論文 / 著書情報
Article / Book Information

題目(和文)	方位選択性を獲得する神経回路網モデルの数学的解析
Title(English)	Mathematical analysis of neural network models for emergence of orientation selectivity
著者(和文)	山崎匡
Author(English)	
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第5176号, 授与年月日:2002年3月26日, 学位の種別:課程博士, 審査員:
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第5176号, Conferred date:2002/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Mathematical Analysis of Neural Network Models for Emergence of Orientation Selectivity

by

Tadashi Yamazaki

A dissertation submitted
to the Department of Mathematical and Computing Sciences
in partial fulfillment of requirements for the degree of

DOCTOR OF SCIENCE

at

TOKYO INSTITUTE OF TECHNOLOGY

January 2002

Acknowledgments

I am deeply indebted to my supervisor, Professor Osamu Watanabe, for his encouragement and support throughout preparing this thesis. He also has been very supportive and has always discussed any type of problems with a positive attitude. I also thank all the members of his laboratory for their enjoyable discussions and friendship.

I am immensely grateful to Professor Shigeru Tanaka, the team leader of the laboratory for Visual Neurocomputing in The Institute of Physical and Chemical Research (RIKEN) for giving me the opportunity to join his laboratory and seminars. It was a great opportunity to study *on-going* neuroscience. I also would like to thank all the members of his laboratory for their valuable discussions and friendly treatment.

I worked for the Centre for Information Science in Tokyo Woman's Christian University throughout my Ph.D course. Without their financial support, I had to quit studying. On the other hand, it was a pleasure to be concerned in working on the whole university network with talented staffs.

I would like to thank anonymous referees of Neural Networks for their rapid review. If it was late, this thesis would not exist.

I would like to thank Professor Masaharu Taniguchi, who proofread this thesis.

All computer simulations were performed under the support of a number of free / open source softwares, which include gcc, Linux, XFree86, TEX, Perl, xemacs, KDE, Qt, gnuplot, MT, mpich, and a number of GNU softwares. I would like to thank the developers of these softwares. May the source be with you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The emergence of orientation selectivity is explained by the self-organization of synaptic connections from the Lateral Geniculate Nucleus (LGN) to the primary visual cortex. That is, by receiving inputs of oriented visual stimuli or random noise, neurons in the primary visual cortex modify these synaptic connections given initially at random and *learn* to be orientation selective in an unsupervised manner. In fact, various models for the emergence of orientation selectivity have been proposed. This thesis aims to mathematically reveal the underlying mechanisms of some well-known orientation selectivity models.

Most neurons in the primary visual cortex are tuned to respond to the presentation of light/dark borders of a particular orientation [36, 38]. In contrast, neurons in the lateral geniculate nucleus (LGN), the inputs from which cortical neurons receive, are not orientation selective; they respond equally to bars of all orientations and well to non-oriented stimuli [37]. The physiological mechanism of orientation selectivity is getting clearer. Hubel and Wiesel [38] proposed that orientation selectivity arises in cortical simple cells. Simple cell receptive fields are composed of adjacent, oriented ON- and OFF-subregions of ganglion cells, and since the subregions respond to elongate light stimuli, simple cells become orientation selective. Much current evidence supports this model [10, 94].

On the other hand, the developmental mechanism of orientation selectivity remains unsolved. It has often been proposed that orientation selectivity emerges through a process of activity-dependent synaptic modification. However, because orientation selectivity arises early in development, before birth in monkeys [41] and before eye opening in kittens [39], it has not been possible to test this hypothesis directly. Many studies have reported that orientation selectivity, once established, is sharpened by neural activity, and have addressed whether preferred orientation can be altered by visual experience.

Meanwhile physiological studies have tackled to the developmental mechanism of orientation selectivity, its theoretical foundation has grown up. Theoretical

study attempts to give a mathematical description of the development, which we refer as a *model*. In 1973, von der Malsburg first proposed a mathematical

Table 1.1: Some models for orientation selectivity.

Class	Author(s)	Remark
Phenomenological	Swindale [90, 92]	
	Rojer and Schwartz [79]	
Correlation-based	Linsker [49–51]	
	Miller [58, 60]	
	Tanaka [64, 95]	
	Shouval and Cooper [81]	
Dimension reduction	Obermayer et al. [73, 74]	Low dim. SOM
	Riesenhuber et al. [78]	High dim. SOM
	Durbin and Mitchison [21]	Elastic net
	Goodhill and Willshaw [31]	Elastic net
Others	Malsburg [103]	

model of the primary visual cortex, and he showed that the model cortex developed biologically realistic orientation columns by a series of presentations of bar type stimuli [103]. However, because orientation selectivity arises early in development before getting visual stimuli of light/dark borders, the presentations of bar type stimuli cannot be justified physiologically. To address this issue, some other models assumed radially symmetric stimuli realized by activity-dependent, correlation-based synaptic competition between ON-center and OFF-center inputs. These, so-called *correlation-based*, models successfully reproduced realistic orientation columns [49, 50, 58, 60, 64, 81, 95]. Another approach to the developmental origin of orientation selectivity also established in view of information processing. A visual stimulus can be assumed as a multi-dimensional vector describing the configuration of activation of geniculate cells, while the visual cortex can be considered as a two-dimensional surface. We expect the brain represents the multi-dimensional vector space into the two-dimensional surface as much as possible. Kohonen's SOM [46, 73, 74, 78] and elastic net [21, 31] are classified in this group. Thus far enormous models for the emergence of orientation selectivity have arisen.

Then, did such model studies completely clarify the developmental mechanism? In other words, did such studies clarify why their models could show orientation selectivity? Rather, they investigated the outputs of the models, not the models themselves. They confirmed that orientation selectivity emerged from random connections and random noise inputs by computer simulations, then they investigated the property of the obtained orientation maps numerically, that is, they focused on whether the simulation results conformed to experimental facts under

the acceptance that the models surely developed orientation columns. For example, in his paper [60], Miller studied his model as follows. First he proposed his correlation-based model for orientation map formation then he performed computer simulations and confirmed that his model developed orientation maps. After that he calculated some properties of the maps numerically including the spatial frequency, phase, power spectra, orientation tuning and singularities, then he compared these results to the physiological facts experimentally obtained from cats and monkeys. Thus the main issue of his paper was the calculation and comparison of the properties, not the understanding why his model could develop orientation maps. Therefore they did not address directly to the above question.

Let us notice again that the model studies so far investigated the outputs of the models, not the models themselves. Why don't we investigate the models straightforwardly? If we understand the core of a model, we may be able to answer to the question. Then, is there a way to do that? Yes, there is. It is a mathematical analysis. A mathematical analysis can reveal everything about the model showing the following. (1) A mathematical description of the solution; (2) A mathematical description of the self-organization process. By these, we can understand the following. (1) The roles of functions; (2) The relationship between a parameter setting and its result.

Furthermore, due to the lack of understanding of the developmental mechanism, a number of models have been proposed which have different descriptions but are essentially identical. Hence the model study seems to be in saturation. Although there have been already some survey and comprehensive reference, they are not yet strong enough to clarify the essential differences of models and roles of assumptions or conditions, because these surveys are also based on computer simulation results. By just proposing a new model, we cannot expect further advance of model study. Also for this, a mathematical analysis can work quite effectively.

In this thesis, we claim the importance of a mathematical analysis by some demonstrations. We show the following facts on these models. (1) The role of a constraint which leads a competition among synapses; (2) The essence of the correlation-based mechanism for a single neuron's case; (3) The essence of the same mechanism for many neurons' case. The first result implies the importance of competition among synapses for the emergence of orientation selectivity. The second and third results clarify the developmental mechanism of orientation selectivity under a correlation-based model, and as a result, we understand the relationships between a parameter setting and the configuration of resulting orientation columns.

In the following, we briefly summarize these topics.

1.1 A role of constraint in self-organization

In many of mathematical models, competitive learning, or more specifically, competitive variants of Hebb's rule have been used as a key principle. A *Hebb rule* is a simple rule for updating, e.g., the weight of connection between two neurons. The rule just says that the connection between two neurons is strengthened if they both become active simultaneously. This rule has been used widely for neural network learning. Von der Malsburg constrained this updating rule so that the total connection weight of one neuron are kept under some bound in his pioneering paper [103]. We call this variation of Hebb rule a *constrained Hebb rule*. He showed through computer simulations that orientation selectivity is surely developed with his constrained Hebb rule.

Since the work of von der Malsburg, many models have been proposed, and some have been theoretically analyzed in depth. For example, a feature of various constrained Hebb rules as a learning mechanism has been discussed in [62]. Yet, the question of why orientation selectivity is obtained by following a constrained Hebb rule has not been addressed. Note that the development of orientation selectivity is different from ordinary learning in the sense that a neuron (or, a group of neurons) establishes a preference to one particular orientation from given (more or less) uniformly random orientation stimuli. We discuss why and how one feature from equally good features is selected with a constrained Hebb rule.

In order to simplify our analysis, we propose a simple probabilistic game called *monopolist game* for abstracting Hebb rules. In monopolist game, an updating rule corresponds to game's rule, and the selectivity is interpreted as that a single winner of a game — monopolist — emerges. Then we prove that a monopolist emerges with probability one in games following a von der Malsburg type rule. On the other hand, we showed theoretical evidence supporting that (i) the chance of having a monopolist is low without any constraint, and (ii) a monopolist emerges even under a rule with a weaker constraint. These results indicates that the importance of constraint in Hebb rules (or, more generally, competition in learning) to select one feature from equally good features.

We also analyzed how fast the monopolist emerges in games following a von der Malsburg type rule. This analysis can be used, in future, to estimate the convergence speed of constrained Hebb rules.

1.2 Emergence of orientation selectivity for one single neuron

Since Linsker [49–51] proposed a multi-layer feed-forward neural network model, which develops center-surround or oriented receptive fields (RFs) closely resembling

those found in the physiological visual system, it has been thoroughly investigated from several viewpoints to explain how these RFs are developed, or why *symmetry breaking* occurs.

Energy function minimization [45], principal component analysis [54, 100], infomax principle [52], and fixed point analysis [25] are the approaches used to address the above issues, and these studies have given deep insights of this model both analytically and numerically.

Nevertheless, a clear explanation seems still lacking, at least that reveals how oriented RFs are developed at layer $F \rightarrow G$. Kammen & Yuille [45] and Linsker [52] have not analyzed the structure of RFs. In particular, Kammen & Yuille indicated that symmetry breaking occurs in Linsker's model, however, they did not analyze the configuration corresponding to the lowest energy level, namely, the structure of RFs. Instead they depended on computer simulations. Other studies [25, 54, 100] concentrated on the Gaussian correlation function at layer $B \rightarrow C$, and the Mexican-hat correlation function at layer $F \rightarrow G$ has not been investigated thus far, although the dynamics is totally different. MacKay & Miller [54] have discussed the development in section 8 of their paper, however, their argument is not strong enough. The condition in which oriented receptive fields are developed has not been shown. We focus on the Mexican-hat correlation function and investigate its role in the development of oriented RFs mathematically.

We also focuses on the role of an arbor function, which determines the area consisting of *effective* synaptic connections to a postsynaptic cell. In previous papers, this function has been assumed to be a Gaussian function, since it has been introduced to define a synaptic density. As a result, synaptic weights are modified in a weighted manner, the weights at the center grow faster than the surrounding ones. However in the original Linsker's simulations, this is not the case. Thus a Gaussian function may not be appropriate. On the other hand, we still need an arbor function, since the size of the presynaptic layer, i.e., the size of the covariance matrix, must be finite. Therefore it is possible that the arbor function should be assumed a step function to determine this size. We consider an arbor function of the step type, as well as of the Gaussian type.

We perform the Fourier transform (FT) of the modification rule twice and obtain the solution under the assumption that the FT of the correlation function is a sharp bi-modal function with two clear peaks. The solution is expressed as a weighted sum of Fourier components defined with respect to the number of radial nodes, where the spatial frequency of these components is determined by the FT of the correlation function, and the weights grow exponentially with respect to the time and the two FTs of the arbor function. Since the difference between the weights of any two components grows exponentially as the self-organization proceeds, the formation of the RF will be conquered by one of the components (or by a combination of several components if the weights of such components grows

at the same speed).

From our analysis, we have the following results. In the Gaussian arbor function case, only center-surround RFs are developed. On the other hand, in the case of the step arbor function, we see that a rich variation of RFs is developed, for example, multi-lobed RFs can be developed in addition to center-surround RFs. In summary, our mathematical analysis clarifies the roles of the arbor function as well as the correlation function. Furthermore we show how the orientation of multi-lobed RFs is determined and finally we discuss the roles of parameters k_1, k_2 in Linsker's model.

1.3 Emergence of orientation selectivity for many neurons with lateral connections

Since von der Malsburg developed a mathematical model for the orientation map formation [103], a large number of models have been proposed, and some of them have been widely reviewed and discussed in depth [23, 93]. These discussions, however, are based on numerical simulations. There have been few attempts to consider the formation mathematically. With mathematical analyses, the roles of functions are clarified and then the properties of developed orientation maps by the model are derived. We consider a correlation-based model proposed by Miller [60] and analyze it mathematically.

Although several mathematical analyses of a correlation-based model have been performed to date, the formation has not been investigated sufficiently. Shouval and Cooper [81] studied a similar correlation based model. They defined an energy function of the model and discussed the structures of receptive fields and orientation columns corresponding to the lowest energy. However, the retinotopy has not been considered in their analysis, that is, in their model, all cells in the visual cortex have the same receptive field. Wimbauer, Gerstner & van Hemmen [107] computed the shapes of receptive fields possibly developed by the principal component analysis; however, they have not mentioned how these receptive fields are arranged on a two-dimensional cortical surface. That is, the development of orientation columns has not been analyzed. To clarify these points, we take another approach and give some explanations for the above problems.

We consider some simplifications in a way similar to those reported in Wimbauer et al.'s paper [107], and then we perform the Fourier transform and compute the principal component (i.e., the fastest growing pattern) of the simplified model. By our analysis, we clarify the roles of functions as follows. (1) A Mexican-hat-type correlation function determines the spatial frequency of receptive fields; (2) The combination of a Gaussian arbor function and a Gaussian intracortical function

develops oriented receptive fields; (3) The intracortical function develops stripes of ON/OFF dominance similar to ocular dominance columns on a layer on which each cell has its receptive field if the range is sufficiently wide; (4) Preferred orientations and phases of receptive fields are determined by the stripes within the receptive fields. These observations indicate that this model develops orientation maps with the following properties. (1) It develops oriented receptive fields; (2) Preferred orientations smoothly change on the cortical surface; (3) The periodicity of orientations on the cortical surface does not appear; (4) The periodicity of phases appears; (5) Singular points appear irregularly on the cortical surface. Finally, we confirm our results by computer simulations.

1.4 Outline of this thesis

The reminder of this thesis is organized as follows.

In Chapter 2, we review physiology of the visual system. We survey the visual pathway from the retina to the primary visual cortex through the Lateral Geniculate Nucleus, and we then discuss the columnar organization of the primary visual cortex and its development.

In Chapter 3, we survey mathematical models of orientation selectivity shown in Table 1.1. We classify them by their mechanisms and first we explain the mechanisms briefly. Then for each model, we review its detail. Moreover, some of them are mathematically analyzed.

In Chapter 4, we study a role of constraint in self-organization.

In Chapter 5, we study emergence of orientation selectivity for one single neuron.

In Chapter 6, we study emergence of orientation selectivity for many neurons with lateral connections.

In Chapter 7, we discuss the details of computer simulations we performed in this thesis.

Finally, in Chapter 8, we conclude this thesis by summarizing the results presented in this thesis with some future works.

Chapter 2

Physiology of the visual system

In this chapter, we review physiology of the visual system. First we look the visual pathway from the retina to the primary visual cortex through LGN. Then we review the columnar structures in the primary visual cortex, in particular, ocular dominance columns and orientation columns. For more details, [11, 20, 70] should be referred.

2.1 Overview of visual pathway

Figure 2.1 shows the structure of visual pathway.

Visual stimulus projected to the visual field are processed by eyes, which are regarded as outpost of the brain. Stimulus projected to the left visual field are processed by the right-side retina of each eye while stimulus given to the right visual field are processed by the left-side retina. Thus the signal processing is starting from this point. Then, the right side of each retina send optic fibers to the right hemisphere while the left side does to the left hemisphere,through the optic chiasm. After that, the information are processed by LGN and sent to the visual cortex.

2.1.1 Retina

Figure 2.2 illustrates the structure of retina.

Light passes through lens and arrives to the deepest layer of the retina. There are 100 million photoreceptor cells consisting of rods and cones, which work in darkness and in daylight respectively. Photoreceptor cells can detect a single photon, besides cones can detect the wavelength of a photon, namely, contrast or color [1].

[1]Cats are color-blind because they do not have cones.

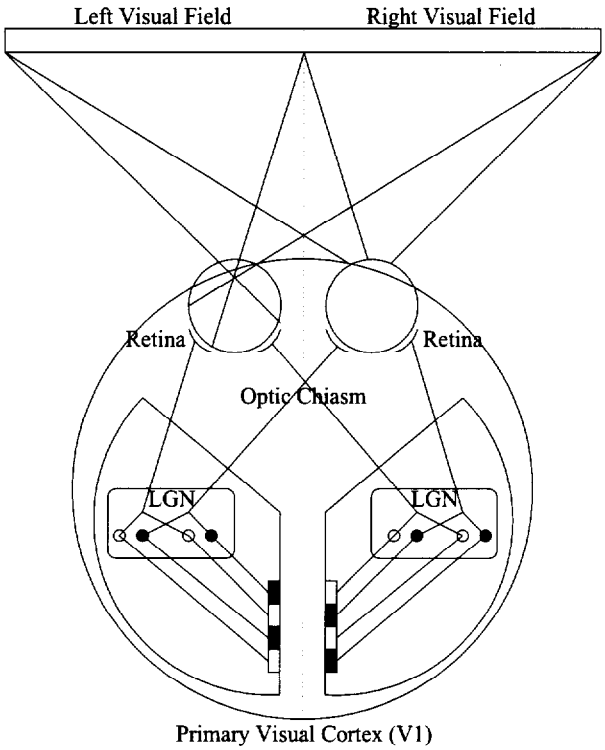

Figure 2.1: Structure of visual pathway.

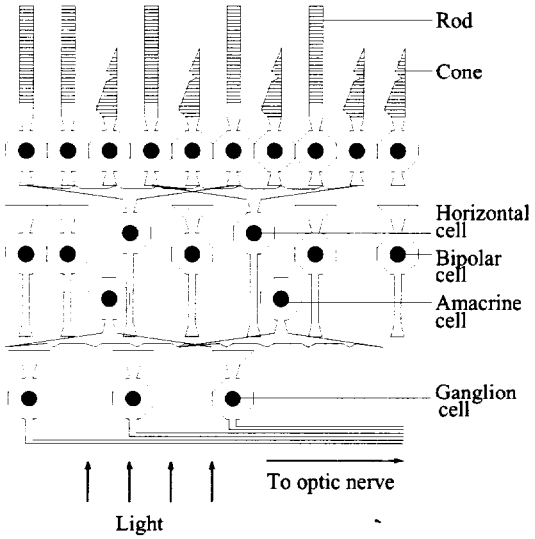

Figure 2.2: Structure of retina.

Signal from photoreceptor cells are transmitted to bipolar cells with interactions mediated through horizontal cells. Horizontal cells convolute signals coming from photoreceptors to bipolar cells. Thus a bipolar cell receives inputs from several photoreceptors. Bipolar cells and horizontal cells do not show spontaneous activity.

Here we define the notion of the receptive field. Each cell has a small area in the visual field and responds only to stimuli presented in the area. We define this area the *receptive field*. Neighboring cells have neighboring receptive fields, which means that neighboring cells respond to stimuli presented a similar area in the visual field.

Some bipolar cells are maintained in an excited (depolarized) state while the others are in an inhibited (hyperpolarized) state. The former cells are excited by the presentation of a dark spot (i.e., an annular illumination) while the latter cells are excited by a light spot (i.e., a central illumination). In the darkness, assume a light spot is presented to the receptive field center of a bipolar cell. Since the presentation inhibits the activity of the corresponding photoreceptors, if the bipolar cell is the former, it increases its inhibition and and thus decreases its activity, on the other hand if it is the latter, it inhibits its inhibition and thus it increases its activity. Meanwhile, assume a light spot is presented to the receptive field surround of a bipolar cell. The inhibited activity of photoreceptors are transmitted through horizontal cells, which increases the postsynaptic cell activity. Thus the former is excited and the latter is inhibited. We call the former and the latter OFF-center and ON-center cells respectively.

