FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Bc. David Nepožitek

Characterizing computations in a model of biological vision using deep-neural-network approaches

Department of Software and Computer Science Education

Supervisor of the master thesis:	Mgr. Ján Antolík, Ph.D.
Study programme:	Computer Science
Study branch:	Artificial Intelligence

Prague 2023

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In . . . . date . . . .

Author’s signature

I would like to thank my supervisor, Mgr. Ján Antolík, Ph.D., for the opportunity to work on this captivating topic and for his continuous guidance throughout the course of this thesis. I am also indebted to Luca Baroni, M.Sc., for his invaluable guidance and insightful suggestions, which have left a profound positive impact on this thesis. Last but not least, I greatly appreciate my family and friends for their support during my academic journey.

Title: Characterizing computations in a model of biological vision using deep-neural-network approaches

Author: Bc. David Nepožitek

Department: Department of Software and Computer Science Education

Supervisor: Mgr. Ján Antolík, Ph.D., Department of Software and Computer Science Education

Abstract: In this thesis, we examine two kinds of models of the primary visual cortex: a deep neural network for system identification and a spiking model of a cat's primary visual cortex. Further progress in modelling visual systems can help us comprehend the brain's inner workings in greater detail; moreover, it can help to develop better visual prosthesis or further improve models that handle visual inputs, such as those used for object classification. We employ the state-of-the-art deep neural network to predict the responses of the spiking model when presented with natural stimuli. We demonstrate that by tuning the hyperparameters, the deep neural network explains approximately 85% of the explainable variance observed in the responses of the spiking model. That is significantly more accurate than predictions of real neural responses, suggesting that real neurons possess certain characteristics not captured in the spiking model. However, we also argue that the network would not be capable of perfect predictions even when a large amount of data is provided. We show that the network encounters notable difficulties in modelling neurons with high noise and precisely predicting high firing rates. Furthermore, we analyse the network's representations by phase, orientation and size tuning. We illustrate that the modelled receptive fields of most layer IV neurons exhibit orientation and phase selectivity. Layer II/III neurons demonstrate orientation selectivity and more varying levels of phase invariance. This observation suggests the predominance of simple cells in layers IV and the presence of complex cells within layers II/III. A small number of neurons exhibit observable surround suppression. However, the neural network has difficulty accurately capturing the precise characteristics of size tuning.

Keywords: computational neuroscience, system identification, deep learning, visual cortex, spiking neural network

Contents

Introduction

Despite the constant advances in neuroscience, a significant portion of the brain's intricate workings remain not understood. This also applies to the brain's visual processing part, which is this thesis's target. One of the essential milestones is the ability to accurately predict responses of the relevant neurons based on given visual stimuli (Carandini et al., 2005). This problem is addressed by *system identification*. Further progress in this field might benefit our understanding of the visual system, help to develop better visual prosthesis, or further improve models that handle visual inputs, such as those used for object recognition.

Therefore, scientists are trying to develop various models of visual systems that aim to predict neural responses accurately. There are several ways how we can approach this task. One of the current prevailing methods is to use a convolutional neural network (CNN) as a so-called core module to extract features from visual stimuli (images). And then, these features are processed by a readout module which predicts the firing rates of individual neurons. These models are trained using pairs of images and responses of the appropriate neurons. This data can be obtained by presenting the images to an animal and recording its neural responses. Such models can achieve prominent results. However, one of the primary goals is to gain insights into the functioning of real neurons. Given that the difficulty to interpret the computations of a neural network, it can be challenging to fulfil this goal with such a model.

There are various other approaches to attaining knowledge about the visual system apart from system identification. For example, we can imitate the system based on our knowledge of neuroanatomy and biophysics and then examine the model's properties. An implementation of this idea is building a so-called spiking neural network (SNN). This network is made of spiking models of neurons designed to mimic an actual neuron's state. Specifically, they mathematically describe the changes in the membrane potential of a neuron and predict the time of spikes of the neurons based on their inputs. The topology and parameters of this network are decided upon the current knowledge of the brain. Compared to deep neural networks (DNNs), the computations of SNNs are much more comparable to the ones of a real brain.

The core idea of the thesis is to use these two types of models in unison in order to examine the capabilities and computations of both of them. Specifically, we are solving the system identification task using the state-of-the-art DNN from Lurz et al. (2020). The authors demonstrated a prominent performance of this network when used for the identification of a mouse's primary visual cortex. However, we utilise the network to predict the responses generated by a spiking model of a cat's primary visual cortex created by Antolík et al. (2018). This model comprises layer IV and layers II/III neurons of a cat's primary visual cortex. Our motivation is to exploit the advantages of using this model compared to a live animal. For instance, we can generate significantly greater amounts of training data, and we understand the design of the SNN completely, contrary to a brain.

The goal is to analyse the behaviour of the DNN used for the identification of the spiking model and to gather insights about the spiking network itself. There are specific areas that we want to address in the thesis:

1. **How well can a neural network predict the responses of the spiking model?** Even the best system identification models are not able to achieve satisfactory performance when tested with natural visual stimuli. The state-of-the-art model for prediction of mouse V1 responses explains less than 50% of explainable variance of the neural responses (Lurz et al., 2020; Willeke et al., 2022). We want to evaluate this model on our artificial dataset. If the model achieves perfect accuracy, it hints that something is missing in the spiking model compared to the actual animal. On the other hand, if the model can not predict the responses well enough, the neural network is probably not capable of handling a phenomenon which Antolík et al. (2018) were able to replicate in their model. Thus, both results might lead to interesting findings. Furthermore, considering the origin of our data, the dataset can be significantly larger compared to typical datasets comprising recordings from live animals. That can help us assess whether the subpar performance is solely a result of data limitations.

2. **Is it possible to transfer the pre-trained CNN core between artificial and live-animal data?** The study conducted by Lurz et al. (2020) demonstrated that pre-training the CNN core of their model on data from multiple subjects (mice) resulted in accurate predictions of neural responses from a distinct subject. We want to examine whether this transfer is possible between the artificial data from the spiking model and data recorded on a real animal. However, it is important to acknowledge that the scope of this experiment is somewhat limited since the spiking model is based on the primary visual cortex of a cat, while the available real data is obtained from a mouse.

3. **What mechanics does the deep neural network capture?** The spiking neural network was purposefully designed and subsequently evaluated under diverse stimulation paradigms to guarantee the manifestation of multiple properties observed in visual neurons. Knowing these characteristics, we want to determine their presence in the representations of the deep neural network. In particular, we are interested in exploring the properties of receptive fields, including orientation selectivity and surround suppression, which apply to all model neurons. We also aim to investigate the characteristic of phase invariance, which applies primarily to layer II/III neurons.

Results and Contributions

Throughout the course of conducting this thesis, we performed numerous experiments intending to gain insights into the system identification deep neural network (Lurz et al., 2020) and the spiking model of a cat's primary visual cortex (Antolík et al., 2018). In this text, we present several interesting findings:

1. **We successfully identify approximately 85% of explainable variance of the artificial neural responses.** By extensive tuning of hyperparameters of the model from Lurz et al. (2020), we trained a model that achieves a fraction of explainable variance explained of roughly 0.85.

2. **We analysed several characteristics of neurons and images to assess what causes the subpar accuracy of some predictions made by the deep neural network.** Particularly, we identified two factors that correlate with poor predictions of the network: the amount of noise in the neuron's responses and the target firing rate of the specific dataset sample. Furthermore, we rule out several characteristics that are currently not influencing the performance of the model, such as the size and position of the receptive field, as well as the frequency spectrum of the stimuli.

3. **We describe the impact of the number of images and neurons used to train the system identification network.** We determined that higher counts of both images and neurons positively impact the model's performance. We note that the influence of images is more significant. However, solely augmenting the dataset would not lead to perfect predictions.

4. **We show that using a fixed CNN core pre-trained on mice subjects performs significantly better than employing a fixed randomly initialised core when used on the artificial dataset.** This result suggests

that the pre-training captured representations that are general enough to be beneficial in predictions of the artificial neuron responses. Surprisingly, when a core pre-trained on the artificial dataset was used for predictions of mouse responses, the accuracy was comparable to random weights of the core.

5. **We demonstrate that using a goal-driven core module pre-trained on the ImageNet object classification task leads to reasonably good predictions.** We evaluated EfficientNetV2-S (Tan and Le, 2021) and VGG16 (Simonyan and Zisserman, 2015) as core modules of the system identification network on the artificial dataset. These models achieve a fraction of explainable variance explained of 0.7 and 0.73, respectively.

6. **We present that the deep neural network captures several essential properties of neurons' receptive fields.** To tune the phase, orientation and size of the receptive fields modelled by the deep neural network, we use patches of sinusoidal gratings as inputs of the model. Using this approach, we show that the well-predicted neurons within layer IV exhibit orientation and phase selectivity. We demonstrate that layers II/III neurons exhibit orientation preference and varying degrees of phase invariance. Lastly, we argue that the size tuning properties are not captured accurately in the DNN.

Organization

This thesis begins by establishing the necessary background knowledge concerning the early visual system in Chapter 1. We describe the path of visual information from the eyes to the primary visual cortex. Furthermore, we describe the key characteristics of neurons in the primary visual cortex that are relevant to this thesis.

In chapter 2, we introduce the field of computational neuroscience, focusing mainly on system identification and biological models of neurons. We explain the classical system identification methods and compare them with the utilisation of deep neural networks, which is the approach employed in this thesis. Additionally, we explain the method called spike-triggered average, a technique used for characterising linear receptive fields, as we use it to analyse the neurons within our datasets. We finish the chapter with an introduction to biological models of neurons. We briefly describe the biophysics of neurons and explain the essentials of integrate-and-fire models that are the foundation of the spiking network used in the thesis.

Moving on to Chapter 3, we delve into the details of the two models we employ throughout the thesis: the deep neural network for system identification and the spiking model of a cat's primary visual cortex. In the last sections of the chapter, we provide descriptions of the two datasets we use.

Lastly, Chapter 4 is dedicated to all the analyses and experiments we conducted throughout the thesis. Firstly, we provide some insights into the dataset. Specifically, we analyse the noise in the neural responses and examine the neurons' linear receptive fields. In the next section, we present a set of experiments focused on the system identification of the spiking model. Moreover, we analyse the predictions made by the model, aiming to assess the factors contributing to subpar predictions. Next, we examine the representations captured by the trained deep neural network. Specifically, we explore whether the captured receptive fields possess some typical properties known to be present in the spiking model. We conclude the chapter with experiments involving transfers of the pre-trained core module across our two datasets.

Chapter 1

Early Visual System

Considering that the thesis examines models of a primary visual cortex, let us first establish the necessary biological background. This chapter serves as an introduction to the neuroanatomy of a mammalian early visual system. This area of research is extensive; thus, we only explain the essentials related to the thesis. For deeper insights and more detailed commentary, please refer to the introductory books to neuroscience from Bear, Connors, and Paradiso (2016) or Goebel, Muckli, and Kim (2012).

The simplistic objective of a visual system as a whole is to extract useful information from the visible light in the environment. The early visual system stands at the beginning of the process. Refer to Figure 1.1 for an illustration of its high-level anatomy. The whole system starts with the eyes. In the retina, the electromagnetic energy is transformed into neural activity. The information then flows via the *optic nerves* and through the *optic chiasm*. Two *optic tracts* emerge at the chiasm. The majority of optic tract axons terminate in the *lateral geniculate nucleus* (LGN) (Bear, Connors, and Paradiso, 2016, p. 355). The predominant output of LGN forms optic radiations that finally innervate the *primary visual cortex*. Now, we will provide a more detailed description of each part of the early visual system.

1.1 Eye

One of the key objectives of an eye is to collect electromagnetic radiation from the environment and project its image on the retina. When the light travels from the outside world, it first goes through the cornea and the aqueous humour, the fluid in front of a lens. Then it moves through the lens and the vitreous humour to eventually reach the retina. All of these elements ensure that the light is refracted correctly and projected on the retina. See Figure 1.4 for a simplified illustration

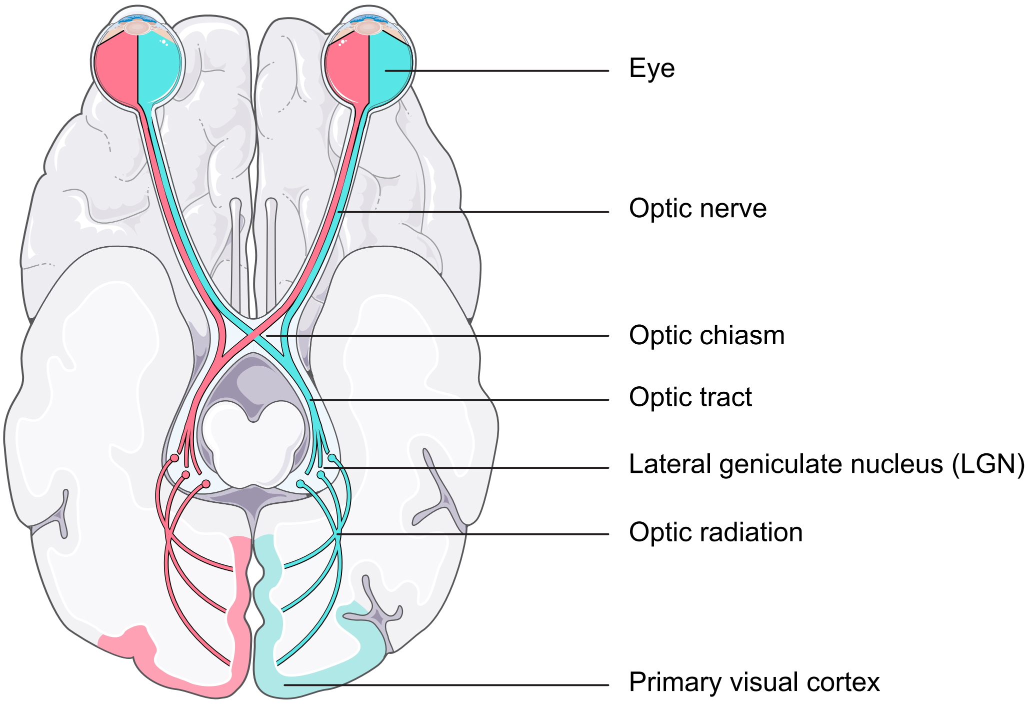


Figure 1.1 A diagram of a human's early visual system. Annotations were added, and colours were edited in the illustration "Optical pathways" from Servier Medical Art. Servier Medical Art by Servier is licensed under a Creative Commons Attribution 3.0 Unported License (https://creativecommons.org/licenses/by/3.0/).

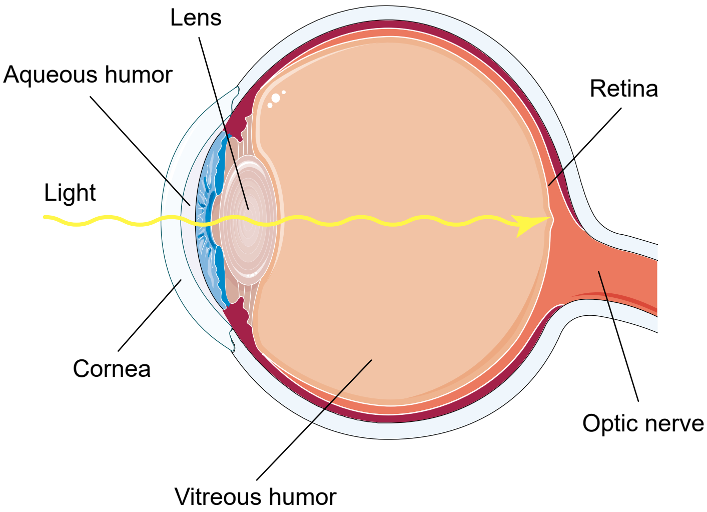


Figure 1.2 A diagram of an eye. The illustration is based on "Eye" from Servier Medical Art. Annotations and the light way were added. Servier Medical Art by Servier is licensed under a Creative Commons Attribution 3.0 Unported License (https://creativecommons.org/licenses/by/3.0/).

of an eye.

An eye's second vital contribution is converting electromagnetic energy into neural activity. This process is initiated when light reaches the retina's photoreceptors (rods and cones). The photoreceptors influence the membrane potential of the so-called bipolar and horizontal cells. The bipolar cells then connect to ganglia. Both bipolar and ganglia cells posses *centre-surround* receptive fields (Bear, Connors, and Paradiso, 2016, p. 321-324). That makes them react to differences of illumination within a small part of the visual field (see Figure 1.3). The ganglia axons form the sole neural output from the eyes.

1.2 The Way to Visual Cortex

As the ganglia axons leave the eyes, they form optic nerves. Eventually, the nerves from both eyes meet and intermingle at the optic chiasm, as illustrated in Figure 1.1. The information from the retinas' nasal parts crosses to the brain's contralateral part. We call this phenomenon a partial decussation. Two optic tracts emerge after the decussation. At this point, each tract carries only the information gathered at the ipsilateral side of the retinas. Hence, the tracts contain only the information from the contralateral visual hemifields (see Figure 1.4).

The optic tracts terminate mainly in the lateral geniculate nucleus (LGN)

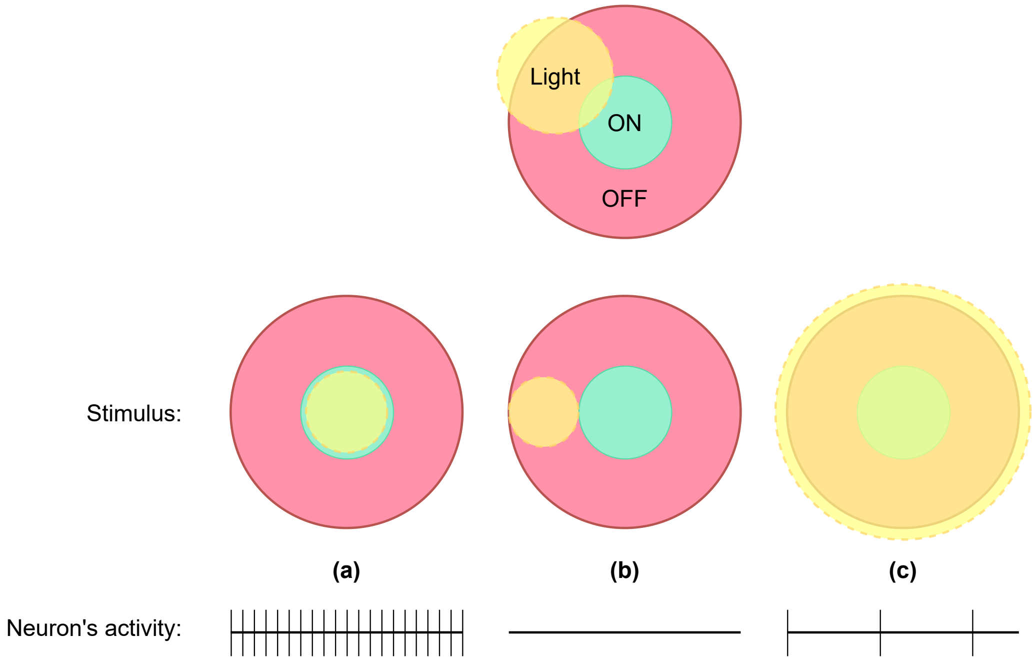


Figure 1.3 The centre-surround receptive field organisation. This illustration shows an on-centre variant of a centre-surround receptive field. The inner circle is an excitatory area, and the surrounding part acts as an inhibitory area. This type of receptive field is common for LGN or ganglia cells. **(a)** When we illuminate the ON region (the centre), the neuron is activated. **(b)** Symmetrically, when the light hits the surrounding part, the firing is inhibited. **(c)** We observe low or no activity when both areas are illuminated simultaneously.

in the thalamus. The processing characteristics of LGN neurons are mostly as straightforward as those of ganglia cells (Piscopo et al., 2013). Accordingly, LGN serves mainly as a relay of visual information from the retinas to the cortical areas. However, the majority of LGN input is actually formed by feedback from V1 and other parts of the brain (Guillery and Sherman, 2002; Bear, Connors, and Paradiso, 2016). Thanks to this backward pathway, it is possible to modulate the transfer of information according to the current attentional needs (Guillery and Sherman, 2002). The predominant receiver of the LGN axons is the primary visual cortex.

1.3 Primary Visual Cortex

The primary visual cortex corresponds to the Brodmann area 17, often called a striate cortex or shortened as V1. It is the first stage of a cortical visual pathway. Therefore, it acts as a gateway to the higher levels of processing that concern more abstract visual concepts (Ungerleider, 1994; Felleman and Van Essen, 1991).

Hubel and Wiesel (1977) describe two main functions of the primary visual cortex. First, it is known that the inputs of V1 neurons are arranged in a way that they respond to short, oriented line segments. Second, the striate cortex is the first area of the visual system where the information from both eyes is united.

1.3.1 Structure

Like other neocortical areas, the primary visual cortex is divided into six layers, usually labelled with Roman numerals. Layer I is the outermost, while Layer VI is the innermost layer. Layer IV is commonly further subdivided into four layers - IVA, IVB, IVCα, and IVCβ.

In this thesis, we will be primarily concerned with layers II/III and IV. Let us briefly describe their connectivity. The major part of LGN axons innervates the layer IVC (Lund, 1988). Layer IVC also receives recurrent input from layer VI Binzegger, Douglas, and Martin, 2004. The information from layer IV is then passed to layers II/III, either directly from IVCβ or via layer IVB. All layers II/III and IV contain horizontal connections between nearby neurons (Buzás et al., 2006). Additionally, layers II/III neurons also receive input from the intra-layer connections of neurons with similar receptive field properties (Ts'o, Gilbert, and Wiesel, 1986). Finally, layer IVB and layers II/III project their axons to other cortical areas, such as V2, V3, V4, and V5.

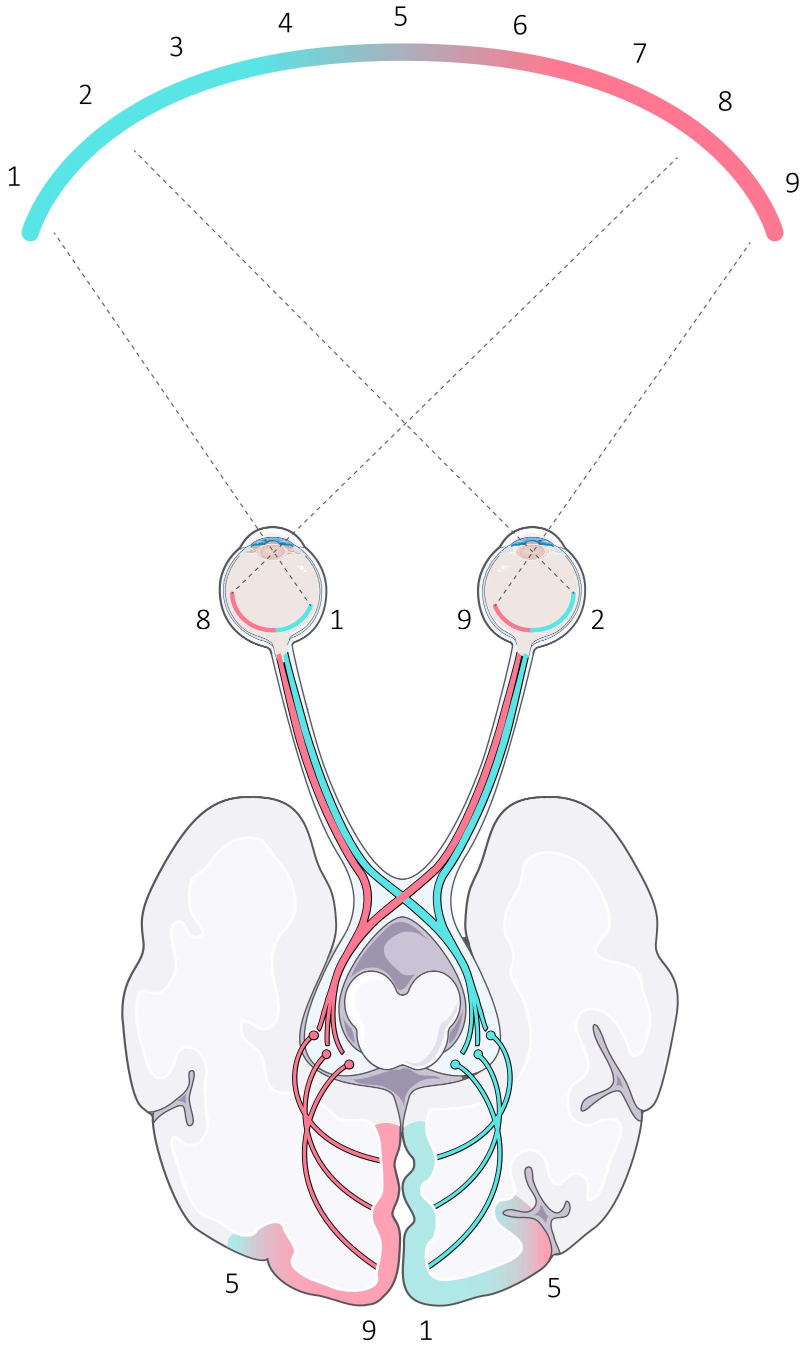


Figure 1.4 Illustration of retinotopy and visual fields. The illustration, in a very simplified way, shows how information from different parts of the visual field flows through the visual pathway. This figure is inspired by *Neuroscience* (Bear, Connors, and Paradiso, 2016, p. 343). The base illustration is "Optical pathways" from Servier Medical Art. Only relevant part of the brain was used, and retinotopy was illustrated. Servier Medical Art by Servier is licensed under a Creative Commons Attribution 3.0 Unported License (https://creativecommons.org/licenses/by/3.0/).

1.3.2 Retinotopy

It has been observed that V1 neurons are organised into several cortical maps (Hubel and Wiesel, 1977). That is, the neurons neighbouring in the horizontal plane of V1 show similarities in particular properties. One of the most essential cortical maps is called *retinotopy*. The retinotopic organisation of V1 neurons ensures that nearby cortical neurons process information from retinal cells that are also close to each other. Thus, a mapping exists between locations in the retina and areas of the primary visual cortex, as depicted in Figure 1.4 in a very simplified manner. Note that the mapping may be distorted, and the visual field is not represented uniformly in the cortex (Hubel and Wiesel, 1977). Specifically, neurons processing the central part of the receptive field occupy more area of the striate cortex than the ones handling the peripheral portion of the field (Hubel and Wiesel, 1977).

1.3.3 Receptive Field

As already suggested, the cells in the retina, LGN and V1 do not respond to the whole visual field but rather to a small portion. Hubel and Wiesel (1962) define *receptive field* as the regions (of the retina or visual field) that can modulate the firing rate of a specific neuron. In more general terms, we describe the receptive field as a set of conditions that can influence the neuron's output. For example, it is not enough to describe a receptive field of a V1 neuron only in the spatial dimensions as it is known that some neurons respond differently to various moving stimuli; therefore, a time dimension is needed as well (Hubel and Wiesel, 1977).

Some receptive fields may be straightforward, e.g. the ones of LGN neurons or ganglion cells in the retina. They both contain separate inhibitory (OFF) and excitatory (ON) regions in the form of concentric circles, as illustrated in Figure 1.3 (Hubel and Wiesel, 1961). Light concentrated on the ON region causes the neuron to fire. On the contrary, when the light hits the OFF area, the neuron responds by inhibiting the firing. Accordingly, when both regions of the receptive field are illuminated, the firing rate is very low, or there is no activity.

However, the receptive fields of the striate cortex neurons can be more complicated. For that reason, Hubel and Wiesel (1962) introduced a categorisation of V1 neurons into two categories based on the properties of their receptive field (Hubel and Wiesel, 1962). Specifically, they classified receptive fields as *simple* if they met all the following: "(1) they were subdivided into distinct excitatory and inhibitory regions; (2) there was summation within the separate excitatory and inhibitory parts; (3) there was antagonism between excitatory and inhibitory regions; and (4) it was possible to predict responses to stationary or moving spots of various shapes from a map of the excitatory and inhibitory areas." (Hubel and Wiesel, 1962). If the receptive field does not comply with these criteria, we term it as *complex*.

Orientation Preference

The receptive fields of the cells preceding the primary visual cortex exhibit circular symmetry. That is, if a line stimulus is presented at various orientations, the cells respond similarly. The neurons of the primary visual cortex act differently under these conditions as it has been observed that V1 neurons show a preference for orientation (also called orientation specificity) (Hubel and Wiesel, 1977). In order to produce a strong response of many V1 neurons, it is not sufficient to present a circular spot of light as a stimulus. They will produce a stronger response when a line segment of light is presented at a specific orientation (Hubel and Wiesel, 1977). Hubel and Wiesel (1977) observed that when the line segment is misoriented from the preferred direction by more than $10^\circ - 20^\circ$ the response of a neuron from macaque V1 significantly decreases. There is no prevailing preference for a particular orientation in the V1 neurons. Moreover, not all the primary visual cortex cells show orientation specificity.

Interestingly, the preference of orientation is also organised into a cortical map (Hubel and Wiesel, 1977). That is, the immediately neighbouring cells in the horizontal plane of a specific neuron usually show only a slight variation of the preferred orientation. By measuring the preference of many neurons across the horizontal plane, we can construct a map similar to the one in Figure 1.5. Moreover, the neurons of V1 are arranged into thin cortical columns that share the same orientation preference as depicted in Figure 1.6. To conclude, the preference stays for the neurons along the vertical axis but gradually changes along the horizontal one.

Surround Modulation

The responses of visual cortex neurons are not only influenced by the visual information directly in their receptive field. It has been demonstrated that a much broader context can impact the neural responses (Hubel and Wiesel, 1965; Nurminen and Angelucci, 2014; Angelucci et al., 2017). The phenomenon is called surround modulation.

