NEURAL NETWORK MODELS FOR NEURAL ENCODING IN VISION AND DECODING BRAIN STATE

by

ELIJAH DOUGLAS CHRISTENSEN

BS, University of Washington, 2011

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Neuroscience Program

2020

This thesis for the Doctor of Philosophy degree by

Elijah Douglas Christensen

has been approved for the

Neuroscience Program

by

Gidon Felsen, Chair

Joel Zylberberg, Advisor

Zachary Kilpatrick

Alon Poleg-Polsky

Angie Ribera

Date: 14-August 2020

Christensen, Elijah Douglas (BS, Neuroscience Program)

Computational Modeling of Neural Encoding in Vision and Neurostimulation

Thesis directed by Assistant Professor Joel Zylberberg

ABSTRACT

There is a significant need for more effective treatments for neurological and psychiatric disease. Implantable neurostimulators are increasingly used as new therapeutic options for these diseases. This work will discuss our approach to mitigate current limitations in two types of implantable neurostimulators: Deep brain stimulation in treating Parkinson's disease and cortical prosthetics in vision.

Deep brain stimulators (DBS) are typically configured to deliver therapeutic stimulation constantly, which can produce unavoidable side-effects and needlessly drains power. Modulating stimulation adaptively, or close-loop stimulation, could mitigate these issues but requires methods to accurately readout physiologically relevant brain states, preferably using only the already implanted electrodes. Another class of implanted neurostimulators, cortical prosthetics, rely on accurate predictions of neural activity in the targeted brain area for arbitrary stimuli. Current models used to predict neural activity in primary visual cortex only achieve 35% predictability overall and predictability declines in subsequent areas of visual processing.

With a focus on DBS and cortical prosthetics, we highlight how deep artificial neural network (ANN) models can be leveraged as a tool in neuroscience for studying neural encoding and decoding. We apply an ANN model trained using supervised learning to decode sleep state continuously from "spectral fingerprints" contained in

local field potential activity of DBS electrodes. Furthermore, we show deep convolutional neural networks can be used to make more accurate predictions of cortical neural encoding of visual stimuli in both early (primary visual cortex) and late (inferior temporal cortex) stages of visual processing.

The form and content of this abstract are approved. I recommend its publication.

Approved: Joel Zylberberg

TABLE OF CONTENTS

CHAPTER

CHAPTER I

INTRODUCTION

Neurological and psychiatric disease represent a significant societal burden in both advanced and developing countries (Collins et al., 2011) and there is a significant need for more effective treatments. Recent advances in brain stimulation and recording technology have enabled development of long-desired treatment options for many of these diseases in the form of implantable devices that directly stimulate populations of neurons. Deep brain stimulation (DBS) is one of these implantable devices utilized to mitigate disease symptoms. Patients with DBS receive electrical pulses via electrodes implanted in their brain. DBS has become an established therapy for movement disorders (Parkinson's Disease (PD) and essential tremor) (Perlmutter and Mink, 2006) as well as epilepsy and psychiatric diseases (Holtzheimer and Mayberg, 2011). Another group of implantable neurostimulators are neural interfaces, such as cortical prostheses, which aim to restore sight in patients with congenital or acquired blindness. The body of work presented here makes progress on two unsolved challenges limiting advances in implantable neurostimulators, namely DBS state detection and more accurate cortical encoding of visual stimuli.

Deep Brain Stimulation

DBS uses a surgically implanted stimulator to apply electrical pulses directly to the brain to mitigate symptoms of neurologic and psychiatric diseases. Historically, drugs have been the primary method of treating these diseases, but DBS has emerged as a promising alternative for patients who do not respond to pharmacotherapy.

Parkinson's disease (PD) was among the first FDA approved uses of DBS for mitigating the disease's motor symptoms. When employed for treating PD, current best practice for DBS therapy uses constant stimulation even though its therapeutic benefits to motor symptoms are needed most when the patient is awake to suppress resting tremor or bradykinesia in movement initiation. Current implanted stimulators are used this way because they have no way to detect when stimulation is not needed, such as when the patient is asleep or when lower levels of stimulation are needed to correct resting tremor. This strategy of constant stimulation, or open-loop stimulation, is less power efficient and comes with side effects such as impaired cognition, speech, gait, and balance (Hariz et al., 2008). However, activating DBS stimulation only when necessary requires a robust method for discerning whether or not the patient's brain needs stimulation. For example, a closed-loop DBS system would read out the patient's brain state and only deliver electrical pulses during periods when the patient is awake. Closed-loop DBS is more power efficient and would have less collateral side effects by only stimulating when necessary.

Neural Interfaces

Cortical prosthetics (Fig 2) are a form of neural interface used to restore sight in blind patients (Lorach et al., 2013). These implantable neurostimulators bypass lost or damaged neurons by stimulating the damaged neuron targets the same way the original neurons otherwise would. Cortical prostheses must reproduce the neural activity patterns that would typically be relayed naturally by neurons of the thalamic lateral geniculate nucleus (LGN) and retina when directly stimulating visual cortex. Neural

encoding, our understanding of how neurons reformat and represent visual stimuli, is key to this goal of properly restoring sight. The ultimate test of our knowledge of neural encoding is to predict neural responses to stimuli. Unfortunately, current models of neural encoding still struggle to accurately predict neuron responses to natural image stimuli. Subsequent sections will review two distinct approaches to developing models capable of predicting cortical responses to visual stimuli: bottom-up encoding models and top-down encoding models.

Bottom-up neural encoding models

Bottom-up encoding models Neurons in early stages of visual processing can be characterized by their response to very specific local features in an image. A visual processing neuron's receptive field (RF) is useful for depicting the properties of an image that modulate the neurons activity. RF's are typically represented in models by linear filters applied at the first stage of processing. The inner product (i.e. dot product) between the filter and corresponding image region predict a given neurons response to that image. The linear RF model was insufficient for predicting several non-linear properties of retinal ganglion cell (RGC) responses to white noise and even worse for more complex stimuli. Subsequent Linear-Nonlinear-Poisson (LNP) (Paninski et al., 2004) models were better predictors of RGC spike rates in responses to white noise image stimuli. LNP combines a linear spatial filter with a single static non-linearity. The LNP model predicts neural responses well for white noise stimuli but does not generalize well when used to predict responses to natural image stimuli. Generalized

Linear Model's (GLM) (Pillow et al., 2008) improve prediction accuracy by accounting for interactions between RGC's.

Top-down neural encoding models

As opposed to bottom-up neural encoding models, top-down models attempt to explain neural encoding as a result of optimizing an overarching goal. The genome likely has insufficient capacity for specifying every neuronal connection (synapse) (Zador, 2019) so what mechanisms ensure that neurons are connected correctly? This has recently been referred to as the "brain wiring problem" (Hassan and Hiesinger, 2015). We'll be looking specifically at how synaptic wiring is determined in the visual cortex. Before the eyes even open, molecular interactions and spontaneous activity of RGC's guide development of the initial "coarse" connectivity between RGC's in the eye, to the neurons of the lateral geniculate nucleus in thalamus (LGN) and on to the primary visual cortex (V1) (Del Rio and Feller, 2006; Katz and Shatz, 1996). After this retinotopic map is established, synaptic connectivity continues refinement but requires environmental stimuli (Pietro Berkes et al., 2011). Identifying this "unifying principle" that guides stimuli-dependent refinement of connectivity would help explain the structure of visual representations in V1 and beyond.

Sparse coding

Shortly after the discovery of simple and complex cells (Hubel and Wiesel, 1959), Horace Barlow proposed efficient coding (Barlow, 1961) as an explanation for the computations performed by neural circuits in sensory cortex. The efficient coding hypothesis posits that the overarching goal of sensory processing is to reduce the high

information redundancy in stimuli from the physical environment. This view was strengthened by findings that the Gabor-like receptive fields of simple cells are an optimal basis set for natural scenes when optimizing for 1) representation sparsity and 2) image reconstruction (D. Field, 1987; Olshausen and D. J. Field, 1996). Due to the highly metabolic nature of neurons, sparse coding was proposed because of its metabolic and information efficient properties (Levy and Baxter, 1996). Sparse coding models were particularly influential after successfully predicting aspects of neural computations in retina (Atick and Redlich, 1992), thalamus (Dan et al., 1996) and V1 (Olshausen and D. J. Field, 1996).

Optimizing for efficient coding would predict information redundancy should decrease as it is processed and relayed by successive visual areas. Information redundancy decreases when the same information can be carried by fewer neurons, which occurs as visual information propagates from photoreceptors to RGCs and from retinal ganglion cells to the LGN in the thalamus (Figure 1). Instead of information redundancy decreasing, as would be predicted by efficient coding, anatomical evidence seems to indicate that information redundancy in primary visual cortex is likely higher than it is prior areas of visual processing (Barlow, 2001; Felleman and Van Essen, 1991). Furthermore, despite some modest successes at explaining the complex response properties of V2 (the next visual area after V1) (Lee et al., 2008; Olshausen et al., 2001) subsequent findings (Pietro Berkes et al., 2009; Willmore et al., 2011) have shown that visual areas beyond V2 cannot be explained by the efficient coding hypothesis. Efficient coding alone as an objective is not sufficient for explaining

response properties of neurons in higher level visual areas like V4 and inferior temporal cortex (IT).

Goal-directed convolutional neural networks

Barlow, when reflecting later on his original idea makes a prescient statement, perhaps without knowing it: “We now need to step back and take a more global view of the brain’s **task** in order to see what lies behind the importance of recognizing redundancy” (Barlow, 2001). Neural networks which optimize behaviorally relevant tasks (Yamins et al., 2014) have shown state of the art performance at predicting neuronal activity across the ventral visual stream.

Summary

Chapter 2 provides an introduction to artificial neural networks (ANN), their similarities and differences to biological neurons, and the machine learning techniques used to train them which serves as a foundation for the technical chapters that follow.

In Chapter 3 we demonstrate ANN models as a tool for decoding sleep state in real-time using only the signals available from intracranial DBS electrodes implanted in the basal ganglia of PD patients. Importantly, this model generalizes decoding to patients never seen by the model and may allow new ways to leverage implantable stimulators for therapeutic benefit.

Chapter 4 explores the effects multi-functional objectives (recognize and visualize) on both learned representations and task performance. This work was motivated by the observation that visual processing areas are reactivated during

visualization tasks indicating their dual role in visual processing and regenerating stimuli.

Finally, Chapter 5 demonstrates the utility of neural network models and machine learning techniques as a way to explain response properties of individual neurons. We use a convolutional neural network (CNN) to achieve performance comparable to state of the art at predicting activity of individual neurons evoked by natural image stimuli in macaque V1. Furthermore, we use this model generatively to explain response properties of cells outside of Hubel and Wiesel's simple- or complex-cell designations.

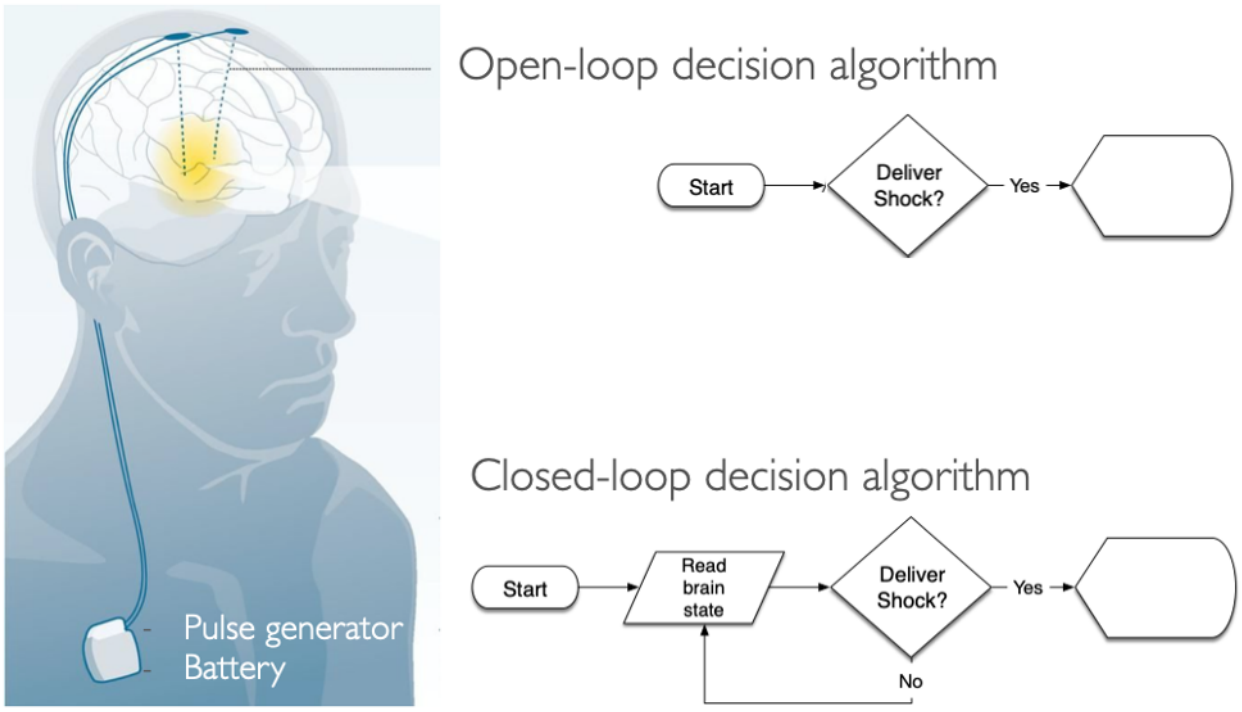

Figure 1.1 Closed-loop DBS system would read out the patient's brain state to modulate stimulation intensity accordingly based on if the patient is awake, stationary, or moving to relieve symptoms of Parkinsons Disease

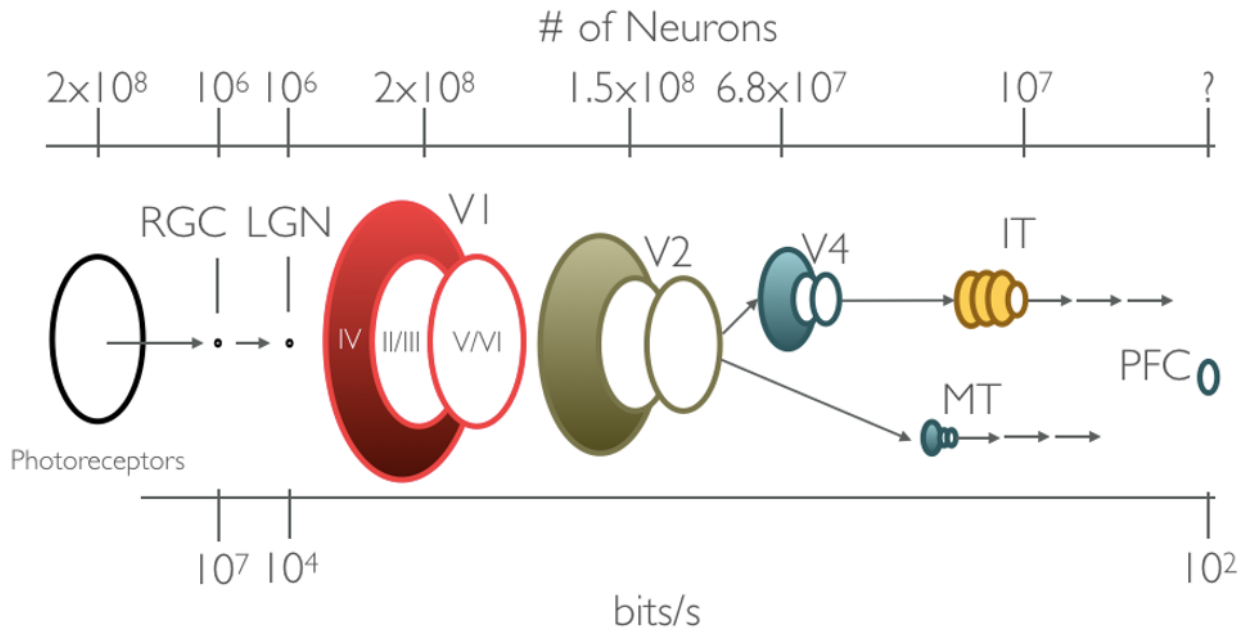

Figure 1.3 Channel capacity, the number of neurons carrying visual information, significantly varies along the ventral visual stream

CHAPTER II

MACHINE LEARNING AND COMPUTATIONAL NEUROSCIENCE

Artificial Neural Networks as a Model of Neural Computation

Despite the importance of computers for conducting machine learning and computational neuroscience research both fields had origins long before contemporary transistor computers. In 1943, inspired by the "all-or-none" nature of neural activity, Warren McCulloch (neuroscientist) and Walter Pitts (logician) formalized a simple mathematical definition of a neuron (McCulloch and Pitts, 1943). McCulloch and Pitts neurons became the fundamental unit of artificial neural networks (ANN). These artificial neurons, often referred to as (artificial) units, reproduce several key properties of real neurons (Fig 2.1). Biological neurons receive input from many other neurons via connections (e.g. synapses) to its dendrites. These synaptic inputs are summated at the soma where the net dendritic input increases or decreases the neurons membrane potential (Fig 2.1A). If the net dendritic input shifts the membrane potential beyond a certain threshold (e.g. the threshold potential) the neuron will fire action potentials.

Artificial Units

Artificial units (Fig 2.1B) are the basic building block of artificial neural networks. Each artificial unit receives input represented as a sequence of inputs x_i and each input has a corresponding synaptic weight w_i. In the artificial unit z loosely represents a neurons membrane potential by adding net dendritic input to a scalar bias term (b) which is meant to represent the unit's intrinsic excitability. Finally, the threshold non-linear response of biological neurons is captured by passing z through an activation

function (g) which gives the unit activation a which is meant to loosely analogize neuronal firing rate.

$$a = g(\mathrm{z}) = g\left(\sum_i (w_i \times \mathrm{x}_i) + b\right)$$

Layers

Just as the brain is comprised of more than one neuron, most models make use of many artificial units. Similar to the functional organization of the neocortex, artificial neural networks (ANN) group individual units together in groups typically referred to as layers (Fig 2.1C). The artificial units within a layer collectively operate on a shared input and the layer's output consists the collective activations of its constituent artificial units. ANN layers in a model between the inputs (x) and final outputs (y) are often referred to as "hidden" layers. The layers of an ANN are often considered analogous to a population of neurons in regions of the brain which perform similar functions. For instance, primary visual cortex (V1) contain a population of neurons which receive visual inputs from the retina (relayed by LGN). As a population of neurons, V1 processes this visual input and this processed visual information is then relayed to area V2 for subsequent processing and so on.

Model archetypes

Deep artificial neural network models typically have multiple layers stacked one after the other, such that the outputs of one layer become the inputs for the subsequent layer. Deep ANN models are often constructed for a specific purpose, or to perform a specific task. Models are often categorized based on purported task and the structure of the inputs it uses to accomplish this task. For instance, many computer vision

researchers train models which, given an image, categorize the object in the image. The work presented in this thesis makes use of three distinct types of neural network models: classifiers, regressors and autoencoders. We will cover these model archetypes briefly in the following sections.

Classifiers

Classifiers are a class of models that attempt to predict the best category that describes the input from a discrete number of categories. For example, a classic machine learning exercise has been to train a model to predict the category of an object depicted in an image. MNIST, Fashion-MNIST(Xiao et al., 2017), CIFAR10/100(Krizhevsky and Hinton, 2009; Krizhevsky et al., n.d.) and ImageNet are examples of large labeled image datasets that have been historically popular for evaluating a model's classification performance. Classifiers are not specific image tasks and can be used on any discrete labeling task. For instance, in Chapter 3 we trained an ANN classifier to predict behavioral sleep state in human PD patients based on features from local field potential spectral decompositions.

Regressors

Regressors use their inputs and attempt to predict a continuous value purportedly derived from the input. Recently, neural network models have been used as functional models of the visual system. These models use images to predict neuronal firing rates observed in animals after viewing the same image and they have been used to successfully for predicting stimulus evoked activity in retina (McIntosh et al., 2016) and Inferior Temporal cortex (IT) (Yamins et al., 2014). We successfully utilized a

convolutional neural network regressor model to predict firing responses for populations of neurons in macaque primary visual cortex (V1) which is the subject of Chapter 5.

