

Project Report:
*Investigating topographic neural map development of the
visual system*
ELEN E6002

Daniel CLARK
djc2184

May 14, 2013

Overview

The human visual system is the most efficient and complex information processor known to us today. It enables us to accurately perceive a vast amount of information in little time, with (seemingly) little effort. The cortical and subcortical circuitry that composes the visual system has been well studied and modeled for years to shed some light on the brilliance of mother nature's design. Its architecture provides for a reliable presentation of visual stimuli from retinal photoreceptors to neurons in the visual cortex. Specifically, the activities and organization of neurons in the primary visual cortex, commonly referred to as "V1", are essential in conveying this stimulus information to the higher centers of the brain. This motivates an investigation into the development process of V1 neural architecture and its overall performance in information encoding.

The goal of this paper is to explain and bring some intuition into a particular software implementation of neural map development called *Topographica*. In order to do so, we will first give an overview of the construction and function of the early visual system. The concept of spatiotemporal receptive fields and the neural responses of simple and complex cells will also be introduced. The theory of self-organizing neural maps will be covered in detail, where we will explore basic models such as LISSOM and GCAL, inter-neuron connectivity - including afferent, feedback and lateral connections, and contrast gain control and its impact in encoding. All of these concepts will be discussed in the context of *Topographica*'s core model, of which, we will give an analysis and justification, as well as thoughts into various applications and further research.

Contents

Background	3
Early visual system	3
Receptive fields and cell responses	6
Gain control	8
Self-organizing maps	9
Plasticity and connectivity	9
LISSOM	10
GCAL	14
Topographica	17
Functionality	17
Implementation	18
Applications and further research	20
Acknowledgements	23
References	24

Background

In order to understand the structure and intuition behind the models used in topographic neural map development, we will introduce a general background to the architecture of the visual system, as well as its associated fundamental concepts.

Early visual system

The visual system is a part of the central nervous system, which provides for the perception of light in the surrounding environment to the brain. It is comprised of a series of nerve-driven components that relay visual information to the visual cortex, including:

- retinal photoreceptors
- bipolar cells
- retinal ganglion cells (RGC)
- optic nerve
- lateral geniculate nucleus (LGN)
- optic radiation
- primary visual cortex (V1)

The optic nerve and optic radiation stages are essentially relay mechanisms composed of the axons of the RGC and LGN, respectively. The neural architecture within each of these stages is complex, consisting of various feedforward, feedback, lateral, and parallel connections of many different types of neurons; approximately 55 in the retina alone [1], however, for simplicity, we will focus on a few main classes of cells. The photoreceptor cells, rods and cones, are responsible for the transduction of photons of light into electro-chemical responses due to the presence of opsin proteins. As the cone photoreceptors are most functional with bright light conditions, they are responsible for color encoding; this arises as a result of the sensitivity of the opsins to different wavelengths of light. However, the wavelength-sensitive opsins are exclusively segregated in cone cells, such that a particular type of cone cell is sensitive to shorter wavelength light, while the others are for longer, yielding ‘blue’ and ‘red’ or ‘red/green’ cones [1]. On the other hand, rod photoreceptors provide for visual function in dim lighting and periphery vision.

Cone photoreceptors provide for one stage of stimulus encoding, however the bipolar cells which interact with them hyperpolarize or depolarize depending on their cell type. These two distinct cell types are commonly referred to as ON and OFF cells, of which each are populated about evenly. The ON cell types respond to high levels of light, while OFF cell types respond to lower levels. Furthermore, these ON and OFF cell types are composed of exposure length-sensitive cells, where certain types of bipolar cells respond to rapid light changes (transient, high frequency) and others to slower light changes (sustained, low frequency) [2]. With this variety, any cone can be “tapped” by several types of bipolar cells so that all relevant visual information (color, brightness, temporal behavior) can be passed on to the ganglion cells for further processing.

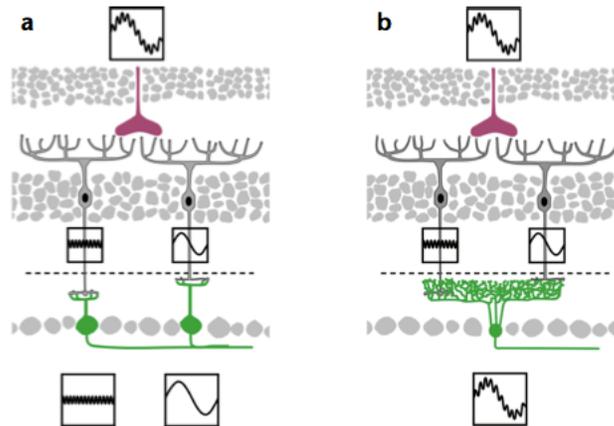


Figure 1: Cones interfacing with different ON bipolar cells and subsequent RGC's [1]

Rod photoreceptors encode information in a very different way than cones. First of all, there is only one known kind of rod photoreceptor and associated rod bipolar cell. However, a given rod bipolar “taps” many more rod photoreceptors than cone bipolars do with cone photoreceptors. The result is a high sensitivity information encoder, enabling visual perception in low-light and peripheral areas. The rod bipolars then drive a special amacrine cell which synapses onto the axons of the cone bipolars. This reveals that the rods don’t actually synapse with the ganglion layer directly, rather they “go around” the cone network to provide information to the ganglion cells.

Within the retinal system, two other general cell types provide for tuning and feedback information for the photoreceptors, bipolar cells, and RGC’s: horizontal and amacrine cells. These cell types have a more “lateral” impact on the neural architecture as a whole, but are not necessarily physiologically connected this way. Generally, the horizontal cells can be thought as feedback mechanisms for the photoreceptors, as they are the source of lateral inhibition of neighboring photoreceptor response. They are also often credited with formation of early stage receptive fields (described in detail below) as well as contributing to adaptability of neural response to overall environmental illumination [3]. The cones and rods interact with the horizontal cells in different ways such that visual statistics, like the different ranges of brightness, can be processed independently, however the overall functionality of the horizontal cells remains the same.

The world of amacrine cells involves a much broader range of roles in stimulus encoding, however they are generally credited with the overall tuning and control of RGC response. As most amacrine cells are predominately structured around their extensive dendritic tree, the number of synapses between amacrine and ganglion cells outnumber those with bipolar cells. This gives a significant controlling function of RGC response to the amacrine cells. In fact, experiments have been conducted to show that the size of the dendritic tree of an amacrine cell has a direct correlation with the receptive field response issued by the RGC [4]. Excitatory, inhibitory, and correlated firing of neighboring RGC’s is credited to amacrine connectivity, which shapes the receptive field response. Finally, the variation in amacrine sensitivity (between narrow, medium and wide-field responses) is often associated with contrast gain control and sensitivity to various visual statistics (e.g. direction, edges).

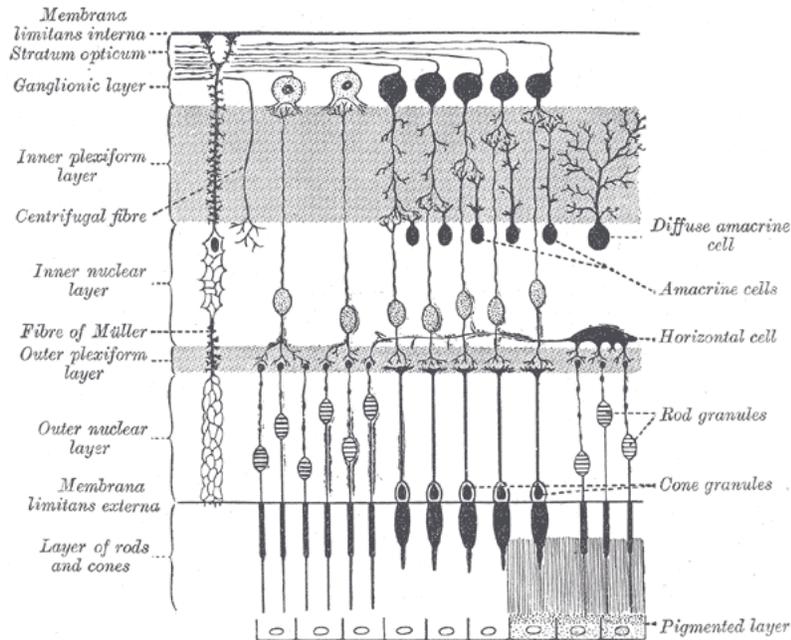


Figure 2: Detailed drawing of retinal architecture, including horizontal and amacrine cells

The LGN serves as a relay center for the input from RGC to V1 and occurs in both the left and right hemispheres of the mammalian brain. In addition to receiving retinal input from their respective (ipsilateral) eyes, the left LGN receives retinal input from the right eye, and the right LGN from the left eye as well, as the optic nerves partially cross mid-brain in the optic chiasm before arriving at the respective LGN's. The LGN is divided into 6 layers, each layer is composed of a sheet of neurons. Interestingly, each layer receives information from only one eye (layers 2,3 and 5 from the ipsilateral eye and layers 1,4 and 6 from the contralateral eye), though the reason for this is not entirely clear [5]. Each neuron in a given sheet is mapped to the same neuron from its respective retina, such that a columnar structure of neurons perpendicular to the layers contains information from the same point in the visual field (for alternating left and right eyes). Along with input from the retina, the LGN also receives feedback from the visual cortex, the thalamus, and the brainstem. Focusing on vision, the LGN is generally credited with temporally integrating and weighting the input stimulus coming from the RGC [6]. In this way, it filters out any temporally redundant information (which can be common in natural images) for efficient encoding. As a result, it is also credited with detection and encoding of temporally-related information such as movement of objects in the environment.