In the context of the primary visual cortex, this effect mostly exhibits as a suppression of firing rate, for instance, when a stimulus is enlarged beyond the central area of the neuron's receptive field (Angelucci et al., 2017). This manifestation is referred to as surround suppression. It was shown that the strongest effect is achieved when the stimulus parameters in the centre of the

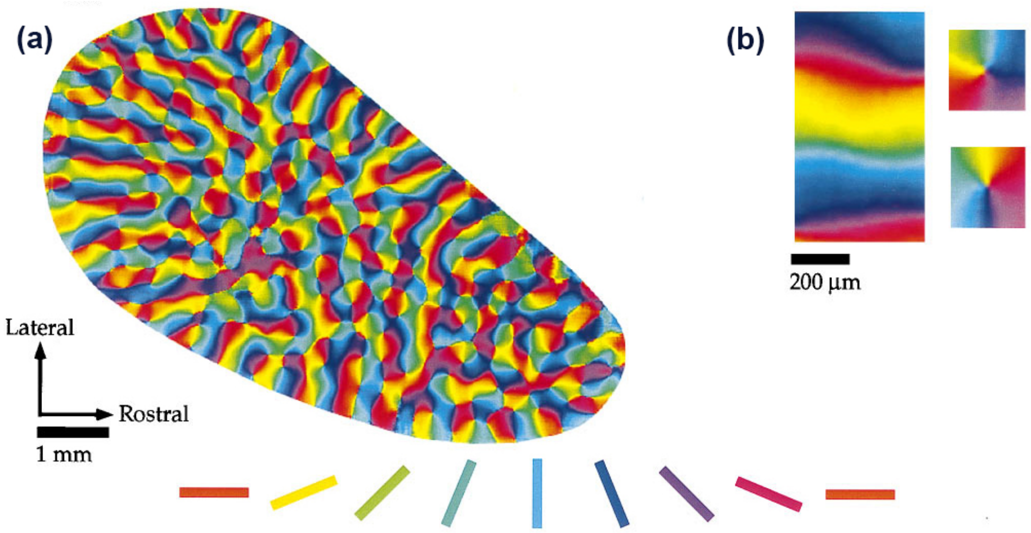


Figure 1.5 An orientation map of a primary visual cortex. This image shows an orientation-specificity map of neurons in a tree shrew's primary visual cortex. The preferred orientation is coded by colour according to the line segments at the bottom of the figure. **(a)** A map corresponding approximately to the striate cortex. **(b)** The common features of the map magnified. Adapted from "Orientation Selectivity and the Arrangement of Horizontal Connections in Tree Shrew Striate Cortex".

receptive field are similar to those in the surrounding area (Angelucci et al., 2017). For example, when drifting gratings of the same orientation, frequency, direction and speed are presented in both areas.

Interestingly, neurons throughout the sensory systems, such as auditory or olfactory, exhibit variants of surround modulation, suggesting a non-negligible impact on sensory processing (Angelucci et al., 2017). However, the functional role of surround suppression in the visual system is still a matter of debate (Angelucci et al., 2017). Some theories suggest that surround modulation might help in several visual tasks, such as the detection of object boundaries (Nothdurft, Gallant, and Van Essen, 2000) or segregation of background and the main figure of the visual field (Lamme, 1995).

Moreover, the is no consensus on the precise mechanism which causes the surround suppression (Angelucci et al., 2017). Angelucci et al. (2017) propose that the effect can be attributed to a mixture of influences from the thalamocortical feed-forward connections, corticocortical lateral connections, and cortical feedback.

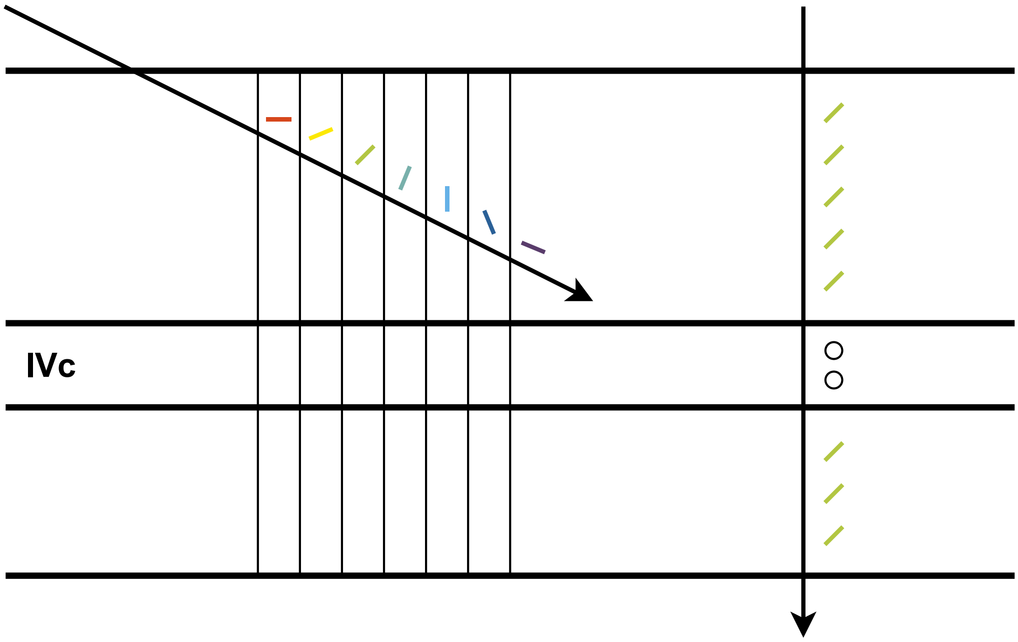


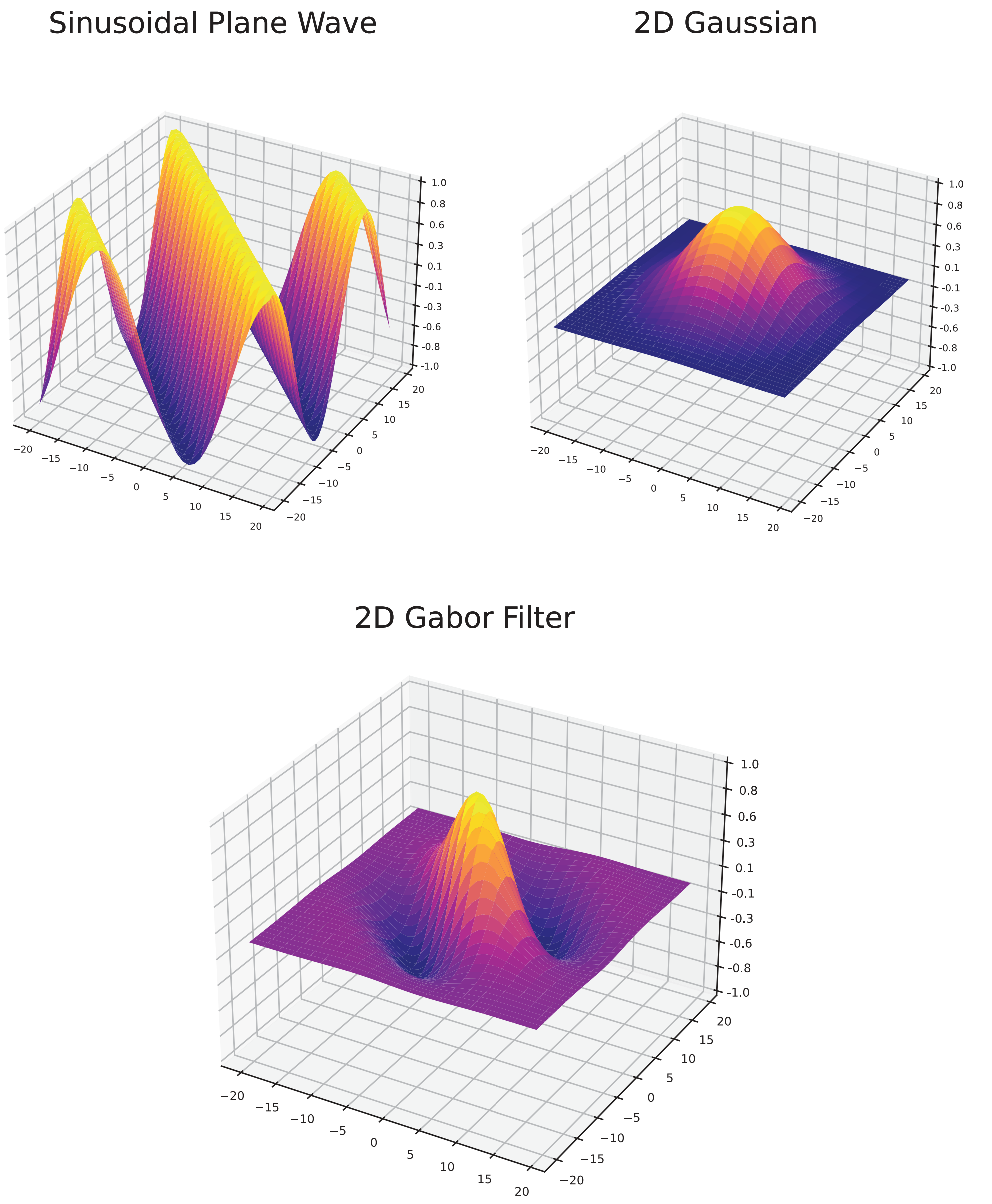
Figure 1.6 An illustration of orientation columns of macaque primary visual cortex. The penetration along the vertical axis on the right shows that the neurons in the same column share the same orientation preference. The preference in layer IVc is depicted as circles, as those receptive fields are circularly symmetric. Conversely, the oblique penetration shows a continuous change of preference. Redrawn based on the original from "Ferrier Lecture - Functional Architecture of Macaque Monkey Visual Cortex"(Hubel and Wiesel, 1977).

Simple Cells Receptive Field

It is possible to precisely approximate the spatial receptive field of a simple cell by a 2D spatial linear filter. A 2D Gabor filter has been demonstrated to be particularly effective in achieving this approximation (Marĉelja, 1980; Jones and Palmer, 1987). This filter can be constructed by multiplying a sinusoidal plane wave by an elliptic Gaussian function as illustrated in Figure 1.7. The advantage of the filter is that it captures both the spatial location and the spatial frequency of the receptive field. Note that the same filters are often used in image processing for features extraction, edge detection, texture segmentation and more (Fogel and Sagi, 1989; Jain and Farrokhnia, 1990; Zalama et al., 2014).

Complex Cells Receptive Field

In comparison to simple cells, much less is known about the function of complex cells (Carandini, 2006). The responses of complex cells cannot be accurately modelled by a linear model (Carandini, 2006). Movshon, Thompson, and Tolhurst (1978) propose and evaluate an explanation of complex cells by a so-called subunit model. According to this model, the receptive field of a complex cell is composed of several subunits that are non-linearly aggregated together, as illustrated in Figure 1.8. A single subunit might correspond to a receptive field of a simple cell (Carandini, 2006). This model explains some general characteristics of these neurons (Movshon, Thompson, and Tolhurst, 1978). Among other properties, Movshon, Thompson, and Tolhurst (1978) demonstrate that complex cells are mostly phase-invariant. Specifically, these neurons exhibit similar responses to stationary sinusoidal gratings, regardless of their phase.



Sinusoidal Plane Wave

2D Gaussian

2D Gabor Filter

Figure 1.7 The construction of a 2D Gabor Filter. A 2D Gabor filter can be obtained by multiplication of a sinusoidal plane wave by a 2D elliptic Gaussian.

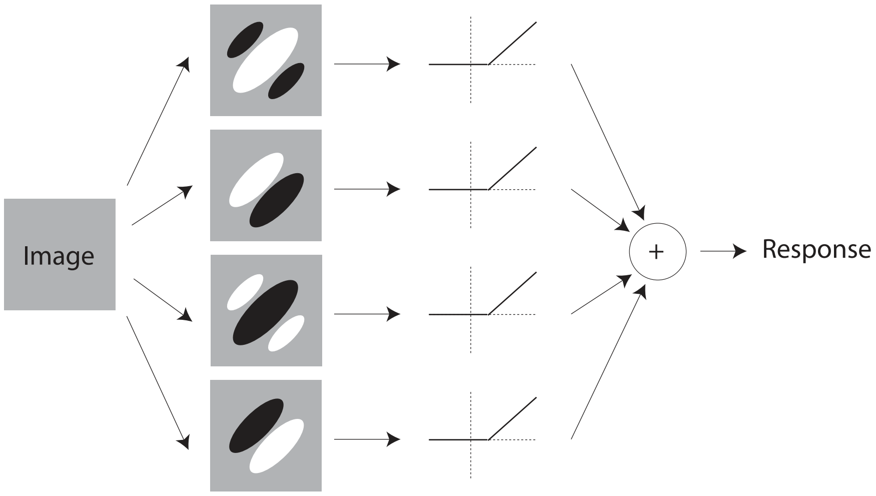


Figure 1.8 The subunit model of a complex cell. The stimulus is first linearly filtered using several simple receptive fields. Subsequently, the outputs are rectified and aggregated. Figure from Carandini (2006)

Chapter 2

Computational neuroscience

Computational neuroscience is a scientific field that employs mathematical theories and computer science techniques to understand the nervous system (Dayan and Abbott, 2001). The emergence of this approach was very natural as we need a reasonable way to analyse and describe the system. Moreover, the advancements in measuring neural activity allow us to collect more data. That enables us to use more complex and data-demanding approaches like deep learning.

A common strategy utilised in computational neuroscience is modelling. Dayan and Abbott (2001) define three categories of models: *descriptive*, *mechanistic* and *interpretative*. Descriptive methods try to summarise the function of neurons in an accurate and condensed way. An example of such a model might be a mathematical description of a receptive field of a particular neuron. Mechanistic models, on the other hand, try to answer how the neuron or group of neurons operate. An instance of such a model might describe a neural circuit's anatomical and physiological properties. Finally, interpretative models are made to help us understand why the neurons work the way they do. This model could, for example, demonstrate that LGN neurons possess specific receptive fields because they efficiently code certain visual features.

In this thesis, we work with representatives of descriptive and mechanistic models. Specifically, we use a deep neural network to solve the system identification problem. This model is descriptive as it only tries to describe the observed function of the neurons mathematically. However, the inner workings of the artificial neural network do not resemble the ones of an actual brain. On the other hand, the spiking neural model that we use was created to mimic the actual visual cortex (Antolík et al., 2018). Despite the necessary simplifications that were made to build the network, it fits the definition of a mechanistic model.

In the following sections, we describe computational neuroscience areas closely related to this thesis. First, we elaborate on *system identification*, which focuses on predicting neurons' activity based on their inputs. Subsequently, we introduce the *spike-triggered average* method, a specific technique for estimating neurons' linear receptive field. Ultimately, we describe the foundations of *spiking models of neurons.*

2.1 System Identification of the Visual System

As Carandini et al. (2005) state, knowing how neurons respond to arbitrary visual stimuli is a necessary condition for understanding how the visual system works. And exactly this is the goal of system identification - prediction of neural activity based on presented stimuli. This approach does not necessarily strive for an interpretable nor mechanistically accurate model; however, it might be beneficial for getting better insights into the modelled system (Butts, 2019). Unsurprisingly, interpretability and accuracy often constitute a trade-off (Butts, 2019).

2.1.1 The Problem

We will first define the system identification problem in vision more formally. Specifically, we define $y = f(s)$, where f is the function of a neuron which maps the stimulus s to a response y. System identification aims to find a mathematical description of the function f. To find this description, we use experimental data in the form of stimulus-response pairs. We will provide a more detailed explanation for each component of the equation.

The Stimulus

Let us first focus on the stimulus. It is usually mathematically described by a vector, matrix or tensor, depending on the real-life stimulus used during the experiments. Similarly to describing a receptive field, the visual stimulus might be defined in time and space. However, it is again common to define the stimulus only as an image (part of a visual field) at one specific point in time. We can also model a function of a neuron with a more straightforward (even scalar) description of a stimulus, such as an orientation of a light bar. Alternatively, it is also possible to incorporate additional information to enhance the visual stimulus, like the animal's behavioural state, such as pupil dilation or eye movement (Willeke et al., 2022).

Hubel and Wiesel (1962) used highly constrained stimuli in their experiments, such as light bars or points of light. However, such stimuli are too simple to unveil the function of more complicated neurons. Another common choice for stimuli is white-noise (Chichilnisky, 2001; Willmore and Smyth, 2003). That is an image whose pixels have random values independent of each other. Considering uniform white noise, this class of stimuli covers all the possible visual stimuli uniformly (i.e. every possible image of a predefined size is sampled with the same probability). However, natural stimuli do not cover this space uniformly; thus, it is unlikely to spot a natural image in white-noise stimuli (Field, 1994). Therefore, we might not be able to identify the function of some neurons that are selective to more sophisticated conditions, as those conditions will probably not appear in the white-noise dataset. To overcome these problems, we can constrain the stimuli by having the already suggested assumption about the neurons. We can assume that the neurons evolved in a way to understand the natural environment (Simoncelli and Olshausen, 2001; Hyvärinen, 2010). Thus, natural stimuli are another frequent choice of input (Lurz et al., 2020; Cadena et al., 2019a; Antolík et al., 2016). This way, the size of the stimuli space is reasonably constrained. Moreover, we also expect that it contains enough complexity to identify the selectivity of neurons in higher visual areas. In contrast to white noise, the pixels of natural stimuli can be highly correlated. Hence, we might need to adapt our techniques to avoid bias (Willmore and Smyth, 2003).

The Response

Neurons transfer information using action potentials, often called spikes. Thus, predicting the response of a neuron is equivalent to predicting the spikes generated by the neuron following the presentation of stimuli. The action potentials may vary in shape, amplitude or duration (Dayan and Abbott, 2001, p. 8). However, for the purpose of system identification, we consider all spikes identical and analyse only their count (Dayan and Abbott, 2001, p. 8). This quantity is usually called *firing rate*. Specifically, we will use the term firing rate solely in the meaning of less ambiguous term *spike-count rate*, which is the number of recorded spikes divided by the duration of the measurement (Dayan and Abbott, 2001, p. 9).

The Mapping Function

As we are familiar with the inputs and the outputs, let us examine the function f itself. Our goal is to derive the function from the observed experimental data. However, the experimental samples cover all the possible inputs very sparsely. That is why we usually assume a specific mathematical form of the function and then tune its parameters utilising the experimental results (Butts, 2019). Butts (2019) declare three aspects to consider when choosing the mathematical form: (1) how easy it is to tune the parameters, (2) the performance of the resulting model, and (3) what insights into the neuron's computation do we get when using the specific model. We will see some possible model choices in the following subsections.

2.1.2 MAP Framework

As already suggested, the goal of system identification is to find a constrained mathematical form of a neuron's function. In other words, the objective is to find its most probable model. Historically, there were many approaches how to obtain such models (Chichilnisky, 2001; Antolík et al., 2016; Cadena et al., 2019a; Lurz et al., 2020). Currently, the majority of those techniques are described using the *maximum a posteriori framework* (MAP) (Butts, 2019; Wu, David, and Gallant, 2006). Within this subsection, we outline the fundamentals of this framework.

Due to the possible complexity of neural computations and limited data, we often need to make additional assumptions about the target function. MAP estimation allows us to do that by including prior knowledge about the system in the parameter search. Particularly, we use the MAP framework to find the best model from a specific *model class* under the assumption of a particular *noise distribution* and a *prior* which determines the plausibility of a model (Wu, David, and Gallant, 2006).

In order to use the framework, we need a dataset made of stimulus-response pairs (s_i, y_i). Let S be the list of N stimuli, and let Y be the list of the corresponding responses. Then, we want to find such parameters θ that are the most probable given the observed data. That can be written in the following fashion.

$$\begin{aligned}
\theta_{\text{MAP}} = \text{argmax}_\theta p(\theta|Y \cap S) &\overset{(1)}{=} \text{argmax}_\theta \frac{p(\theta \cap Y \cap S)}{p(Y \cap S)} \\
&\overset{(2)}{=} \text{argmax}_\theta \frac{p(Y|\theta \cap S) \cdot p(S \cap \theta)}{p(Y \cap S)} \\
&\overset{(3)}{=} \text{argmax}_\theta \frac{p(Y|\theta \cap S) \cdot p(S|\theta) \cdot p(\theta)}{p(Y \cap S)} \\
&\overset{(4)}{=} \text{argmax}_\theta \Big[p(Y|\theta \cap S) \cdot p(\theta)\Big] \\
&\overset{(5)}{=} \text{argmax}_\theta \Big[\log p(Y|\theta \cap S) + \log p(\theta)\Big] \\
&\overset{(6)}{=} \text{argmin}_\theta \Big[\underbrace{- \sum_{i=1}^{N} \log p(y_i|s_i \cap \theta)}_{\substack{\text{Negative log-likelihood} \\ \text{(Loss function)}}} \underbrace{- \log p(\theta)}_{\substack{\text{Negative log-prior} \\ \text{(Regularizer)}}} \Big]
\end{aligned} \tag{2.1}$$

The first three equalities hold from elemental properties of probability. Note that the denominator stays the same for all the parameters θ; thus, we can remove it as we only search for the point of maximal value. Moreover, it holds that

$p(S|\theta) = p(S)$, which also stays the same for all the choices of parameters. By removing these two elements, we obtain the fourth equality. We retrieve the fifth line simply by applying the logarithm function on both terms. This operation, again, does not change the point of maximal value. Finally, we can transform the maximisation into minimisation and sum the log probabilities for each dataset sample. Through this modification, we obtain two terms: the *loss function* in the form of negative log-likelihood; and the *regulariser* in the form of negative log-prior.

The likelihood term $p(y_i|s_i \cap \theta)$ describes how well the model with parameters θ predicts the response to the stimulus s_i. To estimate the value of this term, we assume a specific noise distribution of the data. This distribution then determines the loss function, which controls the penalisation of a specific prediction given the actual response (Wu, David, and Gallant, 2006). For example, we can assume that the noise is Gaussian; in that case, we would use a square loss function. For firing rate prediction, the Poisson noise distribution is often assumed (Wu, David, and Gallant, 2006; Butts, 2019). The corresponding Poisson loss function is the following:

$$L(f_\theta(s_i), y_i) = -\log p(y_i|s_i \cap \theta) = f_\theta(s_i) - y_i \log f_\theta(s_i), \tag{2.2}$$

where f_θ is the evaluated model with parameters θ.

The term $p(\theta)$ corresponds to the prior. It integrates the plausibility of particular parameter choices independently of the observed data. The prior further restricts the model in addition to the experimental data. One of the experimenter's challenges is to find the optimal balance (in the form of weights) between the likelihood and the prior terms. When the dataset is large enough, the benefit of the prior penalty is negligible as the data sufficiently constrain the model. On the contrary, it might even harm the model's performance in this case. Therefore, cross-validation is often used to determine the suitable weights.

The MAP framework can be used to describe most of the statistical models utilised for system identification. Notice that when no assumption about the probability of parameters is made, the MAP estimation is equivalent to maximising the likelihood of the data. The rest of this section is dedicated to the popular model classes that are used for system identification.

2.1.3 The Classical Approach

This subsection presents classical approaches to system identification that have been used throughout history. They are characterised by greater transparency compared to the currently popular deep neural networks. However, they are often not powerful enough to accurately predict responses of neurons with complex receptive fields (Cadena et al., 2019a).

Linear Models

The most straightforward method for approximating neural responses is using a linear filter. Given a multi-dimensional stimulus $\boldsymbol{s} \in \mathbb{R}^d$, the corresponding response can be estimated using a weighted sum as follows:

$$\hat{f}(\boldsymbol{s})_{\text{linear}} = \sum_{i=1}^{d} \beta_i \cdot s_i = \boldsymbol{\beta} \cdot \boldsymbol{s}. \tag{2.3}$$

Each weight β_i indicates how much the particular dimension affects the neural response. Evidently, such estimator has very limited predictive power; however, it has a very clear interpretation. For example, when the stimulus $\boldsymbol{s}$ is an image, the weight vector $\boldsymbol{\beta}$ can also be interpreted as an image. Thus, we can plot the weights and interpret the visualised filter.

Non-linear Methods

Using only the linear filters would be very limiting, as it has been observed that the neural responses exhibit non-linear behaviour (Hubel and Wiesel, 1977). Moreover, the output of the linear model is unbounded and may be negative. We can extend the simple linear model by wrapping the result with one non-linear one-dimensional function:

$$\hat{f}(\boldsymbol{s})_{\text{LN}} = F\left(\sum_{i=1}^{d} \beta_i \cdot s_i\right) = F(\boldsymbol{\beta} \cdot \boldsymbol{s}). \tag{2.4}$$

This model is called the linear-non-linear (LN) model. The LN models retain great interpretability as the linear filter and the non-linearity can be easily separated. Observe that solely the output is scaled using a non-linear function F. This function can be learned or fixed, and it is usually non-negative when predicting the firing rate. The individual dimensions are still weighted using linear weights. Therefore, such a model is not capable of capturing non-linear feature selectivity. LN models were historically very popular for the prediction of visual neuron responses. They were especially effective for firing rate estimation of neurons in the retina, LGN and simple cells in the primary visual cortex (Butts, 2019).

Note that even these simple models can take into account other observed measurements in addition to the visual stimuli. We can enhance the stimuli by other dimensions, such as the neuron's past firing rate, firing rates of other neurons or certain behavioural features like pupil diameter (Butts, 2019). These supplementary dimensions can then be linearly weighted together with the primary visual features.

The neural response might depend on a combination of features; thus, we need a model that can capture these dependencies. One possible approach is to use a generalised LN model which uses a multi-dimensional non-linear function and multiple linear filters:

$$\hat{f}_{\text{GLN}}(\boldsymbol{s}) = F_k(\boldsymbol{B_1} \cdot \boldsymbol{s}, \boldsymbol{B_2} \cdot \boldsymbol{s}, \ldots, \boldsymbol{B_k} \cdot \boldsymbol{s}). \tag{2.5}$$

$\boldsymbol{B_i}$ corresponds to a singular linear filter and F_k is a k-dimensional non-linear function. In other words, this model allows capturing multiple linear filters and weighting them using a non-linear function. This operation can also be viewed as projecting the stimulus into a subspace of a lower dimension and then applying the non-linearity.

Another extension of the LN model is the so-called LNLN cascade. This technique applies a non-linear function on a linear combination of several LN models (subunits). The following equation describes the model:

$$\hat{f}_{\text{LNLN}}(\boldsymbol{s}) = F\left(\sum_{i=1}^{n} w_i \cdot \hat{f}_{LN_i}(\boldsymbol{s})\right), \tag{2.6}$$

where F stands for the final one-dimensional non-linear function, n is the number of subunits and $\hat{f}_{LN_i}$ stands for a single LN model which is weighted by a coefficient w_i. Again, all the non-linearities can be fitted, but they are usually fixed as, for example, rectified linear unit, quadratic function or sigmoid. It is easy to notice that this model can be interpreted as a specific instance of a neural network. Consequently, it can be shown that the LNLN cascade can accurately approximate any non-linear function if enough subunits are used (Cybenkot, 1989; Hornik, 1991).

2.1.4 Deep Learning for System Identification

LN models and LNLN cascades have historically been important baseline models of V1. However, they fail to capture more complex non-linear receptive fields. This deficiency is especially noticeable when presenting natural stimuli and when modelling higher stages of the visual hierarchy (Cadena et al., 2019a; Klindt et al., 2018). The advances in measuring large populations of neurons and the successes of deep learning in various domains stimulated the use of neural networks in system identification (Kindel, Christensen, and Zylberberg, 2017; Klindt et al., 2018). Thanks to that, the models are significantly more accurate and are able to predict the responses to more complex stimuli (Cadena et al., 2019a; Lurz et al., 2020). However, the predictive power of neural networks comes with several disadvantages, particularly their troublesome interpretability and data-demanding nature (Benjamin et al., 2018; Kriegeskorte, 2015).

In this subsection, we briefly review currently used methods and describe the most essential characteristics of utilising deep learning for system identification. To learn in greater detail about the motivations and techniques of modelling biological systems using deep learning, we refer the readers to the valuable review from Kriegeskorte (2015) about the state of the field as of 2015.

Architectures

In the great variety of neural network architectures, convolutional neural networks (CNNs) were proven to be an adequate choice for visual system modelling (Lurz et al., 2020; Cadena et al., 2019a). This is not surprising as CNNs are used for solving many visual tasks such as object detection or image segmentation, and they often achieve super-human performance (Russakovsky et al., 2015; Tan and Le, 2021). Moreover, the representations captured by the layers of CNN, which was trained for object detection, exhibited similarities to various stages of the biological visual processing (Khaligh-Razavi and Kriegeskorte, 2014; Lindsay, 2021). Nonetheless, it is important to acknowledge that the majority of networks used for visual system identification are feed-forward. That is, the computation of the network is strictly serial, despite the recurrent connectivity of the neurons in the brain.

As already stated, deep neural networks can be very data-demanding, and the amount of training data is limited by the measurements on animals. This problem is addressed by fitting the neural network for many neurons simultaneously (Antolík et al., 2016; Lurz et al., 2020). Specifically, most of the current models have a shared component (sometimes referred to as core) that is trained using data from all the neurons available. This shared core can be implemented, for example, as a CNN. Then another module of the network is needed for the prediction of responses for particular neurons. This module is called a readout, and its most straightforward implementation is a weighted sum of the outputs from the core. This specific implementation, however, introduces many parameters for each neuron in the dataset. Smarter readouts with fewer parameters were invented based on the idea that the majority of neurons perform similar computation only in different regions of the visual field (Klindt et al., 2018; Lurz et al., 2020). Thanks to this realisation, a more effective readout layer can be made from two operations: prediction of the receptive field position and processing only the features that correspond to this position. The separation of the shared feature space and the readout leads to richer features learned and less data needed for more accurate predictions.