Autoencoders

Autoencoders are a special class of models which attempt to predict their inputs. This is a trivial task if each of the intermediate hidden layers have similar dimensionality as the input and output; the model can simply learn to copy the input into the output. Instead, these models are more often configured to have far fewer dimensions in their hidden layers. In this configuration the only way to successfully perform the task is to exploit information redundancy in the input to compress the input while retaining as much information as possible. We use an autoencoder in Chapter 4 to better capture the fact that the brain uses its representations for both recognition but also generatively in visualization.

Architectures

Training an ANN model using machine learning typically requires three components. These components are 1) the model's layer architecture, 2) objective or loss function, and 3) the models learning rules. The layer architecture of a model explicitly specifies how the artificial units, organized in layers, are connected from input to output. There are a wide variety of layer types to choose from when constructing a deep ANN but for the sake of brevity only descriptions of layer architectures used in this work will be provided.

Fully connected

Fully-connected layers are the simplest and oldest of layer architectures. In all-to-all layers, every input is connected to every unit in the layer. We can describe this ANN layer mathematically by vectorizing the previous equation wherein inputs and output firing rates are represented as vectors ($\boldsymbol{x}_i, \boldsymbol{a}_j$) instead of scalars ($x, a$):

$$\boldsymbol{a}_j = g(\boldsymbol{z}_j) = g\left(\boldsymbol{x}_i \cdot \boldsymbol{w}_{i,j} + \boldsymbol{b}_j\right)$$

Hyperbolic tangent (tanh), sigmoid, or Rectified Linear Units (ReLU) are often used as activation functions (g) but other more complex ones have also been introduced.

Convolutional

Convolutional layers have many parallels to (and were directly inspired by) the organization of the mammalian visual system. The early layers of visual processing are organized spatially, areas of field of view near each other are encoded near each other in RGCs, LGN, and V1 with nearly one-to-one correspondence. This matching topographic map is often referred to as retinotopic organization. Deeper layers than V1 retain some of this retinotopy but progressively pool these features representing larger and larger receptive field areas. Convolutional layers convolve a series of spatial filters across their 2D inputs to output "feature maps" of patches in the image that match the filter. They typically operate on images that have been separated into three distinct channels (RGB) and normalized as a surrogate for processing steps in retina and thalamus (Dan et al., 1996). However, this is not a defining characteristic, networks are also often trained on grayscale or color images that have not been preprocessed at all.

Loss Function

Loss or cost functions (J) are mathematical definitions of the goal of the learning system. The loss function is used to calculate a scalar metric quantifying the models' task performance as a function of their output. Loss functions can take any form mathematically, but typically differentiable loss functions are preferred for more straightforward optimization. Reconstruction error (sum of squared pixel errors) has traditionally been used for training models which attempt to generate a particular image. As an example of one loss function we can express the sum squared pixel loss between a model's output image ($\hat{y}$) and the target reference (y) as:

$$J(y, \hat{y}) = \sum (y - \hat{y})^2$$

The target reference (y) is sometimes referred to as the teaching signal. In supervised learning the teaching signal is supplied to train the network the right answer for each particular batch of training examples.

Loss functions do not have to depend on a particular dataset or task. For instance, sparse coding models use activation sparseness and reconstruction error as their loss function to learn sparse representations. When minimized over images of natural scenes they learn to represent images using features that resemble localized receptive fields of simple cells in the primary visual cortex (Olshausen and D. J. Field, 1996; Zylberberg et al., 2011)

Learning Rules

Once a model's architecture and loss function are specified "training it" is simply optimizing the parameters of each layer to improve its loss. The first algorithm for defining a method for iteratively updating the ANN model parameters to minimize loss was developed by Rumelhart (Rumelhart et al., 1986) and is still commonly used for training ANNs. We use this algorithm for training our models and it involves a simple 2 step process:

1) Forward pass: Use a batch of x input values to calculate the predicted outputs ($\hat{\boldsymbol{y}}$)
2) Backpropagation: Use prediction error to update weights and biases

To illustrate this process, we will derive it for a simple 2-layer ANN. For simplicity, we change notation when describing deep ANN with multiple layers such that variable and function subscripts denote the variable or function's corresponding layer NOT matrix or vector dimensions. For instance, we define the output activations of the l^{th} layer in a model comprised of sequentially stacked all-to-all layers as:

$$\boldsymbol{a}_l = g(\boldsymbol{a}_{l-1} \cdot \boldsymbol{W}_l + \boldsymbol{b}_l)$$

Forward Pass

First, we pass a batch of training example inputs (x) through the model to get a batch of outputs ($\hat{\boldsymbol{y}}$). Given our simple feedforward layer defined above, the full equation for the models output is given by:

$$\hat{\boldsymbol{y}} = g(\boldsymbol{W}_2 \cdot g(\boldsymbol{W}_1 \cdot \boldsymbol{x} + b_1) + b_2)$$

For simplicity, we will combine all trainable parameters in this model into a variable

$$\theta = \{\boldsymbol{W}_2, \boldsymbol{W}_1, b_2, b_1\}$$

Our loss function (J) defines how to evaluate the model's performance as a function of the model's predicted and target values. The target value is also sometimes referred to as the teaching signal, as it is used to teach the model the correct output for a given input. For this example, we'll use sum-squared-error:

$$J(y, \hat{y}) = \sum (y - \hat{y})^2$$

Backpropagation

To derive the gradient of the loss function with respect to the model parameters (∇_θJ) we take a partial derivative of the loss function with respect to the models parameters:

$$\nabla_\theta \mathrm{J} = \frac{\partial J(\theta; y, \hat{y})}{\partial \theta}$$

Optimizers

Once we know the gradient of each weight with respect to the loss, we simply need to adjust the weights of the model in the direction specified by the weight gradient. Continually descending the gradient of the loss function should result in reaching a minimum of the loss for performing the model's task but may not be the global minimum.

Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is the simplest and oldest optimization algorithm. Model parameters θ, are iteratively updated by subtracting the parameter gradient scaled by a learning rate η according to the following equation

$$\theta \coloneqq \theta - (\eta \cdot \nabla_\theta J)$$

SGD optimization with momentum

Standard SGD works well if the surface of the loss function is smooth but has difficulty navigating "ravines" (the1986, n.d.) which are common near local minima in optimization problems. Momentum (Qian, 1999) is a solution to this issue wherein a fraction γ of the previous update is added to the current weight update. This modification helps SGD accelerate in the relevant descent direction.

$$v_t = \gamma \cdot v_{t-1} + (\eta \cdot \nabla_\theta J)$$

$$\theta \coloneqq \theta - v_t$$

Picture a boulder accumulating speed as it rolls down a hill. Progressively increasing for dimensions gradient direction stays the same and reduces weight updates if the gradient direction changes.

ADAM optimization

Adaptive Moment Estimation (ADAM) computes adaptive learning rates for each parameter. If momentum is a ball rolling down a hill, ADAM optimization is a heavy ball with friction. It accomplishes this by computing decaying averages of past and past squared gradients (g_t, g_t^2).

$$\widehat{m}_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$\hat{v}_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\theta_{t+1} \coloneqq \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

Summary

The purpose of this chapter is not to exhaustively cover the field of machine learning but instead to serve as a brief primer of concepts and terms you will encounter in subsequent chapters. Chapter 2 uses an ANN classifier comprised of fully-connected layers to predict sleep states from LFP spectral decompositions. Chapter 3 utilizes a convolutional autoencoder/classifier hybrid model to test hypotheses about computational objectives employed in primate ventral stream visual representations. Finally, Chapter 4 uses a convolutional neural network (CNN) to directly regress neuronal activity in macaque primary visual cortex.

Hopefully, you can appreciate the similarities between artificial neural networks and the biological neural networks that inspired them. If nothing else, remember that using machine learning to train ANN models hinges on three components:

1) Model architecture

2) Loss function

3) Learning rules

All three components influence both transient and final model performance.

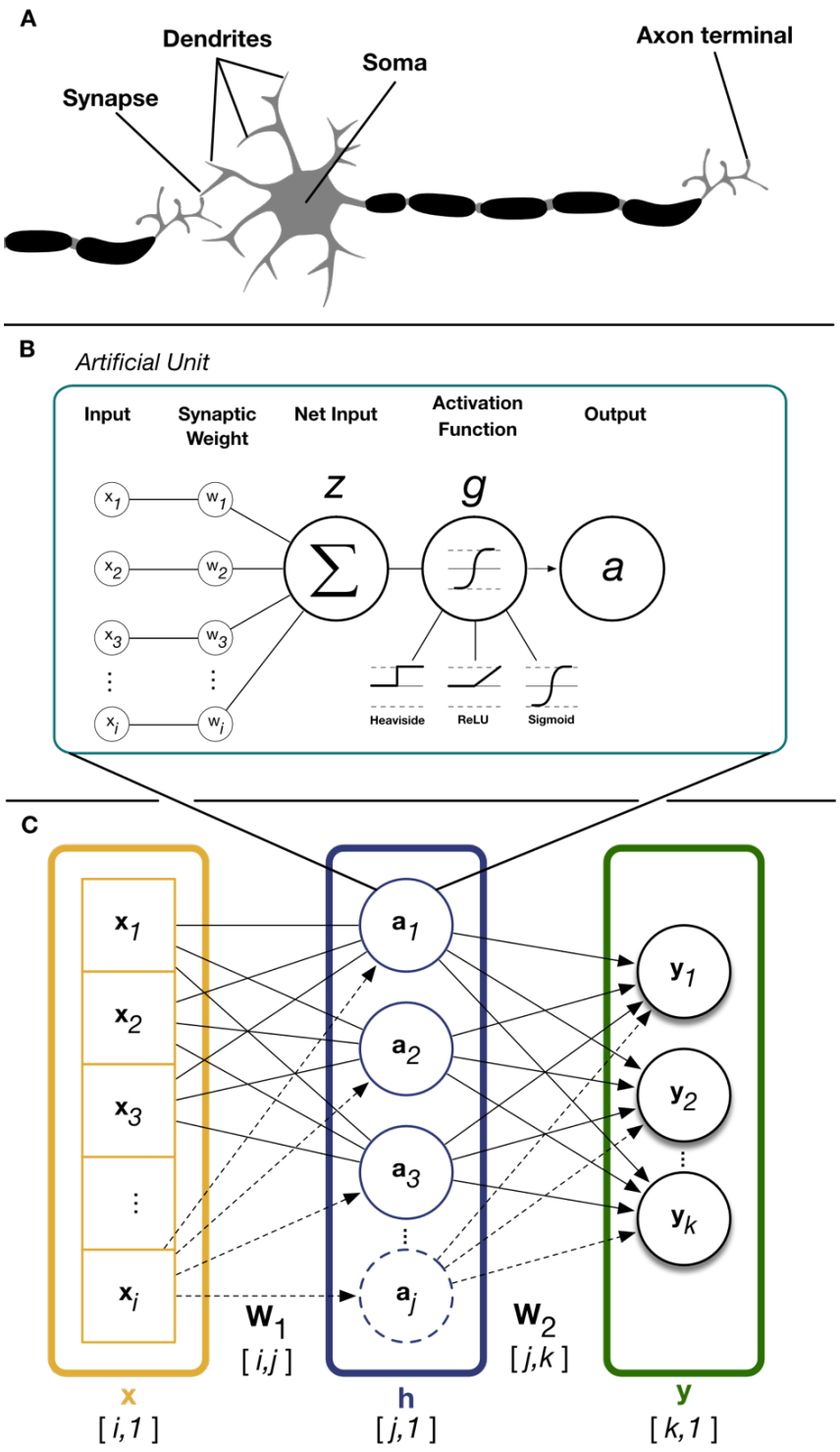

Figure 2.1 Synaptically connected neurons in the brain are the substrate of neural computation in brains

CHAPTER III

PREDICTING SLEEP STATES IN HUMAN PARKINSON'S DISEASE PATIENTS[1]

Introduction

Sleep is crucial to the regulation of physiological and cognitive functions in humans, and when disordered greatly diminishes quality of life (Giuditta et al., 1995; Pace-Schott and Hobson, 2002) and adversely affects nervous system repair (Brager et al., 2016; Lucke-Wold et al., 2015). Parkinson's disease (PD) is a neurodegenerative disorder that exhibits a high degree of comorbidity with a wide range of sleep disorders (De Cock et al., 2011; Tekriwal et al., 2017). The diagnosis and treatment of PD primarily focus on the overt motor symptoms (Postuma et al., 2015). However, there is increasing interest in understanding the impact of non-motor symptoms, such as sleep dysfunction, on overall disease burden (Chaudhuri et al., 2006), and in identifying treatments for these symptoms. With the onset of motor fluctuations or breakthrough tremor despite optimal medical management, subthalamic nucleus (STN) deep brain stimulation (DBS) surgery has become the reference standard for treating the motor symptoms of advanced PD (Bronstein et al., 2011; Hamani et al., 2004). Interestingly, several studies have found that STN-DBS can improve sleep in PD (Arnulf et al., 2000; De Cock et al., 2011; Iranzo et al., 2002). In our previous work, using local field potentials (LFPs) recorded from DBS electrodes implanted in STN for the treatment of PD, we identified unique spectral patterns within STN oscillatory activity that correlated

[1] This chapter was previously published in Christensen, Abosch, Thompson, and Zylberberg (2019) *Journal of Sleep Research* 28:e12806 and are included with the permission of the copyright holder.

with distinct sleep cycles, a finding that might offer insight into sleep dysregulation (Thompson et al., 2017). One extension of this work was to determine whether LFP information recorded from the STN could be used in real time to objectively identify sleep cycles for targeted therapy using DBS. In other words, the sleep benefit derived from STN stimulation could potentially be optimized using an adaptive stimulation algorithm that is aimed at specific sleep stages. In this study, we demonstrate the use of a feedforward artificial neural network that predicts sleep stage from LFP recordings, within the STN, with high precision.

Materials and Methods

Patient Demographics

This study was approved by the Institutional Review Board of the University of Minnesota, where the surgical and recording procedures were performed. All consenting study subjects (n = 9) carried a diagnosis of idiopathic PD (Figure 3.1a). Subjects were unilaterally implanted in the STN with a quadripolar DBS electrode (model #3389: Medtronic Inc., Fridley, MN), per routine surgical protocol (Abosch et al., 2012). Experimental details for the recording setup have been previously published (Thompson et al., 2017). Basic characterization of these data was previously reported in Thompson et al. (2017).

Signal processing and local field potentials

Signal processing of the raw STN LFP signals was previously described in Thompson et al. (2017). Briefly, after preprocessing, the four LFP channels (0, 1, 2 and

3; one recording from each of the four electrical contacts of the implant) were converted into three bipolar derivations (LFP01, LFP12 and LFP23) by sequentially referencing them. Power spectral density (PSD) was estimated using a fast Fourier transform from a 2-s-long sliding window (Hamming) with 1-s overlap. The final time-evolving spectra had 15 s time and 0.5 Hz frequency resolution. For each subject, LFP data selected for further analysis were based on the location of the DBS electrode contact within the STN and this was verified by the following: (a) intraoperative microelectrode recordings that identified cells with firing characteristics consistent with STN neurons; (b) anti-Parkinsonian benefit and side-effects of macrostimulation; (c) preoperative stereotactic T1- and T2-weighted images merged to a postoperative MRI demonstrating the position of the DBS electrode within the borders of STN; (d) the use of Framelink (Medtronic Corp.) software to analyze DBS position on the postoperative MRI; and (e) evaluation of the efficacy of post-programming stimulation for contralateral motor symptoms for each subject (Ince et al., 2010). Selection of which contact(s) to use for study recordings was based on the STN contact (s) associated with peak beta-spectrum activity as this feature correlates with the optimal programming contact(s) for the treatment of contralateral motor symptoms (Ince et al., 2010). These criteria were used to ensure that the selected contact was most reliably in the same relative anatomical location across patients to permit generalizability of the model.

Video-PSG scoring

The polysomnographic electrode montage used was the following: F3–C3, P3–O1, F4–C4 and P4–O2, EOGL–A2, EOGR–A1, and chin EMG (Iber et al., 2007). Sleep

stages were determined by analysis of 30-s epochs of the PSG, by a sleep neurologist, with each epoch classified as Awake or as belonging to one of the following sleep stages: rapid eye movement (REM), or the non-REM (NREM) stages of N1, N2 or N3.

Model description

We trained a feedforward artificial neural network (ANN) with a single hidden layer (Figure 2b) to prospectively identify whether a given 30-s epoch of STN-LFP recording took place during one of three possible states: REM, NREM or Awake. Inputs to the model were eight separate frequency band power bins, averaged over 30 s: delta (0–3 Hz), theta (3–7 Hz), alpha (7–13 Hz), low beta (13–20 Hz), high beta (20–30 Hz), and low gamma (30–90 Hz), high gamma (90–200) and high frequency oscillations (200–350). Each frequency range input feature was normalized independently by subtracting the mean and scaling by the variance of feature. The ANN output is a probability that the measured epoch occurs during one of the three possible states. Optimal ANN architecture was chosen based on the hyperparameter optimization detailed below. The ANN model utilizes a single hidden layer to encode the normalized spectral power bands within 32 features by calculating weighted sums of the input frequency power and scaling them by a non-linear function. Weighted linear combinations of these 32 features are then used by the network to compute sleep state probabilities with application of a softmax non-linearity.

Hyperparameter optimization

The architecture of the ANN model we describe was determined by evaluating classification accuracy across the spectrum of network hyperparameters. We

combinatorically varied the non-linearity of each unit (Sigmoid, ReLu and Tanh), the number of units in the hidden layer(s) (16, 32 or 64) and the number of hidden layers (1 or 2). Randomly initialized models in replicates of five were each trained and tested on a random 80:20 partition of all data. In general, we observed that more complex models with a larger number of total units and multilayer networks produced minor increases in classification accuracy, but these performance variations were not statistically significant. We opted to use 32 units in a single hidden layer with the biology-inspired rectified linear units (ReLu; (Hahnloser et al., 2000)) as the non-linearity. We chose this configuration because it achieved classification accuracy on a par with the best-performing model with 10-fold fewer parameters to minimize overfitting training data.

Results

Model performance and validation

We evaluated the ANN model's sleep stage classification performance and its ability to generalize new predictions under two conditions. Performance was evaluated using accuracy and Cohen's κ. Chance accuracy was calculated as originally described (Cohen, 1960).

First, we tested the model's ability to predict sleep stages on novel examples from patients included in the training set. We pooled 80% of each patient's 30-s STN-LFP recording epochs across all nine patients to train the model. The remaining 20% of the withheld epochs were used to evaluate the model's performance on novel examples from familiar patients. The train-test fractions (80:20) were sampled randomly for each patient and performance was averaged in replicates of five to prevent sampling bias.

The model was able to correctly predict sleep stage from STN-LFP epochs with a mean accuracy of 91% (Figure 3.3a).

Training a model from scratch for each new patient is often intractable. Therefore, the model's ability to perform well on never- seen subjects demonstrates its sensitivity to the salient spectral features of sleep across individual variations. To test this level of generalization, the model was trained on all epochs from eight of the nine patients. Subsequently, model performance was evaluated on all epochs from the kept-out patient. Thus, nine different models were trained, each with a specific patient withheld from its training data. As above, model performance was quantified using accuracy and Cohen's κ (Figure 3.3b). Across all models, mean classification accuracy of 91% was observed. Finally, because the number of epochs of each observed sleep state varies between patients in the dataset, we produced confusion matrices for the test patient of each model and show representative examples from patients with significantly imbalanced sampling as well as a summary matrix averaged across all models (Figure 3.3c). This demonstrates that the model's error rate varies as a function of sleep- stage representation, with less frequent stages showing a higher error rate (see Table 3.1).

Discussion

In this report, we demonstrate the novel use of an optimized ANN to predict sleep stage from 30-s epochs of LFP recorded from the STN of PD subjects. Based on results from hyperparameter optimization, we used a network architecture of a single hidden layer containing 32 artificial neurons with ReLu non-linearities (Figure 3.2b). We

evaluated the model's ability to generalize to new patients by using a LOGO (leave-one-group-out) strategy for cross-validation and attained mean classification accuracy of 91% averaged across all patients.

