors that form the 2D subspace and the overcompleteness. It is possible that for a given angle, a response surface in the 3.2X OC network will have less curvature than a response in the 1.6X OC network. These observations are probably the result of basis vectors outside of the 2D subspace having an effect on the measured curvature.

3.2.2 Averages of Subspace Isocontours

Another way to quantify the curvature is by calculating the average isocontours profiles of response surfaces for pairs of basis vectors within a certain range of angles. For example, the surfaces generated by pairs of neurons at angles between 65° and 75° can be averaged together, as in Fig. 3.8a, and compared to surfaces from pairs at 75-85°, 85-95°, etc.

Fig. 3.9 shows the average contours for neurons grouped by angular separation. This figure is a summary of the information like what is shown in Fig. 3.8, but for all ranges of angles. Blue curves represent the small angles, while red curves represent the large angles. For the critically sampled network, the average isocontours are not curved at all but rather rotate with each other. For the overcomplete networks, dark blue curves represent surfaces from 65°- 75°, and we see an equal amount of curvature for the dark red curves that represent 105°-115°, with the only difference being the orientation of the secondary surface's curves relative to

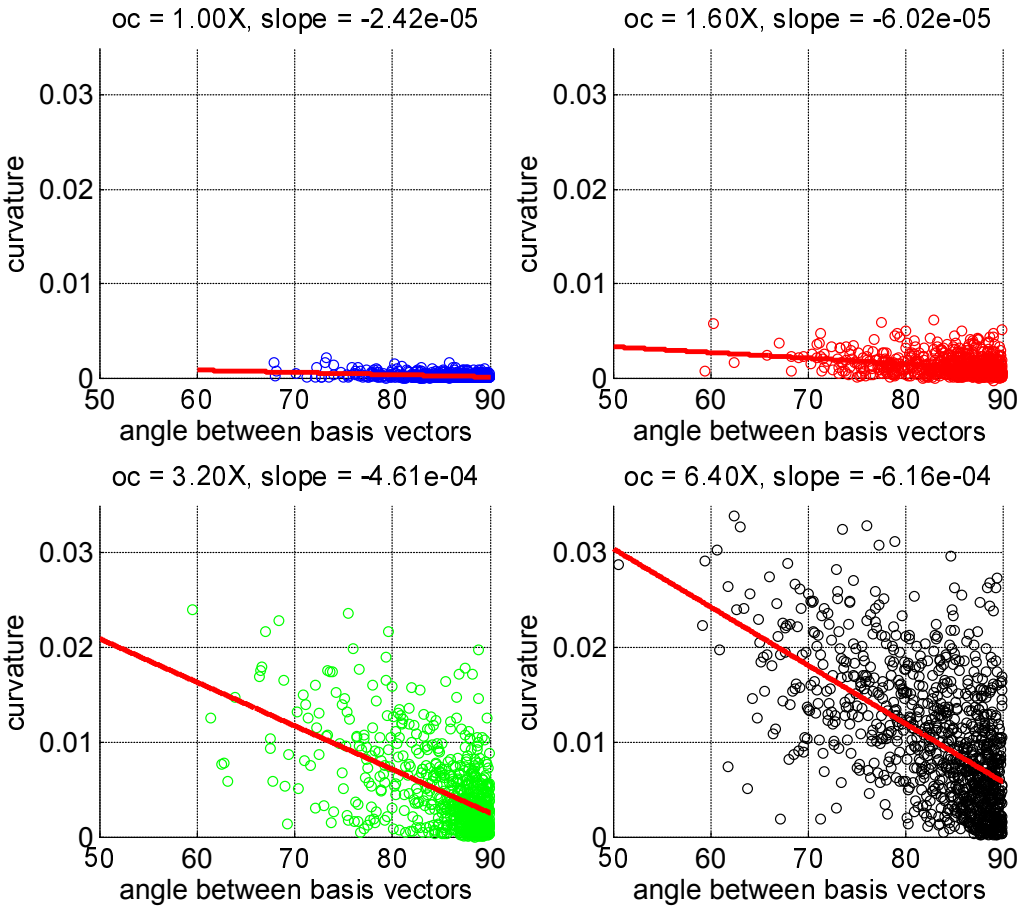

Figure 3.7: The parabolic fit parameter a as a function of angle between basis vectors for networks of 1.0X, 1.6X, 3.2X and 6.4X overcompleteness. There is very little curvature in the networks that are not highly overcomplete. In the more overcomplete networks, the curvature is a clear function of the angle between basis vectors.

the primary vector. As overcompleteness increases, the isocontours move further from the origin and increase in curvature.

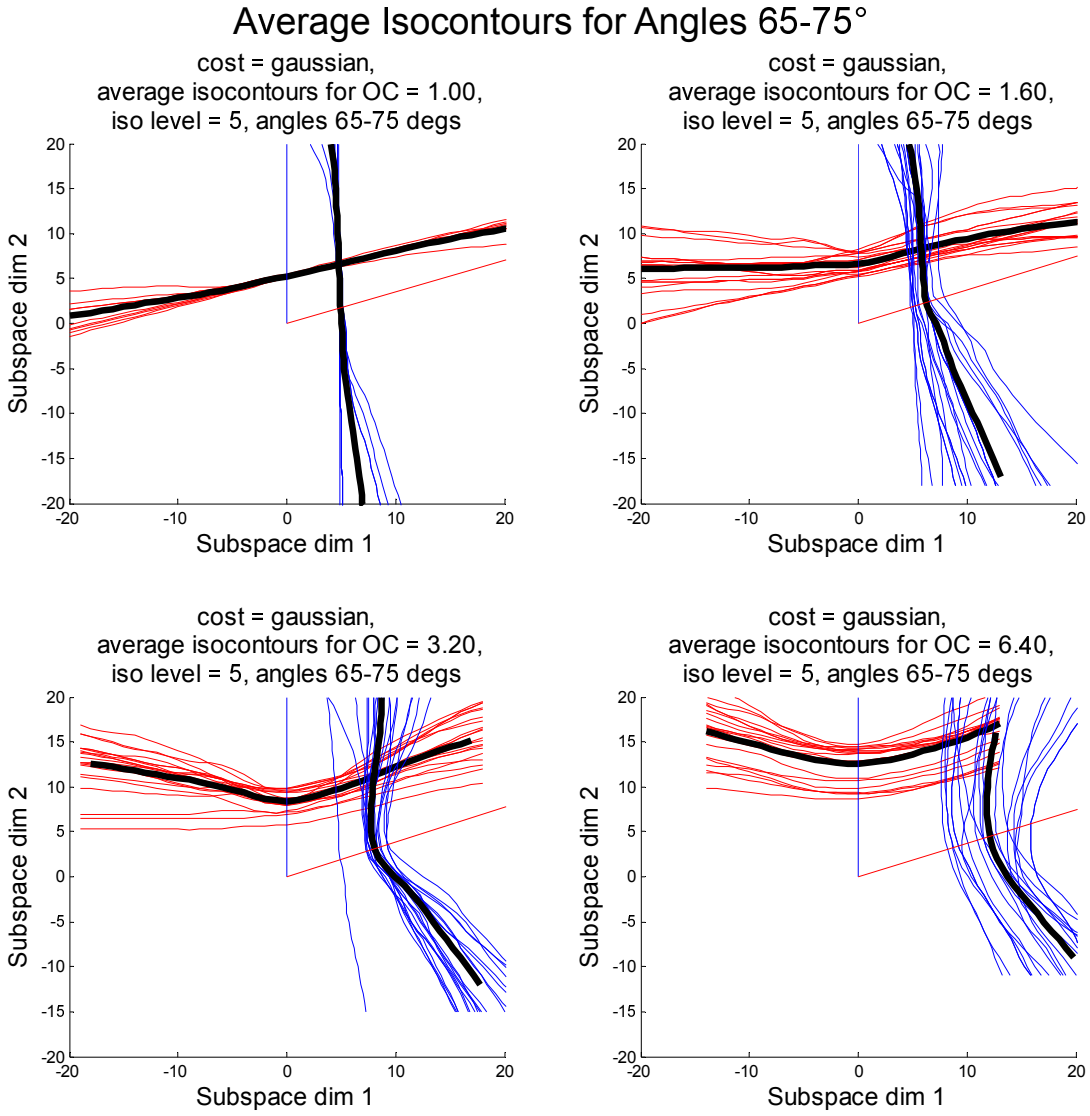

Figure 3.8: Gaussian cost function, four levels of overcompleteness (top left: 1X critically sampled, top right: 1.6X overcomplete, bottom left: 3.2X oc, bottom right: 6.4X oc), 20 random isocontours (blue and red) and the average isocontours (black) for neurons with basis vectors at 65° - 75°. The more overcomplete networks show more curvature for the same angle between basis vectors.

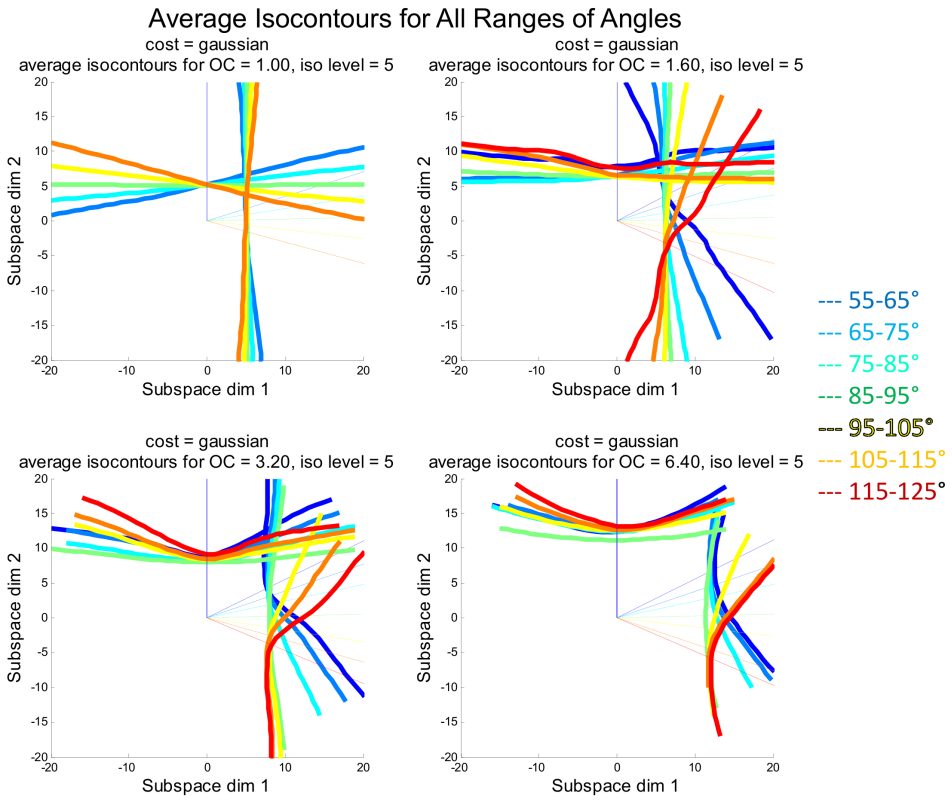

Figure 3.9: The average isocontours for neurons from networks with different degrees of overcompleteness, grouped by angular separation of basis vector pairs. For neurons with basis vectors at a number of ranges of degrees: 55-65°, 65-75°, 75-85°, 85-95°, 95-105°, 105-115°, 115-125°. Rays from origin indicate relative orientation of basis vectors. For the critically sampled network, there is no curvature, and curvature increases with overcompleteness. Neurons at 80° and 100° have very similar average curvatures, the isocontours are just oriented differently to the primary vector.

3.2.3 Fan Fits to Subspace Isocontours

In line with the earlier work of Field and Wu (2004) and Olshausen and Field (2005), it is informative to view the problem of encoding a dataset from the perspective of tessellating the space with encoding vectors and the overlap of their

iso-response contours. Field & Wu proposed a theoretical scheme whereby each data point in a 2-dimensional state space would be optimally encoded only by the two closest vectors, and the weights for any other encoding vectors would be 0 for that data point. This would be the case even for an encoding basis with more vectors than state space dimensions. They termed this a "critically sampled, overcomplete" representation, where even though the number of encoding vectors is overcomplete, the number of active encoding vectors for any data point will only be the same number as the state space dimensionality. Fig 3.10 shows a simple example of a critically sampled, overcomplete representation, where the iso-response contours are described by the fan equation. The red vector is only nonzero for points within 15° on either side. At any of those points, only the blue or only the green vector will also be nonzero. Thus any point where the red vector activation is nonzero leads to a unique encoding with only 2 vectors active.

They proposed that the activation of the encoding vectors in this scheme could be determined by distorting the response of a linear neuron across the angle over which it should be activated. In 2D, a neuron with neighboring neurons at 90° would have iso-response lines that were simply straight. However, if the neighboring neurons were at 45°, the iso-response lines of the neuron would be warped so that its response was zero for stimuli at 45°. The smaller the angle between neurons, the greater the warping of the isocontours. The warping was accomplished using the equation that describes how points on a fan move as the fan is folded. It is a simple modification of the equation for a conic section a = r*cos(θ), where the radius is used to describe an isocontour: $r_{iso} = \frac{isolevel}{\cos(\theta)}$.

It is also possible to fit the isocontours of the neural response surfaces from the sparse coding network with the equation of the folding fan, where the radius of the

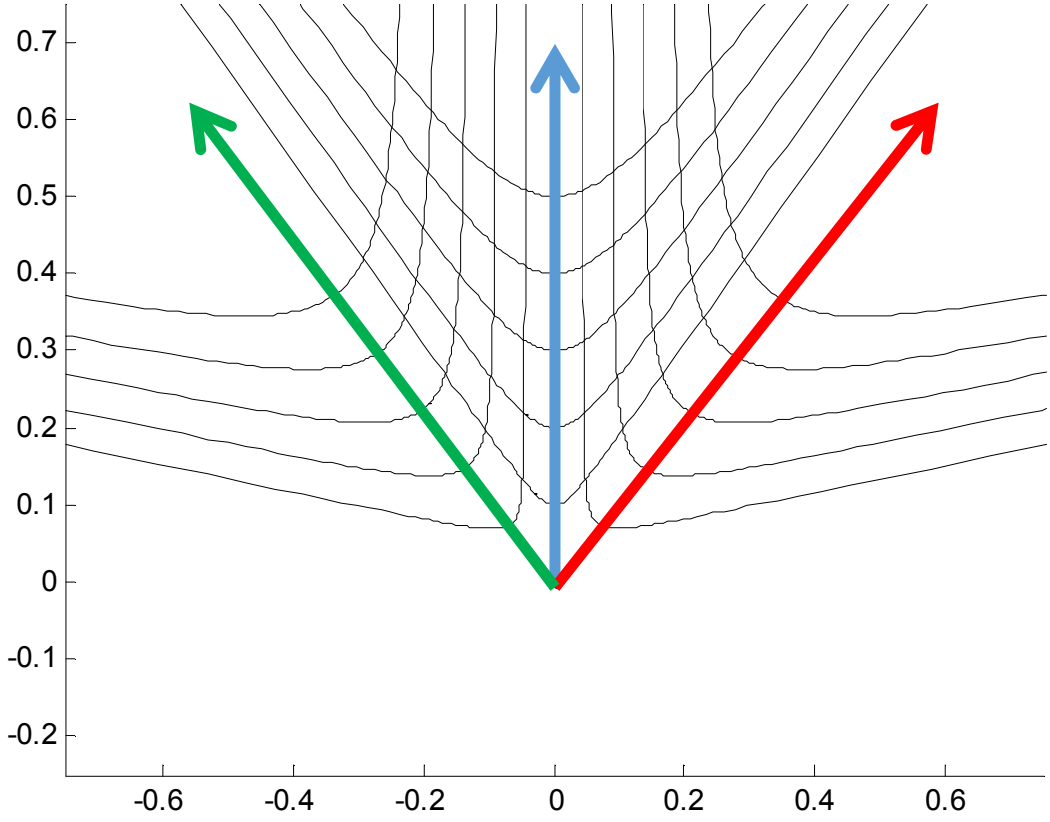

Figure 3.10: The isocontours described by the equation of a folding fan for three vectors. This is a simple example of a critically sampled, overcomplete code. For the red vector, each point in the state space is uniquely encoded by the activation of the red vector and either the activation of the green vector or blue vector.

isocontours is described using the polar equation:

$$r = \frac{isolevel}{cos((\theta/\theta_{fan}) * \pi/2)} \tag{3.1}$$

This equation effectively rescales the cosine operator over an angle less than 90°: when θ_{fan} is set to $\frac{\pi}{2} = 90°$, the isocontours are simply straight lines and

the neural response surface is linear. However, when θ_{fan} is set to a value other than 90°, the surface is rescaled so that the cosine function takes the full range of values 0-1 over the angle θ_{fan}. This has the effect of folding the isocontours into each other the way they would if they were drawn on the surface of a folding fan. The response surfaces of the sparse coding neurons therefore can be fit with the fan equation, just as they were fit with the parabolic equation in the previous section. The same response surface as in Fig. 3.4 is shown below in Fig. 3.11 with the lines of best fit for the fan equation. Interestingly, for the neurons with the Gaussian cost function, the θ_{fan} found by the fitting procedure changed with isocontour level. That is, the θ_{fan} fit for the iso-level with a response of 1 was usually somewhat larger than the θ_{fan} fit for the iso-level of 5. Therefore, each iso-level at responses of 1, 3 and 5 was fit independently for each surface. The R^2 for these fits were high as well, with 90% having $R^2 > 0.95$.

The isocontours of the response surfaces from the sparse coding neurons were also fit with the fan equation, as shown in Fig. 3.12, modified to allow a small degree or rotation. The fit parameter is θ_{fan}, and is shown on the y-axis, as a function of the measured angle between basis functions, θ. For the critically sampled network, the fan angle determined by the fit is highly correlated with the actual angle. The fit values that are greater than 90° correspond to surfaces with values of θ that have been reflected from the $\theta = 90°$ line for the purposes of plotting. For the critically sampled network, the fan angle does not change as a function of the angle between basis vectors because the isocontours show no curvature. For the overcomplete networks, the general trend is that the fan angle tracks the actual angle, but is usually less by some factor. For the 6.4X overcomplete network at iso-level = 5, the fan angle is most closely matched to the actual angle.

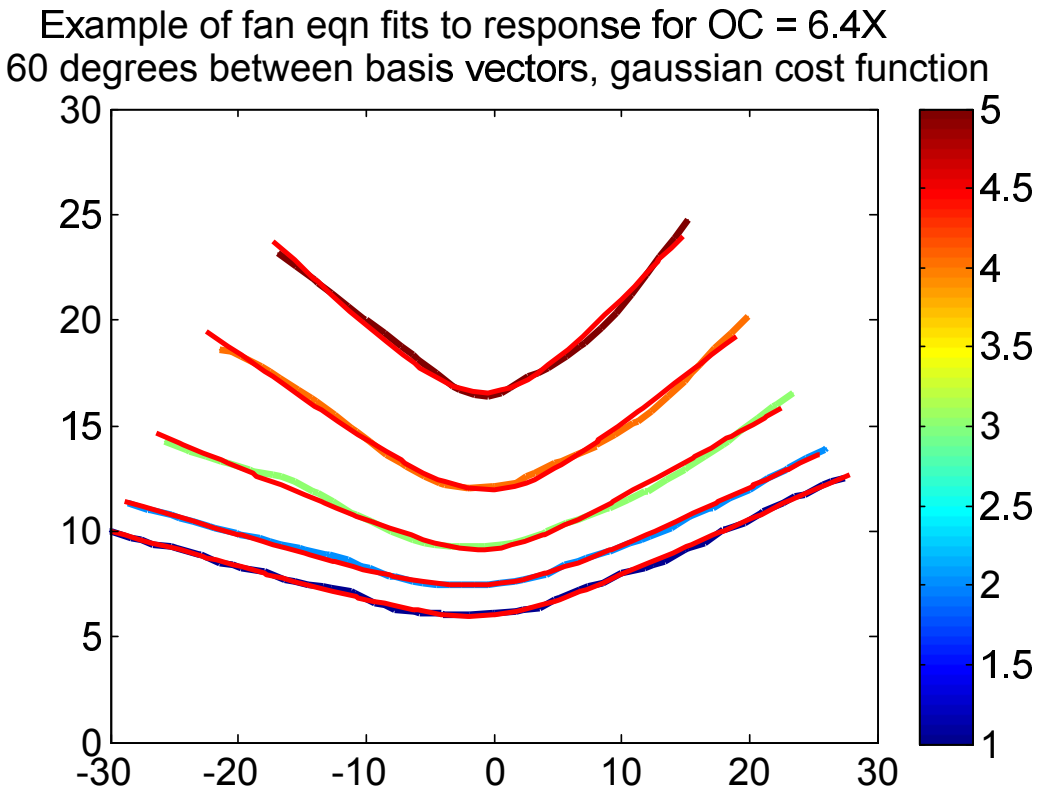

Figure 3.11: The same neural response surface as above in Figs. 3.4 and 3.5 with isocontours fit to the equation of the folding fan. This equation also works well for this particular fit, and 90% of the fits had $R^2 > 0.95$.