The primary visual cortex, or "V1", is similar to the LGN in the fact that it is arranged in 6 layers of neurons that respond to correlated portions of the retina. The columnar structure, as it pertains to a specific portion of the retinal image, of the LGN is seen in V1 as well with a more sophisticated mapping, where feature preferences begin to emerge (e.g. orientation, direction, spatial frequency). Connections in between neurons in different layers of V1 are composed of afferent, lateral, and feedback types. These connections feed forward to higher cortical processing areas V2, V4, and V5/MT and feed back to LGN. Visual input from LGN is typically passed into layer 4 (particularly layer 4C, as layer 4 of V1 is itself subdivided into 4 layers) of V1 [7]. Within a given layer, lateral excitatory and inhibitory connections between neurons also persist and impact the response of neurons in other layers via feedback and afferent connections. It is argued that these connections give way to simple and complex cell formation in the visual cortex [8], which will be discussed in more detail in the next section. It should be noted that each hemisphere of the brain has a primary visual cortex receiving input from its respective LGN.

It is also important to keep in mind that the fundamental organization of neurons and neural connections in all main stages of the visual system, that is, the retina, LGN, and V1, are *topographically* organized. In other words, adjacent areas of the visual field are represented in adjacent neurons in all of the various stages and layers of the subcortical and cortical components.

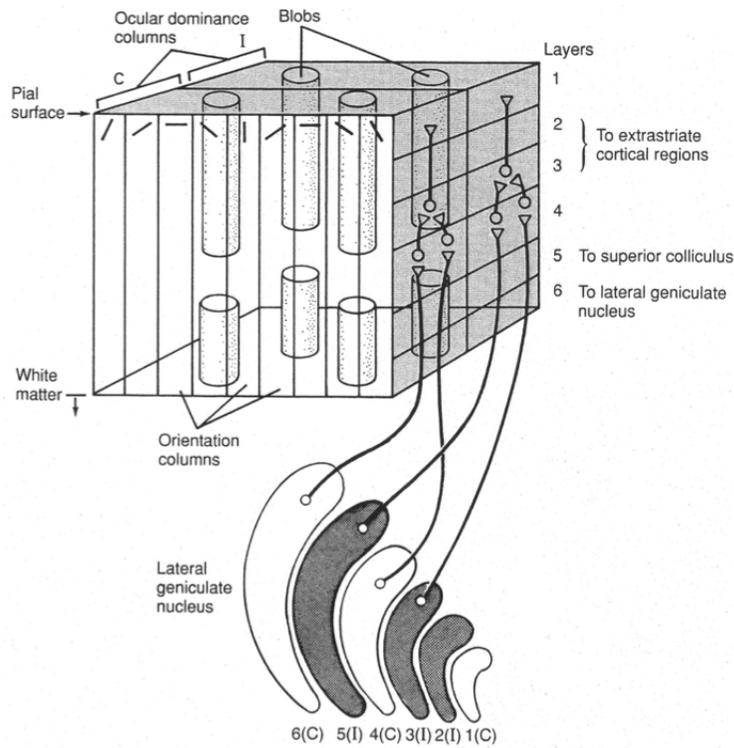


Figure 3: LGN to V1 L4C connectivity and columnar structure [7]

Receptive fields and cell responses

The receptive field, in the context of the visual system, is a region in the visual field which impacts the firing of a neuron. Since neurons are topographically arranged to encode the entire span of the visual field, the “neurological window” to the outside world can be thought of as a grid of receptive fields. The shape of receptive fields vary as one traverses the different stages of the visual system. In the subcortical stages, that is RGC and LGN, the receptive fields are in the form of isotropic “center-surround” patterns. The center-surround receptive fields tend to cause the most response in RGC and LGN when there is light in the center with absence of light surrounding the stimulus, and vice-versa. The ON-type neurons generally respond to the former, where OFF-types to the latter. On the contrary, when light is in the center and the surround of a particular stimulus, neuron firing is very weak. The same case occurs in the absence of light. This demonstrates a form of contrast encoding, where visual stimuli are encoded based on the contrast with their environment. The general function which has been used to characterize the behavior of the spatial-receptive field in the subcortical layers is the difference of Gaussians configuration (DoG).

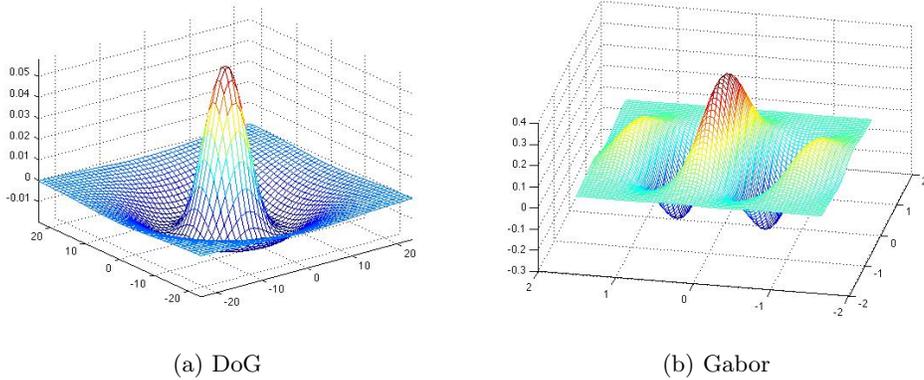


Figure 4: Subcortical and cortical RF functions

In the cortical stages, the receptive fields take on more elaborate shapes. The form of the receptive field is no longer isotropic, but stretched along a particular angle in order to detect edges and objects with a particular orientation. This is typically modeled as a Gabor function. With regard to stationary images, receptive fields are usually only modeled spatially, however, they have a temporal component as well. In general, the spatial patterns of the fields modulate over time, thereby yielding information about relative temporal movement in the visual field to the brain.

Neural response is typically measured as a fixed-point firing rate in relationship to the input stimulus intensity. The functions associated with these responses always yield a positive value (as a neuron cannot have a negative firing frequency). If we momentarily ignore the nonlinearities of any neuron, the linear response estimate of a neuron’s firing frequency can be thought of as a weighting and summation of the spatial components of the stimulus with the neuron’s receptive field convolved over time; the temporal aspects of the input and the receptive field can be thought of as a typical linear convolution of a temporal signal with some kind of filter. Mathematically,

$$L(t) = \int \int g(x, y, s) c(x, y, t - s) dx dy ds, \quad (1)$$

where $c(x, y, t)$ represents the input stimulus and $g(x, y, t)$, the spatiotemporal receptive field, yielding the time-fluctuating linear response $L(t)$. However, $L(t)$ does not match the profile of neural responses very well, due to nonlinearities in neuron behavior. The way this is commonly overcome is by weighting the $L(t)$ with a static nonlinearity, typically of the form given in [6]

$$R(t) = r_0 + F(L(t)), \quad (2)$$

where r_0 is some bias spontaneous firing rate and F represents, at minimum, a rectifying function, but it usually involves a more sophisticated manipulation by processing it through an integrate and fire (IAF) neuron model, Hodgkin-Huxley model, or some variant of a stochastic process (e.g. Poisson). This two-stage process is referred to as the “linear-nonlinear” model, or LN model.

The cortical neurons also demonstrate further complexities in response giving way to two distinct cell types, simple and complex cells. Both simple and complex cells respond best to edges and gratings that are parallel to their receptive fields, however, complex cells produce largely invariant response with regards to spatial phase, whereas simple cells show a strong phase correlation with the input stimulus. In order to model this, a typical approach is to linearly combine and square a number of simple cell responses to form a complex response, as given in [9]

$$R_{cc}(t) = R_{se}^2(t) + R_{so}^2(t), \quad (3)$$

where $R_{cc}(t)$ denotes the temporal response of the complex cell and $R_{se}(t)$ and $R_{so}(t)$ represent two different (opposite in phase) simple cell responses. It should also be noted that many physiological studies show complex cells are typically found in Layers 2/3 of V1, while simple cells are predominately associated with Layer 4 [10].

Gain control

The statistics of natural image content encompass a relatively large deviation in overall light intensities (mean luminance) and local contrasts. The visual system would not be able to encode this broad range of information using a single fixed scale resolution range. An element of adaptability to various contrasts and intensity levels present in the stimulus is hardcoded into the architecture of the visual system through a network of feedback, feedforward and lateral communication (as discussed above). In the context of neural adaptability, it is common to refer to the amplitude of neuronal response (spiking frequency) as “gain”, yielding the automatic adjustment of this quantity for encoding as a “gain control.” Since mean and contrast are statistical quantities which reflect a collection of spatiotemporal “pixels” of the image space, they depend on the sampling characteristics of the perceptual system, namely the receptive field, more specifically, the interaction of the receptive fields of the neurons at every single stage of the encoding process. Additionally, as natural scenes can change in mean and contrast quite drastically (even between visual saccades), the system must adapt quickly (faster than typical 200-600 ms fixation time [11]) to drastic changes in stimuli. Other factors such as neural noise and rapid eye movements during fixation make the understanding of the physiological mechanism behind gain control very difficult to derive. Nonetheless, there have been many experiments and models developed to explain how the visual system accomplishes this.

The driving goal behind the theory of gain control is to reduce the system responses to redundant information, this is commonly known as *efficient coding theory*. The idea is to exploit the statistical dependencies of the input in order to reduce the amount of information encoded; the encoded response will effectively reduce the statistical dependencies in the response. Traditionally, linear coding transforms, such as independent component analysis (ICA) have been utilized for this, as they assume a linear composition of the image and analyze its statistics on this basis (much like the linear receptive field). However, many experimental results on natural images have shown that utilizing linear basis functions for encoding does not significantly reduce the multi-information of the input [12]. As a result, nonlinear transforms have been widely adopted to model the gain control found in neurons. This makes sense for natural images because light reflection off multiple surfaces combines nonlinearly, as defined by occlusion [13]. It also makes sense physiologically, as neural response is highly nonlinear to given stimuli, with elements like rectification and phase invariance (as discussed in the cell response types above).