The ability of this ANN model to accurately predict sleep stages based on STN-LFP data recorded from novel PD patients is a critical improvement over our previously published effort to generate a predictive model. In our prior work, we used a support vector machine (SVM) model that performed well when tested on novel epochs derived from the familiar patient used to train the model but failed to generalize to novel subjects (Thompson et al., 2017). For simplification of model development, the different NREM stages (i.e. NREM 1–3) were aggregated into a single class. However, future development will focus on classification of the non-REM substages, as they represent distinct states and underlie unique sleep processes. Our current study is the first to use direct intracranial recordings from human basal ganglia to classify and match unseen PSG-labelled electrophysiological signals. Although the overall accuracy of the model for all sleep stages combined was well above chance (91%), performance on REM sleep stages was lower than the average performance (77%). Decreased performance for REM could be a result of the lower representation across subjects (see Table 3.1), or it may reflect the challenge in identifying the REM state from PSG in this patient population. This model can be implemented in forthcoming improved DBS neurostimulators to detect sleep stage solely from features of STN-recorded LFP, enabling the implementation of closed-loop stimulation strategies for treating sleep dysregulation in PD patients. This would serve a crucial unmet need in this patient

population (Chaudhuri et al., 2006), as there are currently no effective treatments with a low side-effect burden(Arnulf et al., 2000). Although DBS is an established therapy for the treatment of motor symptoms of Parkinson's disease, the effect of DBS on the sleep disturbances of Parkin- son's disease has not yet been fully characterized, and the mechanism(s) underlying the improvements reported in sleep quality, efficiency and duration remains to be elucidated(Sharma et al., 2018).

Our model's ability to correctly predict sleep stage in novel subjects may imply the existence of a universal LFP spectrum sleep signature within STN. In our investigations to date, this STN localized spectral signature appears conserved across patient demographics, robust to variances in implantation location, and detectable from the aggregate activity of several thousands of neurons. In future work, we intend to characterize this spectral signature space using generative ANN models of LFP oscillations recorded from within the STN. This effort will extend our understanding of the relationship between sleep dynamics and oscillating field potentials in the basal ganglia.

A **PD subject demographics (n = 9)**

				30 second sleep epochs (#)					
	Age (y)	**PD Dur. (y)**	**% Improv.**	**Total**	**Awake**	**REM**	**N1**	**N2**	**N3**
mean	60.11	10.67	61.89	837.11	377.89	62.29	134.78	186.00	88.40
median	61.00	9.50	61.00	791.00	282.00	73.00	106.00	221.00	10.00
std	9.56	4.63	11.40	165.67	246.58	45.61	94.79	122.20	112.90
min	39	6	47	676	97	11	33	3	4
max	70	19	79	1149	850	133	284	322	239
range	31	13	32	473	753	122	251	319	235

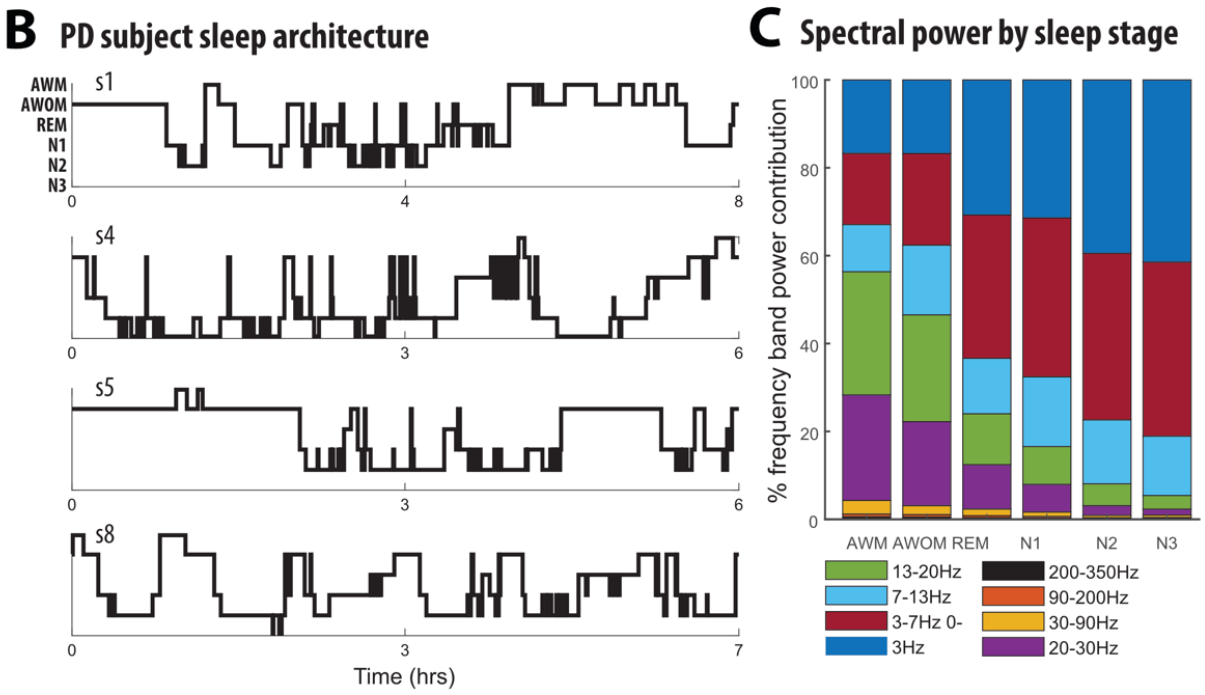

Figure 3.1 (a) Demographic data and sleep stage characteristics for Parkinson's disease (PD) subjects participating in this study (n = 9). Percent improvement in PD reflects the change in the Unified Parkinson's Disease Rating Scale (UPDRS) motor scale before and after DBS surgery. (b) Hypnograms from four representative subjects in this study, indicative of common sleep architecture deficits reported for individuals with PD. (c) Distribution of frequency band power contribution to sleep stage for all subjects. AWM, awake with movement; AWOM, awake without movement; REM, rapid eye movement

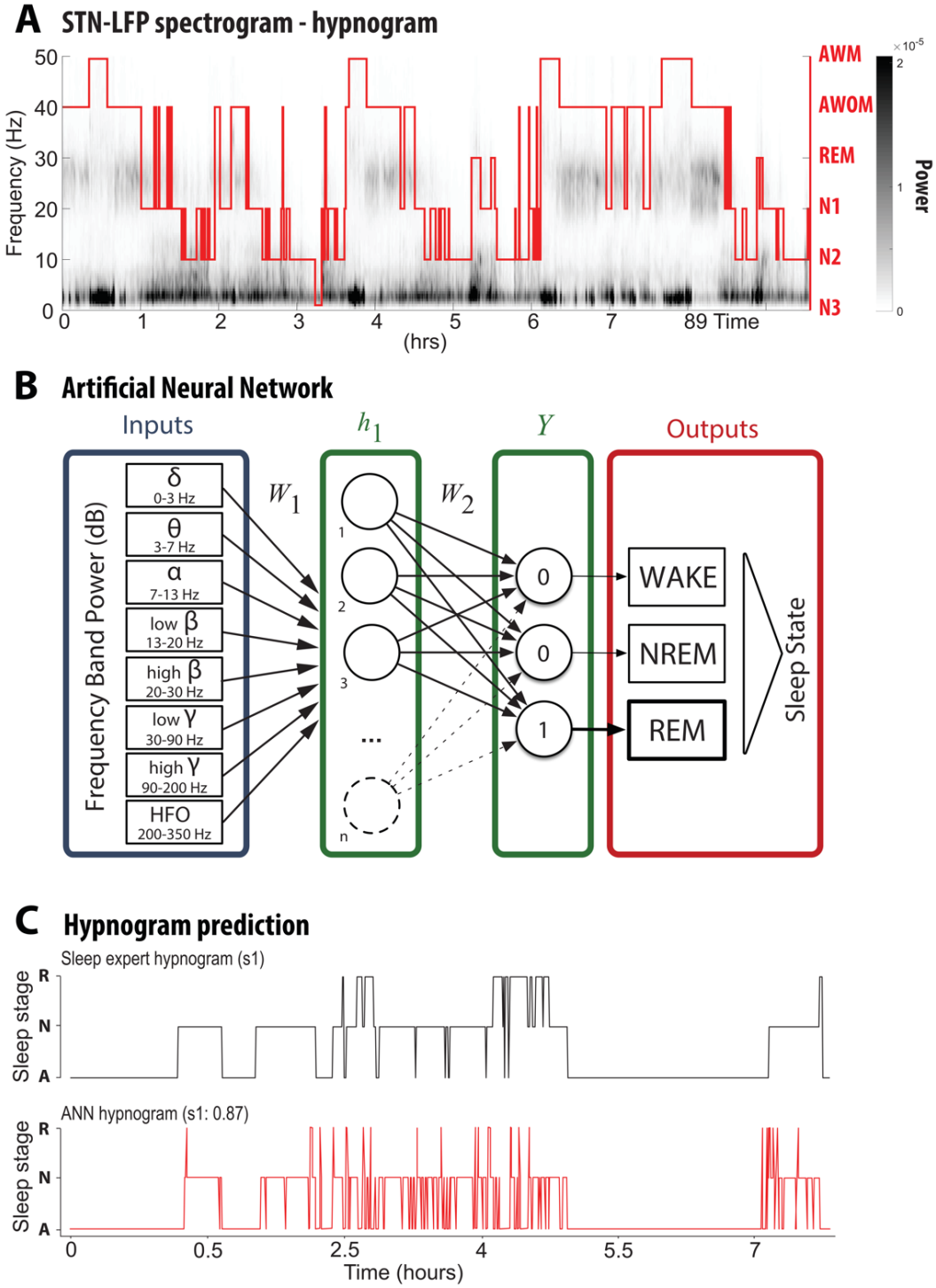

Figure 3.2 (a) Representative spectrogram of a local field potential (LFP) recording acquired over the course of one full night's sleep from a deep brain stimulation (DBS) electrode implanted into the subthalamic nucleus (STN). A PSG- informed hypnogram assessed by a sleep expert is aligned with the LFP recordings (red line; AWM, awake with movement; AWOM, awake without movement; REM, rapid eye movement; N1–3, non-rapid eye movement stages 1–3). (b) Schematic representation of the feedforward classifier used to predict sleep stage from 30-s labelled LFP epochs. The model is composed of an input layer (LFP frequency power bands), a hidden layer and an output layer (predicted sleep stage). (c) Comparison of hypnogram assessed by a sleep expert (top; black) and ANN- predicted hypnogram (bottom; red) from patient 1 with mean classification accuracy of 87%

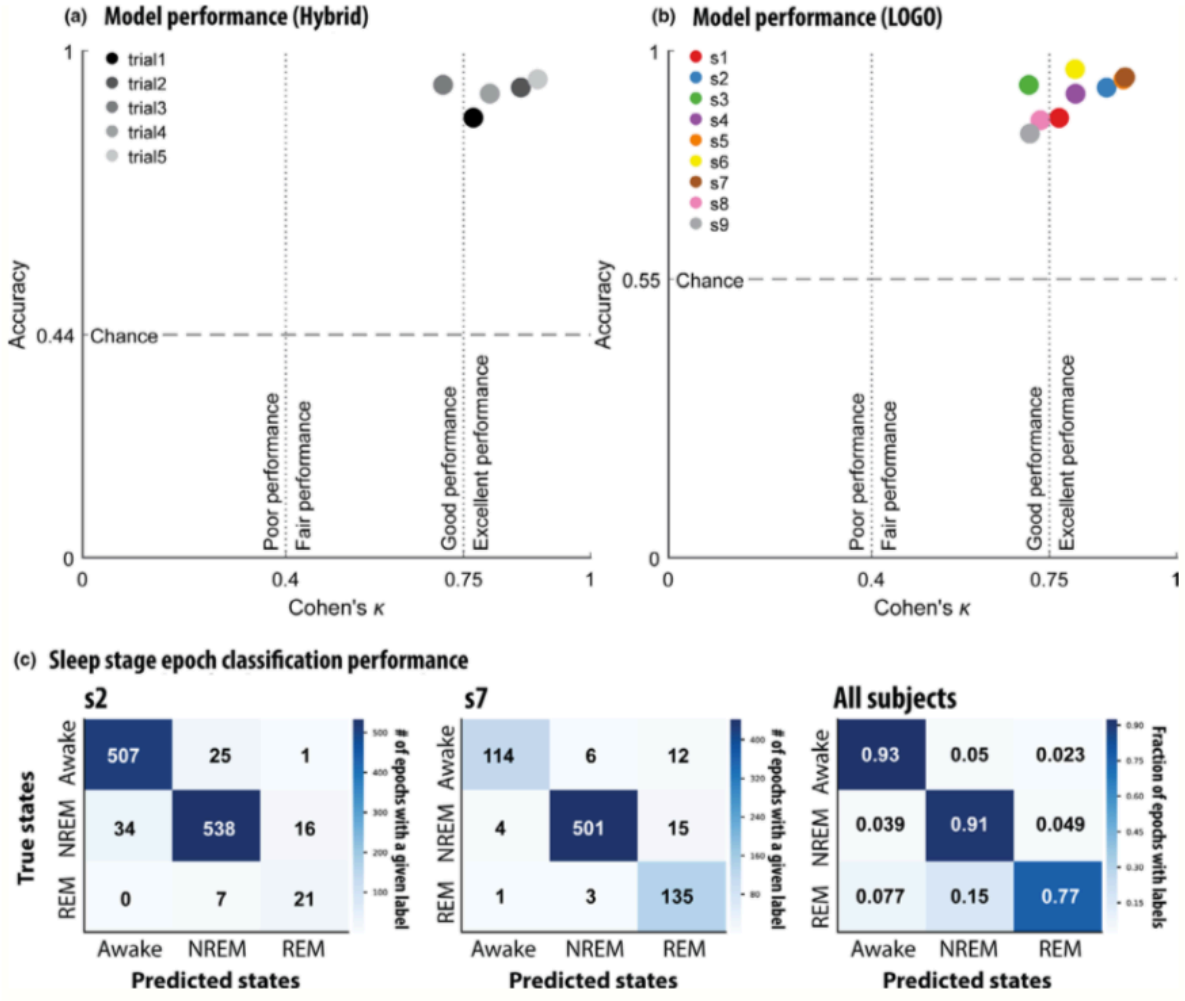

Figure 3.3 (a) In the "hybrid" strategy a random 80% of each patient's local field potential (LFP) recordings were pooled to train the model. Model accuracy and Cohen's κ were evaluated on the withheld 20% from each patient. This analysis was replicated in four other random 80:20 splits to control sampling bias. Cohen's κ magnitude guidelines derived from Fleiss & Cohen (1973). (b) A leave-one-group-out (LOGO) cross-validation strategy was used to test generalizability to unseen patients. Each data point represents a model trained with a specific patient excluded from its training data. Model accuracy and Cohen's κ were evaluated on data from the kept-out patient. (c) Confusion matrices of representative models trained using the LOGO cross-validation strategy. The first two confusion matrices represent individual subjects and the final confusion matrix depicts the fraction of epochs with specific class labels for all subjects. REM, rapid eye movement; NREM, non-rapid eye movement

Table 3.1 Summary for all subjects of the epoch representation and model accuracy for each of the following sleep stages: Awake, rapid eye movement (REM) and an aggregate of the non-rapid eye movement (NREM) substages (N1, N2 and N3)

	Awake		NREM		REM	
Subject ID	% of epochs	% correct	% of epochs	% correct	% of epochs	% correct
1	50	91	42	90	9	47
2	46	94	51	94	2	55
3	88	98	11	69	1	0
4	16	82	73	95	11	84
5	53	99	43	96	3	39
6	88	96	12	100	0	NA
7	17	96	66	98	18	83
8	27	94	61	85	12	83
9	40	96	60	100	0	NA

CHAPTER IV

MODELS OF VENTRAL STREAM THAT CATEGORIZE AND VISUALIZE IMAGES

Introduction

The ventral stream (VS) of visual cortex begins in primary visual cortex (V1), ends in inferior temporal cortex (IT), and is essential for object recognition. Accordingly, the long-standing belief in the field is that the ventral stream could be understood as mapping visual scenes onto neuronal firing patterns that represent object identity(Felleman and Van Essen, 1991). Supporting that assertion, deep convolutional neural networks (DCNN's) trained to categorize objects in natural images develop intermediate representations that resemble those in primate VS (Cadieu et al., 2014; Güçlü and van Gerven, 2015; Yamins et al., 2014; Yamins and DiCarlo, 2016). However, several recent findings appear at odds with the object recognition hypothesis. VS and other visual areas are also engaged during visualization of both prior experience and novel scenes (O'Craven and Kanwisher, 2006; Stokes et al., 2009), suggesting that the VS can generate visual scenes, in addition to processing them as inputs. Furthermore, non-categorical information, about object positions, sizes, etc. is also represented with increasing explicitness in late VS areas V4 and IT(Hong et al., 2016). This is not necessarily expected in a "pure" object recognition system, as the non-categorical information is not necessary for the categorization task. Thus, these recent findings challenge the long-held object recognition hypothesis of ventral stream

and raise the question: What computational objective best explains VS physiology? (Richards et al., 2019)

To address that question, we pursued a recently-popularized approach and trained deep neural networks to perform different tasks: we then compared the trained neural networks' responses to image stimuli to those observed in neurophysiology experiments(Cadieu et al., 2014; Chen and Crawford, 2019; Güçlü and van Gerven, 2015; Yamins et al., 2014), to see which tasks yielded models that best matched the neural data. We trained our networks to perform one of two visual tasks: a) recognize objects; or b) recognize objects while also retaining enough information about the input image to allow its reconstruction. We studied the evolution of categorical and non-categorical information representations along the visual pathway within these models, and compared that evolution with data from monkey VS. Our main finding is that neural networks optimized for task (b) provide a better match to the representation of non-categorical information in the monkey physiology data than do those optimized for task (a). This suggests that a full understanding of visual ventral stream computations might require considerations other than object recognition.

Materials and Methods

Dataset and augmentation

We constructed images of clothing items superimposed at random locations over natural image backgrounds. To achieve this goal, we used all 70,000 images from the Fashion MNIST dataset, a computer vision object recognition dataset comprised of images of clothing articles from 10 different categories. We augmented this dataset by

expanding the background of the image two-fold (from 28x28 pixels to 56x56 pixels) and drawing dx and dy linear pixel displacements from a uniform distribution spanning 75% of the image field {-11,11}. Images were then shifted according the randomly drawn dx and dy values. After applying positional shifts, the objects were superimposed over random patches extracted from natural images from the BSDS500 natural image dataset to produce simplified natural scenes which contain categorical (1 of 10 clothing categories) and non-categorical (position shifts) variation. Random 56x56 pixel patches from the BSDS500 dataset were gray scaled before the shifted object images were added to the background patch (Figure 4.1A). All augmentation was performed on-line during training. That is, every position shift and natural image patch was drawn randomly every training batch instead of pre-computing shifts and backgrounds. This allows every training batch to be composed of unique examples from the dataset and prevents overfitting.

Primate electrophysiology

Neural recordings were originally collected by the DiCarlo lab (Majaj et al., 2015) and shared with us for this analysis. In brief, neural recordings were collected from the visual cortex of two awake and behaving rhesus macaques using multi-electrode array electrophysiology recording systems (BlackRock Microsystems). Animals were presented with a series of images showing 64 distinct objects from 8 classes rendered at varying eccentricity in the animal's visual field. After spike-sorting and quality control this resulted in well-isolated single units from both IT (n=168) and V4 (n=128); higher-

order areas in primate visual cortex. A full description of the data and experimental methods is given by Hong et al. (2016).

Model architecture

Non-convolutional models were constructed by sequentially combining all-to-all (aka densely connected) layers. Any given layer uses the previous layers' output as input, multiplying the inputs (x) from by a weight matrix (w) and adds a bias to each unit in the output. Finally, this value is passed through a nonlinear activation function. Each layer outputs an activation vector of its units (y) which is function of its inputs (x).

Objective functions and training parameters

Models optimized for classification use categorical cross-entropy for the objective function. Categorical cross-entropy (XENT) is a commonly used objective function in machine learning to train neural network classifiers. Multilabel cross-entropy is calculated according to the equation below where M is the total number of classes.

$$XENT = -\sum_{c=1}^{M} y_c \cdot \ln(\hat{y}_c)$$

Here, y_c is the true category label, represented as a one-hot vector, and $\hat{y}_c$ is the network output obtained from the linear readout of population V (see Figure 4.1).

Models with an objective function term for reconstructing the original input scene use pixel-wise sum of squared error (SSE) between the input and the generator's output ($\hat{x}$).

$$SSE = \sum (x - \hat{x})^2$$

We trained each model in our experiment until classification accuracy plateaued on a validation dataset of 512 objects from the 10,000 test images in the fashion MNIST dataset.