Then the signals from bipolar cells are transmitted to retinal ganglion cells (RGCs for short), with interactions mediated through amacrine cells. Amacrine cells receive inputs from bipolar cells and other amacrine cells and transmit signals to bipolar cells, amacrine cells, and retinal ganglion cells. Similary, in RGCs, there are ON-center type OFF-center type. An ON-center RGC responds best to a light spot onto the center of its receptive field, and illumination of the surrounding area reduces or suppresses its responses. An OFF-center RGC responds in opposite. Figure 2.3 illustrates the examples. RGCs send 1 million optic fibers to the higher area. It is known that the shape of receptive fields can be approximated by the

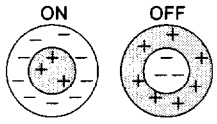

Figure 2.3: Schematic figures of receptive fields of a ganglion cell. + and − symbols respectively represent the regions on which the ganglion cell increases or reduces its response when illuminated.

following difference of Gaussians.

$$\mathrm{RF}(\alpha) = \frac{K_1}{2\pi\sigma_1^2} e^{-|\alpha|^2/2\sigma_1^2} - \frac{K_2}{2\pi\sigma_2^2} e^{-|\alpha|^2/2\sigma_2^2},$$

where α labels the position in the two-dimensional receptive field, and K_1,K_2,σ_1 and σ_2 are parameters which determines the sizes of the excitatory and inhibitory regions in the receptive field. An adaptation of parameters follows that $K_1 = 10.6'$,$K_2 = 31.8'$,$\sigma_1/\sigma_2 = 17/16$, where $30' = 0.5$ degree [84].

Ganglion cells in the cat retina can be grouped into three categories, they are, X, Y, and W type. Cells of Y type have a broad receptive field and transmit signals faster, which concern on the motion detection. X type has a small receptive field and the transmission speed is slow, which concern on the detailed visual information and color. W cells have a broad receptive field and transmit signals slowly, its function is not understood fully. In the monkey retina, the grouping consists of P type and M type, which respectively correspond to X and Y cells.

2.1.2 Lateral Geniculate Nucleus

The optic nerve fibers from each eye terminate on cells of the right and left Lateral Geniculate Nucleus (LGN for short). The principal cell type in LGN is excitatory cells known as *relay cells*; they make up approximately 65%–85% of all cells. Their axons go directly to the cortex giving no local collaterals. The remaining cells are inhibitory interneurons, which connects to the relay cells and each other [2]. Thus in the following we concentrate on the relay cells.

In the LGN of the cat, there are three layers of cells called A, A_1 and C, and layer C is further subdivided. The inputs to layers A and C come from RGCs in the contra-lateral eye. On the other hand, the inputs to layer A_1 come from RGCs in the ipsi-lateral eye [2]. The inputs are all segregated with respect to these types into layers, and terminated without overlap more than two layers. X and Y fibers end in different sublayers of A, C, and A_1. ON-center and OFF-center cells are also segregated into different sublayers. On the other hand, the macaque LGN consists of six layers and cells in layers 1 and 2 are larger than those in layers 3, 4, 5 and 6, which are called Magnocellular and Parvocellular. Layers 1, 4, and 6 are supplied by the contra-lateral eye while layers 2, 3, and 6 by the ipsi-lateral eye.

An important topological feature is the highly ordered arrangement or receptive fields within individual layers. Neighbouring regions of the retina make connections with neighboring geniculate cells, so that the receptive fields of adjacent neurons overlap most of their area, which we call the *retinotopy*. LGN cells also have

[2]Contra-lateral eye is the eye on the other side of the animal while ipsi-lateral eye is on the same side (Notice that there two LGNs in the brain).

center-surround receptive fields, of either the ON-center or the OFF-center, but the contrast mechanism is more finely tuned by more equal matching of the inhibitory and excitatory areas.

In summary the LGN is a way station in which RGC axons are sorted so that nearby cells receive inputs from the same region of the visual field [3].

2.1.3 Primary visual cortex

The primary visual cortex (V1 or area 17 based on the anatomical criteria) consists of folded plate of cells about 2mm thick. The neighboring area of V1 is called V2 (or area 18) and V2 receives inputs from V1. V1 can be divided into six layers and geniculate fibers end at layer 4. Layer 4 is again subdivided into A, B, and C, and layer $4C$ is again subdivided into $4C\alpha$ and $4C\beta$. Geniculate fibers originating in parvocellular (P) or magnocellular(M) ends to the different sublayers. Parvocellular inputs end at layer $4A$, $4C\beta$, and layer 6. Magnocellular inputs end at layer $4C\alpha$ and layer 6.

In layer 4, the inputs derived from the two eyes are segregated. The layer 4 cells receive inputs of both left-eye oriented and right-eye oriented, however, these cells respond only one of the two inputs, that is, some cells prefer the left-eye and others like the right-eye. Cells have the same eye preference are grouped and the grouping develops stripes of blobs, in which cells in the same stripe have the same eye preference. This grouping is called *ocular dominance columns*, where a column means a group of neurons in vertical direction which have the same eye dominance.

The receptive fields of layer 4 cells show several varieties. Although ganglion cells and LGN cell respond to spots of light, layer 4 cells prefer a bar type stimuli or an edge of light than spots. The response is maximal at the preferred orientation and falls off sharply as the stimulus orientation departs from the preferred orientation. The response decreases to zero at about ± 30 degrees away [85]. One type of a receptive field consists of an elongated narrow central portion with flanking two antagonistic areas, which we refer as the form of tri-lobed (Fig. 2.4 left). The center may be either excitatory or inhibitory. This type of receptive fields responds best to the bar type stimuli presented to the center of the receptive field. Another type takes the form of bi-lobed, which consists two opposite types of elongated regions (Fig. 2.4 right). This type responds best to the edge type stimuli.

Different cells have receptive fields of different orientations and positions. There are cells activated by rotating the stimulus or by shifting its position in the visual field. The distribution of inhibitory-excitatory stripes in receptive fields may not be symmetrical, or the field may consist of two elongated regions of opposite signs

[3] It does not mean that there is one-to-one correspondence between a RGC and a LGN cell. a LGN cell receives inputs from several RGCs with preserving the retinotopy.

Figure 2.4: The receptive fields of tri-lobed (left) and of bi-lobed (right). Black and white regions respectively represent optic fibers of OFF-center cells and ON-center cells in LGN.

facing each other. Fig.2.5 illustrates three examples varying its positions.

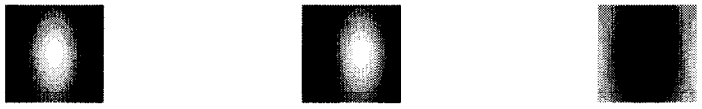

Figure 2.5: Variations of center positions.

How cortical neurons become orientation selective? The key idea is that more elaborate receptive fields are constructed by an ordered synthesis of simpler ones originating at a lower level. This idea is illustrated in Fig.2.6. The centers of concentric receptive fields of geniculate neurons are lined up in such a way that a bar of light through their centers would excite them strongly and a parallel shift of the bar into the inhibitory surround would reduce or stop the excitatory output of the cells. Several other mechanisms has been proposed [102].

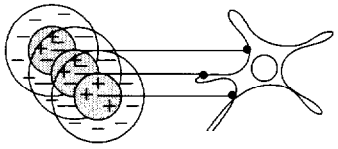

Figure 2.6: Synaptic connections for oriented receptive fields.

Some physiological studies report that the shape of oriented receptive fields can be well approximated by the following Gabor function [14, 15, 44]

$$\mathrm{RF}(\alpha, \mathbf{f}, \phi) = 2\pi\sigma^2 e^{-|\alpha|^2/4\sigma^2} e^{2\pi i(\mathbf{f}\cdot\alpha+\phi)},$$

where α stands for a position in the receptive field, f is a two-dimensional vector representing a spatial frequency, ϕ denotes the phase, and σ determines the width of the receptive field. An adaptation of parameters follows that $|\mathbf{f}| = 2$ cycle/degree and $\sigma = 0.15$ degree [55].

2.2 Columnar structures in the visual cortex

2.2.1 Ocular dominance columns

Hubel and Wiesel reported that many cells in the visual cortex had binocular receptive fields, and that the receptive fields had the same position in visual space, however cells in layer IV of the primary visual cortex are monocular and respond only to one eye [39, 40, 42]. Cells responsive to the ipsi-lateral eye and the contra-lateral eye are segregated on the cortical surface and arranged like stripes or blobs horizontally. Vertically the ocular dominance takes the form of columnar organization, and cells in the same column respond to the same eye.

In macaque monkeys, the regions are branching stripes with a fairly uniform width of around 400 μm and the overall periodicity of about 800 μm [6, 42, 47]. In cat area 17 and 18, the pattern is less regular, although it is still periodic. The overall periodicity is 0.7–1.2mm in area 17 and 1.8–2.2mm in area 18 [18].

Ocular dominance columns emerge from an initially overlapping, unsegregated distribution of left- and right-eye inputs within layer IV. Ocular dominance columns are found in macaque monkeys just after birth, but not found in kittens just after birth. In cats' case, it emerges postnatally. What the important mechanisms for the development of ocular dominance columns are investigated by much studies. Some of them are reviewed as follows.

In monocular deprived animals (animals whose one eye is shut by suturing eyelids so that visual stimuli are prevented), there is a morphological difference of the ocular dominance columns, compared to the normal animals. In normal animals, the width of columns corresponding to the ipsi lateral eye and that to the contra lateral eye are balanced, however in deprived animals the balance changes, columns corresponding to the closed eye shrink while these corresponding to the normal eye increase in width. This has been found for both monkeys [35] (an example is shown in Fig. 6 of Chapter 18 in [70]) and kittens [80]. This observation suggests that the segregation is a result of competitive process between neural activities of both eyes.

In binocular deprived monkeys grown in darkness, a normal pattern of ocular dominance columns are found [48]. This result indicates that visually driven activity is not required for the development of ocular dominance columns in monkeys. However in binocular deprived cats, ocular dominance columns are not found in area 17 [65, 89, 91]. On the other hand in cats' area 18, segragation appears [65, 89, 91].

If spontaneous firing of retinal ganglion cells is stopped, the segregation fails in both area 17 and 18 of kittens [87]. This suggests that retinal activity is required for segregation in kittens. In macaques' case, spontaneous firing of ganglion cells is enough.

2.2.2 Orientation columns

Hubel and Wiesel found that most neurons in the visual cortex selectively respond to oriented light stimulus like a bar or a edge [36, 38, 40].

Cells in the same column have the same orientation preference, thus orientation columns are formed vertically through the primary visual cortex [38, 40]. On the other hand, cells in the neighboring columns have similar orientation preferences, thus the preferred orientations smoothly change on the cortical surface. Hubel and Wiesel first found this smooth change by electrode recording [41] and later the development of the technique of optical recording enabled us to obtain the overall orientation maps [7]. This change is periodic, so that all orientations of 180 degrees appear within about 1mm×1mm cortical tissue in cats' area 17 [1, 18] and in area 18 [53]. Some of important features of the orientation columns are as follows.

1. *periodicity* of preferred orientations.

2. *unperiodicity* of phases [16].

3. *linear zones*, the regions in which the iso-orientation domains are parallel like rainbow.

4. *singularities*, the point at which all orientations meet like a pinwheel [8]. Two types of singularities, positive (rotating clockwise for clockwise movement) and negative (anticlockwise rotation), exist. The distribution of singularities are not random, moreover approximately 80% of nearest-neighbor singularities were of opposite sign [72].

In macaque monkeys, orientation preferences appear soon after the birth [41] and orientation columns can be found by at least 3.5 postnatal weeks [6]. In cats, patterned visual experience begins when the eyes open about a week after birth, and then orientation columns are found to be present by 2 weeks [12]. Does the visual experience of natural scenes play an important role for the development of orientation columns? Clair et al. [12] investigated the role of visual experience of natural scenes so that they performed imaging of intrinsic optical signals and single-unit microelectrode recordings in the visual cortex of binocularly deprived cats by bilateral lid suture from before the time of eye opening. They found that orientation columns were found by 3 weeks and then those columns disappeared. They concluded that the visual experience of natural scenes did not contribute to the development of orientation columns but to its maintenance. It implies that random noise input of light stimulus coming through eye lids are enough to develop orientation columns, which justifies an assumption in correlation-based models [58, 60, 64]. They also found that orientation maps were powerfully dominated by

the contra lateral eye, and experience was needed for responses to the other eye to become strong.

Patterned visual experience can change orientation preferences of neurons. Blakemore and Cooper raised kittens in a cylinder in which vertical stripes are painted on the wall. They found that the kittens could not recognize the horizontal light bars or edges like the border of a table [5].

2.2.3 Structural relationships between ocular dominance columns and orientation columns

Singularities tend to appear in the centers of ocular dominance stripes in macaque monkeys and iso-orientation domains and linear zones tend to cross the borders of ocular dominance stripes at right angle [6, 71].

Chapter 3

Review of models

In this chapter, we review some mathematical models of orientation selectivity listed in Table 3.1.

Table 3.1: Summary of models.

Class	Author(s)	Remark
Phenomenological	Swindale [90, 92]	
	Rojer and Schwartz [79]	
Correlation-based	Linsker [49–51]	
	Miller [58, 60]	
	Tanaka [64, 95]	
	Shouval and Cooper [81]	
Dimension reduction	Obermayer et al. [73, 74]	Low dim. SOM
	Riesenhuber et al. [78]	High dim. SOM
	Durbin and Mitchison [21]	Elastic net
	Goodhill and Willshaw [31]	Elastic net
Others	Malsburg [103]	

Model study was opened up in 1973 by von der Malsburg's pioneering paper [103]. Although this model has been criticized by now in several aspects, this model contains all the essential features on which more recent models based. These are, (1) Hebbian synapses; (2) correlated or spatially patterned activity in the afferents to cortical neurons; (3) fixed connections between cortical neurons which are locally excitatory and inhibitory in surround; (4) normalization of synaptic weights. Linsker [49–51], Miller [58, 60] and Tanaka [64, 95] have extended the scope of this model by devising linear version of the learning equations, by carrying out more extensive computations, and by making modified assumptions about the properties of the input correlations. These are called "correlation-based" models.

Another class of models are so-called "phenomenological" models. While correlation-based models concern on itsx physiological plausibility , phenomenological models do not. What important is that a model can reproduce physiological phenomena, namely, orientation columns, thus any implementation can be accepted. Rojer and Schwartz [79] reported that the application of a bandpass filter to random noise showed a similar pattern of orientation columns. Swindale [90] also considered the same mechanism however he extended synaptic weights to complex values.

Applications of dimension reduction algorithms also succesfully reproduce orientation columns as well as ocular dominance columns and retinotopy. Dimension reduction is a mapping from a high-dimensional space into a two-dimensional surface with keeping the distance between elements in the original space. Therefore dimension reduction is a kind of optimization problems. For example, a visual stimulus to the primary visual cortex is regarded as a six-dimensional vector, consisting of (1) strength $\in \mathcal{R}$; (2) preferred orientation $\in [0, \pi/2)$; (3) origin $\in \{0, 1\}$ (0 for left, 1 for right); (4) color $\in \{0, 1, 2\}$; and (5) position on the visual field $\in \mathcal{R}^2$. The vector space is denoted as a "feature space". On the other hand, cortical neurons in the primary visual cortex are arranged in a two-dimensional surface, and each tends to be selective to a specific vector of the feature space. Thus, we may regard that the six-dimensional feature space is mapped into a two-dimensional cortical surface. Moreover, neibouring neurons tend to be selective to similar stimuli, which implies that two neibouring points (i.e., vectors) in the feature space are mapped to neibouring points in the cortical surface. Hence the distance between two points in the feature space is preserved in the cortical surface.

Thus we can expect that dimension reduction algorithms successfully reproduce cortical maps including orientation columns. An algorithm of this class is Kohonen's self-organizing feature map (SOM) [46]. This algorithm is widely accepted and used in many areas to solve an optimization problem like the traveling salesman problem. Another algorithm is called the "elastic net". While SOM works on "winner-take-all" basis, elastic nets do not.

3.1 Notation

Throughout this thesis, we make use of the following notations. We let bold Roman letters ($\mathbf{x}$, $\mathbf{y}$, $\cdots$) label two-dimensional position in the postsynaptic (output) layer, and Greek letters (α, β, $\cdots$) label position in the presynaptic (input) layer. We often perform the Fourier transform. In the Fourier space, we let bold Roman letters ($\mathbf{l}$, $\mathbf{n}$, $\mathbf{m}$) label two-dimensional position with respect to the Fourier transform of presynaptic position, while we let bold Roman letters ($\mathbf{u}$, $\mathbf{v}$) label two-dimensional position with respect to the Fourier transform of postsynaptic positions.

We define the following functions.

1. $w^{(t)}(\mathbf{x}, \alpha)$: synaptic weight between the postsynaptic cell at position $\mathbf{x}$ and the presynaptic cell at position α at the t th step. Thus $w^{(0)}(\mathbf{x}, \alpha)$ represents the distribution of initial synaptic weights. If the postsynaptic layer contains only one single cell, we omit $\mathbf{x}$ and denote it as $w^{(t)}(\alpha)$ for the sake of simplicity. On the other hand, in the case of which the presynaptic layer is not given explicitly (e.g., in SOM), we denote it as $w^{(t)}(\alpha)$. When we consider continuous time case, we omit the index (t) and denote it as $w(\mathbf{x}, \alpha)$.
2. $\Delta w^{(t)}(\mathbf{x}, \alpha)$: the growth of $w^{(t)}(\mathbf{x}, \alpha)$ at the t th step in descrete time.
3. $\dot{w}(\mathbf{x}, \alpha)$: the growth rate of $w(\mathbf{x}, \alpha)$ in continuous time.
4. $o^{(t)}(\mathbf{x})$: activity of the postsynaptic cell at position $\mathbf{x}$ at the t th step.
5. $i^{(t)}(\alpha)$: activity of the presynaptic layer at the th step, or describing stimulus pattern for Malsburg's model or a feature vector in SOM.
6. $A(\mathbf{x} - \alpha)$: synaptic density or "arbor" function, describing connectivity from the presynaptic layer to the postsynaptic layer. This represents the number of synapses from the presynaptic cell at position α to the postsynaptic cell at position $\mathbf{x}$. This function is assumed time independent and dependent.
7. $I(\mathbf{x} - \mathbf{y})$: intracortical interaction function, describing connectivity from the postsynaptic cell at $\mathbf{x}$ to the postsynaptic cell at $\mathbf{y}$. In general, this function is assumed time independent. In SOM I is time dependent, in that case we denote it as $I^{(t)}$.
8. $C(\alpha - \beta)$: correlation function, describing the correlated activity between the presynaptic cell at α and the presynaptic cell at β. This function is assumed time independent.
9. f^*: value of function f after the application of a threshold.
10. $\widehat{f}$: the Fourier transform of function f.

3.2 Von der Malsburg's model

The model cortex is a hexiagonal array consisting of 169 E (excitatory) cells and the same number of I (inhibitory) cells occupying the same place for every E cells. Cells are connected together in such a way that E cells excite neibouring E cells and inhibit (via I cells) E cells a slightly greater distance away. Each E cell

receives excitatory connections from a set of 19 retinal cells and each retinal cell is connected with every E cell. Thus the configuration of the synaptic weights is represented by the matrix of 19 numbers.

The response $o^{(t)}(\mathbf{x})$ of a cortical cell at $\mathbf{x}$ to a stimulus pattern i_α, is computed as follows.

$$\frac{do^{(t)}(\mathbf{x})}{dt} = -\alpha_{\mathbf{x}} o^{(t)}(\mathbf{x}) + \sum_{\mathbf{y}} I(\mathbf{x}-\mathbf{y}) o^{*(t)}(\mathbf{y}) + \sum_{\alpha} w^{(t)}(\mathbf{x},\alpha) i^{*(t)}(\alpha), \qquad (3.1)$$

where $\alpha_{\mathbf{x}}$ is a decay constant. The first term on the right-hand side represents the decay of the response with time, the second term represents the excitatation and inhibition from the other cortical cells, and the third term represents the effect of afferent inputs.

The network is presented in turn with one of nine oriented patterns of activity in the set of retinal cells as shown in Fig. 3.1. For each presentation of a stimulus, Eqn. (3.1) is computed and the afferent synaptic weights to E cells are modified according to the rule:

$$w^{(t)}(\mathbf{x},\alpha) \propto i^{*(t)}(\alpha) o^{*(t)}(\mathbf{x}).$$

After the modification, the total synaptic weight for each cortical cell is normalized by multiplying the $w^{(t)}(\mathbf{x},\alpha)$ by a factor proportional $1/\sum_\beta w^{(t)}(\mathbf{x},\beta)$. This normalization plays an important role, which is studied in Chapter 4.

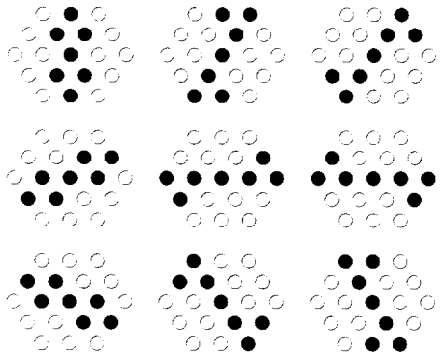

Figure 3.1: The nine different stimuli.