One particular model used to accomplish gain control is based on the divisive normalization (DN) model. Strictly speaking, DN is a scaling of the input space by a nonlinear function of a linear combination of the same space. This can be represented in vector notation, mathematically, as

$$\mathbf{y} = \phi(\mathbf{x}) = \frac{\mathbf{x}}{f(\alpha, \|\mathbf{x}\|)}, \quad (4)$$

where \mathbf{x} represents the input, α represents some function parameters, and $f(\cdot)$ represents some nonlinear kernel function. In the case of neural encoding, specifically in vision, the denominator takes the form of a spatial summation over a portion of the input space. We will see in the models described in the *Self-organizing maps* section that this portion of the input space is taken with respect to a given neuron and its surrounding neighbors; the neuron of consideration adjusts its response by “pooling” the responses of its neighbors. This has physiological grounds at all stages of the visual system, particularly in the retina as bipolar and RGC’s accept input from many neighboring photoreceptors simultaneously to produce an output; this not only helps normalize the response of any given neuron, but it also mitigates neural noise [11]. Additionally, pooling takes advantage of a mutual benefit between rods and cones; in environments of low levels of light, where fidelity of cone photoreceptors is poor, the collection of neighboring light-sensitive rod photoreceptor response yields better perception. A form of pooling was discussed above in the section about complex cell response; we will be coming back to it in the *Implementation* section of *Topographica*.

It is important to keep in mind that DN has a cascading effect up through the cortical levels in terms of gain control, however, the DN mechanism is mainly associated with the subcortical stages of the retina and LGN.

Self-organizing maps

One of the reasons that the brain can accomplish information-processing tasks better than anything we have in silicon is due the large number of neurons, 10^{11} , and synaptic connections, 10^{14} , that are present, while maximum transistor on-chip density is on the order of 10^9 . However, the human genome, which contains all of the hereditary information for how we biologically develop, only has 10^5 genes or less [7]. This poses the question: how do you get 10^{14} from 10^5 ? One predominate notion is that the structure of cortical neurons *self-organizes* based on the input statistics of the environment. This self-organization eventually results in the visual cortex being an optimal “feature detector” of environmental statistics like orientation and direction of motion. We will discuss some of the physiological justifications for this as well as two models for self-organizing maps (SOM): LISSOM and GCAL.

Plasticity and connectivity

It was mentioned above in the *visual system* section that the fundamental organization of all of the neurons in each stage of the visual system is in the form of a topographic map. The organization of this map is such that neighboring neurons respond to neighboring regions of the retina (though this isn't typically modeled as a linear transformation); this is known as *retinotopy*. In a point firing rate model, retinotopy ensures an area of the visual stimulus will cause the corresponding mapped neurons on the retina to respond with a particular firing rate; this process then cascades in some form to the cortex. In order to extract features about the stimulus from this model, variations in neural response have to occur and be correlated with these features in some way. It has been shown that these correlations are not inherent from prenatal development, but are developed, or driven, by visual input [14]. This notion of adaptability of the neural network was discussed in the *gain control* section above, however, now the adaptability is instituted for long-term development, establishing the notion of *plasticity*. Introducing the notion of plasticity to this framework allows for both short and long-term development as a given neuron can adjust its influence over any of its connections; this is often described as *weighting* the connection strengths in Hebbian learning. Connection weighting has physiological grounds in the area of synaptic plasticity, which is a highly active area of research with a huge amount of literature to support its claims. There are many underlying physiological mechanisms for it in different areas of the brain, however, it is mainly attributed to changes in transmembrane receptor concentrations [15] and calcium signaling [9] as a result of cooperative or competitive firing between neurons.

The neural architecture discussed in the *visual system* section describes how neurons project their responses through synaptic connections by feeding the information forward, backward and laterally through the visual system, thus controlling the information sent to higher centers of the brain. These connections grow in number and various levels of strength after birth to reach maturity. The afferent (feed forward) connections are credited with preserving retinotopy through the system as they run perpendicularly to the topographic organization of the neural sheets. On the other hand, lateral connections traverse the neural layers, spanning both long and short ranges, mediating the responses of other neurons in the sheet. These lateral connections are essential in the correlation of firing between afferently connected neurons. [16]. As the system develops, lateral connections tend to form clusters of connections, while others die off (also known as pruning), resulting in a network where similar afferent connection structures are laterally linked on various layers in the cortex [17]. Feedback connections are generally considered between layers in V1 and potentially contribute to the development of complex cells (as will be discussed below).

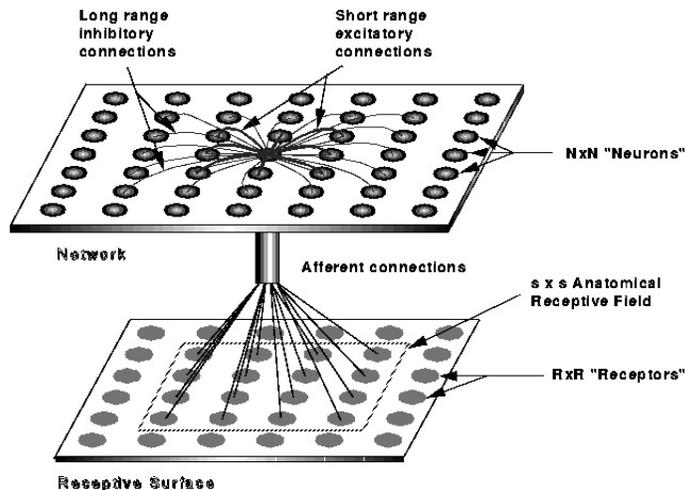


Figure 5: Diagram of afferent and lateral connections [18]

The relationship between these different connection types are reciprocal, in that, the development (or weights) of lateral connections will influence the development (weights) of various afferent connections, and vice-versa, as their development coincides. The motivations for the models below come from recognizing that this underlying self-organizing mechanism can be described as a large-scale process which can be modeled mathematically and simulated in software; these results can then be compared with real animal feature maps for verification.

LISSOM

The LISSOM model, proposed by Sirosh and Mikkulainen in [19], stands for laterally interconnected synergetically self-organizing map. Its basic structure involves a network of neurons, each with afferent (from external input) input connections and lateral (from same neural sheet) input connections. The afferent connections are purely excitatory, while the lateral connections consist of both excitatory and inhibitory contributions. All excitatory connections are short-range while all inhibitory connections span great distances from a given neuron; this yields a network where any neuron's activity will excite its surrounding neighbors while maintaining a functional inhibitory relationship with remote neurons. There is biological plausibility for this metric and has been utilized in many classical self-organizing networks (Belousov-Zhabotinskii reaction) and demonstrated by many experiments, (e.g. [20]). Additionally, the long-range lateral connections are modeled to be significantly weaker than the afferent ones. This also has biological merit in that these connections don't form as many synapses on the dendritic trees of the postsynaptic neurons, and are assumed to have less influence as a result [21].

When a stimulus is exposed to the network, it is represented as a single input vector, and every neuron in the network receives this same input vector through their afferent connections. Each afferent connection has a non-negative associated strength (weight) that scales the influence of that portion of the stimulus; the afferent weights associated with any particular neuron can also be modeled as a vector. From these connections, weights, and the stimulus, the neuron then forms an initial scalar response (firing frequency), η , at time t_0 , through a dot product of the input vector, \mathbf{c} , and the weight vector, ω , such that

$$\eta_j(t_0) = f\left(\sum_{i \in h} \omega_{ij} c_i(t_0)\right), \quad (5)$$

where ω_{ij} represents the connection weight from unit i of the input, which spans h elements, to the j th neuron in the network and $c_i(t_0)$ represents the initial input stimulus at unit i . In actual implementation, the input \mathbf{c} is transformed and normalized so that larger inputs (longer vector) won't result in a greater response than smaller ones. The weighted dot product is a common implementation of a neural network,

as it models the combination of synaptic inputs on a neuron’s dendritic tree. f is a nonlinear activation function which can be defined by

$$f(x) = \begin{cases} 0 & x \leq \delta, \\ (x - \delta)/(\beta - \delta) & \delta < x < \beta \\ 1 & x \geq \beta \end{cases} \quad (6)$$

where δ represents the threshold of activity (the level at which the neuron can begin firing), and β represents the saturation level that defines the maximum firing rate of the neuron.

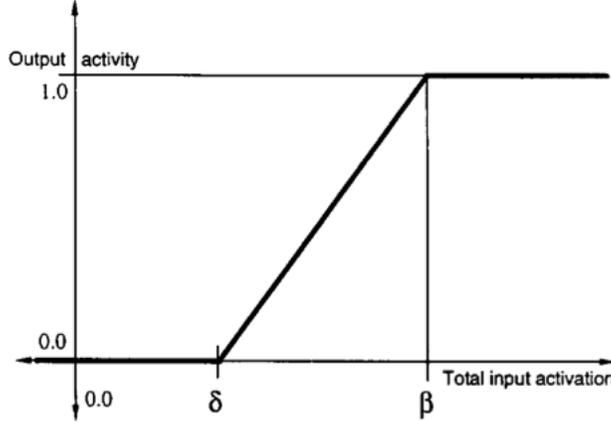


Figure 6: Nonlinear activation function [19]

This activation function is analogous with the non-linear stage of the LN model. The form above is a simple implementation of a piecewise-linear sigmoid function.