3.2.4 The Effect of the Cost Function

When the original sparse coding network was formulated, there were a number of probability distributions that could be imposed upon the activations of the basis functions which would result in a distribution with a high kurtosis (fourth moment) that is sparse. Fig. 3.13 shows intuitively why certain probability distributions will result in sparse solutions. The inner loop of the sparse coding algorithm carries out gradient descent to determine the activations for a given image patch with an

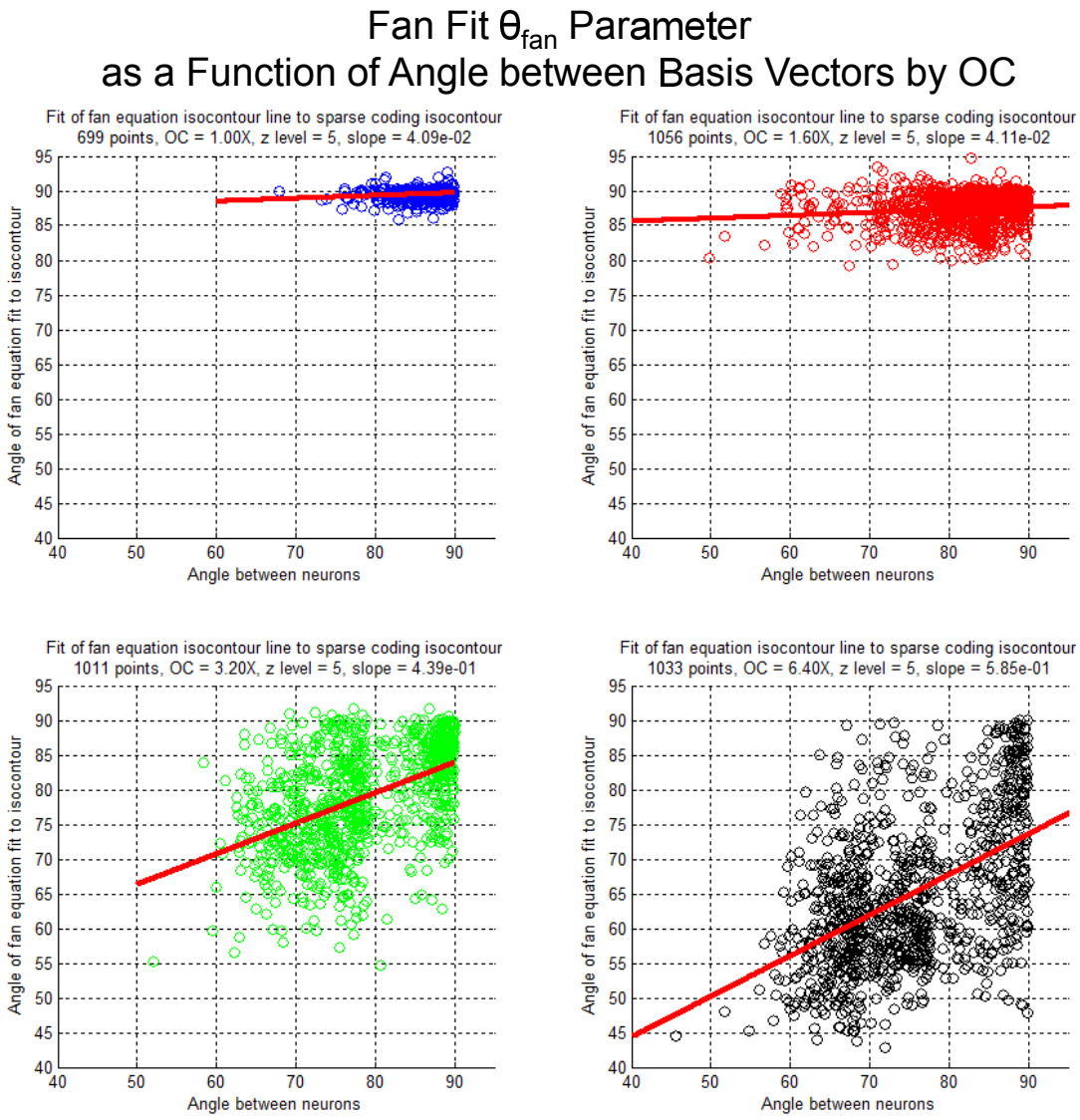

Figure 3.12: The parameter θ_{fan}, as determined by a fit to the iso-response contours from sparse coding neurons as a function of θ, the actual angle between basis functions in the 2D subspace that determines the response surface. Note that the fan angle from the fit is strongly correlated with the actual angle in each case, but that the fan angle more closely matches the basis vector angle in the more overcomplete networks.

equation determined by the assumed sparse distribution. As mentioned before, the most common cost functions are $-\exp(-x^2)$, $\log(1+x^2)$ and $-|x|$, corresponding to the Gaussian, Cauchy and Laplace distributions. All of the above analysis has been performed using the Gaussian cost function, but the same analyses can be done for the sparse network with the other cost functions.

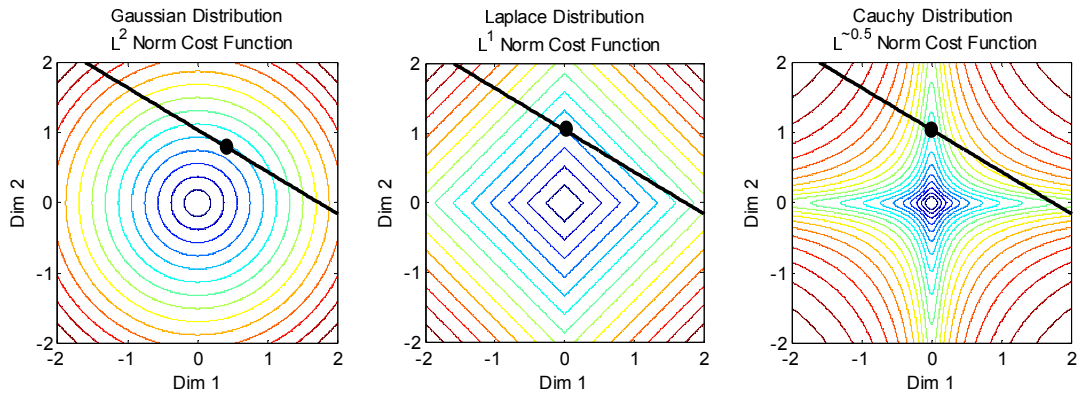

Figure 3.13: After Olshausen & Field (1996). Solutions to an optimization problem are found when the at the intersection of the lowest possible isocontour of the cost function with the plane representing possible solutions. A solution is sparse if it lies on the y-axis in this toy example, because that implies the x-coordinate value is zero. a) A Gaussian L^2 prior is not sparse, because solutions will have nonzero entries for both encoding vectors. Here the solution is (0.5, 0.7). Note, the Gaussian prior is distinct from the Gaussian cost function, which is sparse. b) The L^1 norm or Laplace prior results in sparse solutions; here, the solution is (0, 1). c) The Cauchy prior, which results in a norm between L^1 and L^0, also results in a sparse solution.

Prior work (Körding et al., 2003) has probed the effect of the sparse prior from a different approach, focusing mainly on aggregate statistical properties of the response of the network over large batches of images, as well as the resulting basis functions from the learning process. They found subtle differences in terms of

the mode of the activations as well as qualitative differences in the learned basis functions that were observed by visual inspection. Here we offer a more dramatic comparison of the effects of the sparse prior. Fig. 3.14 shows the isocontours for the same neuron over the same 2D piece of state space, with the inner loop evaluated with the Gaussian cost function on the left and the absolute value cost function on the right. The prior/cost function completely changes the nonlinear response of the network. One of the aspects of the nonlinear response we were interested in was how closely the sparse coding responses match different ideal models: pure contrast gain control, divisive normalization or warping via the fan equation. Fig. 3.14 shows that the sparse neurons can change the curvature of their response surface with contrast, while the absolute value cost function yields isocontours that remain straight and parallel to one another at increasing contrast, which is closer to warping via the fan equation. The sparse coding network not only curves response surfaces in high dimensions, but does it differently according to the sparse prior/cost function.

The gradient descent equation for the inner loop results from the generative model of a sparse code, and by maximizing the posterior probability with respect to the activations x:

$$\frac{d}{dx}[\log p(I)] = \Phi^T[I - \Phi * x] + \lambda * \sum \frac{d}{dx}(cost(x)) \tag{3.2}$$

Just as it is not possible to derive a closed-form analytic expression for the activations x when the network is overcomplete, there is no closed-form expression for the iso-response surfaces of the neurons, although there is obviously an effect of the cost function on the isocontours.

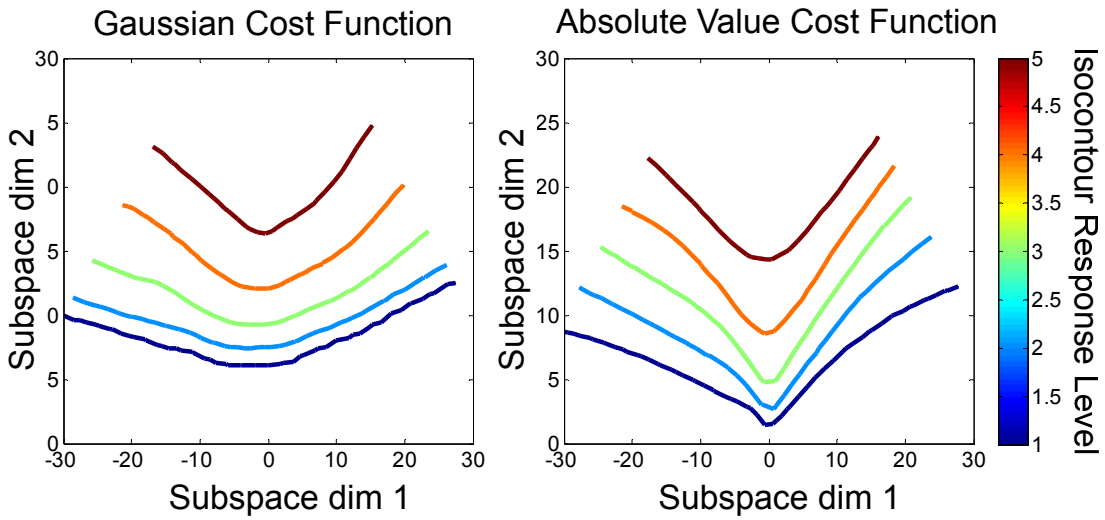

Figure 3.14: a) The iso-response surfaces from Fig. 3.4 a 6.4X overcomplete network with a 60° angle between basis vectors, generated using the Gaussian cost function. b) The iso-response surfaces for the same neuron from the same 2D subspace generated using the Laplace prior/absolute value cost function. The isocontours closest to the neighboring vector are straight and evenly-spaced as the radius (RMS contrast) increases. This demonstrates that the choice of cost function (equivalent to the assumption of a particular sparse prior) has a clear effect on the nonlinear responses of neurons in the sparse coding network.

Another interesting aspect of these plots is that the isocontours are asymmetrical, due to the placement of the two basis vectors. However, we hypothesized that the curvature would be primarily due to the nearest neighbor basis vector, which implies that the isocontours on the left side of the neuron at (0,1) in Fig. 3.14 should not have curvature, and yet they do. We believe that this is due to the cumulative effect of the other vectors in the state space, as the curvature on the far side of the response surfaces increases with overcompleteness.

The average iso-response contours for the networks with four degrees of over-

completeness using the Laplace prior/absolute value cost function are shown below, as a means of comparison between surfaces due to the Gaussian cost function shown in Fig. 3.9. The first difference between is that these are more symmetric, and that the angle between basis vectors does not seem have as much of an effect, because the average curves from each angular range are similar. Unlike Fig. 3.9 the isocontours for degree of overcompleteness and grouping by angle primarily stay centered in the same position, indicative of a type of curvature due to warping by the fan equation as opposed to pure gain control.

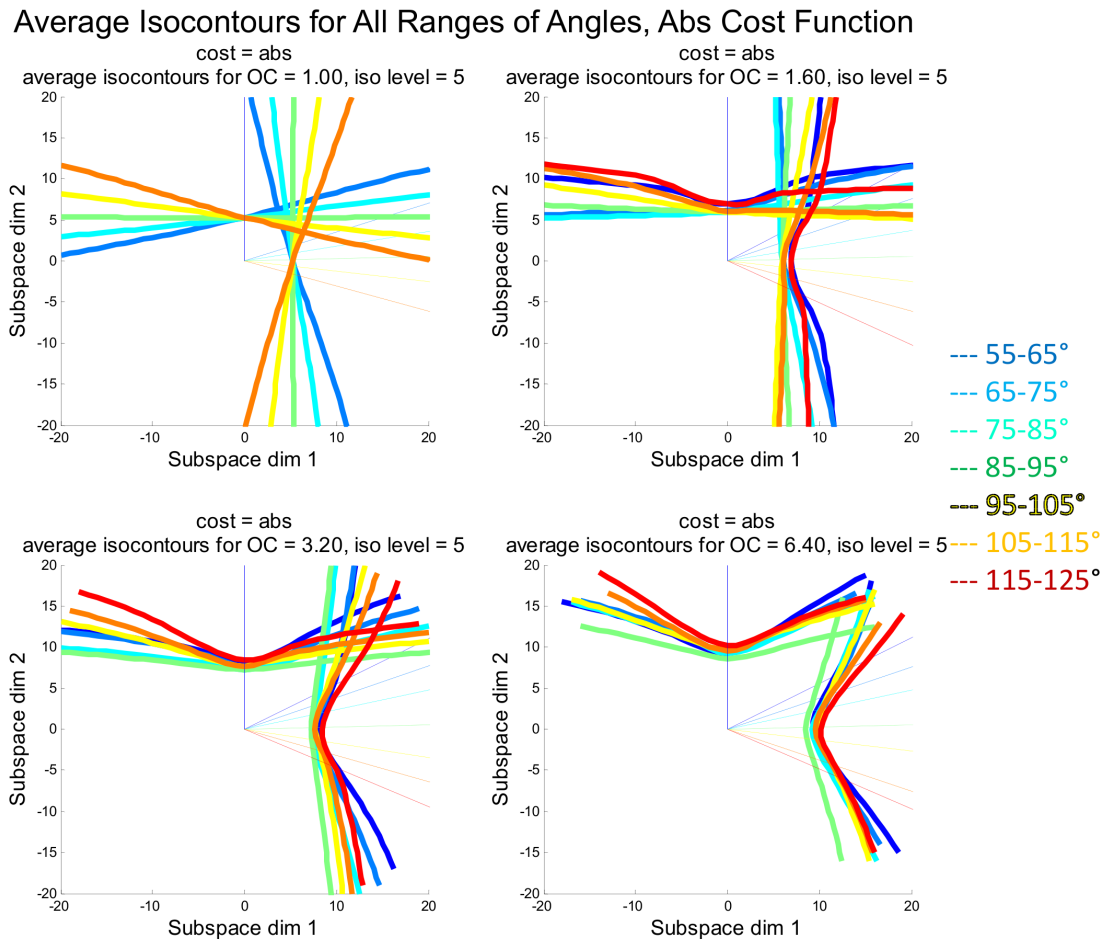

Figure 3.15: Average isocontours for the sparse coding network with the Laplace prior/absolute value cost function with basis vectors at a number of ranges of angles: at 55-65°, 65-75°, 75-85°, 85-95°, 95-105°, 105-115°, 115-125°. In contrast with isocontours due to the Gaussian cost function, contours are more symmetric and less affected by the angle. Note the general resemblance to the isocontours described by the fan equation.

CHAPTER 4

THE CURVATURE OF THE SPARSE CODING NETWORK: IMAGE STATE SPACE

We now invoke a rigorous formulation of the measurement of curvature in high dimensions. We apply these curvature measures to manifolds representing neural responses in order to quantify their curvature locally in the full state space. The sparse coding network, as a model that learns an efficient representation of image data and reproduces some of the nonlinear responses of V1 neurons, will be used to generate the response manifolds. The principal curvatures of a neural response surface are a direct quantitative measurement of the high-dimensional features to which a neuron is selective and invariant. In Section 3.1, we interpreted illustrative toy examples of the sparse coding network in 2D that allow the responses to be visualized over the whole state space, while in Section 3.2 we performed an exhaustive analysis of the 2D subspaces of neural responses in image space. Now we move on to measurements of the curvature of response surfaces in 64D image space.

The possibility that the sparse coding network could capture nonlinear effects like endstopping was mentioned in Olshausen and Field (2005), and the connection between endstopping and curved response manifolds was discussed by Field and Wu (2004). We have provided evidence of the curvature of sparse coding response surfaces in low-dimensional subspaces, and we believe this allows for a new interpretation of how the sparse coding network finds a representation. The network provides an accepted method for placing encoding vectors to learn an efficient code that resemble V1 receptive fields. Here, we demonstrate that the recurrent inner loop step that finds an overcomplete sparse representation warps

the encoding space according to the density of natural images in different regions of state space. A lattice representing state space becomes distorted and twisted when viewed in the representation space. The representation is made efficient by the encoding vectors that tessellate the state space as well as how the state space is warped when viewed in the representation space. This distortion of the space is a way to view what the sparse coding network is doing from a more intuitive perspective, and it can be measured with the curvature of the response surfaces.

The sparse coding network is already a widely accepted model for network-level processing in V1, and serves as the basis for some of the deep learning networks that have become popular in the last decade (Le and Ng, 2013). Therefore, this insight into its mechanics allows us to develop new tools to understand the features to which it is selective and invariant by measuring the curvature of response surfaces. We can also apply these tools to multi-layer networks and hopefully produce more intuitive explanations to exactly which features the show selective and invariant responses.

In this chapter, we first discuss the output of applying the curvature measurement to the response surface of a neuron from the sparse coding network. We provide an interpretation according to feature selectivity and invariance. We discuss the important distinction between the curvature of the full response surface and an iso-response surface. Then we provide summary measurements of curvature over response surfaces and discuss comparisons between different measures and different networks.

4.1 Curvature in high dimensions: a primer

The differential geometry of surfaces is a field of mathematics that has been developed over the last several hundred years (Pressley, 2010). A full account of the mathematics of curved surfaces is available in Appendix A; the reader is urged to turn there if there is any hesitation due to a lack of background in the measurements presented in this chapter. To summarize very briefly, curvature is a measure of how the surface normal changes locally in a quadratic fashion. For a surface in N-dimensional state space, the principal curvatures at a point are found by taking the eigenvector decomposition of the product of the Hessian (matrix of second derivatives) with the surface normal divided by the product of the gradient with itself (see Appendix A for a derivation). For a 2D surface in 3D space, the eigenvector equation is as follows:

$$\begin{bmatrix} f_x * f_x & f_x * f_y \\ f_y * f_x & f_y * f_y \end{bmatrix}^{-1} * \begin{bmatrix} f_{xx} * N & f_{xy} * N \\ f_{yx} * N & f_{yy} * N \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \kappa * \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{4.1}$$

The principal curvatures consist of N vectors of different magnitudes. For neural response surfaces, the principal curvatures are a quantitative measure of the features to which a neuron's response is selective and invariant.

Below in Fig. 4.1 is an example of the output produced by the algorithm for measuring the curvature at a point on a sphere in 64D with $R = 2$. There are 64 principal curvature magnitudes and directions, with the magnitudes all equal to 1/R in the stem plot at left, and the directions in the 8x8 image patch plot at right. A principal direction in 64D space can be shown as an image, and when the curvature algorithm is applied to neural response surfaces, the images will allow

some interpretation in the context of the basis functions/receptive fields of the network. For the hypersphere, since the curvature is the same in any direction at all points, any direction is a principal direction. Here, the algorithm finds the coordinate basis of the state space as the principal directions. In the 8x8-pixel image of each principal direction, all of the gray pixels correspond to a value of zero, while the sole white pixels corresponds to a value of one. Individual measurements of the curvature of neural response surfaces will be presented with this type of figure.