After 100 iterations of one presentation of all the nine different stimuli in a certain order, most of the E cells became orientation selective. Moreover the preferred orientation of neibouring E cells was similar and changed gradually on the cortical surface. The simulated orientation map showed regions of continuous change, periodicity of preferred orientation, and singularities [103].

This model was later extended to the development of retinotopy and ocular dominance columns [104, 106].

3.3 Rojer and Schwartz's band-pass filter model

Rojer and Schwartz found that similar patterns of ocular dominance columns and orientation columns can be obtained by simply band-pass filtering a random noise pattern [79]. First they simulated ocular dominance columns by applying a isotropic band-pass filter (anisotropic filters were also investigated) to a two-dimensional random noise. Let $w^{(0)}(\mathbf{x})$ be the value of the random noise given on the position $\mathbf{x}$ and $I(\mathbf{x})$ be a band-pass filter. A band-pass filter is a function that shows several clear peaks in the Fourier domain, for example, a DOG function. Since such function extracts only (i.e., filters) some components of a certain frequency (i.e., band), it is called a "band-pass" filter. They computed

$$w(\mathbf{x}) = I * w^{(0)}(\mathbf{x}) = \int_{-\infty}^{\infty} I(\mathbf{x}-\mathbf{y})w^{(0)}(\mathbf{y})d\mathbf{y},$$

and then applied a threshold as

$$w^*(\mathbf{x}) = \begin{cases} 1 & \text{if } w(\mathbf{x}) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

They reported that the obtained $w^*(\mathbf{x})$ showed a realistic pattern of ocular dominance columns.

Next they computed a similar pattern of orientation columns from $w^*(\mathbf{x})$. Their idea was the use of the gradient of $w^*(\mathbf{x})$ to represent the preferred orientaion on $\mathbf{x}$. For every position $\mathbf{x} = (x, y)$,

$$\mathrm{grad}(w^*)(\mathbf{x}) = \left(\frac{\partial w^*(\mathbf{x})}{\partial x}, \frac{\partial w^*(\mathbf{x})}{\partial y}\right)$$

represents the orientation of the stripe around $\mathbf{x}$. The preferred orientation on $\mathbf{x}$ was computed as follows.

$$\theta(\mathbf{x}) = \tan^{-1}\left(\mathrm{grad}(w^*)(\mathbf{x})\right) = \tan^{-1}\left(\frac{\partial w^*(\mathbf{x})/\partial y}{\partial w^*(\mathbf{x})/\partial x}\right).$$

Hence the preferred orientations gradually change on the surface and the change is periodic if the stripes is the shape of blobs.

3.4 Swindale's model

Swindale's model, in particular, the model for ocular dominance columns [88], is similar to that of Rojer and Schwartz, although his model was proposed much

before that their model. His model is given a two-dimensional random noise and apply a band-pass filter to the noise. A different point is that in his model the filter is applied iteratively.

Mathematically his model for ocular dominance columns computes the following equation.

$$\dot{w}(\mathbf{x}) = (1 - |w(\mathbf{x})|) \int_{-\infty}^{\infty} I(\mathbf{x} - \mathbf{y}) w(\mathbf{y}) d\mathbf{y}.$$

where $w(\mathbf{x})$ represents the value on position $\mathbf{x}$ initially given uniformly at random, and $I(\mathbf{x})$ is the following DOG function.

$$I(\mathbf{x}) = Ae^{-\lambda_1 |\mathbf{x}|^2} - Be^{-\lambda_2 |\mathbf{x}|^2}.$$

The term $(1 - |w(\mathbf{x})|)$ on the right-hand side constrains the value of $w(\mathbf{x})$ within the range $[-1, 1]$. Here we briefly analyze this model. We consider the dynamics at the very initial stages. Assuming that the fluctuation of $w(\mathbf{x})$ from 0 is very small, $(1 - |w(\mathbf{x})|)$ is almost 1. Under this situation, the above equation might be approximated by simply omitting $(1 - |w(\mathbf{x})|)$.

$$\dot{w}(\mathbf{x}) = \int_{-\infty}^{\infty} I(\mathbf{x} - \mathbf{y}) w(\mathbf{y}) d\mathbf{y}.$$

This equation is solvable performing the Fourier transform.

$$w(\mathbf{x}) = \int_{-\infty}^{\infty} \widehat{w}^{(0)}(\mathbf{n}) e^{\widehat{I}(\mathbf{n})t} e^{i\mathbf{n}\mathbf{x}} d\mathbf{n},$$

where $\widehat{f}$ denotes the Fourier transform of f, and $\widehat{w}^{(0)}(\mathbf{n})$ is the Fourier transform of the initial weights. Since I is a DOG function, $\widehat{I}$ takes the form of bi-modal. Let σ be the frequency at which $\widehat{I}$ takes the maximum value. Then we can see that the term $e^{\widehat{I}(\mathbf{n})t}$ extracts only $\mathbf{n}$ such that $|\mathbf{n}| = \sigma$, thus the equation can be restated as follows.

$$w(\mathbf{x}) \propto \int_{|\mathbf{n}|=\sigma} \widehat{w}^{(0)}(\mathbf{n}) e^{i\mathbf{n}\mathbf{x}} d\mathbf{n}.$$

This describes the band-pass filtering of the frequency σ to noise like Rojer and Schwartz's model. Hence a pattern of stripes like ocular dominance columns can be developed.

Although his model for orientation columns performs the similar computation, a different point is that $w(\mathbf{x})$ is no longer a real value but a complex value. At

initialization $w(\mathbf{x})$ is given a complex value and then this is modified by the above rule. After the computation, the orientation is calculated as follows.

$$\theta(\mathbf{x}) = \tan^{-1}\left(\Im(w)(\mathbf{x})/\Re(w)(\mathbf{x})\right).$$

It was reported that the orientation columns developed by this model have bandpass, rather than lowpass power spectra shown by more biological models. Erwin et al. evaluated and compared several models with the recent experimental findings from macaque striate cortex, and they concluded that this model showed the best performance [23].

Later these two models were combined and proposed as the model for simultaneous development of ocular dominance columns and orientation columns [92].

3.5 Linsker's model

Linsker's model [49–51] consists of seven neuronal layers called layers $A, B, \ldots, G$, respectively (Fig. 3.2). Each cell of a presynaptic layer projects connections approximating a Gaussian distribution centered on a corresponding point in the postsynaptic layer.

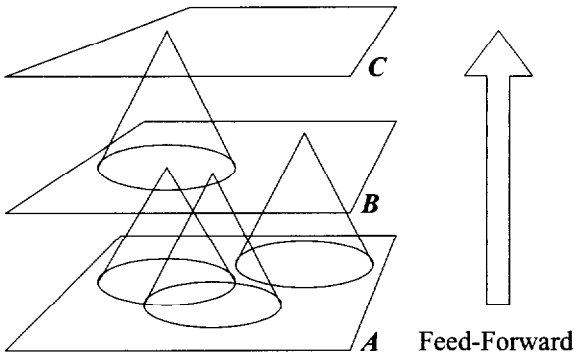

Figure 3.2: Network of Linsker's model

The activity level o of a cell in the postsynaptic layer can be computed by the linear sum of its synaptic weights $w(\alpha)$ and the activity level of cells in the presynaptic layer, plus a constant a_1; i.e.,

$$o = \sum_{\alpha} w(\alpha) i(\alpha) + a_1.$$

Here it should be noted that o can be negative or positive. Synaptic weights are modified according to a linear Hebbian rule;

$$\dot{w}(\alpha) = c_1 (o - c_2)(i(\alpha) - c_3) + c_4,$$

where $c_1, \ldots, c_4$ are constants. Synaptic weights are modified following each random presentation of an activity pattern in the presynaptic layer to the postsynaptic layer, however assuming that $w(\alpha)$ changes on a much longer time scale than the presentations, we can take an average over the ensemble of patterns $i(\alpha)$. Then, the mean rate of change of $w(\alpha)$ can be computed as

$$\dot{w}(\alpha) = k_1 + \sum_{\beta} (C(\alpha - \beta) + k_2)\, w(\beta), \tag{3.2}$$

where k_1, k_2 are constants and $C(\alpha - \beta) = \langle (i(\alpha) - i_*)(i(\beta) - i_*) \rangle$ is the covariance matrix of points α and β, and i_* is an average activity of presynaptic cells over the ensemble of presentations. Synaptic weights are constrained by an upper bound and a lower bound, so that $-w_{\min} \leq w(\alpha) \leq w_{\max}$ holds.

This network is trained sequentially, namely, layer by layer basis. We first train the connections from layer A to layer B using this noise. We assume that random and uncorrelated noise is given to layer A as inputs to the network. From the nature of uncorrelated noise, its correlation is zero. Thus the correlation function of the input activity is represented by Kronecker's $\delta(\alpha, \beta)$. Hence we make use of the identity matrix as $C^A(\alpha - \beta)$. Once the connections stabilize, the correlation of activities between layer B cells is computed and $C^B(\alpha - \beta)$ is defined. This procedure is repeated for subsequent layers.

3.5.1 Development of center-surrond receptive fields

Linsker reported that during the training of layer A to B, the activity of cells B became correlated with respect to those distance, in particular, the correlation was proportional to a Gaussian. Thus he used a Gaussian correlation function as $C^B(\alpha - \beta)$ and trained connections from layer B to layer C. Then he found that cells on layer C as developed center-surround receptive fields (Fig.2 in [51]).

This dynamics has been studied in depth. Tang [100] and MacKay and Miller [54] made the principal component analysis and they found that the combination of a Gaussian arbor function and a Gaussian correlation function led a centre-surround receptive field. MacKay and Miller also studied the roles of parameters k_1 and k_2, and they reported that by taking k_2 appropriately (e.g., large negative), this model could develop a bi-lobed receptive field. They also discussed the effects of the upper and lower bounds for synaptic weights. Feng et al. [25] considered the stability of a weight configuration varying k_1 and k_2 for fixed arbor and correlation functions. They considered three configurations, they are, a circular symmetric type, a bi-lobed type, and a tri-lobed type. They found that a circular symmetric type was stable for almost choices of k_1 and k_2. On the other hand, they reported that a bi-lobed type and a tri-lobed type were stable in a very small choices. Here,

an important thing is that a stable configuration does not mean the dominant configuration. Even if a bi-lobed type is a stable configuration under a choice of k_1 and k_2, a circular symmetric type can be the dominant configuration.

3.5.2 Development of oriented receptive fields

He then pursued the training on layers D, E, F. Since the correlational activities of cells on these neurons were the form of Mexican-hat, he made use of DOG functions as the correlation functions. Linsker reported that cells on layer G developed receptive fields of bi-modal or tri-modal shapes (Fig.1,2 in [50]).

This dynamics has not been studied so much. MacKay and Miller discussed the development in their paper [54], the discussion is not mathematical. Therefore we do not know what situation leads the development of orientation selective cells. Kammen & Yuille proposed a similar network model and developed the energy function which described the dynamics of the model. They numerically investigated the equiliblium status of the energy function and found that "symmetry breaking" occurs in their model. However, they did not discuss the configuration corresponding to the equiliblium status, namely, the structure of RFs. Linsker himself studied the development by following infomax principle [52], however he also did not discuss the structure of RFs.

In chapter 5 we study the dynamics mathematically.

3.5.3 Development of orientation columns

Since Linsker's model did not include lateral connections between cells on the same layer, it did not develop orientation columns. To address this issue, he considered the pattern of orientation columns which minimize (to 0) an "energy" E' defined as Eqn.(8) in [49] under the assumption that all cells became orientation selective.

E' is given as follows.

$$E' = -\sum_{x}\sum_{x'} \rho(|x - \mathbf{x}'|) Q^G(\theta_x, \theta_{x'}, x' - x),$$

where the sum is over all pairs of cells at x and x', and θ_x is the preferred orietation of the cell at x. $\rho(d)$ is a decreasing function (e.g., a Gaussian). $Q^G(\theta_x, \theta'_{x'}, x' - x)$ is the lateral interaction function defined as follows.

$$Q^G(\theta, \theta', d) \propto \sum_{i}\sum_{j} Q^F(|d + t_j - t_i|) c_i^{\theta} c_j^{\theta'},$$

where i indexes the connections from layer F to layer G, t_i and t_j are the positions of i and j relative to the center of their respective receptive field, c_i^{θ} is $+0.5$ or -0.5

depending on whether the connection i belongs the positive region or the negative region of the receptive field.

The developed orientation columns show a strong bias to preferred orientations. Iso-orientation domains for vertical orientations tend to be elongated in a vertical direction, conversely, iso-orientation domains for horizontal orientations tend to run in a horizontal direction (Fig. 2 in [49]).

3.6 Miller's model

The model consists of two neuronal layers (Fig. 3.3). The pre- and postsynaptic layers correspond to the LGN and the visual cortex, respectively, and cells in each layer are arranged on a two-dimensional grid. Two types of synaptic connections, 'ON' type and 'OFF' type, are assumed, and each corresponds to the synaptic connection from an ON-center LGN cell and an OFF-center LGN cell. Both layers conserve *the retinotopy*, that is, neighboring cells in the cortex and in the LGN correspond to neighboring positions on the retina. Thus, the neighborhood relationships of the visual input are conserved. The initial synaptic weights are assigned uniformly at random.

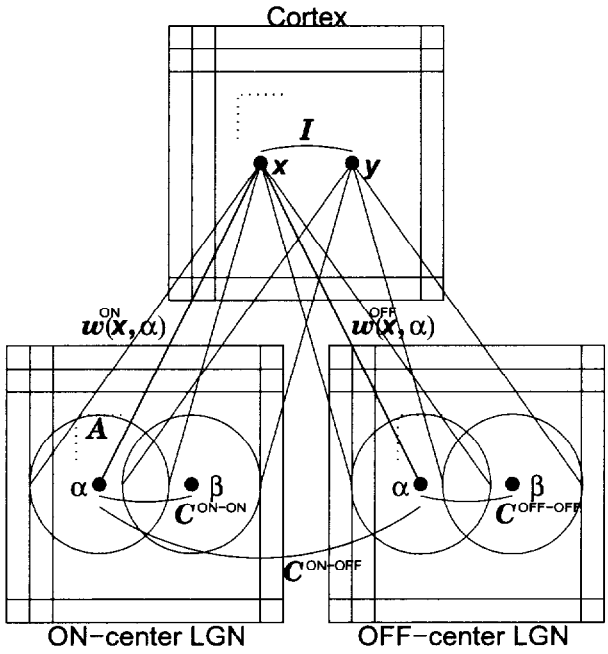

Figure 3.3: Network of Miller's model

The modification rule of synaptic weights at the t th step is

$$\Delta w^{\mathrm{ON}(t)}(\mathbf{x},\alpha) = \lambda A(\mathbf{x}-\alpha)\sum_{\mathbf{y},\beta} I(\mathbf{x}-\mathbf{y})\left[C^{\text{ON-ON}}(\alpha-\beta)w^{\mathrm{ON}(t)}(\mathbf{y},\beta) + C^{\text{OFF-OFF}}(\alpha-\beta)w^{\mathrm{OFF}(t)}(\mathbf{y},\beta)\right]. \tag{3.3}$$

A, I, C, and w are respectively the arbor function, the intracortical interaction function, the correlation functions, and the synaptic weights. The vectors $\mathbf{x},\mathbf{y}$ denote two-dimensional postsynaptic positions, and α,β indicate two-dimensional presynaptic positions. The equation for OFF synaptic weights is identical after exchange of 'ON' and 'OFF' everywhere. This linear equation was derived from more detailed nonlinear equation under some assumptions [59].

In addition to these, this model includes an upper and a lower bound for each synaptic weight and the normalization to keep the sum of synaptic weights for each postsynaptic cell constant.

This model was inspired by his previous model for ocular dominance columns development [61]. Changing ON and OFF labels in the above equations to Left and Right, we obtain the model for ocular dominance columns development. Recently, Erwin and Miller proposed the model for simultaneous development of ocular dominance and orientation maps [22].

This model has not been mathematically investigated in depth. Wimbauer et al. [107] investigated the dynamics by the principal component analysis; however, due to the improper choice of the base components, their analysis became quite complicated, and they could only compute the shapes of receptive fields possibly developed. They have not mentioned how these receptive fields are arranged on a two-dimensional cortical surface. We also study this model mathematically in Chapter6.

3.7 Shouval and Cooper's model

This model is identical to Miller's model except that this model exclude the effect of the retinotopy, that is, all cortical neurons receive inputs from the same set of retinal cells [81].

First of all, they considered a linear equation identical to Miller's model without constraints, and derived the energy function.

The output $o(\mathbf{x})$ of each neuron is defined as

$$o(\mathbf{x}) = \sum_{\mathbf{y},\alpha} I(\mathbf{x}-\mathbf{y})A(\alpha-c(\mathbf{y}))i(\alpha)w(\mathbf{y},\alpha),$$

where c is a function of $\mathbf{x}$ which represents the center of the symmetric arbor function A.

We derive the energy function H and minimize it by a Hebbian rule.

$$\begin{aligned} H &= -\sum_{\mathbf{x}} o^2(\mathbf{x}) \\ &= -\sum_{\mathbf{x},\mathbf{y},\mathbf{y}'} I(\mathbf{x}-\mathbf{y})I(\mathbf{x}-\mathbf{y}') \\ &\quad \sum_{\alpha,\alpha'} A(\alpha - c(\mathbf{y}))A(\alpha' - c(\mathbf{y}'))w(\mathbf{y},\alpha)w(\mathbf{y}',\alpha')i(\alpha)i(\alpha'). \end{aligned}$$

Then we take ensemble average on H and obtain its correlation-based model.

$$H = -\sum_{\mathbf{x},\mathbf{y},\mathbf{y}'} \tilde{I}(\mathbf{y}-\mathbf{y}') \sum_{\alpha,\alpha'} A(\alpha - c(\mathbf{y}))A(\alpha' - c(\mathbf{y}'))w(\mathbf{y},\alpha)w(\mathbf{y}',a')C(\alpha - a'),$$

where $\tilde{I}(\mathbf{y}-\mathbf{y}') = \sum_{\mathbf{x}} I(\mathbf{x}-\mathbf{y})I(\mathbf{x}-\mathbf{y}')$ is the new interaction function, and $C(\alpha - \alpha') = \langle i(\alpha)i(\alpha')\rangle$ is the correlation function.

In the following we assume that A is a step function as

$$A(\mathbf{x}) = \begin{cases} 1 & |\mathbf{x}| \le \mu_{\max}; \\ 0 & \text{otherwise.} \end{cases}$$

Then the gradient descent dynamics implied by this energy function is as

$$\begin{aligned} \frac{\partial w(\mathbf{x},\alpha)}{\partial t} &\stackrel{\text{def}}{=} -\frac{\partial H}{\partial w(\mathbf{x},a)} \\ &= \begin{cases} \sum_{\mathbf{y}} \tilde{I}(\mathbf{x}-\mathbf{y}) \sum_{|\beta - c(\mathbf{y})| \le \mu_{\max}} w(\mathbf{y},\beta)C(\alpha-\beta) & |\alpha - c(\mathbf{x})| \le \mu_{\max}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

This equation is the same as Miller [58,60]. Therefore we obtain the energy function which describe the dynamics of Miller's model. They made the simplification that $c = 0$ for further investigation. It follows

$$\frac{\partial w(\mathbf{x},\alpha)}{\partial t} = \sum_{\mathbf{y}} \sum_{\beta} \tilde{I}(\mathbf{x}-\mathbf{y})A(\beta)C(\alpha-\beta)w(\mathbf{y},\beta).$$

Here an arbor function A is again introduced to determine the effective range of β.

This model can be analyzed mathematically by the similar method shown in Chapter 6. First we restate it as follows.

$$\dot{w}(\mathbf{x}, \Delta\alpha) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(\mathbf{x}-\mathbf{y})C(\Delta\alpha - \Delta\beta)A(\Delta\beta)w(\mathbf{y}, \Delta\beta)d\beta d\mathbf{y}.$$

Its Fourier transform is as

$$\dot{\widehat{w}}(\mathbf{u}, \mathbf{n}) = \int_{-\infty}^{\infty} \widehat{I}(\mathbf{u})\widehat{C}(\mathbf{n})\widehat{A}(\mathbf{n}-\mathbf{m})\widehat{w}(\mathbf{u}, \mathbf{m})d\mathbf{m}.$$

When we fix $\mathbf{u}$, this equation is identical to Linsker's model [50, 51]. This is mathematically studied in Chapter 5, and the solution is as follows. In the case of Gaussian A,

$$w(\mathbf{x}, \Delta\alpha) \propto \int_{|\mathbf{u}|=\sigma_{\mathbf{u}}}\int_{|\mathbf{n}|=\sigma_{\mathbf{n}}} \widehat{w}^{(0)}(\mathbf{u}, \mathbf{n})e^{\widehat{I}(\mathbf{u})t}\cos(\mathbf{n}\Delta\alpha)e^{i\mathbf{u}\mathbf{x}}d\mathbf{n}d\mathbf{u}.$$

On the other hand, in the case of the step A,

$$w(\mathbf{x}, \Delta\alpha) \propto \int_{|\mathbf{u}|=\sigma_{\mathbf{u}}}\int_{|\mathbf{n}|=\sigma_{\mathbf{n}}} \widehat{w}^{(0)}(\mathbf{u}, \mathbf{n})e^{\widehat{I}(\mathbf{u})t}\sin(\mathbf{n}\Delta\alpha)e^{i\mathbf{u}\mathbf{x}}d\mathbf{n}d\mathbf{u}.$$

Here $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{n}}$ are the peak frequencies of $\widehat{I}$ and $\widehat{C}$.