Model Evaluation

After training performance plateaus, 192 randomly chosen unit activations from Layers 1-3 in the encoder model (Fig 4.1B) were used in comparisons with primate ventral stream electrophysiology. Unit activations were generated using a random sample from held out test images (not used during training). As in a (simulated) electrophysiology experiment, each image was input to the network, and the corresponding unit activations were recorded. We then analyzed these unit activations in the same way as we did the firing rates recorded in monkey visual cortex.

We measured selectivity of our artificial neurons in the same way as Hong et al 2016 (they call these measures “performance” instead of selectivity). For continuous-valued scene attributes (e.g. horizontal position) we measured selectivity as the absolute value of the Pearson correlation between the neuron’s response and that attribute in the stimulus image. For categorical properties (e.g. object class) we measure selectivity as the one-vs-all discriminability (d’).

We quantified the similarity of each models’ layer-wise selectivity to corresponding layers in primate ventral stream using Fisher’s Combined Probability Test (FCT). As discussed in the main paper, we first used the Welch’s unpaired t-test to calculate p-values model-VS pairs for all selectivity metrics in the corresponding layers, then used the FCT to combine those p-values into a single likelihood measure that

reflects the likelihood of observing the monkey physiology data, under the hypothesis that those data are drawn from the same distribution as the units computational model: a larger p-value corresponds to a model that more closely matches the monkey data.

Results

Computational models

To identify the degree to which different computational objectives describe ventral stream physiology, we optimized computational neural network models for different objectives, and compared them to neural recordings from the primate ventral stream. Each computational model was constructed out of a series of layers of artificial neurons, connected sequentially. The first layer takes as input an image x and outputs at the final layer outputs a set of neuronal activities that represent the visual scene input (Fig 4.1B), including object identity. We refer to this output as the latent representation. The input images, x, consisted of images of clothing articles superimposed over natural image backgrounds (see Methods). Each image used a single clothing article rendered in a randomly chosen position and placed over a natural image background (Fig. 4.1A).

The models each had a total of three layers of processing (corresponding to cortical areas V1, V2, ad V4) between their inputs and these latent representations; the latent representations correspond to area IT, for reasons we discuss below. The visual inputs to the model had normalized luminance values, mimicking the normalization observed in thalamic inputs to V1(Carandini and Heeger, 2011). The connectivity between neurons in each layer (and the artificial neurons' biases) were optimized within

each model, so as to achieve the specified objective (see Methods). We repeated this process for two different objectives, yielding two different types of models.

The first type of model was optimized strictly for object recognition: the optimization maximized the ability of a linear decoder to determine the identity of the clothing object in the visual scene from the latent representation. (This mirrors the observation that neural activities in area IT can be linearly decoded to recover object identity(Majaj et al., 2015)). The second type of model was optimized for two tasks simultaneously: the ability of a linear decoder to determine object identify from latent representation, and the ability of a decoder to reconstruct the object from the latent representation. (See Methods for details about the optimization procedure). We repeated this procedure with both convolutional, and non-convolutional neural network architectures, yielding a total of four models.

In all cases, the models were optimized using sets of images containing randomly sampled objects, until their object classification performance saturated on a set of held-out validation images. Good performance on the categorization task was obtained in all models (Fig 4.1D). Having developed models optimized for these different objectives, we could evaluate how well each model matched observations from primate VS, and use that comparison to determine which computational objective provides the best description of primate VS.

Comparisons to macaque electrophysiology

To compare our neural network models to ventral stream physiology, we used the experimental data from a previously published study (see methods and (Hong et al.,

2016) for details). These data consisted of electrode array recordings from areas V4 and IT of monkeys that were viewing images; many neurons in each area were simultaneously observed. Within these data, we assessed each neuron's selectivity for object identify, and for category-orthogonal image properties (e.g. horizontal object position), as in Hong et al(Hong et al., 2016) (see methods). We performed this analysis for the monkey data, and for the artificial neurons in each layer of each of our computational models. We then compared the trends in image property selectivity displayed by primate VS neurons and units from each of our models along the visual processing pathway.

In the primate VS, selectivity for both categorical and category-orthogonal scene attributes increased along the ventral stream (Fig 4.2A), as reported by Hong et al8. This indicates that both types of attributes are more explicitly represented in progressively deeper ventral stream areas.

Within our computational models, those models optimizing the composite objective showed the same trends observed in primate ventral stream neurons (Fig 4.2C, 4.2E): both category and category-orthogonal properties of the visual scene are represented more explicitly with each subsequent layers of the model. This observation persisted for both the convolutional and the non-convolutional architectures. For contrast, models optimized solely for object recognition (without the image reconstruction component of the objective function) did not show consistent increases in position selectivity along the visual pathway (Fig 4.2B, 4.2D). Again, this observation held for both convolutional and non-convolutional model architectures.

Thus, models optimizing the composite objective function qualitatively recapitulate the trends in neuronal selectivity along the visual pathways better than do models optimized strictly for object recognition. This observation motivated us to quantify how well each model matched the primate VS data. To achieve this goal, we performed the following analysis on each computational model. First, we used unpaired t-tests to estimate the probability that there is no difference in object category selectivity between the primate IT data and the model's latent representation. We then performed a t-test comparing the primate V4 category selectivity to the corresponding layer of the computational model. Next, we performed t-tests comparing the horizontal, and vertical, position selectivities in primate V4 and IT to the corresponding layers of the computational model. This procedure yielded 6 p-values, describing the probability that the model matched each of these attributes observed in the primate VS. Finally, we used Fisher's method (Li et al., 2014) to combine those 6 p-values into a single number, that quantified the likelihood of there being no difference between the computational model and the primate VS.

Comparing these likelihood values, we found that the convolutional models overall provided better descriptions of the primate VS than did the non-convolutional ones (i.e., they had higher likelihood values), and that the best model overall was the convolutional neural network optimized for the composite classify-and-reconstruct objective (See Figure 4.4).

Noise Robustness

We found that the convolutional model, optimizing the composite objective (classify-and-reconstruct) best matched the depth-dependent increase in position selectivity seen in single unit activities recorded from primate ventral stream. This led us to ask whether there might be functional benefits for networks optimizing this composite objective function, as compared with ones that are just trained to classify their inputs.

Further motivating this question, we note that previous work has shown that convolutional neural networks optimized for object recognition tend to perform poorly on object recognition tasks when the images are corrupted by noise. Specifically, classification performance has been seen to decrease significantly when networks are evaluated under noise conditions even marginally different from the conditions under which it was trained (Geirhos et al., 2018). This is different from the primate visual system, where object recognition performance is more robust to image noise, leading us to speculate that the convolutional networks trained for the composite classify-and-reconstruct task – which provide the best match to primate VS data – might have classification performance that is more robust to image corruption than do the networks trained purely for object recognition.

To test that hypothesis, we took each of our previously trained models, and measured their accuracy at categorizing the clothing objects in test images corrupted by increasing levels of additive pixel noise (see methods). Similar to previous work, the convolutional model trained purely for object recognition showed a decrease in performance as the noise level increased. For the convolutional model trained on the

composite task, the decrease in performance with increasing noise level was less severe. This suggests that, consistent with our hypothesis, there is a functional benefit to systems optimizing the composite objective over “pure” object recognition systems: their object recognition performance is more robust to noise.

The same finding also holds for the non-convolutional model architectures, and they are overall more robust to image noise than are the convolutional ones. We repeated this analysis with multiplicative (instead of additive) pixel noise and made very similar observations (see Figure 4.5). This shows that our findings are not specific to the additive noise model.

Discussion

Here we report evidence that convolutional neural networks (CNNs) optimizing a two-part composite objective (recognize and visualize) describe the depth-dependent evolution of categorical and non-categorical information in primate VS better than do networks optimized for object recognition alone. This is unexpected, as prior work posits that networks optimized strictly for object recognition should form the best models of primate VS(Cadieu et al., 2014; Hong et al., 2016; Richards et al., 2019; Yamins and DiCarlo, 2016). Our results suggest that the evolution of category-orthogonal information along the visual pathway could require a different functional explanation. Moreover, consistent with previous work(Cadieu et al., 2014; Hong et al., 2016; Richards et al., 2019; Yamins and DiCarlo, 2016), our CNNs optimized for image classification resemble primate VS more closely than do non-convolutional models optimizing the same objective.

Our findings may help reconcile discrepancies between the object recognition hypothesis of VS and results which appear at odds with this interpretation(Freud et al., 2016; O'Craven and Kanwisher, 2006; Sereno and Lehky, 2011; Stokes et al., 2009), for example the finding that primate VS explicitly retains information not useful for object recognition(Hong et al., 2016). The composite objective promotes retention of both category and category-orthogonal information because both are necessary to reconstruct the stimulus.

Importantly, we used a different method to compare our neural networks to the primate VS than have previous studies that compared the representational dissimilarity matrices (RDMs) for their models, with those of the primate VS(Cadieu et al., 2014; Hong et al., 2016; Richards et al., 2019; Yamins and DiCarlo, 2016). While RDMs assay the (dis)similarity (Nili et al., 2014) in how different images are represented by the models, or primate VS, our approach was to focus instead on the depth-dependent evolution of neuronal selectivity to categorical and non-categorial variations in the input images. That we came to a different conclusion than did prior studies -- E.g., that an objective other than pure object categorization could best describe the computations in primate VS – suggests that there could be aspects of visual computation that are not fully captured by RDM analysis.

Furthermore, our findings suggest noise tolerance as another independent explanation for why the VS might use a composite computational objective. VS classification accuracy measured in humans tolerates noise corrupted images much better than DCNNs optimized for image classification alone(Geirhos et al., 2018). In

contrast, convolutional model's optimizing the composite objective demonstrate better noise tolerance compared to identical models trained solely for classification (Fig 4.3). Importantly, improved noise tolerance occurs without having to augment training images with noise. These findings complement the expanding body of work to explain the neuronal computations in visual processing and have applications in the computer vision models that emulate them.

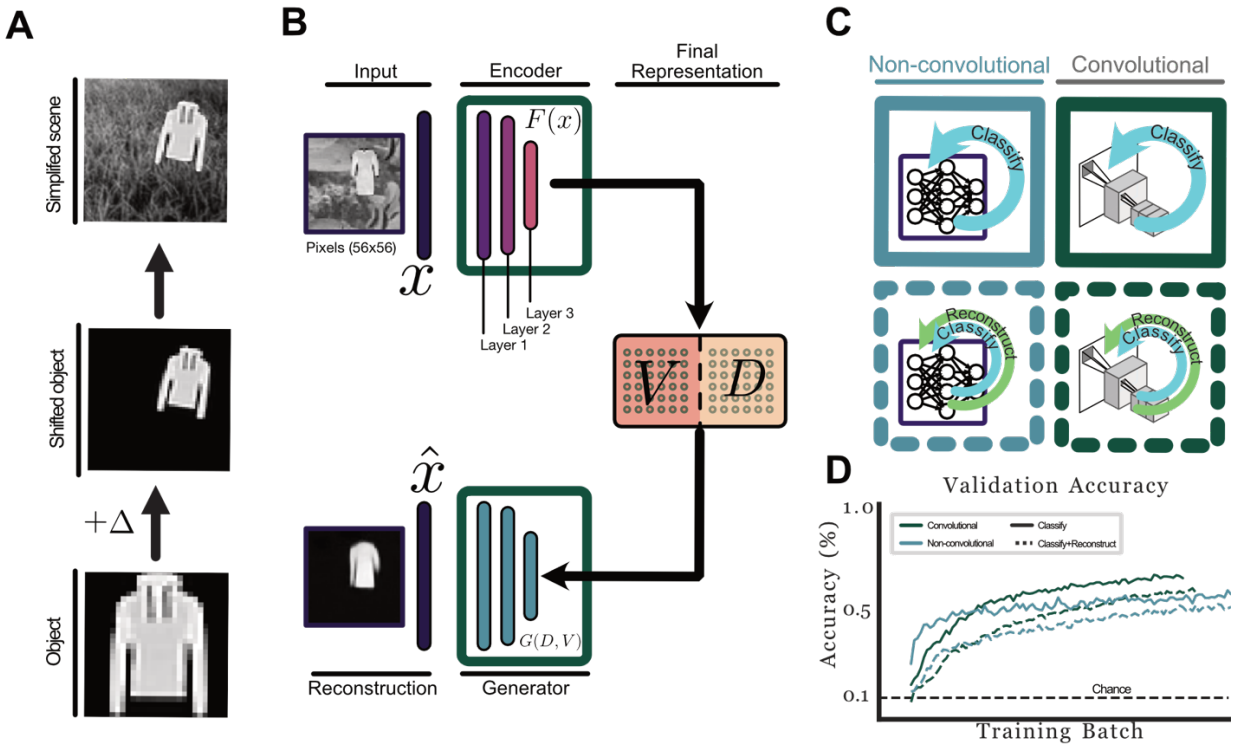

Figure 4.1 Overview. A) We constructed images of clothing items superimposed over natural image backgrounds at random eccentricities. B) We model the ventral stream as an encoder whose objective is to map input image (x) onto more abstract "latent" representations (D and V). In our models this entire latent space is represented by 70 artificial neurons (35 units in each of D and V) The generator network uses these latent representations (D and V) as input to reconstruct the object and its location within the scene. A separate linear decoder attempts to determine the object identity from the activities of the units in V. C) We trained both convolutional, and non-convolutional neural network architectures, on one of two tasks: object categorization ("classify"), or object categorization with concurrent image reconstruction. We note that, for the "pure" object recognition task, the generator network is superfluous. D) Neural networks with both architectures achieve comparable object recognition performance (accuracy) when using either classify-only and classify+reconstruct objective functions. This performance was assessed on held-out images, not used in training the networks.

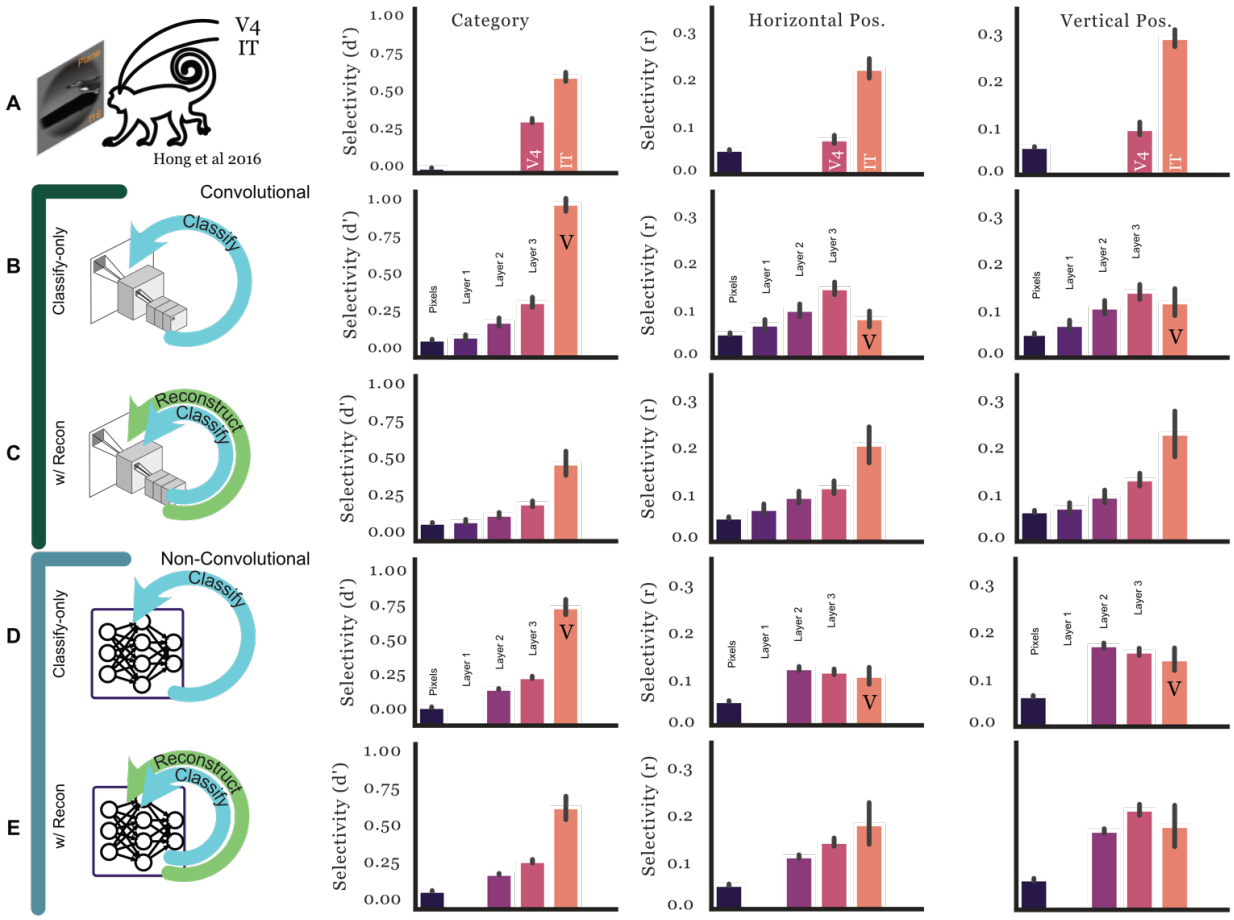

Figure 4.2 Comparisons of selectivity for visual scene properties. A) Category and position selectivity of single units recorded from macaque ventral stream (see Methods and Hong et al. 2016). B&C) Selectivity of units in the fully trained convolutional models optimized under classify-only objective (categorical cross-entropy) and the composite classify+reconstruct autoencoder objective. D&E) Non-convolutional or "all-to-all" models were also trained on both classify-only and classify+reconstruct. We measured property selectivity of both categorical and continuous valued category-orthogonal properties on units in the multi-electrode array data and each layer of the computational model encoders. We defined selectivity for categorical information on each unit in the dataset as the absolute value of that unit's discriminability (one-vs-all d-prime). We defined selectivity for continuous valued attributes (horizontal and vertical position) on each unit as the absolute value of the Pearson correlation coefficient. Unit activities for models were sampled using 10000 held out test images to generate activations at each layer of the model. For layers containing more than 192 units we randomly sampled 192 units for the analysis (to have a number of units similar to the number of IT units in the neural recordings).

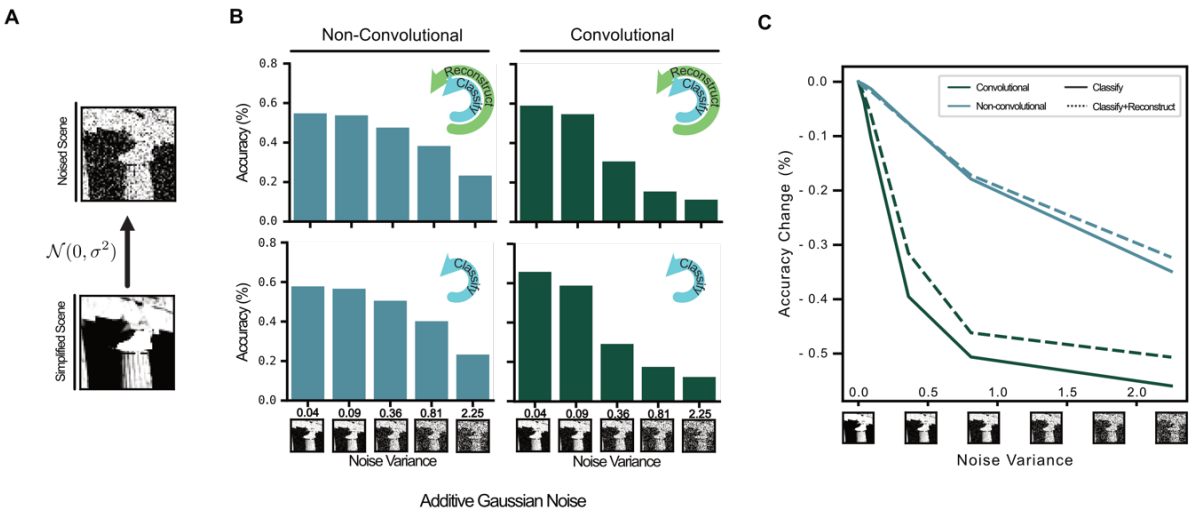

Figure 4.3 Noise generalization properties of models. A) Additive gaussian noise (mean=0) was used to corrupt 10,000 testing images at increasing levels. B) Each model (defined its architecture – convolutional or non-convolutional -- and the objective on which it was trained) was evaluated on images corrupted with increasing levels of gaussian noise. We show the accuracy at categorizing the objects in the noise-corrupted images. These images were from a held-out dataset, not used in training the neural networks. C) Convolutional neural networks are more sensitive to noise than are non-convolutional ones; they show a larger decrease in accuracy with increasing noise variance. Adding a reconstruction component to the network objective reduces this sensitivity. Similar results were obtained with a multiplicative noise model (Fig. S2), indicating that this result is not sensitive to the specific type of noise that corrupts the images.