Goal-driven Networks and Transfer Learning

So far, we have considered fitting the models only directly on the stimulus-response pairs. However, as in other branches of deep learning, we can exploit pre-training and transfer learning. That is, training the network on a different but related task and then employing this model (with appropriate modifications) for our original objective. This technique can be advantageous because obtaining neural data can be quite demanding. When an appropriate goal is chosen, the model can be fitted with significantly more training data, which can result in better inner representations and performance. In computational neuroscience, this approach is called *goal-driven* in opposition to the standard *data-driven* approach.

Currently, the standard method is to use a CNN trained for the task of image classification as the core module (Cadena et al., 2019a). A readout model that takes the features of the CNN as its input is then fitted using the standard stimulus-response dataset. The core can be optionally also fine-tuned. Depending on which area of the visual cortex is modelled, we can use different levels of the CNN as the core. For instance, when modelling V1, it would probably be better to use one of the first layers of the CNN, whereas the higher stages of visual processing would probably require using deeper layers (Butts, 2019). Refer to Figure 2.1 for a diagram of a goal-driven model.

One challenge of using goal-driven networks is specifying precise goals and providing relevant training data. As stated above, the object recognition goal is often chosen. However, it might not be sufficient to use this goal only as some neurons might be contributing to a very different objective. And thus, important features for some neurons might not be present in the pre-trained shared feature space. For example, Cadena et al. (2019b) show that a specific CNN trained on object recognition reaches a similar performance as the same network with random weights when predicting the responses of the mouse's primary visual cortex. Therefore, object recognition might not be the principal goal of the mouse visual system.

Interpretability and Gaining Insights

The use of deep neural networks for system identification is often criticised for the challenging interpretability (Kriegeskorte, 2015; Benjamin et al., 2018). One could argue that this approach is replacing one black-box model with another and therefore providing no relevant insights. Let us present some arguments made in favour of using DNNs in the context of system identification.

Firstly, the capability of a complex neural network to perfectly predict neural responses implies that the network captured all the necessary biological computations. In a deep neural network, these computations are captured at a very

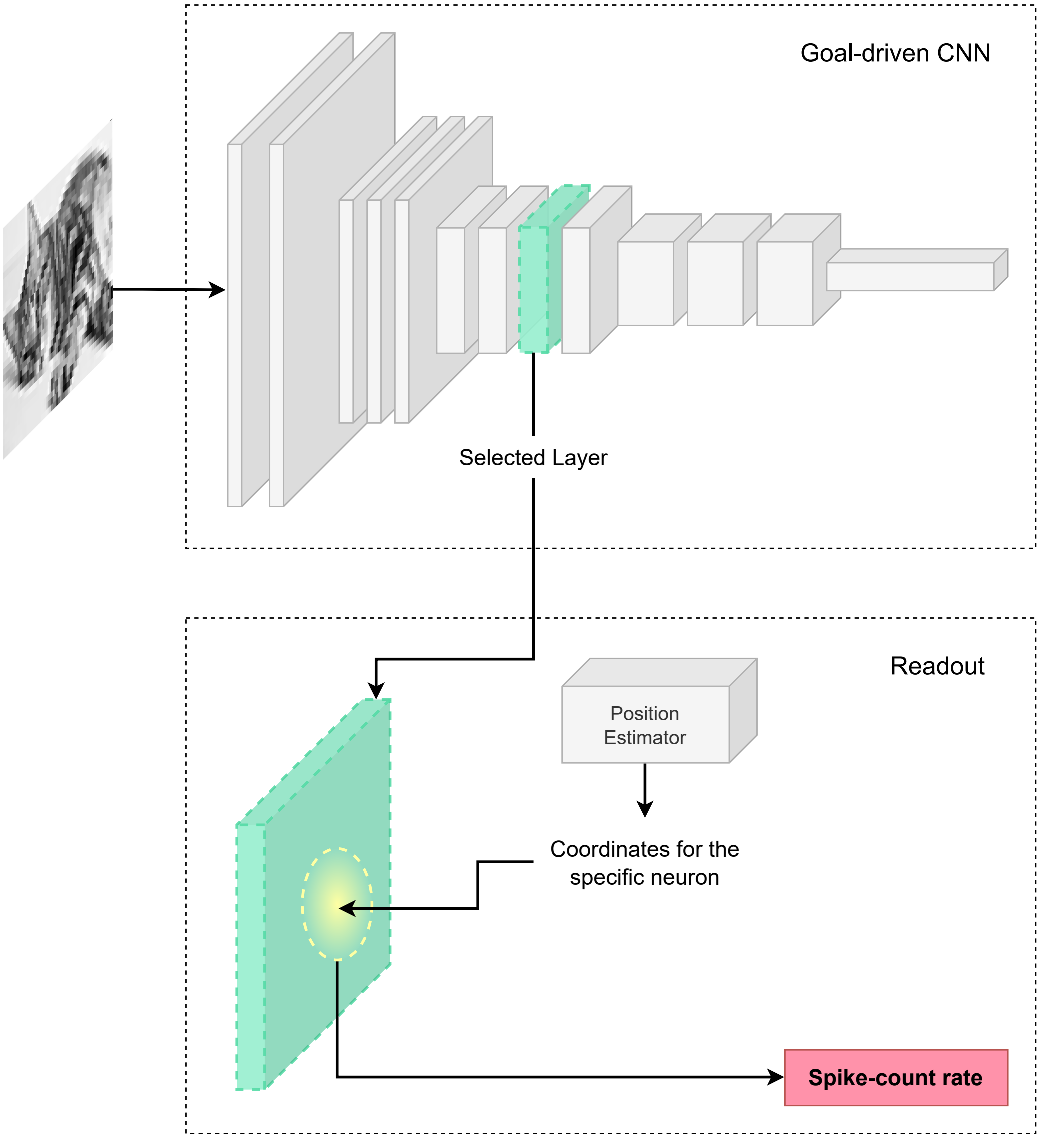


Figure 2.1 An example of a goal-driven network for identification of a visual system. First, the CNN is trained on a different task, such as object recognition. The experimenter then selects a specific layer of the pre-trained network to be the layer of features. This layer is the input to the readout module, which extracts relevant information from the features, e.g., by predicting the receptive field position for the specific neuron. Finally, the extracted features are transformed into the firing rate prediction.

low level of description. Nonetheless, having this perfect predictor would enable the scientists to perform better in silico experiments with the goal of extracting higher-level descriptions (Kriegeskorte, 2015; Lurz et al., 2020).

Kriegeskorte (2015) further argues that using such complex models like deep neural networks might be necessary to capture the neural inner workings properly. The classical approach to modelling or other approaches that would give us a concise description of the systems might not be sufficient when it comes to complex biological systems.

Moreover, interpretability might be wrongly interchanged with the transparency of a model. Lipton (2018) presents two areas of interpretability: transparency and post hoc interpretability. Transparency denotes how easy it is to understand how a model works. For example, how complicated it would be to manually predict a response step-by-step given the model parameters and the inputs. It can also describe how intuitive each part of the computations is or how complex the fitting algorithm is. On the other hand, post hoc interpretability is a specific approach to exploiting the fitted model for gathering insights. This can be done, for instance, by inspecting the inner representations or visualisations of examples. Lipton (2018) actually suggests that "linear models are not strictly more interpretable than deep neural networks".

Finally, Benjamin et al. (2018) assert that the generalised linear models can be, in fact, difficult to interpret when used for modelling higher areas of visual systems. Moreover, he suggests that many scientific questions about a neural system can be answered solely by exploiting post hoc interpretability. For example, we can use this approach to examine the relevance of a specific input just by comparing the relative performance of the model.

To conclude, using deep learning techniques for system identification can undoubtedly be beneficial despite the deficient transparency of the models.

2.1.5 Performance Evaluation

There is currently no consensus on what measure should be used for determining goodness of fit of neural system identification models (Pospisil and Bair, 2021; Carandini et al., 2005). In this subsection, we introduce several metrics and describe their attributes. In order to understand why it is challenging to select a suitable metric, we first describe the correlation coefficient and the coefficient of determination. Subsequently, we define specialised metrics created with the intention of overcoming some disadvantages of the two basic metrics.

Correlation

Correlation, in our case more precisely called sample Pearson's correlation coefficient, expresses the extent of linear correlation between two sets of data. We define $corr_{\hat{y}y}$ (often also referred to as r) for one neuron as a correlation between the model outputs $\hat{y}$ and the true responses y as follows:

$$corr_{\hat{y}y} = \frac{\sum_{i=1}^{m}(\hat{y}_i - \mu_{\hat{y}})(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{m}(\hat{y}_i - \mu_{\hat{y}})^2}\sqrt{\sum_{i=1}^{m}(y_i - \mu_y)^2}}\,, \tag{2.7}$$

where m is the number of stimuli in the dataset, $\hat{y}_i$ is the model's output for i-th image and $\mu_{\hat{y}}$ is the sample mean of $\hat{y}$. Analogously, y_i denotes the neuron's true response to i-th stimulus, and μ_y is the mean of all the neuron's responses.

Recall that the value of $corr$ lies in the interval $[-1, 1]$. The value 0 indicates no linear correlation; conversely, 1 and -1 indicate perfect linear correlation. However, there is no simple intuition on the scale of the correlation coefficient (Puth, Neuhäuser, and Ruxton, 2014). Thus, it might be confusing to compare two models using this measure. Another disadvantage is the lack of consideration for trial-to-trial variance, i.e. the changes in neuron responses when repeating an experiment with the same stimulus. Nonetheless, correlation can be a useful metric to report during the training of a model because it does not require repeated trials (those are usually present only in the test dataset).

Coefficient of Determination

Related to the correlation coefficient is the so-called *coefficient of determination*, also known as R^2 or R-squared. The coefficient of determination aims to solve the interpretability problem of the correlation coefficient (Puth, Neuhäuser, and Ruxton, 2014). R-squared indicates the portion of the variance in the data explained by the model. For the purpose of defining R-squared, let us first define the total sum of squares (TSS) and the sum of squares of residuals (RSS):

$$\text{TSS} = \sum_{i=1}^{m}(y_i - \mu_y)^2$$
$$\text{RSS} = \sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

using the same notation as in the definition of correlation 2.7. Subtracting the ratio of RSS and TSS from 1, we obtain the R-squared:

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}\,. \tag{2.8}$$

Interestingly, R-squared in some occasions, such as linear regression, equals to the correlation coefficient squared (hence the name) (Devore, 2012, p. 510). Be aware that this equality does not hold in general. For instance, when the prediction is worse than predicting the mean (in terms of squared error), the R-squared is negative.

A coefficient of determination might be more intuitive when comparing several models because the interpretation is more explicit compared to the correlation coefficient. Nonetheless, the issue of not accounting for the trial-to-trial variability remains. Thus, the metric might often underestimate the actual performance of a model when there is a high variability of the responses.

Metrics Accounting for Trial-to-Trial Variance

To overcome the issue of trial-to-trial variance, several authors proposed new metrics (Pospisil and Bair, 2021; Cadena et al., 2019a; Walker et al., 2019). Let us introduce the *fraction of explainable variance explained* (introduced by Cadena et al. (2019a)) and the *fraction of oracle correlation* (proposed by Walker et al. (2019)).

Cadena et al. (2019a) estimate each neuron's amount of explainable variance (EV). This value is obtained by withdrawing a specific noise portion of the overall variance:

$$\text{EV} = \text{Var}[Y] - \sigma^2_{\text{noise}} \,. \tag{2.9}$$

Here Y stands for the matrix of actual responses with shape $m \times t$, where m again denotes the number of stimuli and t stands for the number of repetitions. Accordingly, $\text{Var}[Y]$ is defined as a variance of all the elements of the matrix. Finally, they refer to σ^2_{noise} as the "observation noise", and it is computed as follows:

$$\sigma^2_{\text{noise}} = \mathbb{E}_i[\text{Var}_j[Y_{i,j}]] \,. \tag{2.10}$$

That is, we first compute the variances across repetitions (receiving m values), and then we average the variances. To make the explainable variance comparable across neurons, we will often use a fraction of explainable variance (FEV):

$$\text{EV} = \frac{\text{EV}}{\text{Var}[Y]} \tag{2.11}$$

We can now compute the actual performance metric called *fraction of explainable variance explained* (FEVE) using the following formula:

$$\text{FEVE} = 1 - \frac{\text{MSE}(\hat{y}, Y) - \sigma^2_{\text{noise}}}{\text{Var}[Y] - \sigma^2_{\text{noise}}} \,. \tag{2.12}$$

The mean square error (MSE) is computed between the predictions $\hat{y}$ and the true responses from all the trials:

$$\mathrm{MSE}(\hat{y}, Y) = \frac{1}{m \cdot t} \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq t}} (\hat{y} - Y_{i,j})^2 \, .$$

Note that the value of FEVE might be negative when the mean square error is greater than the variance of the responses. Moreover, FEVE can also reach values greater than 1. This can happen when the mean square error is lower than the observation noise (remember that the observation noise is only an estimate). Apart from the corner cases, the intention of FEVE is to compare our model with the best model possible in the form of a fraction.

Walker et al. (2019) proposed a metric called *fraction of oracle correlation* with a similar intention but different approach. The first step is again estimating the maximal possible performance. In this case, it is called *oracle correlation.* As the name suggests, it is an estimate of the maximum reachable correlation. It is estimated by computing the correlation of a jackknife estimator on a dataset with repetitions.

Let us explain how to compute the oracle correlation for one neuron. For each stimulus and repetition, the jackknife estimator computes a mean from the rest of the trials to estimate the response of the selected repetition. Using this technique, we create $m \cdot t$ pairs in a form of $(\tilde{y}_{i,j}, Y_{i,j})$, where $\tilde{y}_{i,j}$ is the jackknife estimate for i-th stimulus and j-th repetition. The oracle correlation of a neuron is then simply a correlation of the pairs. To compute the fraction of the oracle correlation, we pair the per-neuron oracle correlations with the per-neuron correlations of our model. Then, we fit a line without an offset to the pairs using the least-squares approach. Finally, the fraction of oracle correlation is the coefficient received from the least-squares solution, i.e. the slope of the line. Notice again that this measure is also not theoretically bounded by the interval $[0, 1]$. It might reach values lower than zero when some of the correlations are negative. Furthermore, the fraction can be greater than one when the correlations of the model are higher than the estimated oracle correlations.

2.2 Spike-Triggered Average

Spike-triggered average (STA) is a method for analysing neuron computations. It can also be thought of as a specific example of a linear model, as defined in the previous section. Under specific conditions, it allows us to characterise a linear part of a neuron's receptive field (Chichilnisky, 2001). This approach is often used due to its simplicity and straightforward interpretation.

2.2.1 STA and White Noise Analysis

STA is often employed in the context of a method called *white noise analysis.* White noise analysis is performed by presenting white noise stimuli and then analysing the neurons' responses. In this subsection, we present the basic idea behind this technique.

Firstly, the stimuli used for white-noise analysis are required to be drawn from a radially symmetric distribution (Chichilnisky, 2001). That is, when stimuli $\boldsymbol{s}$ and $\boldsymbol{s}^*$ are vectors of equal lengths from k-dimensional stimuli space S, the probability of being drawn must be the same for both vectors. To put it differently, the following implication must hold:

$$|\boldsymbol{s}| = |\boldsymbol{s}^*| \Rightarrow p(\boldsymbol{s}) = p(\boldsymbol{s}^*). \tag{2.13}$$

Note that the Gaussian distribution satisfies this condition; thus, the Gaussian white noise is a usual choice of stimuli. Each stimulus generated by this distribution is a k-dimensional vector where each element is independently drawn from a Gaussian distribution.

This approach assumes a model of a neuron where the stimulus is first linearly weighted with weights $\boldsymbol{w}$ and then goes through a static, possibly non-linear function F:

$$f(\boldsymbol{s}) = F(\boldsymbol{w} \cdot \boldsymbol{s}). \tag{2.14}$$

Observe that this model matches the description of an LN model from Section 2.1. Recall that the weights have the same shape as the stimulus and can be interpreted as the neuron's selectivity.

The spike-triggered average $\boldsymbol{a}$ is a vector of the same size as the stimulus. It can be obtained by summing all the stimuli that triggered a spike and dividing this sum by the total count of spikes. We can express it in a more concise form as follows:

$$\boldsymbol{a} = \frac{\sum_{i=1}^{T} \boldsymbol{s_i} \cdot \boldsymbol{y_i}}{\sum_{i=1}^{T} \boldsymbol{y_i}}. \tag{2.15}$$

The number T denotes the total count of stimuli, s_i is the i-th stimuli and y_i stands for the number of spikes triggered when the i-th stimulus was presented. The formula can be rewritten as a matrix multiplication:

$$\begin{aligned} \boldsymbol{a} &= \frac{S^T \cdot \boldsymbol{y}}{\sum_{i=1}^{T} \boldsymbol{y_i}} \\ &\propto S^T \cdot \boldsymbol{y}, \end{aligned} \tag{2.16}$$

where S is the matrix of stimuli and $\boldsymbol{y}$ is the vector of corresponding responses.

Chichilnisky (2001) proves that under the above-stated assumptions, the vector $\boldsymbol{a}$ obtained by computing the STA is proportional to the weights $\boldsymbol{w}$. As already suggested in Section 2.1, obtaining these weights can be very useful as they indicate the linear selectivity of a specific neuron.

Willmore and Smyth (2003) arrive at the same result by the assumption that a response to a stimulus can be approximated by weighting the stimulus by linear weights as in:

$$S \cdot \boldsymbol{f} = \boldsymbol{y}, \tag{2.17}$$

where $\boldsymbol{f}$ is the vector of weights. When we multiply the equation by S^{-1}, we obtain the following:

$$\boldsymbol{f} = S^{-1} \cdot \boldsymbol{y}. \tag{2.18}$$

However, this operation is naturally possible only if the inverse of S exists. Furthermore, when the stimuli matrix S is orthogonal, it holds that $S^{-1} = S^T$. From Equation 2.16, it follows that, under these conditions, the weights $\boldsymbol{f}$ are proportional to the spike-triggered average $\boldsymbol{a}$.

2.2.2 STA for Natural Stimuli

Natural stimuli, unfortunately, do not satisfy the conditions of white noise analysis. Thus, extracting the linear receptive field just by transposing the stimuli matrix is not possible. However, we can approximate the solution by finding a pseudo-inverse of S (Willmore and Smyth, 2003). Specifically, we can search for S^p in:

$$\hat{\boldsymbol{f}} = S^P \cdot \boldsymbol{y}, \tag{2.19}$$

where $\hat{\boldsymbol{f}}$ stands for the approximation of linear weights $\boldsymbol{f}$ defined in Equation 2.17.

We can find this pseudo-inverse by standard least-square solution, which results in the following equation:

$$\hat{\boldsymbol{f}} = S^P \cdot \boldsymbol{y} = (S^T S)^{-1} S^T \boldsymbol{y}. \tag{2.20}$$

Observe that this corresponds to finding weights of linear regression. Even though this method minimises the mean-square error, the variability of neural responses substantially corrupts the solutions with noise (Willmore and Smyth, 2003). Theoretically, this would not be a problem when a great amount of data is presented. However, in practice, it shows that additional constraints are necessary to find a satisfactory solution. This problem naturally leads to the introduction of prior as explained in Section 2.1 in the context of the MAP framework. In this specific case, we can use a prior in a form of L_2 regularisation, which minimises the square of weights. Thanks to that, the high-frequency portion of the linear filter is penalised, which is desirable for many neurons (Willmore and Smyth, 2003). The final regularised form is equivalent to the so-called ridge regression:

$$\hat{\boldsymbol{f}} = S^P \cdot \boldsymbol{y} = (S^T S + \delta I_k)^{-1} S^T \boldsymbol{y}, \tag{2.21}$$

where $\delta \in \mathbb{R}^+$ is the regularization parameter and I_k stands for an identity matrix of size $k \times k$ with k being the size of a stimulus. The parameter δ balances the strength of the regularisation and the constraints from the stimuli. Lower δ results in weaker regularisation and vice-versa. An appropriate value of δ can be determined via cross-validation. That is, fitting the ridge regression on a training portion of data with various choices of δ and then evaluating the models on the responses from the validation set.

2.3 Spiking Neural Models

The system identification methods predict how a neuron would respond when a specific stimulus is presented. However, they do not attempt to do so by modelling the biological neural network but rather by estimating a mathematical description of the neuron's operation. Spiking neural networks, on the other hand, try to mimic real neural systems by modelling each neuron and connecting these models together. There is a wide range of biological neuron models available. Some aim to replicate the intricate biophysical processes of neurons, while others employ various simplifications to facilitate computational simulations.

In this section, we first present a condensed introduction to the biophysics of neurons. Subsequently, we present a series of models, starting with the perfect integrate-and-fire model and progressing towards the widely utilised exponential integrate-and-fire model. This section aims to build a foundation of spiking neural models, enabling the reader to better comprehend the particular network we are analysing. For more details about the biological aspects of neurons, we recommend *Neuroscience* (Bear, Connors, and Paradiso, 2016). For better grasp of neural systems modelling, we suggest reading *Theoretical Neuroscience* (Dayan and Abbott, 2001) and *Spiking Neuron Models* (Gerstner and Kistler, 2002).

2.3.1 Neural Physiology and Neuroelectronics

As illustrated in Figure 2.2, a neuron consists of three main parts: *dendrites*, *soma* and *axon*. Simply put, dendrites provide inputs to the neuron, a soma aggregates the inputs, and an axon acts as the neuron's output. Most neurons transmit

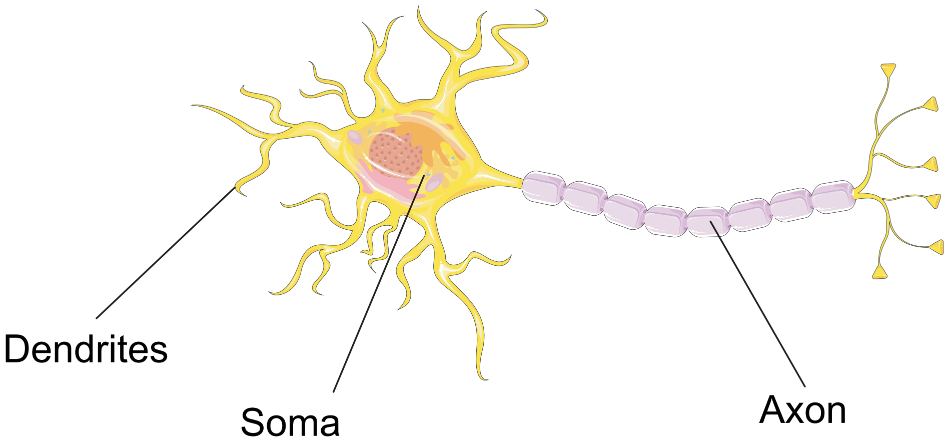


Figure 2.2 Illustration of a neuron. Annotations were added to the "Neuron" illustration from Servier Medical Art. Servier Medical Art by Servier is licensed under a Creative Commons Attribution 3.0 Unported License (https://creativecommons.org/licenses/by/3.0/).

information using electrical signals that flow along the neural body. We call these signals *spikes* or *action potential*. Spikes are usually considered identical; the information is encoded in their count and distribution (Gerstner and Kistler, 2002). Neurons are interconnected through synapses, facilitating the transmission of information between them.

Excitable Membrane

The electricity in neurons is induced by the flow of ions (electrically charged atoms). Specifically, a neuron is enclosed in a special excitable membrane, which controls the flow of these ions. The membrane itself isolates the cytosol of the neuron and extracellular fluid as it is not permeable to most charged atoms. However, it contains pores and special *channels* that allow a selective flow of specific ions. Moreover, the channels can react to changes in the surrounding environment, such as changes in electrical potential or the presence of specific molecules (neurotransmitters). Apart from the channels, the membrane also contains so-called *ion pumps* that move particular ions from and into the neuron resulting in a rising concentration of the respective ion on one of the sides.

The movement of ions across the pores and channels in the membrane is caused by two main factors: the concentration gradient and the difference of electrical potential across the membrane (Bear, Connors, and Paradiso, 2016). The particles move from the areas of high concentration to the ones with a lower concentration; this movement is called diffusion. On the other hand, the difference of potentials pushes the ions according to their charge and the charges inside and outside of the neuron: positive particles move to the side with a negative charge and vice-versa. Note that, for each ion, there is a specific value of membrane potential which perfectly balances the electrical and diffusion forces.

Given all of these properties, when no input is received, the membrane potential remains steady at the value of approximately -65 mV (Bear, Connors, and Paradiso, 2016). We call this value a *resting potential.*

Action Potential

As stated above, the membrane potential at rest is negatively polarised. When a neuron is depolarised enough, the potential may rapidly turn positive - generating an action potential. As illustrated in Figure 2.3, the action potential follows a sequence of specific stages. First, we observe a rising phase, i.e. a slight depolarisation of a neuron. It can be induced in various ways, for example, by the entry of Na^+ ions through channels that are sensitive to neurotransmitters released by another neuron at a synapse. When the membrane potential reaches a threshold value of approximately -55 to -50 mV, the membrane generates a rapid spike, and the membrane potential briefly turns positive (Dayan and Abbott, 2001). We call this phase overshoot, and it is caused by the opening of voltage-gated channels that allow a further influx of Na^+ ions into the neuron. Quickly after, the voltage-gated sodium channels close and the potassium voltage-gated channels open, which allows a flow of K^+ ions out of the neuron. These two events bring the neuron into a falling phase when it swiftly re-polarises. The re-polarisation actually reaches a potential lower than its original resting value because the voltage-gated potassium channels do not close that quickly. Therefore, we call this phase a hyper-polarisation. In the final stage, the membrane potential slowly reaches the resting value.

Even when the electrical current flowing into the neuron is constant, the neurons cannot generate one action potential right after another. There is an absolute refractory period of approximately 1 ms after each spike when creating another one is impossible. Moreover, there is also a relative refractory period during which a greater amount of input current is needed in order to generate spikes.

Synapses

When an action potential is generated, it propagates along the axon. However, the signal also needs to spread to other neurons. This transfer happens at a synapse, a place where the axon of a pre-synaptic neuron meets a dendrite of a

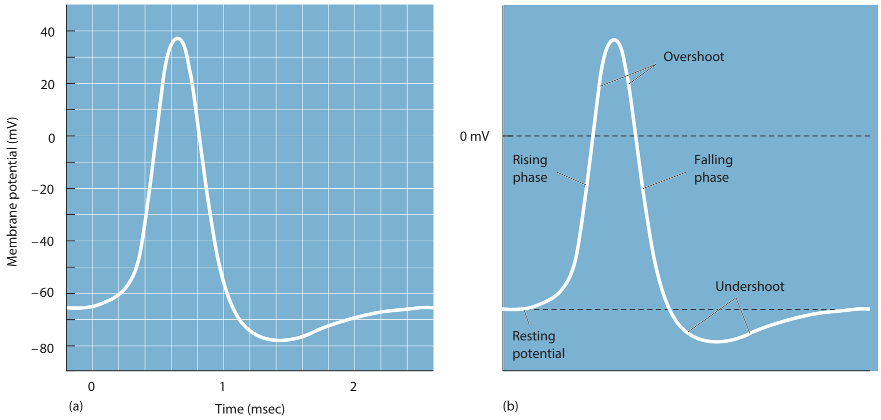


Figure 2.3 An action potential. (a) An action potential recorded by an oscilloscope. **(b)** Stages of action potential. A figure from *Neuroscience* (Bear, Connors, and Paradiso, 2016, p. 84).

post-synaptic neuron. Based on the signal transfer mechanism, we distinguish two types of synapses: electrical and chemical (Bear, Connors, and Paradiso, 2016). They both ultimately result in depolarisation, or hyperpolarisation of the post-synaptic neuron, which either excites or inhibits the creation of a spike in the post-synaptic neuron.

2.3.2 Integrate-and-Fire Models

Scientists try to invent various models that would mimic the spike-generating mechanism of a neuron (Abbott, 1999; Hodgkin and Huxley, 1952). This subsection presents the essential spiking models from the integrate-and-fire family. These models do not attempt to imitate low-level biological and physical mechanics. They primarily focus on changes in the membrane potential of a neuron caused by the incoming current (Dayan and Abbott, 2001). Specifically, the changes in the membrane potential are described using differential equations. When the membrane potential reaches a particular threshold, a spike is generated, and the membrane potential is restored.

Perfect Integrate-and-Fire

The key to integrate-and-fire approaches is modelling the neural membrane as a capacitor (Dayan and Abbott, 2001). It acts as a capacitor because it separates the charges of the intracellular and extracellular environments. With membrane capacitance C_m, the relation between the excess charge Q and the voltage across the membrane V follows the capacitance formula:

$$Q = C_m V. \tag{2.22}$$

Using a time-derivative of this formula, we obtain an equation corresponding to the so-called perfect integrate-and-fire model:

$$\begin{aligned} C_m \frac{dV}{dt} &= \frac{dQ}{dt} \\ C_m \frac{dV}{dt} &= I_C \end{aligned} \tag{2.23}$$

This formula describes how much current $I_C = \frac{dQ}{dt}$ is needed to change the potential of a membrane with capacitance C_m at a rate of $\frac{dV}{dt}$. Apart from this relation, we only need to set a threshold V_{th} when a spike should be fired. After reaching this threshold, a spike is triggered, and the membrane potential is set to V_{reset}. Optionally, we can extend the model by introducing refractory periods. By using all of this together, we obtain a very simple model of a neuron.

Leaky Integrate-and-Fire

In order to maintain the membrane potential at a non-resting value, an additional current is required (Dayan and Abbott, 2001, p. 157). However, the perfect integrate-and-fire model does not incorporate this factor. When a current flows into the perfect integrate-and-fire model, the membrane potential is increased and stays the same even after the flow is stopped. To address this issue, we present a leaky integrate-and-fire model.

Specifically, we introduce "leakage" into the model in the form of a resistor which is connected in parallel to the capacitor. Consider a membrane resistance R_m, a small current I_R entering the neuron and a difference of potential from the resting value $\Delta V = V - V_{rest}$. The relationship between these values can be described using Ohm's law as follows: $I_R = \frac{\Delta V}{R_m}$.