When I is Gaussian, $\sigma_{\mathbf{u}} = 0$. It follows,

$$w(\mathbf{x}, \Delta\alpha) \propto \begin{cases} \displaystyle\int_{|\mathbf{n}|=\sigma_{\mathbf{n}}} \widehat{w}^{(0)}(\mathbf{0}, \mathbf{n})\cos(\mathbf{n}\Delta\alpha)d\mathbf{n} & \text{Gaussian } A, \\ \displaystyle\int_{|\mathbf{n}|=\sigma_{\mathbf{n}}} \widehat{w}^{(0)}(\mathbf{0}, \mathbf{n})\sin(\mathbf{n}\Delta\alpha)d\mathbf{n} & \text{Step } A. \end{cases}$$

Thus if A is Gaussian, then all neurons develop the same center-surround receptive fields. If A is a step function, then all neurons develop the same oriented receptive fields of even type.

On the other hand when I is DOG, $\sigma_{\mathbf{u}}$ is no longer zero. Then

$$w(\mathbf{x}, \Delta\alpha) \propto \begin{cases} \displaystyle\int_{|\mathbf{u}|=\sigma_{\mathbf{u}}}\int_{|\mathbf{n}|=\sigma_{\mathbf{n}}} \widehat{w}^{(0)}(\mathbf{u}, \mathbf{n})e^{\widehat{I}(\mathbf{u})t}\cos(\mathbf{n}\Delta\alpha)e^{i\mathbf{u}\mathbf{x}}d\mathbf{n}d\mathbf{u} & \text{Gaussian } A, \\ \displaystyle\int_{|\mathbf{u}|=\sigma_{\mathbf{u}}}\int_{|\mathbf{n}|=\sigma_{\mathbf{n}}} \widehat{w}^{(0)}(\mathbf{u}, \mathbf{n})e^{\widehat{I}(\mathbf{u})t}\sin(\mathbf{n}\Delta\alpha)e^{i\mathbf{u}\mathbf{x}}d\mathbf{n}d\mathbf{u} & \text{Step } A. \end{cases}$$

If A is Gaussian, all neurons develop center-surround receptive fields, and depending on the cortical position, they change the center types like ocular dominance columns. If A is a step function, all neurons develop oriented receptive fields of odd type, and preferred orientations gradually change on the cortical surface. The change is determined by $\widehat{w}^{(0)}(\mathbf{u}, \mathbf{n})$. This analysis is consistent with the numerical results reported by Shouval and Cooper [81].

3.8 Tanaka's model

While the components in Tanaka's model are similar to those of Miller or Linsker, its mathematical treatment is totally different. He focused on the similarity of the formations between cortical maps like ocular dominance columns and magnetic domain in the statistical physics, and he made a general theory which described the formation of cortical maps on an analogy with the system which can be characterized by a discrete state variable known as Potts spin. To justify this analogy, he made a rigorous mathematical analysis [95]. The analysis showed that in a stable equiliblium the synaptic strengths tend to assume upper or lower limiting values. The resulting binary state variable was the Potts spin of the system, and from it an associated temperature, which is related to the lifetime of an average synapse, and Hamiltonian function were derived. A standard Monte Carlo technique is used to find configurations of the system for which the value of the associated Hamiltonian is the minimum.

The formulation is as follows [1]. Let $\rho_{j,k}$ be the strength of the synaptic connection from the presynaptic cell k to the postsynaptic position j. It is modified. by the following rule.

$$\frac{d\rho_{j,k}}{dt} = \rho_{j,k}\left(1 - \sum_{k'} \rho_{j,k'}\right) + g\zeta_j \rho_{j,k}\eta_k + \varepsilon_{j,k}. \tag{3.4}$$

The first term on the right hand side represents a competition of nutrition factors supplied by a postsynaptic cell among synaptic terminals connecting to the postsynaptic cell. The second term represents the Hebbian rule, where g is a connection efficacy, ζ_j stands for the menbrane potential at the postsynaptic position j, and η_k denotes the firing rate of the presynaptic cell k. The third term is introduced to prevent $\rho_{j,k}$ from fixing and is a small constant. Defining $W_{j,k} \stackrel{\text{def}}{=} g\zeta_j\eta_k$, Eqn. (3.4) is restated as follows.

$$\frac{d\rho_{j,k}}{dt} = \rho_{j,k}\left(1 - \sum_{k'} \rho_{j,k'} + W_{j,k}\right). \tag{3.5}$$

Here Eqn. (3.5) suggests that for any postsynaptic position j, only one synaptic terminal is survived and the others are eliminated. Thus $\rho_{j,k}$ can be represented by a spin variable $\sigma_{j,k}$, where

$$\sigma_{j,k} = \begin{cases} 1 & j \text{ and } k \text{ are connected,} \\ 0 & \text{otherwise, and,} \end{cases}$$

[1] We do not follow our notation in this section because the formulation is quite different from the others.

$$\sum_k \sigma_{j,k} = 1, \quad \forall j.$$

We then compute $\overline{W}_{j,k}$, the average value of $W_{j,k}$. Let $\delta\zeta_j$ be the fluctuation of ζ_j, which can be represented as follows.

$$\delta\zeta_j = \sum_{j',k} V^{\text{post}}_{j;j'} \sigma_{j',k} \delta\eta_k,$$

where $V^{\text{post}}_{j;j'}$ stands for the lateral connection nof postsynaptic positions j and j', and $\delta\eta_k$ is the fluctuation of η_k. On the other hand, let $\Gamma^{\text{pre}}_{k,k'}$ be the correlation function which represents the correlational activity between presynaptic cells k and k'. It can be represented as follows.

$$\Gamma^{\text{pre}}_{k,k'} = \langle \delta\eta_k \delta\eta_{k'} \rangle,$$

where $\langle x \rangle$ stands for the ensemble average of x. Now $\overline{W}_{j,k}$ can be stated as follows.

$$\begin{aligned}
\overline{W}_{j,k} &= g\langle \zeta_j \eta_k \rangle \\
&= g\langle (\overline{\zeta}_j + \delta\zeta_j)(\overline{\eta}_k + \delta\eta_k) \rangle \\
&= g\overline{\zeta}_j \overline{\eta}_k + g\langle \delta\zeta_j \eta_k \rangle \quad (\langle \delta\zeta_j \rangle \text{ and } \langle \delta\eta_k \rangle \text{ are } 0) \\
&= g\overline{\zeta}_j \overline{\eta}_k + \langle g \sum_{j',k'} V^{\text{post}}_{j;j'} \sigma_{j',k'} \delta\eta_{k'} \delta\eta_k \rangle \\
&= g\overline{\zeta}_j \overline{\eta}_k + g \sum_{j',k'} V^{\text{post}}_{j;j'} \sigma_{j',k'} \langle \delta\eta_{k'} \delta\eta_k \rangle \\
&= g\overline{\zeta}_j \overline{\eta}_k + g \sum_{j',k'} V^{\text{post}}_{j;j'} \Gamma^{\text{pre}}_{k;k'} \sigma_{j',k'},
\end{aligned}$$

where $\overline{\zeta}_j$ and $\overline{\eta}_k$ are respectively the average membrane potential of the postsynaptic position j and the average firing rate of the presynaptic cell k.

The average value of the transition probability including fluctuations is calculated as follows.

$$\Pr(\sigma_{j,k'} = 1 \to \sigma_{j,k} = 1) = \langle \Theta(W_{j,k'} - W_{j,k}) \rangle_{\delta\widetilde{W}_{j,k}},$$

where $\delta\widetilde{W}_{j,k}$ is the time averaged value of $\delta W_{j,k}$. Since this can be approximated by the logistic function $T'/\exp(\overline{W}_{j,k'} - \overline{W}_{j,k})$, the separation function in the equiblium status is denoted as follows.

$$\pi_{\text{eq}}(\{\sigma_{j,k}\}) = \frac{1}{Z} \exp\left(-\frac{H}{T}\right),$$

where H is the hamiltonian, T is the effective temperature, and Z is the separation function, which are defined as follows.

$$H = -\sum_{j,j'}\sum_{k,k'} V^{\text{post}}_{j;j'}\Gamma^{\text{pre}}_{k;k'}\sigma_{j,k}\sigma_{j',k'},$$

$$Z = \sum_{\sigma_{j,k}=\{0,1\}} \exp\left(-\frac{H}{T}\right).$$

As explained the above, the following energy function is defined which describes the dynamics of the model.

$$E = -\sum_{j,j'}\sum_{k,\mu}\sum_{k',\mu'} V^{\text{post}}_{j;j'}\Gamma^{\text{pre}}_{k,\mu;k',\mu'}\sigma_{j,k,\mu}\sigma_{j',k',\mu'},$$

where $V^{\text{post}}_{j;j'}$ stands for the lateral connections between postsynaptic cells, $\Gamma^{\text{pre}}_{k,\mu;k',\mu'}$ is the correlation function between the presynaptic cell k of type μ and cell k' of type μ'. Here $\mu, \mu' \in \{-1, 1\}$ represents the type of inputs, for example, left and right eye in the case of ocular dominance columns, ON- and OFF-center in the case of orientation columns, and $\sigma_{j,k,\mu}$ stands for the synaptic connection from the presynaptic cell k of type μ to the postsynaptic cell j.

This general theory is constructed in [95], and is applicable to a variety of cortical map formations, including, ocular dominance columns [96,97], periodic orientation columns [69], CO Blobs [68], direction maps [63,99], as well as the simultaneous development of retinotopy, ocular dominance, and orientation columns [98].

In other models, a "synaptic connection" means the connection between pre- and postsynaptic cells via the axon projected from the presynaptic cell. On the other hand, in this model, a synaptic connection means the connection via a synaptic terminal on the axon of the presynptic cell. Thus we can interpret a "synaptic weight" between cells in other models as the number of synaptic terminals which connect these cells.

3.9 Applications of Kohonen's self-organizing map algorithms

Self-organizing map is an algorithm which maps a high-dimensional space into a two-dimensional space proposed by Kohonen [46]. Obermayer et al. applied this algorithm to simultaneous development of retinotopy, ocular dominance columns, and orientation columns. First we review their algorithm [73, 74].

Obermayér et al.'s application is a low dimensional version of SOM, that is, an input vector is represented as a low dimensional vector. In this version, input

vectors explicitly represent features. For example of orientation map formation, the preferred orientation is explicitly represented in the input vector. On the other hand, there is a version of SOM referred as a high dimensional model. In this version inputs do not explicitly represent features. Instead, an input consists of a number of elements, for example, the configuration of activities of a number of cells. Thus the input is similar to that of correlation-based models. While a correlation-based model changes synaptic weights of all cells for each step, a high dimensional SOM changes weights of the winner and its neibouring cells. Recently in 1998, Riesenhuber et al. reported that the high dimensional model also successfully reproduce the orientation map and the joint map of ocular dominance and orientation, although the parameter space in which such maps are obtained is quite small. We then review their model [78].

3.9.1 Low dimensional model

A visual stimulus is encoded by a five-dimensional vector, $(x, y, q\sin(2\phi), q\cos(2\phi), z)$, where x, y correspond to retinotopic position, q describes orientation selectivity, ϕ is preferred orientation, and z is the ocular dominance value. Let V be this five-dimensional space such that

$$V = \{i | i = (x, y, q\sin(2\phi), q\cos(2\phi), z)\}.$$

We map V into a two-dimensional surface $w^{(t)}(\mathbf{x}, \alpha)$. First of all, $w^{(0)}(\mathbf{x}, \alpha)$ is initialized, for any position $\mathbf{x}$, a feature vector $i(\alpha)$ chosen uniformly at random from V is assigned, thus $w^{(0)}(\mathbf{x}, \alpha) = i(\alpha)$. Then, the following procedure is repeated. (1) a feature vector $i(\alpha)$ is chosen uniformly at random from V, (2) find $\mathbf{x}$ such that $||w^{(t)}(\mathbf{x}, \alpha) - i(\alpha)||$ is the minimum (i.e., the position on which the nearest vector of the input is assigned), (3) for all neibours of $\mathbf{x}$, say $\mathbf{y}$, $w^{(t)}(\mathbf{y}, \alpha)$ is modified as follows.

$$w^{(t+1)}(\mathbf{y}, \alpha) = w^{(t)}(\mathbf{y}, \alpha) + \varepsilon(t) I^{(t)}(\mathbf{x} - \mathbf{y}) \left(i(\alpha) - w^{(t)}(\mathbf{x}, \alpha)\right),$$

where I is the following anisotropic Gaussian,

$$I^{(t)}(\mathbf{x} - \mathbf{y}) = \exp\left(-\frac{(x_1 - y_1)^2}{\sigma_{h1}^2(t)} - \frac{(x_2 - y_2)^2}{\sigma_{h2}^2(t)}\right),$$

(4) if the change of w is sufficiently small, then halt, (5) otherwise, goes to (1). During the procedure, $\varepsilon(t)$ decreases, on the other hand, $\sigma_{h1}^2(t)$ and $\sigma_{h2}^2(t)$ increases, to ensure that the change of w gradually decreases, that is, the self-organization process finishes.

3.9.2 High dimensional model

Initially $w(\mathbf{x}, \alpha)$ are given randomly from $i(\alpha)$ and these are modified as follows. For a given stimulus $i(\alpha)$, the winner cell $\mathbf{x}$ is determined by the following criteria.

$$\mathbf{x} = \arg\max_{\mathbf{x}} \sum_{\alpha} \mathbf{w}(\mathbf{x}, \alpha) i(\alpha).$$

Then the receptive fields of cell $\mathbf{y}$ are modified as follows.

$$\Delta w^{(t+1)}(\mathbf{y}, \alpha) = \varepsilon I(\mathbf{x} - \mathbf{y}) \left(i(\alpha) - w^{(t)}(\mathbf{y}, \alpha) \right).$$

This equation implies that a neibouring cell of the winner changes its receptive field to be similar to the winner's one. The neighborhood function $I(\mathbf{x} - \mathbf{y})$ is the following Gaussian.

$$I(\mathbf{x} - \mathbf{y}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{2\sigma^2} \right).$$

Stimuli are represented as a DOG function.

$$i(\alpha - \beta) = \exp\left(-\frac{||\alpha - \beta||^2}{2\sigma_1^2} \right) - k \exp\left(-\frac{||\alpha - \beta||^2}{2\sigma_2^2} \right),$$

where α stands for the stimulus center position. The ON-type stimulus i_{ON} and OFF-type stimulus i_{OFF} are defined as follows. $i_{\bullet}(\alpha - \beta) = [i(\alpha - \beta)]_+$ denotes the activity distribution of the central peak of the DOG, $i_{\circ}(\alpha - \beta) = [-i(\alpha - \beta)]_+$ the surround activity distribution corresponding the negative part of the DOG. Then we define i_{ON} and i_{OFF} as

$$i_{\mathrm{ON}} = (i_{\bullet}, i_{\circ}), \quad i_{\mathrm{OFF}} = (i_{\circ}, i_{\bullet})$$

At each step of the simulation, the center position α and the center type (ON or OFF) are given at random.

With this setting, Reisenhuber et al. analysed the parameter space in which orientation maps are obtained. They defined the energy function which computes the total distance between the receptive field of each cell and stimulus virtually given to the cell in the equiliblium status, under the assumption that there exists a equiliblium status and this algorithm converges the arrangement of receptive fields to the status. Then they found a very small parameter region for orientation map development. They discussed that since the region was too small, the former studies could not found a suitable parameter setting.

3.10 Elastic net models

The elastic net algorithm [21,31] is a variant of the dimension reduction framework.

At each iteration of the model, $w^{(t)}(\mathbf{x}, \alpha)$ are changed by an amount

$$\Delta w^{(t)}(\mathbf{x}, \alpha) = \alpha \sum_{\beta} o^{(t)}(\mathbf{x})(i(\beta) - w^{(t)}(\mathbf{x}, \alpha)) + \beta K \sum_{\mathbf{y} \in \text{nearest neibour of } \mathbf{x}} (w^{(t)}(\mathbf{y}, \alpha) - w^{(t)}(\mathbf{x}, \alpha)),$$

where K is a distance scaling parameter. The output $o(\mathbf{x})$ is computed as

$$o^{(t)}(\mathbf{x}) = \sum_{\alpha} \frac{\exp\left(-\frac{|i(\alpha) - w^{(t)}(\mathbf{x}, \alpha)|^2}{2K^2}\right)}{\sum_{\mathbf{y}} \exp\left(-\frac{|i(\alpha) - w^{(t)}(\mathbf{y}, \alpha)|^2}{2K^2}\right)}.$$

Durbin and Mitchson [21] applied this algorithm to the formation of orientation and retinotopic maps, and Goodhill and Willshaw [31] did to the formation of the retinotopic map. Later, Erwin et al. [23] combined these models and developed simultaneously the retinotopy, ocular dominance columns, and orientation columns.

Recently, Goodhill and Cimponeriu considered the application to the formation of ocular dominance and orientation columns, and they analyzed the behavior in the very early stage [30]. They investigated the effect of relative order of OR and OD development on overall map structure, then the result suggested that developmental order can be predicted from the final OR and OD periodicities.

3.11 Other models

In this section, we review some others which are not developmental models.

Feed-forward models suggest that cortical neurons obtain orientation selectivity from elongated patterns of converging LGN inputs [38]. Experiments qualitatively support this idea [10, 44], however, many simple cells subfield length-to-width ratios are quantitatively insufficient to account for the sharp orientation tuning [44]. Moreover, pure feed-forward models also cannot account for the loss of orientation selectivity observed under iontophoresis of bicuculline, a $GABA_A$ antagonist, which reduces inhibition over a localized population of cortical neurons [82, 101]. Furthermore, it has been shown that the layout of orientation maps is stable and independent of visual experience. Orientation maps are present in area 18 of binocularly deprived cats [12]. A reverse suture experiment has shown that two eyes without common visual experience develop similar orientation maps [28].

To address these problems, Somers, Nelson, and Sur proposed a recurrent neural network models [84]. Weak orientation selectivity by elongated feed-forward connections from LGN and recurrent cortical connections via inhibitory neurons are assumed. They found that the recurrent cortical excitation among cells preferring similar orientations, combined with iso-orientation inhibition from a broader range of orientations, integrates and amplifies a weak thalamic orientation bias. As a result, neurons showed a sharp orientation tuning and contrast invariance.

Besides, Goldberg, Shouval, and Cooper [29] focused on the anisotropic cortical connections and assumed that these connections are innate and arranged in which the orientations of connections become periodic. Then they trained the network according to the BCM rule, they reported that a sharp orientation tuning in neurons can be found even if the feed-forward connections show only a weak orientation selectivity.

Chapter 4

A role of constraint in self-organization

In this chapter we introduce a neural network model of self-organization. This model uses a variation of Hebb rule for updating its synaptic weights, and surely converges to the equilibrium status. The key point of the convergence is the update rule that constrains the total synaptic weight and this seems to make the model stable. We investigate the role of the constraint and show that it is the constraint that makes the model stable. For analyzing this setting, we propose a simple probabilistic game that models the neural network and the self-organization process. Then, we investigate the characteristics of this game, namely, the probability that the game becomes stable and the number of the steps it takes.

Results presented in this chapter partially appeared at [19].

4.1 Von der Malsburg's Model and Monopolist Game

Although von der Malsburg's original model consists of a set of neurons with lateral connections as reviewed in Section 3.2, we do not take the effects of lateral connections into account. Thus, we consider a neural network model consisting of one single postsynaptic neuron, so that we concentrate on the role of constraint for one single neuron.

In the following we give a brief overview of our simplified von der Malsburg's model with a new notation.

Neural Network Structure

We consider two-layer neural network. In particular, we discuss here the orientation selectivity for one neuron, and thus, we assume that there is only one output cell.

On the other hand, the input layer consists of 19 input cells that are (supposed to be) arranged in a hexagon like the ones in Fig.4.1. We use i for indicating the ith input cell, and IN for the set of all input cells.

Stimuli and Firing Rule

We use 9 stimuli with different orientations (Fig.4.2), which are given to the network randomly. Here • indicates an input cell that gets input 1, and ○ indicates an input cell that gets input 0.