With this initial response populated for every neuron j in the network, the lateral connections can then refine neural response on the next time iteration in the following form:

$$\eta_j(t + dt) = f\left(\sum_{i \in h} \omega_{ij} c_i(t + dt) + \gamma_e \sum_{k \in \ell} \xi_{e,kj} \eta_k(t) - \gamma_i \sum_{k \in \ell} \xi_{i,kj} \eta_k(t)\right), \quad (7)$$

where $\xi_{e,jk}$ and $\xi_{i,jk}$ represent the excitatory and inhibitory weights from neuron j to neuron k . It should be noted that in simulation, all weights are initialized from a normalized Gaussian random distribution. This is scaled by the previous activity of those neurons $\eta_k(t)$ and summed over every neuron k in neuron j ’s lateral connection field, which spans ℓ neurons. γ_e and γ_i represent contribution constants which determine the overall strength of the lateral connections.

It should be noted here that the weight vectors are kept constant throughout the activation update process until the network has “settled,” over a number of iterations (each iteration of is of duration dt). The weight vectors are then updated on this secondary time scale, which is indexed by n below. The weight update equations obey the form

$$\omega_{ij}^{n+1} = \frac{\omega_{ij}^n + \alpha \eta_j(t_n) c_i}{\{\sum_{i \in h} [\omega_{ij}^n + \alpha \eta_j(t_n) c_i]^2\}^{1/2}} \quad (8)$$

for afferent weights and

$$\xi_{jk}^{n+1} = \frac{\xi_{jk}^n + \alpha_L \eta_j(t_n) \eta_k(t_n)}{\sum_{k \in \ell} [\xi_{jk}^n + \alpha_L \eta_j(t_n) \eta_k(t_n)]} \quad (9)$$

for lateral weights (both excitatory and inhibitory). t_n represents the time step at which the weights are updated, and the α terms represent the learning rates for each process (α_L can be for excitatory, α_e , or inhibitory, α_i , learning rates). The above processes follow the Hebbian principle that large products of

activities between neurons result in large weight changes. In addition, the above implementation makes a number of assumptions about the underlying biological processes: (1) afferent and lateral connections develop simultaneously, (2) connection strengths are normalized over the entire neuron as opposed to relying on realistic synaptic behavior. The simultaneous development of these connection types has been demonstrated in the cat [22], where, the afferent connections form ocular dominance columns in V1 develop on the same time frame as lateral connections grow, mature, and form clusters for feature preferences. At the end of 6 weeks, both lateral and afferent connections have reached maturity, suggesting that they progressed at the same time. The normalizing of all activity relationships between a postsynaptic neuron and its presynaptic partners seems like an artificial simplification of the underlying biological process actually contributing to its response: namely synaptic behavior. It is argued in [19] that a biologically-plausible model for synaptic adaptation (in regards to neural efficacy) converges to the same form as Eq. (8), given that the synapses for that connection type (afferent or lateral) occur locally on the neuron. This model is represented mathematically below:

$$\frac{d\mu_{js}}{dt} = \left(\psi_s - \lambda \mu_{js} \sum_r \mu_{jr} \psi_r \right) \cdot \sum_l h_{jl} \eta_l, \quad (10)$$

where ψ_s and ψ_r represent the pre and post synaptic activity of synapse s on neuron j , μ represents the weight of that particular synaptic connection, r indexes the postsynaptic cell's inhibitory influences, and the outer term is a summation of all the chemical influences from l neighboring presynaptic cells, based on their activation η_l and an influence term h_{jl} [23]. This model is a more physiologically grounded one and can serve as a justification for Eq. (8) and (in cases of numerous lateral connections) Eq. (9).

The activation function is updated at the same frequency that the stimulus \mathbf{c} changes, which is not necessarily every time step. In the LISSOM model, \mathbf{c} can change continuously or be presented in discrete steps, however, both the weight vectors and the activation function should only be updated after the input changes in either case. The activation function is updated as follows

$$\begin{aligned} \delta_j^{n+1} &= \min(\delta_j^n + \alpha_\delta \eta_j, \delta_{max}) \\ \beta_j^{n+1} &= \max(\beta_j^n + \alpha_\beta \eta_j, \beta_{min}). \end{aligned}$$

This provides for an additional nonlinearity, in that, the neurons will actually mature faster at higher-activity portions of the network map. This is because as the activation function, f , becomes narrower in its responsive zone, the firing rate of any neuron within the responsive region will vary more and more. This type of evolution occurs in observations of neuron response throughout its growth. Immature neurons are easier to trigger and more linear in their response than after they mature [24]. As the cell develops, it becomes harder to trigger spiking, but because of a higher density of ion channels, when it does respond, it exhibits higher spiking rates, coinciding with faster saturation (as modeled above).

Finally, connection death is implemented by pruning weak lateral connections at specified time periods. The parameter t_o determines the initial onset of connection death, where any lateral connection value ξ_{kj} that falls below the threshold w_c is effectively removed from the network. Subsequent intervals, defined by t_c , mark periods where lateral connections are pruned after initial onset. Physiologically, this has been observed in many cases [22], where the majority of long-range lateral connections don't survive cortical development. The assumption in the model indicates that these connections must be weak for a duration of time before they are eliminated.

The model has been implemented and evolved in different ways to yield realistic topographic neural maps. Initial experimentation showed how the randomly initialized afferent weight vectors rotated towards their connected input vectors with each iteration. After enough iterations, the (initially random) weight vectors formed a realistic topographic map, resembling the retinotopy from visual receptors to V1.

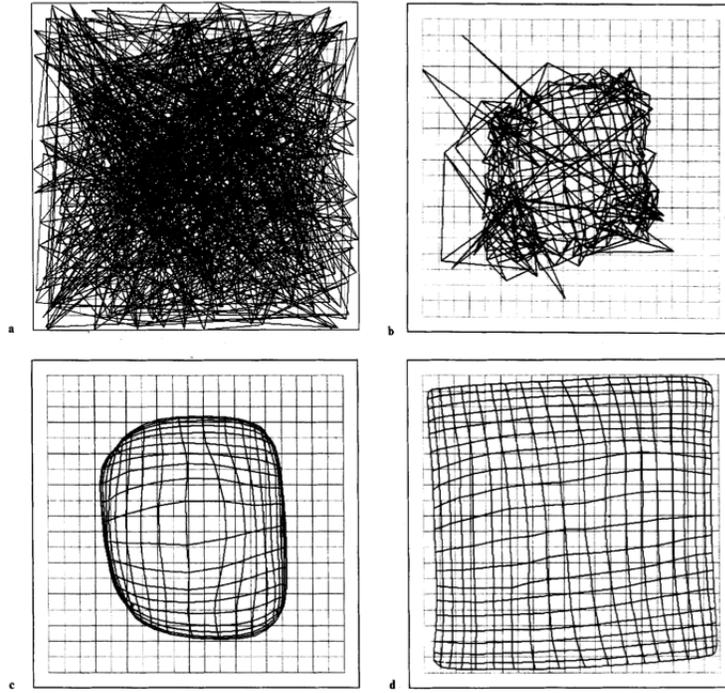


Figure 7: Self-organizing of afferent input weights (retinotopy) [19]

This principle also carries over to lateral connection weight vectors, as they rotate the greatest towards correlated activity between neurons. This is the essential driving force in neural map development in regards to feature extraction since the neurons that are firing with a correlated response rate are presumably seeing similar features in the input; as a result, the lateral connection between them becomes stronger, and they become increasingly correlated to those same features (e.g. ocular dominance, orientation). An altered version of LISSOM called RF-LISSOM (receptive field LISSOM) instantiates the connection fields of any particular neuron to a local area of the input as opposed to the entire input, where the center of the connection fields are topographically arranged in proximation to the neurons in the network. This makes sense in simulation because the amount of calculations becomes significantly lower; it does make the assumption, however, that the receptive field of any neuron is relatively small compared to the entire input field and is spatially centered around that particular neuron. This assumption is experimented against others (such as having the entire retina act as the inputs for each neuron) in [25]. Using the local receptive field centers and two photoreceptor inputs (modeling left and right eye), the authors achieved a realistic mapping of ocular dominance columns in the neural sheet.

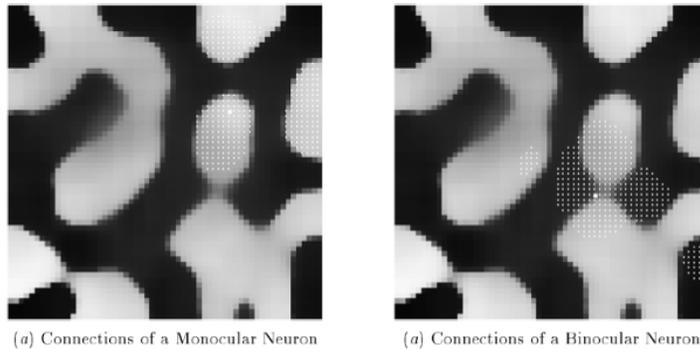


Figure 8: Self-organized ocular dominance map using RF-LISSOM [25]

The LISSOM model provides a great framework for map development, however there are some limitations in its biological plausibility. The determination of the activation threshold δ in LISSOM is a supervised process, as a maximum and minimum function performs a logical compare between two values. This is not necessarily a completely input-driven approach; it also artificially drives the lateral connection radii to become smaller, faster than they would normally be. Additionally, there's no concept of gain control implemented in the system; the amount that the input, \mathbf{c} , varies in contrast is interpreted on a fixed scale.