By the law of current conservation, we can now combine the current I_R with I_C from Equation 2.23 as $I = I_C + I_R$ (Gerstner and Kistler, 2002). In other words, we split the total incoming current I into I_C portion, which goes through a capacitor and I_R, which goes through a resistor. By substitutions, we obtain the following:

$$I = \frac{V - V_{rest}}{R_m} + C_m \frac{dV}{dt}. \tag{2.24}$$

We multiply the equation by R_m and rearrange it into the following form:

$$\begin{aligned} C_m R_m \frac{dV}{dt} &= -(V - V_{rest}) + R_m I \\ \tau_m \frac{dV}{dt} &= V_{rest} - V + R_m I \end{aligned} \tag{2.25}$$

where τ_m is the product of membrane resistance and capacitance. It is called a membrane time constant, and it scales the changes in the membrane potential (Dayan and Abbott, 2001). Observe that when the input current I is zero, the membrane potential converges to V_{rest} exponentially with the membrane time constant τ_m (Dayan and Abbott, 2001).

Again, the leaky integrate-and-fire model is fully defined by Equation 2.25, the threshold V_{th} and the reset value V_{reset}.

Exponential Integrate-and-Fire

The leaky integrate-and-fire model can be further refined by the introduction of a non-linearity into the equation. Particularly, an exponential function is used as its behaviour was shown to fit the measured data very well (Badel et al., 2008). The change of membrane potential follows this equation:

$$\tau_m \frac{dV}{dt} = V_{rest} - V + \Delta_T \exp\left(\frac{V - V_{rh}}{\Delta_T}\right) + R_m I, \tag{2.26}$$

where Δ_T denotes the sharpness of spike initiation and V_{rh} is a so-called rheobase threshold (Gerstner and Kistler, 2002). Observe that the equation differs from the one of the leaky integrate-and-fire solely by the exponential term.

Similarly, as for the previous models, when the potential reaches a threshold V_{th}, a spike is generated, and the potential is reset. Then the integration restarts after the absolute refractory period that we choose. Note that after the spike is initiated in this model, the membrane potential reaches infinity quickly (in finite time) (Gerstner and Kistler, 2002). Thus, the threshold V_{th} is introduced rather for numerical convenience.

Chapter 3

Related Work and Data

In this chapter, we introduce the models used for our analysis: the neural network for system identification and the spiking model of a cat's primary visual cortex. Then, we describe the two datasets used in this thesis.

3.1 State-of-the-art SI Model

In this thesis, we use the state-of-the-art model for the prediction of V1 neural responses. Lurz et al. (2020) designed this model with the usual architecture of shared CNN core and a readout module. However, they invented a novel Gaussian readout with the intention of lowering the number of neuron-specific parameters. They evaluated the model on a dataset recorded from a mouse's primary visual cortex. Let us present their approach in greater detail.

3.1.1 Architecture

As stated above, the model is divided into two modules: a core and a readout. The goal of the core module is to capture non-linear representations that can be shared across all the neurons. It is implemented as a sequence of four convolutional layers. The first layer is a standard 2D convolution, whereas the rest uses the depth-separable variation. Each convolutional layer is followed by a batch normalization layer and an ELU activation function. The output of a shape $width \times height \times channels$ ($w \times h \times c$) from the last block is used as a feature tensor for the readout.

The readout module assigns a position of the receptive field for each neuron. That is, it assigns coordinates (x, y) in the feature tensor. To generate the estimated response, the feature vector at the predicted position is linearly weighted. Each neuron has its own learned weight vector of length c for this purpose. Note that the coordinates may be continuous. To address this, the model bi-linearly interpolates the four vectors neighbouring that exact position.

As it would be challenging to train this module as is, the authors introduce a special training mechanism in order to help the gradient flow. Concretely, the model assigns a bi-variate Gaussian distribution to each neuron. Thus, for i-th neuron, the network learns the coordinates μ_i, a covariance matrix Σ_i and a scaling factor s. During the training, for each neuron, image and batch, the readout position is drawn from the Gaussian distribution. This mechanism enables the gradient flow and thus allows the network to effectively learn the receptive field positions. When evaluating the model, the centre of the distribution is used to prevent non-determinism.

Moreover, their model can optionally exploit the retinotopic organization of V1 neurons - the preservation of visual field topology as explained in Section 1.3. The position of each neuron is recorded when measuring the neural responses. The readout module then transforms the actual neuron's position into the location in the feature tensor. This transformation is done by a linear fully connected neural network. That is, the network linearly projects the positions on the cortical surface into the receptive field location. Thanks to that, the number of neuron-specific parameters is further lowered. Specifically, the model with this enhancement needs to learn $c + 5$ parameters per neuron: c parameter for the linear weights of features, 4 parameters for the covariance matrix of 2D Gaussian distribution, and 1 parameter represents the scaling factor of the distribution.

3.1.2 Training

The network is trained by the Adam optimizer (Kingma and Ba, 2014) using the Poisson loss function (as described in Subsection 2.1.2). The learning rate is gradually lowered by a multiplicative factor when the correlation on the validation set do not improve for 5 consecutive epochs. When the validation performance reaches a plateau for the third time, the training is stopped and the model with the highest validation correlation is restored.

3.1.3 Results

Apart from introducing a new model, the authors examined the generalization power of the core. Particularly, they evaluated the transferability of the core to recordings of different neurons of the same subject (mouse) but also neurons of different mice. For all the core transfer experiments, the authors first train the core and a readout on a *core* set of neurons, then freeze the weights of the core and learn the parameters for a new readout that corresponds to the new unseen neurons.

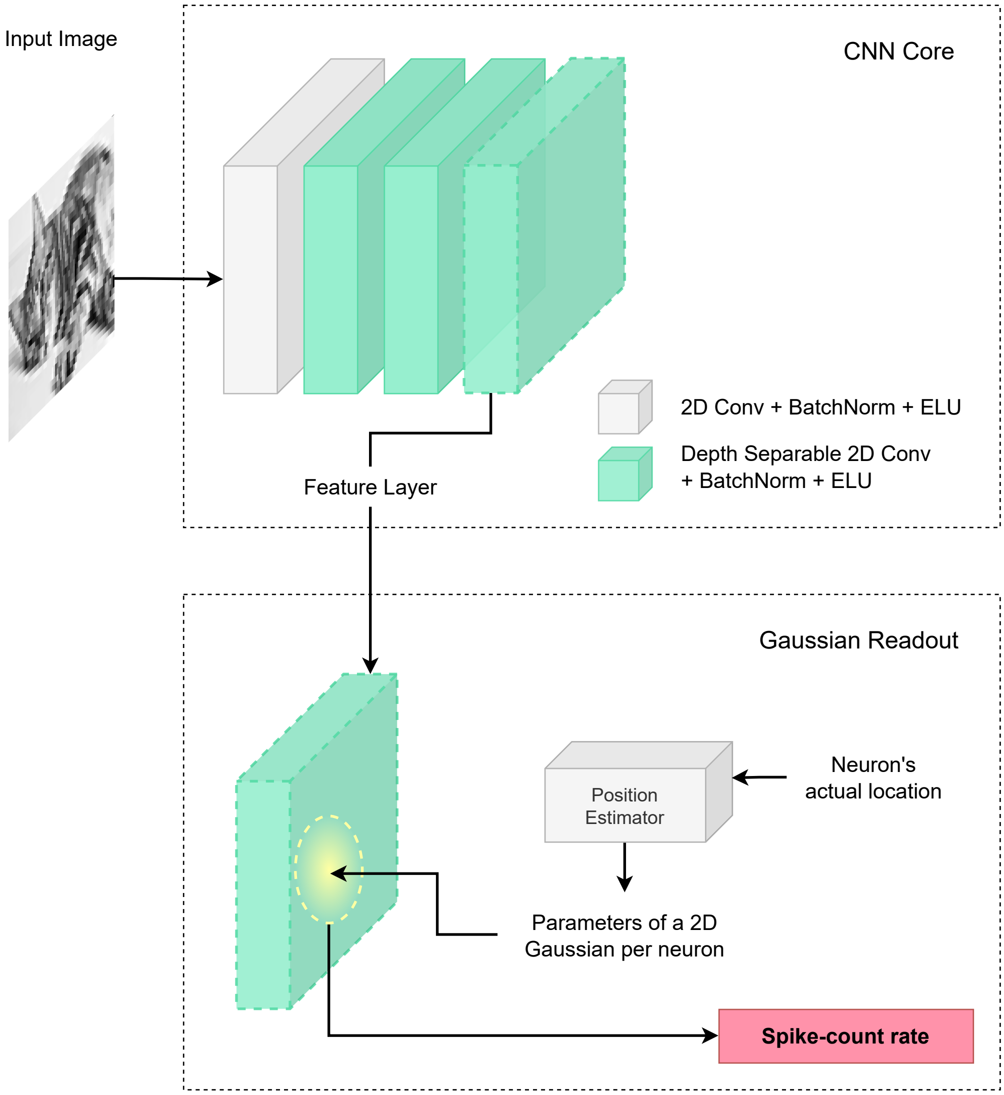


Figure 3.1 The architecture of the DNN model from Lurz et al. (2020). First, features are extracted from the image using a convolutional neural network. The position estimator then projects the neuron's actual location into the particular position in the tensor of features. Finally, the model outputs a linear combination from the selected features vector.

When training the whole model using neurons from a single individual and then evaluating the model on the same neurons (but different images), the model reaches the performance of approximately 0.886 of fraction oracle. In this case, the dataset was made of 17,596 images and 4,597 neurons.

Using the best training configuration of transferring the core to new neurons within the same animal, the model reaches a fraction oracle of 0.834. In this setup, the core was trained using data from 3,597 neurons and all 17,596 images, while the readout was trained on 4,000 images and the rest 1,000 neurons.

In their last experiment, the authors showed that using many neural recordings from different subjects leads to faster convergence when transferring the core to a new set of neurons from a different animal. Moreover, the final performance was even better than training the core directly on the set of target neurons. In other words, the representations learned during the pre-training on a greater amount of different neurons were general enough to provide better overall results.

3.2 Spiking Model of Cat Primary Visual Cortex

The focus of this thesis is an analysis of the spiking model of a cat's primary visual cortex introduced by Antolík et al. (2018). It spans the thalamocortical pathway and layers IV and II/III of the primary visual cortex. The network was designed using the current knowledge about the cat neural system from a great amount of experimental data and previous research. While the model displays inherent spontaneous activity, it can also receive visual inputs to elicit stimulus-driven activity. To assess the validity of the model, the network was tested under several stimulation paradigms, such as drifting gratings or natural stimuli. Let us provide a more comprehensive overview of the model.

3.2.1 V1 and Thalamo-Cortical Pathway

The cortical part of the model comprises a 4×4 mm patch of layers IV and II/III. That corresponds roughly to 4×4 degrees of visual field around the area centralis. Constructed from 60,000 neurons and approximately 60 million synapses, the network's scale corresponds to roughly 10% of the real neuron density. The total number of neurons is evenly distributed between layers IV and II/III, which follows the empirical evidence. Both layers contain excitatory and inhibitory neurons in a ratio of 4:1.

The neurons are modelled using an exponential integrate-and-fire approach; thus, it mostly matches the description presented in Subsection 2.3.2. The only deviation lies in the input current. Particularly, the synapses are modelled as changes of conductances rather than an input current. The change of membrane potential can be fully described by the following equation:

$$\begin{aligned}\tau_m \frac{dV}{dt} =& E_L - V + \Delta_T \exp\left(\frac{V - V_{rh}}{\Delta_T}\right) \\ &+ R_m g_{exc}(E_{exc} - V) \\ &+ R_m g_{inh}(E_{inh} - V),\end{aligned} \tag{3.1}$$

where E_L, E_{exc} and E_{inh} are respectively leakage, excitatory and inhibitory reverse potentials. g_{exc} and g_{inh} denote the incoming excitatory and inhibitory synaptic conductances. The specific parameter values were once again determined based on the latest experimental knowledge. For precise values and the rationale behind their selection, please refer to the original paper.

The model contains both feed-forward and recurrent connections. Layer IV's excitatory neurons form connections with both types of neurons in layer II/III. Both layer IV and II/III incorporate short-range intra-layer connections. These connections are sampled randomly from a distribution following two principles: the probability is higher for closer neurons, and there is a preference for connection with functionally similar neurons. Additionally, the model also accounts for long-range functionally specific connections within layers II/III. Ultimately, there is also direct feedback from layers II/III to layer IV. Even though these connections are not common in reality, this feedback exists transitively via layers V and VI (Binzegger, Douglas, and Martin, 2004).

Apart from the cortical part, the network also contains a model of LGN. All the neurons in layer IV receive inputs from this model. In order to induce the typical properties of layer IV neurons, the connectivity from LGN is sampled from a Gabor distribution. The neurons' orientation selectivity was determined by the orientation parameter of the distribution. Specifically, the parameter for each neuron was chosen based on the location of the neuron and a pre-computed orientation map. The phase of the distribution was set randomly, and the rest of the parameters (size, frequency and aspect ratio) were fixed to constant values matching the averages from experimental findings.

3.2.2 Input Model

The LGN model corresponds to 5 × 5 degrees of visual angle in order to fully cover the receptive fields of V1 neurons. Similarly, the overall input visual field corresponds to 11 × 11 degrees. The LGN neurons directly process the visual stimuli (retinal circuitry is omitted). Based on the findings of Allen and Freeman, 2006, these neurons are modelled using a spatiotemporal field. The spatial part is implemented as a difference-of-Gaussians while the temporal profile corresponds to a difference-of-gamma-functions. The visual field is uniformly covered by the LGN neurons with a density of 100 neurons per square degree. The output of LGN is then injected into the cortical neurons in the form of an electrical current.

3.2.3 Results

Thanks to the fact that the model was designed in line with the current experimental evidence, it exhibits a great range of phenomena observed in an actual visual cortex. Here, we highlight several findings that are important for our experiments. Firstly, most neurons in both layers demonstrate orientation selectivity with realistic orientation tuning curves. Additionally, by size tuning of drifting sine gratings, Antolík et al. (2018) illustrate that the neurons express surround suppression. The suppression is again observed in both layers, but the effect is stronger for layers II/III. And lastly, the majority of neurons of layer IV were classified as simple cells, whereas most neurons from layers II/III were classified as complex.

3.3 Data

The datasets used for the analysis in the thesis come from two sources. First, we use the dataset published by Lurz et al. (2020), which contains recordings of the mouse's primary visual cortex. The second dataset is an artificial dataset generated by the spiking model of a cat's primary visual cortex, which was designed by Antolík et al., 2018. The rest of the section is dedicated to the description of these datasets.

3.3.1 Mouse Dataset

The mouse dataset from Lurz et al. (2020) consists of greyscaled images from ImageNet (Russakovsky et al., 2015) and neural responses (firing rates). The images are cropped and isotropically downsampled to the size of 64×36 px with a resolution of 0.53 pixels per degree of visual angle. The responses come from layers II/III of the primary visual cortex of a mouse. See Figure 3.2 for examples of the stimulus-response pairs from this dataset.

The dataset consists of recordings of 5,335 neurons. The training partition contains responses to approximately 4,500 different stimuli, while the validation part contains about 500 stimuli. The test set was created by presenting 100 different images ten times. Thus, there are 1,000 responses for each neuron in the test set.

The stimulation protocol, the recording itself and the necessary processing of the data follow the description from Walker et al. (2019). The images are presented to one eye of the mouse on a 25-inch monitor with a resolution of 2,560 × 1,440 px. The monitor is placed 15 cm away from the eye. Each image was presented for 500 ms. A blank screen was shown as a gap between stimuli for a time interval uniformly sampled from 300 to 500 ms. The spike-count rate was then computed from a 500 ms long interval starting 50 ms after a stimulus was presented.

3.3.2 Artificial Dataset

The artificial dataset was generated using the cat primary visual cortex model from Antolík et al. (2018). The dataset shares the same composition as the mouse data, encompassing stimuli from ImageNet (Russakovsky et al., 2015) and corresponding neural responses as visible in Figure 3.3. The images were presented to the LGN model at a resolution of 220 × 220 px. The input comprises 11 × 11 degrees of the visual field resulting in a resolution of 20 pixels per degree of visual angle. The spike-count rate of neurons was calculated from the time interval of 25 to 350 ms following the stimulus presentation.

Responses from 10,000 neuron models from layers IV and II/III were recorded to create this dataset. The portion for training and validation contains 40,000 different stimuli and the corresponding responses. Similarly to the mouse data, the test set was generated by repeating the same stimuli ten times. Specifically, 500 images were used to create the test set, which results in a total of 5,000 stimulus-response pairs for each neuron.

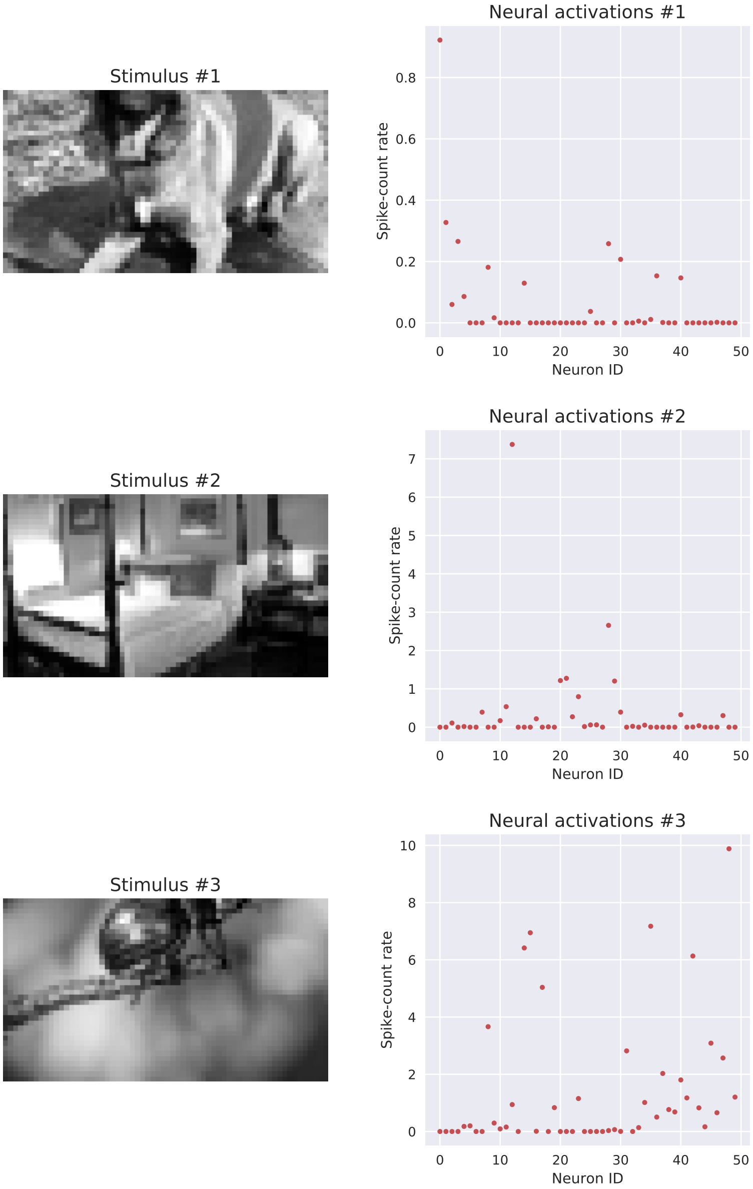


Figure 3.2 Samples from the mouse dataset published by Lurz et al. (2020). The left column contains the visual stimuli that were presented to the mouse. The plots in the right column show the responses of V1 neurons.

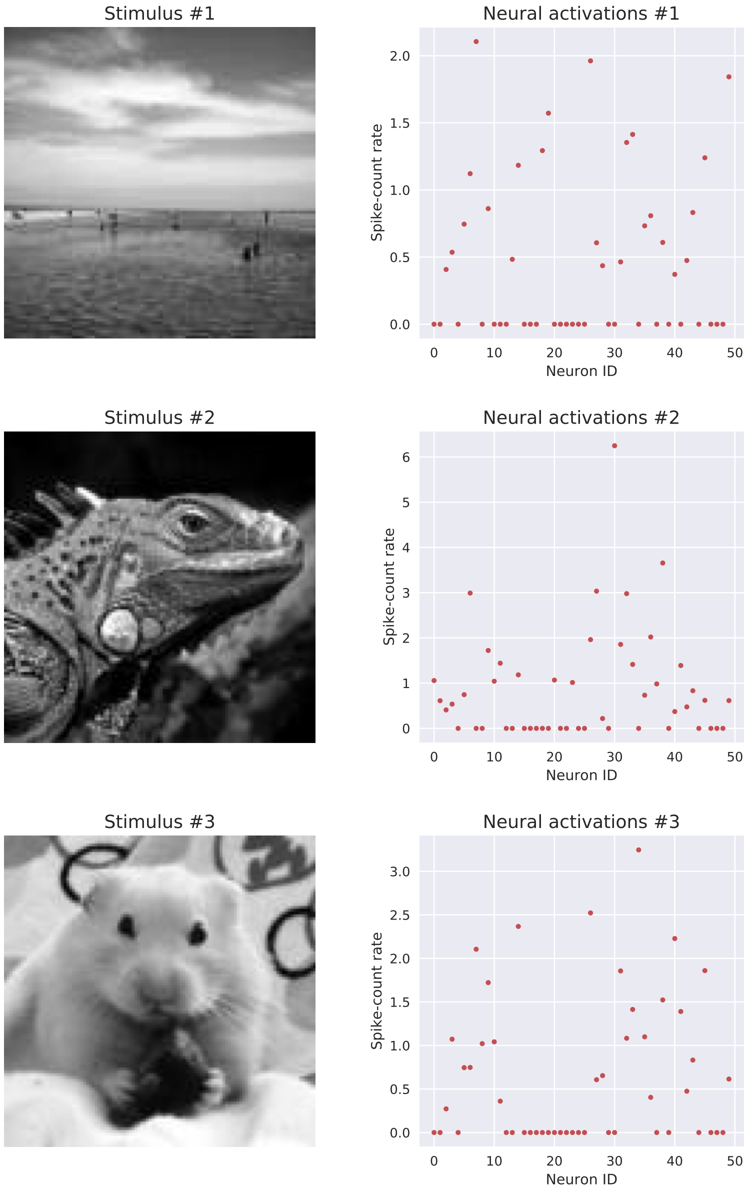


Figure 3.3 Samples from the artificial dataset. The left column contains the visual stimuli that were presented to the spiking neural network. The plots in the right column show the responses of V1 neurons.

Chapter 4

Results and Discussion

This chapter presents several analyses and experiments that we conducted in this thesis. We begin by presentation of two exploratory analyses. The first one illustrates the level of noise in the datasets, while the second one estimates and compares the sizes of receptive fields of the mouse and cat neurons. In the next section, our focus shifts towards system identification of the spiking model by means of deep neural networks. First, we tune the hyperparameters of the neural network by Lurz et al. (2020). Then, we assess the performance of goal-driven neural networks when employed for predictions on the artificial dataset. We continue with an analysis that determines the effect of the number of neurons and the number of images used for training. And we finish the section by examining the predictions made by the trained neural network to find the cause for subpar predictions. The subsequent section is dedicated to examining the representations captured by the DNN. We finish this chapter with attempts to transfer the core module of the DNN across the two datasets.

4.1 Insights into the Datasets

In this section, we present the results of two analyses conducted on the available datasets. Firstly, we evaluate the amount of noise in the datasets by computing the fraction of explainable variance of the neurons. And secondly, we analyse the datasets through spike-triggered averages. Thanks to that, we obtain linear approximations of the receptive fields. We use these approximations to estimate the sizes of the receptive fields and compare the sizes between the two datasets.

4.1.1 Noise Estimation

Before commencing the system identification process, we wanted to determine the amount of noise in the data. Or, more specifically, how big portion of the neural responses is directly caused by the presented stimuli. To estimate this value, we choose to compute the fraction of explainable variance as defined in Subsection 2.1.5. It is necessary to use a dataset with multiple trials in order to calculate the FEVs. Therefore, we use the test sets from both the mouse and artificial datasets for this purpose.

As indicated in Table 4.1, the median FEV of the artificial neurons is noticeably higher than that of the neurons in the mouse dataset, with values of 0.28 and 0.2, respectively. We also exploit the extra metadata of the artificial dataset to analyse each group of neurons individually. Specifically, we create four groups based on the cortical layer and the type of neuron. Apart from Table 4.1, you can view the FEVs as a plot of densities in Figure 4.1. We observe that the inhibitory neurons from both layers exhibit less noise than their excitatory counterparts. The difference is significantly more noticeable in layer IV, where the median FEV of the excitatory neurons is 0.24, while the inhibitory neurons have a median FEV of 0.56. For layers II/III excitatory and inhibitory neurons, the median FEVEs are 0.27 and 0.31, respectively.

Furthermore, we analysed the influence of neurons' firing rate on their FEV. First, we computed the mean firing rate from the training data for each neuron. Then we grouped the neurons into 50 bins according to their firing rate and computed the mean FEV of the neurons in the bins. Inspect Figure 4.2 for the resulting plot. Observe that the neurons with higher mean firing rates tend to have higher FEVs. To put differently, a greater portion of their responses can be attributed to the presented stimuli.

Discussion

Even though the artificial neurons exhibit noticeably less noise, the disparity is not particularly significant. For both datasets, it remains the case that less than 50% of the observed variance in the responses can be attributed solely to the presented stimuli.

Interestingly, the median FEV of layer IV excitatory neurons is slightly lower than the ones from layer II/III. This finding is somewhat counter-intuitive as layer IV neurons receive the input directly from the model of LGN. Moreover, layers II/III neurons of the model do not receive direct input from layer IV inhibitory neurons (which exhibit a relatively low level of noise) but solely from the excitatory ones. Therefore, we would expect layer IV excitatory neurons to have higher FEV than layer II/III neurons.

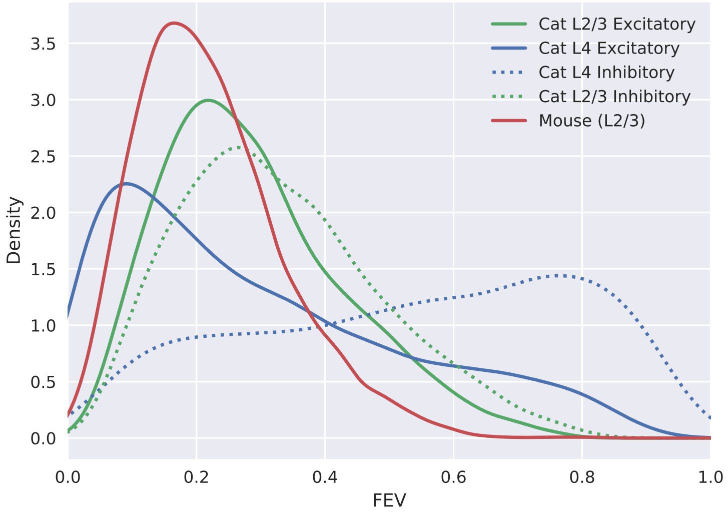


Figure 4.1 A comparison of fractions of explainable variance. The plot shows the estimated densities of neurons' FEV. The artificial (cat) neurons are divided into four different groups based on their layer and type.

Neuron Group	Mean FEV	Median FEV
Mouse (L2/3)	0.22	0.20
Cat (whole)	0.32	0.28
Cat L4 Excitatory	0.3	0.24
Cat L4 Inhibitory	0.54	0.56
Cat L2/3 Excitatory	0.29	0.27
Cat L2/3 Inhibitory	0.33	0.31

Table 4.1 Results of FEV analysis. The table shows the mean and median values of FEV for both datasets. Furthermore, we present the FEVs for individual groups of neurons from the artificial dataset. Recall that the excitatory and inhibitory neurons are represented in a ratio of 4:1.

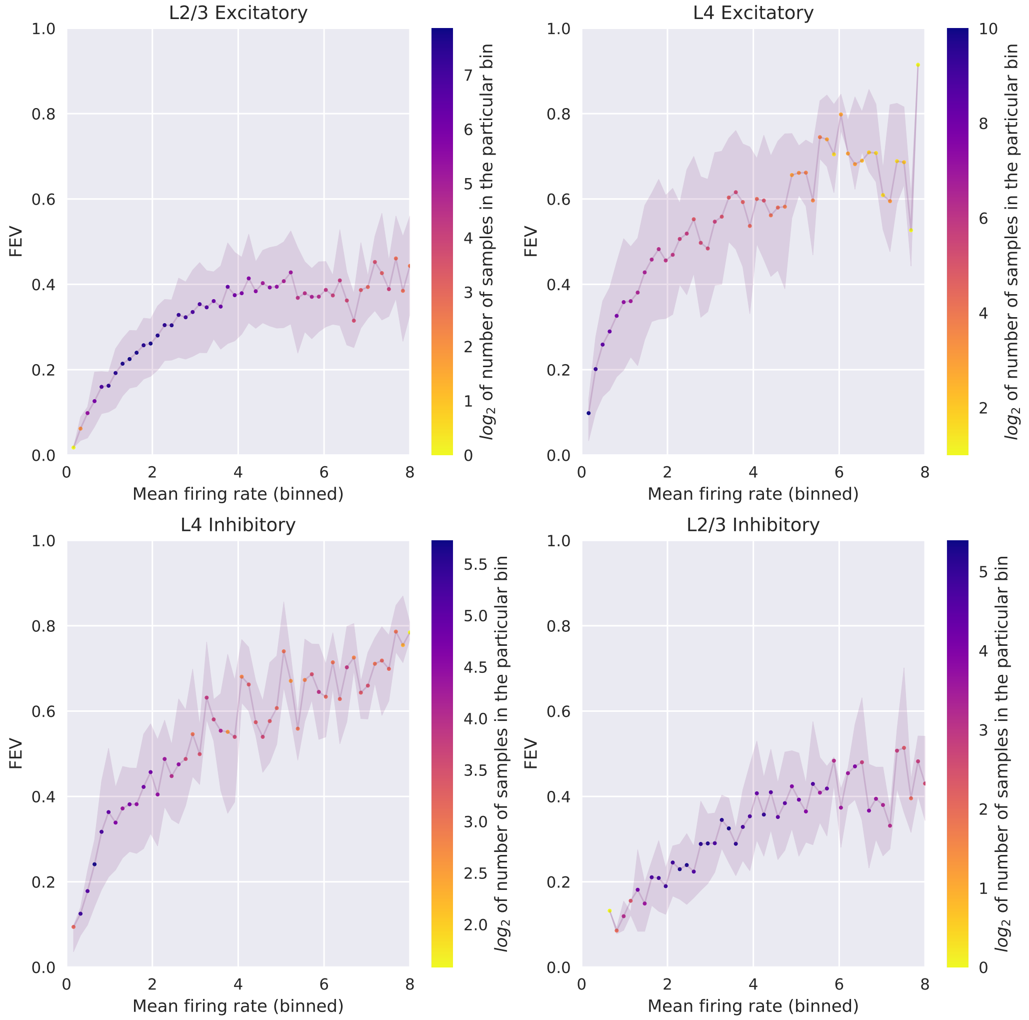


Figure 4.2 The influence of neurons' firing rate on their FEV. We binned the neurons by their mean firing rate and computed the FEVs of these bins. The dots correspond to the mean FEV of each bin, while the filled area bounds the lower and upper quartile values. The colours of the dots indicate the number of neurons in the particular bin.