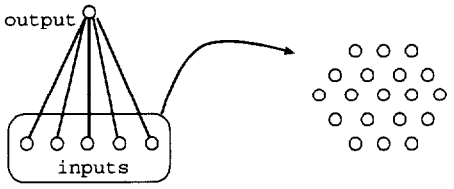

Figure 4.1: Neural network model.

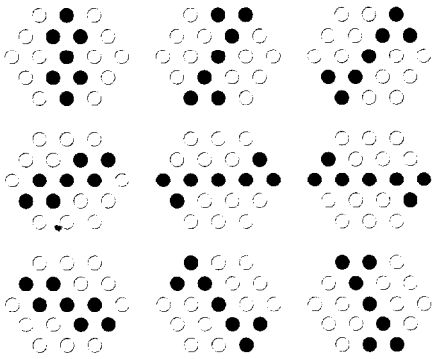

Figure 4.2: Nine stimuli.

We use a_i to denote input value (either 0 or 1) to the ith input cell. Then output value V is computed as $V = \mathrm{Th}_p\left(\sum_{i\in IN} w_i a_i\right)$, where w_i is the current synaptic strength between the output cell and the ith input cell. $\mathrm{Th}_p(x)$ is a threshold function that gives $x - p$ if $x > p$ and 0 otherwise, where p is given as a parameter.

Updating Rule

Initially, each weight w_i is set to some random number between 0 to some constant. Then weights are updated each time according to the following variation of Hebb's rule, which we call the *constrained Hebb rule* (of von der Malsburg).

$$w_i' = w_i + c_{\mathrm{inc}} a_i V, \quad \text{and} \quad w_i = w_i' \times \left(W_0 / \sum_{k\in I} w_k' \right).$$

Where c_{inc} (which is called a *growth rate*) and W_0 (*total weight bound*) are constants given as parameters. The first formula may be considered as the original Hebb's rule; on the other hand, the second one is introduced in order to keep the total weight within W_0. (In fact, it is kept as W_0.)

With this setting, von der Malsburg demonstrated that the selectivity is developed through computer simulations. Thus, it seems likely that some selection occurs even from uniformly random examples, and that the constraint of the von

der Malsburg's rule is a key for such a selection. In this paper we would like to study this feature of the constrained Hebb rule. For this, we further simplify von der Malsburg's computation model, and propose the following simple probabilistic game.

Monopolist Game
Basic Rule: Consider a finite number of players. Initially they are given the same amount of money. The game goes step by step, and at each step, one of the players wins with the same probability. The winner gets some amount of money, while the other loses some.
Details: A player who loses all his money is said become *bankrupt*. Once a player becomes bankrupt, he cannot get any amount of money, though he can still win with the same probability. (See below for the motivation.)
Goal: The game terminates if all but one player become bankrupt. If the survived player keeps enough money at that point, then he is called a *monopolist*. We call a situation where a monopolist appears *monopoly*.

Notations. We use n and n' to denote the number of players and that of remaining (not being bankrupt) players, and use i, $1 \leq i \leq n$, to denote players' indices. Throughout this paper, each player's wealth is simply called a *weight*, and let w_i denote the player i's current weight. Let I and W_0 respectively denote the initial weight of each player and the total amount of initial weights; that is, $W_0 = nI$.

The connection of this game with von der Malsburg's computation model is clear; each player's weight corresponds to total synaptic strength between the output cell and a set of input cells corresponding to one type of stimulus, and the emergence of a monopolist means that the network develops preference to one orientation. From this correspondence, it is natural to require that even a bankrupt player can win with the same probability $1/n$, which reflects the fact that the probability of a stimulus of each orientation appears is the same no matter how neural connections are organized.

An updating rule of players' weights corresponds to a rule of changing synaptic strength in the network. Here we can state updating rules in the following way. (In the following, let i_0 denote the player who wins at the current step.)

$$w_i = \begin{cases} w_i + f_{\text{inc}} - f_{\text{dec}}, & \text{if } i = i_0, \text{ and} \\ w_i - f_{\text{dec}}, & \text{otherwise.} \end{cases}$$

Here f_{inc} and f_{dec} are the amount of increment and decrement at each step respectively, and one type of monopolist game is specified by defining f_{inc} and f_{dec}. In the following, we assume that these values are determined from w_i, w_{i_0}, n,

and n'. From the relation to von der Malsburg's computation model, we require that both f_{inc} and f_{dec} are 0 if $w_i = 0$; that is, once a player loses all money, he stays forever in the 0 weight state. (In the following, we will omit stating this requirement explicitly.)

Now we consider the rule that corresponds to the constrained Hebb rule of von der Malsburg's rule. It is defined as follows with constant c_{inc}.

$$f_{\text{inc}} = c_{\text{inc}}, \text{ and } f_{\text{dec}} = c_{\text{inc}}/n'. \tag{4.1}$$

(Recall that n' is the number of currently remaining players.)

Note that with this rule, the total amount of wealth is kept constant. Thus, in this sense, it corresponds to von der Malsburg's rule, and we call it *constrained rule*. Note that we may also consider a similar rule such that f_{inc} is not constant but proportional to w_i. (Similarly, f_{dec} is also proportional to w_i.) This rule might be closer to the original von der Malsburg's rule. This difference is, however, not essential for discussing the probability of having a monopolist, i.e., for our discussion in Section 3. On the other hand, there is a significant difference in convergence speed; but roughly speaking, the difference disappears if we take the log of weight. Thus, we will discuss the above simpler rule.

4.2 Importance of Competition

Here we compare three different updating rules for monopolist game, and show that constraint is important to derive a monopolist. From this, we could infer that some sort of constraint, (or, competition in more general) is important in learning rules for selecting one among the others through random process.

In the following, we consider the following three updating rules: (1) constrained rule, (2) local rule, and (3) semi local rule. Below we define these rules (except (1) that has been defined in the previous section) and discuss the probability P_* that a monopolist emerges.

Constrained Rule

We show that under constrained rule, P_* is 1, that is, a monopolist emerges with probability 1.

A monopolist game in general expressed by an one-dimensional random walk. More precisely, for any i, we can express the player i's wealth w_i as the following random walk.

Note that the particle (i.e., the weight w_i) moves to the left (resp., to the right) with probability $1 - 1/n$ (resp., $1/n$). The left (resp., right) end of the interval means that the player i becomes bankrupt (resp., a monopolist). Thus, these two ends are absorbing walls.

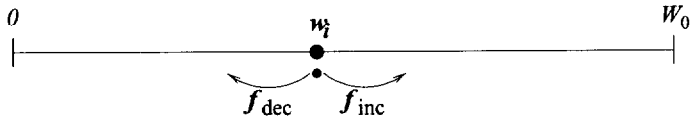

Figure 4.3: One-dimensional random walk.

In a monopolist game under constrained rule with $n = 2$, we have $f_{\mathrm{inc}} = c_{\mathrm{inc}}/2$ and $f_{\mathrm{dec}} = c_{\mathrm{inc}}/2$. Hence, the above random walk becomes standard one (see, e.g., [24]), and it is well-known that the particle in such a standard random walk goes to one of the absorbing walls in finite steps with probability 1. This proves that $P_* = 1$ when $n = 2$. Then by induction, we can prove $P_* = 1$ when $n > 2$; thus, we have the following theorem.

Theorem 4.2.1. Under constrained rule, a monopolist emerges (in finite steps) with probability 1.

Local Rule

In constrained rule, for computing f_{dec}, we need the number of remaining players; that is, weights cannot be updated locally. In general, in order to be competitive, an updating rule must not be local. Thus, to see the importance of competition, we consider here the following purely local updating rule.

$$f_{\mathrm{inc}} = c_{\mathrm{inc}}, \text{ and } f_{\mathrm{dec}} = c_{\mathrm{dec}}. \tag{4.2}$$

Notice that for this local rule (and the next semi local rule), the notion of monopolist is less clear than constrained rule, because the notion of "enough amount of money" is not clear. Here we simply consider it as $W_0/2$, a half of the total initial weight. That is, we regard a single surviver as a monopolist if his weight is more than $W_0/2$; hence, P_* is the probability that the game reaches to the state where $w_{i_1} \geq W_0/2$ for some i_1 and $w_i = 0$ for the other i.

We first discuss one feature of this updating rule. In the following, let us fix $c_{\mathrm{dec}} = 1$. Our computer experiments show that the probability to have a single surviver (in a reasonable amount of steps) drops rapidly when $c_{\mathrm{inc}} \geq n + 1$. The reason is clear from the following fact.

Theorem 4.2.2. Fix c_{dec} to be one, and consider one player's weight. For any t, it increases, by $t\left(\frac{c_{\mathrm{inc}}}{n} - 1\right)$ on average, after t steps.

Thus, if $c_{\mathrm{inc}} > n$, then it is quite likely that all players increase their weights, and thus no bankrupt appears in the game. On the other hand, if $c_{\mathrm{inc}} < n$, then

every player dies quickly, and hence, no monopolist occurs even though someone may become the last player. This means that the most crucial case is the case $c_{\mathrm{inc}} = n$. Next we discuss P_* for such a case.

Recall that P_* is the probability that, at some point in the game, all but one become bankrupt and that the survived player has weight $\geq W_0/2$. Since it is difficult to estimate P_* directly, we analyze the following probability P'_* instead of P_*: P'_* is the probability that at least one player's weight reaches to $W_0/2$ and no more two players have weight larger than sufficiently large value, say, kW_0 for some $k > 0$. Notice that if a monopolist emerges at some point, then clearly, someone needs to reach $W_0/2$ in the game. Furthermore, it is unlikely that two players reach to kW_0 and one of them become bankrupt afterwards. Thus, we may regard P'_* as an upper bound of P_*. For this P'_*, we have the following bound.

Theorem 4.2.3. For any W_1 and sufficiently large W_2, we have

$$P'_* < \Big(1 - \frac{I}{W_2 + n}\Big)^n + \frac{nI}{W_2}\Big(1 - \frac{I}{W_2 + n}\Big)^{n-1} - \Big(1 - \frac{I}{W_1}\Big)^n.$$

Proof. We first estimate, the probability $P_{I,W}$ that one player, say player 1, whose initial weight is I obtains weight W at some point in the game. By using results in random work [24], we can easily analyze this probability and obtain the following bounds.

Fact 1. For any I and W, we have $\dfrac{I}{W+n} \leq P_{I,W} \leq \dfrac{I}{W}$.

Then we consider P'_* with W_1 and sufficiently large W_2. Note first that

$$\begin{aligned}
&\Pr\{\text{less than two players' weight reach to } W_2\} \\
&\qquad = (1 - P_{I,W_2})^n + nP_{I,W_2}(1 - P_{I,W_2})^{n-1} \\
&\Pr\{\text{no player's wealth reaches to } W_1\} \\
&\qquad = (1 - P_{I,W_1})^n.
\end{aligned}$$

Then by definition of P'_*, we have

$$\begin{aligned}
P'_* &= (1 - P_{I,W_2})^n + nP_{I,W_2}(1 - P_{I,W_2})^{n-1} - (1 - P_{I,W_1})^n \\
&< \Big(1 - \frac{I}{W_2 + n}\Big)^n + \frac{nI}{W_2}\Big(1 - \frac{I}{W_2 + n}\Big)^{n-1} - \Big(1 - \frac{I}{W_1}\Big)^n.
\end{aligned}$$

■

For example, by taking W_1 and W_2 as $\frac{nI}{2}$ and knI respectively, we have

$$P'_* < e^{-I/(kI+1)} + e^{-I(n-1)/(kI+1)n}/k - e^{-2} \approx (1+1/k)e^{-1/k} - e^{-2},$$

which is less than 0.6 if $k = 1$. On the other hand, our computer experiments show that P_* is less than 0.5 for various sets of parameters.

Semi Local Rule

As a third updating rule, we consider somewhat mixture of the above two rules. It keeps a certain amount of locality, but it still has some constraint. This rule is defined as follows.

$$f_{\text{inc}} = \min(c_{\text{inc}}, W_0 - \sum_j w_j), \text{ and } f_{\text{dec}} = c_{\text{dec}}. \tag{4.3}$$

That is, we want to keep the total weight smaller than W_0, where W_0 is the total initial weight. Thus, a winner can gain c_{inc} (in net, $c_{\text{inc}} - c_{\text{dec}}$) if there is some room to increase its weight. In this case, only the winner needs to know the current total weight, or the amount of room to the limit W_0, and the other players can update its weight locally.

Our computer experiments show that the probability P_* that a monopolist emerges is fairly large if c_{inc} is large enough, say $c_{\text{inc}} \geq 2n$. On the other hand, P_* gets small when c_{inc} is small, which is explained in the same way as local rule. Although we have not been able to prove that P_* is large for sufficiently large c_{inc}, we can give some analytical result supporting it.

Here instead of analyzing P_*, we estimate (i) the average number of steps until all but one players become bankrupt, and (ii) the average number of steps until the total weight (which is initially W_0) becomes $W_0/2$. Let $T_{n\to 1}$ and $T_{W_0\to W_0/2}$ denote the former and the latter numbers respectively. We prove below that $T_{n\to 1}$ is smaller than $T_{W_0\to W_0/2}$ if W_0 is large enough. This means that it is likely that at the time when all but one players become bankrupt, the total weight, which is the same as the survivor's weight, is larger than $W_0/2$, that is, the surviver is a monopolist.

Theorem 4.2.4. Fix again c_{dec} to be one. If $I \geq (\ln 6)n(n-2)$ and $c_{\text{inc}} \geq 2n$, then we have $T_{W_0\to W_0/2} > T_{n\to 1}$.

To prove this, for a given $k \leq n$, we consider the situation that k players are left, and analyze their total weight and the weight of the poorest player. For this, we use random walks representing respectively, the total weight, *a random walk for the total*, and the weight of currently the poorest player, *a random walk for the poorest*. These random walks are expressed as follows.

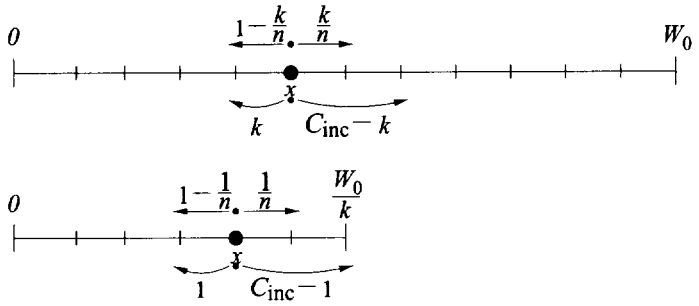

Figure 4.4: Random walk for the total (above) and for the poorest (below).

Since the total weight is at most W_0, the poorest player cannot have more than W_0/k. Also as soon as the poorest player obtains W_0/k, he cannot be the poorest, and he (or, his role) is replaced with some currently poorest player, whose weight is again less than W_0/k. In any case, if we only consider the weight of the poorest player, we can assume that the rightmost end is a reflecting wall. On the other hand, by definition of semi local rule, the rightmost end of a random walk for the total is a reflecting wall. Thus, in both random walks, the rightmost walls are reflecting walls; hence, the particles are eventually absorbed in the leftmost walls. Here we discuss the difference between the number of steps until the particles are absorbed.

First we prove the following two lemmas.

Lemma 4.2.5. Consider the situation in the game that k players are left, and let $\mathrm{TP}(k)$ be the average number of steps that some player becomes bankrupt from this situation. Then we have $\mathrm{TP}(k) \leq ne^{\frac{W_0}{nk}}$.

Proof. Consider a random walk for the poorest. As explained above, we may assume that the random walk has a reflecting wall at W_0/k and an absorbing wall at 0. Here for showing an upper bound, we assume, as the worst case, that the initial weight of the poorest is W_0/k; that is, the random walk starts off at W_0/k. Then $\mathrm{TP}(k)$ is bounded by the average number of steps until the particle reaches to the absorbing wall at 0.

Let t_x be the random variable denoting the number of steps starting off at x and arriving at $x-1$ for the first time. Then we can evaluate t_x by

$$t_x = \begin{cases} 1, & \text{with probability } 1-1/n, \text{ and} \\ 1+t_{x+c_{\mathrm{inc}}-1}+\ldots+t_{x+1}+t_x, & \text{otherwise.} \end{cases}$$

Define e_x to be the expectation of t_x. Then we have

$$e_x = \frac{n}{n-1} + \frac{1}{n-1}\left(e_{x+c_{\mathrm{inc}}-1} + \ldots + e_{x+1}\right).$$

Then by induction on x, we can show that $e_{W_0/k-x} \leq \left(1 + 1/(n-1)\right)^x$. Thus, we have

$$\mathrm{TP}(k) \leq \sum_{x=0}^{W_0/k-1} e_{W_0/k-x} \leq (n-1)\left(1 + \frac{1}{n-1}\right)^{\frac{W_0}{k}} \approx (n-1)e^{\frac{W_0}{nk}}.$$

■

Lemma 4.2.6. Let $\mathrm{TT}(k, x, y)$ be the average number of steps until the total weight becomes y from x under the condition that none of remaining k players become bankrupt. For any x and y, $x > y$, we have

$$\mathrm{TT}(2, x, y) > \frac{n-2}{2}\left(\left(1 + \frac{2}{n-2}\right)^{\frac{W_0-y}{2}} - \left(1 + \frac{2}{n-2}\right)^{\frac{W_0-x}{2}}\right)(1 - e^{-2}).$$

Proof. Consider a random walk for the total (see Fig.4.4). By dividing all parameters by $k = 2$, we can modify it to essentially the same random walks as Fig.4.5.

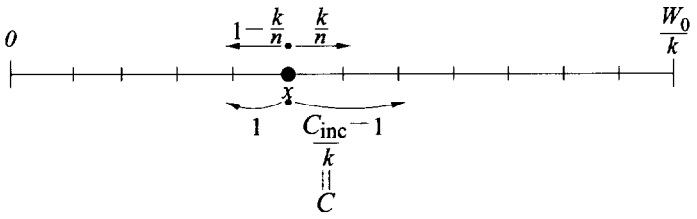

Figure 4.5: A modification of a random walk for the total.

Note that this random walk is quite similar to a random walk for the poorest (see Fig.4.4 below). Thus, for e_x, the average number of steps starting off at x and arriving at $x-1$ for the first time, a similar argument derives that the following relation (here $c = c_{\mathrm{inc}}/2$).

$$e_x > 1 + \frac{2}{n-2}\left(e_{x+c-1} + \ldots + e_{x+1}\right).$$

Then it is not so hard to see that

$$\begin{aligned} e_{W_0/2-x} &\geq \left(1+\frac{2}{n-2}\right)^{x}-\left(1+\frac{2}{n-2}\right)^{x-c+1} \\ &= \left(1+\frac{2}{n-2}\right)^{x}\left(1-\left(1+\frac{2}{n-2}\right)^{-c+1}\right). \end{aligned}$$

Now by using our assumption that $c = c_{\mathrm{inc}}/2 \geq n$, we can derive the following bound.

$$\begin{aligned} \mathrm{TT}(2,x,y) &\geq \sum_{x=(W_0-x)/2}^{(W_0-y)/2-1}\left(1+\frac{2}{n-2}\right)^{x}\left(1-\left(1+\frac{2}{n-2}\right)^{-c+1}\right) \\ &\geq \frac{n-2}{2}\left(\left(1+\frac{2}{n-2}\right)^{\frac{W_0-y}{2}}-\left(1+\frac{2}{n-2}\right)^{\frac{W_0-x}{2}}\right)(1-e^{-2}). \end{aligned}$$

■

Now we are ready to prove Theorem 4.2.4.

Proof of Theorem 4.2.4. We first give an upper bound to $T_{n\to 1}$, the average number of steps until all but one players become bankrupt. For any $k \leq n$, consider the situation in the game that k players are left.

By Lemma 4.2.5, we have

$$\begin{aligned} T_{n\to 1} &\leq \sum_{k=1}^{n}\mathrm{TP}(k) \\ &\leq n(e^{W_0/2n}+e^{W_0/3n}+\ldots+e^{W_0/(n^2)}) \\ &\leq ne^{W_0/2n}+n(n-2)e^{W_0/3n} \\ &\leq 2ne^{W_0/2}, \end{aligned}$$

where the last inequality holds from our assumption $W_0 \geq n(n-2)\ln 6$.