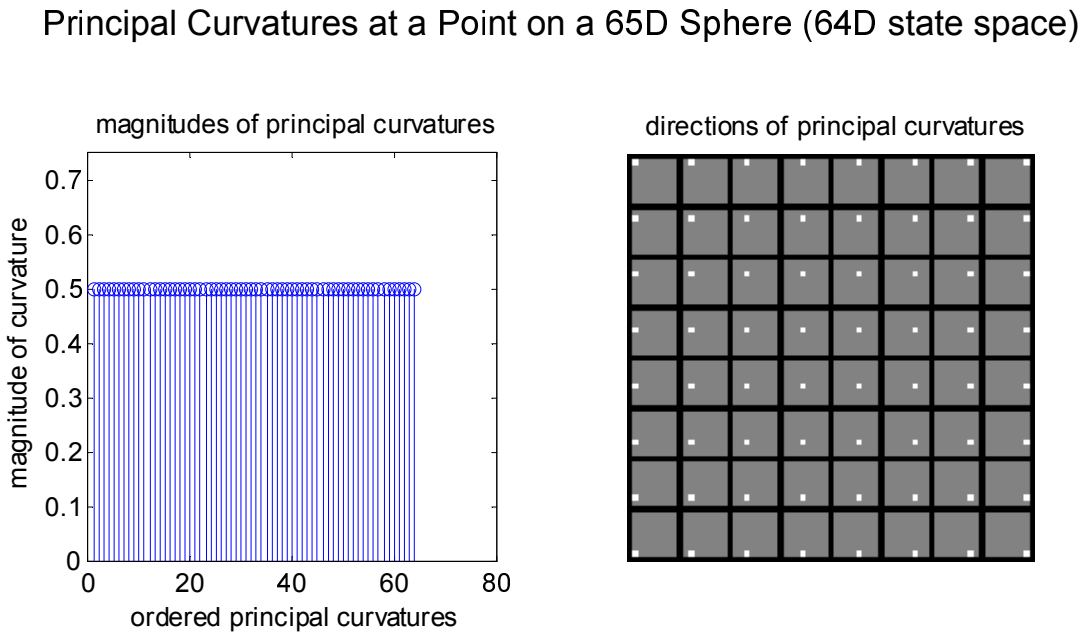

Figure 4.1: The principal curvatures and principal directions of the hypersphere with R = 2 in 64D state space. Note the curvature magnitudes are all equally $1/R = 0.5$, and the principal directions can be represented as 8x8-pixel images in the state space, here corresponding to the coordinate basis.

4.2 Results of curvature measurements in high dimensions

The measurement of curvature of the response surfaces of sparse coding neurons will yield exactly how much and in what direction they are nonlinear in high dimensions. The results described in this chapter can be summarized briefly. Do predictions from the low-dimensional results hold? For the most part, curvature is still a function of the sparsity, the angle between basis vectors and the overcompleteness of the network. Are the response surfaces in relevant subspaces purely hyperbolic, since the network ought to be purely selective? Yes, with the caveat that the neurons are only purely selective at their maximum response value for an image of a particular contrast. We predict that the response surfaces will show the strongest curvature in the direction of other neurons - is that the case? Specifically, the principal directions of the curvature of response surfaces ought to resemble the basis functions (or receptive fields) of other neurons in angular proximity. The principal directions could have turned out to be any vector in the state space, but for high principal curvatures they are almost always basis functions of other neurons in the network. Since V2 neurons tend to be more nonlinear than V1 neurons, do the response surfaces of neurons from the Karklin & Lewicki network show greater curvature than those of the sparse coding network? We have found that the curvature is greater, and that the maximum responses show positive curvature that is evidence of tolerance/invariance. The evidence for these findings is detailed below.

These ideas point to statistical statements that we can try to make about the curvature as a function of properties of the point on the surface. For example, if principal directions are the basis functions of nearby neurons, then perhaps the curvature magnitude is correlated with the angle between the surface's basis

function and other basis functions. As has been described before, we know the curvature of response surfaces is tied to the sparsity parameter λ, and therefore curvature will increase as the network is forced to be sparser. Along these same lines, curvature should also increase if there are simply more encoding vectors for the same state space. We will provide quantitative evaluations of these predictions.

Another question is whether we will be able to observe global patterns in the curvature measurements (primarily in the principal directions). This seems somewhat unlikely, as the curvature should be more dependent on the point from which it is being measured. If a point on the neural response surface of neuron 1 is orthogonal to the basis function of neuron 2, there should not be much curvature in the direction of neuron 2; but if another point is only 20° away from neuron 2, then there will likely be curvature in that direction. However, it may be possible to derive an approximate formula that predicts the curvature of a neuron at a particular point as a function of the basis set, which could allow for a feedforward approximation of the response of the sparse coding network. This approximation would resemble the network that implements slow feature analysis (Wiskott and Sejnowski, 2002), although it would still be more complex as the curvature for SFA neurons is the same at every point.

Consider the output of the measurement for a neuron in the sparse coding network shown in Fig. 4.2. In order to completely characterize a neuron with nonlinear behavior throughout the high-dimensional state space, some form of a lookup table would be necessary (although impossible in the practical sense in high dimensions). This curvature measurement of a response manifold yields only local information, but gives a measure of what the manifold is doing locally in the full high-dimensional space. It will allow testing of the previously mentioned

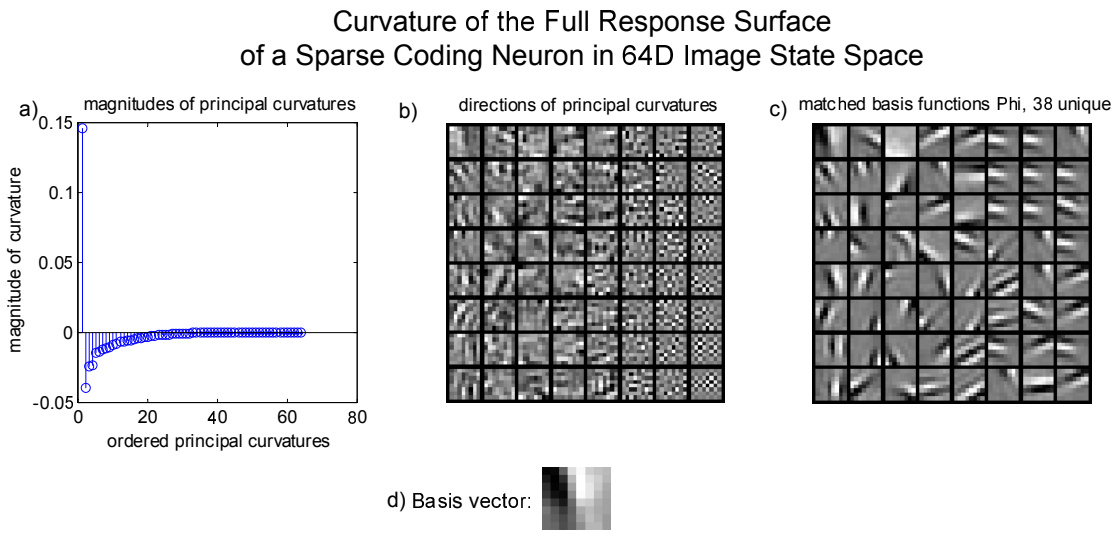

Figure 4.2: The curvature on the response manifold of one neuron at one point. a) The principal curvatures, which represent the magnitude of the curvature. b) The principal directions of the curvature. c) The closest basis function to each principal direction in terms of inner product. Note that the principal curvatures with the strongest magnitudes have directions that match reasonably well with basis functions, although they are not perfectly matched in terms of spatial frequency and phase.

hypotheses, but those can only be supported by measuring curvature at many points on many different response manifolds.

Like the measurement for curvature on a sphere in high dimensions above in Fig. 4.1, we have both the principal curvature magnitudes and directions in Fig. 4.2. This one measurement already provides some support for the hypothesis that the directions of strongest curvature will be towards other basis vectors from the sparse coding network. As we have seen, the principal curvatures result from an eigenvector decomposition of a particular combination of the normal and the first and second derivatives of the surface in high dimensions. The eigenvector

decomposition is also used in principal components analysis (PCA). In PCA, the largest eigenvector is the direction of greatest variance in the data; the second largest eigenvector is the direction of greatest variance in the data orthogonal to the first eigenvector, and so on. For the principal curvature eigenvectors, the largest is the direction of greatest curvature, the second largest is the direction of greatest curvature orthogonal to the first, etc. (Berkes and Wiskott, 2006). The principal directions could have been any vector in 64D for this particular response surface, but they are clearly related to the basis functions of other neurons in the network. In other words, the directions of greatest curvature on the response surface are the basis vectors of other neurons. The inhibition due to the nonlinear inner loop stage that finds the overcomplete sparse representation results in large degrees of curvature. To demonstrate this, Fig. 4.2c shows the basis functions that most closely correspond with each of the principal directions (in terms of greatest inner product). There is a striking correspondence between the principal directions and the basis vectors of the network. The principal directions with the largest curvature correspond to basis functions that are less than 90° away from the basis vector representing the response surface. These relationships will be quantified in a statistical manner below.

One possible confounding problem with this analysis is that the largest curvature is in the direction of the neuron's own basis vector. This is information about the contrast response of the neuron: when the magnitude of the stimulus is increased in the direction of the basis vector, the neuron's response increases at a nonlinear rate. This is unfortunately the least interesting type of nonlinearity into which the curvature analysis could yield insight. As discussed earlier, the curvature of the iso-response surface of the neuron's response manifold will not have this confounding problem, as the curvature will be measured in with the ba-

sis vector as the surface normal. Below in Fig. 4.3 is the curvature analysis for an iso-response surface of the neuron. This is quite similar to the curvature of the full surface, except there is no curvature in the direction of the encoding vector. The encoding vector is the normal vector to an isosurface, so the isosurface curvature measure serves as the more interesting characterization. Fig. 4.4 shows a comparison between the curvature of the full surface in (a) as well as iso-response surface in (b).

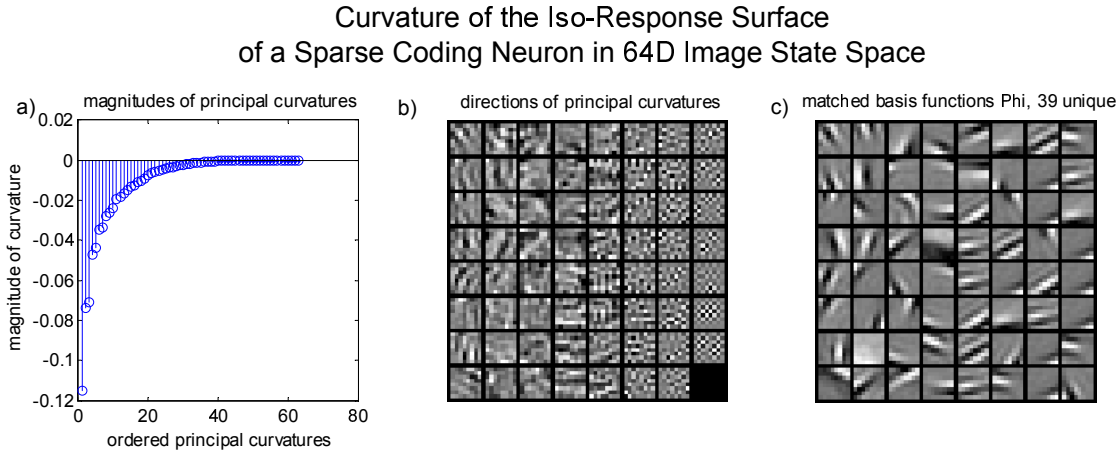

Figure 4.3: a) Curvature of the iso-response surface. b) The principal directions of the iso-response surface curvature do not include the neuron's own basis vector. Therefore, the curvature of the iso-response surface is more informative about how the response manifold is warped by the presence of other neurons. For a neuron from the sparse coding network, at a point in state space identical with the neuron's basis function, all of the principal curvatures of the isosurface are negative, indicating pure selectivity. More examples of this measure are shown in Section B for networks of different degrees of overcompleteness.

As for the magnitudes of the principal curvatures, the interpretation is not as obvious. First, note that the magnitudes of curvature are different for the

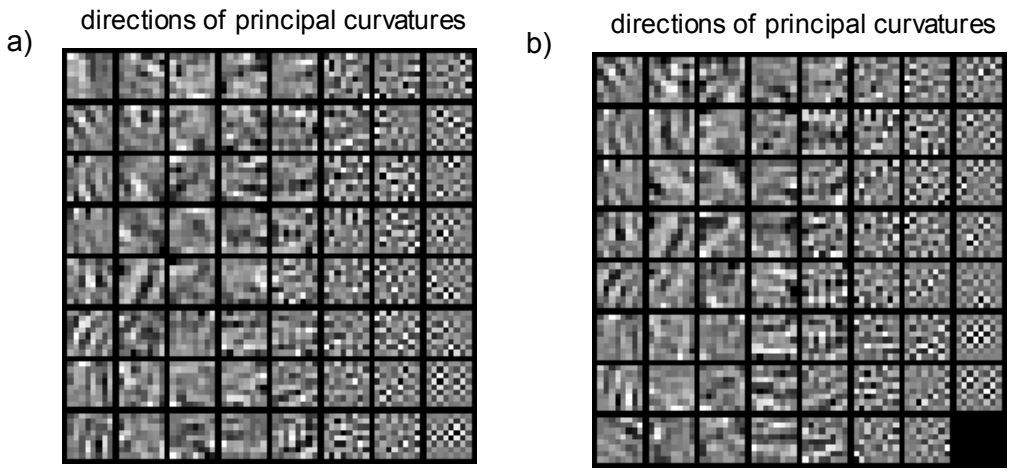

Figure 4.4: Note that in terms of the matched basis functions, the principal directions of both a) the full response surface and b) the iso-response surface are quite similar. The first five principal directions of the iso-response surface are the second through sixth principal directions of the full response surface. As expected, the iso-response surface and the full response surface are curved in similar directions.

iso-response surface and the full surface. This is in part an inherent difference in iso-surface curvature. Consider a sphere in 3D, $x^2+y^2+z^2 = R^2$, with iso-response surfaces that are circles, as in Fig. 4.5. For a constant z value, the radius of the iso-response circle is defined by a choice of the point on the sphere. When x^2+y^2 is small, z is big, and the iso-response circle has a small radius and therefore a large curvature. When x^2+y^2 is big, the radius of the iso-response circle approaches the radius of the sphere. The curvature magnitude of any iso-response surface therefore has a lower bound defined by the radius of the full surface. All of this is to simply say we should not expect iso-response curvature magnitudes to completely match curvature magnitudes of the full surface. As long as we only compare isosurface

curvatures to other isosurface curvatures, this should not matter.

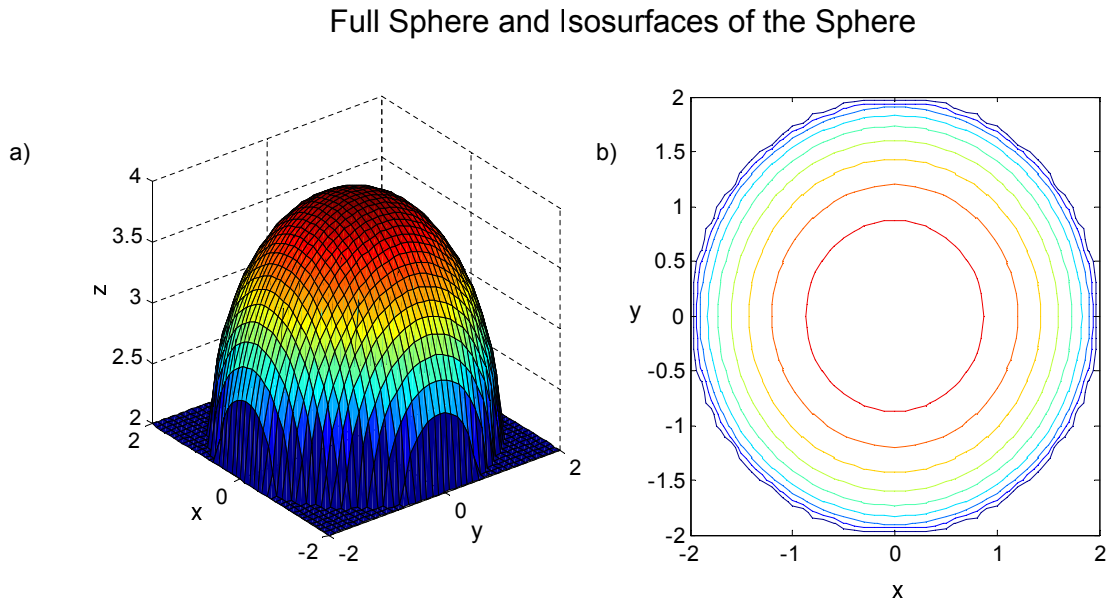

Figure 4.5: a) The sphere has the same curvatures (magnitudes and directions) at any point. b) The isosurfaces of the sphere have different curvature (inverse of the radius of the isosurface, in this case) at different points in the state space. However, for a sphere, the curvature at any two points on the same isosurface is identical. When comparing isosurface curvature, only curvatures from the same iso-level ought to be compared to one another.

Here we will also consider the curvature of neurons from the second layer of the Karklin & Lewicki network. Based on Rust and DiCarlo (2012), neurons farther up the hierarchy of visual cortex will likely show both increased selectivity and increased invariance. The neurons of the Karklin & Lewicki network have been shown to capture aspects of invariance that are not seen in sparse coding neurons. The curvature due to selectivity and invariance should be stronger than the curvature of sparse coding neurons. The results for the measurement of the selectivity and invariance of a second layer neuron is shown below in Fig. 4.6. There are

clearly strong positive and negative curvatures, and the absolute magnitudes are much greater than what is seen in sparse coding responses. Finally, note the invariant responses to vertically-oriented features, while selective responses are due to horizontally-oriented features.

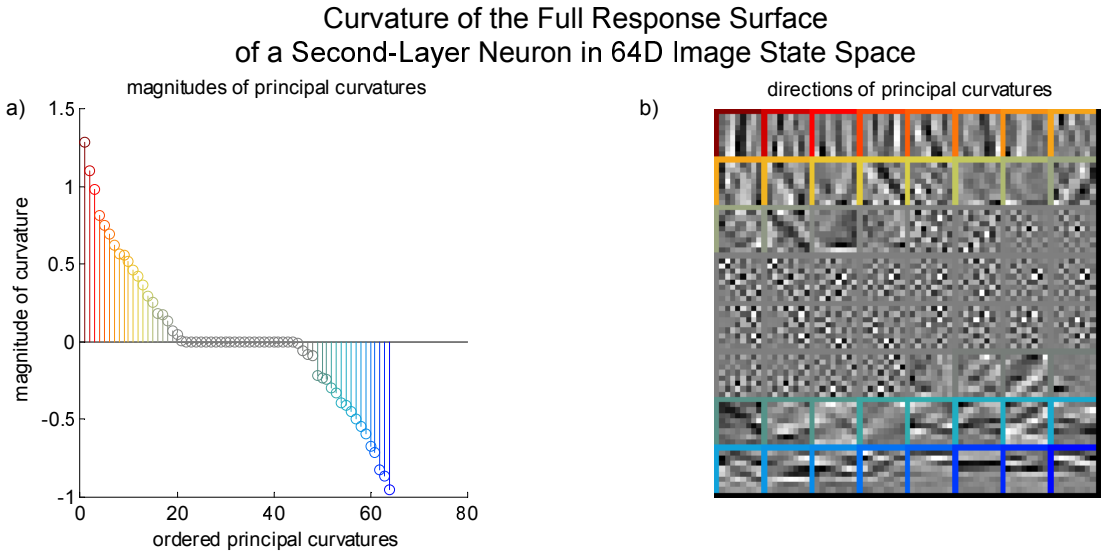

Figure 4.6: The curvature of a second-layer neuron from the Karklin & Lewicki network. Note the curvature is several orders of magnitude larger than that of the sparse coding neuron above in Fig. 4.2 and that there are strong positive and negative components. The neuron is invariant toward vertically-oriented features and selective for horizontally-oriented features.

The measurement of the curvature of a neuron's response in this full high-dimensional space provides support for a theory created in a toy low-dimensional space, and results in the idea that selectivity is identical to negative curvature of the response manifold. By making exact numerical measurements of the magnitude of the selectivity and the directions in which the neuron is selective, we have a wholly new way of conceptualizing a neuron's selectivity. Along with that comes an interpretation of the curvature, such that we can use the method to describe

exactly how selective a neuron is to a set of specific image features. These measurements may not be feasible for physiological experiments because they are reliant on exact equations describing a response, but it may be possible to use them to make predictions from a model that captures aspects of the physiology. For example, if we know a neuron's receptive field, and the receptive fields of others nearby in state space (Tsai and Cox, 2015), then we ought to be able to simulate responses using the sparse coding network to make predictions about where the isocontours will fall in the state space. Further, it should have clear importance in deep belief networks, which are governed by systems of equations and exact numerical derivatives may be calculated. Simple ways of testing feature selectivity and invariance based on a network finding the representations for a large batch of images (Goodfellow et al., 2009) may be compared with this theoretical measurement of selectivity and invariance.

4.3 Summary Measures of Neural Response Curvature

We collected summary statistics on the curvature of neural response surfaces from the sparse coding network. The response surfaces are functions of some high-order polynomial, so the curvature, which is a second-order measure, changes at every point. Since the surfaces are high-dimensional objects, it is not practical to measure the curvature over the whole state space. We chose to measure the curvature at a subset of the images of a constant contrast (on the surface of a sphere in image state space). For one category, we chose sample points within 90° of each basis vector, because we have seen in Chapter 3 that there is strong curvature at those points of the surfaces. In addition to the points in state space near the basis vectors, we have also measured curvature at natural image points from the training set. This

allows us to measure the response surface curvature for points of the state space that the network actually visits when it is encoding real natural image data. As a point of comparison, we have also examined the curvature at random points in the state space. A further complexity is that we are interested in the curvature of the iso-response surfaces, which we have also probed. Measuring the curvature at a large number of sample points throughout the state space allows us to look at summary statistics, like how many eigenvectors show nozero curvature and how much of the curvature is concentrated in some number of the eigenvectors. We can compare these summary measures to each other for curvature points near the basis vectors, curvature at natural image points and curvature of the iso-response surfaces at both of these sets of points. This is also carried out for the second-layer neurons from the Karklin & Lewicki network.