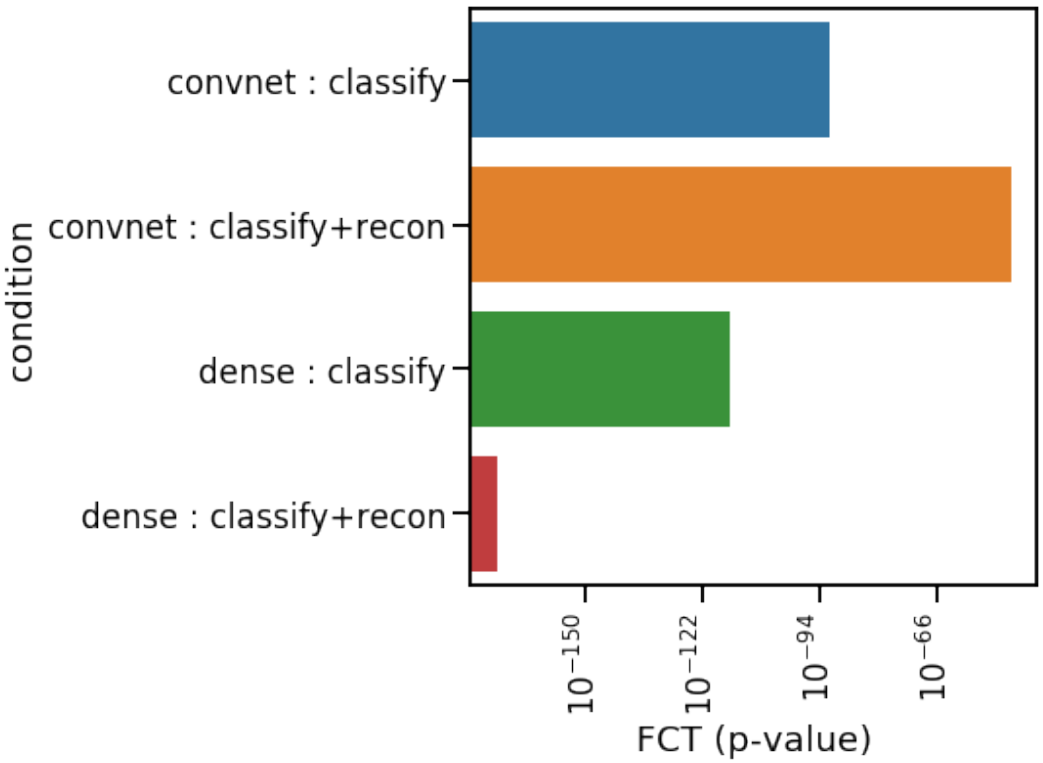

Figure 4.4 Fisher combined probability test. We used the FCT to compute the likelihood of each model's category and position selectivity matching the data observations made in monkey ventral stream recordings. Those likelihoods (p-values) are shown for each model. Higher p-values (taller bars) correspond to models that more closely match the neural data.

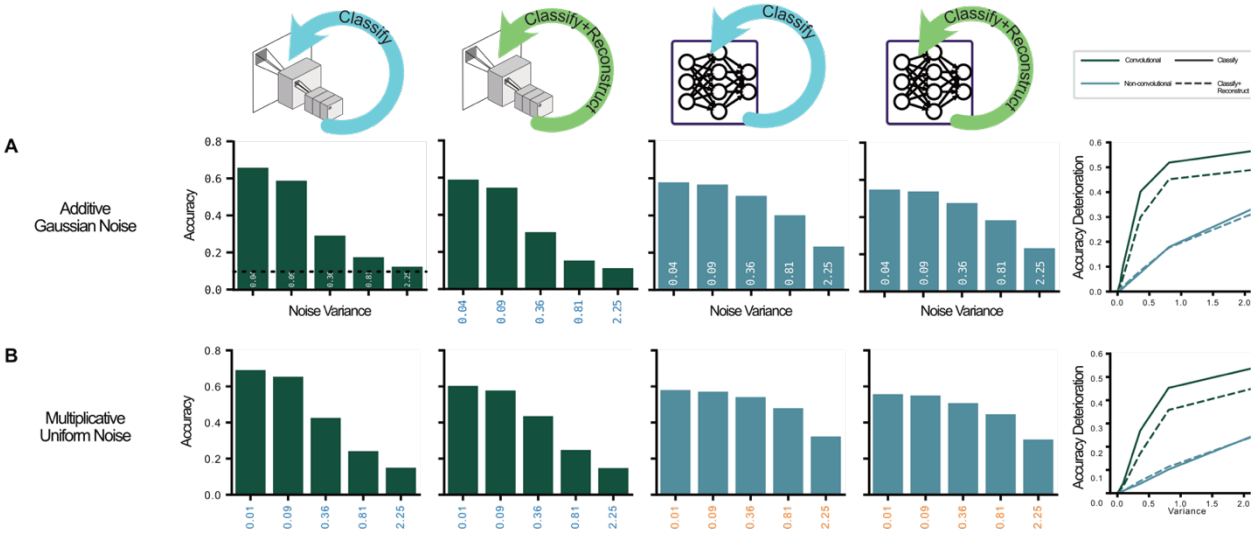

Figure 4.5 Noise generalization properties of models across noise types. Each model (defined its architecture – convolutional or non-convolutional -- and the objective on which it was trained) was evaluated on images corrupted with increasing levels of noise. A) Additive gaussian noise (mean=0) was used to corrupt 10,000 testing images at increasing levels. B) Multiplicative uniform noise) was used to corrupt 10,000 testing images at increasing levels. Bar plots show the accuracy of each neural network model at categorizing the objects in those noisy images. C) We show the deterioration in accuracy at each noise level, for each model. This comparison shows that the convolutional neural networks are more sensitive to noise but adding a reconstruction objective appears to improve this sensitivity.

CHAPTER V

PREDICTING SINGLE NEURON RESPONSES IN MACAQUE V1[2]

Introduction

Our ability to see arises because of the activity evoked in our brains as we view the world around us. Ever since Hubel and Wiesel (Hubel and Wiesel, 1959) mapped the flow of visual information from the retina to thalamus and then cortex, understanding how these different regions encode and process visual information has been a major focus of visual systems neuroscience. In the first cortical layer of visual processing—primary visual cortex (V1)—Hubel and Wiesel identified neurons that respond to oriented edges within image stimuli. These are called simple or complex cells, depending on how sensitive their responses are to shifts in the position of the edge. The simple and complex cells are well studied (David et al., 2004; Lehky et al., 1992; Montijn et al., 2016). However, many V1 neurons are neither simple nor complex cells, and the classical models of simple and complex cells often fail to predict how those neurons will respond to naturalistic stimuli (Olshausen and D. J. Field, 2005). Thus, much of how V1 encodes visual information remains unknown. We use deep learning to address this longstanding problem.

Recent advances in neural-recording technology and machine learning have put solving the V1 neural code within reach. Experimental technology for simultaneously recording from large populations of neurons— such as multielectrode arrays—has

[2] Portions of this chapter are previously published by Kindel, W; Christensen, E; and Zylberberg, J in (2019) *Journal of Vision* 19:29 and are included with the permission of the copyright holder. WK and JZ conceived the original project. EC and WK analyzed and interpreted the data. WK and EC drafted the manuscript. EC, WK, and JZ critically revised the manuscript.

opened the door to studying how the collective behavior of neurons encodes sensory information. Moreover, methods of machine learning inspired by the anatomy of the mammalian visual system, known as convolutional neural networks, have achieved impressive success in increasingly difficult image-classification tasks (Krizhevsky et al., 2012; LeCun et al., 2015). Recently, these artificial neural networks have been used to study the visual system (Yamins and DiCarlo, 2016), setting the state of the art for predicting stimulus-evoked neural activity in the retina (McIntosh et al., 2016) and inferior temporal cortex(Yamins et al., 2014). Despite these successes, we have not yet achieved a full understanding of how V1 represents natural images.

In this work, we present a convolutional neural network that predicts V1 activity patterns evoked by natural image stimuli. We use this network to predict the activity of 355 individual neurons in macaque monkey V1, in which it represents the neural visual code for many neurons regardless of cell type. On held-out validation data, the network predicts firing rates that are highly correlated ($CC_{norm} = 0.556 \pm 0.015$) with the neurons' actual firing rates. This performance value is quoted for all neurons, with no selection filter. Performance is better for more active neurons: When evaluated only on neurons with mean firing rates above 5 Hz, our predictors achieve correlations of $CC_{norm}=0.69 \pm 0.01$ with the neurons' true firing rates. Our deep network is overall more accurate than a library of other models used as a baseline for comparison.

Methods

Experimental Data

We used publicly available multielectrode recordings from macaque V1 downloaded from the Collaborative Research in Computational Neuroscience website (http://crcns.org; Coen-Cagli, Kohn, & Schwartz, 2015). In these experiments, macaque monkeys were anesthetized and then presented with a series of images while the experimenters recorded the spiking activity of a population of neurons in V1 (Figure 5.1A and 5.1B) with a multielectrode array. Each image was presented for 100 ms, and there was a 200-ms blank screen shown between images. These recordings were conducted in 10 experimental sessions with three different animals, resulting in recordings from a total of 392 spike-sorted neurons whose receptive fields were centered on the stimulus. In the publicly available data, both well- isolated single units and small multiunit clusters are present. In our main analysis, we consider all of these as neurons; we also separately performed an analysis in which we attempted to distinguish between the single neurons and the small multiunit clusters. That result is included in the Discussion. A full description of the data and experimental methods is given by Coen-Cagli et al. (Coen-Cagli et al., 2015). Unlike those researchers, who used selection criteria based on responses to visual stimuli and reported results from a subset of 207 neurons, we used no further selection criteria and used all 392 spike-sorted and centered neurons. We used 37 of these neurons from one experimental session to determine how to construct our network (its hyperparameters), and the

remaining 355 neurons to evaluate its performance. For each neuron n, we calculated the mean firing rate $A_{n.i}$ evoked by each image i by averaging its firing rate across the 20 repeated presentations of that image. The firing rates were calculated over a window from 50 to 100 ms after the image was presented, to account for the signal- propagation delay from retina to V1 (Figure 5.1D; V1 firing rates increase dramatically at; 50 ms after stimulus onset). We separately analyzed firing rates computed over a longer (100-ms) window, from 50 to 150 ms after stimulus onset; the results of that analysis are presented in the Discussion section.

We analyzed the responses to 270 natural images circularly cropped with a 1º aperture (Figure 5.1B). All 392 neurons are centered such that the 1º image aperture fully contains every neuron's receptive field. The full data set contains responses to natural and artificial stimuli, both full-size and cropped. We used only natural images because we are interested in the real-world behavior of the visual system, and we used only the cropped images because they have the same visual field as the grating stimuli that we used to characterize the neurons as either orientation selective or not.

Model

To construct our predictive network, we used a convolutional neural network (CNN) whose input is an image and whose output is the predicted firing rates of every neuron in a given experimental session. Prior to training the neural network, we down-sampled the images using a nonoverlapping 2 x 2 window and cropped them to a size of 33 x 33 pixels. As shown in Figure 5.2, the network consists of a series of linear–

nonlinear layers. The first layer(s) performs local feature extraction on the image by sweeping banks of convolutional filters over the image and then applying a maximum pooling operation. These local features are then globally combined at the all-to-all layer(s) to generate the predicted firing rate for every neuron in that data session.1

The number of each type of layer (convolutional with maximum pooling or all-to-all) and the details about each layer (number of units, convolution stride, etc.) were optimized to maximize the accuracy of the neural-activity predictions on the 37 neurons recorded in the second experimental session. We did this using a combination of manual and automated searches, where the results of our manual search informed the range of the hyperparameter space for an automated random search (Bergstra and Bengio, 2012). A subset of the results from the manual search is shown in Figure 5.3A and 5.3B. In Figure 5.3A, the number of convolutional layers, the kernel size of the convolutions, the pooling stride, and the loss function are adjusted. During training, units are randomly silenced (dropped out), which is a commonly used method for preventing overfitting in neural networks (Srivastava et al., 2014). In Figure 3B, we take the best-performing networks with one, two, and three convolutional layers and adjust the dropout keep rate. Using the best-performing set of parameters, we defined our best CNN, denoted CNN2 because it is a two-convolutional-layer network. We trained and evaluated CNN2 using the data from the remaining nine experimental sessions.

For each experimental session, we trained our network using a cross-validation procedure where we randomly subdivided the given data set into a training subset (80% of the images and corresponding V1 activity patterns) and an evaluation subset (20% of

the images). We then trained all layers of our network using the TensorFlow Python package with the gradient-descent optimizer. Based on the results of our hyperparameter search, which showed that this loss function outperforms the alternative log-likelihood one, we attributed a loss

$$L_n = \frac{\sum_i \left(y_{n,i} - A_{n,i}\right)^2}{var_i(A_{n,i})}$$

to each neuron (indexed by n), where i is the image index, $A_{n,i}$ the measured response, and $y_{n,i}$ the network's predicted response. The neurons' losses are summed, yielding the total loss used by the optimizer. To ensure that the performance generalizes, the training data were subdivided into data used by the optimizer to train the weights (66% of the images) and another small subset (14% of the images) to stop the training when accuracy stops improving (early stopping). To quantify the performance of the predictor, we compared the network's predicted firing rates to the neurons' measured firing rates using a held-out evaluation set. This set was used neither to determine the hyperparameters nor to train the weights in our neural network. We calculated the Pearson correlation coefficient CC_{CNN2} between the predicted and measured absolute firing rates for each neuron. Following the convention of Schoppe et al. (Schoppe et al., 2016), we scaled the Pearson correlation coefficient by its theoretical maximum value given neural variability to yield the normalized Pearson correlation coefficient

$$\mathrm{CC}_{norm}^{CNN2} = \frac{\mathrm{CC}_{abs}^{CNN2}}{\mathrm{CC}_{max}}$$

that we use to quantify our results. Thus, in principle, a perfect model can achieve $CC_{norm} = 1$. To compute CC_{max}, we followed a bootstrapping procedure (in contrast to Schoppe et al., 2016) where we generated fake data by drawing random numbers from Gaussian distributions with the same statistics as the measured neural data. For each neuron and image, we averaged over 20 of these values to obtain a simulated prediction. We then computed the correlation between these simulated predictions and the neurons' actual mean firing rates to find the maximum correlation CC_{max} possible given the variability in stimulus-evoked neural firing rates. While we acknowledge that neural firing rates are not Gaussian distributed, the CC_{max} estimate, being a second-order statistic of the neural firing rates (and their estimates via the predictor networks), is sensitive only to the first- and second-order statistics of the neural data. A Gaussian distribution captures these first- and second-order statistics while making as few assumptions as possible about the higher order statistics in the data (i.e., it is a second-order maximum entropy model). As a result, our use of Gaussian distributions does not affect the reliability of our estimates of CC_{max}: Using more complex, harder-to-estimate probability distributions would yield the same result. For this reason, we are confident that our bootstrapping procedure, while different from that of Schoppe et al., is comparable to their method.

Comparisons with other models

We compared the results obtained from CNN2 to those of a variety of other models. In implementing our comparisons, we used identical cross-validation protocols to determine the training and evaluation data that were used to train CNN2. When the models contained hyperparameters (including regularization parameters), these parameters were optimized on data from the same experimental session used to optimize the hyperparameters of CNN2. We also evaluated all models in the same way, using the normalized Pearson correlation between predicted and actual neural firing rates.

We organized our models for comparison in two broad groups: models that are fully data driven, where all the model parameters were learned from our neural-activity data sets, and models where only a linear regression is performed on neural-activity data sets using regularization by the least absolute shrink- age and selection operator (LASSO). The models using LASSO regression, denoted ‘‘trained with regression,’’ often use external information about visual processing. The fully data-driven models are denoted ‘‘trained in TensorFlow.’’ Our pixel model could fit into either category but is grouped with the LASSO models. The LASSO comparison models are pixels, SAILnet, Berkeley Wavelet Transform, and five VGG-16-based models. The fully data-driven comparison models are linear–nonlinear (LN), LN-LN, and a one- and a three-convolutional-layer network.

Pixels

First, we constructed a linear model by performing a weighted sum over all pixel values of an image stimulus with a bias to yield a predicted neural activity for each neuron. That is, we formed a prediction

$$y_{n,i}^{pixels} = b_n + \sum_j W_{n,j} x_{j,i}$$

for the activity $A_{n,i}$ of neuron n, where $x_{j,i}$ is the j-th pixel value in image i and the constants $W_{n,j}$ and b_n are determined from linear regression using LASSO regularization, a type of L1 (sparse) regularized linear regression. The LASSO regularization parameter was optimized on data from the same experimental session used to optimize the hyperparameters of CNN2. Then, leaving this term fixed, we evaluated the model using cross-validation on data from the other nine experimental sessions.

SAILnet

Next we constructed a SAILnet implementation of a sparse-coding model. In the SAILnet model the images are first whitened, using the whitening filter defined by Olshausen and Field (Olshausen and D. J. Field, 1996). The whitened images are then passed into a sparse-coding model, which outputs the activations of 1,089 different image features; the number of features is chosen to match the number of pixels. The image features, and the activations, are optimized so as to maximize the fidelity of image encoding while minimizing the number of active features. As an alternative to the

SparseNet imple- mentation(Olshausen and D. J. Field, 1996), we used the SAILnet model(Zylberberg et al., 2011).

After training SAILnet on whitened natural-image patches, we froze the weights and passed in whitened versions of the images shown to the monkeys, to obtain the activations $z_{j,i}$ of each feature (indexed by *j*) for each image (indexed by *i*). We then constructed a linear predictor of the neuron firing rate, from the activations of the sparse-coding features, with prediction

$$y_{n,i}^{SAILnet} = b_n + \sum_j W_{n,j} z_{j,i}$$

Similar to the pixels model, we optimized the biases and weights of this predictor using linear regression with LASSO regularization.

Berkeley Wavelet Transform

We constructed a Gabor model called the Berkeley Wavelet Transform (BWT) model. To construct the BWT model, we trimmed the outer edges of the small images by cropping the images down to 243 3 243 pixels, removing part of the gray background (the BWT requires square images with edge sizes of a power of 3). We then passed each image through the BWT using code shared by the authors(Willmore et al., 2008). We did this for all of the small images and then selected those wavelets whose outputs had nonzero variance over the set of images (there are 16,478 of those, out of the total of 59,049 wavelets); the ones with zero variance occurred because they look at the gray parts of the images (see Figure 5.1B). We used the coefficients of these 16,478 wavelets to predict the neurons' mean firing rates, using LASSO regression with an identical protocol to that of the SAILnet model. The regression was on the weights *W*

and biases *b* according to the previous equation, where the variables $z_{i,j}$ are BWT wavelet activations.

VGG

To add a comparison to the work(Cadena et al., 2018), we constructed five models from a deep CNN called VGG-16 that has been pretrained on an image classification task (Simonyan and Zisserman, n.d.). We constructed these models out of the activations of VGG at five different depths along the deep network in response to our image set. To do this, we trimmed the outer edges of the small images and cropped down to 224 x 224 pixels, then copied the grayscale images into each of the R, G, and B channels to match the 224 x 224 x 3 input size of VGG. (This duplicates the fact that the monkey has the three input channels but saw grayscale images.) We then passed these images through the (already trained) VGG-16 model and extracted the activations from each layer. Of the layers, we focused on convolutional blocks 2 and 3 because the LASSO fitting is much slower on such large inputs (e.g., >590,000 units in convolutional 3 block 2), and Cadena et al. (2017) show that these blocks provide the best predictions of V1 firing rates. For each layer's activations, we selected those units whose activations had nonzero variance over the set of images; the ones with zero variance occurred because they look at the gray parts of the images. We used the activations of these units to predict the neurons' mean firing rates, using L1-regularized (sparse) LASSO regression. The regression is on the weights W and biases b according to Equation 4, where the variables $z_{i,j}$ are VGG activations within the given layer. The five

VGG layers we considered are $Conv_{2,1}$, $Conv_{2,2}$, $Conv_{3,1}$, $Conv_{3,2}$, and $Conv_{3,3}$ (where $Conv_{a,b}$ denotes convolutional layer *b* within block *a*).