Next consider $T_{W_0\to W_0/2}$, the average number of steps needed until the total weight becomes $W_0/2$ from W_0. Note first that for any k and k' such that $k < k'$, we have $\mathrm{TT}(k,x,y) \leq \mathrm{TT}(k',x,y)$. Then, for any $x_0,\ldots,x_{n-3}$, we have

$$\begin{aligned} \mathrm{TT}(n,W_0,x_0)+\mathrm{TT}(n-1,x_0,x_1)+\cdots+\mathrm{TT}(2,x_{n-3},W_0/2) \\ \geq \mathrm{TT}(2,W_0,W_0/2). \end{aligned}$$

Hence, $T_{W_0\to W_0/2} \geq \mathrm{TT}(2,W_0,W_0/2)$. On the other hand, $\mathrm{TT}(2,W_0,W_0/2)$ is bounded in the following way by using Lemma 4.2.6 and the assumption that $c_{\mathrm{inc}} \geq 2n$.

$$\begin{aligned}\mathrm{TT}(2, W_0, W_0/2) &> \frac{n-2}{2}\left(\left(1+\frac{2}{n-2}\right)^{\frac{W_0}{2}}-1\right)\left(1-e^{-2}\right) \\ &\geq \frac{n-2}{2}\cdot\left(e^{\frac{W_0}{2(n-2)}}\left(1-e^{-2}\right)\right) \geq \frac{n-2}{3}\cdot e^{W_0/2(n-2)}.\end{aligned}$$

Comparing both bounds, we have $T_{W_0 \to W_0/2} \geq T_{n \to 1}$ if $W_0 \geq n(n-2)\ln 6$. ■

4.3 Efficiency Analysis

In this section we discuss how fast a monopolist emerges in games with constrained rule. We estimate an upper bound on the average number of steps needed for monopoly to emerge, and we give some justification (not a rigorous proof) supporting that it is $O(n^2(In/c_{\mathrm{inc}})^2)$.

We start with some definitions and notations that are used through the section. Here we modify our monopolist game and define a variant of monopolist game. Let game_0 denote the original monopolist game. We will denote by game_1 a variant of game_0 in which no bankrupt player can win. Thus, in game_1, the winning probability of remaining players is $1/n'$ instead of $1/n$. As we will see game_1 is useful for induction and it is easier to analyze.

These two game types are defined on different probability spaces. Let us define them more precisely. For all two game types, (the execution of) a game is specified by a *game sequence*, i.e., a string from $\{1, \ldots, n\}^*$ that defines a history of winners. (Precisely speaking, we also need to consider infinite strings; but as we see below, we may ignore infinite strings.) We say that a game sequence x *kills* a player i if w_i becomes 0 (or, negative) in the game following x just after the end of x, and we say that x *derives a monopolist* if the second last player is killed and monopoly emerges just after x. We say that a game sequence x is *valid* (resp., *strongly valid*) if it derives a monopolist and no prefix of it derives a monopolist (resp., x contains no indices of previously killed players). Note that the meaning of these notions may vary depending on game types. Now for any n, let X_n (resp., Y_n) be the set of game sequences for n player games that are strongly valid w.r.t. game_0 (resp., valid w.r.t. game_1). For each x in X_n, its probability $\Pr\{x\}$ is $n^{-|x|}$. On the other hand, the probability $\widetilde{\Pr}\{y\}$ of $y \in Y_n$ depends on the number of remaining players, and it is rather complicated. (We omit specifying $\widetilde{\Pr}\{y\}$ because it is not important for our discussion.) Note that X_n and Y_n are all prefix free. Also it is not hard to show that $\Pr\{X_n\}$ and $\widetilde{\Pr}\{Y_n\}$ are one. (For example, $\Pr\{X_n\} = 1$ follows from Theorem 4.2.1.) Therefore, we may regard X_n and Y_n as a probability space of the corresponding game, and we do not have to worry about infinite strings.

We denote by $T(n, I_1, \ldots, I_n)$ (resp., $T_1(n, I_1, \ldots, I_n)$) the number of steps needed until monopoly emerges in game_0 (resp, game_1) with n players and initial

weight $I_1, \ldots, I_n$. When all the weights are equal, we use the simpler notation $T(n, I)$. Our goal is to get some upper bound on $\mathrm{E}[T(n, I)]$. But instead, we will analyze an upper bound on $\mathrm{E}[T_1(n, I)]$, which gives us an upper bound on $\mathrm{E}[T(n, I)]$, as the following lemma guarantees.

Lemma 4.3.1. There exists c_1 such that for any sufficiently large n and any I, we have $\mathrm{E}[T(n, I)] \leq c_1 n \mathrm{E}[T_1(n, I)]$.

Now we analyze the convergence speed of game$_1$. For our analysis, we split a game execution into stages where each stage is a part of the game until some amount of players become bankrupt. More specifically, we denote by $t_1(n, I_1, \ldots, I_n)$ the number of steps needed in a game with n players and initial weights $I_1, \ldots, I_n$ until at least 1 player becomes bankrupt. The following lemma relates the two terms $T_1(n, I_1, \ldots, I_n)$ and $t_1(n, I_1, \ldots, I_n)$.

Lemma 4.3.2. For any n and $I_1, \ldots, I_n$, there exists a constant c_2, $c_2 \geq 1$, and weights $I'_1, \ldots, I'_{n-c_2}$ such that the following inequality holds.

$$\mathrm{E}[T_1(n, I_1, \ldots, I_n)] \leq \mathrm{E}[t_1(n, I_1, \ldots, I_n)] + \mathrm{E}[T_1(n - c_2, I'_1, \ldots, I'_{n-c_2})].$$

Proof. Let $Y \subseteq Y_n$ be the set of all valid game sequences y such that the number of players becomes strictly smaller than n for the first time just after y. By definition of $\mathrm{E}[T_1(n, I_1, \ldots, I_n)]$, we have the following equality.

$$\mathrm{E}[T_1(n, I_1, \ldots, I_n)] = \sum_{x \in Y_n} \widetilde{\Pr}\{x\} \cdot |x| = \sum_{\substack{x \in Y_n, y \in Y \\ x = yz}} \widetilde{\Pr}\{yz\}(|y| + |z|).$$

Notice here that we can split $\widetilde{\Pr}\{yz\}$ in two factors, $\widetilde{\Pr}\{y\}$ and $\widetilde{\Pr}_y\{z\}$, where $\widetilde{\Pr}_y\{z\}$ determines the probability of z after the game follows y. Also note that the set Y_y of strongly valid z depends on y. Thus, we can rewrite the above expression as

$$\begin{aligned}
&\mathrm{E}[T_1(n, I_1, \ldots, I_n)] \\
&= \sum_{y \in Y} \sum_{z \in Y_z} \widetilde{\Pr}\{y\}\widetilde{\Pr}_y\{z\} \cdot |y| + \sum_{y \in Y} \sum_{z \in Y_z} \widetilde{\Pr}\{y\}\widetilde{\Pr}_y\{z\} \cdot |z| \\
&= \left(\sum_{y \in Y} \widetilde{\Pr}\{y\}|y|\right) \sum_{z \in Y_z} \widetilde{\Pr}_y\{z\} + \sum_{y \in Y} \widetilde{\Pr}\{y\} \left(\sum_{z \in Y_z} \widetilde{\Pr}_y\{z\} \cdot |z|\right) \\
&= \mathrm{E}[t_1(n, I_1, ..., I_n)] \sum_{z \in Y_z} \widetilde{\Pr}_y\{z\} + \sum_{y \in Y} \widetilde{\Pr}\{y\} \mathrm{E}[T_1(n_y, I_{i_1(y)}, \ldots, I_{i_{n_y}(y)})] \\
&\leq \mathrm{E}[t_1(n, I_1, \ldots, I_n)] + \mathrm{E}[T_1(n - c', I'_1, \ldots, I'_{n-c'})].
\end{aligned}$$

where the values of n_y and $I_{i_1(y)}, \ldots, I_{i_{n_y}(y)}$ are determined by the result of the game following y. On the other hand, c' and I'_i are chosen so that the value of $\mathrm{E}[T_1(n_y, I_{i_1(y)}, \ldots, I_{i_{n_y}}(y))]$ is maximized. These values always exist since even if there is an infinite number of game sequences y that appear on the summation, there is only a finite number of possible values for n_y (since n_y must be between 1 and $n-1$) and $I_{i_j(y)}$ (since $\sum_j I_{i_j(y)} = In$). ■

By this lemma we can use induction for bounding the expected value of T_1. Recall that when analyzing $t_1(n, I_1, \ldots, I_n)$, by the way it is defined, no player becomes bankrupt, and thus, the amount of decrement is fixed to c_{inc}/n. Thus, game_1 until at least one player becomes bankrupt is regarded as a n-dimensional random walk, which is much easier to be analyzed. In fact, we can use the following lemma.

Lemma 4.3.3. Let X be a random variable that is 1 with probability $1/n$ and 0 with probability $1 - 1/n$, and let $S = X_1 + \ldots + X_t$, the sum of the outcomes of t random trials of X. Then, for some constant $\alpha > 0$, the following holds for any t and n.

$$\Pr\left\{S \leq \frac{t}{n} - \alpha\sqrt{\frac{t}{n}}\right\} > 1/3.$$

Proof. We estimate the probability that the statement of the lemma is false and show that it is less than 2/3. That is, we upper bound the following probability.

$$\Pr\left\{S > \frac{t}{n} - \alpha\sqrt{\frac{t}{n}}\right\} = \Pr\left\{S > \frac{t}{n}\right\} + \Pr\left\{\frac{t}{n} \geq S > \frac{t}{n} - \alpha\sqrt{\frac{t}{n}}\right\}.$$

The first term in the sum is bounded by 1/2; see, e.g., [43]. Let s be the smallest integer such that $s > t/n - \alpha\sqrt{t/n}$. We calculate, by using Stirling's approximation, the second term of the above sum as follows.

$$\begin{aligned}
\Pr\left\{\frac{t}{n} \geq S > \frac{t}{n} - \alpha\sqrt{\frac{t}{n}}\right\} &= \sum_{i=s}^{t/n} \binom{t}{i} \left(\frac{1}{n}\right)^i \left(1 - \frac{1}{n}\right)^{t-i} = \sum_{i=s}^{t/n} \binom{t}{i} \frac{(n-1)^{t-i}}{n^t} \\
&\leq \sum_{i=s}^{t/n} \sqrt{\frac{t}{2\pi i(t-i)}} \left(\frac{t}{t-i}\right)^t \left(\frac{t-i}{i}\right)^i \frac{(n-1)^{t-i}}{n^t} \\
&\leq \sum_{i=s}^{t/n} \sqrt{\frac{t}{2\pi i(t-i)}} \left(\frac{t(n-1)}{n(t-i)}\right)^t \left(\frac{t-i}{i(n-1)}\right)^i.
\end{aligned}$$

Also routine calculations show that $\left(\frac{t(n-1)}{n(t-i)}\right)^t \left(\frac{t-i}{i(n-1)}\right)^i$ is always less than 1 for $s \leq i \leq t/n$ and that this factor is maximized when $i = t/n$. From this by simple calculation, we obatain the desired bound with $\alpha = \sqrt{\pi/12}$. ■

Now we are now ready to make the following claim[1].

Claim.

$$\mathrm{E}[T(n, I)] = O\left(n^2 \left(\frac{In}{c_{\mathrm{inc}}}\right)^2\right).$$

Justification. We start with estimating $\mathrm{E}[t_1(n, I_1, \ldots, I_n)]$ by using the above lemma. For a given t and for any i, Let t_i be the number of times that player i wins within t steps. Then w_i, the weight of player i, in game_1 is expressed as follows.

$$w_i = I_i + c_{\mathrm{inc}} t_i - c_{\mathrm{dec}} t \leq I_i + c_{\mathrm{inc}} \left(t_i - \frac{t}{n}\right).$$

(For simplifying our notation, we use c to denote c_{inc} in the following.)

Moreover, since game_1 until at least one player becomes bankrupt is regarded as a n-dimensional random walk, we can use Lemma 4.3.3 to show that the following event happens with probability bigger than 1/3.

$$w_i = I_i + c\left(t_i - \frac{t}{n}\right) \leq I_i + c\left(\frac{t}{n} - \alpha\sqrt{\frac{t}{n}} - \frac{t}{n}\right) = I_i - c\alpha\sqrt{\frac{t}{n}}.$$

Therefore, with probability more than 1/3, the weight of player i becomes zero or negative if $c\alpha\sqrt{t/n} \geq I_i$, that is, $t \geq (I_i/c\alpha)^2 n$. Now sort players by their initial weights, and define P to be the set of the first (i.e., the smallest) $n/2$ players. Since the total weight is W_0 (at any step), all players in P have weight at most $2W_0/n$ and therefore,

$$\Pr\{\, w_i \leq 0 \text{ in } t_0 = n\left(\frac{2W_0}{nc\alpha}\right)^2 \text{ steps} \mid i \in P \,\} > \frac{1}{3}.$$

[1]We do not have a rigorous proof for this result, and for this reason we stated it as a claim.

Moreover, *if* we can assume that each player in P become bankrupt independently, we also have the following probability:

$$\Pr\{\text{ There exists } i \in P, \text{ such that } w_i < 0 \text{ in } t_0 \text{ steps}\} > 1 - \left(\frac{2}{3}\right)^{n/2}.$$

From this observation, it is reasonable to bound $\mathrm{E}[t_2(n, I_1, \ldots, I_n)]$ by $c_3 t_0$ for some constant c_3 since for most of the valid game sequences (a $1 - (2/3)^{n/2}$ fraction of them) this bound holds.

Now combining the above lemmas and the obtained bound, we have

$$\begin{aligned}
\mathrm{E}[T_1(n, I)] &\leq \mathrm{E}[t_1(n, I_1, \ldots, I_n)] + \mathrm{E}[T_1(n - c_2, I'_1, \ldots, I'_{n-c_2})] \\
&\leq c_3 n \left(\frac{2W_0}{nc\alpha}\right)^2 + \mathrm{E}[T_1(n - c_2, I'_1, \ldots, I'_{n-c_2})] \\
&\leq c_3 n \left(\frac{2W_0}{nc\alpha}\right)^2 + c_3(n-1)\left(\frac{2W_0}{(n-1)c\alpha}\right)^2 \\
&\quad + \mathrm{E}[T_1(n - c'_2, I'_1, \ldots, I'_{n-c'_2})] \cdots \\
&\leq c_3 n \left(\frac{W_0}{c\alpha}\right)^2 \leq c_4 n \left(\frac{In}{c}\right)^2, \quad \text{for some constant } c_4.
\end{aligned}$$

From this and Lemma 4.3.1, we obtain the desired bound. ■

4.4 Conclusions

In thic chapter, we considered a stochastic process of the self-organization in von der Malsburg's model. In particular, we focused on a constraint for the synaptic weight modification and studied its role for the stability of the self-organization. We proposed a probabilistic game called the monopolist game which represents the stochastic process of the weight modification. The monopolist game consisted of players each corresponding to an orientation and a player was initially given an amount of money corresponding the strength of the synaptic connection. The amount of money changed by following a game rule, and a player lost all money was ruled out. The goal of this game was to survive with much money until the other players were ruled out, which corresponded to that a preferred orientation was determined with a sufficient connection strength. Thus the dynamics of Malsburg's model could be expressed in terms of random walks. Analyses of the random walks stated that a Malsburg-type normalization surely enabled that the monopolist emerged. Besides, we also calculated the average number of steps needed until the monopolist emerged.

Chapter 5

Emergence of orientation selectivity for one single neuron

In this chapter, we demonstrate a mathematical analysis of the emergence of orientation selectivity for one single neuron, in other words, how one single neuron develops its oriented receptive field (RF). We consider Linsker's neural network model reviewed in Section 3.5 and mathematically analyze its dynamics, in particular, the dynamics between layer F and layer G, in which a neuron develops oriented RFs. We concentrate on the Mexican-hat correlation function and show that this function determines the spatial frequency of RFs. We also focus on the role of the arbor function. Two types of arbor functions, a Gaussian type and a step type, are considered, and it is shown that the Gaussian arbor function develops center-surround RFs while the step arbor function enables the development of multi-lobed RFs in addition to center-surround receptive fields. In summary, our mathematical analysis clarifies the roles of the arbor function as well as the correlation function. Furthermore we show how the orientation of multi-lobed RFs is determined and finally we discuss the roles of parameters k_1, k_2 in Linsker's model.

Results presented in this chapter partially appeared at [112].

5.1 Preliminaries

5.1.1 Model to study

We consider the dynamics of Eqn. (3.2) with some simplifications. First we consider the continuous form of Eqn. (3.2). Second, we assume that the absolute values of an upper bound and a lower bound are large enough and the dynamics of the self-organization is determined before synaptic weights reach these bounds. Therefore we omit these bounds and consider the linear regime only.

In summary, the following equation is derived, which is Eqn. (7) in [54].

$$\dot{w}(\alpha) = k_1 + \int_{-\infty}^{\infty} (C(\alpha - \beta) + k_2) A(\beta) w(\beta) d\beta, \tag{5.1}$$

$w(\alpha)$ is the synaptic weight of the cell in α and $\dot{w}(\alpha)$ is the growth rate. $A(\alpha)$ is the arbor function and we make use of the following functions.

$$A(\alpha) = \begin{cases} 1 & |x| \leq R, \\ 0 & \text{otherwise.} \end{cases}, \quad \text{and} \quad A(\alpha) = \exp\left(-|\alpha|^2/2\sigma_A^2\right), \tag{5.2}$$

where R and $\sqrt{2}\sigma_A$ are positive constants that determine the radius of effective synaptic density, which we call *arbor radius*. We denote the arbor radius by R_A and we call these functions *Step* and *Gaussian*, respectively. On the other hand, since Q_{ij}^F can be expressed as a Mexican-hat function, we make use of the following DOG function as our correlation function.

$$C(\alpha) = K_1 \exp\left(-|\alpha|^2/2\sigma_1^2\right) - K_2 \exp\left(-|\alpha|^2/2\sigma_2^2\right), \tag{5.3}$$

where $\sigma_1 < \sigma_2$. DOG functions of this type are used to express Mexican-hat functions [60, 64]. Initial synaptic weights $w^{(0)}(\alpha)$ are assigned uniformly at random.

Besides, since we want to consider only $w(\alpha)$ within the arbor, we consider the dynamics of $A(\alpha)w(\alpha)$. Let $w_*(\alpha)$ be synaptic weights modified by the following rule.

$$\dot{w}_*(\alpha) \stackrel{\text{def}}{=} A(\alpha) \left(k_1 + \int_{-\infty}^{\infty} (C(\alpha - \beta) + k_2) w_*(\beta) d\beta \right). \tag{5.4}$$

If we assign the initial weights $w_*^{(0)}(\alpha)$ as

$$w_*^{(0)}(\alpha) = A(\alpha) w^{(0)}(\alpha),$$

then it follows that

$$w_*(\alpha) = A(\alpha) w(\alpha). \tag{5.5}$$

Thus the dynamics of $w_*(\alpha)$ is essentially identical to that of $A(\alpha)w(\alpha)$. Therefore in the following sections, we consider the dynamics of $w_*(\alpha)$, namely Eqn. (5.4), instead of Eqn. (5.1). We denote $w_*(\alpha)$ as simply $w(\alpha)$ by omitting the $*$ symbol.