A simple summary of the curvature measure over many points is the distribution of the principal curvature values. For each point at which the curvature is measured, there will be 40 values contributing to the distribution (because the 64D state space has been reduced to 40D by whitening). From the histograms below, we see that the distribution depends on the type of surfaces that are being measured. The insight these summary measures yield comes from comparisons between the distributions.

Fig 4.7 shows that the curvature of the iso-response surfaces measured at the basis vector image were nearly all negative, indicative of pure selectivity. There were a handful of positive values that showed up for the more overcomplete networks, and this is probably due to the fact that the basis vectors were not the exact point in state space that evoked the maximum response for a given contrast. The magnitude of every positive value was less than 0.001 at a radius of 1, so

the positive curvature was extremely small. The sparse coding network therefore produces neurons with almost pure selectivity.

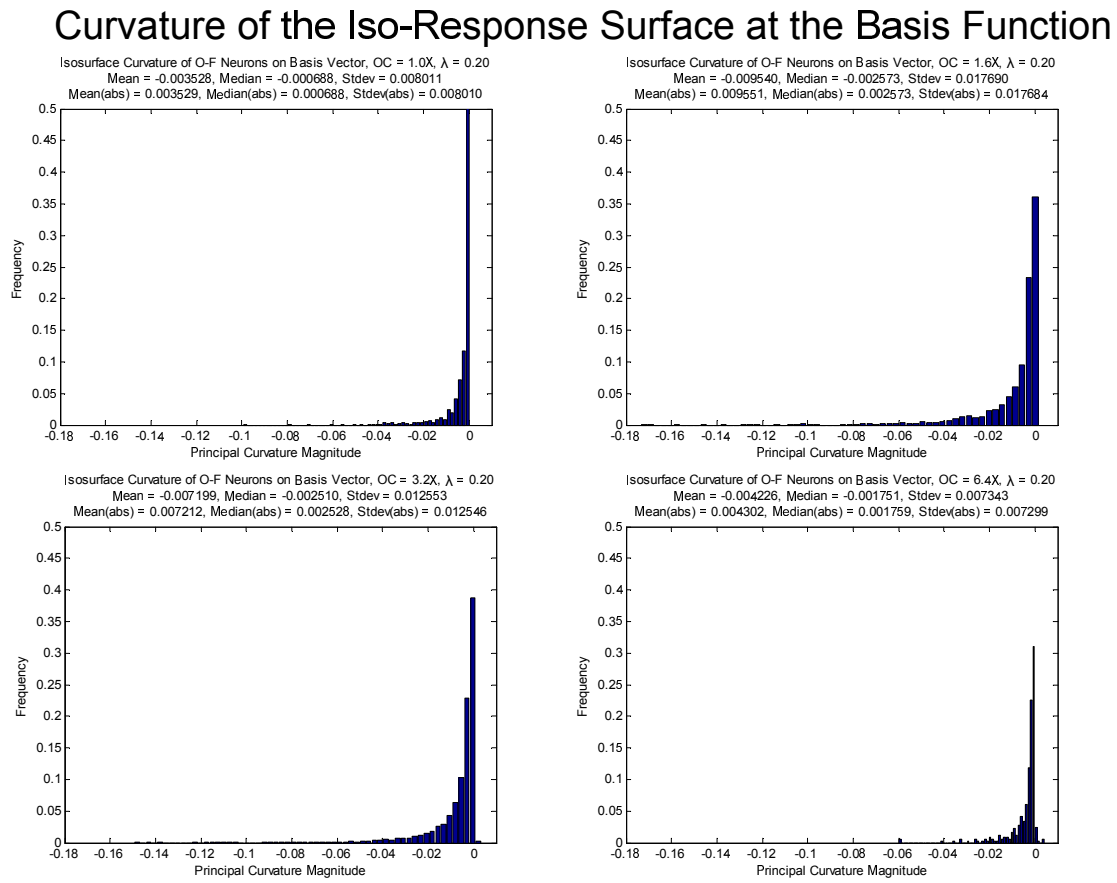

Figure 4.7: The distribution of principal curvature magnitudes for points on the basis vectors on the iso-response surfaces of neurons from the Olshausen & Field network at four degrees of overcompleteness.

A simple comparison to make is between the curvatures of response surfaces for the Olshausen & Field network in Fig. 4.8a with the second-layer neurons from the Karklin & Lewicki network in Fig. 4.8b. The immediate difference is simply the magnitude of the distributions (note the different scales on both axes). The principal curvatures for the second-layer neurons are two orders of magnitude greater than what is found in the single-layer network.

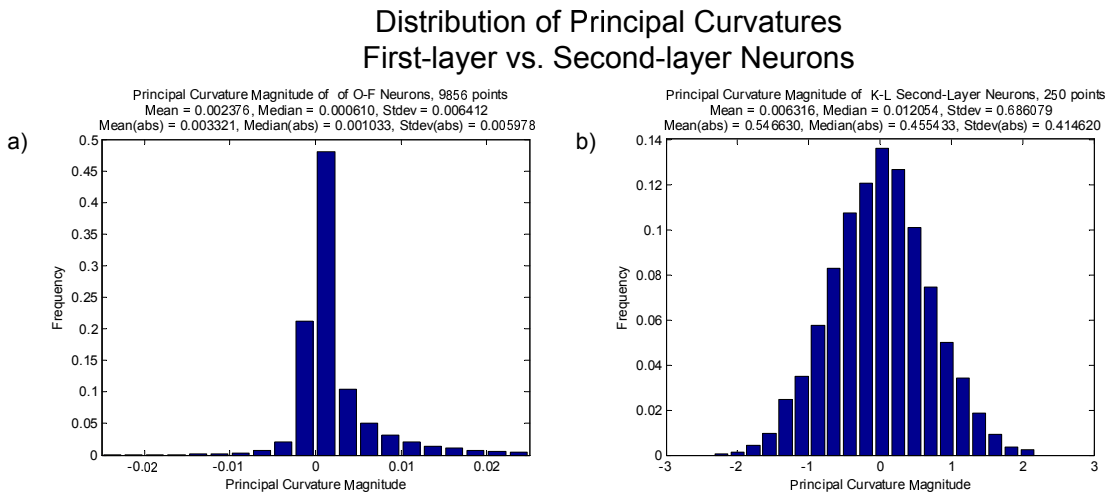

Figure 4.8: a) The distribution of principal curvature magnitudes for points near the basis vectors on the full response surfaces of neurons from the Olshausen & Field network. b) The distribution for second-layer neurons from the Karklin & Lewicki network. The median value is two orders of magnitude larger than the median value in a).

Fig. 4.9a shows the distribution of curvatures of the response isosurfaces for points near the basis vectors, while Fig. 4.9b shows the distribution for isosurfaces at natural scene points. The magnitude of curvatures does not seem to be different for white noise images and natural scenes.

A related measurement is the mean curvature of a surface at a point. This is the mean value of the principal curvatures. Here, we compare the distributions of mean curvature for the full response surfaces of Olshausen & Field neurons to the isosurfaces of Olshausen & Field neurons in Fig. 4.10. The isosurface mean curvature values are strongly skewed positive, with 87% of the points having mean curvature greater than zero.

From an example of the curvature at a single point on a neural response surface,

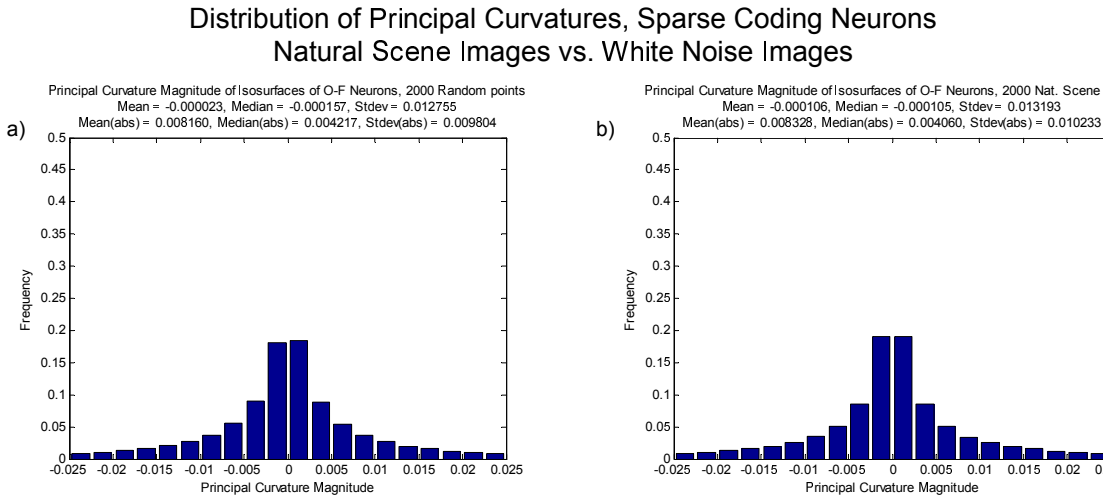

Figure 4.9: a) The distribution of principal curvature magnitudes for points on the isosurfaces of neurons from the Olshausen & Field network probed at white noise images at response magnitude 0.08. b) The distribution of principal curvature magnitudes for points on the isosurfaces of neurons from the Olshausen & Field network probed at natural scene points at magnitude 0.08. These are quite similar, so at this level of analysis there is not much of a difference between the curvature for natural scenes and white noise images.

it is clear that a large percentage of the curvature is concentrated in a small number of principal curvatures. In order to quantify this, we calculated the number of principal curvatures/eigenvalues which account for 95% of the total curvature. This is a measure of the degree of nonlinearity of the response surfaces. For 64D, 8x8-pixel natural image patches, the whitened space is 40D, and the mean and median number of principal curvatures that account for 95% of the total curvature over 9856 points are both about 18 dimensions, shown in Fig. 4.11. This means that 95% of the principal curvature is concentrated in less than half of the dimensionality of the response surface. For second-layer Karklin & Lewicki

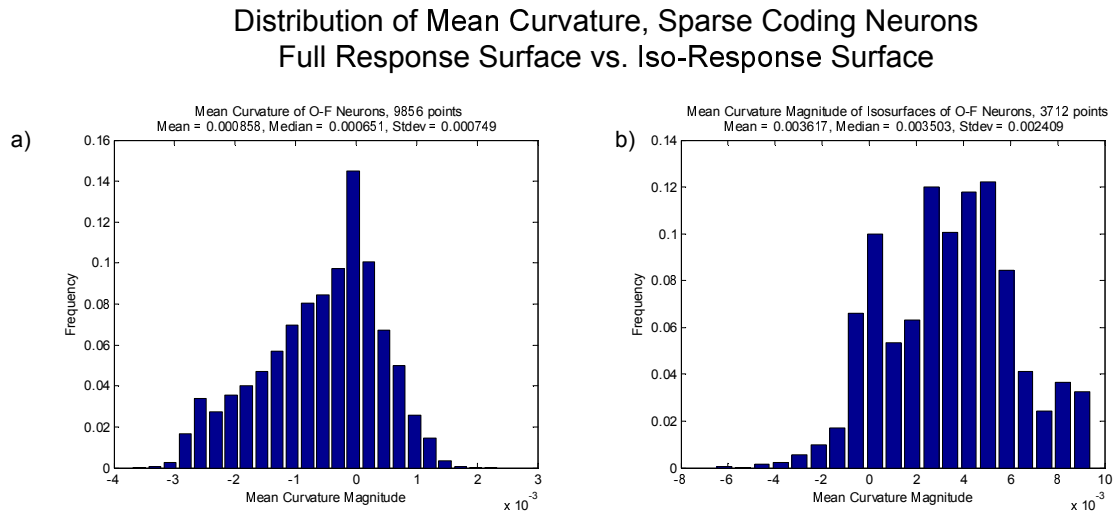

Figure 4.10: a) The distribution of the mean curvature (average of all principal curvatures) for response surfaces of neurons from the Olshausen and Field network. b) The distribution of the mean curvature isosurfaces of neurons from the Olshausen and Field network. Note 87% are positive.

neurons, the median value is 31 dimensions, so the curvature of these neurons is spread out over far more of the state space than the single-layer neurons.

The curvature depends both on the neural response surface and the point at which it is being measured. The curvature tends to fit the idea of pure selectivity closest to the basis vectors, and there is also a dependence on the total magnitude of the curvature as a function of the angular distance of the basis function from the point at which curvature is measured. This was quantified by calculating the absolute mean curvature for the response surfaces at each point (the sum of the absolute value of the principal curvatures) and plotting that data as a function of the angle. There is clearly a roughly linear relationship, as in Fig. 4.12, which simply confirms the idea that curvature is stronger the closer the point of measure is to the basis function.

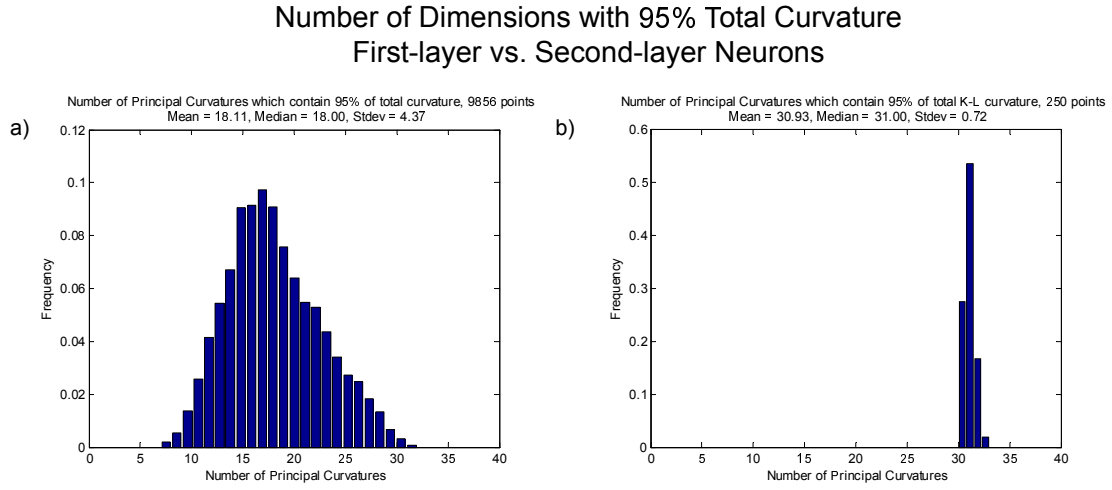

Figure 4.11: A histogram of the number of principal curvatures/eigenvalues for isosurfaces which account for 95% of the total absolute curvature. a) Olshausen & Field neurons, with a median of 18 dimensions. b) Karklin & Lewicki second-layer neurons, with a median value of 31 dimensions.

In Fig. 4.13, we present a quantification of the effect of both overcompleteness and the value of λ. We measured the curvature of the isoresponse surfaces of neurons from the Olshausen & Field network at 1000 natural image points for different values of overcompleteness and λ. Fig. 4.13 shows plots of the mean curvature (the average of the principle curvature magnitudes) as a function of the angle between the basis vector and the natural image point at which curvature was measured. Generally, the closer the natural image is to the basis vector, the higher the curvature of the isoresponse surface of that basis function. When the image point is 90° from the basis vector, the curvature is generally zero. Finally, the curvature generally increases with overcompleteness and with the value of λ.

The plots of Fig. 4.13 can be summarized by the slope of the fit from each. The slope relates the angle between neurons to the curvature of the response surfaces,

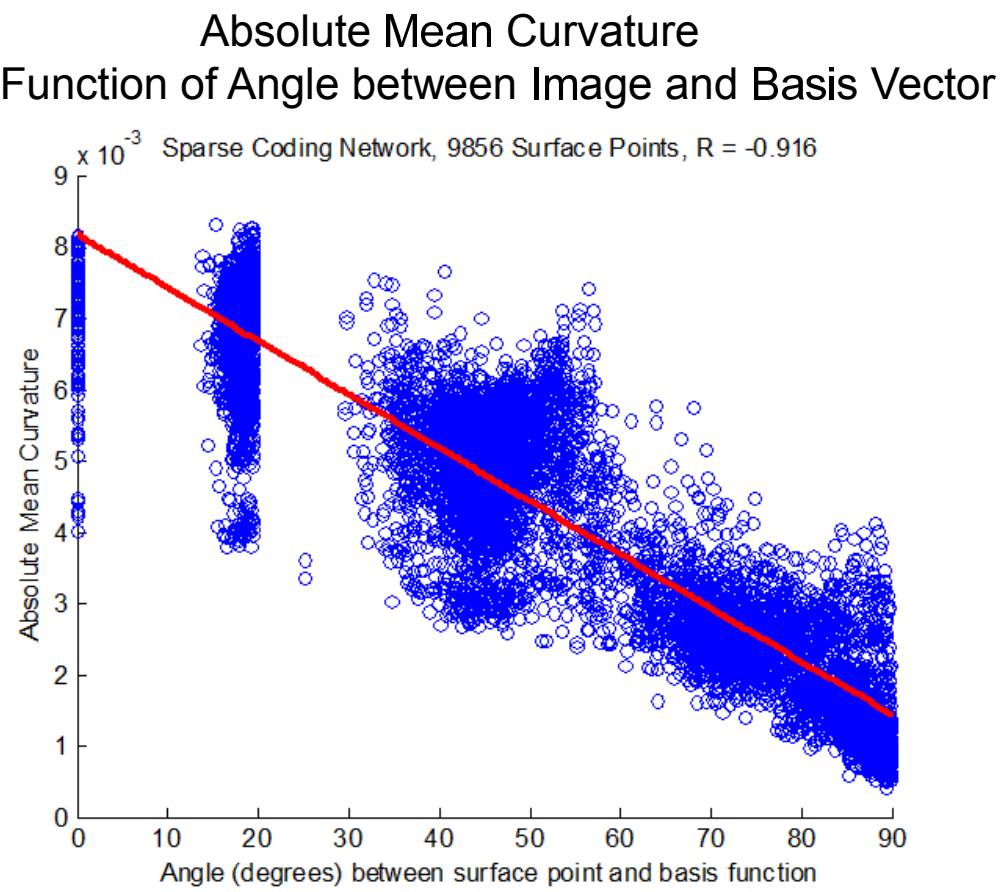

Figure 4.12: The absolute mean curvature as a function of the angle between the basis function of the response surface and the point at which the curvature is measured. Response surfaces tend to be flatter as the point moves away from the basis function. The gaps are due to the particular points that were sampled to be at a range of angles, and a full measurement of every surface would have points at every angle value.

and the slope is clearly a function of both the sparsity (λ) and the overcompleteness of the network. Fig. 4.14 shows a plot of the slope from each subplot of 4.13 as a function of sparsity and overcompleteness. Note that as a rule curvature increases with overcompleteness for a given lambda, but interestingly the highest slope is due to the medium level of sparsity.

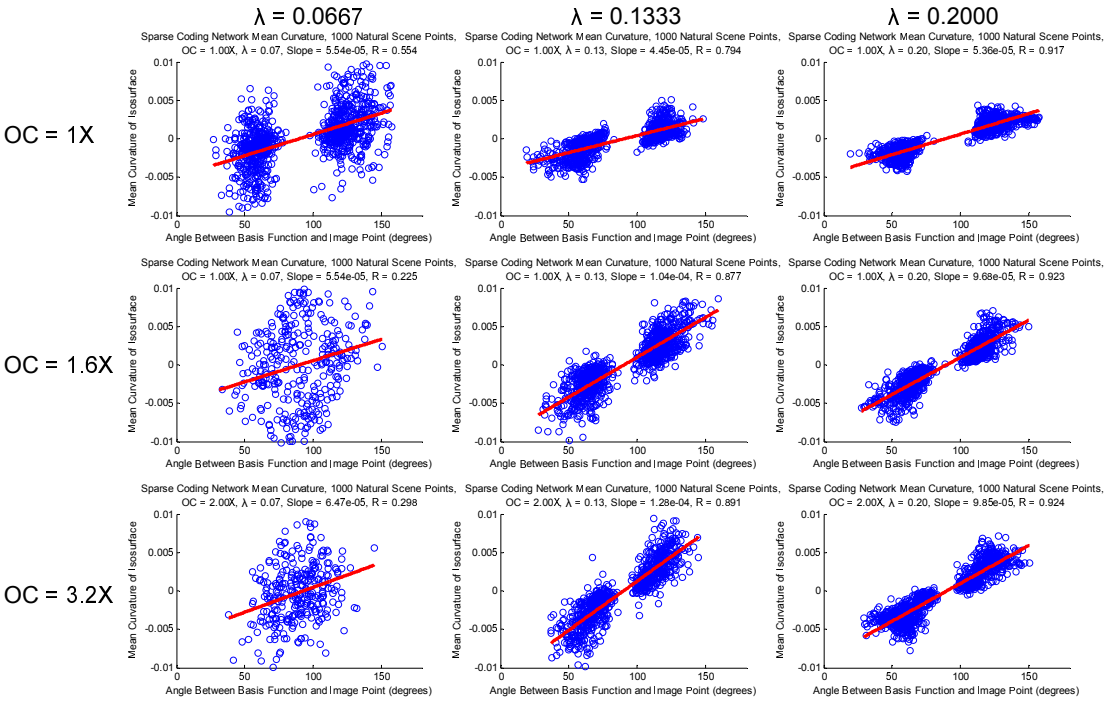

Figure 4.13: Plots of the mean curvature for isosreponse surfaces of Olshausen & Field neurons at natural scene points. Each plot is for response surfaces from a network trained at a certain degree of overcompleteness and a particular lambda value. The mean curvature is the average of the principal curvatures, and it is plotted as a function of the angle between the basis vector and the natural image point at which the curvature of the response surface was measured. Note that the mean curvature is about zero at 90°, and increases most dramatically for high overcompleteness and high λ. These plots demonstrate that mean curvature is a function of angle, sparsity and overcompleteness.