GCAL

The GCAL (gain control, adaptation, laterally connected) model uses the LISSOM model as a base framework with two key improvements: (1) the incorporation of feed-forward gain control and (2) single neuron adaptation, also referred to as homeostatic plasticity. As GCAL was developed with software implementation in mind (namely *Topographica*), its fundamental properties are much more specific than in the LISSOM model. In fact, the GCAL model should not be thought of as a model on the same theoretical level as LISSOM, it is more of an implementation-specific instantiation of self-organizing maps and thus, give more specific parameters for real experiments. Specifically, its implementation generally infers the presence of at least four stages:

- Photoreceptor stage
- RGC/LGN On stage
- RGC/LGN Off stage
- V1

The model can be described, starting with the photoreceptors to RGC/LGN. This connection field is now defined to be retinotopically aligned with the neurons of RGC/LGN and has a predefined radius upon initialization. In the base model, there is no plasticity in the afferent weights from the photoreceptors to RGC/LGN. The weights are instead fixed, defined by a DoG function in the following form

$$\omega_{ij} = \alpha \exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\sigma_c^2}\right) + \beta \exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\sigma_s^2}\right), \quad (11)$$

where x_i and y_i define the location of photoreceptor i and x_j and y_j define the location of postsynaptic neuron j in RGC/LGN, α and β are constants that can be scaled, positive or negative, to generate the correct DoG, σ_c is the width of the center Gaussian and σ_s is the width of the surrounding Gaussian. The activity in RGC/LGN is updated at every time step dt , as in Eq. (5) but with a few differences, such that

$$\eta_j(t + dt) = f\left(\gamma_L \sum_{i \in F_j} \omega_{ij} c_i(t)\right), \quad (12)$$

where f is a nonlinear rectifying function, γ_L is an arbitrary multiplier for the connection strength, and F_j is the span of the connection field of neuron j . f , in this case, is a pure half-wave rectifier without specific thresholds predefined (we'll get to that soon). The feedforward gain control is implemented in the RGC/LGN layer as a divisive normalization (as discussed in the *Gain control* section) from lateral connections of neighboring neurons. The weights of these connections are set to fixed strengths in the form of a Gaussian, such that

$$\omega_{ij,S} = \exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\sigma_S^2}\right), \quad (13)$$

where x_j and y_j signify the location of the postsynaptic neuron, x_i and y_i signify the presynaptic one and σ_S specifies the width of the Gaussian. The divisive normalization is implemented as follows

$$\eta_j^{n+1}(t_n) = f\left(\frac{\eta_j^n(t_n)}{\gamma_S \sum_{i \in L_j} \omega_{ij,S} \eta_i^n(t_n)}\right), \quad (14)$$

where n represents the separate time scale which specifies when the activity is normalized, t_n is the time at which this normalization happens, γ_S is a constant strength scaling factor, and L_j is the span of neuron j 's lateral connection field. The activity normalization is updated after multiple time steps, so that the activity in all of the layers can settle from a given input sequence. This is implemented as such because GCAL is designed for use of discontinuous inputs, which represent training in "snapshots" over a long duration. For this reason, it only makes sense to update the normalizing term when a new input is presented, which is on the same time-scale as n , not dt . The feed-forward normalization contributes to stabilizing the presynaptic activity of the neurons in V1 so that the values don't swing wildly between input presentations. Additionally, this allows the network to self-organize on a much wider range of contrast levels. Without gain control, the response in V1 is completely linear with regard to input contrast, as a result, the network doesn't form properly at areas of low and high contrast levels [26]. This mechanism provides an overall robustness to the model.

The connections to a given neuron in V1 is the same as defined by LISSOM, with afferent, and lateral (short-range excitatory and long-range inhibitory) connections. This can be described succinctly, similar to Eq. (7) from LISSOM as

$$\eta_j(t + dt) = f\left(\sum_p \gamma_p \left(\underbrace{\sum_{i \in F_{jp}} \omega_{ij,p} \eta_i(t)}_{X_{jp}(t+dt)}\right)\right), \quad (15)$$

where γ_p is a constant scaling strength factor (positive for excitatory, negative for inhibitory) for projection p , $\omega_{ij,p}$ is the connection weight between neuron i and j of that projection p , $\eta_i(t)$ is the activity of the neighboring neuron i at the previous time step, and $X_{jp}(t + dt)$ is the total activity contribution for projection p (p can be afferent, lateral inhibition, or excitation). This activity update is known as the settling process for V1 in the GCAL model and occurs throughout the entire duration of an image presentation. At the end of the image presentation (before a new one is presented to the photoreceptor stage), the V1 activity values are "settled" and represents the trained network as a result of a particular input presentation (a typical amount of settling steps for a given presentation is ≈ 16). Right before the next image is presented, the V1 activity is reset to 0, allowing the activity in the sheet to better represent the statistics of the new input (without decaying the response from the old one).

As in LISSOM, GCAL also implements threshold-sensitive firing via the activation function f . Despite the addition of gain control in the model, the activity map can still develop discontinuities due to postsynaptic variations. For example, as the receptive fields in V1 evolve to better represent the input presentations, the activity in V1 will increase, in turn, increasing the strength of lateral connections. This will only amplify changes further for neurons where activity is strongest. But neighboring neurons that have been activated less strongly will lag behind the others, resulting in discontinuities in the feature map [26]. Homeostatic adaptation remedies this by mediating the postsynaptic response. This is implemented based on a smoothed exponential averaging of any single neuron's previous activity, via

$$\bar{\eta}_j^{n+1} = (1 - \beta)\eta_j(t_n) + \beta\bar{\eta}_j^n, \quad (16)$$

where β is a smoothing parameter ($\beta \approx 0.999$) and $\bar{\eta}_j^{n+1}(t_n)$ is the average activity of neuron j used to determine the threshold of activity for the next, $n + 1^{th}$, image presentation. This average activity value is utilized to update the threshold of activity, such that

$$\delta_j^{n+1} = \delta_j^n + \lambda(\bar{\eta}_j^{n+1} - \mu), \quad (17)$$

where δ_j^{n+1} is the updated activation threshold, $\lambda \approx 0.0001$ is the homeostatic learning rate, and μ is desired target activity of any neuron after a long exposure to the same stimuli; this can be thought of as the spontaneous firing rate of the neuron (or r_0 from Eq. (2)). It can be seen in the equation that as average activity drifts away from μ , the threshold is raised or lowered to bring it back closer to μ . However, because the learning rate λ is usually modeled to be so small, this convergence towards spontaneous activity will only happen if the stimulus remains constant for a long duration. This makes intuitive sense as the neurons will eventually only spontaneously fire when not presented with new information to encode. This homeostatic plasticity in the activation threshold introduces stability to postsynaptic neural

response. This is needed in addition to feed-forward gain control as the gain control only influences the presynaptic activities of the neurons; the post synaptic activity still requires an input-driven stabilizing mechanism as shown above.

The connection weights in V1 for afferent, lateral excitatory and inhibitory connections are initialized via a similar equation to Eq. (13)

$$\omega_{ij,p} = \begin{cases} u \exp\left(-\frac{(x_i-x_j)^2+(y_i-y_j)^2}{2\sigma_p^2}\right), & (x_i - x_j) \leq r_p, \\ 0, & otherwise \end{cases} \quad (18)$$

where $u = 1$ for the lateral excitation weights, and u is a random scalar drawn from a uniform distribution for lateral inhibition and afferent weights, p specifies the projection type (afferent, lateral exc., lateral inh.), and σ_p determines the width of the Gaussian function for projection p . r_p determines the radius for which the weights are defined over neighboring neurons for neuron j . Finally, the weights for each projection type p are updated in the same way (for all connections in V1) as in Eq. (9):

$$\omega_{ij,p}^{n+1} = \frac{\omega_{ij,p}^n + \alpha \eta_j(t_n) \eta_i(t_n)}{\sum_k (\omega_{kj,p}^n + \alpha \eta_j(t_n) \eta_k(t_n))}, \quad (19)$$

where α is the Hebbian learning rate, η_i and η_k represent the presynaptic neuron activities (where k indexes all neurons in the connection field for neuron j), and η_j is the postsynaptic activity. It can be seen that divisive normalization is implemented here in order to prevent a constant increase in the weighting values. This type of weight adjustment isn't necessarily biologically motivated (or plausible). A more realistic approach would be adjusting the weights by modeling chemical activity of synapses throughout the training process and scaling the connection strengths by the amount of synaptic activity between neurons, however this is a computationally demanding approach.

The robustness and stability of GCAL was tested experimentally using a stability index and level of selectivity of orientation maps for different levels of contrast of an input stream of two elongated Gaussian functions at random orientations and locations in the visual field [26]. A similarity index (judging stability) for the map was developed by comparing the orientation preference of individual units throughout development with the final orientation preference after 20,000 training iterations. The selectivity of the map was measured as an average of the orientation selectivity of the afferent connections from RGC/LGN to V1. This average is computed by finding the orientation preference magnitude of each neuron, where the maximum activation of a given neuron in V1 is found by measuring neural response over sine grating inputs (combinations of all orientations, frequencies and phases) - the angle of associated with this maximum activation is the neuron's orientation preference. This activation is then computed and averaged over all of the neurons in the map to yield a quantifiable orientation selectivity. The experiments in [26] compared a network without gain control and homeostatic adaptation (LISSOM) with GCAL over a range of contrasts of the input. The selectivity and stability was found strongest across all contrasts in the GCAL model.

The biological plausibilities of the mechanisms implemented in GCAL are analogous to those described in the *LISSOM* section. However, the addition of feedforward gain control in the RGC/LGN layer is consistent with the behavior of amacrine and horizontal cells in the retina described in the *Early visual system* section along with its divisive normalization method, consistent with the neurological pooling mentioned in the *Gain control* section. The homeostatic plasticity of activation thresholding is a more plausible biological realization than in LISSOM since the threshold is computed for each neuron, purely input-driven (based on average activity), and revolving around a targeted spontaneous rate. The weights of the lateral connections in RGC/LGN are almost certainly not fixed in vivo; this mechanism is ignored in the GCAL model. However, it is important to keep in mind that these models are intended to describe realistic neural map development in V1 and are validated and judged by empirical comparison with real data. Additionally, other mechanisms, such as constant weight adjustment and average computing at every time-step, are left out for computational efficiency. In the end, if the result bears resemblance to real data, the model implemented is considered valid.