What follows proves the equivalence in Eqn. (5.5). We consider the difference versions of Eqn. (5.1) and Eqn. (5.4). Let $w^{(t)}(\alpha)$ and $\Delta w^{(t)}(\alpha)$ be the weight and the growth at the t th step respectively and consider that $\Delta w^{(t)}(\alpha)$ is given by the following difference version of Eqn. (5.1).

$$\Delta w^{(t)}(\alpha) = k_1 + \int_{-\infty}^{\infty} (C(\alpha - \beta) + k_2) A(\beta) w^{(t)}(\beta) d\beta.$$

In contrast to this, the difference version of Eqn. (5.4) is as

$$\Delta w_*^{(t)}(\alpha) = A(\alpha)\left(k_1 + \int_{-\infty}^{\infty}(C(\alpha-\beta)+k_2)\, w_*^{(t)}(\beta)d\beta\right).$$

We assign the initial weights $w_*^{(0)}(\alpha)$ as

$$w_*^{(0)}(\alpha) = A(\alpha)w^{(0)}(\alpha),$$

then inductively we prove for any t

$$w_*^{(t)}(\alpha) = A(\alpha)w^{(t)}(\alpha).$$

1. At the initial step,

$$w_*^{(0)}(\alpha) = A(\alpha)w^{(0)}(\alpha).$$

2. Assume that for some t it follows that

$$w_*^{(t)}(\alpha) = A(\alpha)w^{(t)}(\alpha),$$

then we consider the case of the $t+1$ th step.

$$\begin{aligned}
w_*^{(t+1)}(\alpha) &= w_*^{(t)}(\alpha) + \Delta w_*^{(t)}(\alpha) \\
&= A(\alpha)w^{(t)}(\alpha) \\
&\quad + A(\alpha)\left(k_1 + \int_{-\infty}^{\infty}(C(\alpha-\beta)+k_2)\, w_*^{(t)}(\beta)d\beta\right) \\
&= A(\alpha)w^{(t)}(\alpha) \\
&\quad + A(\alpha)\left(k_1 + \int_{-\infty}^{\infty}(C(\alpha-\beta)+k_2)\, A(\beta)w^{(t)}(\beta)d\beta\right) \\
&= A(\alpha)\cdot \\
&\quad \left(w^{(t)}(\alpha) + k_1 + \int_{-\infty}^{\infty}(C(\alpha-\beta)+k_2)\, A(\beta)w^{(t)}(\beta)d\beta\right) \\
&= A(\alpha)w^{(t+1)}(\alpha).
\end{aligned}$$

■

5.1.2 Characterization of a receptive field

We characterize the shape of a receptive field by its spatial frequency and the number of angular nodes. Let $c_{s,k}(\alpha)$ be the receptive field that has the spatial frequency s and k angular nodes. $c_{s,k}(\alpha)$ can be computed by the following integral.

$$\begin{aligned}
c_{s,k}(\alpha) &\stackrel{\text{def}}{=} \int_0^{2\pi} \exp(ik\theta)\exp(is_\theta\alpha)d\theta, \\
s_\theta &\stackrel{\text{def}}{=} (s\cos\theta, s\sin\theta).
\end{aligned}$$

Figure 5.1 shows 16 examples of $c_{s,k}(\alpha)$ for $s = 0, \pi/R_A, 2\pi/R_A, 3\pi/R_A$, and $k = 0, 1, 2, 3$ (Note that R_A is the arbor radius defined in the previous subsection. Thus, $c_{a\pi/R_A,k}(\alpha)$ denotes the receptive field that has a radial nodes and k angular nodes). We make use of the symbol $*$ as a wild card, for example, $c_{s,*}(\alpha)$ indicates one of $c_{s,0}(\alpha), c_{s,1}(\alpha), c_{s,2}(\alpha), \cdots$. A similar notation has been introduced

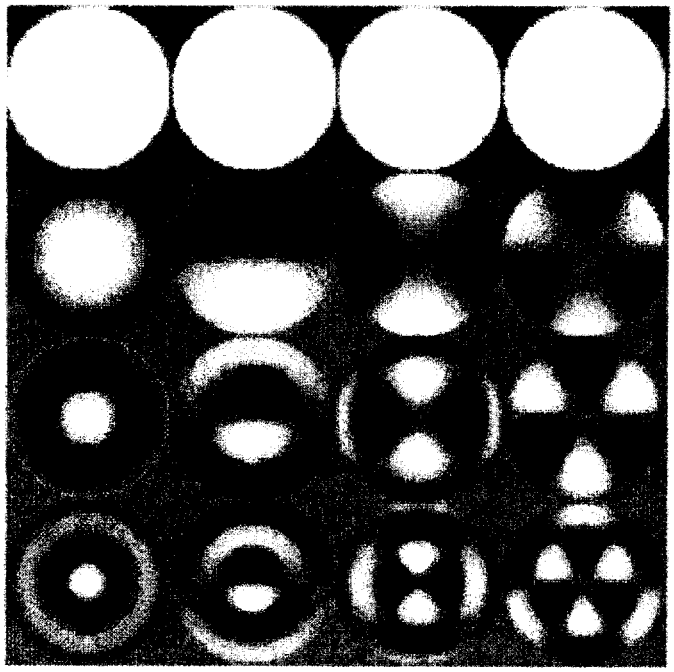

Figure 5.1: 16 examples of $c_{s,k}(\alpha)$ for $s = 0, \pi/R_A, 2\pi/R_A, 3\pi/R_A$, and $k = 0, 1, 2, 3$. Each row corresponds to a spatial frequency (i.e., s), and each column corresponds to a number of angular nodes (i.e., k). For example, the top left example is $c_{0,0}(\alpha)$ and the bottom right one is $c_{3\pi/R_A,3}(\alpha)$.

by MacKay & Miller [54] and it is an alternative to our notation. For example, $c_{0,0}(\alpha)$ and $c_{\pi/R_A,1}(\alpha)$ respectively correspond to '1s' and '2p' in their notation. $c_{*,0}(\alpha)$ corresponds to 's-mode'.

5.2 Mathematical analysis

In our analysis we consider the case that k_1 and k_2 are 0; later we will discuss the roles of these parameters independently.

We consider expressing the shape of a developed receptive field as the sum of its Fourier components with respect to the spatial frequency and the number of

angular nodes as shown in Fig.5.1, that is, we assume that $w(\alpha)$ at the t-th step can be written as the following summation.

$$w(\alpha) = \int_0^\infty \sum_{k=0}^{\infty} a_{s,k} c_{s,k}(\alpha) \exp(g(s,k)t) ds, \tag{5.6}$$

where $a_{s,k}$ is a coefficient, $c_{s,k}(\alpha)$ is a component defined in the previous section, and $g(s,k)$ is the function which defines the "growth rate" of $c_{s,k}(\alpha)$. Since for any pair of (s,k) and (s',k') the difference between $\exp(g(s,k)t)$ and $\exp(g(s',k')t)$ becomes exponential as t increases, we can consider $w(\alpha)$ will be dominated by $c_{s_*,k_*}(\alpha)$ such that $(s_*,k_*) = \arg\max_{s,k}(f(s,k))$. Thus, in the following, we attempt to obtain the solution of Eqn. (5.4) in the form of Eqn. (5.6). Table of integrals should be referred for calculations of integrals which result the Bessel functions, for example, [32].

The FT of Eqn.(5.4) is

$$\dot{\widehat{w}}(\mathbf{n}) = \int_{-\infty}^{\infty} \widehat{A}(\mathbf{n}-\mathbf{m})\widehat{C}(\mathbf{m})\widehat{w}(\mathbf{m}) d\mathbf{m},$$

where $\widehat{C}(\mathbf{n})$, the FT of $C(\alpha)$, is the following DOG function.

$$\widehat{C}(\mathbf{n}) = 2\pi\sigma_1^2 K_1 \exp\left(-\sigma_1^2|\mathbf{n}|^2/2\right) - 2\pi\sigma_2^2 K_2 \exp\left(-\sigma_2^2|\mathbf{n}|^2/2\right). \tag{5.7}$$

On the other hand, $\widehat{A}(\mathbf{n})$, the FT of $A(\alpha)$, is one of the following functions.

When A is Gaussian:

$$\widehat{A}(\mathbf{n}) = 2\pi\sigma_A^2 \exp\left(-\sigma_A^2|\mathbf{n}|^2/2\right)$$

When A is Step:

$$\widehat{A}(\mathbf{n}) = 2\pi R J_1(R|\mathbf{n}|)/|\mathbf{n}|,$$

where $J_1(x)$ is the Bessel function of the first kind of order 1.

As evident from Fig.5.2, the shape of $\widehat{C}(\mathbf{n})$ is either uni-modal or bi-modal depending on the choice of parameters. In the following analysis, we consider only the bi-modal case, because the uni-modal case is essentially the same as the case in which the correlation function is Gaussian, and this case has been extensively studied by MacKay & Miller [54].

As shown in Fig.5.2 left, $\widehat{C}(\mathbf{n})$ of the bi-modal case can be regarded as follows.

$$\widehat{C}(\mathbf{n}) = \begin{cases} C_*, & \sigma - \varepsilon \le |\mathbf{n}| \le \sigma + \varepsilon, \\ 0, & \text{otherwise}, \end{cases}$$

where σ is a constant determined by $K_1, K_2, \sigma_1, \sigma_2$, and $\varepsilon > 0$ is a sufficiently small constant.

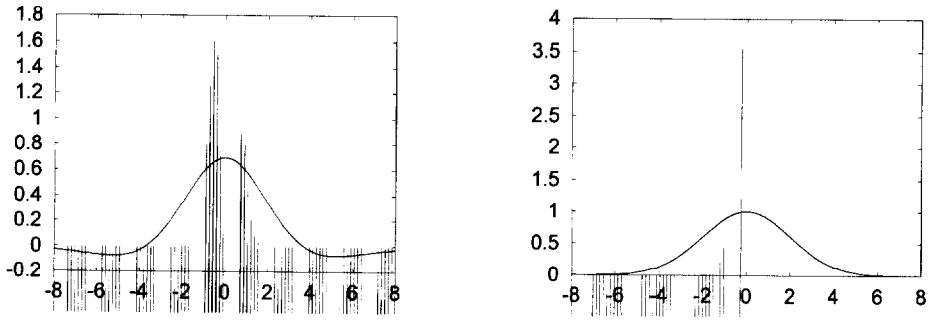

Figure 5.2: A correlation function (real line) and its FT divided by 2π (dashed line). (Left: under the following parameter settings. $\sigma_1^2 = 4.0$, $\sigma_2^2 = 13.0$,$K_1 = 1.0$, $K_2 = (\sigma_1/\sigma_2)^2$.) (Right: the parameters are the same as the left ones except for $K_2 = 0.001$.)

From this approximation, it follows that

$$\dot{\widehat{w}}\,(\mathbf{n}) = \int_{|\mathbf{m}|\approx\sigma} \widehat{A}(\mathbf{n}-\mathbf{m})\widehat{w}(\mathbf{m})d\mathbf{m}. \tag{5.8}$$

Here by $|\mathbf{m}| \approx \sigma$, we mean the integral domain $\sigma - \varepsilon \leq |\mathbf{m}| \leq \sigma + \varepsilon$. This notation will be used for the sake of simplicity. Since the integral domain is limited to the circle such that $|\mathbf{m}| \approx \sigma$, $\widehat{w}(\mathbf{n})$ is effective only when $|\mathbf{n}| \approx \sigma$. Thus we assume here that

$$\widehat{w}(\mathbf{n}) = \begin{cases} \widehat{w}(\mathbf{n}) & |\mathbf{n}| = \sigma, \\ 0 & \text{otherwise.} \end{cases}$$

On the circle such that $|\mathbf{n}| = \sigma$, $\widehat{w}(\mathbf{n})$ can be considered as one-dimensional periodic function. Thus, we introduce polar coordinates such that $\mathbf{n} \stackrel{\text{def}}{=} \sigma e^{i\theta_n}$ and $\mathbf{m} \stackrel{\text{def}}{=} \sigma e^{i\theta_m}$. Then we have

$$\begin{aligned} |\mathbf{n}-\mathbf{m}| &= \sqrt{|\mathbf{n}|^2 - 2|\mathbf{n}||\mathbf{m}|\cos(\theta_n - \theta_m) + |\mathbf{m}|^2} \\ &= \sigma\sqrt{2\,(1-\cos(\theta_n - \theta_m))}. \end{aligned}$$

Since $\widehat{w}(\mathbf{n})$ and $\widehat{A}(\mathbf{n})$ are functions of θ, we denote these by $\widehat{w}(\theta)$ and $\widehat{A}(\theta)$. Therefore Eqn. (5.8) can be written as

$$\dot{\widehat{w}}\,(\theta_n) = \int_0^{2\pi} \widehat{A}(\theta_n - \theta_m)\widehat{w}(\theta_m)d\theta_m, \tag{5.9}$$

where

When A is Gaussian:

$$\widehat{A}(\theta) = 2\pi\sigma_A^2 \exp\left(-\sigma_A^2\sigma^2|1-\cos\theta|\right),$$

When A is Step:

$$\widehat{A}(\theta) = 2\pi R^2 J_1\left(R\sigma\sqrt{2|1-\cos\theta|}\right)/R\sigma\sqrt{2|1-\cos\theta|}.$$

The FT of Eqn. (5.9) is

$$\dot{\widehat{w}}(k) = \widehat{A}(k)\widehat{w}(k) \quad (k = 0, 1, 2, \cdots),$$

and its solution is

$$\widehat{w}(k) = \widehat{w}^{(0)}(k)\exp(\widehat{A}(k)t). \tag{5.10}$$

By computing the inverse Fourier transform of Eqn. (5.10), we have the solution of Eqn. (5.9) as

$$w(\theta) = \sum_{k=0}^{\infty}\widehat{w}(k)\exp(ik\theta) = \sum_{k=0}^{\infty}\widehat{w}^{(0)}(k)\exp(ik\theta)\exp(\widehat{A}(k)t).$$

Now we obtain the solution of Eqn. (5.4) in the form of Eqn. (5.6). Let $a_{\sigma,k}$, $c_{\sigma,k}(\alpha)$, and σ_θ be

$$a_{\sigma,k} \stackrel{\text{def}}{=} \sigma\widehat{w}^{(0)}(k), \quad c_{\sigma,k}(\alpha) \stackrel{\text{def}}{=} \int_0^{2\pi}\exp(ik\theta)\exp(i\sigma_\theta\alpha)d\theta, \quad \text{and}$$

$$\sigma_\theta \stackrel{\text{def}}{=} (\sigma\cos\theta, \sigma\sin\theta),$$

then we have by the inverse Fourier transform

$$w(\alpha) = \sigma\int_0^{2\pi}\widehat{w}(\theta)\exp(i\sigma_\theta\alpha)d\theta = \sum_{k=0}^{\infty}a_{\sigma,k}c_{\sigma,k}(\alpha)\exp(\widehat{A}(k)t).$$

As mentioned previously, for any k, k', the difference between $\exp(\widehat{A}(k)t)$ and $\exp(\widehat{A}(k')t)$ will be exponential as t increases. Thus, we can consider that $w(\alpha)$ will be dominated by $c_{\sigma,k_*}(\alpha)$ such that $k_* = \arg\max_k(\widehat{A}(k))$. Therefore in the following, we consider k that maximizes $\widehat{A}(k)$ in each case of arbor functions.

When $A(\alpha)$ is Gaussian, $\widehat{A}(k)$ is calculated as follows.

$$\widehat{A}(k) = 4\pi^2\sigma_A^2\exp(-\sigma_A^2\sigma^2)I_k(\sigma_A^2\sigma^2),$$

where I_k is the modified Bessel function of the k-th order. Some calculation shows that $I_0(x) > I_1(x) \Longrightarrow I_0(x) > I_k(x)$ for any x and k. We can numerically verify that $I_0(x) > I_1(x)$ holds for any k and any x less than a sufficiently large constant (e.g., 1,000,000). Hence $\widehat{\widehat{A}}(0) > \widehat{\widehat{A}}(k)$ holds for any k and $\sigma_A^2\sigma^2$ in an appropriate range. As a result, $w(\alpha)$ is dominated by $c_{\sigma,0}(\alpha)$ (or s-mode in MacKay & Miller's notation) and circular symmetric receptive fields are developed.

On the other hand, when A is Step, we have

$$\widehat{\widehat{A}}(k) = \pi^2 R^2 \left(J_k^2(R\sigma) + (-1)^k J_{1-k}(R\sigma) J_{1+k}(R\sigma) \right).$$

By some calculation, we can prove that $\widehat{\widehat{A}}(k) \geq \widehat{\widehat{A}}(k+2)$ holds for any k and $R\sigma$. Figure5.3 shows $\widehat{\widehat{A}}(k)$ of $k = 0, 1, 2, 3$ varying $R\sigma \in [0, 10]$. As can be seen, $\widehat{\widehat{A}}(0)$ and $\widehat{\widehat{A}}(1)$ become the maximum alternatively, and $\widehat{\widehat{A}}(0) \geq \widehat{\widehat{A}}(2)$ always holds. Therefore in this case $w(\alpha)$ is almost always dominated by $c_{\sigma,0}(\alpha)$ or $c_{\sigma,1}(x)$ (corresponds to s-mode or p-mode in MacKay & Miller's notation). If $c_{\sigma,1}(x)$ is dominant, the developed receptive field is oriented (e.g., bi-lobed, quadri-lobed, and so on).

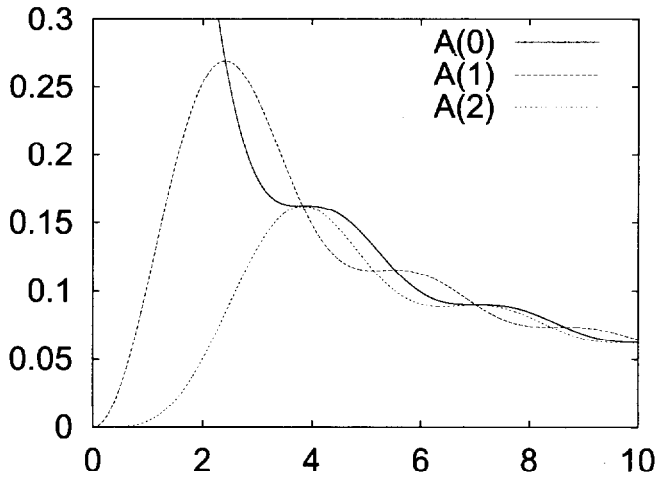

Figure 5.3: Graphs of $\widehat{\widehat{A}}(0)$, $\widehat{\widehat{A}}(1)$, and $\widehat{\widehat{A}}(2)$. The horizontal and vertical axes respectively show values of $R\sigma$ and these functions. As can be seen, $\widehat{\widehat{A}}(0) \geq \widehat{\widehat{A}}(2)$ always holds and $\widehat{\widehat{A}}(0)$ or $\widehat{\widehat{A}}(1)$ becomes the maximum alternatively.

In $c_{\sigma,1}(\alpha)$, the orientation of elongation of stripes within the receptive field may

be of interest. Assuming that the receptive field is $c_{\sigma,1}(\alpha)$ for some σ. Then

$$\begin{aligned} w(\alpha) &\approx a_{\sigma,1} c_{\sigma,1}(\alpha) \exp(\widehat{\widehat{A}}(1)t) \\ &= \sigma \int_0^{2\pi} \widehat{\widehat{w}}^{(0)}(1) \exp(i\theta) \exp(i\sigma_\theta \alpha) d\theta \exp(\widehat{\widehat{A}}(1)t) \\ &= \sigma \int_0^{2\pi} (\Re(\widehat{\widehat{w}}^{(0)}(1)) \cos\theta + \Im(\widehat{\widehat{w}}^{(0)}(1)) \sin\theta) \exp(i\sigma_\theta \alpha) d\theta \exp(\widehat{\widehat{A}}(1)t) \\ &= \sigma c \int_0^{2\pi} \cos(\theta - \phi) \exp(i\sigma_\theta \alpha) d\theta \exp(\widehat{\widehat{A}}(1)t) \\ &= \sigma c \int_0^{2\pi} \cos(\theta - \phi) \sin(\sigma_\theta \alpha) d\theta \exp(\widehat{\widehat{A}}(1)t), \end{aligned}$$

where

$$c = \sqrt{\Re^2(\widehat{\widehat{w}}^{(0)}(1)) + \Im^2(\widehat{\widehat{w}}^{(0)}(1))}, \quad \text{and,} \quad \phi = \tan^{-1}(\Im(\widehat{\widehat{w}}^{(0)}(1))/\Re(\widehat{\widehat{w}}^{(0)}(1))).$$

Since $\cos(\theta - \phi)$ takes the maximum when $\theta = \phi$, this ϕ is the orientation of the elongation. It should be noted that we can predict the orientation *a priori*, because this ϕ is a constant determined by the initial weights.

Finally, as is also shown in Fig.5.3, there are several points at which several $\widehat{\widehat{A}}(k)$ take the same maximal value. This suggests that it is possible to develop receptive fields that have two or more radial nodes, or the combination of several components in Fig.5.1. For example, when $R\sigma \approx 3.81$, the values of $\hat{A}(0)$, $\hat{A}(1)$, and $\hat{A}(2)$ are approximately 0.162. Thus if $R\sigma \approx 3.81$, the values of $\widehat{\widehat{w}}^{(0)}(0)$ and $\widehat{\widehat{w}}^{(0)}(2)$ are almost equal, and the value of $\widehat{\widehat{w}}^{(0)}(1)$ is much smaller than these, then the developed receptive field could be tri-lobed shape, which cannot be obtained by a single component.

5.3 The roles of k_1, k_2

So far we saw that the bi-modal shape of $\widehat{C}(\mathbf{n})$ is crucial. Here we will see that a good choice of k_1, k_2 gives similar effects even in the case that $\widehat{C}(\mathbf{n})$ is uni-modal (e.g., Gaussian). MacKay & Miller [54] have also obtained similar results.

First we discuss the role of k_1. As we saw, our modification rule (Eqn. (5.1)) can be expressed in terms of the FT as follows.

$$\dot{\widehat{w}}(\alpha) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \widehat{A}(\mathbf{n} - \mathbf{m}) \widehat{C}(\mathbf{m}) \widehat{w}(\mathbf{m}) e^{i\mathbf{n}\alpha} d\mathbf{m} d\mathbf{n}.$$

The following equation is identical to the above one.

$$\dot{w}(\alpha) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\widehat{C}(\mathbf{n})\int_{-\infty}^{\infty}\widehat{A}(\mathbf{n}-\mathbf{m})\widehat{w}(\mathbf{m})e^{i\mathbf{n}\alpha}d\mathbf{m}d\mathbf{n}.$$

We separate the right hand side into the component $\mathbf{n} = \mathbf{0}$ and the other components. Then the rule becomes

$$\begin{aligned}\dot{w}(\alpha) = &\frac{1}{4\pi^2}\widehat{C}(\mathbf{0})\int_{-\infty}^{\infty}\widehat{A}(-\mathbf{m})\widehat{w}(\mathbf{m})d\mathbf{m}\\ &+\frac{1}{4\pi^2}\int_{\mathbf{n}\neq\mathbf{0}}\widehat{C}(\mathbf{n})\int_{-\infty}^{\infty}\widehat{A}(\mathbf{n}-\mathbf{m})\widehat{w}(\mathbf{m})e^{i\mathbf{n}\alpha}d\mathbf{m}d\mathbf{n}.\end{aligned}$$

The first term on the right hand side is some constant independent of α. If $\widehat{C}(\mathbf{n})$ is uni-modal then this term dominates. Therefore, it is possible to define k_1 to suppress the effect of this term in computer simulations.