4.1.2 Receptive Field Size Estimation by STA

One of our goals is to transfer the CNN core of the model by Lurz et al. (2020). If there is a substantial disparity in the sizes of receptive fields between the two datasets, the fixed size of the CNN kernels could potentially have a negative impact on the model's performance. Therefore, we want to estimate and compare the sizes of receptive fields between the mouse neurons and the spiking neurons. For this purpose, we utilised the STA analysis as described in Section 2.2.2. Besides the estimates of receptive field size, we can also examine the obtained filters for certain expected properties, such as the Gabor-like shape of simple cells.

Method

We used the regularised STA as defined by Equation 2.21 on our training sets. That way, we obtained a linear filter for each of the neurons. To estimate the size of the receptive field, we first computed the absolute value of the filter and then fitted a two-dimensional symmetrical Gaussian function on the result. We approximated the size of the receptive field as a radius of the Gaussian's central part containing 50% of its volume. To make the sizes comparable, we normalised them by the image dimensions. That is, we divided the obtained radii by $(\text{width} + \text{height})/2$. Note that we compute the receptive field sizes mainly for comparison; thus, a rough estimate is sufficient as long as the same approach is used for both datasets.

Results and Discussion

Inspect Figure 4.3 and Figure 4.4 for the samples of the STAs of the mouse neurons and the artificial neurons, respectively. We observe that the positions and sizes of receptive fields were especially well estimated for the spiking neurons. The STAs computed for the mouse neurons are significantly more noisy. However, recall that we use roughly eight times more images for the estimate of the spiking neurons. Notice that, for some neurons, this technique was able to extract a filter that greatly resembles the Gabor function.

The mouse neurons' median relative receptive field size is 0.098, and 0.092 is the relative median size for the spiking neurons. That is 4.9 px and 4.6 px when an image of size 50×50 px is used. See the box plots in Figure 4.5 for a better idea about the relative size distribution. Notice that the receptive field size of mouse neurons exhibits broader distribution, which might be partly caused by the noisy STAs.

Note that these receptive fields correspond roughly to 8.7 degrees of visual angle for the mouse and 1 degree of visual angle for the cat. However, the relative receptive field size in pixels is the important value to analyse. That is because the neural network will process images scaled to the same size. Thus, the disparity

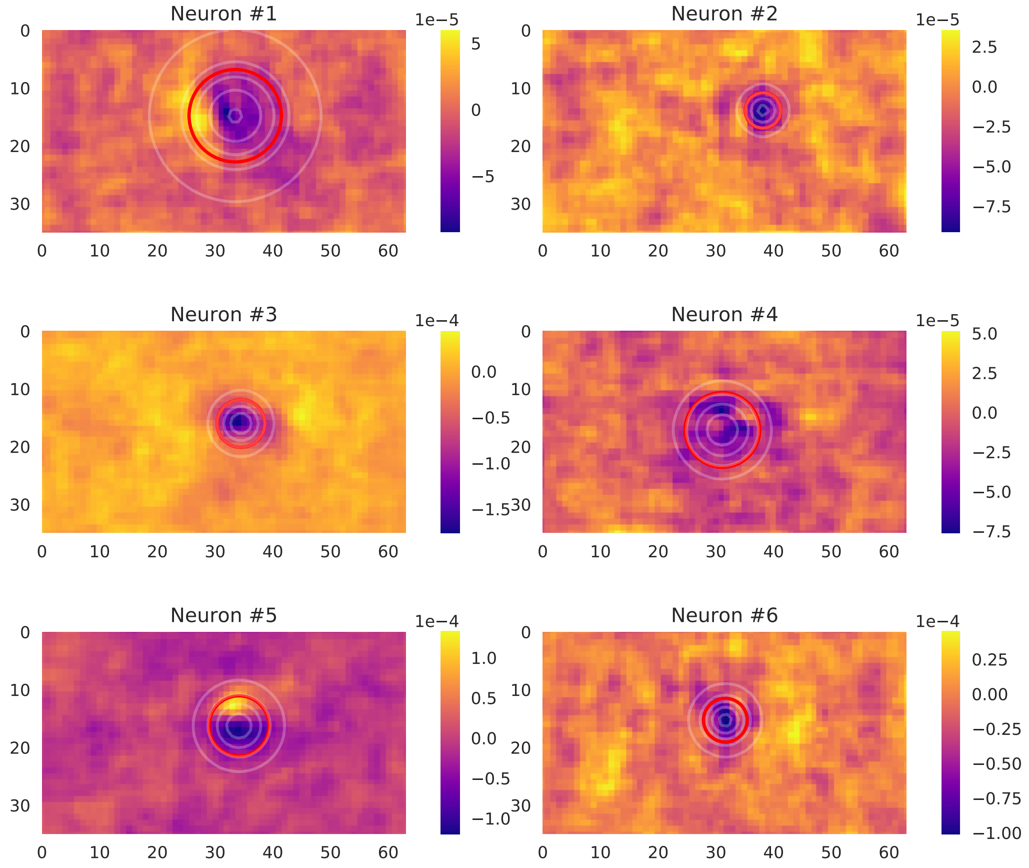


Figure 4.3 Sample STAs of neurons from the mouse dataset. The red circles indicate the estimated receptive field size as described in Subsection 4.1.2. The semitransparent circles illustrate the contours of the fitted Gaussian function.

in degrees of visual angle should not have a substantial impact. Based on the obtained results, the sizes of the CNN kernels are unlikely to be a major concern, as the receptive fields are similarly big in both datasets.

4.2 Identification of the Spiking Model

In this section, we present the results of our efforts to system identification of the spiking model of a cat's primary visual cortex from Antolík et al. (2018). First, we describe the performance of the original model from Lurz et al. (2020). Then, we perform a hyperparameter tuning in order to obtain the best model possible. We also present the performance of goal-driven networks used on the artificial dataset. Subsequently, we analyse the influence of the number of images and the

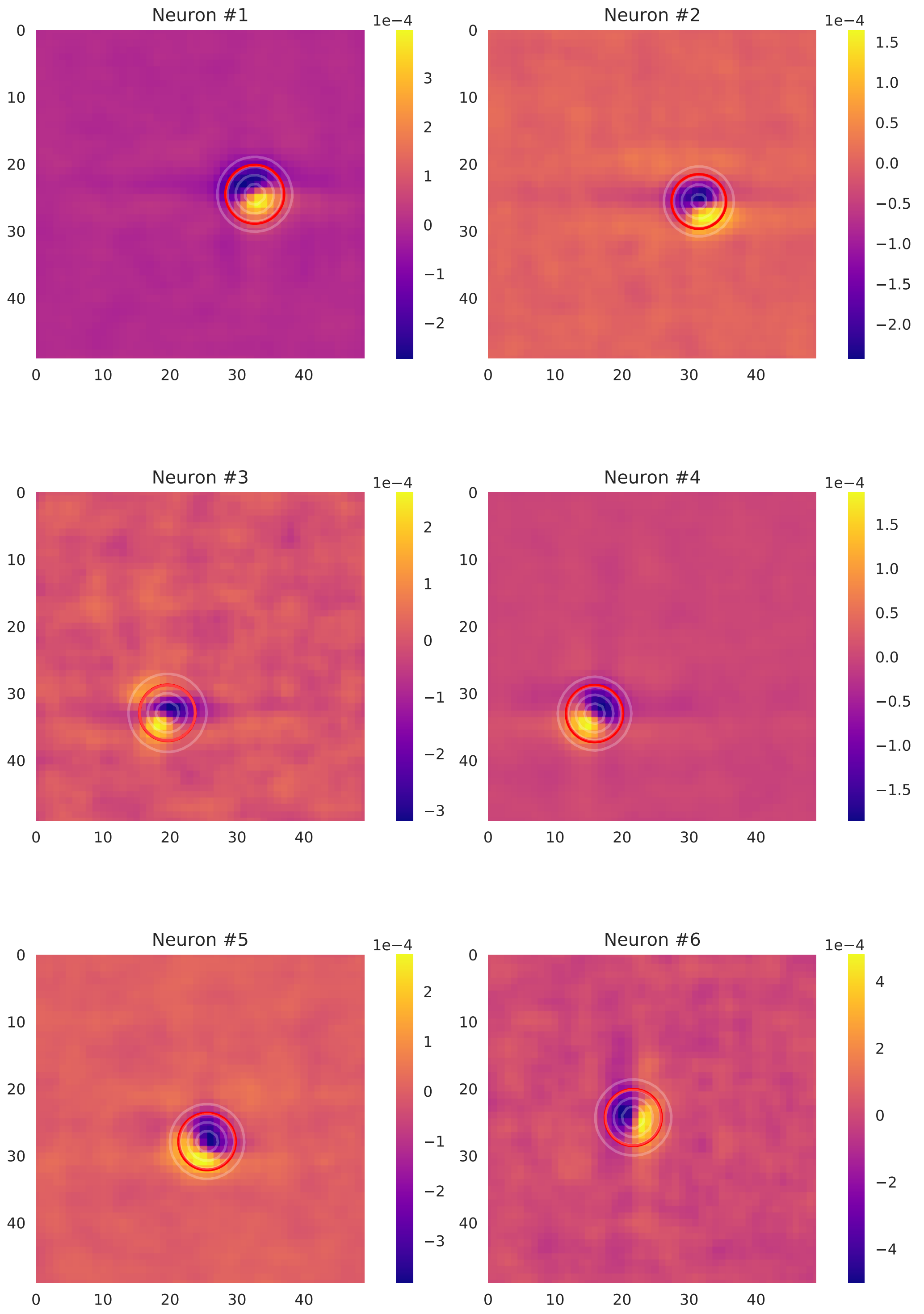


Figure 4.4 Sample STAs of neurons from the artificial cat dataset. The red circles indicate the estimated receptive field size as described in Subsection 4.1.2. The semitransparent circles illustrate the contours of the fitted Gaussian function.

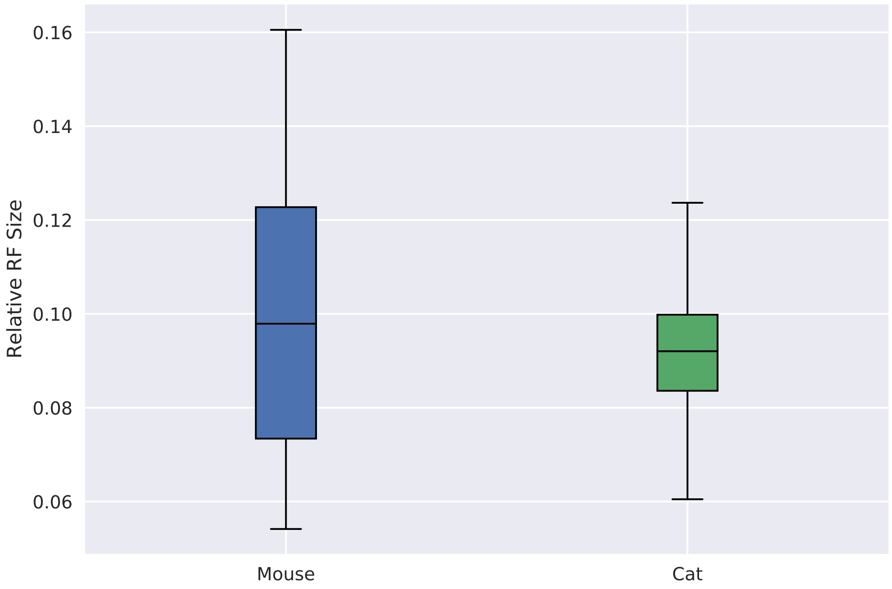


Figure 4.5 The distribution of relative receptive field sizes in the datasets. The y-axis in this plot corresponds to receptive field sizes normalised by the image size.

Hyperparameter	**Values**
Number of channels in the hidden CNNs	16, 24, 32, 64*
Hidden CNNs kernel size	11, 13*, 15, 17
Input CNN kernel size	13, 15*, 17, 19
Number of layers	3, 4*, 5

Table 4.2 The space of architecture hyperparameters that was explored. The original values of the hyperparameters are marked with asterisks.

number of neurons used for training. We finish the section with an analysis of the predictions made by our best model.

For the purpose of this section, we performed over 4,500 training runs which resulted in more than 300 hours of computational time invested throughout the course of this project. Here we present some of the interesting insights that we acquired.

Note that for better time performance, we downscaled the images from the artificial dataset to dimensions of 50 × 50 px for all the experiments in this section.

4.2.1 Hyperparameter Tuning

We begin the system identification with the model from Lurz et al. (2020) using the same set of hyperparameters as presented in their paper. Using this model directly leads to decent results. The network reaches a FEVE of 0.79 when trained on the whole dataset. Furthermore, we divided the neurons based on their cortical layers. As written in Table 4.5, the model achieves a slightly better FEVE of 0.82 on the layer II/III neurons compared to 0.78 on the layer IV neurons. This minor difference might be caused by the slightly higher FEV of the neurons in layer II/III, as will be discussed in Section 4.2.4.

We performed an extensive hyperparameter search to obtain the model that fits the artificial dataset as well as possible. After a series of experiments across various sets of hyperparameters, we decided to tune the hyperparameters in two stages. First, our focus was solely on assessing the architecture hyperparameters. Therefore, we performed a grid search on the hyperparameters described in Table 4.2, utilising a set of only 20,000 images for both training and validation purposes. Other hyperparameters were used with their original values as presented by Lurz et al. (2020). To compare the models, we utilised the correlation between the predictions and target firing rates on the validation set. We selected the five best performing models, and compared their performances with five different initializations to limit the non-determinism of the process. See the results of this stage in table 4.3.

# Channels	Hidden Kernel	Input Kernel	# Layers	Val. Corr.
24	17	17	5	0.4963
24	13	19	5	0.4936
32	17	19	4	0.4928
24	15	17	4	0.4904
24	13	13	5	0.4859

Table 4.3 The best performing models of the architecture hyperparameter search. Each row shows the model's hyperparameters: number of channels in the hidden CNN layers, size of the hidden CNN kernels, size of the input CNN kernel, and number of layers. Each configuration is evaluated using mean validation correlation computed from five runs with varying initializations.

Hyperparameter	Values
Batch size	32, 64*, 128, 256, 512, 768, 1024
Initial learning rate	0.003, 0.005*, 0.007
Number of decays steps	3*, 5, 8, 10
Decay factor	0.3*, 0.4, 0.5
Minimal learning rate	0.00001, 0.0001*

Table 4.4 The space of training hyperparameters that was explored. The original values of the hyperparameters are marked with asterisks.

Subsequently, we used the three best models from the grid search and iteratively evaluated the training hyperparameters outlined in Table 4.4. This time, we used the whole dataset of 40,000 stimuli. The best model that we discovered has 32 channels in the hidden layers, hidden CNNs kernels of size 17, input CNN kernel of size 19, and 4 layers in total. It was trained using batches of size 256, initial learning rate of 0.005, minimal learning rate of 0.00001, and 8 decays steps with a decay factor 0.4. This model achieves a FEVE of 0.85, while the model with original hyperparameters reached a FEVE of 0.79. See the comparison with additional metrics in Table 4.6.

Observe that all the metrics order the models in the same way. However, the tuned model achieved an oracle fraction higher than 1, indicating that the oracle correlation might be underestimated in the dataset.

Discussion

Our main takeaway from the previous experiments is the fact that the model achieves significantly better performance on our artificial dataset when compared

Model	FEVE	Oracle Fraction	Corr.	Training Time
Original	0.79	0.99	0.47	40m
Original (L2/3 only)	0.82	1.00	0.47	34m
Original (L4 only)	0.78	0.98	0.48	34m

Table 4.5 The performance of the original model by Lurz et al. (2020) on the artificial dataset. The table shows the performance of the model from Lurz et al. (2020) when used with its original hyperparameters. We also present the differences when the model is trained and evaluated only on a single cortical layer. For each model, we computed the fraction of explainable variance explained (FEVE), the oracle fraction, and the correlations between the predictions and responses on the test set. Moreover, we include an approximate training time when using NVIDIA V100.

Model	FEVE	Oracle Fraction	Corr.	Training Time
Original	0.79	0.99	0.47	40m
Tuned	0.85	1.03	0.50	35m
VGG16	0.73	0.96	0.47	43m
EfficientNetV2-S	0.70	0.95	0.46	13m

Table 4.6 A comparison of different models trained on the artificial dataset. The table shows performances of the model from Lurz et al. (2020) with its original hyperparameters, the model obtained from the hyperparameter tuning, and the models with goal-driven cores. For each model, we computed the fraction of explainable variance explained (FEVE), the oracle fraction, and the correlations between the predictions and responses on the test set. Moreover, we include an approximate training time when using NVIDIA V100.

to the system identification of real neurons. To illustrate, the top-performing model from Lurz et al. (2020) achieves an FEVE of approximately 0.42 on the mouse dataset, and Cadena et al., 2019a explains around 0.52% of FEV observed in responses from neurons in a macaque's primary visual cortex. In subsection 4.2.3, we show that the superior performance is not solely an impact of using a bigger dataset. These findings collectively suggest that real neurons possess certain characteristics that are currently not captured by the spiking model.

4.2.2 Goal-Driven Networks

In addition to the original design of the system identification network from Lurz et al. (2020), we also experimented with goal-driven networks as the core modules. Particularly, we used EfficientNetV2-S (Tan and Le, 2021) and VGG16 (Simonyan and Zisserman, 2015) as shared cores together with the Gaussian readout module. Both the CNN networks were pre-trained on the ImageNet object classification task (Russakovsky et al., 2015). We allowed fine-tuning of the core simultaneously with the training of the readout on the artificial dataset. To determine the best number of layers to be used as the core, we tested several options in three training runs with different random seeds.

Both networks, perform reasonably well, as can be seen in Table 4.7. The best model utilizing VGG16 reached an FEVE of 0.74. Employing EfficientNetV2-S as a core resulted in an FEVE of 0.7 for the model that achieved the best validation correlation. Both networks performed the best when only four first layers are used. Despite the subpar performance of EfficientNetV2-S, note that the training took only 13 minutes compared to 43 minutes of VGG16. Inspect table 4.6 for a comparison with the model from Lurz et al., 2020.

4.2.3 Neuron and Image Count Influence

Deep neural networks often require substantial amounts of data, while simultaneously acquiring neural recordings can be fairly challenging. In this section, we analyse the influence of the number of images and the number of neurons used for training the neural network. Specifically, we aim to answer the following questions. Firstly, we want to investigate whether extending the dataset in both dimensions can result in further improvement of our predictions. Then, we aim to determine whether augmenting the number of neurons contributes to improved performance, which would indicate that the network is effectively learning more robust representations. Finally, we want to assess whether the superior performance of the model on the artificial dataset is solely attributed to its larger size.

Model	Number of Layers	Mean Val. Corr.	Mean FEVE
VGG 16	4	**0.4778**	**0.7332**
	3	0.4712	0.7217
	5	0.4527	0.6654
	2	0.4421	0.6191
	1	0.4397	0.6137
EfficientNetV2-S	4	0.4697	0.6881
	5	0.4694	0.6767
	2	0.4683	0.7060
	3	0.4518	0.6622
	1	0.3693	0.4635

Table 4.7 A comparison of different goal-driven cores with varying number of layers. The table compares the performances of EfficientNetV2-S and VGG16 used as a core module in the system identification network. For each configuration, we report mean validation correlation and mean FEVE, both computed from three runs with different seeds.

Method

For the purpose of this experiment, we use the best model from our hyperparameter search, and we exploit the large amount of data that we have generated from the model of a cat's primary visual cortex. We trained the model using datasets with the number of images ranging from 2,500 to 40,000 and the number of neurons starting at 1,000 and going up to 10,000. For all the runs, 10% of the stimuli are used as a validation set. This results in 45 different configurations that we evaluated and averaged across five random initialisations,

Results and Discussion

We illustrate the results in the heatmap shown in Figure 4.6. As expected, the best performance was achieved by the model, which was trained on the biggest dataset. We observe that adding more images to the dataset significantly improves the performance. Similarly, increasing the number of neurons also contributes to better FEVE. The effect of the high number of neurons is mainly noticeable when not enough images are provided. However, increasing the number of neurons does not improve the model's performance when the dataset consists of a very low number of images (2,500).

To better illustrate the influence, we plotted and extrapolated the first row and the last column of the heatmap from Figure 4.6. In other words, we examined the effect of augmenting the dataset in one dimension with the other dimension at its maximum value. See Figure 4.7 for the results of this analysis. It is apparent that adding more neurons to the dataset has negligible effect, while providing more images would still improve the performance. According to the local linear extrapolation from the last three data points, adding 12.5% more images would result in an increase of FEVE by approximately 0.002 in contrast to the addition of 25% more neurons which would improve the FEVE by less than 0.001.

To conclude, augmenting the dataset in both dimensions leads to better performance. Even though further expansion of the dataset would induce slightly better results, there would still remain a significant margin to perfect predictions. Moreover, as little as 2,500 neurons and 15,000 images are sufficient to surpass the FEVE of 0.8, suggesting that the model's superior performance is not caused only by the size of the dataset.

4.2.4 Predictions Analysis

With the neural network predicting the responses of artificial neurons reasonably well, we wanted to comprehend whether there are some specific characteristics of the neurons or images where the model underperformed. With this objective in mind, we conducted several analyses to assess the impact of various properties of both the images and neurons on the quality of predictions. In this section, we present two factors that influence the predictive errors: the noise of the neuron in terms of FEV and the target firing rate within a single stimulus-response pair. We finalise this part by describing other properties of neurons and images that did not demonstrate a significant impact on the quality of predictions.

Effect of Noise on Prediction Quality: FEV vs FEVE

First, we analyse the effect of noise in the neural responses. For this purpose, we compute the fraction of explainable variance (as explained in Section 2.1.5) for each neuron and pair it with the fraction of explainable variance explained of that neuron as modelled by our best network.

See Figure 4.8, where we plotted the relation of these two values for a randomly sampled subset of neurons. Notice the evident decrease of FEVE for neurons with lower FEV. Recall our findings about the noise of neurons from specific groups that we presented in Subsection 4.1.1. Figure 4.8 clearly shows that the responses of inhibitory neurons from layer IV are the most accurately predicted. On the contrary, the network struggles to predict the responses of the excitatory neurons from layer IV as they exhibit a significantly higher noise-to-signal ratio. Furthermore, we used the same approach using the mouse dataset and obtained similar results as shown in Figure 4.9. Again, the plot shows that the model

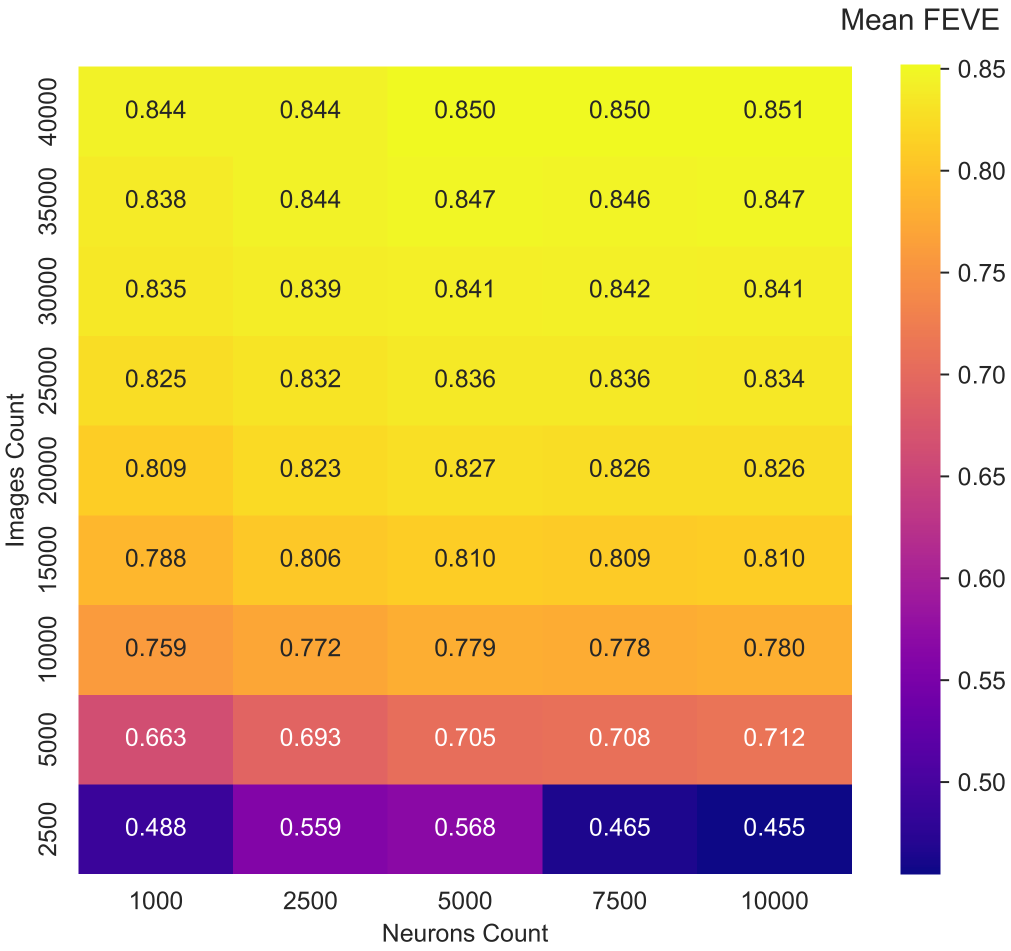


Figure 4.6 The influence of neurons and images counts on the model's performance. Each heatmap cell corresponds to the mean FEVE obtained from five runs of our best model. These runs were conducted on a dataset with the number of neurons and images determined by the horizontal and vertical axes.

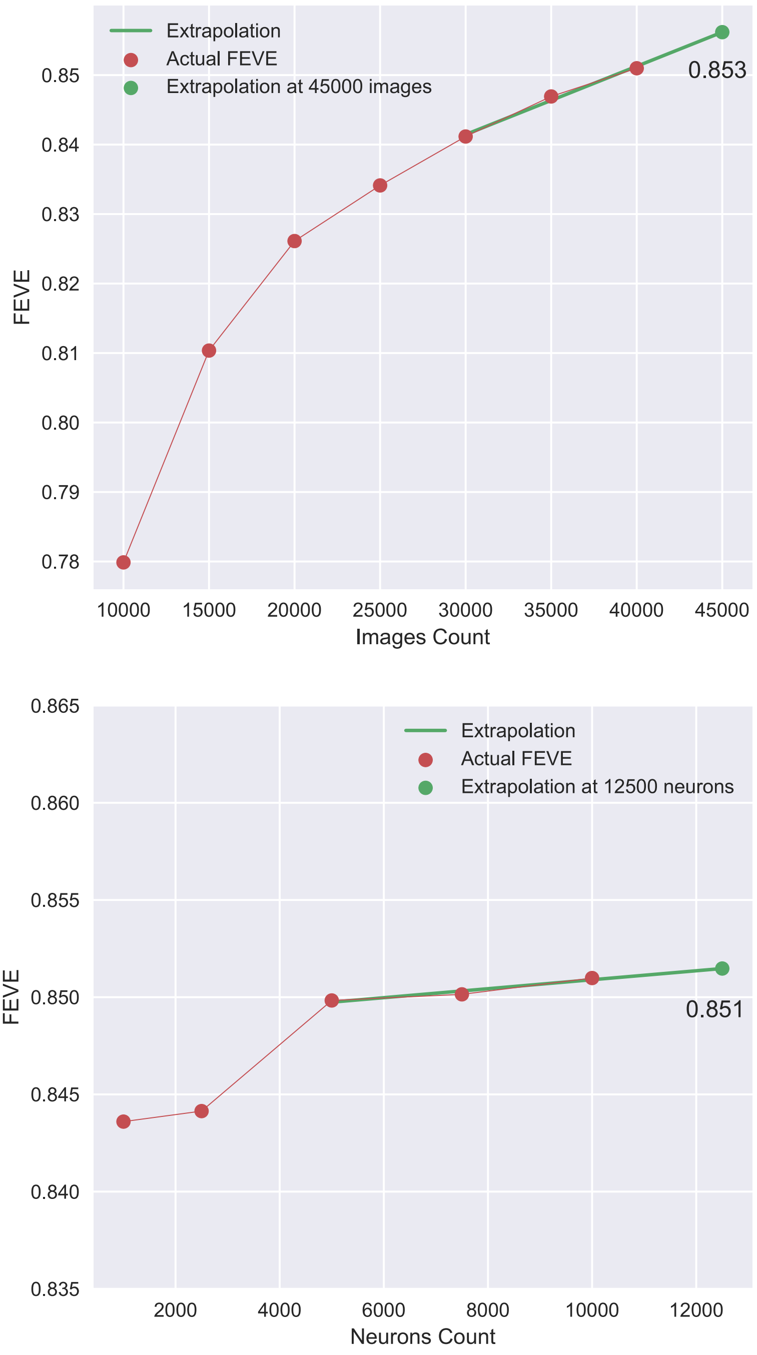


Figure 4.7 Extrapolations of the model's performance when augmenting the dataset in a single dimension. The upper plot corresponds to a dataset with all 10,000 neurons, while the number of images is based on the x-axis. Conversely, the lower plot describes a dataset with 40,000 images and a variable number of neurons. Both plots contain extrapolated values that were derived through linear interpolation of the last three data points.