Topographica

Topographica is an open source software package that is designed for topographic neural map development. It can be used to simulate the models discussed above as well as any other network that relies on the formation of neural maps (e.g. auditory, motor, touch areas) on a large scale of thousands of neurons and millions of connections. It is built to interface with other programming frameworks, such as *C++*, *Python*, *MATLAB* as well as small-scale neuron simulators like *GENESIS*, *NEST*, and *NEURON*. We will first discuss some of the functionality and features of the package, then some specifics on how the software is structured and implements the models above. Finally, we'll go over an example of how the package and the models above can be used to generate complex cells in Layer 2/3 of V1 and thoughts about further applications and research.

Functionality and features

The package can be interacted with both through the command-line and scripts as well as its GUI to model any type of sensory system that is based on sheets of neurons. The architecture of the model is set before training/testing and is entirely up to the user. Minimally a trainable model will consist of at least a photoreceptor sheet and V1, but more often contains intermediary layers such as RGC/LGN and several cortical layers. The connection fields for any neuron in the non-photoreceptor layers can then be specified and tuned with parameters. Similarly, how these connection fields project to postsynaptic neurons can be specified. With this framework, a network can easily be trained using various input patterns. The package incorporates a diverse library of input functions that can be customized via parameters. These functions include, but are not limited to: Gabor functions, Gaussian noise, various gratings, geometrical patterns, and access to image files. In addition these patterns can be combined to form composite patterns as well as be translated and rotated in different ways for each input presentation to provide a dynamic stimulus environment.

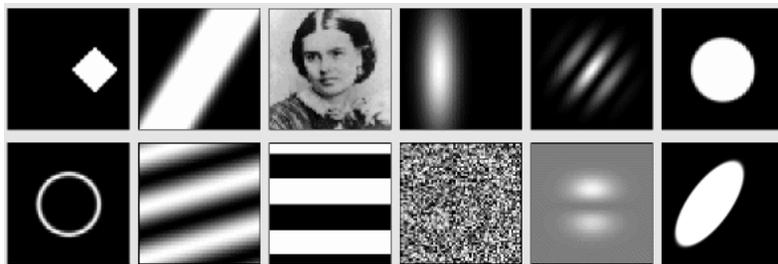


Figure 9: Examples of patterns available in Topographica [27]

The calculation of weight updates, presynaptic normalization, and postsynaptic normalization is completely customizable with a library of functions (discussed in the *Implementation* section). Once the network is defined with a training pattern and connected architecture, it can be ran or stepped through at any desired rate (on either dt or n timescales). As this training period is occurring, one is able to observe various events and statistics of the network happening in real time. This includes the neural response on any sheet, the connection field strength of any projection on any sheet, and a collection of projection between any sheets. In addition, the feature preferences in V1 (e.g. orientation, direction, ocular dominance, tuning curves) can be measured at any point in time; however the calculations for these statistics may take a while to perform.

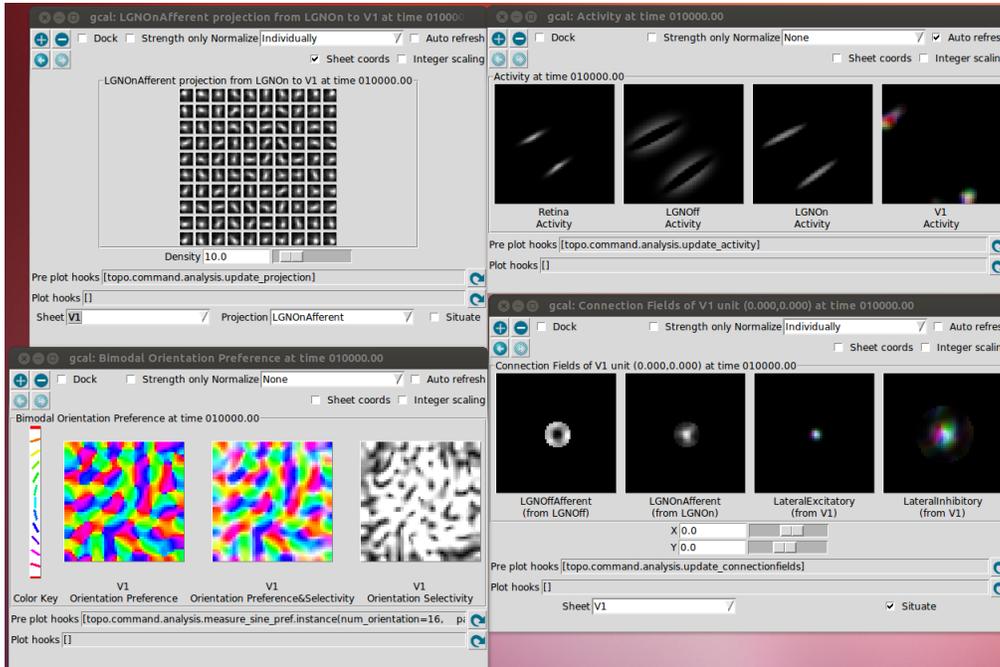


Figure 10: Screen shot of network behavior from Topographica

It is important to note that Topographica’s internal response functions are not the only method for calculating neural response of a given model; it can use external simulators to calculate the response as well, including spiking neuron simulators, such as *NEST* or *NEURON*. This is accomplished by defining a *EventProcessor* class (any object in a simulation which is capable of sending and receiving events) that exports the input presentation generated in Topographica to the external model through a Python wrapper, such as *PyNN*, specifies the run time and the number of neurons to simulate, and then receives the spike times as an input matrix and computes an average firing rate from these spike times. This firing rate is then treated as the input to the next layer stage and the simulation can continue [28]. An example of Topographica utilizing the *NEST* simulator through the *PyNN* wrapper is included in the downloadable package and can be ran provided the wrapper packages and *NEST* simulator are installed locally. Additionally, Topographica can interface to *MATLAB* through the Python package *mlaburap* to execute .m files as well as C/C++ via the *weave* package.

Implementation

Topographica implements the SOM models described above by representing each sheet of neurons as a two-dimensional array of firing-rate units. Each sheet represents two levels of dimensionality: (1) sheet coordinates and (2) matrix coordinates. Sheet coordinates have the origin located at the center and the coordinates increment outward, typically on the order of ± 0.5 to ± 2.0 in the x and y axis. Meanwhile, matrix coordinates are all positive where the (0,0) element starts in the upper left-hand corner, and the (N, N) element in the lower right. The N is determined from a parameter called the sheet density, which describes how many units are in a length of 1.0 sheet coordinates. Typically sheet coordinates of 1.0×1.0 are used for V1 sheets while the sheets become increasingly larger as one traverses the subcortical layers down to the photoreceptors ($\approx 2.75 \times 2.75$). This is to avoid any boundary artifacts for neurons receiving projections from units close to the border as the size of the layer below a given network should be large enough to provide for complete connection fields for every postsynaptic unit. The radius of each postsynaptic connection field can be specified by a parameter or a function. The weight vector for any particular postsynaptic neuron is defined only within the limits of its connection field.

The package provides for a library of different functions which can be utilized for different aspects of network development. In addition to the pattern generator functions, there are various transfer functions,

response functions and learning functions. Transfer functions take a matrix of input activations and modify it somehow, usually for the alteration of an input stimulus or connection field of presynaptic activities (e.g. activation function). Response functions take a matrix of weights and activity values as its arguments and computes a postsynaptic response matrix, typically through a weighted norm; it is also capable of divisive normalization. Learning functions will modify a matrix of weights given presynaptic and postsynaptic activity matrices and is typically utilized to implement Hebbian updates on the connection strengths in the network.

In addition, there are different types of sheets that can be used for each processing stage. GeneratorSheets, for example, are able to produce patterns and images as activations on each unit of the sheet, so they are typically used as photoreceptor sheets in simulation. Projection and CFSheets take matrices of activities from other sheets projections and uses its *activate()* function to compute the sum of the activity values per projection to yield a postsynaptic response. Projection sheets and CFSheets differ only in that CFSheets provide access to the connection field strengths for visualization and plotting. JointNormalizingCF and JointNormalizingCF.Continuous sheets take into consideration all of the projections and corresponding weights to normalize the activity on a broader (sheet-wise) level. LISSOM sheets are also available, which are similar to the JointNormalizingCF sheets except they enforce a strict settling time on the activity between presentations. The JointNormalizingCF.Continuous sheet has no resetting mechanism in between image presentations, allowing continuous behavioral response to any set of inputs. This is typically utilized in network training where one wishes to develop a system on continuous inputs. In general, training a system without resetting the activity in between inputs is not biologically realistic as an image never changes instantaneously in the visual field. However, having a continuous input (like a video of natural scenes) that changes with the time step defined in the model can be used to train a network without the implicit resetting.