4.4 Conclusion

In this chapter, we have expanded the measurements of Ch. 3.2 into the full high-dimensional state space. We have considered how the curvature in the state space provides a direct quantification of the selectivity and invariance of a neu-

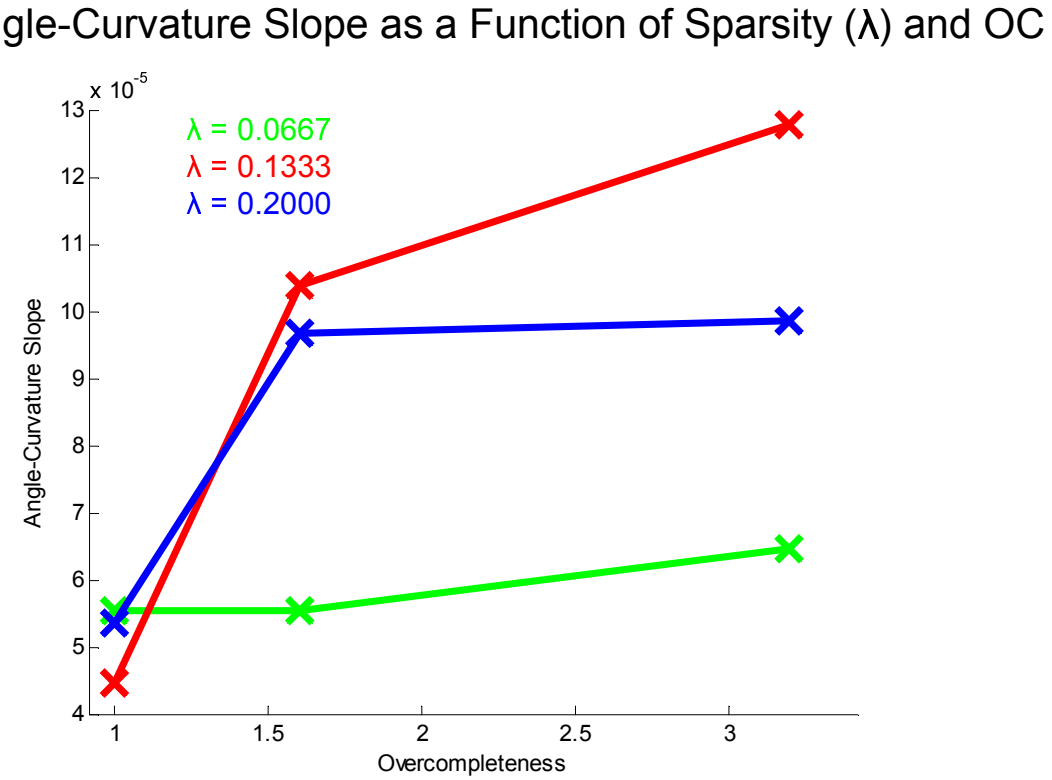

Figure 4.14: A summary of the effect of sparsity and overcompleteness on the curvature as a function of angle in the sparse coding network.

ron's response, and observed these measurements for a number of neurons. The increased complexity of the response surface of neurons from the second layer of the Karklin & Lewicki network was demonstrated, showing their increased curvature and feature selectivity as compared to neurons from the sparse coding network. We measured summary statistcs and showed that most of the curvature of a sparse coding neuron's response is concentrated in 18 of 40 possible dimensions, while it is concentrated in 31 dimensions for second layer neurons. We quantified the effects of sparsity and overcompleteness on the curvature of response surfaces, and showed that both generally increase curvature as a function of angle. This new measure of selectivity and invariance provides a new window into the features that drive

a neuron's response, both in the classic sparse coding network and a multi-layer network with more complexity.

CHAPTER 5

CONCLUSION AND FUTURE WORK

The application of methods from differential geometry to the high-dimensional surfaces that describe a neural response provides new insight into the nonlinearities of these neurons as well as the features to which they are selective and invariant. The sparse coding network is an example of a simple system defined by an energy equation that gives rise to great complexity in its solutions. The curved surfaces of neural responses in high dimensions result from the nonlinear optimization of the representation, and the surfaces have a polynomial complexity that certainly exceeds that of the local quadratic approximation used to measure curvature. It is interesting to consider the limits of what we can do in terms of mapping a function in a high dimensional space. Perhaps these surfaces could be accurately parameterized in some sort of feedforward network where the local curvature is dependent on the point at which curvature is being measured, its distance from the surface's basis function, and its distance from other basis vectors. This would likely allow for a concise understanding of the neuron's high-dimensional curvature.

A final theory of vision will include a generative model with units that learn response properties of visual neurons, both from low-level statistics and, in terms of object recognition, supervised training. The complexity of current deep networks that are extremely successful in certain object recognition tasks further highlights the decreasing importance of quantifying the behavior of single units if the network as a whole is accurate. The idea of a functional parameterization of neuronal responses in these network is not the end goal for those who design the networks; the performance in the task is the primary focus. However, there is some discussion that a better grasp of object manifolds as low-dimesnional subspaces within image

state space is necessary (Edelman, 1999; Field and Wu, 2004; DiCarlo and Cox, 2007). Since progress in object recognition with deep networks is necessarily linked to learning those object manifolds, it may be that this understanding has already been achieved in the field, but simply in a different way than what has been imagined.

Of course, as they say, two dollars and a grand unified theory of vision will still get you just a cup of coffee if there isn't any physiological evidence supporting it. A final theory of vision will include a model that can accurately predict the response of a visual neuron to a stimulus. These models will necessarily be highly nonlinear. It may be possible to map the nonlinearity of physiological responses using the methods discussed in Ch. 3, where responses over a particular low-dimensional subspace can be plotted to find iso-response contours. The methods of Ch. 4 may be somewhat more difficult to carry out, as precise measurements of curvature require exact differentials, and given the noisy nature of physiological responses it may not be possible to measure these accurately. However, that is an empirical question, and it is possible that certain approximations could also be used in place of the curvature measurement to quantify selectivity and invariance in the high-dimensional state space (Tsai and Cox, 2015).

Some of the greatest achievements in science include discoveries that came as unforeseen predictions of a mathematical framework. When Einstein completed the theory of general relativity, he had proposed a precise quantitative relationship between local mass/energy and the curvature of spacetime. Within one of the first solutions lurked what seemed to be an impossible outcome, but what was later called a black hole. As for vision science, we have seen a method for seeing "chimerical colors" (by putting cone photoreceptors into unnatural states

of fatigue) emerge from the opponent-theory of color vision (Churchland, 2005), and arguably many illusions fit the bill to some degree. Is it possible that more extreme predictions could fall out of future theories of vision?

It is interesting to consider whether a mathematical framework like relativity could be formulated to quantify how neural representations curve high-dimensional image space, but it seems that the sparse coding network is already such an example, and that deep networks are as well. The trend toward more unconstrained databases of labeled objects is certainly a step in the right direction, although this is of course limited by the extent and accuracy of the labor-intensive labeling process. The work of Rumelhart and Hinton has already revolutionized the field (Rumelhart et al., 1988), and with future increases in computing power this may be all that is necessary. Once deep networks surpass human performance at object labeling, an aspect of vision may indeed be solved. Of course, vision is so much more than object recognition (Edelman, 2009; Lewicki et al., 2014). When we take in a scene, the knowledge we both bring in to interpret what we see and take away for interpretive and behavioral use is far from being replicated in an artificial system. It may be that vision will not be truly solved until everything from photoreceptors to behavior is replicated in an artificial system.

APPENDIX A

MEASURING CURVATURE IN HIGH DIMENSIONS

The space that represents the set of possible images that humans can see is extremely large, likely on the order of a million dimensions (corresponding to the number of photoreceptors, or retinal ganglion cells, or an orthogonal set of those) (Edelman, 1999), and the response of each visual neuron is a surface in this high-dimensional space. How can we probe the curvature of these high-dimensional surfaces? The techniques for visualizing nonlinear responses described in Ch. 2 are obviously limited to low-dimensional projections of the image state space, and as a result there is necessarily a great deal of geometric complexity that is ignored. A surface in 3D can be projected onto 3 2D cross-sections, but the real structure of the surface is much more apparent when it is plotted in the full 3D space and can be observed from all perspectives. Unfortunately, for surfaces in image state space, visualizations will necessarily reveal only a hint of the true structure of the data. Fortunately, there is an extensive mathematical formalism for measuring certain aspects of high-dimensional data that turns out to be quite useful for this investigation. The goal of this chapter is to review the mathematical techniques that have been developed to measure the curvature of surfaces in high dimensions so that they can be applied to responses of neurons from the sparse coding network in Chapter 54.

Differential geometry was developed initially for cartography primarily by Gauss (Pressley, 2010), and its methods were extended to higher dimensions for use in physics as well as pure geometry. Insights were achieved formally through proof, but here we are forced to use these techniques for numerical measurements of curvature in high dimensions as a result of the probabilistic formulation of the

sparse coding network. Methods from differential geometry allow us to quantify responses of neurons in these networks in a new way that has the potential to be applied more broadly.

In this chapter, we will begin with an overview of the initial investigations into curvature with examples from cartography and physics. Next, an argument will be made for classifying a neural response surface as a particular type of manifold, which will allow the application of a specific type of curvature analysis. The equations for measuring several types of curvature will be derived and discussed at first in low dimensions and then expanded into high dimensions. The related measurement of the curvature of an isosurface will also be discussed. The limitations on curvature measurements of neural response surfaces due to the sparse coding network will be outlined. Finally, the curvature will be measured and discussed for several examples of quadratic forms in high dimensions.

Note that although the mathematics of curvature is well understood, there are no general purpose algorithms for measuring curvature numerically in high dimensions. The mathematics described in this chapter follows my personal path to writing the algorithms used to measure the curvature of neural response surfaces in high dimensions. With this in mind, some may find this chapter a bit too detailed, although I feel it is necessary to put the algorithms on a firm mathematical footing, and also to build an intuition for the output of the curvature measurements.

A.1 History of Curvature

Differential geometry and curvature can be used to explain why a map of the earth on a flat sheet of paper necessarily distorts distances. If one is uses a flat map to plot flights between destinations, the shortest routes will appear to be com-

plex nonlinear transforms of a straight-line path between origin and destination, and routes going north-south will involve different transforms than those going southeast-northwest. We believe this is roughly analogous to modeling nonlinearities in V1 neurons, as responses to different classes of stimuli each seem to be some complex transform of the input (Golden et al., 2015). In the case of the map, once one realizes that the underlying geometry is spherical, the shortest-distance routes are found by connecting the origin and destination with a great circle, or geodesic, on the surface of a globe. It is simply not possible to preserve distances (to create an isometry) between all points on a plane and a sphere. In some sense, this could be the case for V1 neurons as well, in that the nonlinearities may reveal themselves to be different manifestations of the underlying geometry.

Curved geometry has also been used extensively by physicists to describe light and gravity. The theory of special relativity, which quantifies the movement of objects approaching the speed of light, uses a hyperbolic geometry (via the Minkowski metric, $dx^2 + dy^2 + dz^2 - dt^2 = 0$) to describe empty spacetime (Lee, 1997). In general relativity, spacetime is distorted by mass and energy to produce gravity, and Einstein found a set of rules that quantify the resulting geometry. The equations for general relativity provide the form for how the metric describing spacetime deviates from that of special relativity given the local mass and energy landscape, for example at a point in space near a star (Egan, 2005). As one moves closer or further from a star, gravity changes because the spacetime metric changes. As opposed to the map/globe and special relativity, there is therefore no single underlying geometric structure that describes spacetime everywhere.

The sparse coding network learns its responses based on the probability density of natural images in image state space. The distribution of natural scenes is not

likely to be uniform in state space, like white noise images, so the closest analogue seems to be that of general relativity, which can describe how light will move through spacetime due to gravity for non-uniform distributions of mass. With this connection in mind, it is unlikely that one underlying geometry (or, more specifically, metric) will describe both selective and invariant V1 responses, because if neural response surfaces have curvature like what is found in general relativity, the metric will change over the high-dimensional image space. It may be possible, though, to generalize enough to formulate equations that describe how neural responses become nonlinear depending on, for example, the density of natural images in that volume of state space, as well as the response surfaces of other neurons in the state space. The responses of neurons in mammalian visual cortex certainly can be described using this language, but it remains to be seen what the geometry will turn out to be like.

A.2 The curvature of surfaces in high dimensions

As in Ch. 2, the response of a neuron to a 2D stimulus can be plotted in a third dimension, as in Fig. A.1. For each unique stimulus, or point in 2D state space, a neuron always produces the same response. The response is a function of the stimulus:

$$z = f(x, y) \tag{A.1}$$

The response is therefore a 2D surface in a 3D space, as it is a sheet with extent in the x- and y- dimensions but no extent in the z-direction, although it takes on different z-values depending on the point in the stimulus space. The important

point here is that the surface that describes the response is the same dimensionality as the state space and exists in a space that has one extra dimension. From the plots in Ch. 1, it seems reasonable to expect these 2D surfaces are nicely behaved in that they are continuously differentiable, although this is an open question that will be explored more below.

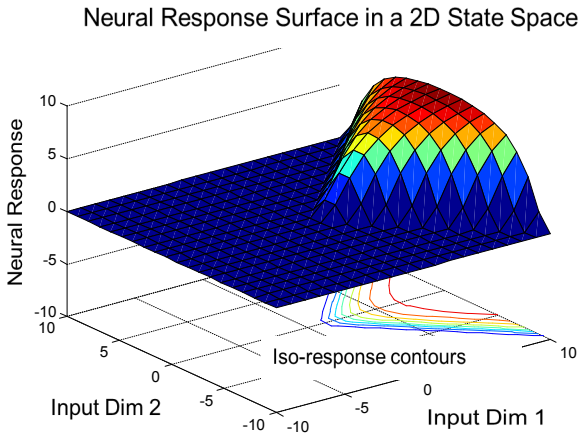

Figure A.1: A toy model of a neural response surface. The x- and y-axes represent the stimulus dimensions, while the response is plotted in a third dimension. A neural response over a 2D state space is described by a 2D response surface in a 3D space.

This idea of a response surface or manifold can be extended to higher dimensions: a neuron that has unique responses at each point in a 3D stimulus space can be represented by the response in a fourth dimension, or, equivalently, a 3D manifold in 4D space.

$$z = f(x_1, x_2, x_3) \tag{A.2}$$

Likewise, the response of a neuron in N-dimensional image state space can be plotted in an N+1th dimension, so the response is an N-dimensional manifold in an N+1-dimensional state space:

$$z = f(x_1, x_2, x_3, ..., x_{N-1}, x_N) \tag{A.3}$$

The N-dimensional manifold representing the nonlinear response of the neuron to every point in state space is an N-dimensional subset of N+1-dimensional Euclidean space. This fits the definition of what is known as an embedded submanifold of Euclidean space. This class of manifold with one value of z at each point in state space is also known as a hypersurface, and, as a result of its simplicity when compared with manifolds in non-Euclidean spaces, there are many results from differential geometry that can be used to describe hypersurfaces in high dimensions (Lee, 1997).

Before moving into high dimensions, I will describe the concept of curvature and derive the formula to calculate it, first for a one-dimensional path and then for a 2D surface. The curvature of a 1D path through 2D space refers generally to how much the path deviates from being a straight line. This can be formalized using derivatives: given a path through space, the curvature is how the tangent vector changes with path distance, when the tangent is a function of the distance along the curve (Pressley, 2010). If the path length s and tangent angle θ are parameterized as functions of x, the curvature κ is defined as:

$$\kappa = \frac{d\theta}{ds} \tag{A.4}$$

We can rewrite θ as $\tan^{-1}(\frac{dy}{dx})$, and if we assume that the path has been rotated so that the direction coincides with the x-axis, then we can assume $ds = dx$. This results in the following derivation (Weisstein, 2001a):

$$\kappa = \frac{d\theta}{ds} = \frac{d}{dx}\tan^{-1}(\frac{dy}{dx}) = \frac{\frac{d^2y}{dx^2}}{[(1+\frac{dy}{dx}^2]^{3/2}} \tag{A.5}$$

With a first derivative equal to zero, the curvature is simply defined as the second derivative of the function at a point.

Thus the curvature is the second derivative (after a translation to the origin and a rotation to the canonical axes) of a second-order approximation of the function locally. Curvature measures are always quadratic fits to a surface around a point of interest: they are local, not global, measures. Even though the local fit is quadratic, it is still an informative measure about a surface whose true polynomial degree is much higher. Ideally, a parameterization of the curvature could be made, which would be a global measure, because the curvature could be calculated for any particular point. This is much easier to do when the variables that affect the curvature are clearly defined; in relativity, a system will consist of something like a star centered at the origin of a coordinate space. For natural images, the dimensionality of the state space is so high that we cannot easily describe the distribution in that space that causes neural responses to curve. The distribution of natural images in state space cannot be known *a priori*, but the sparse coding networks serve as approximations to measuring this distribution; therefore curvature, initially as a local measure, is an important first step for characterizing neural responses in high dimensions. Ideally the curvature could be parameterized based on induction from many local measurements.

A.3 Deriving a Formula for Curvature

At this point, we have classified neural responses surfaces as a particular type of high-dimensional manifold called a hypersurface. We have examined the simple definition of curvature for a line in a 2D space. The goal of this section is to derive a formula for the curvature of a hypersurface in N-dimensional space. This formula requires the surface normal, the first derivative (gradient vector) and the second derivative (Hessian matrix) at a particular point on the surface. The initial deriva-

tion will be made for a surface in a 3D space, but this is trivially generalized to N dimensions. The curvature of a high-dimensional surface is somewhat cumbersome because there are many quantities of interest for different applications. We will define principal (extrinsic) curvature, mean curvature and Gaussian (intrinsic) curvature and link these to selectivity and invariance of a neural response surface. The reader is free to skip this section if she or he is not interested in the derivation of the curvature measure.

The curvature of a circle in 2D will serve as a useful point of reference for more complex manifolds. It can be shown that the curvature for a circle in 2D is the inverse of its radius (see Weisstein (2001a) for an analytic derivation of equation A.5). This fits well with intuition: the smaller a circle it is, the greater its curvature, while the larger a circle is, the smaller its curvature. This turns out to be useful in higher dimensions as well, because a 2D subspace of an N-dimensional sphere shows this $\frac{1}{R}$ curvature.

The analysis is more involved for a 2D surface in a 3D Euclidean space. The first derivative of the surface results in the gradient, which is a vector of partial derivatives in orthogonal directions, while the second derivative results in the Hessian, which is a matrix of second partial derivatives in all pairs of directions. The curvature can be calculated for an arbitrary point on the surface by translating the surface to the origin and rotating it such that the first derivatives are properly aligned with the axes. In order to derive how to calculate the characteristic or principal curvatures at a point on a surface, we will describe the first and second fundamental forms, which are matrices of constructed using the gradient and Hessian, as well as the shape operator, which is a particular combination of the fundamental forms. The shape operator and principal curvatures quantify how the

normal vector to the surface at a point changes with direction.