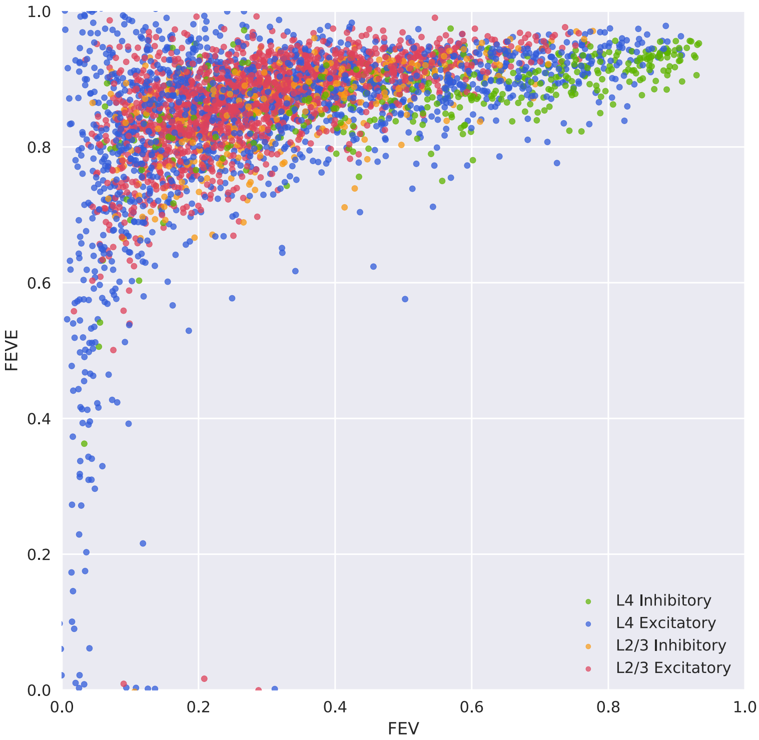


Figure 4.8 The impact of noise in the responses of neuron models on the quality of predictions. The plot illustrates the relation between the fraction of explainable variance and the fraction of explainable variance explained for a subset of neurons from the artificial dataset. The neurons are divided into groups based on their type and cortical layer.

achieves higher FEVEs for neurons with greater FEVs. In conclusion, these findings suggest that the model encounters difficulties in learning the neuron-specific weights for neurons that exhibit higher noise levels.

Firing Rate Impact on Prediction Accuracy

The following analysis unravels the influence of firing rate on the accuracy of predictions. Particularly, we found out that the model's performance gets worse for samples with higher target firing rates. See Figure 4.10, where we plotted the relationship between the target firing rate and predictions of the model. The network slightly overshoots the firing rates for the samples with low target

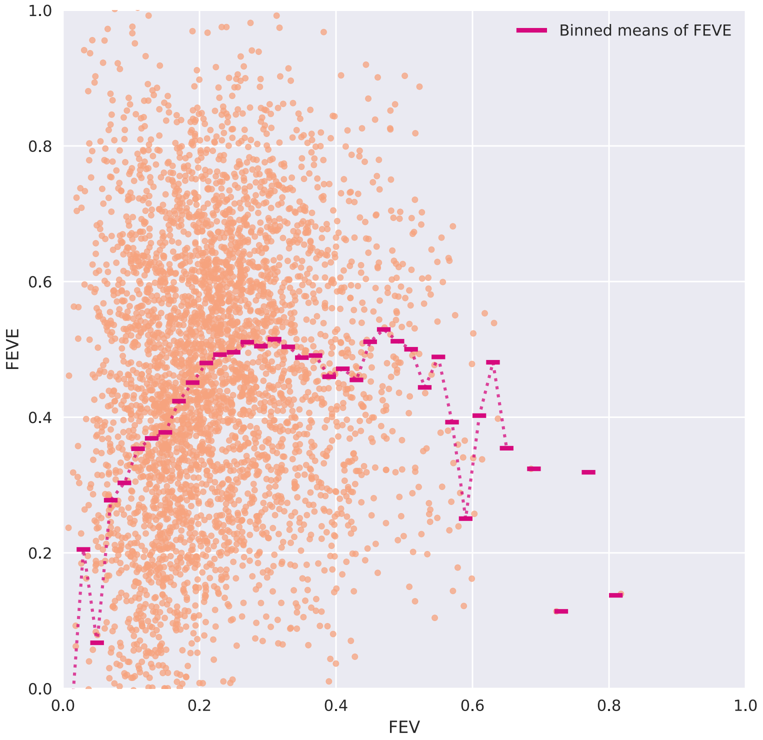


Figure 4.9 The impact of noise in the responses of mouse neurons on the quality of predictions. The plot illustrates the relationship between the fraction of explainable variance and the fraction of explainable variance explained for a subset of neurons from the mouse dataset.

firing rates. However, when the target firing rate is higher, the model starts to significantly underestimate the responses.

This finding might be a consequence of the "bi-modality" of the dataset, as roughly 50% of the target responses are equal to zero. One potential way to address this issue and improve the model could be first predicting whether the neuron should fire or not and naturally penalising the model for incorrect guesses. Then, only if the neurons fire, the firing rate prediction error would be used for learning the network parameters. By these extensions of the network and the loss function, the firing rate predicting part of the network might not be that biased towards predicting low numbers. Furthermore, the network may predict the zero responses more precisely as well.

Factors with Insignificant Influence on Predictions

Besides the two presented analyses, we inspected several other properties of neurons and images, which showed an insignificant impact on the quality of predictions. Namely, we analysed the receptive field size and position. To assess the relationship between these features and the model's performance, we conducted a correlation analysis between these values and the FEVEs of the specific neurons. However, none of these properties significantly impacted the model's performance.

Furthermore, we wanted to assess whether the frequency spectrum of the images influenced the accuracy of the predictions. That is, for example, if the images with sharp changes of intensity cause better or worse performance. For this purpose, we computed the frequency spectrum for each image using a two-dimensional fast Fourier transform and then calculated the average frequency of the image. We correlated these values with the Poisson losses obtained on the corresponding dataset samples. These values demonstrated only a low correlation. Apart from the frequencies, we also examined the effect of the image's mean pixel value by correlating it with the Poisson losses. Once again, we found no significant correlation between these two values.

4.3 Characteristics Captured by the DNN

To further understand the predictions of the deep neural network utilised for system identification, we want to explore what mechanics the network captures for individual neurons. Specifically, we chose to examine whether the neural network models some typical properties of receptive fields, such as orientation selectivity or surround suppression. We also investigate the presence of phase invariance. We perform this experiment on our artificial dataset as the spiking

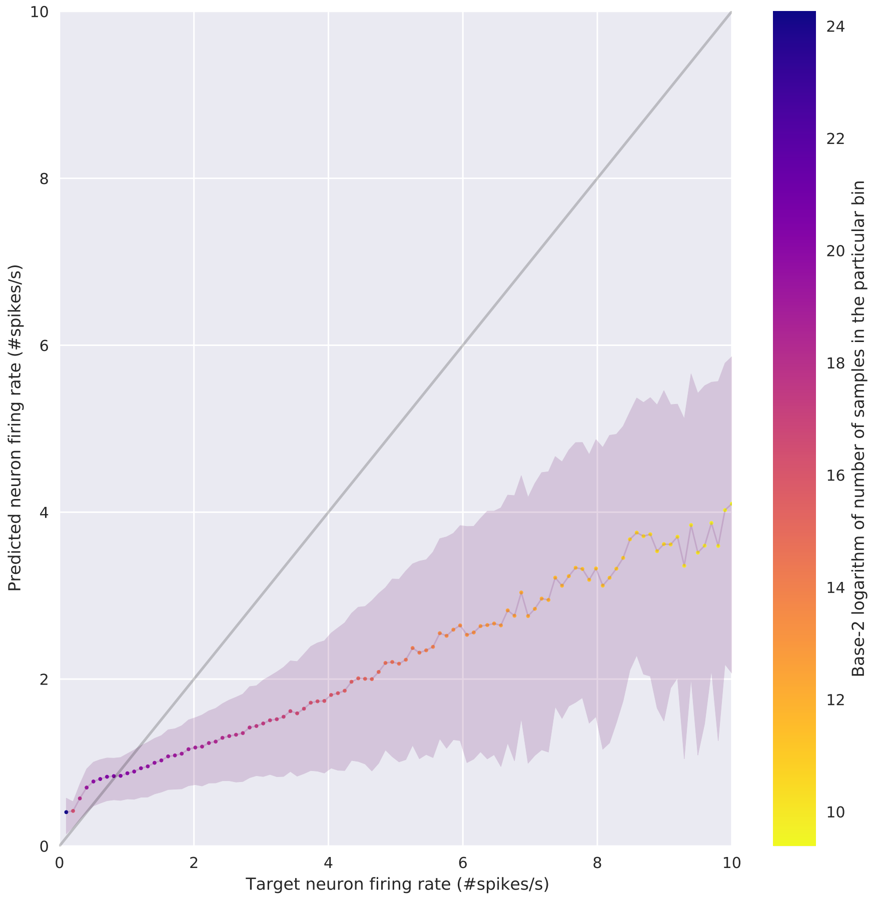


Figure 4.10 The relationship between the target firing rates and the predicted rates. All training samples were binned according to their target firing rate on the x-axis, whereas the y-axis shows the mean predictions of our best system identification model. The filled area around the means illustrates the lower and upper quartiles of the predictions.

network was purposefully designed to guarantee the manifestation of these properties.

4.3.1 Orientation, Phase and Size Tuning Methodology

To conduct this experiment, we utilise our best model from the hyperparameters tuning. Therefore, we use stimuli of size 50×50 px throughout this examination. The core idea of this analysis is to use circular patches of sinusoidal gratings as inputs to the neural network and observe the predicted responses. More specifically, for a single neuron, we first tune its phase and orientation by generating sinusoidal gratings of various orientations and phases. For this part, we used fixed radii of 4.4 degrees of visual angle (20 px) and fixed sinus frequency of 0.8 cycles per visual degree, as these values worked reasonably well in our experiments. The patches are placed at the centre of the neuron's receptive field (this information is part of the artificial dataset). We let the neural network process these stimuli and predict the corresponding responses. We mark the phase and orientation that resulted in the highest firing rate and plot all the responses as displayed in the top-left plot of Figure 4.11.

With the neuron's phase and orientation selectivity determined, we tune the size of the stimulus. That is, we again present circular patches of sinusoidal gratings, in this case, with varying sizes. We illustrate the relationship between the stimulus size and neural response as shown in the upper-right plot of Figure 4.11. We finish the examination by plotting the grating that generated the highest response and the spike-triggered average of the neuron as in the bottom plots of Figure 4.11. Thanks to that, we can compare whether the orientation of the most exciting grating matches the STA filter.

We repeated these analyses employing models with varying initialisations. Thanks to that, we can better understand the origin of the results that we observe. Similar results across different seeds suggest that the neural network truly captures the properties of the spiking network. Conversely, if the results differ significantly, it implies that the properties of receptive fields captured by the DNN do not correspond to the ones of the spiking neurons.

4.3.2 Results

Using these steps, we analysed several hand-picked neurons from the artificial dataset. Particularly, we first examined the neurons based on the quality of predictions of their responses in terms of FEVE. We discovered that the neurons that are poorly predicted exhibit vague phase and orientation tuning as depicted in Figure 4.12. Compare the result with the plot of the well-predicted neuron in Figure 4.11. Observe that the phase-orientation tuning is much more pronounced for the well-predicted neuron. Furthermore, notice that the orientation of the most exciting grating matches the corresponding spike-triggered average for the well-predicted neuron. However, for many poorly predicted neurons, the orientations do not match. Finally, observe that the neuron with high FEVE exhibits surround suppression which is visible from the top-right plot of Figure 4.11. However, most representations of the receptive fields demonstrate low or no surround modulation. See Appendix A for more receptive fields captured by the neural network.

Next, we examined the differences between neurons from layer IV and layers II/III. We observed that most layer IV neurons show phase and orientation selectivity as illustrated in Figure 4.11. Many well-predicted neurons from layers II/III demonstrate similar phase and orientation tuning patterns as those from layer IV. However, some layers II/III neurons exhibit signs of phase invariance. Figure 4.13 shows an example of such a neuron. This phase-invariance plot implies that the neuron prefers a specific orientation while being mostly agnostic to phase. Nonetheless, the phase invariance is imperfect, as the spiking diminishes for specific phases. From our observations, the quality of predictions of these phase invariant neurons is lower on average in comparison to the phase selective ones; however, we did not quantitatively evaluate this result.

Finally, the influence of differently initialised models on these analyses can be seen in Figure 4.14. The phase-orientation plots reveal mostly consistent preferences for orientation and phase. However, observe that the functions acquired through size tuning exhibit varying shapes, despite reaching the plateaus at similar sizes.

4.3.3 Discussion

This experiment brought several insights that we find interesting:

- The neural network clearly learns the orientation selectivity of the well-predicted neurons as the most exciting gratings match the STA filters. Furthermore, the captured preferences are mostly similar across different initialisations of the network, indicating that the network captures the actual properties of the spiking neurons. However, the neural network is evidently unable to capture the orientation and phase selectivity of poorly predicted neurons.

- Only a few well-predicted neurons exhibit observable surround suppression effects. Moreover, the neural network appears to have difficulty accurately capturing the precise characteristics of size tuning. This is evident from the fact that differently initialised networks capture different shapes in the size tuning plots.

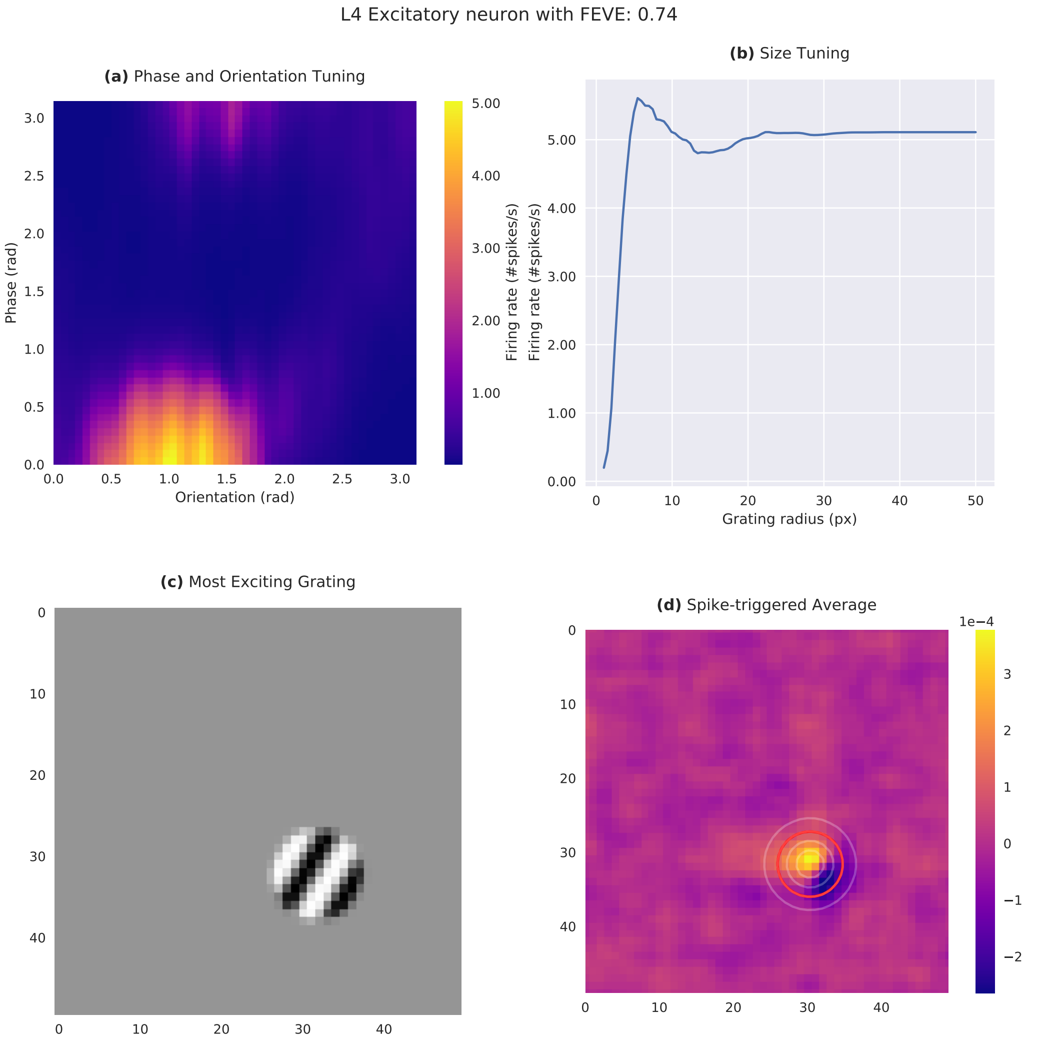


Figure 4.11 The results of orientation, phase and size tuning of a well-predicted neuron from layer IV. (a) The plot shows predicted firing rates for gratings of varying orientations and phases. **(b)** The size tuning chart obtained depicts the relationship between the size of stimuli and predicted responses. **(c)** The patch of sinusoidal grating that resulted in the highest firing rate prediction during the tuning experiments. **(d)** The spike-triggered average filter computed for the specific neuron.

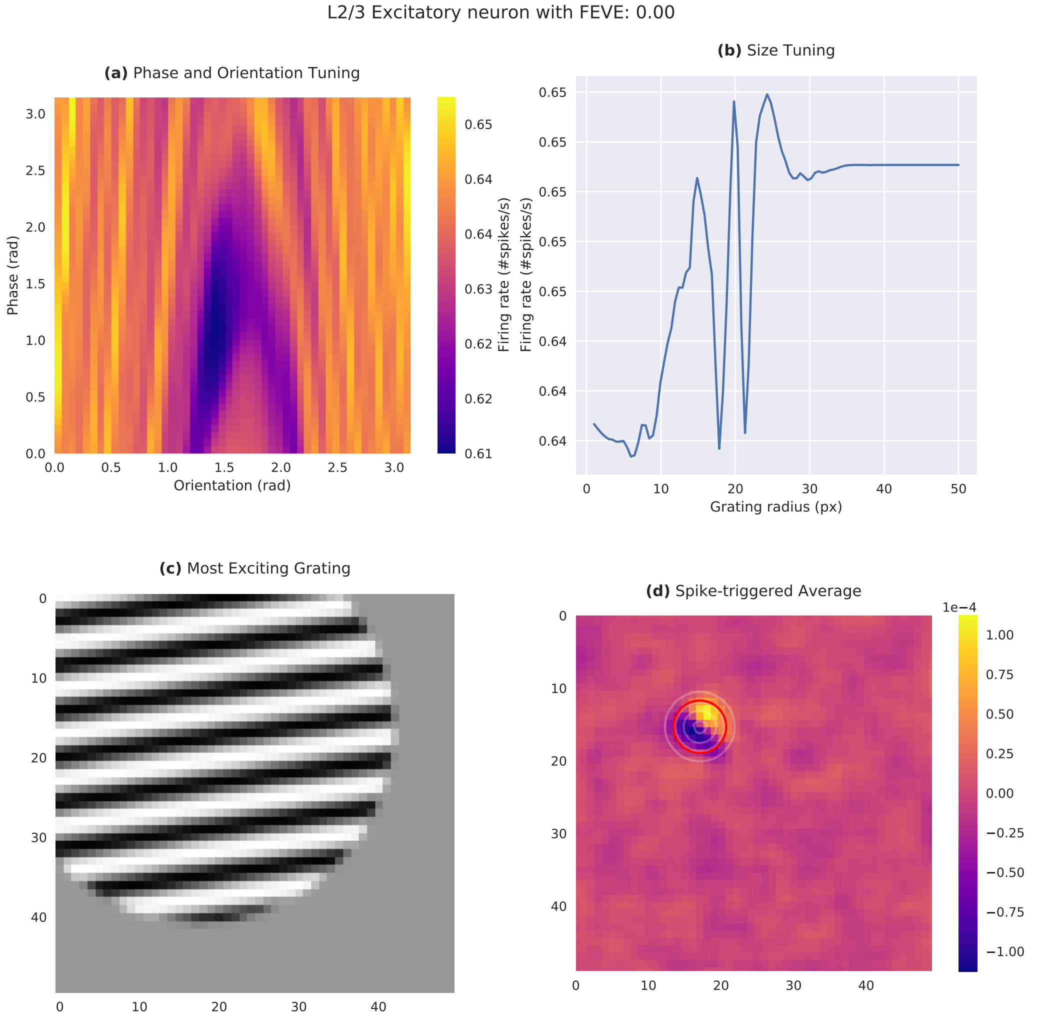


Figure 4.12 The results of orientation, phase and size tuning of a poorly predicted neuron from layers II/III. (a) The plot shows predicted firing rates for gratings of varying orientations and phases. **(b)** The size tuning chart obtained depicts the relationship between the size of stimuli and predicted responses. **(c)** The patch of sinusoidal grating that resulted in the highest firing rate prediction during the tuning experiments. **(d)** The spike-triggered average filter computed for the specific neuron.

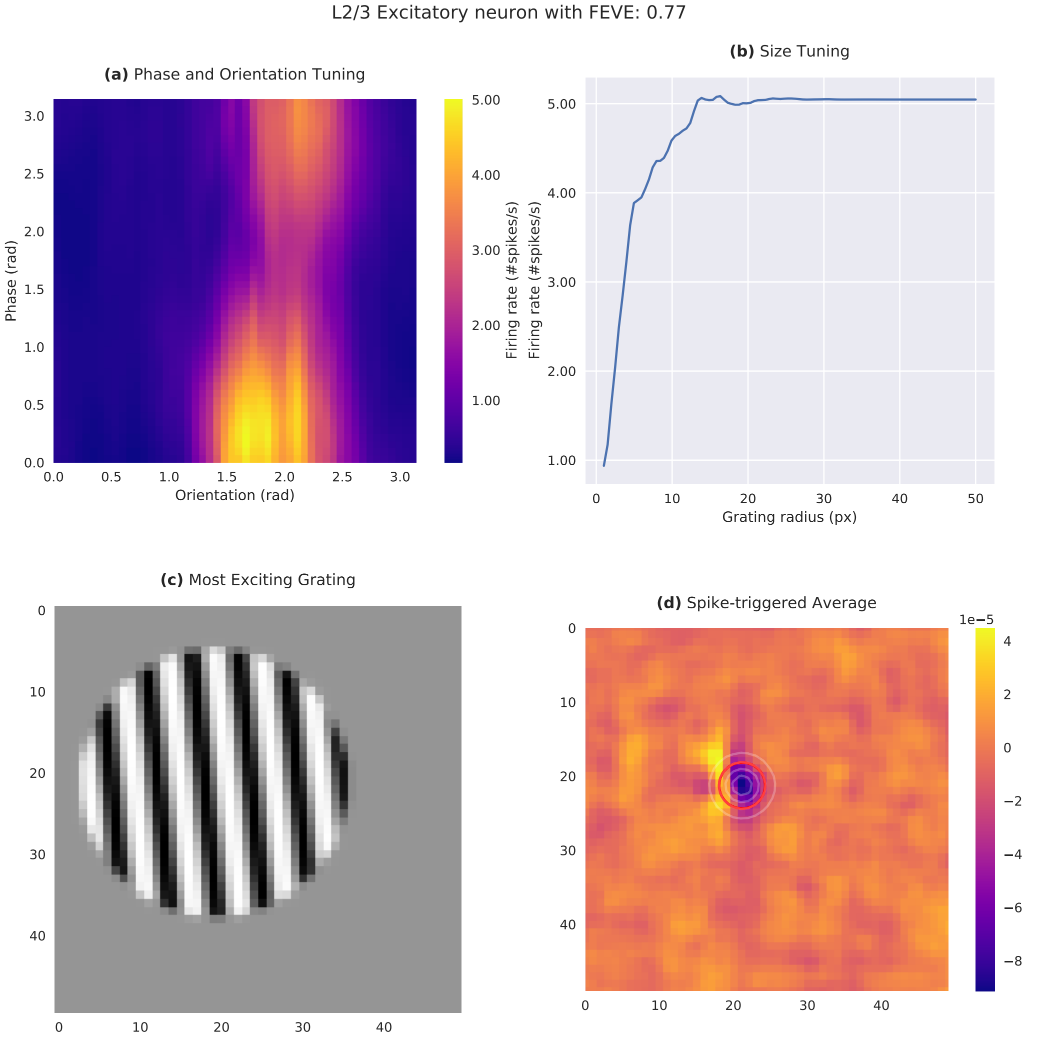


Figure 4.13 The results of orientation, phase and size tuning of a well-predicted neuron from layers II/III. (a) The plot shows predicted firing rates for gratings of varying orientations and phases. **(b)** The size tuning chart obtained depicts the relationship between the size of stimuli and predicted responses. **(c)** The patch of sinusoidal grating that resulted in the highest firing rate prediction during the tuning experiments. **(d)** The spike-triggered average filter computed for the specific neuron.

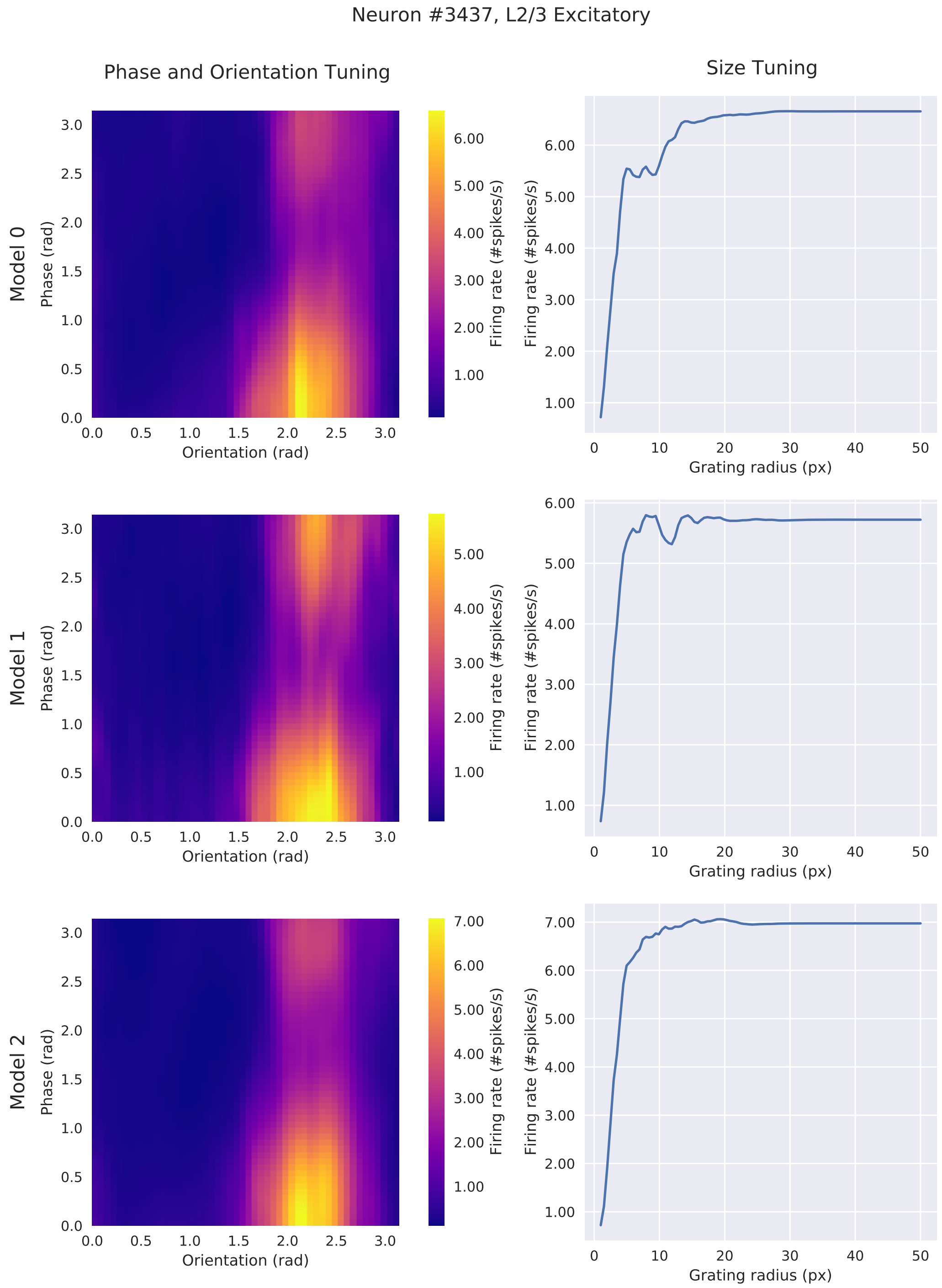


Figure 4.14 Phase, orientation and size tunings on models with varying initialisations. Each row represents a model trained with a different initialisation. The left column shows the phase-orientation selectivity plots, while the right column illustrates the size tunings.