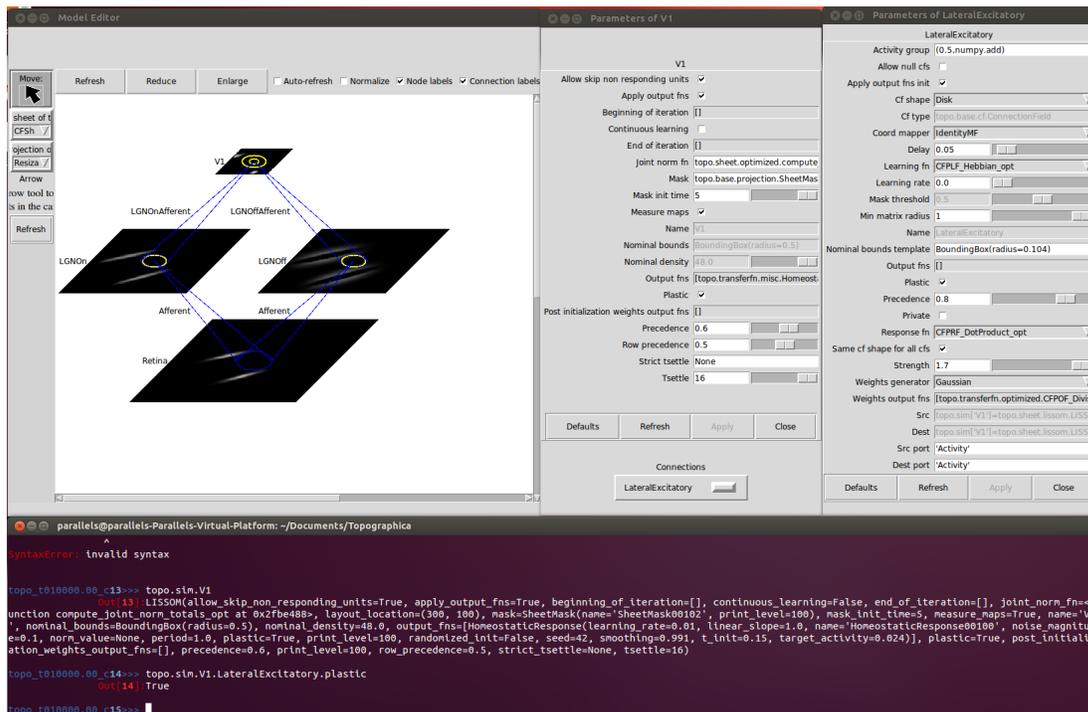


Figure 11: Parameters of the network can be modified in the GUI or through Python

Finally, the coordinate mapping between layers can also be customized to simulate a given layer of neurons residing on something besides a two-dimensional plane. Utilizing these objects, functions, and plotting commands, one can collect statistics and visualize the performance and impact of various components of a network. Any network can be properly tuned to try and extract specific features from the input (e.e. using two photoreceptor sheets to find ocular dominance) which opens up a realm for

in-depth experimentation. It should be kept in mind, however, that the nature of the images presented to the network are crucial to how it develops. For example, in the images below, three different sets of training inputs were utilized: randomly placed Gabors, straight lines, and a human face on a trained version of the *gal.ty* example file over 10,000 iterations. Below are the resulting orientation projections from the LGN On layer to V1.

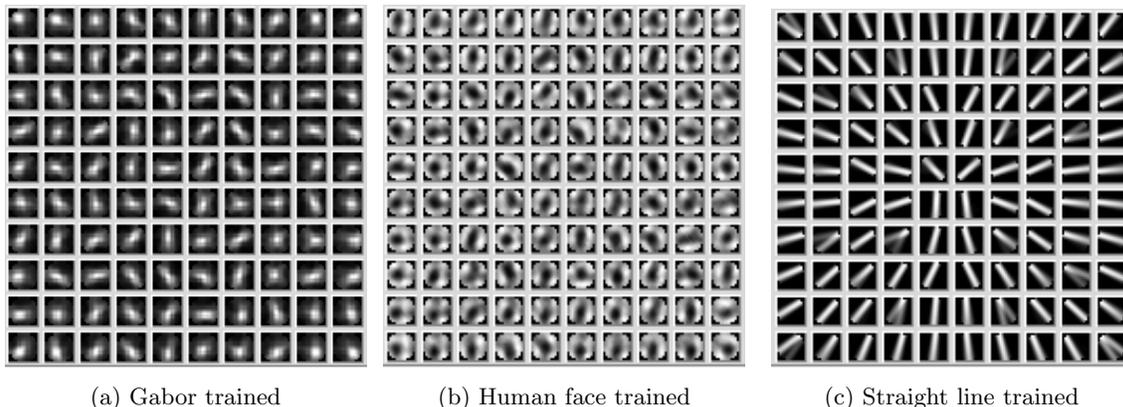


Figure 12: Projection weights from LGN On to V1

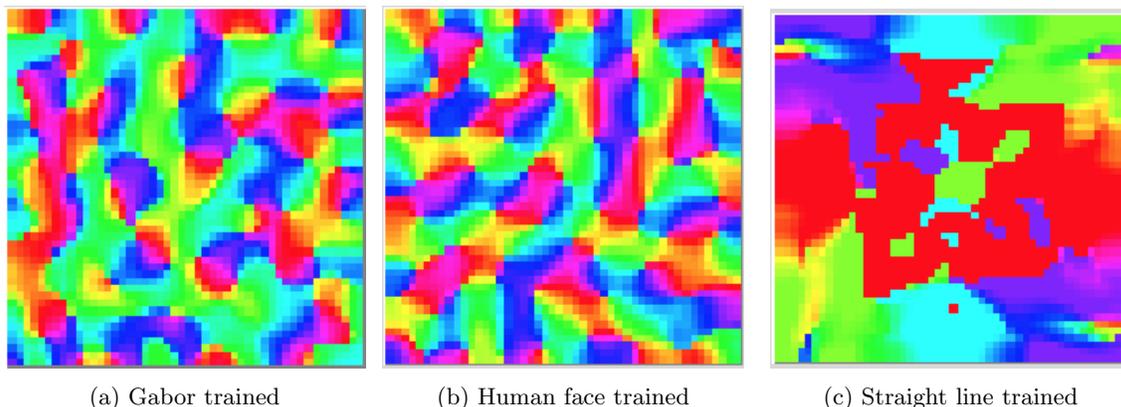


Figure 13: Orientation preference in V1

The model above utilized a single photoreceptor sheet, two RGC/LGN sheets (ON/OFF), and a single V1 cortical sheet. As the images were presented discontinuously, a settling and reset sheet was used to allow the activity in V1 to settle, and then give a non-biased estimate of the activity on a completely new independent image.

The architecture and hierarchy of the software itself is well described in the online documentation at topographica.org along with great tutorials and a user manual.

Applications and further research

In [29], Topographica was utilized to show how the development of complex cells can arise from the laterally-connected models. The basis for this implementation was to use two layers of neural sheets to represent V1, one representing layer $4C\beta$ (the layer that receives direct connection from the thalamus), and layer 2/3 (no direct connection from the thalamus, but intricately connected with layer $4C\beta$). Each V1 layer has lateral connections (short-range excitatory and long-range inhibitory), however the lateral connections in layer $4C\beta$ are several times weaker than those in layer 2/3; this burdens the majority of map development to layer 2/3. Between the two layers, there are afferent connections from layer $4C\beta$ to

2/3, but there's also feedback in the reverse direction. This brings about an interesting dynamic. Because the local connection fields in layer $4C\beta$ are initially retinotopically scattered with intrinsic noise variations in activity, the units in that layer develop more variation in phase preference locally, then they would with stronger connections. The afferent projection from layer $4C\beta$ pools together this local scattering of phase to generate a pseudo-phase invariant response in the postsynaptic layer 2/3 units. This phase-invariance is characteristic of a complex cell. The complex cells still have orientation preferences as the lateral connections in layer 2/3 are significant enough to correlate firing between similar orientation features in the input. Additionally, the feedback projection onto layer $4C\beta$ allows for the orientation preferences to be consistent while traversing the cortical layers (as was mentioned in the *Early visual system* section above).

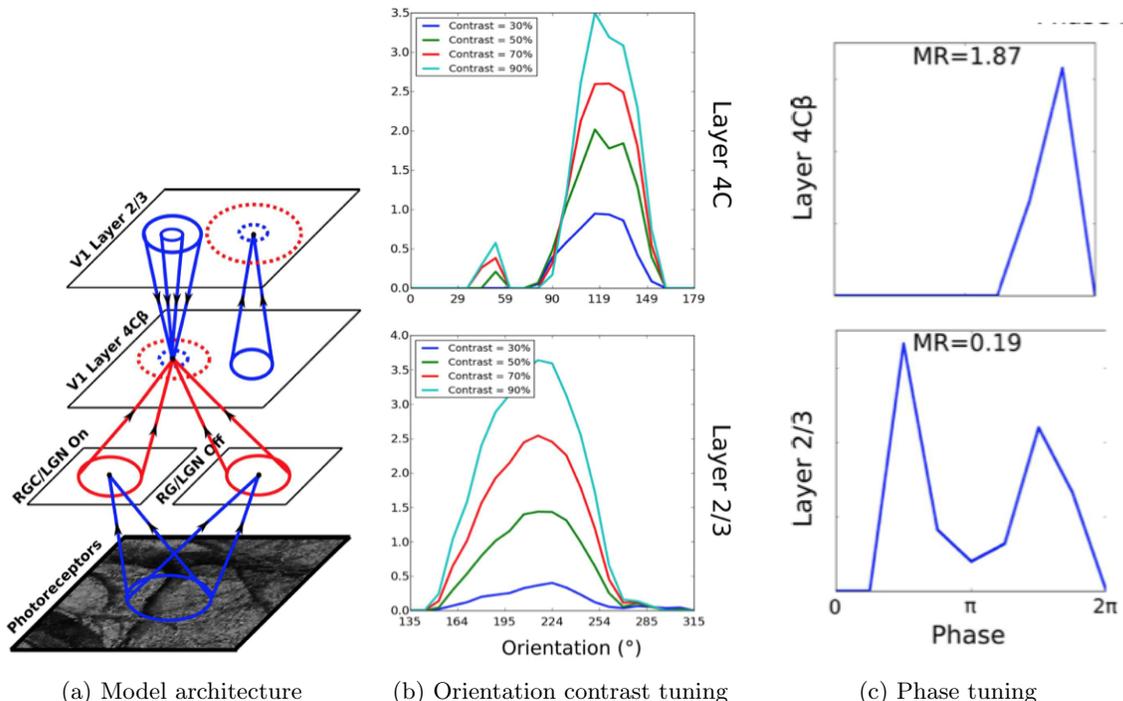


Figure 14: Simple and complex performance in model

The training of the network that went into this model was very elaborate to simulate the development of the visual system from prenatal to maturity. An image presentation of retina waves were exposed to the network a consecutive number of times (translating and rotating the image slightly each time), allowing the network to settle in between over several time steps. In between the translation/rotation of the next stage, a blank constant stimulus was introduced for one iteration to allow the cortical activities to settle. This retinal wave stage is supposed to represent the prenatal architecture development, prior to eye opening. After 50,000 iterations of this, the natural images from [30] began to train the network, simulating an already trained network being exposed to natural stimuli to hone and shape the pre-existing architecture. This was done over another 50,000 iterations with the same update scheme as before.