Linear-Nonlinear (LN)

We constructed an LN model by applying a nonlinearity to a linear model to yield a prediction for each neuron. According to the LN model we formed a prediction

$$y_{n,i}^{LN} = \sigma\left(b_n + \sum_j W_{n,j}\, x_{j,i}\right)$$

for the activity of neuron *n*, where $\sigma(x)$ is a nonlinear function. A parametric rectified linear was chosen as the nonlinearity because it outperformed a parameterized sigmoid. The parameters of the model were trained in TensorFlow using the same learning process as for the convolutional models, with early stopping as the primary form of regularization.

LN-LN

We constructed an LN-LN model by cascading two LN models. Thus,

$$y_{n,i}^{LN-LN} = \sigma_2\left(b_n + \sum_k W_{n,k}\, \sigma_1\left(b_k + \sum_j W_{k,j}\right)\right)$$

forms the LN-LN model, where $\sigma(\cdot)$ is the rectified linear function, and the subscripts on $\sigma_1(\cdot)$ denote the layer 1. This model was trained in TensorFlow using the same learning process as the convolutional models, with early stopping as the primary form of

regularization. Its hyperparameters, such as the number of hidden elements, were optimized on the same experimental session as CNN2. Our LN-LN model is a non-convolutional LN-LN. There are more complex versions that use convolutions and pooling at the input stage; those are more similar to our CNN1(Vintch et al., 2015).

CNN1 and CNN3

In order to show the importance of model depth or lack thereof, we compared our chosen best model—the two-convolutional-layer network (CNN2)—to a single-convolutional-layer network (CNN1) and a three- convolutional-layer network (CNN3). The hyperparameters of CNN1 and CNN3 were optimized on data from the same experimental session used to optimize CNN2, and the models were regularized using a combination of dropout and early stopping.

Characterizing the selectivity of cells

To show that our model describes the activity of a broad class of cell types, we grouped the cells into functional classes and looked at how well the firing rates from each class could be predicted by our neural- network model. We classified cells by their selectivity to specific natural images, their selectivity to specific orientations of grating stimuli, their average firing rate over all images A, and their reliability CC_{max}.

The selectivity of each neuron to specific natural images is quantified by

$$image\ selectivity\ index = \left(N - \frac{(\sum_i A_i)^2}{\sum_i (A_i^2)}\right)\frac{1}{N-1}$$

where A_i is the cell's firing rate indexed *i* over the set of N images(Zylberberg and DeWeese, 2013). This index has a value of 0 for neurons that fire equally to all images and a value of 1 for cells that spike in response to only one of the images.

The neuron's orientation selectivity is measured by

$$circular\ variance = 1 - \frac{\left|\sum_\theta A_\theta e^{i2\theta}\right|}{\sum_\theta A_\theta}$$

where Ah is the neuron's firing rate in response to a grating oriented at angle h. The circular variance is less sensitive to noise than the more commonly used orientation-selectivity index (Mazurek et al., 2014). Following the results of Mazurek et al. we used thresholds of circular variance < 0.6 to define orientation-selective cells (the simple and complex cells according to Hubel & Wiesel, 1959) and circular variance > 0.75 non-orientation-selective cells. We omitted all other cells from these two groupings.

Results

Using our optimal network, we predicted firing rates that were highly correlated with the measured firing rates for most neurons (Figure 5.4A) when evaluated on held-out data. The correlation between the predicted and actual neural firing rates is $\overline{CC}_{norm}^{CNN2} = 0.556 \pm 0.015$ ($\overline{CC}_{abs}^{CNN2} = 0.493 \pm 0.014$) averaged over all 355 neurons in the evaluation set without using any selection criteria (Figure 5.4B). To benchmark the accuracy of our model, we compared it to a variety of other models (Figure 5.4B). We found that CNN2 is, indeed, the best- performing model. In comparison with fully data-driven models (denoted ‘‘trained in TensorFlow’’), we found that our two-convolutional-layer CNN2 is more accurate than single- (CNN1) and triple-convolutional-layer (CNN3) models, and far more accurate than shallower models such as LN. Compared to pretrained models where only LASSO regression was performed on the neural-activation data, we found that our optimized data-driven CNN outperforms models based on VGG, the Berkeley Wavelet Transform, and the SAILnet sparse-coding

algorithm (see Methods for details). Because simple and complex cells have been extensively studied, we were motivated to compare the predictability of simple and complex cells to the predictability of the other neurons in the data set. Grouping the cells into orientation-selective (simple and complex like cells) and non-orientation-selective cells (see Methods), we found that our network predicts non-orientation-selective cell responses with $\overline{CC}_{norm}^{CNN2} = 0.50 \pm 0.02$ and orientation-selective cell responses with $\overline{CC}_{norm}^{CNN2} = 0.55 \pm 0.04$. Therefore, our model predicts the firing rates of both cell types, performing slightly better on the simple- and complexlike cells than the non-orientation-selective cells.

Given that some neurons' firing rates are well predicted by the network (CNN2) while others are not, we were motivated to ask what distinguishes predict- able from unpredictable cells. Furthermore, we found that the cells that are well predicted CNN2 are also well predicted by CNN1 (Figure 5.5D) and CNN3 (Figure 5.5E), indicating these differences in predictability are set by the cells themselves rather than the neural-network architecture. To better understand what is driving these differences among the cells, we characterized the cells according to several metrics and then saw how these metrics can explain the distribution of predictability over the population of cells. We quantified the cells according to their orientation selectivity (see Methods), their image selectivity (see Methods), their average firing rate over all images and trials A□, and their reliability over repeat image presentations, as quantified by the theoretical upper bound on predictability CC_{max}. Comparing the predictability of each cell's firing rates with its respective image-selectivity index (Figure 5.5A) and circular variance (Figure

5.5B), we found that the predictability depends only weakly on these characteristics. Thus, orientation selectivity and image selectivity are only minor factors in determining how well our model performs.

We found that a neuron's activation, or mean firing rate over all images $\bar{A}$ (Figure 5.5C), and its limit neural reliability CC_{max} (Figure 5.5F) are both strongly related to the model's performance. Cells with a low mean firing rate $\bar{A} < 5$ Hz are less well described by our model, with $\overline{CC}_{norm}^{CNN2} = 0.29 \pm 0.03$. Selecting only the norm more active cells ($\bar{A} <$ 5 Hz) yields improved predictability, with $\overline{CC}_{norm}^{CNN2} = 0.69 \pm 0.01$, increased for neurons with greater mean firing rates. Similarly, we found that the model performs much better on reliable neurons than on those with low neural reliability. As the limit CC_{max} on predictably set by the neural reliability decreases, the model performance decreases by far more, meaning that overall the model does far worse at predicting the activity of these neurons. Selecting only the reliable neurons, $CC_{max} > 0.80$, yields improved predictability, with $\overline{CC}_{norm}^{CNN2} = 0.68 \pm 0.01$. Thus, we found that our model describes particularly well the neural encoding of both the cells that are more active ($\bar{A} < 5$ Hz) and the neurons that are more reliable ($CC_{max} > 0.80$).

Discussion

We trained a deep convolutional neural network to predict the firing rates of neurons in macaque V1 in response to natural image stimuli. In contrast to shallow models, such as linear–nonlinear models that can only describe simple cells, we find

that our convolutional neural network can describe a broad range of cells. Firing rates of both orientation-selective and non-orientation-selective neurons can be predicted with high accuracy. Our network describes the more active and more reliable cells particularly well. Additionally, we find that the two-convolutional-layer network outperforms a variety of other models.

Our results take a key step toward cracking the neural code for how visual stimuli are translated into neural activity in V1. This would be a major step forward in sensory neuroscience and would enable new technologies that could restore sight to the blind. For example, cameras could continuously feed images into networks that would determine the precise V1 activity patterns that correspond to those images: a camera-to-brain translator. Brain-stimulation methods like optogenetics (Ozbay et al., 2015) could then be used to generate those same activity patterns within the brain, thereby restoring sight.

Model comparisons and depth

Comparing across all of our fully data-driven models (Figure 5.4B, fully trained) of visual processing in V1, we find that increasing the complexity or depth of the models increases the ability of these models to replicate the visual processes that take place in V1, up to a convolutional neural network with two convolutional layers. Increasing the depth saturates or modestly decreases this CNN2 network's performance. We also find some difference between networks of comparable depths. For instance, the CNN1 and LN-LN networks are both the same depth, with two hidden layers. However, CNN1 does far better at predicting the firing rates in V1. The increased performance of CNN1 is

perhaps due to the constraints of the convolutional filters. We want to emphasize that our LN-LN model represents only a small subset of all the possible LN-LN models, and our CNN1 model could be classified as an LN-LN model. Overall, our results support the hypothesis that a model architecture with two convolutional layers and an all-to-all layer well represents the visual processing that takes place in V1.

Comparisons to other work

Although it is difficult for a variety of reasons to fairly compare the performances of published results, we predict neural activity with performance that is comparable to the state of the art. Over all neurons, the correlation between our network predictions and the actual neural firing rates is $\overline{CC}_{abs}^{CNN2} = 0.493 \pm 0.014$. For comparison, Lau, Stanley, and Dan (2002) achieved predictability of $\overline{CC}_{abs} = 0.45$ for simple cells and $\overline{CC}_{abs} = 0.31$ for complex cells; Vintch et al. (2015) achieved predictability of $\overline{CC}_{abs} = 0.55$ for simple cells and $\overline{CC}_{abs} = 0.42$ for complex cells; and Prenger, Wu, David, and Gallant (2004) achieved $\overline{CC}_{abs} = 0.24$ averaged over all cells. Lehky et al. (1992) achieved$\overline{CC}_{abs} = 0.78$, and Willmore, Prenger, and Gallant (2010) achieved a predictability (quantified as fraction of variance explained) of 0.4. However, some contextual factors confound direct comparison to these results. Specifically, Lehky et al. selected neurons that are easier to predict by specifically choosing neurons that responded strongly to the presentation of bars of light; Vintch et al. analyzed direction-selective neurons; and Willmore et al. adjusted their image to match the receptive field of each neuron they predicted. We, by contrast, neither tailored our stimulation to our

neurons nor selected well-behaved neurons. By selecting on either reliability or activation, we could easily achieve $\overline{CC}_{abs}^{CNN2} > 0.6$.

Consistent with Cadena et al. (2017), we find that the VGG layer most predictive of V1 neural firing rates is Conv3,1. However, in contrast with Cadena et al., we find that our data-driven CNN outperforms even this best VGG layer. In this comparison, confounds include having different images sets, using different methods for optimizing hyperparameters of CNNs, and using anesthetized monkeys rather than awake monkeys.

Identifying visual features that elicit high activity

In addition to making predictions of neural activity, the CNN represents the underlying visual processing that drives the population of neurons to spike. As an example of how to use the model as a tool to investigate the functions of individual neurons, we used Deep-Dream-like techniques (Mahendran and Vedaldi, 2015) to identify the visual features that cause each cell to spike. We inverted our network by finding input images that cause a given cell to spike at a prespecified level. To do this, we first took the fully trained network and set Gaussian-white-noise images as the input. We then used back-propagation to modify the pixel values of the input image to push the chosen neuron's predicted firing rate toward the prespecified level. Thus, we found an input image that induced the prespecified response. We applied this procedure to several different neurons that are well described by the model, and at several different target firing rates (Figure 5.6). Cells A ($\overline{CC}_{abs}^{CNN2} = 0.88$) and B ($\overline{CC}_{abs}^{CNN2} = 0.89$) appear to function like previously characterized cells. Cell A responds to a center–surround image

feature, and cell B's receptive field is a Gabor wavelet. In contrast, cells C ($\overline{CC}_{abs}^{CNN2} = 0.91$) and D ($\overline{CC}_{abs}^{CNN2} = 0.90$) appear to respond to more abstract image features that are not well represented by simple localized image masks. For comparison, we plot the receptive fields according to the LN model (Figure 5.6, left).

By inverting our network, we showed that we can use it as a tool to investigate neurons' response properties that cannot be found with shallower models. Going forward, this method shows potential for characterizing the response properties of more cells in V1 and precisely defining functional cell types that have been previously overlooked. Looking beyond V1, these methods could be applied to understanding higher level cortical processing, such as visual encoding in V2. By finding the features that elicit a response in V2 neurons, this tool could help fill the visual-encoding knowledge gap (Ziemba et al., 2016) that exists between the abstract encoding of inferior temporal cortex and V4 and the low-level encoding of the retina and V1.

Window length and well-isolated neurons

In our main analysis, we focused on predicting the initial neural response to exclude influence of top-down feedback from higher cortical areas. That is, we focused on the timescale when biological neural processing is most analogous to the feed-forward architecture of the artificial neural networks in our study. Because we considered only the initial response of the neurons to the stimulus, we were motivated to ask how well our network architecture can predict the neurons' firing rates, estimated by counting spikes over the full 100-ms window in the data of Coen-Cagli et al. (2015). Repeating our analysis with 100-ms windowed data, we found that our predictions have

correlation $\overline{CC}_{norm}^{CNN2} = 0.506 \pm 0.006$ to the measured firing rate over all neurons. This is slightly worse than our main analysis, where we used a 50-ms window. This result is not surprising, because we optimized the hyperparameters of our model using a 50-ms window.

Because the dataset we use groups both well-isolated neurons and small multiunit clusters, we were motivated to see how our best CNN2 model performs at predicting firing rates of each of these unit types. Following Coen Cagli et al. (2015), we identified the most well-isolated neurons by choosing only those whose signal-to-noise ratio in the spike sorting is greater than 2.75, and the remaining neurons (spike-sorting signal-to-noise ratio < 2.75) are an indistinguishable mixture of small multiunit clusters and single neurons. We found that the most well-isolated neurons have a predictor performance of $\overline{CC}_{norm}^{CNN2} = 0.414 \pm 0.016$, whereas the mixture of clusters and single neurons has $\overline{CC}_{norm}^{CNN2} = 0.635 \pm 0.012$. We were initially surprised by this finding, as we expected the well-isolated single units to be the most predictable. However, the multiunit clusters, being aggregates of several neurons, have higher average firing rates: 12.6 ± 0.6 spikes/s on average ($M \pm SEM$), compared with 8.4 ± 0.8 spikes/s for the well-isolated single units (estimated during the 50-ms spike-counting window). Recall that neurons with higher firing rates were generally more predictable (Figure 5.5C). We thus attribute the higher predictability of the multiunit clusters to their higher mean firing rates.

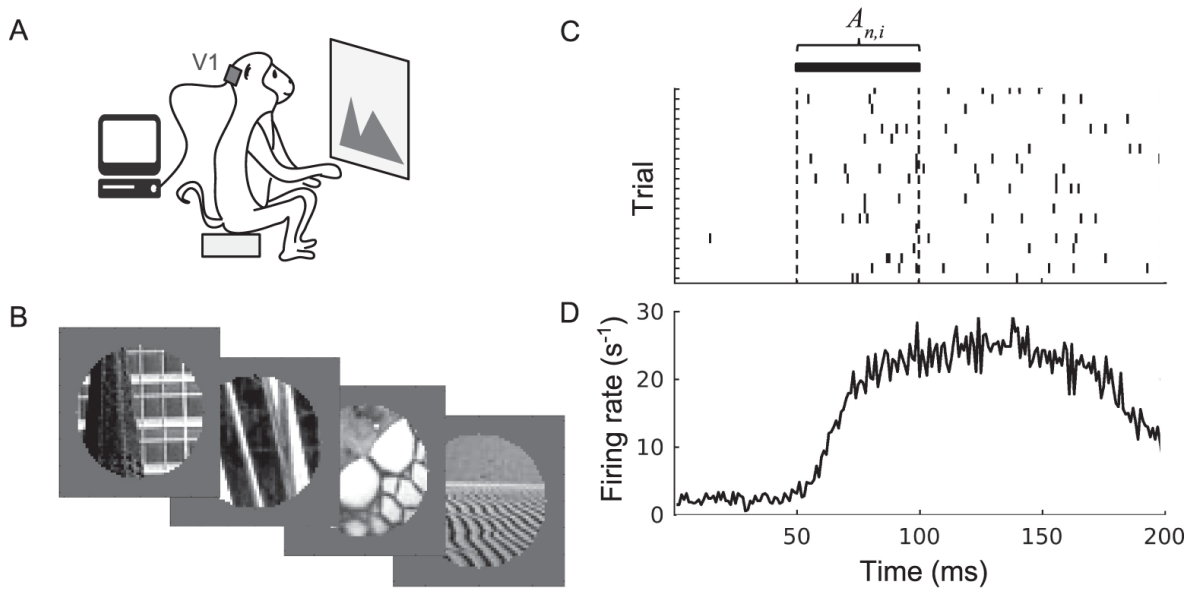

Figure 5.1 Experimental data collection and processing. (A) Neural activity was recorded in monkeys' V1 as they were shown a series of images. (B) The image set contains 270 circularly cropped natural images. (C) The response of a single neuron over repeated presentations of an image. Ticks indicate the neuron's spiking; each row corresponds to a different image-presentation trial. During the response window, the firing rate is computed and then averaged over trials to yield the average response An,i used in our analysis. (D) The neuron responds to image stimuli with a latency of ;50 ms from the image onset at t 1⁄4 0, as seen in the peristimulus time histogram (firing rate plotted against time, averaged over all 270 images).

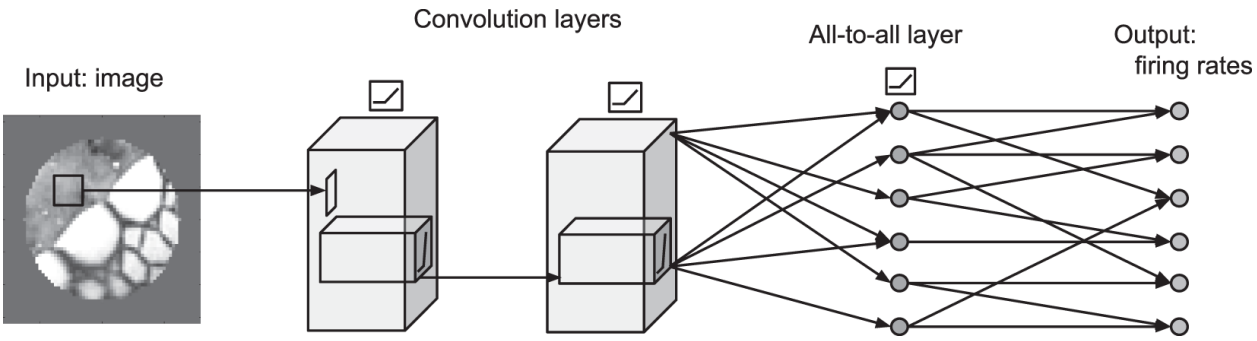

Figure 5.2 The optimized architecture of the deep convolutional-neural-network model. The network's inputs are the pixel values of an image, and each output unit gives the predicted firing rate of a single neuron in monkey V1.