- Most layer IV neurons captured by the network exhibit phase and orientation selectivity, suggesting simple receptive fields of these cells. This finding aligns with the analysis conducted by Antolík et al. (2018), indicating simple cell prevalence in layer IV.

- The captured representations of neurons from layers II/III demonstrate a broader range of receptive fields. Many phase-orientation tunings exhibit some level of phase invariance. However, we also observed a significant number of neurons within this layer that display similar characteristics to those in layer IV. As described in Subsection 1.3.3, phase invariance is typical for complex cells, which is again in line with the results from Antolík et al. (2018) regarding the higher occurrence of complex cells in layers II/III. Furthermore, it would be interesting to quantitatively evaluate the relationship between phase-invariance and FEVEs of the neurons. This analysis might reveal whether it is more challenging to predict responses of complex cells.

4.4 Core Transfers

Lurz et al. (2020) demonstrated successful transfers of their model's core module to a dataset with a different set of neurons, even from another subject. In this section, we explore the possibility of core transfers between the artificial and mouse datasets. Recall that the artificial dataset is generated by a model of a cat's primary visual cortex (Antolík et al., 2018). Hence, the transfers are performed across species in our case, while the transfers done by Lurz et al. (2020) were all conducted on mice.

4.4.1 Cat to Mouse

First, we explored the possibility of transferring the core pre-trained on the artificial dataset for use on the mouse data. The generated dataset contains significantly more images and neurons; thus, we expected that the core might capture some features useful for the prediction of mouse neurons' responses.

To conduct this experiment, we used the original network from Lurz et al. (2020). We downscaled the images from the artificial dataset to match the dimensions of the ones from the mouse dataset. We trained the network on the whole artificial dataset and saved the weights of the core module. Subsequently, we used this pre-trained core for learning the readout weights on the mouse dataset. Throughout this training, we froze the weights of the core. As the mouse dataset contains only neurons from layer II/III, we repeated the experiment with a core pre-trained solely using the neurons from a single cortical layer. We compared these training runs with a fixed randomly initialised core, fine-tuned randomly initialised core, and the fixed pre-trained core from Lurz et al. (2020).

As illustrated in Figure 4.15, all the cores pre-trained on the artificial dataset demonstrate comparable performance as the fixed core with random weights. The performance is notably lower than both the network trained from scratch and the network using the pre-trained core from Lurz et al. (2020).

As already mentioned, the findings from Cadena et al. (2019b) suggest that the principal goal of a mouse's primary visual cortex is not object recognition. However, we demonstrated in Section 4.2.2 that goal-driven networks trained on object recognition predict the responses from the artificial dataset fairly well. These results hint that the visual systems of the two species solve rather different tasks. However, it is still surprising that the representations captured during the training on the artificial dataset did not capture features general enough to facilitate the model's predictions compared to a core with random weights.

4.4.2 Mouse to Cat

We also transferred the core the other way. Specifically, we used the pre-trained core from Lurz et al. (2020) that was published alongside their paper. Similarly, as before, we froze the weights of this core and trained the Gaussian readout on the artificial dataset. Again, to determine the effect of different cortical layers, we trained the readouts also on datasets consisting of a single cortical layer. We compared the performances of the networks with the baselines represented by a model with a fixed randomly initialised core and a whole model trained from scratch.

See Figure 4.16 for the results of this experiment. Observe that all the training runs with the pre-trained core significantly outperform the models with fixed randomly initialised core. Nonetheless, their performance is still significantly lower than the of the network trained from scratch. The best model with a pre-trained core achieves mean FEVE of 0.44, while the model trained from scratch reaches 0.79. The division into cortical layers does not seem to make a noteworthy difference.

To our surprise, the outcomes from this experiment differ considerably compared to the previous one. These findings suggest that the network pre-trained on the mouse dataset learns fairly general representations, which results in quite accurate predictions of the artificial neurons. Nonetheless, as anticipated, the performance of the pre-trained model does not exceed the model trained on the artificial dataset from scratch.

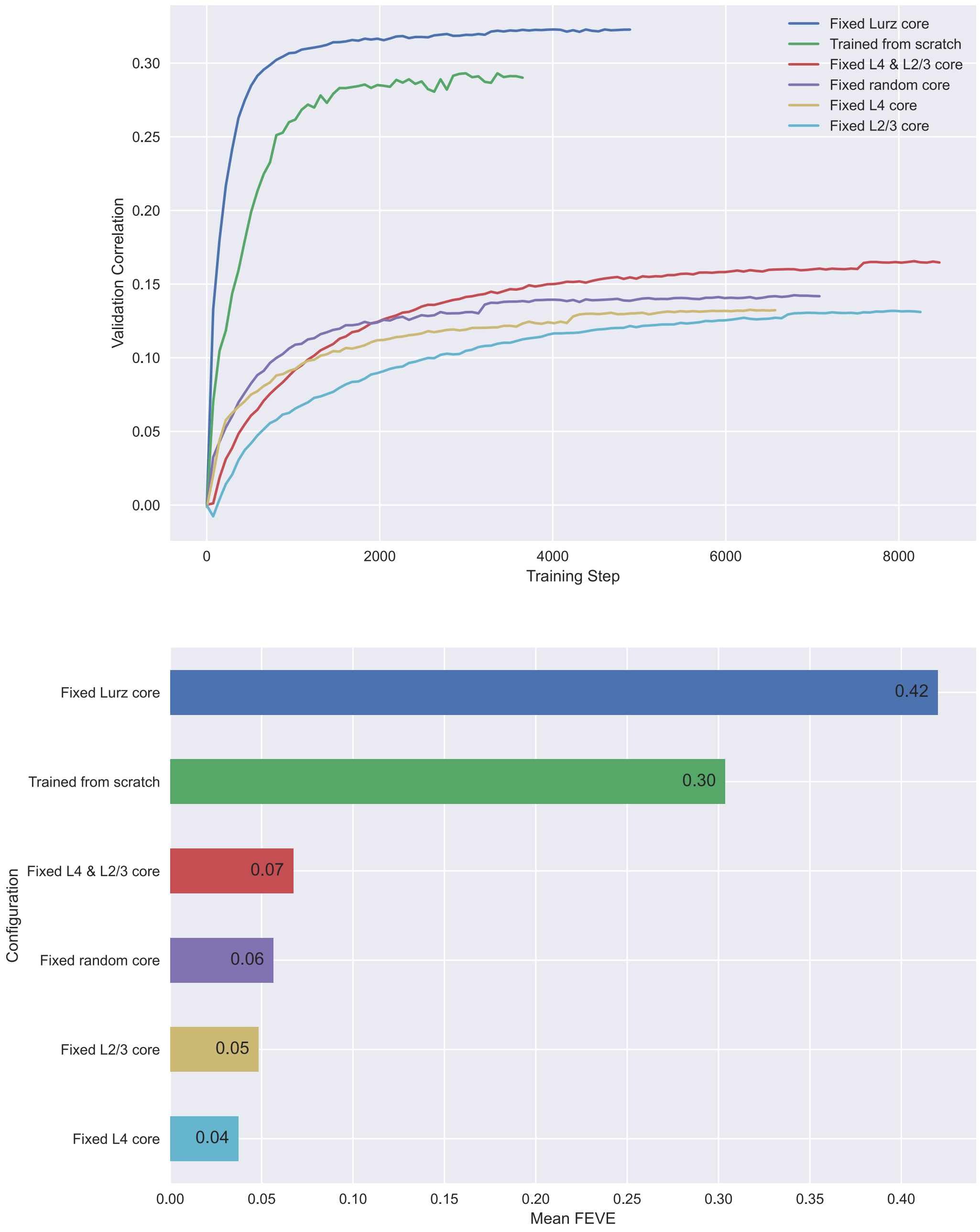


Figure 4.15 The results of transferring the core module pre-trained on the artificial dataset for use on the mouse data. The upper plot displays the progress of individual training runs, while the bottom plot shows the resulting FEVE obtained.

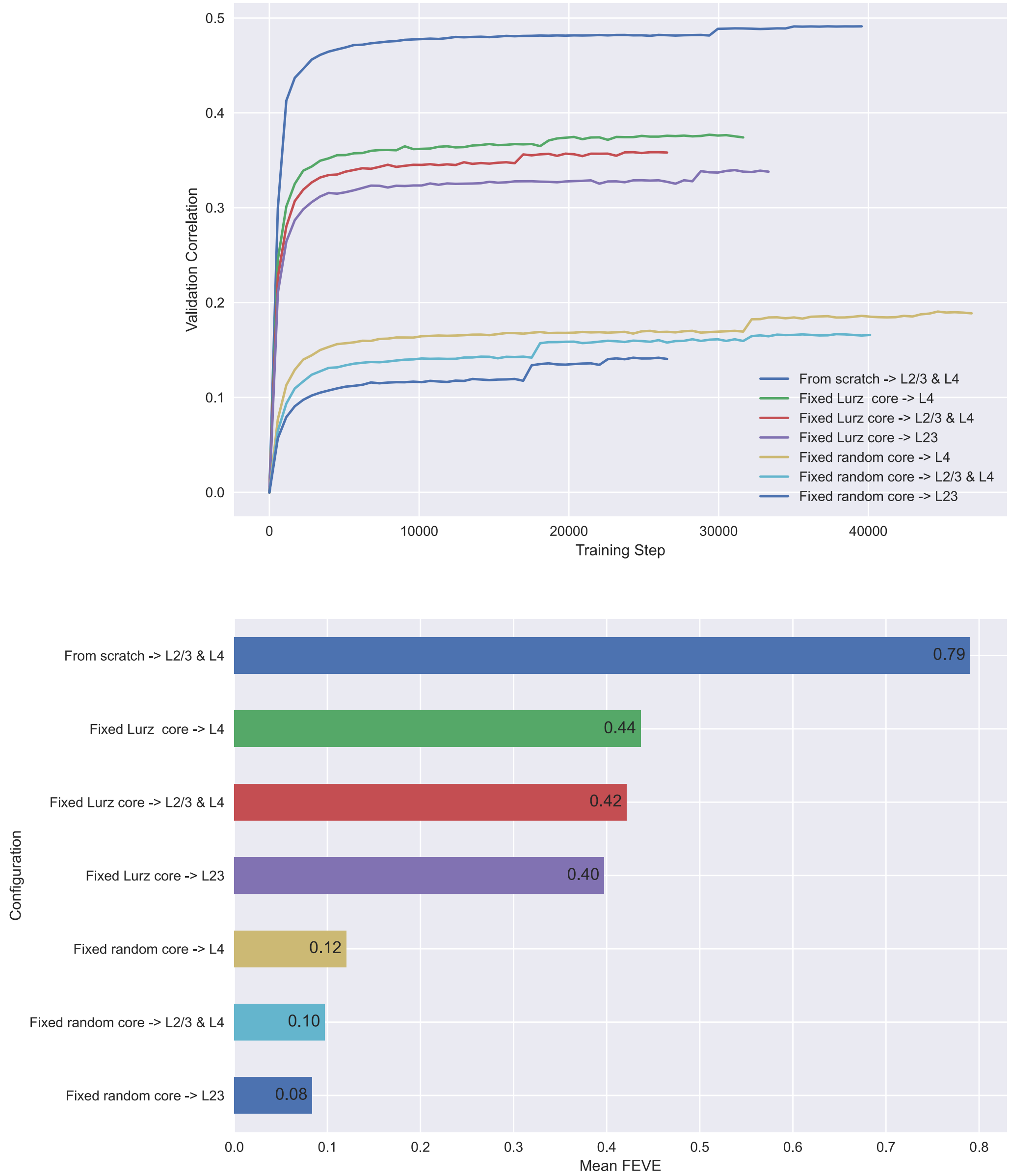


Figure 4.16 The results of transferring the core module pre-trained on the mouse dataset for use on the artificial cat data. The upper plot displays the progress of individual training runs, while the bottom plot shows the resulting FEVE obtained.

Conclusion

The thesis aimed to explore the mechanics of two different kinds of models of the visual system: a deep neural network used for system identification (Lurz et al., 2020) and a spiking model of a cat's primary visual cortex (Antolík et al., 2018). We discovered several interesting findings by using these networks simultaneously in a series of experiments and analyses.

First, we used the deep neural network from Lurz et al. (2020) for system identification of the spiking model. By tuning the hyperparameters, we successfully predicted approximately 85% of the explainable variance of the responses generated by the spiking models of neurons. Moreover, we analysed the impact of the number of neurons and the number of images used for training the network. Based on these two experiments, we conclude that the neural network, as presented by Lurz et al. (2020), would not be capable of predicting the responses perfectly even when a reasonably large amount of data is provided. We also provide an analysis of possible causes of subpar predictions on some data samples. We showed that the noise in neural responses obstructs learning adequate weights. Furthermore, we show that the network encounters difficulties in precisely predicting higher firing rates. Nonetheless, the network's performance on this dataset is significantly higher when compared to datasets made from recordings of live animals. These findings suggest that real neurons possess certain characteristics that the spiking model does not capture. We finished the system identification of the spiking model by evaluating pre-trained goal-driven networks. Specifically, we used EfficientNetV2-S (Tan and Le, 2021) and VGG16 (Simonyan and Zisserman, 2015) as core modules in combination with the Gaussian readout. These networks achieved a satisfactory fraction of explainable variance explained of 0.7 and 0.73, respectively. Nonetheless, the networks did not outperform the model from Lurz et al. (2020).

In the next set of experiments, we further analysed the representations of the system identification neural network trained on the artificial dataset. By presentation of sinusoidal gratings, we examined whether the network captures several properties of receptive fields. We found out that for well-predicted neurons from layer IV, the receptive fields captured by the network exhibit orientation and phase selectivity, indicating a prevalence of simple cells within the layer. On the other hand, the learned representations of layers II/III neurons demonstrate varying degrees of phase invariance, suggesting a higher concentration of complex cells. Furthermore, only a few receptive fields exhibit surround suppression, and the size tuning properties of the spiking neurons seem to be captured quite poorly.

We finalised the experiments by exploring the possibilities of transferring the core module of the neural network from Lurz et al. (2020). Specifically, we wanted to assess the effect of a core pre-trained on the mouse dataset and subsequently used for training on the artificial dataset and vice versa. We show that a model with a fixed core pre-trained on the mouse dataset performs significantly better than a model with a fixed, randomly initialised core. This result implies that the core captures some general features that are also usable for system identification of the spiking model of a cat's primary visual cortex. However, we demonstrate that conducting the transfer the other way leads to performance similar to using a randomly initialised core.

We believe that many of our experiments and results encourage additional research. To further improve the system identification neural network, one might investigate ways how to directly handle the great number of zero target responses in the datasets, as this seems to be one of the factors that negatively skews the network's predictions.

Another area of research might aim to assess what part of the spiking model prevents the neural network from predicting the responses perfectly. One potential approach is to systematically eliminate certain mechanisms (such as recurrent connections) from the spiking model and use the neural network for the system identification of these limited models. Then, by comparison of the neural network's performances, we might get additional insights into which specific mechanic of the spiking model makes the predictions challenging.

Furthermore, to determine the validity of the phase, orientation and size tuning patterns obtained in our experiments, it would be useful to conduct analogous analyses directly on the spiking model. That way, we would know for sure whether the learned representations correspond to the properties of the actual spiking neurons.

And finally, it would be intriguing to use the core module of the system identification network pre-trained on the spiking model of a cat's primary visual cortex for predictions of responses from a live cat. This would clarify whether the underwhelming performance of using such a pre-trained core was caused solely by the interspecies transfer. Moreover, given the ability to generate large datasets using the spiking model, this approach could improve the quality of predictions for neural responses of an actual cat's primary visual cortex.

Bibliography

Abbott, L.F (Nov. 1999). "Lapicque's Introduction of the Integrate-and-Fire Model Neuron (1907)". In: *Brain Research Bulletin* 50.5-6, pp. 303–304. ISSN: 03619230. DOI: 10.1016/S0361-9230(99)00161-6. URL: https://linkinghub.elsevier.com/retrieve/pii/S0361923099001616 (visited on 07/11/2023).

Allen, Elena A. and Ralph D. Freeman (Nov. 8, 2006). "Dynamic Spatial Processing Originates in Early Visual Pathways". In: *The Journal of Neuroscience* 26.45, pp. 11763–11774. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.3297-06.2006. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.3297-06.2006 (visited on 06/30/2023).

Angelucci, Alessandra et al. (July 25, 2017). "Circuits and Mechanisms for Surround Modulation in Visual Cortex". In: *Annual Review of Neuroscience* 40.1, pp. 425–451. ISSN: 0147-006X, 1545-4126. DOI: 10.1146/annurev-neuro-072116-031418. URL: https://www.annualreviews.org/doi/10.1146/annurev-neuro-072116-031418 (visited on 07/09/2023).

Antolík, Ján et al. (June 27, 2016). "Model Constrained by Visual Hierarchy Improves Prediction of Neural Responses to Natural Scenes". In: *PLOS Computational Biology* 12.6. Ed. by Matthias Bethge, e1004927. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1004927. URL: https://dx.plos.org/10.1371/journal.pcbi.1004927 (visited on 05/07/2022).

Antolík, Ján et al. (Sept. 24, 2018). *A Comprehensive Data-Driven Model of Cat Primary Visual Cortex.* preprint. Neuroscience. DOI: 10.1101/416156. URL: http://biorxiv.org/lookup/doi/10.1101/416156 (visited on 03/05/2022).

Badel, Laurent et al. (Feb. 2008). "Dynamic *I-V* Curves Are Reliable Predictors of Naturalistic Pyramidal-Neuron Voltage Traces". In: *Journal of Neurophysiology* 99.2, pp. 656–666. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.01107.2007. URL: https://www.physiology.org/doi/10.1152/jn.01107.2007 (visited on 06/27/2023).

Bear, Mark F., Barry W. Connors, and Michael A. Paradiso (2016). *Neuroscience: Exploring the Brain.* 4. ed. Philadelphia: Wolters Kluwer. 975 pp. ISBN: 978-1-4511-0954-2 978-0-7817-7817-6.

Benjamin, Ari S. et al. (July 19, 2018). “Modern Machine Learning as a Benchmark for Fitting Neural Responses”. In: *Frontiers in Computational Neuroscience* 12, p. 56. ISSN: 1662-5188. DOI: 10.3389/fncom.2018.00056. URL: https://www.frontiersin.org/article/10.3389/fncom.2018.00056/full (visited on 05/01/2022).

Binzegger, Tom, Rodney J. Douglas, and Kevan A. C. Martin (Sept. 29, 2004). “A Quantitative Map of the Circuit of Cat Primary Visual Cortex”. In: *The Journal of Neuroscience* 24.39, pp. 8441–8453. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.1400-04.2004. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.1400-04.2004 (visited on 06/30/2023).

Bosking, William H. et al. (Mar. 15, 1997). “Orientation Selectivity and the Arrangement of Horizontal Connections in Tree Shrew Striate Cortex”. In: *The Journal of Neuroscience* 17.6, pp. 2112–2127. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.17-06-02112.1997. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.17-06-02112.1997 (visited on 05/08/2023).

Butts, Daniel A. (Sept. 15, 2019). “Data-Driven Approaches to Understanding Visual Neuron Activity”. In: *Annual Review of Vision Science* 5.1, pp. 451–477. ISSN: 2374-4642, 2374-4650. DOI: 10.1146/annurev-vision-091718-014731. URL: https://www.annualreviews.org/doi/10.1146/annurev-vision-091718-014731 (visited on 03/05/2022).

Buzás, Péter et al. (Dec. 20, 2006). “Model-Based Analysis of Excitatory Lateral Connections in the Visual Cortex”. In: *The Journal of Comparative Neurology* 499.6, pp. 861–881. ISSN: 00219967, 10969861. DOI: 10.1002/cne.21134. URL: https://onlinelibrary.wiley.com/doi/10.1002/cne.21134 (visited on 07/14/2023).

Cadena, Santiago A. et al. (Apr. 23, 2019a). “Deep Convolutional Models Improve Predictions of Macaque V1 Responses to Natural Images”. In: *PLOS Computational Biology* 15.4. Ed. by Wolfgang Einhäuser, e1006897. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1006897. URL: https://dx.plos.org/10.1371/journal.pcbi.1006897 (visited on 05/07/2022).

Cadena, Santiago A et al. (Sept. 11, 2019b). “How Well Do Deep Neural Networks Trained on Object Recognition Characterize the Mouse Visual System?” In: p. 5.

Carandini, Matteo (Dec. 1, 2006). “What Simple and Complex Cells Compute: Classical Perspectives”. In: *The Journal of Physiology* 577.2, pp. 463–466. ISSN: 00223751. DOI: 10.1113/jphysiol.2006.118976. URL: https://onlinelibrary.wiley.com/doi/10.1113/jphysiol.2006.118976 (visited on 07/08/2023).

Carandini, Matteo et al. (Nov. 16, 2005). “Do We Know What the Early Visual System Does?” In: *The Journal of Neuroscience* 25.46, p. 10577. DOI: 10.1523/

JNEUROSCI.3726-05.2005. URL: http://www.jneurosci.org/content/25/46/10577.abstract.

Chichilnisky, E J (Jan. 1, 2001). "A Simple White Noise Analysis of Neuronal Light Responses". In: p. 15.

Cybenkot, G (Dec. 1, 1989). "Approximation by Superpositions of a Sigmoidal Function". In.

Dayan, Peter and L. F. Abbott (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems.* Computational Neuroscience. Cambridge, Mass: Massachusetts Institute of Technology Press. 460 pp. ISBN: 978-0-262-04199-7.

Devore, Jay L. (2012). *Probability and Statistics for Engineering and the Sciences.* Eighth edition. Boston, MA: Brooks/Cole, Cengage Learning. 687 pp. ISBN: 978-0-538-73352-6.

Felleman, D. J. and D. C. Van Essen (Jan. 1, 1991). "Distributed Hierarchical Processing in the Primate Cerebral Cortex". In: *Cerebral Cortex* 1.1, pp. 1–47. ISSN: 1047-3211, 1460-2199. DOI: 10.1093/cercor/1.1.1. URL: https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/1.1.1 (visited on 04/11/2022).

Field, David J. (July 1994). "What Is the Goal of Sensory Coding?" In: *Neural Computation* 6.4, pp. 559–601. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/neco.1994.6.4.559. URL: https://direct.mit.edu/neco/article/6/4/559-601/5794 (visited on 05/07/2022).

Fogel, I. and D. Sagi (June 1989). "Gabor Filters as Texture Discriminator". In: *Biological Cybernetics* 61.2, pp. 103–113. ISSN: 0340-1200, 1432-0770. DOI: 10.1007/BF00204594. URL: https://link.springer.com/10.1007/BF00204594 (visited on 05/13/2023).

Gerstner, Wulfram and Werner M. Kistler (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity.* Cambridge, U.K. ; New York: Cambridge University Press. 480 pp. ISBN: 978-0-521-81384-6 978-0-521-89079-3.

Goebel, Rainer, Lars Muckli, and Dae-Shik Kim (2012). "Visual System". In: *The Human Nervous System.* Elsevier, pp. 1301–1327. ISBN: 978-0-12-374236-0. DOI: 10.1016/B978-0-12-374236-0.10037-9. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780123742360100379 (visited on 03/08/2022).

Guillery, R W and S Murray Sherman (Jan. 17, 2002). "Their Role in Corticocortical Communication: Generalizations from the Visual System". In: p. 13.

Hodgkin, A. L. and A. F. Huxley (Aug. 28, 1952). "A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve". In: *The Journal of Physiology* 117.4, pp. 500–544. ISSN: 0022-3751, 1469-7793. DOI: 10.1113/jphysiol.1952.sp004764. URL: https://

onlinelibrary.wiley.com/doi/10.1113/jphysiol.1952.sp004764 (visited on 07/11/2023).

Hornik, Kurt (Mar. 1, 1991). "Approximation Capabilities of Muitilayer Feedforward Networks". In.

Hubel, D. H. and T. N. Wiesel (Feb. 1, 1961). "Integrative Action in the Cat's Lateral Geniculate Body". In: *The Journal of Physiology* 155.2, pp. 385–398. ISSN: 00223751. DOI: 10.1113/jphysiol.1961.sp006635. URL: https://onlinelibrary.wiley.com/doi/10.1113/jphysiol.1961.sp006635 (visited on 04/22/2022).

— (Jan. 1, 1962). "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex". In: *The Journal of Physiology* 160.1, pp. 106–154. ISSN: 00223751. DOI: 10.1113/jphysiol.1962.sp006837. URL: https://onlinelibrary.wiley.com/doi/10.1113/jphysiol.1962.sp006837 (visited on 04/19/2022).

— (July 28, 1977). "Ferrier Lecture - Functional Architecture of Macaque Monkey Visual Cortex". In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 198.1130, pp. 1–59. ISSN: 0080-4649, 2053-9193. DOI: 10.1098/rspb.1977.0085. URL: https://royalsocietypublishing.org/doi/10.1098/rspb.1977.0085 (visited on 04/19/2022).

Hubel, David H. and Torsten N. Wiesel (Mar. 1965). "RECEPTIVE FIELDS AND FUNCTIONAL ARCHITECTURE IN TWO NONSTRIATE VISUAL AREAS (18 AND 19) OF THE CAT". In: *Journal of Neurophysiology* 28.2, pp. 229–289. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.1965.28.2.229. URL: http://www.physiology.org/doi/10.1152/jn.1965.28.2.229 (visited on 07/09/2023).

Hyvärinen, Aapo (Apr. 2010). "Statistical Models of Natural Images and Cortical Visual Representation". In: *Topics in Cognitive Science* 2.2, pp. 251–264. ISSN: 17568757, 17568765. DOI: 10.1111/j.1756-8765.2009.01057.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1756-8765.2009.01057.x (visited on 05/07/2022).

Jain, K and Farshid Farrokhnia (Nov. 4, 1990). "Unsupervised Texture Segmentation Using Gabor Filters". In.

Jones, J. P. and L. A. Palmer (Dec. 1, 1987). "An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex". In: *Journal of Neurophysiology* 58.6, pp. 1233–1258. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.1987.58.6.1233. URL: https://www.physiology.org/doi/10.1152/jn.1987.58.6.1233 (visited on 05/13/2023).

Khaligh-Razavi, Seyed-Mahdi and Nikolaus Kriegeskorte (Nov. 6, 2014). "Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation". In: *PLoS Computational Biology* 10.11. Ed. by Jörn Diedrichsen, e1003915. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003915. URL:

https://dx.plos.org/10.1371/journal.pcbi.1003915 (visited on 07/08/2023).

Kindel, William F., Elijah D. Christensen, and Joel Zylberberg (June 19, 2017). *Using Deep Learning to Reveal the Neural Code for Images in Primary Visual Cortex.* arXiv: 1706.06208 [cs, q-bio]. URL: http://arxiv.org/abs/1706.06208 (visited on 06/10/2023). preprint.

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". Version 9. In: DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980 (visited on 07/16/2023).

Klindt, David A. et al. (Jan. 29, 2018). "Neural System Identification for Large Populations Separating "What" and "Where"". arXiv: 1711.02653 [cs, q-bio, stat]. URL: http://arxiv.org/abs/1711.02653 (visited on 03/05/2022).

Kriegeskorte, Nikolaus (Nov. 24, 2015). "Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing". In: *Annual Review of Vision Science* 1.1, pp. 417–446. ISSN: 2374-4642, 2374-4650. DOI: 10.1146/annurev-vision-082114-035447. URL: https://www.annualreviews.org/doi/10.1146/annurev-vision-082114-035447 (visited on 05/14/2022).

Lamme, Va (Feb. 1, 1995). "The Neurophysiology of Figure-Ground Segregation in Primary Visual Cortex". In: *The Journal of Neuroscience* 15.2, pp. 1605–1615. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.15-02-01605.1995. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.15-02-01605.1995 (visited on 07/09/2023).

Lindsay, Grace W. (Sept. 1, 2021). "Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future". In: *Journal of Cognitive Neuroscience* 33.10, pp. 2017–2031. ISSN: 0898-929X, 1530-8898. DOI: 10.1162/jocn_a_01544. URL: https://direct.mit.edu/jocn/article/33/10/2017/97402/Convolutional-Neural-Networks-as-a-Model-of-the (visited on 07/08/2023).

Lipton, Zachary C (2018). "In Machine Learning, the Concept of Interpretability Is Both Important and Slippery." In: *machine learning.*

Lund, J S (1988). "Anatomical Organization of Macaque Monkey Striate Visual Cortex". In: p. 36.

Lurz, Konstantin-Klemens et al. (Oct. 7, 2020). *Generalization in Data-Driven Models of Primary Visual Cortex.* preprint. Neuroscience. DOI: 10.1101/2020.10.05.326256. URL: http://biorxiv.org/lookup/doi/10.1101/2020.10.05.326256 (visited on 03/05/2022).

Marĉelja, S. (Nov. 1, 1980). "Mathematical Description of the Responses of Simple Cortical Cells*". In: *Journal of the Optical Society of America* 70.11, p. 1297. ISSN: 0030-3941. DOI: 10.1364/JOSA.70.001297. URL: https://opg.optica.org/abstract.cfm?URI=josa-70-11-1297 (visited on 05/13/2023).

Movshon, J A, I D Thompson, and D J Tolhurst (Oct. 1, 1978). “Receptive Field Organization of Complex Cells in the Cat’s Striate Cortex.” In: *The Journal of Physiology* 283.1, pp. 79–99. ISSN: 00223751. DOI: 10.1113/jphysiol.1978.sp012489. URL: https://onlinelibrary.wiley.com/doi/10.1113/jphysiol.1978.sp012489 (visited on 07/08/2023).

Nothdurft, Hans-Christoph, Jack L. Gallant, and David C. Van Essen (May 2000). “Response Profiles to Texture Border Patterns in Area V1”. In: *Visual Neuroscience* 17.3, pp. 421–436. ISSN: 0952-5238, 1469-8714. DOI: 10.1017/S0952523800173092. URL: https://www.cambridge.org/core/product/identifier/S0952523800173092/type/journal_article (visited on 07/09/2023).