Another experiment was designed to use Topographica's trained model to derive spatiotemporal receptive fields via autocorrelation. The autocorrelation technique involves assuming the presence of a linear spatiotemporal filter, $D(x, y, t)$, as the reduction of the entire processing pathway from photoreceptors to the cortex, the response of which is be given by

$$L(t) = \int_0^\infty ds \int dx dy D(x, y, s) c(x, y, t - s), \quad (20)$$

where $c(x, y, t)$ is the spatiotemporal stimulus, and $L(t)$ is the time-varying response of any neuron [6]. Assuming a white noise input for $c(x, y, t)$, the autocorrelation, Q_{Lc} can then be calculated using

$$Q_{Lc}(x, y, \tau) = \frac{1}{T} \int_0^T dt L(t) c(x, y, t + \tau). \quad (21)$$

From the autocorrelation function the original filter can be reconstructed by scaling by the variance of the white noise stimulus. For unit variance,

$$D(x, y, \tau) = \frac{Q_{Lc}(x, y, -\tau)}{\sigma_c^2} = Q_{Lc}(x, y, -\tau) \quad (22)$$

For this type of analysis to be done, some notion of continuity must exist in the network. Both the GCAL and LISSOM models implemented in Topographica are trained on discontinuous inputs, so the sheets reset their activity after each image presentation to compensate. A relatively new model being worked on by the Topographica developers, TCAL (Temporally CALibrated GCAL), is designed precisely for this. Instead of facilitating the settle-reset cycle for each image presented to the network, the TCAL model implements continuously adapting activities in the neural sheets, allowing the residual activity from a previous presentation to impact the connection strengths and activities of the following presentation. With this model in place, a network can be trained using the GCAL model (assuming no continuous input stream is available), and its connection weights exported to a TCAL model. This TCAL model can then be ran, and the activity of V1 and the photoreceptors can be recorded for every input presentation of a uniform variance Gaussian white noise signal. After many iterations, Eq. (21) and Eq. (22) can be used to derive a spatiotemporal receptive field for each cortical unit. This experiment has not been conducted in full yet, but will be in further exploration of the package.

There are other experiments yet to be developed or conducted that are of interest here as well. Finding a correlation between the calculated receptive field of a cortical neuron and its afferent projections from the subcortical layers is of interest. With this correlation, a quantitative impact of the lateral connections in V1 can begin to be formed. As Topographica is primarily suited for map development, there are not a lot of built-in tools for evaluating the encoding ability of a given network. Developing a model to evaluate this is also of interest. The integration of spiking neuron models into this large-scale approach is a valuable insight to the validity of the models implemented in Topographica. It would be of value to develop a protocol for spike time integration into this framework to test the theory of self-organizing map on spiking neurons. Finally, it is important to note that a lot of the additions made to the models came about from observations during simulation. Though these justifications were presented and explained in various papers, repeating these experiments oneself will help develop an intuition on why these additions are necessary and perhaps provide insight on what modifications may help performance.

Acknowledgements

Special thanks to Yiyin Zhou and Nikul Ukani for directing my progress and giving feedback and suggestions on a daily basis. Thanks to Prof. Lazar for supervising the project. And thanks to the rest of the Bionet lab for various tips and consulting when I had questions.

References

- [1] Richard H. Masland. The fundamental plan of the retina. *Nature neuroscience*, 4(9):877–886, 2001.
- [2] Gautam B. Awatramani and Malcolm M. Slaughter. Origin of transient and sustained responses in ganglion cells of the retina. *The Journal of Neuroscience*, 20(18):7087–7095, September 2000.
- [3] Robert Shapley and Christina Enroth-Cugell. Visual adaptation and retinal gain controls. *Progress in Retinal Research*, 3:263–346, 1984.
- [4] Stewart A. Bloomfield. Relationship between receptive and dendritic field size of amacrine cells in the rabbit retina. *Journal of Neurophysiology*, 68(3):711–725, September 1992.
- [5] David H. Hubel. *Eye, Brain, and Vision*. W.H. Freeman, 2 edition, May 1995.
- [6] Peter Dayan and L.F. Abbot. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press, 2001.
- [7] Risto Miikkulainen, James A. Bednar, Yoonsuck Choe, and Joseph Sirosh. *Computational Maps in the Visual Cortex*. Springer Science+Business Media, Inc., 2005.
- [8] James A. Bednar. Building a mechanistic model of the development and function of the primary visual cortex. *Journal of Physiology (Paris)*, 106:194–211, September 2012.
- [9] Fabrizio Gabbiani and Steven James Cox. *Mathematics for Neuroscientists*. Academic Press, first edition, August 2010.
- [10] Heather J. Chisum and David Fitzpatrick. The contribution of vertical and horizontal connections to the receptive field center and surround in v1. *Neural Networks*, 17(5-6):681–693, May 2004.
- [11] Fred Rieke and Michael E. Rudd. The challenges natural images pose for visual adaptation. *Neuron*, 64(5):605–616, December 2009.
- [12] Matthias Bethge. Factorial coding of natural images: how effective are linear models in removing higher-order dependencies? *Journal of the Optical Society of America*, 23(6):1253–1268, 2006.
- [13] Odelia Schwartz and Eero P. Simoncelli. Natural signal statistics and sensory gain control. *Nature neuroscience*, 4(8):819–825, August 2001.
- [14] David H. Hubel and Torsten N. Wiesel. Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology*, 28:229–289, 1965.
- [15] Kimberly Gerrow and Antoine Triller. Synaptic stability and plasticity in a floating world. *Current Opinion in Neurobiology*, 20(5):631–639, 2010.
- [16] W. Singer. The formation of cooperative cell assemblies in the visual cortex. *Journal of Experimental Biology*, 153:177–197, October 1990.
- [17] Dan D. Stettler, Aniruddha Das, Jean Bennett, and Charles D. Gilbert. Lateral connectivity and contextual interactions in macaque primary visual cortex. *Neuron*, 36:739–750, November 2002.
- [18] Joseph Sirosh, Risto Miikkulainen, and James A. Bednar. Self-organization of orientation maps, lateral connections, and dynamic receptive fields in primary visual cortex.
- [19] Joseph Sirosh and Risto Miikkulainen. Cooperative self-organization of afferent and lateral connections in cortical maps. *Biological Cybernetics*, 71:65–78, 1994.
- [20] Armen Stepanyants, Luis M. Martinez, Alex S. Ferecskó, and Zoltán F. Kisvárdy. The fractions of short- and long-range connections in the visual cortex. *Proceedings of the National Academy of Sciences*, 2009.

- [21] Charles D. Gilbert, Joseph A. Hirsch, and Torsten N. Wiesel. Lateral interactions in visual cortex. *Cold Spring Harbor Symposia on Quantitative Biology*, 55:663–677, 1990.
- [22] Edward M. Callaway and Lawrence C. Katz. Emergence and refinement of clustered horizontal connections in cat striate cortex. *The Journal of Neuroscience*, 10(4):1134–1153, April 1990.
- [23] Tuevo Kohonen. Physiological interpretation of the self-organizing map algorithm. *Neural Networks*, 6:895–905, 1993.
- [24] D.A Prince and J. Huguengard. Functional properties of neocortical neurons. In *Neurobiology of Neocortex*, Dahlem Conference Reports, pages 153–176, New York, 1988. John Wiley and Sons.
- [25] Joseph Sirosh and Risto Miikkulainen. Topographica receptive fields and patterned lateral interaction in a self-organizing model of the primary visual cortex. Technical Report AI94-225, The University of Texas at Austin, Austin, TX, December 1994.
- [26] Judith S. Law, Jan Antolik, and James A. Bednar. Mechanisms for stable and robust development of orientation maps and receptive fields. Technical Report EDI-INF-RR-1404, Institute for Adaptive and Neural Computation, University of Edinburgh, Edinburgh, Scotland, UK, 2011.
- [27] May 2013.
- [28] James A. Bednar. Topographica: building and analyzing map-level simulations from python, c/c++, matlab, nest, or neuron components. *Frontiers in Neuroinformatics*, 3(8):1–9, March 2009.
- [29] Jan Antolik and James A. Bednar. Development of maps of simple and complex cells in the primary visual cortex. *Frontiers in Computational Neuroscience*, 5(17):1–19, April 2011.
- [30] Christoph Kayser, Wolfgang Einhauser, and Peter Konig. Temporal correlations of orientations in natural scenes. *Neurocomputing*, 52-54:117–123, 2003.