In order to find the general formula for curvature, we must introduce the first fundamental form, also known as the metric. It is a matrix of the outer product of the gradient vector g of a manifold with itself, and is used to calculate distance of a path on a manifold. The metric of any N-dimensional Euclidean space is simply the N-D identity matrix I. For the metric on an arbitrary surface, if $g = [\frac{\partial f}{\partial x} \frac{\partial f}{\partial y}]$ and $\frac{\partial f}{\partial x} = f_x$, then the first fundamental form is:

$$FF = g * g^T = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \begin{bmatrix} f_x & f_y \end{bmatrix} = \begin{bmatrix} f_x * f_x & f_x * f_y \\ f_y * f_x & f_y * f_y \end{bmatrix} \tag{A.6}$$

In order to use the metric to find the distance along a particular vector v over the surface, for a constant metric one computes $v^T * FF * v$ (or, if the metric is a function of position, integrates this quantity over the path length). For Euclidean space, where $FF = I$, this results in a statement of the Pythagorean theorem (Pressley, 2010), so computation of distance on a manifold with an FF different than the identity matrix results in a generalized version of the Pythagorean theorem. Different surfaces with the same FF are *isometric* to Euclidean space because they have the same distance measure; the canonical example is the plane and the cylinder. Gauss found a method for finding the curvature of a manifold that can be measured by an observer on the manifold based on the FF and its derivative, but for our purposes it is also useful for translating and rotating a manifold to the origin so that its curvature at a particular point can be calculated.

The next calculation necessary for the curvature formula is the second fundamental form. It is computed by taking an element-wise inner product of the Hessian matrix, composed of all the second partial derivatives of a function, with

an oriented normal vector to the surface at that point. As with our first definition of curvature, we are interested in how the normal vector of the surface changes as we move over the surface through the state space.

$$SF = \begin{bmatrix} f_{xx} * N & f_{xy} * N \\ f_{yx} * N & f_{yy} * N \end{bmatrix} \tag{A.7}$$

The FF and the SF can be combined to create the shape operator, which is another matrix that represents the second derivatives of a manifold at a point, but rotated so that its first derivatives are aligned with the coordinate vectors and the normal to the manifold is aligned with the z vector (Pressley, 2010).

$$SO = FF^{-1} * SF = \begin{bmatrix} f_x * f_x & f_x * f_y \\ f_y * f_x & f_y * f_y \end{bmatrix}^{-1} * \begin{bmatrix} f_{xx} * N & f_{xy} * N \\ f_{yx} * N & f_{yy} * N \end{bmatrix} \tag{A.8}$$

The shape operator allows the calculation of the principal curvatures, which will be the primary measure of neural response manifolds in high dimensions. With the SO, we can define the principal curvatures as the eigenvalues of the SO, and the principal directions as the eigenvectors of the SO. If we require that the tangent vectors to the manifold are parallel to the principal directions, we can define an equation for the curvature κ and substitute eigenvectors e for the displacement vector:

$$\begin{bmatrix} f_x * f_x & f_x * f_y \\ f_y * f_x & f_y * f_y \end{bmatrix}^{-1} * \begin{bmatrix} f_{xx} * N & f_{xy} * N \\ f_{yx} * N & f_{yy} * N \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \kappa * \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{A.9}$$

$$\begin{bmatrix} f_{x_1} * f_{x_1} & \cdots & f_{x_1} * f_{x_n} \\ \vdots & \ddots & \vdots \\ f_{x_n} * f_{x_1} & \cdots & f_{x_n} * f_{x_n} \end{bmatrix}^{-1} * \begin{bmatrix} f_{x_1 x_1} * N & \cdots & f_{x_1 x_n} * N \\ \vdots & \ddots & \vdots \\ f_{x_n x_1} * N & \cdots & f_{x_n x_n} * N \end{bmatrix} \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix} = \kappa * \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix} \tag{A.10}$$

By calculating the eigenvector decomposition of the shape operator, the eigenvectors represent the directions along which the second partial derivatives are parallel to the gradient $[dx\ dy]$. The eigenvectors of the shape operator are called the principal directions of curvature, and the eigenvalues correspond to their magnitudes. In 2D, the principal curvatures are an orthogonal basis where the eigenvectors represent the maximum and the minimum curvature of the manifold at that point.

With this definition of principal curvature in mind, we can define two other curvature measures based on principal curvature, and illustrate the calculation with a sphere. At each point on the sphere, the curvatures $\kappa_1 = \kappa_2 = \frac{1}{R}$. Each point is therefore an *umbilic* of the surface, and every direction in the tangent space has the same curvature. This is only true for the sphere, and these facts will be useful below as a point of comparison for manifolds in higher dimensions. To connect this to the earlier result that the curvature of a circle was $\frac{1}{R}$, we can define the intrinsic curvature at a point on the manifold as the product of the two principal curvatures, which will yield $\frac{1}{R^2}$. Another measure of curvature, called the mean curvature, is defined as the average of the sum of the principal curvatures, which yields $\frac{1}{R}$.

What about the curvature of a neural response surface in high dimensions? A neuron's response in N-dimensional image state space is described by a con-

tinuously differentiable N-dimensional manifold in an N+1-dimensional Euclidean space, or equivalently an N-dimensional hypersurface. Hypersurfaces are fortunately the simplest type of manifold in differential geometry, and the preceding curvature analysis generalizes to N dimensions in a straightforward manner. Consider a hypersphere in a 4D space (which corresponds to a 3D state space): $x_4 = \sqrt{(R^2 - x_1^2 - x_2^2 - x_3^2)}$. The manifold is 3D, and now there are three principal curvatures at each point, so there is an intermediate curvature between the maximum and minimum. The intermediate curvature is like the second principal component in PCA, in that it is the largest curvature in a direction orthogonal to the maximum curvature (Berkes and Wiskott, 2006). Again, the three principal directions will be orthogonal, as they are eigenvectors of the shape operator. For a 3D sphere in 4D space, we can also define the sectional curvature at a point on the sphere of a particular orthonormal 2D subspace of the 4D state space as the product of principal curvatures in each of the two dimensions of the desired subspace. The sectional curvature for a sphere in any dimension is therefore $\frac{1}{R^2}$. The derivation of the shape operator as well as the principal curvatures can just as easily be done for a point on a sphere in N dimensions. The dimensions of the matrices become N-dim x N-dim, and there are N resulting principal curvatures/eigenvalues.

A.4 The curvature of isosurfaces in high dimensions

We have derived the formula for measuring the curvature of a neural response surface in a high-dimensional state space. However, as described in Ch. 1, we are not directly interested in the curvature of the hypersurface in the full state space, but rather the curvature of an iso-response surface of the full response surface. In order to measure the curvature of an iso-response surface in high dimensions, one

further derivation is necessary.

We are ultimately interested in utilizing the measurement of principal curvatures to calculate the features to which a neuron is selective and invariant. We have argued in Ch. 1 that a selective response is defined by curvature of the iso-response surface toward the normal at the point of maximum response for a given RMS contrast, and that tolerance/invariance is defined by curvature away from the normal. These two types of curvature are identical to hyperbolic curvature (selectivity), where the iso-response line bends toward the normal, and spherical curvature (tolerance/invariance), where the iso-response line bends away from the normal. Here we define these to be respectively negative (selective) and positive (spherical). These definitions hold in high dimensions: if all of the N principal curvatures at a point are positive, then the isosurface curves away from the normal, and the response is purely convex or purely tolerant. A sphere is purely convex, because all of its principal curvatures are positive. If all of the N principal curvatures are negative, then isosurface is purely concave and purely selective. We need a measure of the high-dimensional curvature of the isosurfaces in order not only to find the features of interest, but to test for the dimensionality of the selective and invariant subspaces of a neuron's response.

Although there is clearly a link between the curvature of a full surface and the curvature of one of its isosurfaces, this is not straightforward. A purely convex full surface necessarily has purely convex isosurfaces, but there could be a full surface that is not purely convex with purely convex isosurfaces. It is necessary to define an isosurface as a function on the full surface in order to measure its curvatgure. Following the derivation in (Chang et al., 2010) and (Bian et al., 2011), we start with an N-dimensional hypersurface $u(x_1, x_2, x_3, ..., x_{n-1}, x_n)$. To find an

iso-response surface, we assume

$$u(x_1, x_2, x_3, ..., x_{n-1}, x_n) = u_0 \tag{A.11}$$

The iso-response surface is the set of coordinates in the state space that give rise to u = u0. We can define this surface by generating a function $x_n = v(x_1, x_2, x_3, \hat{a}\check{A}\c{e}, x_{n-1})$. This says that the n-th coordinate of the iso-response surface is a function of the first through the (N-1)-th coordinate. For a sphere in 3D space, where $z2 = \sqrt{(R^2 - x^2 - y^2)}$, we would be searching for a surface with a constant value, which would be a circle. For each value of x, there would be some value $y(x)$ that would define the iso-response circle. The iso-response surface in N dimensions is then:

$$u(x_1, x_2, x_3, ..., x_{n-1}, v(x_1, x_2, x_3, ..., x_{n-1})) = u_0 \tag{A.12}$$

To find the function $v(x_1, x_2, x_3, ..., x_{n-1})$, we take the derivative of A.12 with respect to an arbitrary dimension i:

$$u_i + u_n * v_i = 0 \tag{A.13}$$

By the chain rule, we see $v_i = -u_i/u_n$. The gradient of v is $(v_1, v_2, ..., v_{n-1})$, and the first fundamental form matrix may be calculated by taking the outer product of this vector with itself.

To find the components of the second fundamental form, the Hessian matrix of v may be found by differentiating eqn. A.13 again:

$$u_{ij} + u_{in} * v_j + u_{nj} * v_i + u_{nn} * v_i * v_j + u_n * v_{ij} = 0 \tag{A.14}$$

where solving for v_{ij} and substituting for v_j from A.13 above yields

$$v_{ij} = [u_n^2 * u_{ij} + u_{nn} * u_i * u_j - u_n * u_j * u_{in} + u_n * u_i * u_{jn}]/u_n^3 \qquad \text{(A.15)}$$

The inner product of this second derivative with the unit normal to the iso-response surface in each dimension yields the second fundamental form for the iso-response surface. The surface normal in this case is simply the gradient of the full surface at the point of interest. In order to obtain the principal curvatures, we use eqn. A.14 to find the shape operator for the surface at this point and take the eigenvector decomposition.

A.5 The meaning of curvature in high dimensions

With the algorithm for surface curvature in high dimensions in hand, the next step is to test the algorithm and verify its results. The principal curvatures of a 2D surface in a 3D Euclidean space have a clear geometric interpretation that can be visualized. The curvatures of an N-dimensional surface in N+1-dimensional Euclidean space cannot be visualized except within low-dimensional subspaces. The principal curvatures of surfaces in high dimensions are defined as a measure of how quickly the surface bends toward or away from the normal at a point (Fig. A.2). We have made the argument that principal curvatures of a neuron's response surface correspond to the magnitude and direction of their selective and invariant responses. Here, we will visually examine the principal curvatures of analytically-defined quadratic surfaces in order to get an intuitive understanding for what the principal curvatures mean.

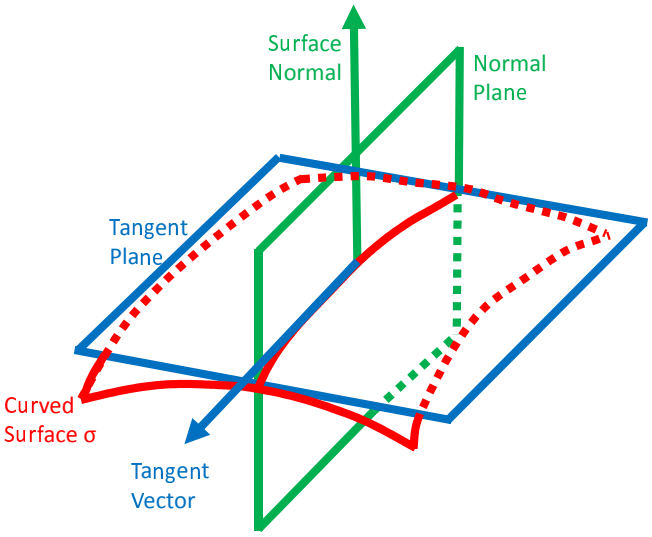

Figure A.2: Redrawn from (Pressley, 2010), pg. 128. A surface σ (red), with the normal vector and normal section in green and the tangent vector and plane blue. These normal and tangent vectors define the plane called the normal section, which is orthogonal to the tangent plane of the surface at the point.

It is important to keep in mind that we do not know how complex neural response surfaces are in terms of polynomial degree; curvature is a local quadratic fit, so it is likely that the curvature of neural response surfaces changes at every point. Plots of 2D subspaces of neural responses can certainly be fit to quadratic surfaces locally, but they are unlikely to be a good fit globally. The subspace plots may therefore not clearly correspond to our intuitive ideas about curvature from quadratic surfaces, even though the principal curvatures are accurate local descriptions. The familiar 2D quadratic form is defined by the following equation:

$$R^2 = c_1 * x^2 + c_2 * y^2 + z^2 \tag{A.16}$$

$$z = \sqrt{R^2 - (c_1 * x^2 + c_2 * y^2)} \tag{A.17}$$

When $c_1 = c_2 > 0$, the surface is a sphere and the principal curvatures are both $\frac{1}{R}$. If $c_1 \neq c_2$, but $c_1 > 0$ and $c_2 > 0$, then the surface is an ellipse. When $c_1 = 0$ or $c_2 = 0$, the surface is a parabola, and $\kappa = \frac{1}{R}$, while $\kappa = 0$. For $c_1 > 0$ and $c_2 < 0$ (or *vice versa*), the surface is a hyperbola, and $\kappa = \frac{1}{R}$, $\kappa_2 = -\frac{1}{R}$.

For a hyperboloid in 3D ($x^2 + y^2 - z^2 = R^2$), for $R = 1$, the curvature can be shown to be $\frac{1}{(1+2*z^2)^2}$ at every point (Weisstein, 2001b). This surface is more interesting than the sphere, because its curvature is different at every point and the numerical curvature measurements can be compared to the analytical result. A highly accurate numerical derivative toolbox called ADMAT (Automated Differentiation for Matlab) was used and the curvature was calculated using the derivatives. The numerical curvature calculation was exact, as shown in Fig. A.3, and accurate to 16 decimal places. This served as an initial validation that the numerical curvature algorithm was producing measurements that agreed exactly with analytical results.

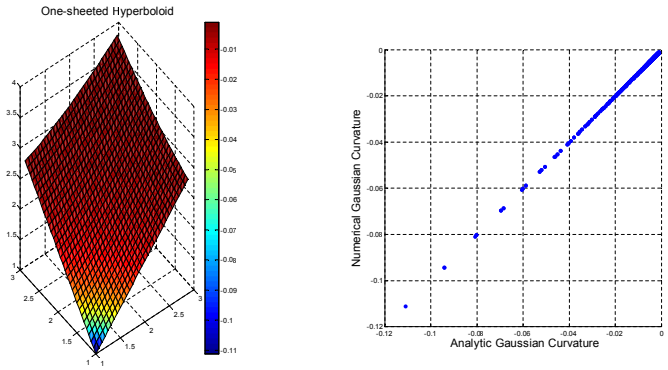

Figure A.3: For $x^2 + y^2 - z^2 = 1$, we show the surface with the numerical curvature on the left, and a plot comparing the numerical curvature and the analytically determined curvature. Note the perfect correspondence, supporting the accuracy of the numerical measurement process.

Let us consider the general quadratic form for a 3D surface in 4D defined by

the equation:

$$R^2 = c_1 * x_1^2 + c_2 * x_2^2 + c_3 * x_3^2 + c_4 * z^2 \tag{A.18}$$

$$z = \sqrt{[-(c_1 * x_1^2 + c_2 * x_2^2 + c_3 * x_3^2) + R^2]/c_4} \tag{A.19}$$

The coefficients of the variables again determine the nature of the surface. When $c_1 = c_2 = c_3 = c_4 > 0$, the surface is a sphere. When $c_i > 0$, the surface is an ellipse. If $c_i > 0$ and $c_j < 0$, then the surface is a hyperboloid.

For a visualization, we will begin with a cylinder, where $c_1 = c_2 > 0$ and $c_3 = 0$. Since the state space is 3D we can visualize the surface using its isosurfaces in Fig. A.4.

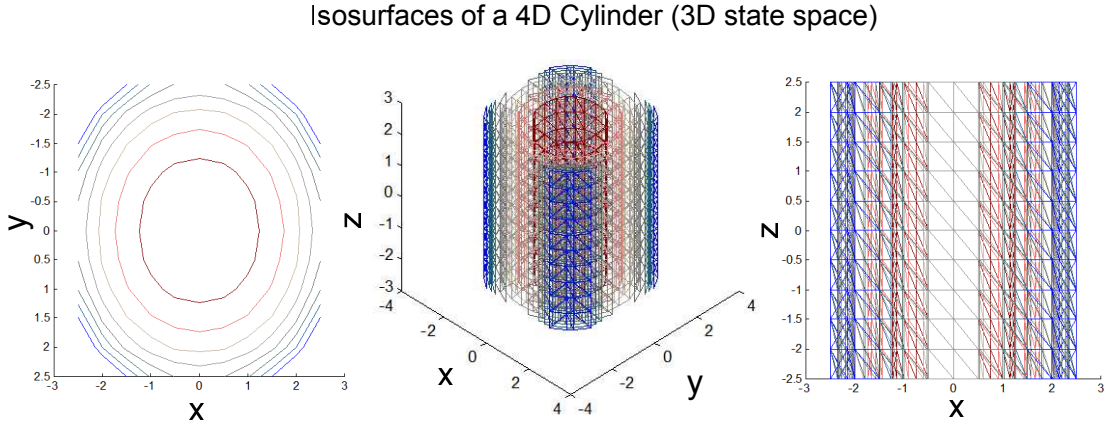

Figure A.4: A cylinder in a 3D state space defined by $z = \sqrt{[-(x_1^2 + x_2^2 + 0 * x_3^2) + 64]}$. Note c_3, the coefficient of x_3, is set to zero. In other words, the isosurfaces are circles in the $x_1 - x_2$ plane and unevenly spaced planes in the $x_1 - x_3$ and $x_2 - x_3$ planes.

Now we can also view the surface normal (green arrows) as well as the principal directions scaled by the principal curvatures (red arrows) at a number of points on the surface, as in Fig. A.5. The surface normal should always point to the center of the cylinder, and two of the principal curvatures will be equal (for the circular isosurfaces), while the third will be zero.

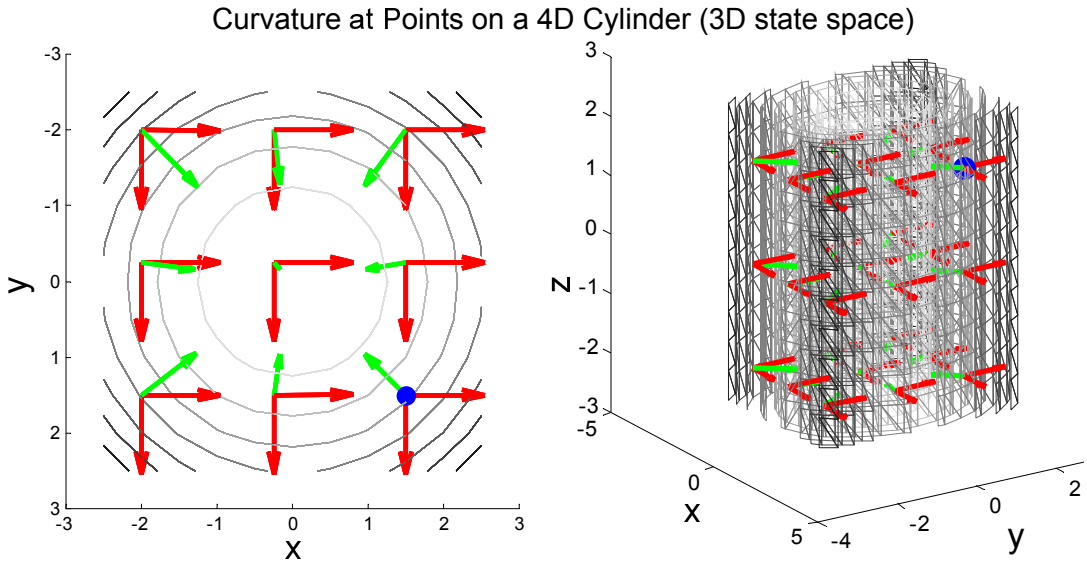

Figure A.5: Green arrows indicated the surface normal, while the red arrows are the principal directions scaled by the principal curvatures (red if $\kappa > 0$, blue if $\kappa < 0$; no κ is less than zero for this surface). Isosurfaces are represented in grayscale. A) Points on the cylinder in 3D state space in the $x_1 - x_2$ plane. Note how the surface normal always points to the origin. B) The full 3D state space; note the cylindrical isosurfaces.

In order to form an intuition for principal curvatures that will be useful in higher dimensions, we will examine the principal curvatures via the normal sections of the surface in Fig. A.6. A normal section is a two-dimensional subspace defined by the normal vector to the isosurface (which is the gradient vector of the full surface in state space) and one of the principal directions (vectors in the tangent space of the

surface) centered around the point at which curvature is measured. Since there are three principal curvatures for a point, there will be three normal sections. It is clear from Fig. A.5a that the principal curvatures (red) are not always orthogonal to the isosurface normal vector (green, which is also the gradient of the full surface).