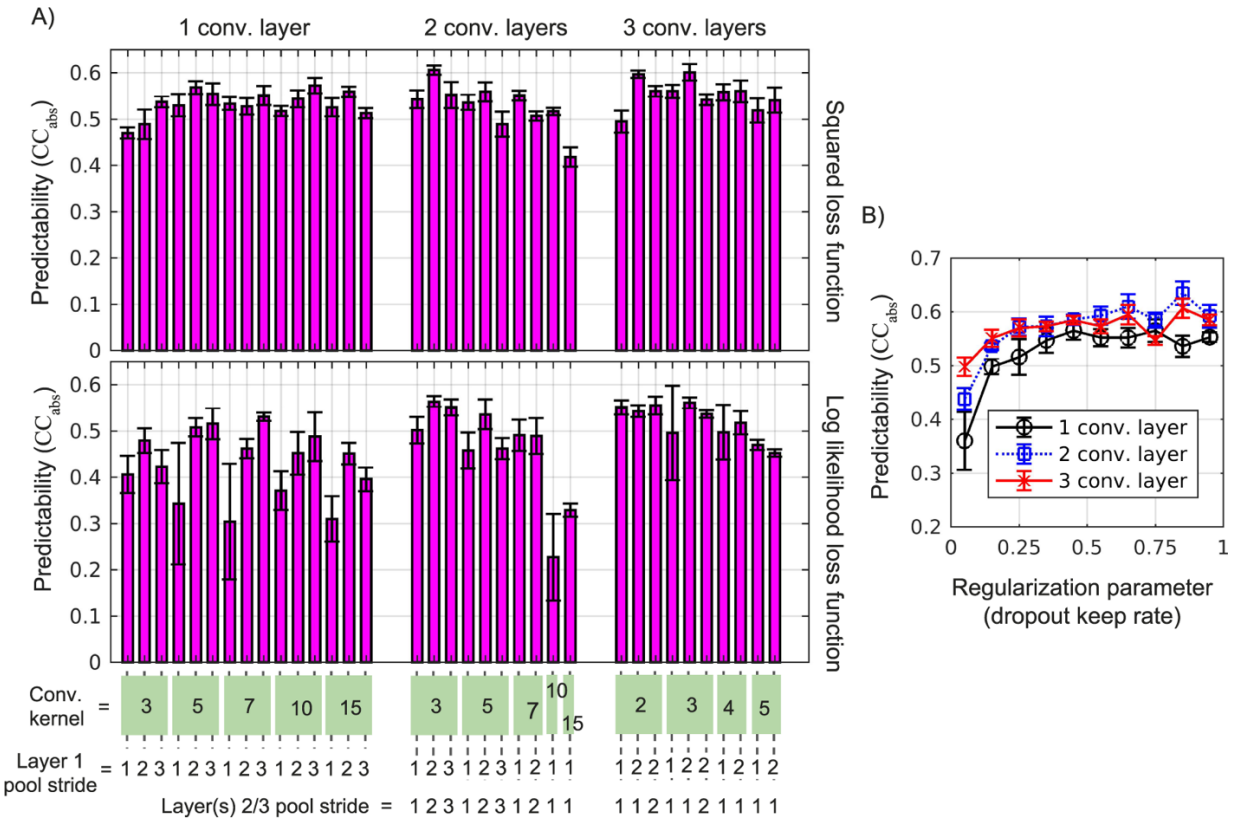

Figure 5.3 The hyperparameter optimization of the deep convolutional-neural-network model. (A) Adjusting the number of convolutional layers, loss function, convolutional kernel size (size of filters), and maxpool strides (scale of down-sampling) for just Layer 1 and both Layers 2 and 3. Each point is computed from the average Pearson correlation coefficient between the model's predictions and measured firing rates on one of the 10 experimental sessions with the standard error computed from five distinct partitions of training and evaluation data. (B) Adjusting the dropout keep rate for the best-performing networks with one, two, and three convolutional layers.

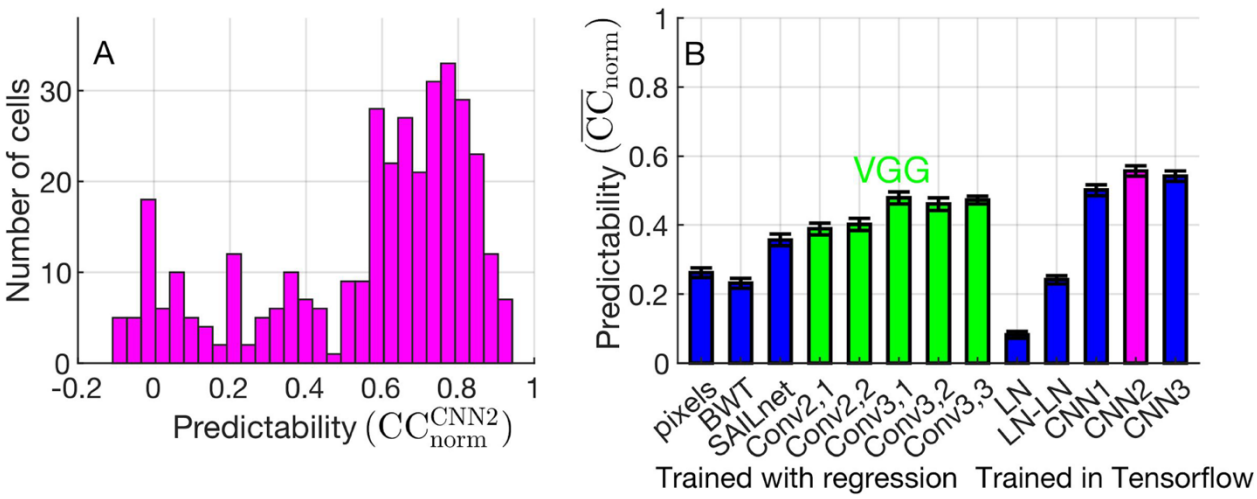

Figure 5.4 The performance of the best convolutional network model, CNN2. (A) A histogram of the normalized Pearson correlation coefficients between the network predictions and the actual firing rates CCCNN2 of all 355 neurons. (B) The average performance of norm the convolution-neural-network predictor (CNN2) compared to a variety of other models. The models are grouped as models that are trained only with regularized linear regression by least absolute shrinkage and selection operator on the neural-activity data (pixels, Berkeley Wavelet Transform [BWT], SAILnet, and our VGG models) and models where all the parameters are fully trained on the neural activity using TensorFlow (linear–nonlinear [LN], LN-LN, CNN1, and CNN2). The five VGG models in green are denoted Conva,b for convolutional layer b within block a.

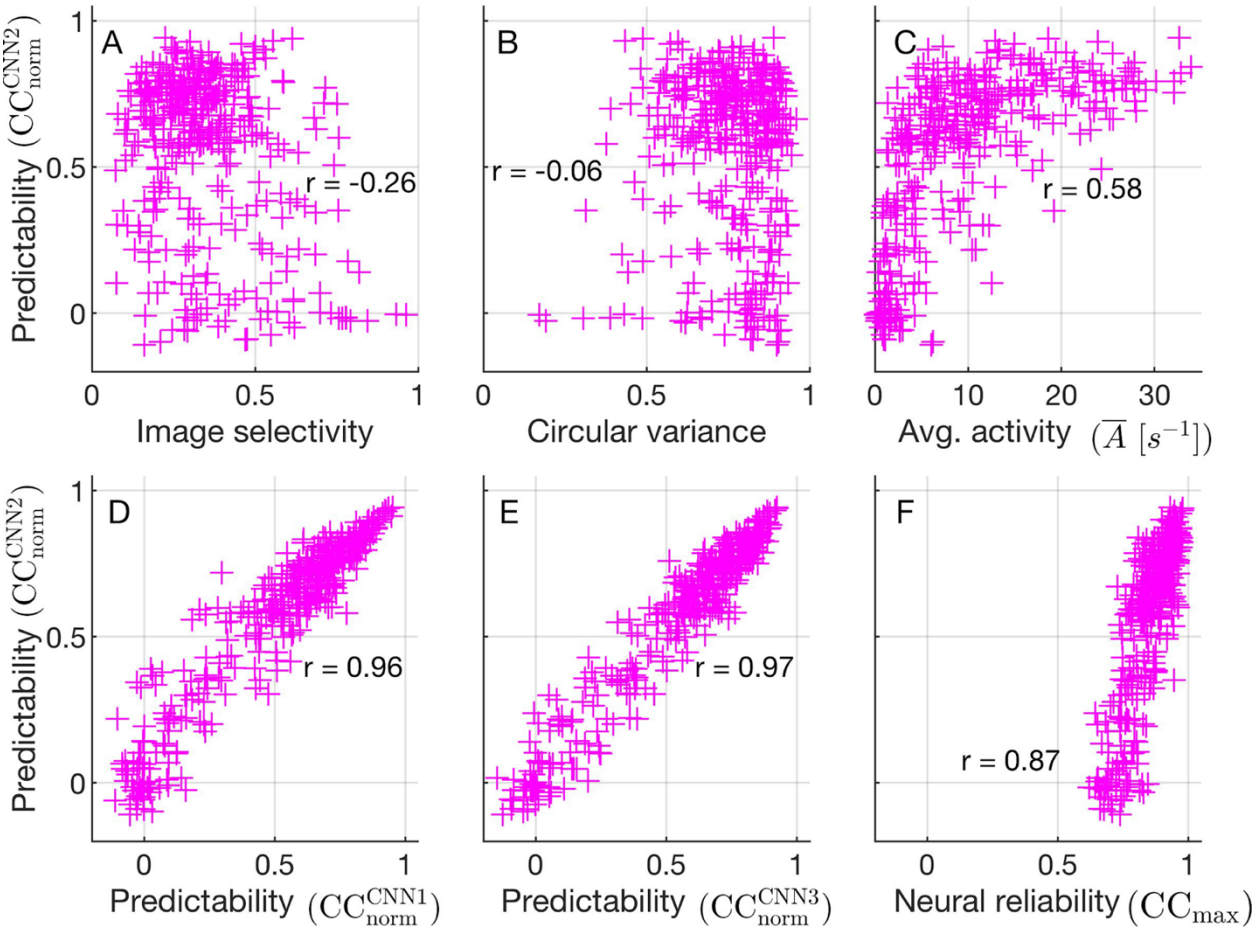

Figure 5.5 Characterizing the predictability of CNN2 (CC^{CNN2}) over the population of neurons; each data point corresponds to a single norm neuron. (A) Scatterplot of how well the predictor can predict each neuron's firing rate CC^{CNN2} (vertical axis) against the neuron's image selectivity (horizontal axis). (B) Scatterplot against the neuron's circular variance (horizontal axis). (C) Scatterplot against the neuron's average firing rate A□ (horizontal axis). (D) Scatterplot against the predictability CC^{CNN1} of CNN1 (horizontal axis). (E) Scatterplot against the predictability CC^{CNN3} of CNN3 (horizontal axis). (F) Scatterplot against the neural reliability CC_{max} (horizontal axis)

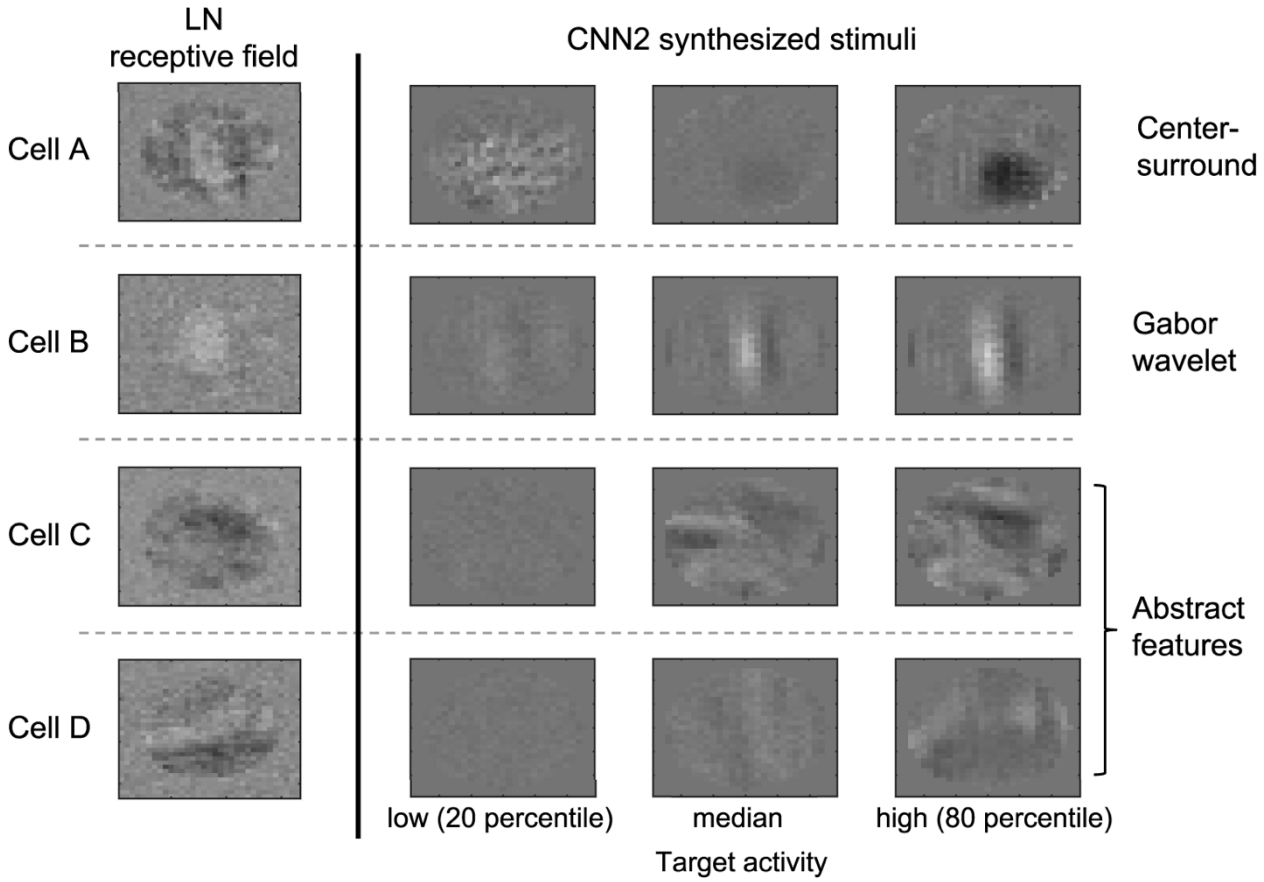

Figure 5.6 Using the network model to reveal the visual features that drive individual neurons. (Left) Receptive-field filters from the LN model for four neurons. (Right) For each neuron, we synthesized images that drove the predicted firing rates to the specified target values using the convolutional-neural-network model. These target firing rates were chosen to be different percentiles of the neuron's firing-rate distribution. Cells A and B appear to respond to localized image features, whereas cells C and D respond to more abstract image features.

CHAPTER VI

SINGLE NEURONS TO BRAIN-WIDE STATES

A persistent challenge in neuroscience is to bridge the gap between the complex tasks we know brains can perform and the physical components (neurons) that enable them. In vision, this divide is particularly wide, and much effort has been devoted to understanding how our brain processes visual information. For instance, we know the visual cortex receives complex spatiotemporal patterns of light relayed by the retina and reformats these patterns to infer what caused them (i.e. the identity of the object present) (DiCarlo et al., 2012). Answering this question requires first understanding how visual information is encoded at each sequential stage of processing along brain areas in visual cortex. Computational modeling has much to offer neuroscience in addressing this knowledge gap. One approach is to build computational models to replicate the neural encoding which takes place when the brain receives sensory stimuli.

Modeling Neural Encoding with ANN's

Predicting Single Units

At the most granular level, information is encoded in the spiking activities of individual neurons. Traditional systems neuroscience approaches have advanced our understanding of neural encoding by providing explanations for what individual neurons compute. In visual systems neuroscience, Hubel and Wiesel performed the seminal work in the field showing that individual simple and complex cells in primary visual cortex (V1) "tuned" to respond to oriented edges. This approach has worked well for

cells that respond well to simple stimuli, but the encoding properties of many other cells in V1 are still unexplained (Olshausen et al., 2001).

We demonstrate ANN's trained with machine learning techniques are a robust model of neural encoding in V1 capable of accurately predicting single neuron responses to natural stimuli. Importantly, this model achieves equally good predictability for both orientation selective and non-orientation selective cells for natural image stimuli. We show this approach is a useful tool for studying responses properties of previously difficult to study cells in V1.

Objectively Useful

David Marr proposed the idea that understanding the computational goals of visual processing is equally important and complementary to an understanding of the parts (e.g. individual neurons) that physically implement it. For early visual areas, efficient coding as a computational goal has been successful at explaining response properties of cells in primary visual cortex (V1) but has not worked as well at explaining responses in higher visual areas. The prevailing view for higher visual areas like inferior temporal cortex (IT) is that object recognition best describes its objective but directly testing this hypothesis is difficult.

Questions of this nature are fundamentally challenging to test, especially when limited to only analyzing responses for a handful of neurons. Recent results have demonstrated ANN's may be better suited for evaluating higher visual area objectives. Deep convolutional neural networks (DCNN's) trained to categorize objects in images also develop internal representations which also match IT responses to natural images.

It has been posited that matching representations could only arise if both the model and ventral stream are optimized for the same objective. We tested this explanation by optimizing models for both image categorization and a composite categorize and reconstruct objective. We find models which optimize the composite objective have representations which match IT better than representations formed from categorize alone. This is surprising, if strictly object recognition best describes the objective of visual processing in the ventral stream, optimizing an alternate objective should develop more poorly matching representations. However, this finding may help reconcile two observations at odds with the strictly object recognition hypothesis. First, it's been shown that visual processing areas show matching activation patterns in response to both viewing a scene and mentally visualizing the same scene (Freud et al., 2016; O'Craven and Kanwisher, 2000; Sereno and Lehky, 2011; Stokes et al., 2009). Second, ventral stream areas explicitly retain information not useful for object recognition (Hong et al., 2016).

Half of nonhuman primate neocortex is devoted to visual processing (Felleman and Van Essen, 1991), underscoring both its complexity and evolutionary importance. Furthermore, the 'No free lunch theorems' demonstrate objective function choice is not arbitrary; no learning algorithm performs well on all tasks. These pressures dictate an alternate objective choice, should have a compelling advantage. Our work suggests that an advantage such as noise robustness might explain why the alternate categorize and reconstruct objective provides a better match to IT representations.

Deep Brain Stimulation

Current neurostimulators are non-adaptive, limited by lack of robust methods for reading out a patient's brain state necessary for adaptive neurostimulation. This work has also shown the promise of ANN's as useful tool decoding information from neural activity. In chapter 3 we described our efforts to build such a model capable of detecting sleep stage from local field potential (LFP) recordings taken from DBS electrodes implanted in the subthalamic nucleus (STN). In this work we trained an ANN classifier model to predict the sleep state of PD patients from spectral decomposition features in their local field potentials.

Looking forward

In this work we leverage ANN models as a powerful tool to improve our understanding of neural encoding, predict brain-wide states, replicate population response properties in IT, and predict individual neuron responses to stimuli. We demonstrate that the objective best describing higher visual areas may be more complex than solely object recognition and show why this may be important for practical reasons. While these advances take important steps forward in modeling neural encoding, our model performance is likely limited by several factors.

Current shortcomings

Model input medium

More accurate models of cortical visual processing can likely be achieved by training models using media more aligned with natural vision. For instance, the brain's

visual processing system has been evolutionarily optimized to operate on sequences of images (e.g. video) and not just a single snapshot. Motion signals, which are present changes in object location frame by frame, convey import information used by the visual system. The models described in this work do not take advantage of extra information present in the time domain of video. Future models will likely take advantage of this kind of information.

Attention

One of the important uses of motion signals is to help direct our attention to focus on parts of visual field that is more important than others. Attention mechanisms allow our visual system to use its visual processing resources more efficiently. Attention layers have been used successfully in both machine translation and some classification networks, yet our current models do not have an attention component. Future regression and classification models which attempt to predict neural encoding will likely benefit from the addition of attention as a component of the model.

Interaction

Another key aspect of the development of visual representations is interaction. While little definitive evidence in human brains exists, it is highly likely that our visual representations are influenced by our ability to interact with our environment as agents. Developing training environments in which models can move and interact in 3D rendered environments as agents when learning object recognition tasks will likely improve model accuracy and training efficiency. Models which incorporate agency typically also intersect reinforcement learning, another important field of machine learning, which we did not cover in our work.

Future Challenges

Stimulation resolution

Beyond the models themselves, ideal implantable neurostimulators still have several challenges which need to be addressed for widespread use in human patients. One of these challenges is the current resolution limits of implantable neurostimulators. Even with a perfect model for predicting individual neuronal responses to visual stimuli, large-scale devices with stimulation resolutions down to single neurons are still not widely available. Optogenetic methods have shown promise for devices with single neuron resolution (Ozbay et al., 2015). However, optogenetic devices face even more regulatory hurdles than traditional implantable devices due to the gene therapy components necessary to deliver the photosensitive ion channels to neurons. In addition to challenges intrinsic to the interface hardware, another issue facing practical cortical prosthetics is the variability in neural encoding across individuals.

Individual variability

Clinical use in human patients will require ways to tune models to an individual's specific encoding, which is currently missing when we train models on a dataset containing recordings from 1-2 subjects. We are optimistic that overcoming issues of individual variability is achievable. Our work on predicting individual brain states in human LFP patterns shows that for some domains of neuronal encoding, model generalization across individual patients is possible. These results still require clinical validation on a novel cohort of patients. This validation study could also include a component to determine optimal stimulation settings for each state. For instance, the

stimulation parameter space could be sampled during a sleep study to identify parameters that might influence sleep state transitions. This could have important implications due to the fact that PD patients often transition into REM sleep states less often than healthy patients.