Nurminen, Lauri and Alessandra Angelucci (Nov. 2014). “Multiple Components of Surround Modulation in Primary Visual Cortex: Multiple Neural Circuits with Multiple Functions?” In: *Vision Research* 104, pp. 47–56. ISSN: 00426989. DOI: 10.1016/j.visres.2014.08.018. URL: https://linkinghub.elsevier.com/retrieve/pii/S0042698914002053 (visited on 07/09/2023).

Piscopo, D. M. et al. (Mar. 13, 2013). “Diverse Visual Features Encoded in Mouse Lateral Geniculate Nucleus”. In: *Journal of Neuroscience* 33.11, pp. 4642–4656. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.5187-12.2013. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.5187-12.2013 (visited on 04/10/2022).

Pospisil, Dean A. and Wyeth Bair (Aug. 4, 2021). “The Unbiased Estimation of the Fraction of Variance Explained by a Model”. In: *PLOS Computational Biology* 17.8. Ed. by Frédéric E. Theunissen, e1009212. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1009212. URL: https://dx.plos.org/10.1371/journal.pcbi.1009212 (visited on 05/14/2022).

Puth, Marie-Therese, Markus Neuhäuser, and Graeme D. Ruxton (July 2014). “Effective Use of Pearson’s Product–Moment Correlation Coefficient”. In: *Animal Behaviour* 93, pp. 183–189. ISSN: 00033472. DOI: 10.1016/j.anbehav.2014.05.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0003347214002127 (visited on 05/29/2022).

Russakovsky, Olga et al. (Dec. 2015). “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115.3, pp. 211–252. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-015-0816-y. URL: http://link.springer.com/10.1007/s11263-015-0816-y (visited on 07/03/2023).

Simoncelli, Eero P and Bruno A Olshausen (Mar. 2001). “Natural Image Statistics and Neural Representation”. In: *Annual Review of Neuroscience* 24.1, pp. 1193–1216. ISSN: 0147-006X, 1545-4126. DOI: 10.1146/annurev.neuro.24.1.1193. URL: https://www.annualreviews.org/doi/10.1146/annurev.neuro.24.1.1193 (visited on 05/07/2022).

Simonyan, Karen and Andrew Zisserman (Apr. 10, 2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition.* arXiv: 1409.1556 [cs]. URL: http://arxiv.org/abs/1409.1556 (visited on 07/03/2023). preprint.

Tan, Mingxing and Quoc V. Le (June 23, 2021). *EfficientNetV2: Smaller Models and Faster Training.* arXiv: 2104.00298 [cs]. URL: http://arxiv.org/abs/2104.00298 (visited on 07/03/2023). preprint.

Ts'o, Dy, Cd Gilbert, and Tn Wiesel (Apr. 1, 1986). "Relationships between Horizontal Interactions and Functional Architecture in Cat Striate Cortex as Revealed by Cross-Correlation Analysis". In: *The Journal of Neuroscience* 6.4, pp. 1160–1170. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.06-04-01160.1986. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.06-04-01160.1986 (visited on 04/19/2022).

Ungerleider, L (1994). "'What' and 'where' in the Human Brain". In: *Current Opinion in Neurobiology* 4.2, pp. 157–165. ISSN: 09594388. DOI: 10.1016/0959-4388(94)90066-3. URL: https://linkinghub.elsevier.com/retrieve/pii/0959438894900663 (visited on 04/11/2022).

Walker, Edgar Y. et al. (Dec. 2019). "Inception Loops Discover What Excites Neurons Most Using Deep Predictive Models". In: *Nature Neuroscience* 22.12, pp. 2060–2065. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-019-0517-x. URL: http://www.nature.com/articles/s41593-019-0517-x (visited on 03/05/2022).

Willeke, Konstantin F. et al. (June 17, 2022). *The Sensorium Competition on Predicting Large-Scale Mouse Primary Visual Cortex Activity.* arXiv: 2206.08666 [cs, q-bio]. URL: http://arxiv.org/abs/2206.08666 (visited on 04/30/2023). preprint.

Willmore, Ben and Darragh Smyth (2003). "Methods for First-Order Kernel Estimation: Simple-Cell Receptive Fields from Responses to Natural Scenes". In.

Wu, Michael C.-K., Stephen V. David, and Jack L. Gallant (July 21, 2006). "COMPLETE FUNCTIONAL CHARACTERIZATION OF SENSORY NEURONS BY SYSTEM IDENTIFICATION". In: *Annual Review of Neuroscience* 29.1, pp. 477–505. ISSN: 0147-006X, 1545-4126. DOI: 10.1146/annurev.neuro.29.051605.113024. URL: https://www.annualreviews.org/doi/10.1146/annurev.neuro.29.051605.113024 (visited on 03/05/2022).

Zalama, Eduardo et al. (May 2014). "Road Crack Detection Using Visual Features Extracted by Gabor Filters: Road Crack Detection". In: *Computer-Aided Civil and Infrastructure Engineering* 29.5, pp. 342–358. ISSN: 10939687. DOI: 10.1111/mice.12042. URL: https://onlinelibrary.wiley.com/doi/10.1111/mice.12042 (visited on 05/13/2023).

List of Figures

Appendix A

Orientation, Phase and Size Tuning of the DNN

In this appendix, we provide more plots that we obtained through our tuning experiments in Section 4.3. Specifically, we show the phase-orientation tuning plots from random neurons selected from layer IV and layers II/III in Figures A.1 and A.2, respectively. Next, we present size tuning plots of the same neurons in Figures A.3 and A.4. Lastly, we demonstrate the impact of different model initializations on the representations in Figure A.5 for a neuron from layer IV, and in Figure A.6 for a layers II/III neuron.

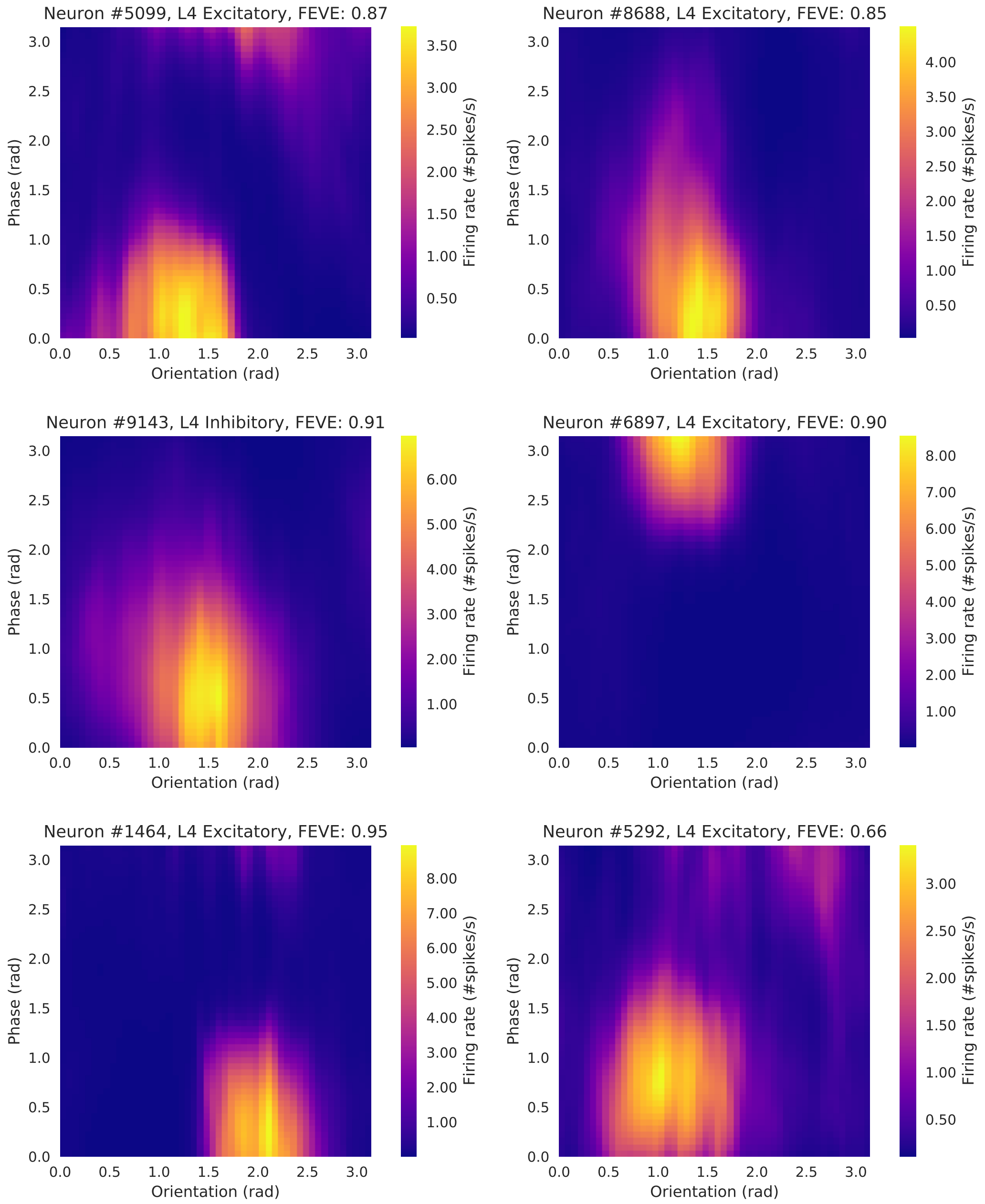
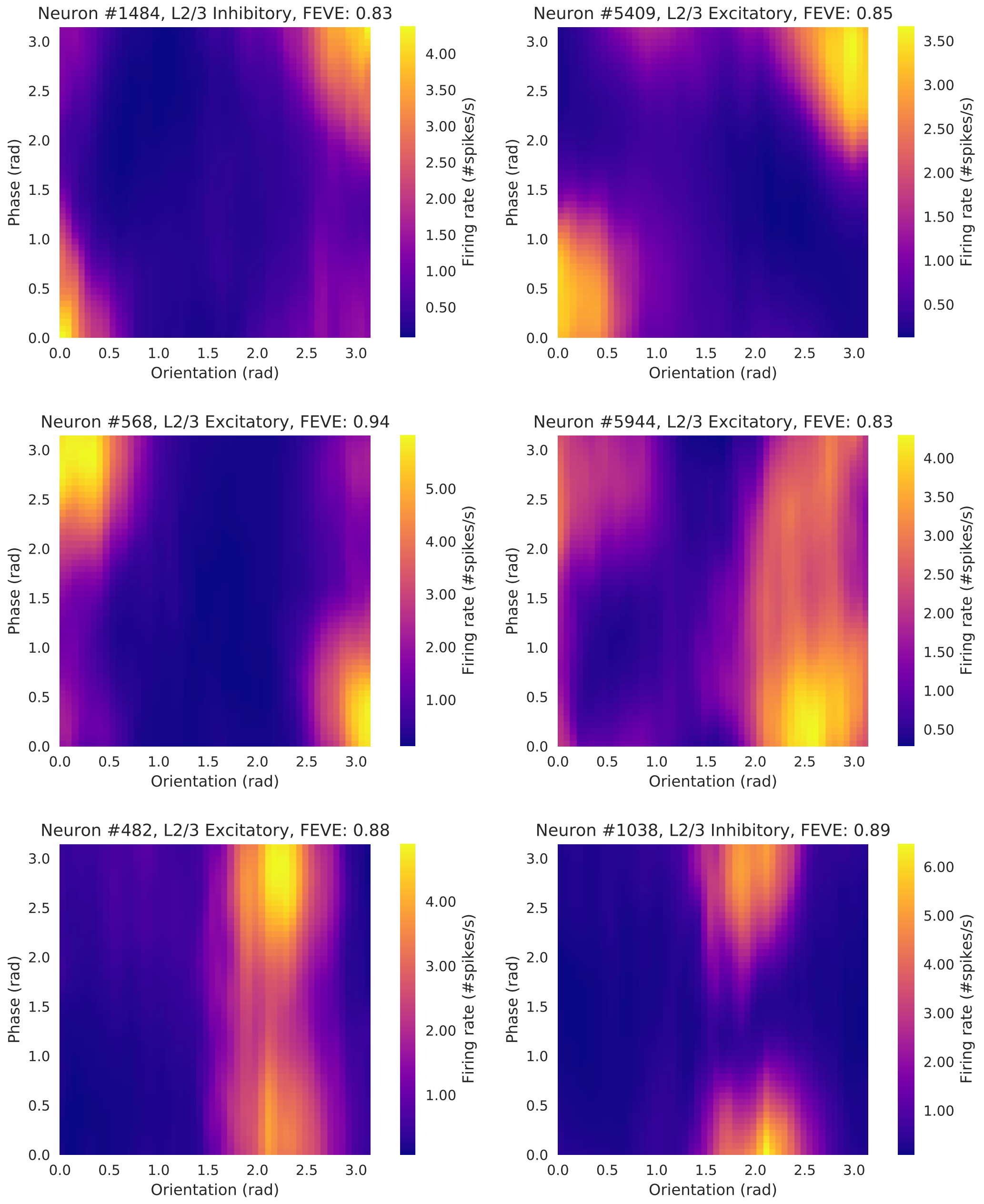


Figure A.1 Phase and orientation tunings of layer IV neurons. Each plot illustrates the phase and orientation selectivity of a single neuron. The titles specify the types of neurons and the goodness of prediction in terms of FEVE.



Neuron #1484, L2/3 Inhibitory, FEVE: 0.83

Neuron #5409, L2/3 Excitatory, FEVE: 0.85

Neuron #568, L2/3 Excitatory, FEVE: 0.94

Neuron #5944, L2/3 Excitatory, FEVE: 0.83

Neuron #482, L2/3 Excitatory, FEVE: 0.88

Neuron #1038, L2/3 Inhibitory, FEVE: 0.89

Figure A.2 Phase and orientation tunings of layer II/III neurons. Each plot illustrates the phase and orientation selectivity of a single neuron. The titles specify the types of neurons and the goodness of prediction in terms of FEVE.

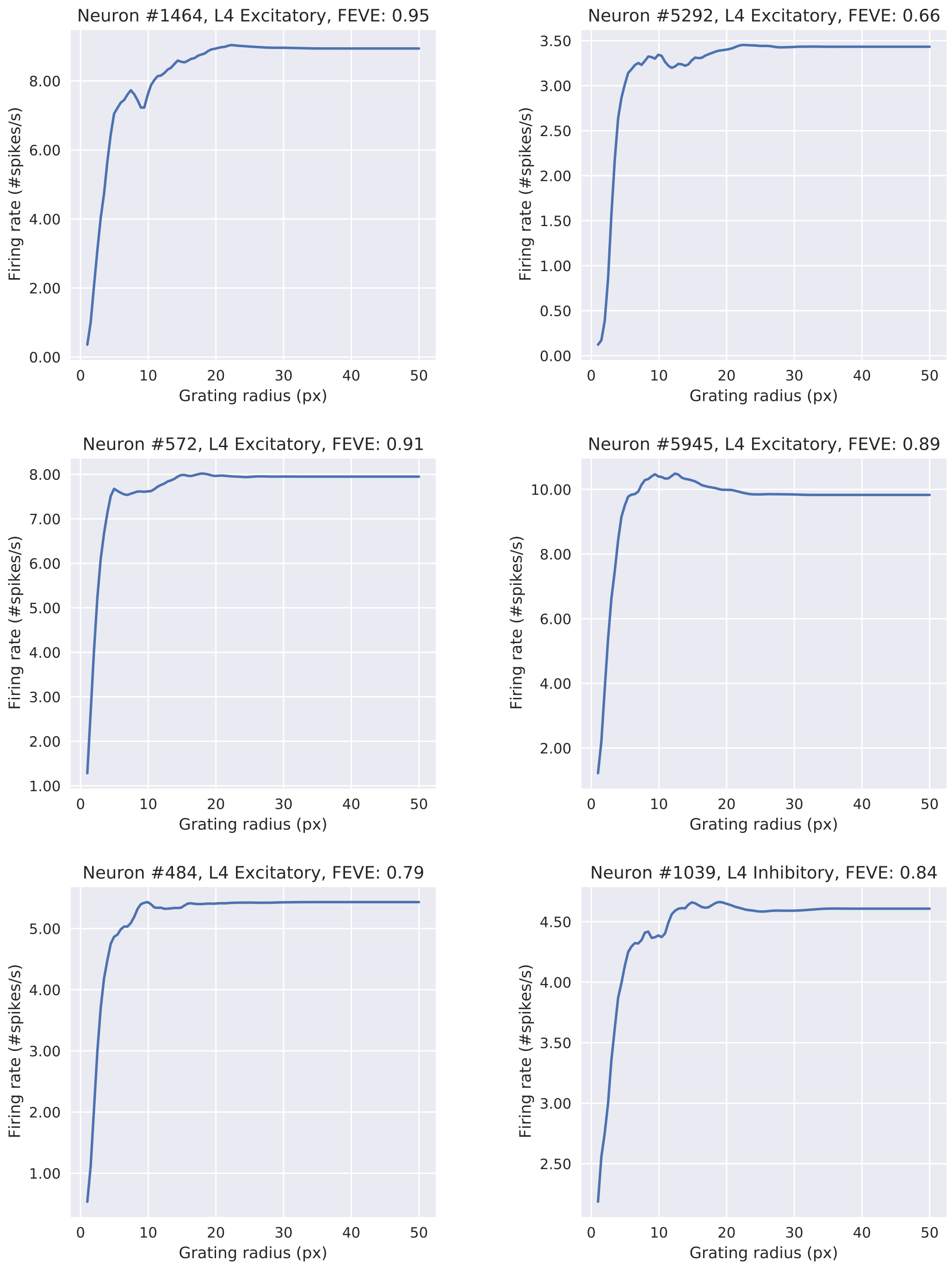


Figure A.3 Size tunings of layer IV neurons. Each plot illustrates the relationship between the stimulus size and the response of a single neuron. The titles specify the types of neurons and the goodness of prediction in terms of FEVE.

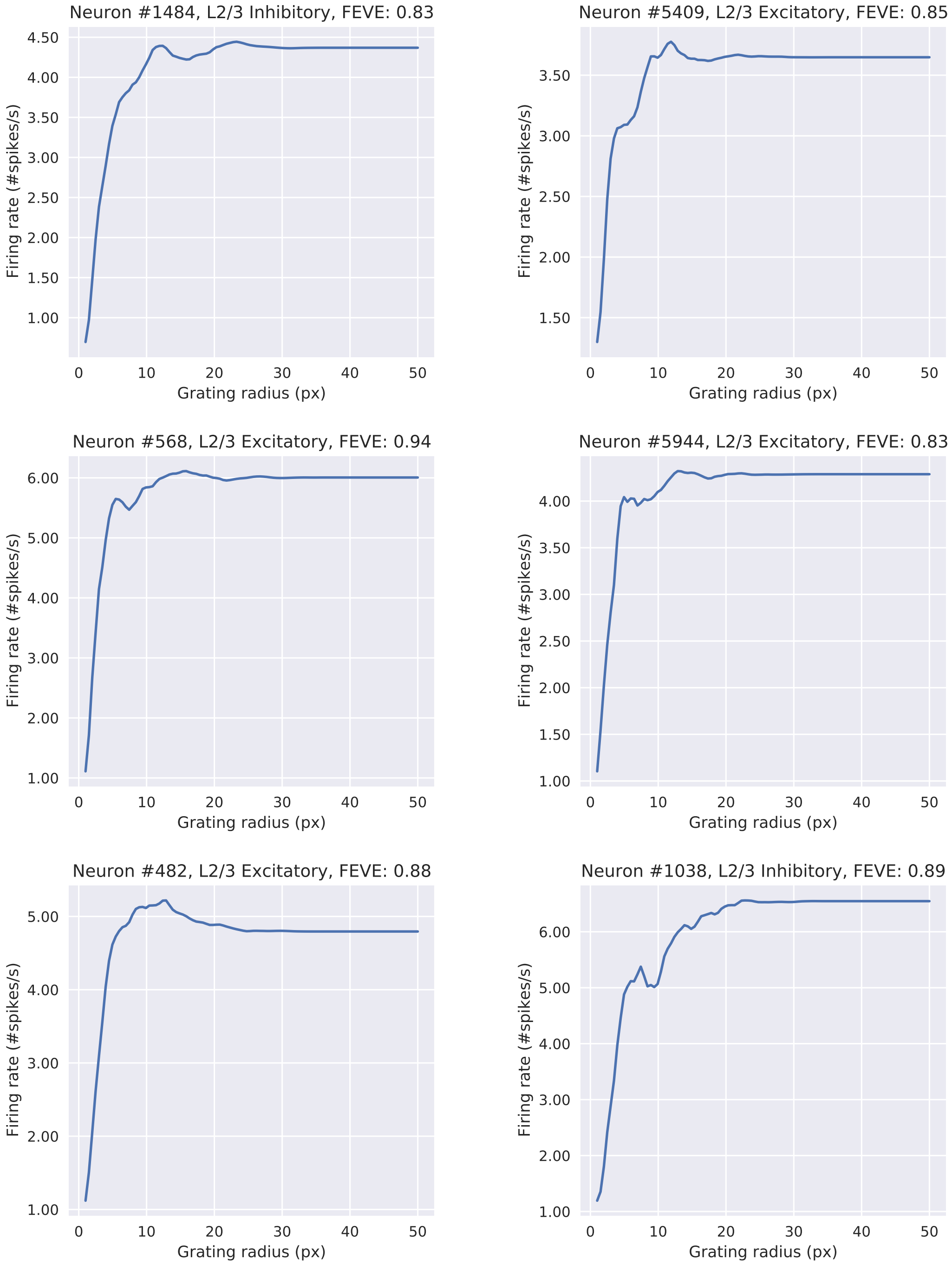


Figure A.4 Size tunings of layer II/III neurons. Each plot illustrates the relationship between the stimulus size and the response of a single neuron. The titles specify the types of neurons and the goodness of prediction in terms of FEVE.

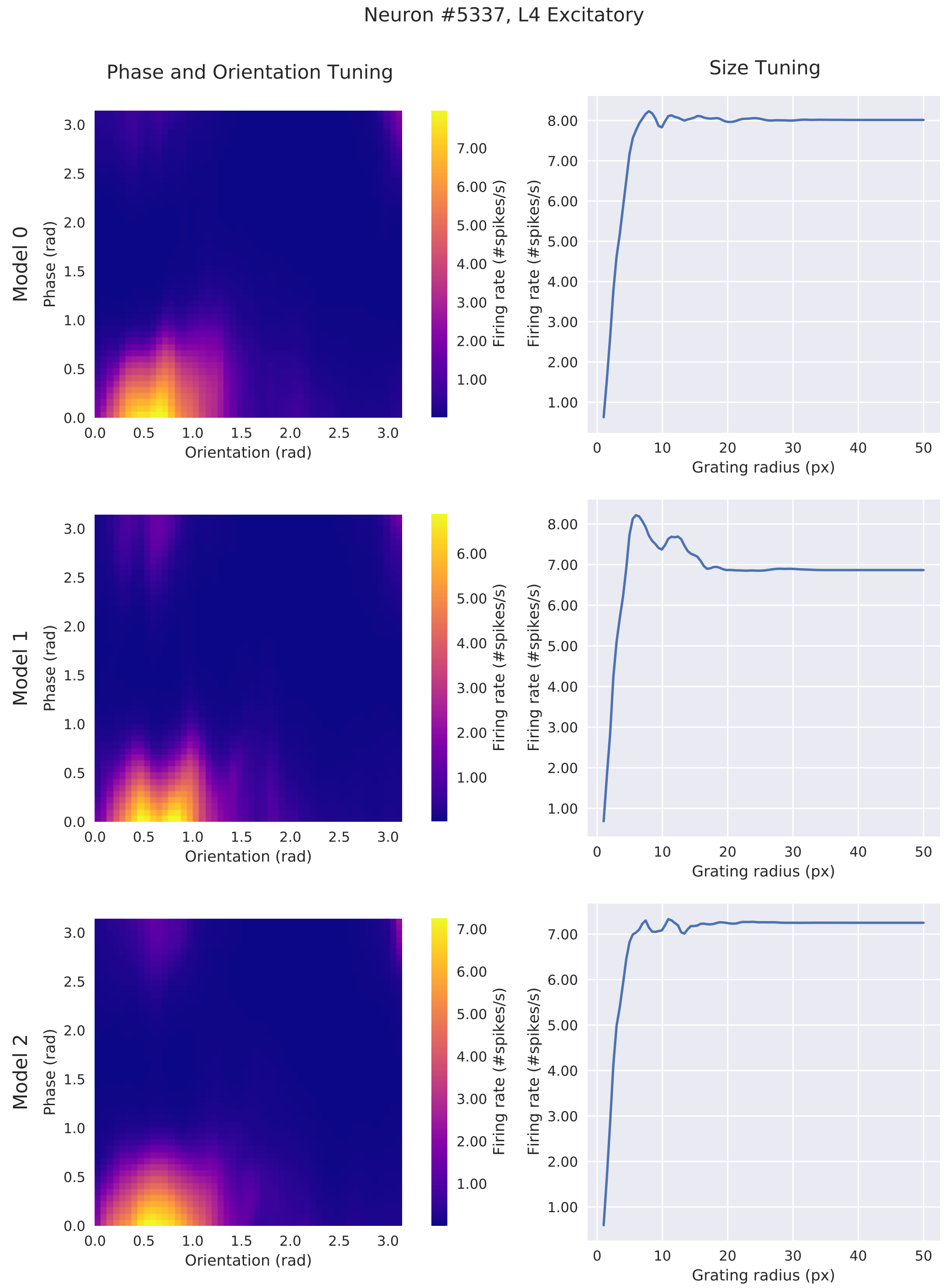


Figure A.5 Phase, orientation and size tunings on models with varying initializations. Each row represents a model trained with a different initialization. The left column show the phase-orientation selectivity plots while the right column illustrates the size tunings.

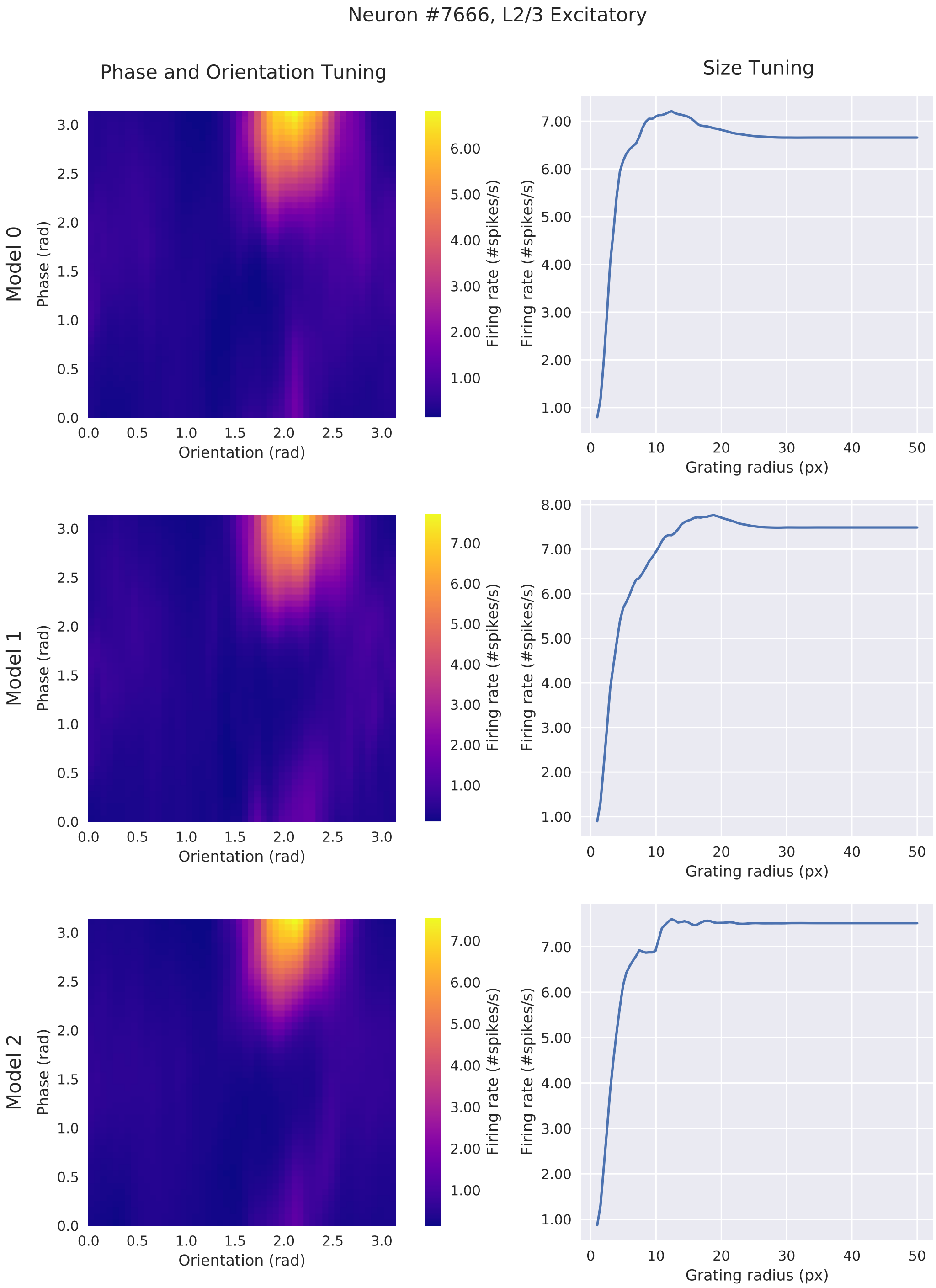


Figure A.6 Phase, orientation and size tunings on models with varying initializations. Each row represents a model trained with a different initialization. The left column show the phase-orientation selectivity plots while the right column illustrates the size tunings.