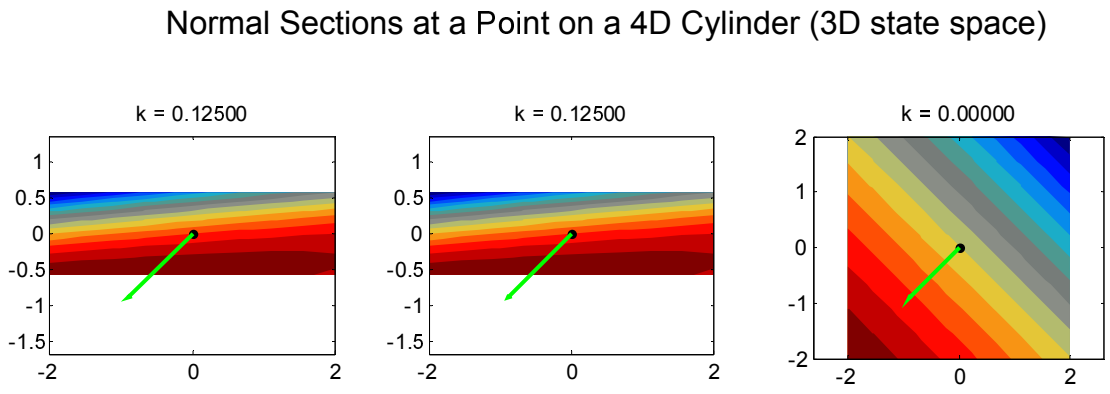

Figure A.6: The normal sections of the cylinder at the blue point indicated in Fig. A.4. The principal curvatures are measured locally at the origin in these plots (note: this does not correspond to the origin in Figs. 2 and 3) and the direction of the surface normal is indicated by the green arrow. The magnitude of the principal curvature is labeled at the top of each normal seciton. Note that for the two sections with nonzero principal curvatures show curvature in the isocontours, while the section with zero principal curvature shows straight, evenly-spaced contours.

This lack of orthogonality between the normal vector and the principal vector is the reason the section is compressed in Fig. A.6a and A.6b. Generally, the normal sections of the function convey that positive curvature corresponds to a local curvature toward the surface normal, and zero curvature results in straight and evenly spaced isocontours. We will see below how negative curvature results in isocontours that bend away from the surface normal.

Now consider a function with $c_1 = 1, c_2 = 0.33$ and $c_3 = -0.5$. This results

in a function with isosurfaces corresponding to both ellipses and hyperboloids, as show in Fig. A.7. Again, the principal curvatures and the surface normals are drawn on the surface for a range of points. Principal curvatures greater than zero are drawn in red, and less than zero in blue. Unlike the cylinder above, there are actually points with negative principal curvatures on this surface. This basically corresopnds to elliptic isosurfaces of Fig. A.7a as well as hyperbolic isosurfaces of Fig. A.7c. Additionally, each point now has three nonzero principal curvatures, making the sectional curvatures more interesting.

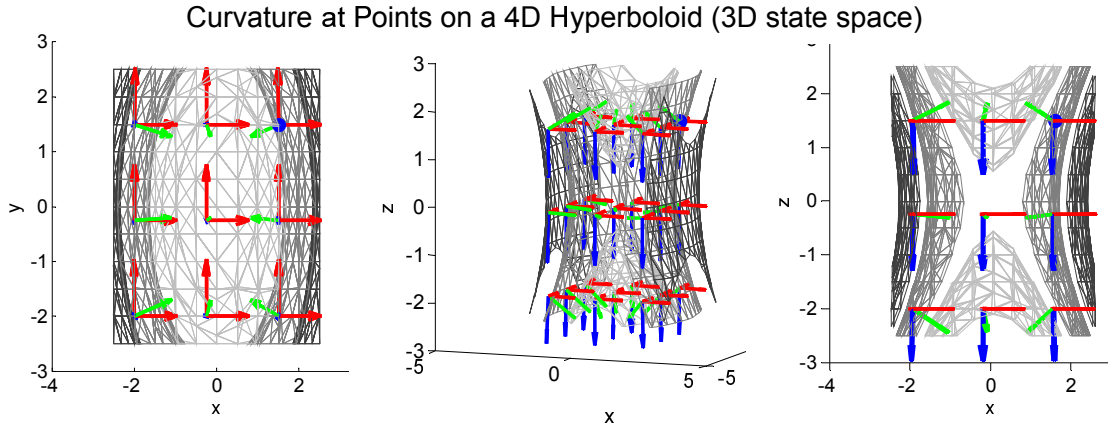

Figure A.7: The isosurfaces for a quadratic surface defined by $z = \sqrt{[64 - (1 * x_1^2 + 0.33 * x_2^2 - 0.5 * x_3^2)]}$. Principal curvature vectors are red if $\kappa > 0$ and blue if $\kappa < 0$. Note the elliptic view of the isosurfaces present in a) along the $x_1 - x_2$ plane (both positive coefficients) and the hyperbolic isosurfaces in c) along the $x_1 - x_3$ plane (a positive and a negative coefficient). The fact that there are three nonzero curvatures at each point makes the sectional curvatures more interesting than in the case of the cylinder.

The normal sections for the blue point from Fig. A.7 show two positive principal curvatures, as we saw above in Fig. A.5 for the cylinder. The third principal curvature is negative, which clearly curves away from the normal vector in the

opposite manner of the two positive curvatures. As for the sectional curvatures, one is elliptic and two are hyperbolic, as must be the case for $\kappa_1, \kappa_2 > 0$ and $\kappa_3 < 0$. Note the similarities between the two hyperbolic sectional curvatures, and how they differ from the elliptic section.

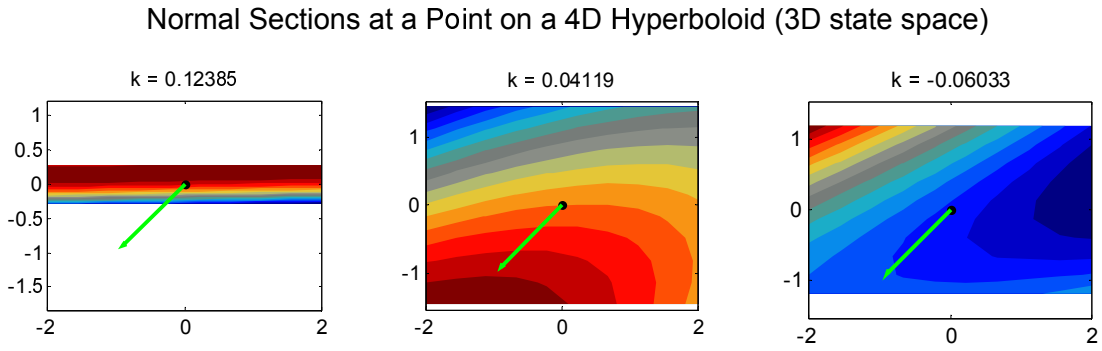

Figure A.8: The normal sections for the quadratic surface shown in Fig. A.7. Note the negative curvature denoted by the blue vectors, where the surface curves away from the normal vector.

These visualizations reinforce the analytic results that describe the curvature of a function. They can also be used in higher dimensions as well. For a sphere in N = 9 dimensions with $R = 4$, every normal section should show positive curvature equal to $\frac{1}{R} = 0.25$. This is indeed confirmed by the sections in Fig. A.9.

For a surface with positive as well as negative curvature, the normal sections through a point demonstrate that negative curvature corresponds to curvature toward the surface normal.

The principal curvatures were calculated for hyperspheres in dimensions 4, 9, 16, 25 and 64 for a range of radius values, and again all the principle curvatures were exactly $\frac{1}{R}$. Below in Fig. A.11 is an example of the output produced by the algorithm for a sphere in 64D with $R = 2$. There are 64 principal curvature

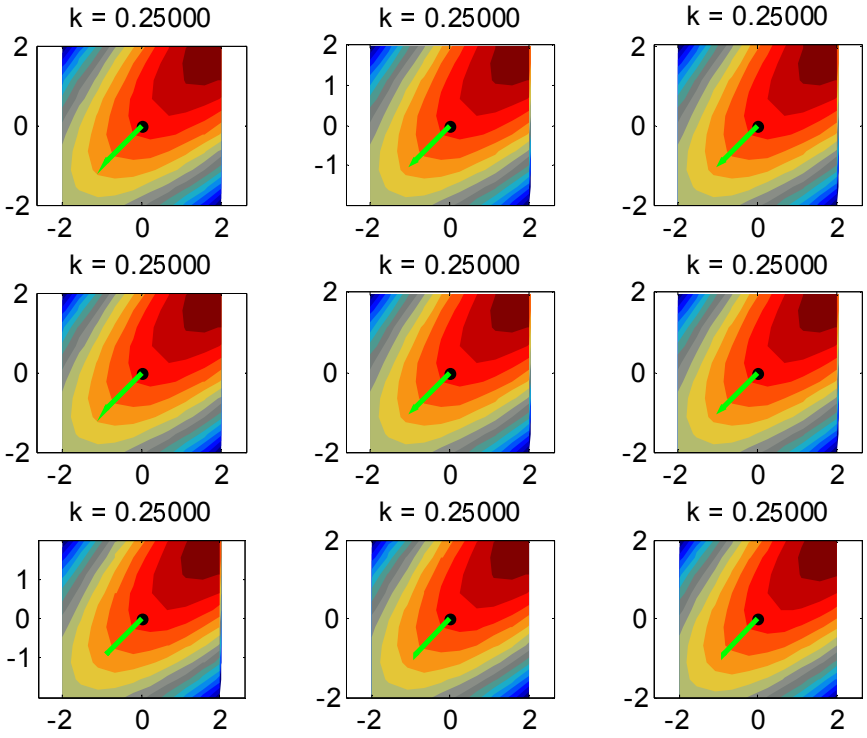

Figure A.9: The normal sections for a 9-dimensional sphere with R = 4, supporting our intuitions about positive principal curvature indicating that the surface bends away from the normal.

magnitudes and directions, with the magnitudes all equal to 1/R in the stem plot at left, and the directions in the 8x8 image patch plot at right. A principal direction in 64D space can be shown as an image, and when the curvature algorithm is applied to neural response surfaces, the images will allow some interpretation in the context of the basis functions/receptive fields of the network. For the hypersphere, since the curvature is the same in any direction at all points, any direction is a principal direction. Here, the algorithm finds the coordinate basis of the state space as the principal directions. In the 8x8-pixel image of each principal direction, all of the gray pixels correspond to a value of zero, while the sole white pixels corresponds

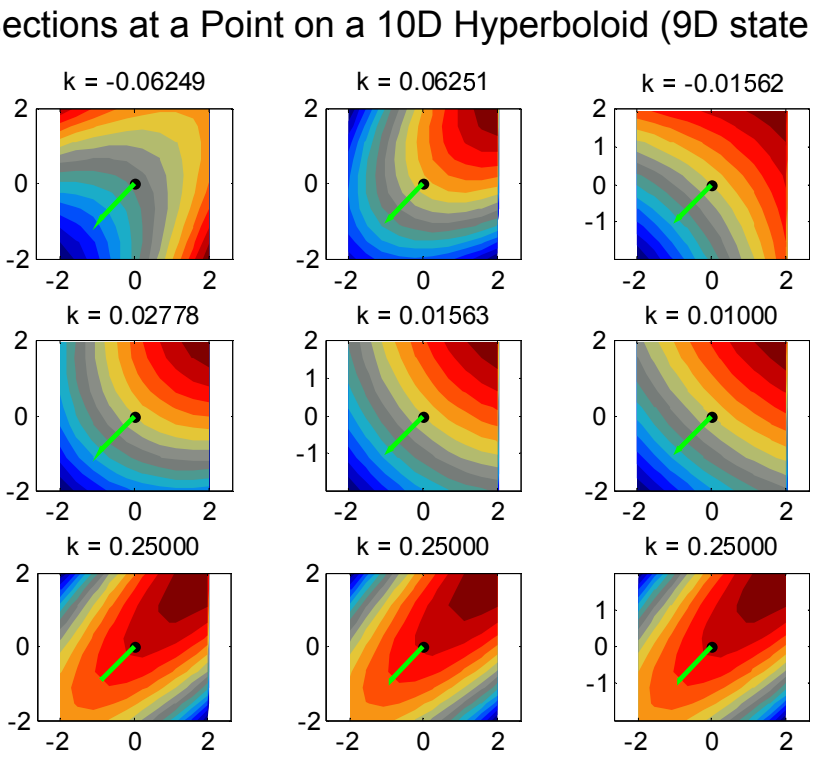

Figure A.10: A quadratic surface in 9D with simultaneous positive and negative curvature. Negative principle curvature implies the surface curves away from the normal.

to a value of one.

A.6 Conclusion

The next step is to apply measures of curvature to responses of model neurons in high-dimensional state spaces with the hope that these techniques (or similar ones) could be applied to responses of real visual neurons. In Chapter 1, we argued that the responses of neurons that are described as selective nonlinearities can all be described by curvature toward the encoding vector, and that invariant nonlinear-

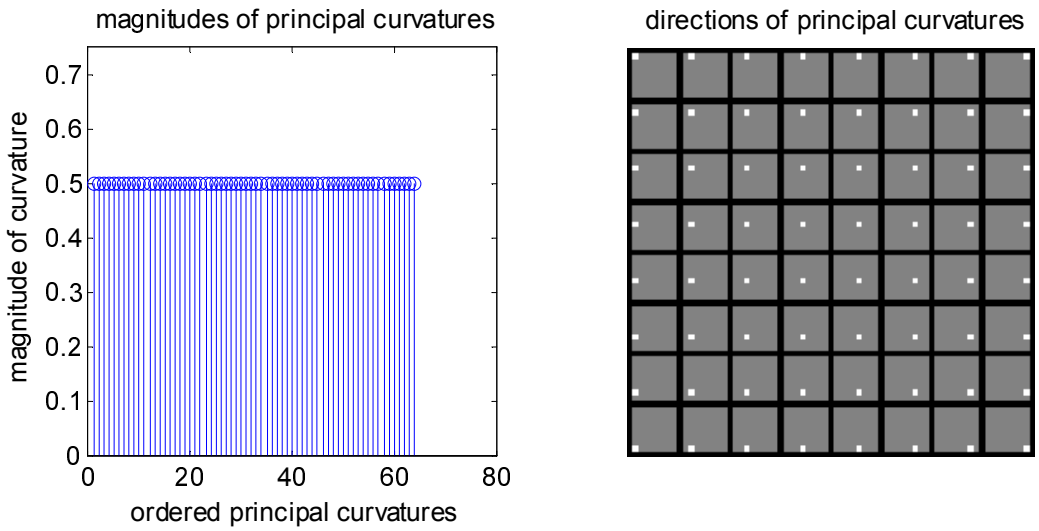

Figure A.11: The principal curvatures and principal directions of the hypersphere with R = 2 in 64D state space. Note the curvature magnitudes are all equally $1/R = 0.5$, and the principal directions can be represented as 8x8-pixel images in the state space, here corresponding to the coordinate basis.

ities can all be described by curvature away from the encoding vector. With the mathematical framework for describing curvature in high dimensions now in place, we can formulate and test specific hypotheses about the high-dimensional geometry of responses. The first is that selective nonlinearities are manifestations of negative principal curvature (toward the encoding vector), and that invariant nonlinearities come from positive curvature (away from the encoding vector). In high dimensions, a manifold can simultaneously have positive and negative curvature in different principal directions. The curvature algorithm that has been described will allow us to measure this directly. The original sparse coding network can only operate between a linear representation and an increasingly selective representation, so it ought to show only negative curvature in its responses. The Karklin

& Lewicki network, however, shows selectivity as well as invariance not found in the Olshausen & Field network, which will be manifested by simultaneous positive and negative curvature.

The real benefit of measuring curvature in high dimensions is that it will provide a more quantitative measure of selectivity and invariance. Typically, these are measured by the fraction of images a neuron responds to given some transformation (Rust and DiCarlo, 2012). Selectivity for conjunctions of features was measured by the fraction of instances a neuron fired above a threshold to an image of a particular object as well as a texture-scrambled version of the same image. Invariance to features was measured by the fraction of instances a neuron fired above a threshold to an object image as well as the same image with an altered position, scale or background image. These measures certainly capture the notions of selectivity and invariance, and they work for real neurons. The proposed curvature measurement for simulated neurons, however, would reveal a set of the features to which the neuron is selective and invariant by the principal directions, as well as how sensitive they are to those features in terms of the magnitude of the principal curvature. The curvature of the Karklin & Lewicki network will reveal that feature selectivity and invariance is determined by the direction of the particular basis functions in the first layer and how they are combined by the second layer bases. The features to which visual neurons in these networks (and in physiology) are selective and invariant can be found based primarily on the directions of the basis functions in state space, which are in turn learned from samples from the distribution of natural images. The power of this method is that it provides direct measurements about the image state space in a way that is obscured by previous methods.

APPENDIX B

ADDITIONAL FIGURES

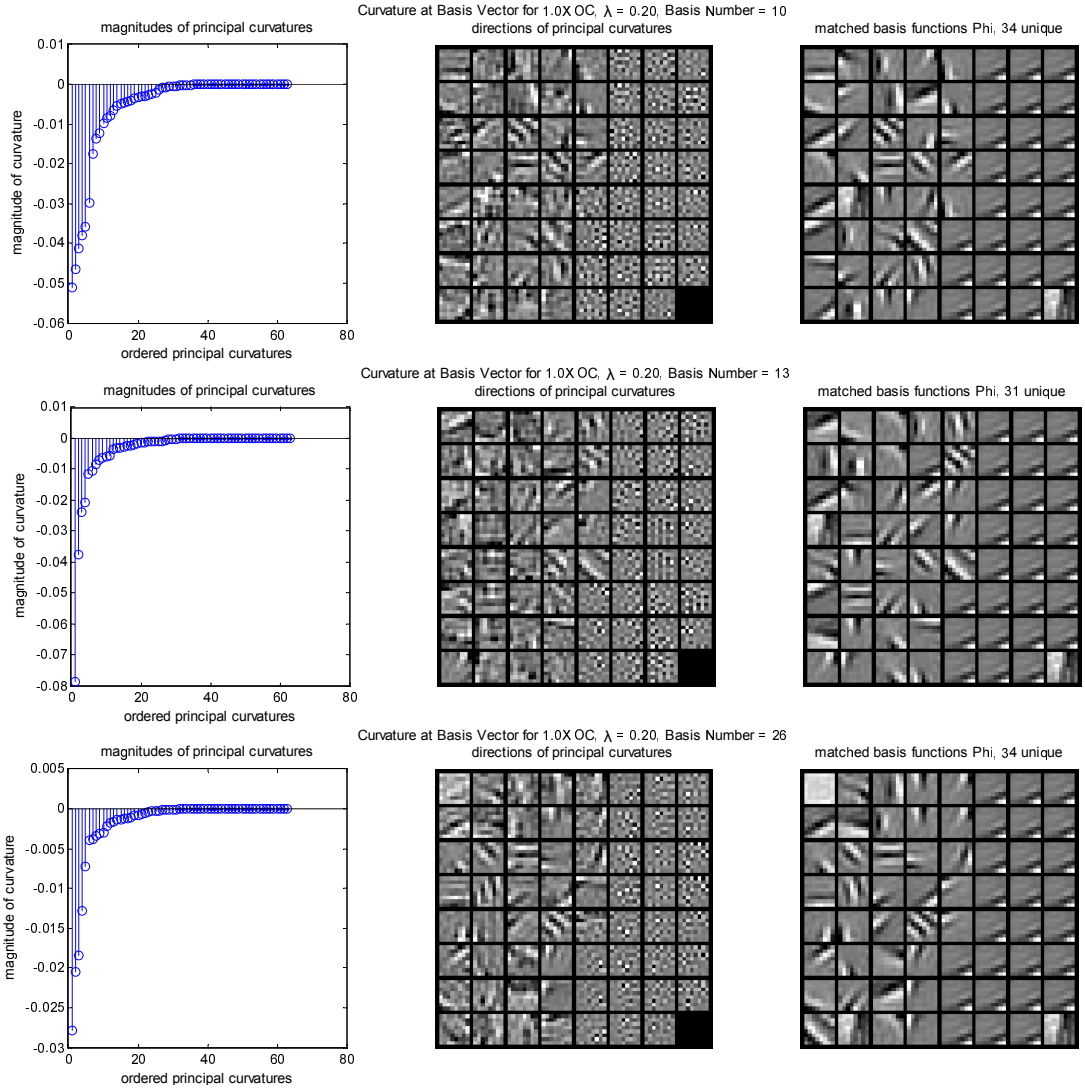

Figure B.1: Examples of isosurface curvature for OC = 1X (critically sampled) and $\lambda = 0.2$. Note that all are less than zero.