REFERENCES

Abosch, A., Lanctin, D., Onaran, I., Eberly, L., Spaniol, M., Ince, N.F., 2012. Long-term recordings of local field potentials from implanted deep brain stimulation electrodes. Neurosurgery 71, 804–814. doi:10.1227/NEU.0b013e3182676b91

Arnulf, I., Bejjani, B.P., Garma, L., Bonnet, A.M., Houeto, J.L., Damier, P., Derenne, J.P., Agid, Y., 2000. Improvement of sleep architecture in PD with subthalamic nucleus stimulation. Neurology 55, 1732–1734. doi:10.1212/wnl.55.11.1732

Atick, J.J., Redlich, A.N., 1992. What Does the Retina Know about Natural Scenes? Neural Computation 4, 196–210. doi:10.1162/neco.1992.4.2.196

Barlow, H., 2001. Redundancy reduction revisited. Network 12, 241–253.

Barlow, H.B., 1961. The coding of sensory messages. Current Problems in Animal Behavior 331–360.

Bergstra, J., Bengio, Y., 2012. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research 13, 281–305.

Brager, A.J., Yang, T., Ehlen, J.C., Simon, R.P., Meller, R., Paul, K.N., 2016. Sleep Is Critical for Remote Preconditioning-Induced Neuroprotection. Sleep 39, 2033–2040. doi:10.5665/sleep.6238

Bronstein, J.M., Tagliati, M., Alterman, R.L., Lozano, A.M., Volkmann, J., Stefani, A., Horak, F.B., Okun, M.S., Foote, K.D., Krack, P., Pahwa, R., Henderson, J.M., Hariz, M.I., Bakay, R.A., Rezai, A., Marks, W.J., Moro, E., Vitek, J.L., Weaver, F.M., Gross, R.E., DeLong, M.R., 2011. Deep brain stimulation for Parkinson disease: an expert consensus and review of key issues., in:. Presented at the Archives of neurology, American Medical Association, pp. 165–165. doi:10.1001/archneurol.2010.260

Cadena, S.A., Denfield, G.H., Walker, E.Y., Gatys, L.A., Tolias, A.S., Bethge, M., Ecker, A.S., 2018. Deep convolutional models improve predictions of macaque V1 responses to natural images. bioRxiv 201764. doi:10.1101/201764

Cadieu, C.F., Hong, H., Yamins, D.L.K., Pinto, N., Ardila, D., Solomon, E.A., Majaj, N.J., DiCarlo, J.J., 2014. Deep neural networks rival the representation of primate IT cortex for core visual object recognition. PLoS Comput. Biol. 10, e1003963. doi:10.1371/journal.pcbi.1003963

Carandini, M., Heeger, D.J., 2011. Normalization as a canonical neural computation. Nat Rev Neurosci 13, 51–62. doi:10.1038/nrn3136

Chaudhuri, K.R., Healy, D.G., Schapira, A.H.V., National Institute for Clinical Excellence, 2006. Non-motor symptoms of Parkinson's disease: diagnosis and management. Lancet Neurol 5, 235–245. doi:10.1016/S1474-4422(06)70373-8

Chen, Y., Crawford, J.D., 2019. Allocentric representations for target memory and reaching in human cortex. Annals of the New York Academy of Sciences 46, 774. doi:10.1111/nyas.14261

Coen-Cagli, R., Kohn, A., Schwartz, O., 2015. Flexible gating of contextual influences in natural vision. Nat. Neurosci. 18, 1648–1655. doi:10.1038/nn.4128

Cohen, J., 1960. A Coefficient of Agreement for Nominal Scales Educational and Psychological Measurement, vol. 20.

Collins, P.Y., Patel, V., Joestl, S.S., March, D., Insel, T.R., Daar, A.S., Scientific Advisory Board and the Executive Committee of the Grand Challenges on Global Mental Health, Anderson, W., Dhansay, M.A., Phillips, A., Shurin, S., Walport, M., Ewart, W., Savill, S.J., Bordin, I.A., Costello, E.J., Durkin, M., Fairburn, C., Glass, R.I., Hall, W., Huang, Y., Hyman, S.E., Jamison, K., Kaaya, S., Kapur, S., Kleinman, A., Ogunniyi, A., Otero-Ojeda, A., Poo, M.-M., Ravindranath, V., Sahakian, B.J., Saxena, S., Singer, P.A., Stein, D.J., 2011. Grand challenges in global mental health. Nature 475, 27–30. doi:10.1038/475027a

Dan, Y., Atick, J.J., Reid, R.C., 1996. Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. Journal of Neuroscience 16, 3351–3362.

David, S.V., Vinje, W.E., Gallant, J.L., 2004. Natural Stimulus Statistics Alter the Receptive Field Structure of V1 Neurons. Journal of Neuroscience 24, 6991–7006. doi:10.1523/JNEUROSCI.1422-04.2004

De Cock, V.C., Debs, R., Oudiette, D., Leu, S., Radji, F., Tiberge, M., Yu, H., Bayard, S., Roze, E., Vidailhet, M., Dauvilliers, Y., Rascol, O., Arnulf, I., 2011. The improvement of movement and speech during rapid eye movement sleep behaviour disorder in multiple system atrophy. Brain 134, 856–862. doi:10.1093/brain/awq379

Del Rio, T., Feller, M.B., 2006. Early retinal activity and visual circuit development. Neuron 52, 221–222. doi:10.1016/j.neuron.2006.10.001

DiCarlo, J.J., Zoccolan, D., Rust, N.C., 2012. How does the brain solve visual object recognition? Neuron 73, 415–434. doi:10.1016/j.neuron.2012.01.010

Felleman, D.J., Van Essen, D.C., 1991. Distributed hierarchical processing in the primate cerebral cortex. Cereb. Cortex 1, 1–47.

Field, D., 1987. Relations between the statistics of natural images and the response properties of cortical cells. Journal of Optical Society of America.

Freud, E., Plaut, D.C., Behrmann, M., 2016. “What” Is Happening in the Dorsal Visual Pathway. Trends in Cognitive Sciences 20, 773–784. doi:10.1016/j.tics.2016.08.003

Geirhos, R., Temme, C.R.M., Rauber, J., Schuett, H.H., Bethge, M., Wichmann, F.A., 2018. Generalisation in humans and deep neural networks.

Giuditta, A., Ambrosini, M.V., Montagnese, P., Mandile, P., Cotugno, M., Grassi Zucconi, G., Vescia, S., 1995. The sequential hypothesis of the function of sleep. Behav. Brain Res. 69, 157–166. doi:10.1016/0166-4328(95)00012-i

Güçlü, U., van Gerven, M.A.J., 2015. Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream. J. Neurosci. 35, 10005–10014. doi:10.1523/JNEUROSCI.5023-14.2015

Hahnloser, R.H.R., Sarpeshkar, R., Mahowald, M.A., Douglas, R.J., Seung, H.S., 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature 405, 947–951. doi:10.1038/35016072

Hamani, C., Saint-Cyr, J.A., Fraser, J., Kaplitt, M., Lozano, A.M., 2004. The subthalamic nucleus in the context of movement disorders. Brain 127, 4–20. doi:10.1093/brain/awh029

Hariz, M.I., Rehncrona, S., Quinn, N.P., Speelman, J.D., Wensing, C., Multicentre Advanced Parkinson's Disease Deep Brain Stimulation Group, 2008. Multicenter study on deep brain stimulation in Parkinson's disease: an independent assessment of reported adverse events at 4 years. Mov. Disord. 23, 416–421. doi:10.1002/mds.21888

Hassan, B.A., Hiesinger, P.R., 2015. Beyond Molecular Codes: Simple Rules to Wire Complex Brains. Cell 163, 285–291. doi:10.1016/j.cell.2015.09.031

Holtzheimer, P.E., Mayberg, H.S., 2011. Deep brain stimulation for psychiatric disorders. Annu. Rev. Neurosci. 34, 289–307. doi:10.1146/annurev-neuro-061010-113638

Hong, H., Yamins, D.L.K., Majaj, N.J., DiCarlo, J.J., 2016. Explicit information for category-orthogonal object properties increases along the ventral stream. Nat. Neurosci. 19, 613–622. doi:10.1038/nn.4247

Hubel, D.H., Wiesel, T.N., 1959. Receptive fields of single neurones in the cat's striate cortex. The Journal of Physiology 148, 574–591.

Iber, C., Ancoli-Israel, S., Chesson, A., Quan, S., 2007. The AASM manual for the scoring of sleep and associates events: Rules, Terminology, and technical specifications. Westchester, IL: American Academy of Sleep Medicine.

Ince, N.F., Gupte, A., Wichmann, T., Ashe, J., Henry, T., Bebler, M., Eberly, L., Abosch, A., 2010. Selection of optimal programming contacts based on local field potential recordings from subthalamic nucleus in patients with Parkinson's disease. Neurosurgery 67, 390–397. doi:10.1227/01.NEU.0000372091.64824.63

Iranzo, A., Valldeoriola, F., Santamaría, J., Tolosa, E., Rumià, J., 2002. Sleep symptoms and polysomnographic architecture in advanced Parkinson's disease after chronic bilateral subthalamic stimulation. J Neurol Neurosurg Psychiatry 72, 661–664. doi:10.1136/jnnp.72.5.661

Katz, L.C., Shatz, C.J., 1996. Synaptic activity and the construction of cortical circuits. Science 274, 1133–1138. doi:10.1126/science.274.5290.1133

Krizhevsky, A., Hinton, G., 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A., Nair, V., Hinton, G.E., n.d. CIFAR10.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks, in:. Presented at the Advances in Neural Information Processing Systems NIPS, pp. 1097–1105.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. doi:10.1038/nature14539

Lee, H., Ekanadham, C., Ng, A.Y., 2008. Sparse deep belief net model for visual area V2 873–880.

Lehky, S.R., Sejnowski, T.J., Desimone, R., 1992. Predicting responses of nonlinear neurons in monkey striate cortex to complex patterns. Journal of Neuroscience 12, 3568–3581. doi:10.1523/JNEUROSCI.12-09-03568.1992

Levy, W.B., Baxter, R.A., 1996. Energy efficient neural codes. Neural Computation 8, 531–543. doi:10.1162/neco.1996.8.3.531

Li, Q., Hu, J., Ding, J., Zheng, G., 2014. Fisher's method of combining dependent statistics using generalizations of the gamma distribution with applications to genetic pleiotropic associations. Biostatistics 15, 284–295. doi:10.1093/biostatistics/kxt045

Lorach, H., Marre, O., Sahel, J.-A., Benosman, R., Picaud, S., 2013. Neural stimulation for visual rehabilitation: advances and challenges. J. Physiol. Paris 107, 421–431. doi:10.1016/j.jphysparis.2012.10.003

Lucke-Wold, B.P., Smith, K.E., Nguyen, L., Turner, R.C., Logsdon, A.F., Jackson, G.J., Huber, J.D., Rosen, C.L., Miller, D.B., 2015. Sleep disruption and the sequelae associated with traumatic brain injury. Neurosci Biobehav Rev 55, 68–77. doi:10.1016/j.neubiorev.2015.04.010

Mahendran, A., Vedaldi, A., 2015. Understanding deep image representations by inverting them, in:. Presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 5188–5196. doi:10.1109/CVPR.2015.7299155

Majaj, N.J., Hong, H., Solomon, E.A., DiCarlo, J.J., 2015. Simple Learned Weighted Sums of Inferior Temporal Neuronal Firing Rates Accurately Predict Human Core Object Recognition Performance. J. Neurosci. 35, 13402–13418. doi:10.1523/JNEUROSCI.5181-14.2015

Mazurek, M., Kager, M., Van Hooser, S.D., 2014. Robust quantification of orientation selectivity and direction selectivity. Front Neural Circuits 8, 92. doi:10.3389/fncir.2014.00092

McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity., Bulletin of mathematical biology.

McIntosh, L.T., Maheswaranathan, N., Nayebi, A., Ganguli, S., Baccus, S.A., 2016. Deep Learning Models of the Retinal Response to Natural Scenes. arXiv 29, 1369–1377.

Montijn, J.S., Meijer, G.T., Lansink, C.S., Pennartz, C.M.A., 2016. Population-Level Neural Codes Are Robust to Single-Neuron Variability from a Multidimensional Coding Perspective. Cell Rep 16, 2486–2498. doi:10.1016/j.celrep.2016.07.065

Nili, H., Wingfield, C., Walther, A., Su, L., Marslen-Wilson, W., Kriegeskorte, N., 2014. A Toolbox for Representational Similarity Analysis. PLoS Comput. Biol. 10, e1003553. doi:10.1371/journal.pcbi.1003553

O'Craven, K.M., Kanwisher, N., 2006. Mental Imagery of Faces and Places Activates Corresponding Stimulus-Specific Brain Regions. http://dx.doi.org/10.1162/08989290051137549 12, 1013–1023. doi:10.1162/08989290051137549

O'Craven, K.M., Kanwisher, N., 2000. Mental imagery of faces and places activates corresponding stiimulus-specific brain regions. J Cogn Neurosci 12, 1013–1023.

Olshausen, B.A., Field, D.J., 2005. How Close Are We to Understanding V1? Neural Computation 17, 1665–1699. doi:10.1162/0899766054026639

Olshausen, B.A., Field, D.J., 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381, 607–609. doi:10.1038/381607a0

Olshausen, B.A., Sallee, P., Lewicki, M., 2001. Learning sparse image codes using a wavelet pyramid architecture. papers.nips.cc

papers.nips.cc.

Ozbay, B.N., Losacco, J.T., Cormack, R., Weir, R., Bright, V.M., Gopinath, J.T., Restrepo, D., Gibson, E.A., 2015. Miniaturized fiber-coupled confocal fluorescence microscope with an electrowetting variable focus lens using no moving parts. Optics Letters 40, 2553–2556. doi:10.1364/OL.40.002553

Pace-Schott, E.F., Hobson, J.A., 2002. The neurobiology of sleep: genetics, cellular physiology and subcortical networks. Nat Rev Neurosci 3, 591–605. doi:10.1038/nrn895

Paninski, L., Pillow, J.W., Simoncelli, E.P., 2004. Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. Neural Computation 16, 2533–2561. doi:10.1162/0899766042321797

Perlmutter, J.S., Mink, J.W., 2006. Deep brain stimulation. Annu. Rev. Neurosci. 29, 229–257. doi:10.1146/annurev.neuro.29.051605.112824

Pietro Berkes, Ben White, Fiser, J., 2009. No evidence for active sparsification in the visual cortex 108–116.

Pietro Berkes, Orbán, G., Lengyel, M., Fiser, J., 2011. Spontaneous Cortical Activity Reveals Hallmarks of an Optimal Internal Model of the Environment. Science 331, 83–87. doi:10.1126/science.1195870

Pillow, J.W., Shlens, J., Paninski, L., Sher, A., Litke, A.M., Chichilnisky, E.J., Simoncelli, E.P., 2008. Spatio-temporal correlations and visual signalling in a complete neuronal population. Nature 454, 995–999. doi:10.1038/nature07140

Postuma, R.B., Berg, D., Stern, M., Poewe, W., Olanow, C.W., Oertel, W., Obeso, J., Marek, K., Litvan, I., Lang, A.E., Halliday, G., Goetz, C.G., Gasser, T., Dubois, B., Chan, P., Bloem, B.R., Adler, C.H., Deuschl, G., 2015. MDS clinical diagnostic criteria for Parkinson's disease. Mov. Disord. 30, 1591–1601. doi:10.1002/mds.26424

Qian, N., 1999. On the momentum term in gradient descent learning algorithms. Neural Netw 12, 145–151. doi:10.1016/s0893-6080(98)00116-6

Richards, B.A., Lillicrap, T.P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R.P., de Berker, A., Ganguli, S., Gillon, C.J., Hafner, D., Kepecs, A., Kriegeskorte, N., Latham, P., Lindsay, G.W., Miller, K.D., Naud, R., Pack, C.C., Poirazi, P., Roelfsema, P., Sacramento, J., Saxe, A., Scellier, B., Schapiro, A.C., Senn, W., Wayne, G., Yamins, D., Zenke, F., Zylberberg, J., Therien, D., Kording, K.P., 2019. A deep learning framework for neuroscience. Nat. Neurosci. 22, 1761–1770. doi:10.1038/s41593-019-0520-2

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536. doi:10.1038/323533a0

Schoppe, O., Harper, N.S., Willmore, B.D.B., King, A.J., Schnupp, J.W.H., 2016. Measuring the Performance of Neural Models. Front Comput Neurosci 10, 1929. doi:10.3389/fncom.2016.00010

Sereno, A.B., Lehky, S.R., 2011. Population coding of visual space: comparison of spatial representations in dorsal and ventral pathways. Front Comput Neurosci 4, 159. doi:10.3389/fncom.2010.00159

Sharma, V.D., Sengupta, S., Chitnis, S., Amara, A.W., 2018. Deep Brain Stimulation and Sleep-Wake Disturbances in Parkinson Disease: A Review. Front Neurol 9, 697. doi:10.3389/fneur.2018.00697

Simonyan, K., Zisserman, A., n.d. Very deep convolutional networks for large-scale image recognition. arxiv.org

arXiv:1409.1556.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15, 1929–1958.

Stokes, M., Thompson, R., Cusack, R., Duncan, J., 2009. Top-down activation of shape-specific population codes in visual cortex during mental imagery. J. Neurosci. 29, 1565–1572. doi:10.1523/JNEUROSCI.4657-08.2009

Tekriwal, A., Kern, D.S., Tsai, J., Ince, N.F., Wu, J., Thompson, J.A., Abosch, A., 2017. REM sleep behaviour disorder: prodromal and mechanistic insights for Parkinson's disease. J Neurol Neurosurg Psychiatry 88, 445–451. doi:10.1136/jnnp-2016-314471

the, R.S.P.O.T.E.A.C.O., 1986, n.d. Two problems with back propagation and other steepest descent learning procedures for networks, ci.nii.ac.jp

.

Thompson, J.A., Tekriwal, A., Felsen, G., Ozturk, M., Telkes, I., Wu, J., Ince, N.F., Abosch, A., 2017. Sleep patterns in Parkinson's disease: direct recordings from the subthalamic nucleus. J Neurol Neurosurg Psychiatry 89, jnnp–2017–316115–104. doi:10.1136/jnnp-2017-316115

Vintch, B., Movshon, J.A., Simoncelli, E.P., 2015. A Convolutional Subunit Model for Neuronal Responses in Macaque V1. J. Neurosci. 35, 14829–14841. doi:10.1523/JNEUROSCI.2815-13.2015

Willmore, B., Prenger, R.J., Wu, M.C.K., Gallant, J.L., 2008. The berkeley wavelet transform: a biologically inspired orthogonal wavelet transform. Neural Computation 20, 1537–1564. doi:10.1162/neco.2007.05-07-513

Willmore, B.D.B., Mazer, J.A., Gallant, J.L., 2011. Sparse coding in striate and extrastriate visual cortex. Journal of Neurophysiology 105, 2907–2919. doi:10.1152/jn.00594.2010

Xiao, H., Rasul, K., Vollgraf, R., 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.

Yamins, D.L.K., DiCarlo, J.J., 2016. Using goal-driven deep learning models to understand sensory cortex. Nat. Neurosci. 19, 356–365. doi:10.1038/nn.4244

Yamins, D.L.K., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., DiCarlo, J.J., 2014. Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proc. Natl. Acad. Sci. U.S.A. 111, 8619–8624. doi:10.1073/pnas.1403112111

Zador, A.M., 2019. A Critique of Pure Learning: What Artificial Neural Networks can Learn from Animal Brains. bioRxiv 29, 582643. doi:10.1101/582643

Ziemba, C.M., Freeman, J., Movshon, J.A., Simoncelli, E.P., 2016. Selectivity and tolerance for visual texture in macaque V2. Proceedings of the National Academy of Sciences 113, E3140–E3149. doi:10.1073/pnas.1510847113

Zylberberg, J., DeWeese, M.R., 2013. Sparse Coding Models Can Exhibit Decreasing Sparseness while Learning Sparse Codes for Natural Images. PLoS Comput. Biol. 9, 1–10. doi:10.1371/journal.pcbi.1003182

Zylberberg, J., Murphy, J.T., DeWeese, M.R., 2011. A Sparse Coding Model with Synaptically Local Plasticity and Spiking Neurons Can Account for the Diverse Shapes of V1 Simple Cell Receptive Fields. PLoS Comput. Biol. 7, 1–12. doi:10.1371/journal.pcbi.1002250