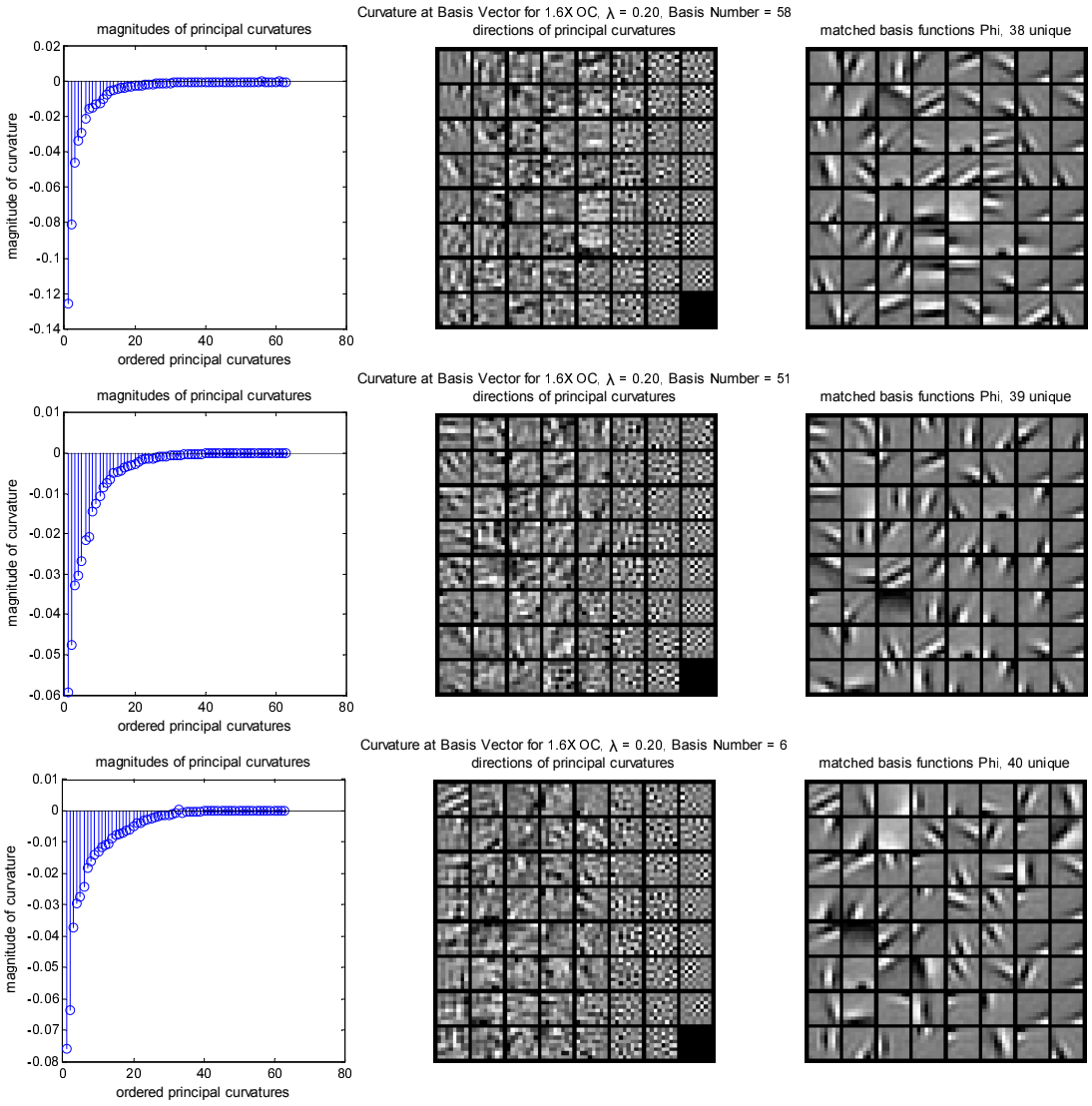

Figure B.2: Examples of isosurface curvature for OC = 1.6X (critically sampled) and $\lambda = 0.2$. Note that all are less than zero.

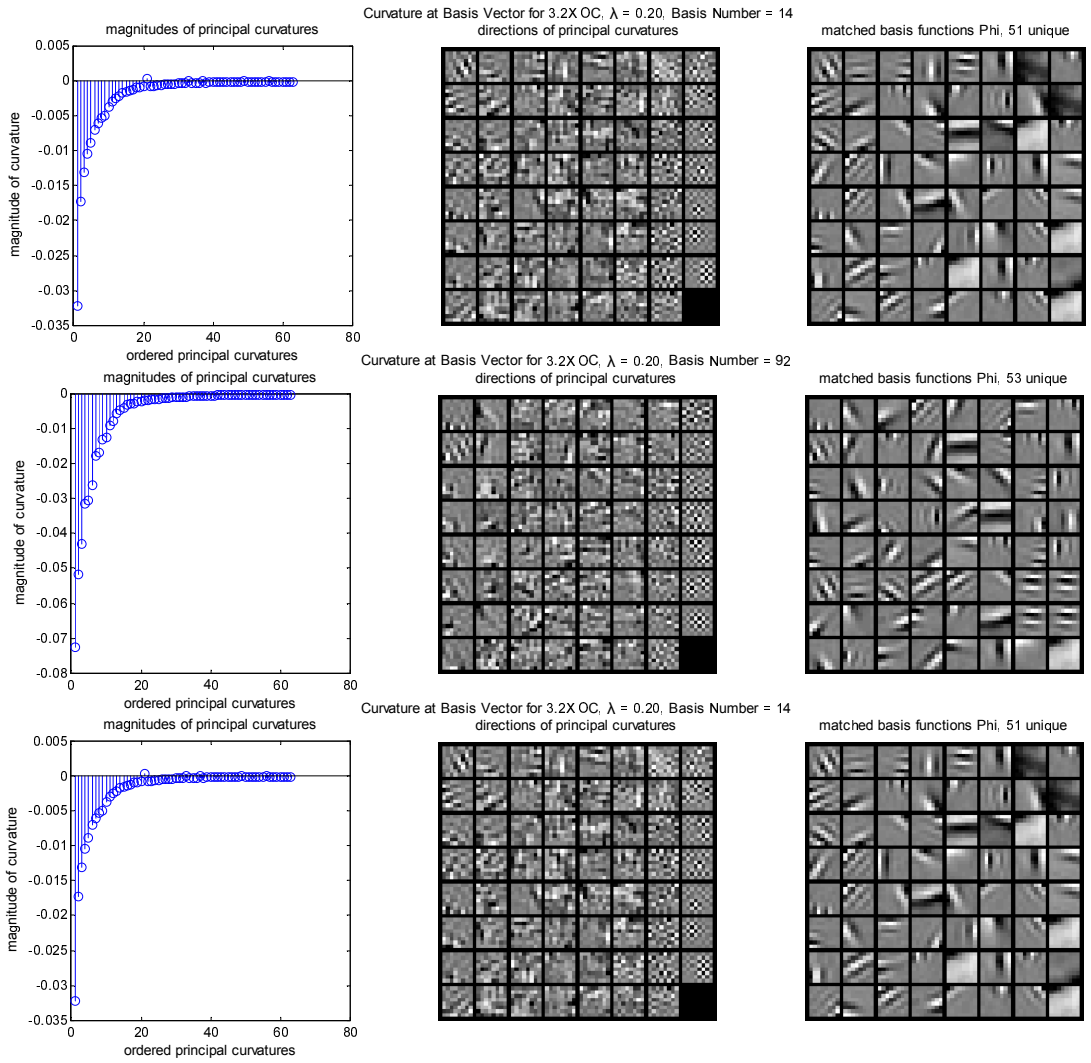

Figure B.3: Examples of isosurface curvature for OC = 3.2X (critically sampled) and $\lambda = 0.2$. Note that all are less than zero.

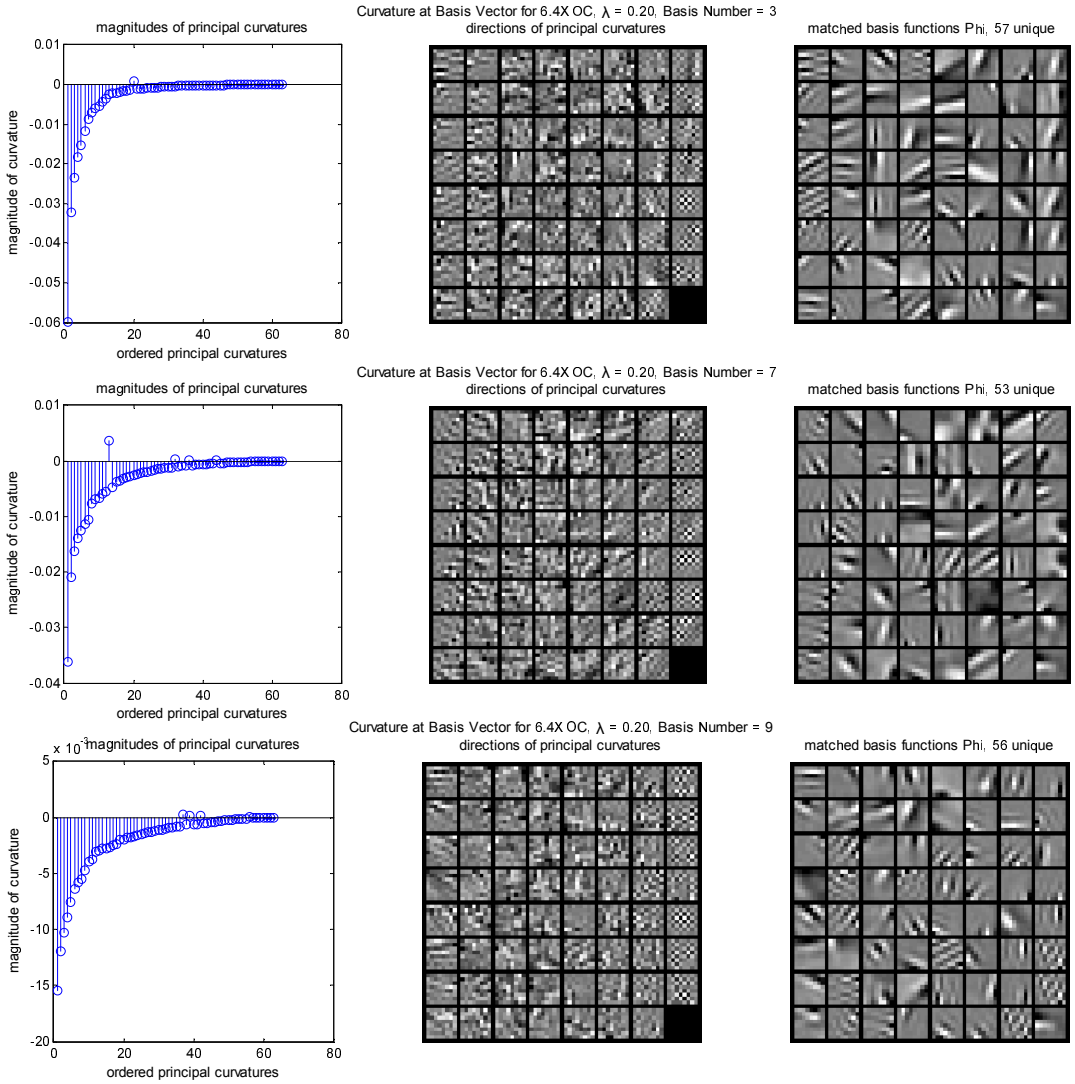

Figure B.4: Examples of isosurface curvature for OC = 6.4X (critically sampled) and $\lambda = 0.2$. Note that all are less than zero.

REFERENCES

Adelson, E. H. and Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *JOSA A*, 2(2):284–299.

Adrian, E. D. and Matthews, R. (1928). The action of light on the eye. *The Journal of physiology*, 65(3):273–298.

Aigner, M., Ziegler, G. M., Hofmann, K. H., and Erdos, P. (2010). *Proofs from the Book*, volume 274. Springer.

Albrecht, D. G., Geisler, W. S., and Crane, A. M. (2003). Nonlinear properties of visual cortex neurons: Temporal dynamics, stimulus selectivity, neural performance. *The visual neurosciences*, 1:747–764.

Attneave, F. (1954). Some informational aspects of visual perception. *Psychological review*, 61(3):183.

Barlow, H. B. (1972). Single units and sensation: a neuron doctrine for perceptual psychology? *Perception*, (38):795–8.

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

Berkes, P. and Wiskott, L. (2006). On the analysis and interpretation of inhomogeneous quadratic forms as receptive fields. *Neural computation*, 18(8):1868–1895.

Bian, B., Guan, P., Ma, X.-N., Xu, L., et al. (2011). A constant rank theorem for quasiconcave solutions of fully nonlinear partial differential equations. *Indiana Univ. Math. J*, 60(1):101–120.

Blakemore, C. t. and Campbell, F. (1969). On the existence of neurones in the human visual system selectively sensitive to the orientation and size of retinal images. *The Journal of physiology*, 203(1):237–260.

Brodie, S. E., Knight, B. W., and Ratliff, F. (1978). The response of the limulus retina to moving stimuli: a prediction by fourier synthesis. *The Journal of general physiology*, 72(2):129–166.

Cadieu, C. F. and Olshausen, B. A. (2012). Learning intermediate-level representations of form and motion from natural movies. *Neural computation*, 24(4):827–866.

Campbell, F. W. and Robson, J. (1968). Application of fourier analysis to the visibility of gratings. *The Journal of physiology*, 197(3):551–566.

Carlson, E. T., Rasquinha, R. J., Zhang, K., and Connor, C. E. (2011). A sparse object coding scheme in area v4. *Current Biology*, 21(4):288–293.

Chang, S.-Y. A., Ma, X.-N., and Yang, P. (2010). Principal curvature estimates for the convex level sets of semilinear elliptic equations. *Discrete Contin. Dyn. Syst*, 28(3).

Churchland, P. (2005). Chimerical colors: some phenomenological predictions from cognitive neuroscience. *Philosophical psychology*, 18(5):527–560.

DiCarlo, J. J. and Cox, D. D. (2007). Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341.

Edelman, S. (1999). *Representation and recognition in vision*. MIT press.

Edelman, S. (2009). On what it means to see, and what we can do about it. *Object Categorization: Computer and Human Vision Perspectives*, pages 69–86.

Egan, G. (2005). Foundations: General relativity and black holes.

Erhan, D., Courville, A., and Bengio, Y. (2010). Understanding representations learned in deep architectures. *Dept. Inf. Res. Oper., Univ. Montréal, Montréal, QC, Canada, Tech. Rep*, 1355.

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *JOSA A*, 4(12):2379–2394.

Field, D. J. (1994). What is the goal of sensory coding? *Neural Computation*, 6(4):559–601.

Field, D. J. and Wu, M. (2004). An attempt towards a unified account of non-linearities in visual neurons. *Journal of vision*, 4(8):283–283.

Fitzgerald, J. D., Rowekamp, R. J., Sincich, L. C., and Sharpee, T. O. (2011). Second order dimensionality reduction using minimum and maximum mutual information models. *PLoS computational biology*, 7(10):e1002249.

Freeman, J., Ziemba, C. M., Heeger, D. J., Simoncelli, E. P., and Movshon, J. A. (2013). A functional and perceptual signature of the second visual area in primates. *Nature neuroscience*, 16(7):974–981.

Gabor, D. (1946). Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93(26):429–441.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE.

Golden, J. R., Vilankar, K. P., Wu, M. C., and Field, D. J. (2015). Conjectures regarding the nonlinear geometry of visual neurons. *Vision Research, Special Issue: Natural Scenes*.

Goodfellow, I., Lee, H., Le, Q. V., Saxe, A., and Ng, A. Y. (2009). Measuring invariances in deep networks. In *Advances in neural information processing systems*, pages 646–654.

Hartline, H. K. (1928). A quantitative and descriptive study of the electric response to illumination of the arthropod eye. *American Journal of Physiology–Legacy Content*, 83(2):466–483.

Hartline, H. K. (1930). The dark adaptation of the eye of limulus, as manifested by its electric response to illumination. *The Journal of general physiology*, 13(3):379–386.

Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154.

Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243.

Hyvärinen, A., Hurri, J., and Hoyer, P. O. (2009). *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.*, volume 39. Springer Science & Business Media.

Karklin, Y. (2007). *Hierarchical Statistical Models in the Visual Cortex*. PhD thesis, Carnegie Mellon University.

Karklin, Y. and Lewicki, M. S. (2003). Learning higher-order structures in natural images. *Network: Computation in Neural Systems*, 14(3):483–499.

Karklin, Y. and Lewicki, M. S. (2005). A hierarchical bayesian model for learning nonlinear statistical regularities in nonstationary natural signals. *Neural computation*, 17(2):397–423.

Karklin, Y. and Lewicki, M. S. (2009). Emergence of complex cell properties by learning to generalize in natural scenes. *Nature*, 457(7225):83–86.

Konorsky, J. (1967). Integrative activity of the brain.

Körding, K. P., Kayser, C., and König, P. (2003). On the choice of a sparse prior. *Reviews in the Neurosciences*, 14(1-2):53–62.

Köster, U. and Olshausen, B. (2013). Testing our conceptual understanding of v1 function. *arXiv preprint arXiv:1311.0778*.

Köster, U., Sohl-Dickstein, J., Gray, C. M., and Olshausen, B. A. (2014). Modeling higher-order correlations within cortical microcolumns. *PLoS computational biology*, 10(7):e1003684.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kuffler, S. W. (1953). Discharge patterns and functional organization of mammalian retina. *Journal of neurophysiology*, 16(1):37–68.

Le, Q. V. and Ng, A. (2013). Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE.

Lee, J. M. (1997). *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media.

Lettvin, J. Y., Maturana, H. R., McCulloch, W. S., and Pitts, W. H. (1959). What the frog's eye tells the frog's brain. *Proceedings of the IRE*, 47(11):1940–1951.

Lewicki, M. and Sejnowski, T. (2000). Learning overcomplete representations. *Neural computation*, 12(2):337–365.

Lewicki, M. S., Olshausen, B. A., Surlykke, A., and Moss, C. F. (2014). Scene analysis in the natural environment. *Frontiers in psychology*, 5.

Marčelja, S. (1980). Mathematical description of the responses of simple cortical cells*. *JOSA*, 70(11):1297–1300.

Marr, D. (1982). Visoin: A computational investigation into the human representation and processing of visual information. *WH San Francisco: Freeman and Company*.

Minsky, M. L. and Papert, S. A. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT press Boston, MA.

Movshon, J., Thompson, I., and Tolhurst, D. (1978). Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *The Journal of physiology*, 283(1):53–77.

Olshausen, B. (2008). Sparse coding and ica.

Olshausen, B. and Lewicki, M. S. (2014). What natural scene statistics can tell us about cortical representation.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325.

Olshausen, B. A. and Field, D. J. (2005). How close are we to understanding v1? *Neural computation*, 17(8):1665–1699.

Oppenheim, J. N. and Magnasco, M. O. (2013). Human time-frequency acuity beats the fourier uncertainty principle. *Physical review letters*, 110(4):044301.

Pagan, M., Urban, L. S., Wohl, M. P., and Rust, N. C. (2013). Signals in inferotemporal and perirhinal cortex suggest an untangling of visual target information. *Nature neuroscience*, 16(8):1132–1139.

Pentland, A. P. (1984). Fractal-based description of natural scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):661–674.

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E., and Simoncelli, E. P. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999.

Pressley, A. N. (2010). *Elementary differential geometry*. Springer Science & Business Media.

Priebe, N. J. and Ferster, D. (2006). Mechanisms underlying cross-orientation suppression in cat visual cortex. *Nature neuroscience*, 9(4):552–561.

Rao, R. P. and Ruderman, D. L. (1999). Learning lie groups for invariant visual perception. *Advances in neural information processing systems*, pages 810–816.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5.

Rust, N. C. and DiCarlo, J. J. (2012). Balanced increases in selectivity and tolerance produce constant sparseness along the ventral visual stream. *The Journal of Neuroscience*, 32(30):10170–10182.

Schwartz, O. and Simoncelli, E. P. (2001). Natural signal statistics and sensory gain control. *Nature neuroscience*, 4(8):819–825.

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

Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21.

Sherrington, C. (1941). Man on his nature. *The Journal of Nervous and Mental Disease*, 94(6):762–763.

Simoncelli, E. P. and Olshausen, B. A. (2001). Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216.

Tsai, C.-Y. and Cox, D. D. (2015). Measuring and understanding sensory representations within deep networks using a numerical optimization framework. *arXiv preprint arXiv:1502.04972*.

Vinje, W. E. and Gallant, J. L. (2000). Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, 287(5456):1273–1276.

Weisstein, E. W. (2001a). Curvature. From MathWorld—A Wolfram Web Resource.

Weisstein, E. W. (2001b). One-sheeted hyperboloid. From MathWorld—A Wolfram Web Resource.

Wiskott, L. and Sejnowski, T. J. (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624.

Yazdanbakhsh, A. and Livingstone, M. S. (2006). End stopping in v1 is sensitive to contrast. *Nature neuroscience*, 9(5):697–702.

Zetzsche, C. and Krieger, G. (1999). Nonlinear neurons and higher-order statistics: new approaches to biological vision and digital image processing. In *Human vision and electronic imaging iv*, pages 2–33.

Zhu, M. and Rozell, C. J. (2013). Visual nonclassical receptive field effects emerge from sparse coding in a dynamical system. *PLoS computational biology*, 9(8):e1003191.