Next we consider the role of k_2. Here we assume that k_2 is the following function $k_2(\alpha)$.

$$k_2(\alpha) = \begin{cases} k_2 & |\alpha| \leq R_A, \\ 0 & \text{otherwise}, \end{cases}$$

that is, k_2 is effective only within the arbor radius. The FT of $k_2(\mathbf{n})$ can be computed as

$$\widehat{k_2}(\mathbf{n}) = \int_{-\infty}^{\infty}k_2(\alpha)e^{-i\mathbf{n}\alpha}d\alpha = 2\pi k_2 R_A^2\frac{J_1(|\mathbf{n}|R_A)}{|\mathbf{n}|R_A}.$$

This function takes the maximum value $2\pi k_2 R_A$ at the center and decreases to 0 towards $\mathbf{n}$ such that $|\mathbf{n}| \approx 3.8/R_A \approx \pi/R_A$. After that, this function oscillates and finally converges to 0. Thus, we can regard $\widehat{k_2}(\mathbf{n})$ as a sharp uni-modal function which is "effective" only within the range $|\mathbf{n}| \leq \pi/R_A$ (Fig.5.4 left). Hence even if $\widehat{C}(\mathbf{n})$ is uni-modal, by taking k_2 appropriately, we can make $\widehat{C}(\mathbf{n}) - \widehat{k_2}(\mathbf{n}) = \widehat{(C-k_2)}(\mathbf{n})$ bi-modal (Fig.5.4 right). Moreover, the peak of $\widehat{(C-k_2)}(\mathbf{n})$ appears around $|\mathbf{n}| \approx \pi/R_A$. Thus, the number of radial nodes of a developed receptive field is 1.

5.4 Conclusions

We considered Linsker's neural network model and studied how oriented RFs were developed at layer $F \rightarrow G$ mathematically. We concentrated on the Mexican-hat correlation function and showed that this function determined the spatial frequency

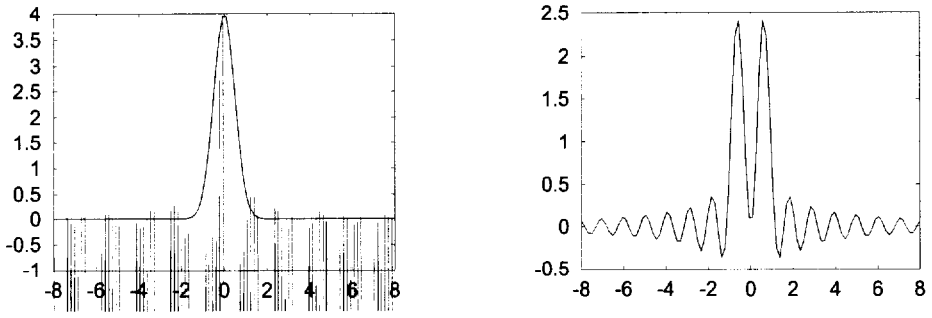

Figure 5.4: Left: The shape of $\widehat{C}(\mathbf{n})$ (real line) and that of $\widehat{k_2}(\mathbf{n})$ (dashed line). We assume that $C(\alpha) = \exp(-x^2/2\sigma_c^2)/2\pi$, then $\widehat{C}(\mathbf{n}) = \sigma_c^2 \exp(-\sigma_c^2 x^2/2)$ is uni-modal, where $\sigma_c^2 = 4.0$. Therefore the peak value, i.e., the value of $\widehat{C}(\mathbf{0})$ is 4.0. On the other hand, k_2 is chosen to suffice $\widehat{k_2}(\mathbf{0}) = 4.0$, that is, $2\pi k_2 R_A^2 = 4.0$. Under this parameter, $\widehat{k_2}(\mathbf{n}) = 4.0 J_1(|\mathbf{n}|R_A)/|\mathbf{n}|R_A$, where $R_A = 6.5$. Right: the shape of $(\widehat{C - k_2})(\mathbf{n})$. As can be seen, the shape is bi-modal.

of RFs. We also focused on the role of the arbor function. Two types of arbor functions were considered, and it was shown that the Gaussian arbor function developed center-surround RFs while the step arbor function enabled the development of multi-lobed RFs in addition to center-surround RFs.

We performed the FT of the modification rule twice and solved the modification rule under the assumption that the FT of the correlation function was a sharp bi-modal function with two clear peaks. The solution was expressed as a weighted sum of Fourier components defined with respect to the spatial frequency and the number of angular nodes, where the weights increased exponentially with respect to the time and the two FTs of the arbor function. Since the difference between weights of any two components grew exponentially as the self-organization proceeded, the formation of the RF was conquered by one of the components (or by a combination of several components if the weights of such components grows at the same speed).

In summary, our mathematical analysis indicated the roles of the correlation function and the arbor function. Furthermore we showed how the orientation of multi-lobed RFs is determined and finally we discussed the roles of parameters k_1, k_2 in Linsker's model.

Chapter 6

Emergence of orientation selectivity for many neurons with lateral connections

In this chapter, we demonstrate a mathematical analysis of the emergence of orientation selectivity for many neurons with lateral connections, in particular, how each neuron develops its oriented receptive field and how the preferred orientations and phases are arranged on the cortical surface. We consider a correlation-based model for orientation map formation proposed by Miller [60] reviewed in Section 3.6 and study the formation mathematically. We perform the Fourier transform and compute the principal component of the model. With our analysis, the roles of functions are clarified as follows. (1) A Mexican-hat-type correlation function determines the spatial frequencies of receptive fields; (2) The combination of a Gaussian arbor function and a Gaussian intracortical function develops oriented receptive fields; (3) The intracortical function develops stripes similar to ocular dominance columns on a layer on which each cell has its receptive field if the range is sufficiently wide; (4) The preferred orientations and the phases of receptive fields for each cell are determined by the stripes within the receptive fields. These results indicate that the developed orientation maps have the following properties. (1) It develops oriented receptive fields; (2) Preferred orientations smoothly change on the cortical surface; (3) The periodicity of preferred orientations does not appear; (4) The periodicity of phases appears; (5) Singular points appear irregularly on the cortical surface. Our analytical results are justified by computer simulations.

Results presented in this chapter partially appeared at [110, 111].

6.1 Preliminaries

6.1.1 Model to study

We investgate the dynamics of Eqn. (3.3) in the case of which A, I, and C are given by the following equations.

$$\begin{aligned} A(\mathbf{x}) &\stackrel{\text{def}}{=} \exp\left(-|\mathbf{x}|^2/2\sigma_A^2\right), \quad I(\mathbf{x}) \stackrel{\text{def}}{=} \exp\left(-|\mathbf{x}|^2/2\sigma_I^2\right), \\ C(\alpha) &\stackrel{\text{def}}{=} C_1 \exp\left(-|\alpha|^2/2\sigma_{C1}^2\right) - C_2 \exp\left(-|\alpha|^2/2\sigma_{C2}^2\right), \end{aligned} \tag{6.1}$$

$$\begin{aligned} C^{\text{ON-ON}}(\alpha) &\stackrel{\text{def}}{=} C(\alpha) \quad (\text{resp., } C^{\text{OFF-OFF}}), \\ C^{\text{ON-OFF}}(\alpha) &\stackrel{\text{def}}{=} -\varepsilon C(\alpha) \quad (\text{resp., } C^{\text{OFF-ON}}). \end{aligned}$$

We consider some simplifications. First, we consider the continuous version of the modification rule. Second, we omit the lower and upper bounds and the normalization. We assume that the dynamics of the model is determined in the early stages of the self-organization, i.e., before synaptic weights do not reach the saturation limits. On the other hand, Miller and MacKay [62] have shown that the subtractive normalization does not change the principal component. Thus, without the normalization, the dynamics of the model may not change. Third, we introduce $w^{(t)}(\mathbf{x}, \alpha)$ as the difference between $w^{\text{ON}(t)}(\mathbf{x}, \alpha)$ and $w^{\text{OFF}(t)}(\mathbf{x}, \alpha)$ as $w^{(t)}(\mathbf{x}, \alpha) \stackrel{\text{def}}{=} w^{\text{ON}(t)}(\mathbf{x}, \alpha) - w^{\text{OFF}(t)}(\mathbf{x}, \alpha)$. Fourth, for each $w^{(t)}(\mathbf{x}, \alpha)$, we consider the position α as the relative position $\Delta\alpha$ from the position $\mathbf{x}$. We define $w^{(t)}(\mathbf{x}, \Delta\alpha)$ as $w^{(t)}(\mathbf{x}, \Delta\alpha) \stackrel{\text{def}}{=} w^{(t)}(\mathbf{x}, \alpha - \mathbf{x})$. From the above simplifications, it follows that

$$\begin{aligned} \Delta w^{(t)}(\mathbf{x}, \Delta\alpha) = A(\Delta\alpha) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} & I(\mathbf{x} - \mathbf{y}) \\ & C\left((\mathbf{x} - \mathbf{y}) + (\Delta\alpha - \Delta\beta)\right) w^{(t)}(\mathbf{y}, \Delta\beta) d\Delta\beta d\mathbf{y}. \end{aligned} \tag{6.2}$$

In this chapter, we analyze the dynamics of the modification rule defined by Eqn.(6.2).

6.1.2 Mathematical Framework

Since Eqn. (6.2) is a linear equation, it can be solved by expanding $w^{(t)}(\mathbf{x}, \Delta\alpha)$ into some independent bases. One obtains

$$w^{(t)}(\mathbf{x}, \Delta\alpha) = \sum_{k=0}^{\infty} a_k \exp(f_k t) p_k(\mathbf{x}, \Delta\alpha),$$

where a_k and f_k are constants while $p_k(\mathbf{x}, \Delta\alpha)$ is a basis. Since

$$\lim_{t\to\infty} |\exp(f_k t) - \exp(f_{k'} t)| = \infty$$

for some $f_k \neq f_{k'}$, $p_{k_*}(\mathbf{x}, \Delta\alpha)$ such that $k_* = \arg\max_k f_k$ will dominate $w^{(t)}(\mathbf{x}, \Delta\alpha)$. Therefore, our goal is to find such decomposition and $p_{k_*}(\mathbf{x}, \Delta\alpha)$ which we call *the dominant component.*

In a similar way, we assume that $\Delta w^{(t)}(\mathbf{x}, \Delta\alpha)$ decomposes into coefficients c_k of t and bases $q_k(\mathbf{x}, \Delta\alpha)$.

$$\Delta w^{(t)}(\mathbf{x}, \Delta\alpha) = \sum_{k=0}^{\infty} c_k(t) q_k(\mathbf{x}, \Delta\alpha).$$

Then, $w^{(t+1)}(\mathbf{x}, \Delta\alpha)$ can be computed as follows.

$$\begin{aligned} w^{(t+1)}(\mathbf{x}, \Delta\alpha) &= \Delta w^{(t)}(\mathbf{x}, \Delta\alpha) + w^{(t)}(\mathbf{x}, \Delta\alpha) = \cdots \\ &= \sum_{k=0}^{\infty} \sum_{s=0}^{t} c_k(s) q_k(\mathbf{x}, \Delta\alpha) + w^{(0)}(\mathbf{x}, \Delta\alpha). \end{aligned}$$

Thus, if $c_{k_0}(t) > c_{k_1}(t) > c_{k_2}(t) > \cdots$ holds for some $k_0, k_1, \cdots$ and for any t, and if each $c_{k_i}(t)$ increases exponentially as t increases, it follows that the structure of $w^{(t+1)}(\mathbf{x}, \Delta\alpha)$ is conquered by $q_{k_0}(\mathbf{x}, \Delta\alpha)$. Therefore, our goal now is to find such decomposition of $\Delta w^{(t)}(\mathbf{x}, \Delta\alpha)$ and $q_{k_0}(\mathbf{x}, \Delta\alpha)$ which we also call the dominant component.

6.2 Mathematical Analysis

This section contains our main results. First we perform the Fourier transform of Eqn. (6.2) and compute the decomposition of $\Delta w^{(t)}(\mathbf{x}, \Delta\alpha)$ in the Fourier space, namely, the decomposition of $\Delta\widehat{w}^{(t)}(\mathbf{u}, \mathbf{n})$, where the symbol $\widehat{\ }$ denotes the Fourier transform. Then we obtain the dominant component of $\Delta\widehat{w}^{(t)}(\mathbf{u}, \mathbf{n})$. After that, by computing the Fourier inverse transform, we obtain the dominant component of $\Delta w^{(t)}(\mathbf{x}, \Delta\alpha)$, that is, the dominant component of $w^{(t)}(\mathbf{x}, \Delta\alpha)$. Finally, we give an interpretation of how the dominant component describes the formation of orientation maps.

The Fourier transform of Eqn. (6.2) is as follows.

$$\Delta\widehat{w}^{(t)}(\mathbf{u}, \mathbf{n}) = \widehat{I}(\mathbf{u} - \mathbf{n}) \int_{-\infty}^{\infty} \widehat{A}(\mathbf{n} - \mathbf{m}) \widehat{C}(\mathbf{m}) \widehat{w}^{(t)}(\mathbf{u}, \mathbf{m}) d\mathbf{m},$$

where

$$\widehat{A}(\mathbf{l}) = \sigma_A^2 e^{-\sigma_A^2|\mathbf{l}|^2/2}, \quad \widehat{I}(\mathbf{l}) = \sigma_I^2 e^{-\sigma_I^2|\mathbf{l}|^2/2},$$
$$\widehat{C}(\mathbf{l}) = C_1\sigma_{C1}^2 e^{-\sigma_{C1}^2|\mathbf{l}|^2/2} - C_2\sigma_{C2}^2 e^{-\sigma_{C2}^2\mathbf{l}^2/2}.$$

It is known that the Fourier transform of a DOG function takes the form of bi-modal (Fig.6.1) depending on the choice of parameters. Since $C(\alpha)$ is a DOG function, we assume $\widehat{C}(\mathbf{l})$ is a bi-modal function which shows its peaks at $\sigma \stackrel{\text{def}}{=} |\mathbf{l}|$. Then, the integral domain could be restricted to the circle of radius σ because only $\widehat{w}^{(t)}(\mathbf{u},\mathbf{n})$ such that $|\mathbf{n}| = \sigma$ are effective. Thus, we have

$$\Delta\widehat{w}^{(t)}(\mathbf{u},\mathbf{n}) \approx \widehat{I}(\mathbf{u}-\mathbf{n}) \int_{|\mathbf{m}|=\sigma} \widehat{A}(\mathbf{n}-\mathbf{m})\widehat{w}^{(t)}(\mathbf{u},\mathbf{m})d\mathbf{m}. \tag{6.3}$$

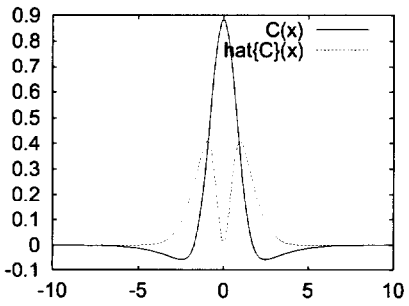

Figure 6.1: Shapes of a DOG function $C(\alpha)$ in Eqn. (6.1) (real line) and its Fourier transform (dashed line) under the following parameters. $C_1 = 1.00$, $C_2 = 1/9$, $\sigma_{C1}^2 = 0.61$, and $\sigma_{C2}^2 = 5.45$.

Since $|\mathbf{n}|$ and $|\mathbf{m}|$ are σ, we represent Eqn. (6.3) in polar coordinates. We denote $\mathbf{u}$,$\mathbf{n}$,$\mathbf{m}$ as

$$\mathbf{u} \stackrel{\text{def}}{=} ue^{i\theta_u}, \quad \mathbf{n} \stackrel{\text{def}}{=} \sigma e^{i\theta_n}, \quad \text{and} \quad \mathbf{m} \stackrel{\text{def}}{=} \sigma e^{i\theta_m},$$

then

$$|\mathbf{u}-\mathbf{n}| = \sqrt{u^2+\sigma^2-2u\sigma\cos(\theta_u-\theta_n)}, \quad \text{and}$$
$$|\mathbf{n}-\mathbf{m}| = \sigma\sqrt{2(1-\cos(\theta_n-\theta_m))}.$$

Let $\widetilde{I}(u,\theta)$, $\widetilde{A}(\theta)$, and $\widetilde{w}^{(t)}(u,\theta_u,\theta_n)$ be as follows.

$$\widetilde{I}(u,\theta) \stackrel{\text{def}}{=} \widehat{I}\left(\sqrt{u^2+\sigma^2-2u\sigma\cos\theta}\right),$$
$$\widetilde{A}(\theta) \stackrel{\text{def}}{=} \widehat{A}\left(\sigma\sqrt{2(1-\cos\theta)}\right), \quad \text{and}$$
$$\widetilde{w}^{(t)}(u,\theta_u,\theta_n) \stackrel{\text{def}}{=} \widehat{w}^{(t)}(\mathbf{u},\mathbf{n}).$$

Thus, Eqn. (6.3) is written as

$$\Delta\widetilde{w}^{(t)}(u,\theta_u,\theta_n) = \widetilde{I}(u,\theta_u-\theta_n)\int_0^{2\pi}\widetilde{A}(\theta_n-\theta_m)\widetilde{w}^{(t)}(u,\theta_u,\theta_m)d\theta_m.$$

Defining $\Delta\theta_n$ and $\Delta\theta_m$ respectively as $\Delta\theta_n \stackrel{\text{def}}{=} \theta_u-\theta_n$ and $\Delta\theta_m \stackrel{\text{def}}{=} \theta_u-\theta_m$, and rewriting $\Delta\widetilde{w}^{(t)}(u,\theta_u,\theta_u-\Delta\theta_n)$ as $\Delta\widetilde{w}_*^{(t)}(u,\theta_u,\Delta\theta_n)$, we obtain

$$\Delta\widetilde{w}_*^{(t)}(u,\theta_u,\Delta\theta_n) = \widetilde{I}(u,\Delta\theta_n)\int_0^{2\pi}\widetilde{A}(\Delta\theta_n-\Delta\theta_m)\widetilde{w}_*^{(t)}(u,\theta_u,\Delta\theta_m)d\Delta\theta_m.$$

By the Fourier series expansion, we can restate it as follows.

$$\Delta\widetilde{w}_*^{(t)}(u,\theta_u,\Delta\theta_n) = \widetilde{I}(u,\Delta\theta_n)\sum_{k=0}^{\infty}\widehat{\widetilde{A}}(k)\widehat{\widetilde{w}}_*^{(t)}(u,\theta_u,k)e^{ik\Delta\theta_n}, \tag{6.4}$$

where

$$\widehat{\widetilde{A}}(k) = 4\pi^2\sigma_A^2 e^{-\sigma_A^2\sigma^2}I_k(\sigma_A^2\sigma^2), \tag{6.5}$$

and I_k is the modified Bessel function of the k th order.

We assume that $\Delta\widetilde{w}_*^{(t)}(u,\theta_u,\Delta\theta_n)$ can be decomposed into a series of $\widehat{\widetilde{w}}_*^{(0)}(u,\theta_u,k)$ $\widetilde{I}(u,\Delta\theta_n)\,e^{ik\Delta\theta_n}$ such that $k=0,1,2,\cdots$. Let $c(k,t)$ be the coefficient of the k th component at the t th step. We obtain

$$\Delta\widetilde{w}_*^{(t)}(u,\theta_u,\Delta\theta_n) = \sum_{k=0}^{\infty}c(k,t)\widehat{\widetilde{w}}_*^{(0)}(u,\theta_u,k)\widetilde{I}(u,\Delta\theta_n)e^{ik\Delta\theta_n}. \tag{6.6}$$

We first show that the 0 th component, i.e., $\widehat{\widetilde{w}}_*^{(0)}(u,\theta_u,0)\widetilde{I}(u,\Delta\theta_n)$, grows faster.

This is stated inductively. At the 0 th step, it follows from Eqn.(6.4),

$$\Delta\widetilde{w}_*^{(0)}(u,\theta_u,\Delta\theta_n) = \sum_{k=0}^{\infty}\widehat{\widetilde{A}}(k)\widehat{\widetilde{w}}_*^{(0)}(u,\theta_u,k)\widetilde{I}(u,\Delta\theta_n)e^{ik\Delta\theta_n}.$$

Since initial weights are assigned uniformly at random, $\widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, k)$ are almost equal for any k. Moreover for any k and feasible range of $\sigma_A^2\sigma^2$ in Eqn.(6.5) (e.g., less than 1000000), we can numerically verify that $\widehat{\widetilde{A}}(0) > \widehat{\widetilde{A}}(k)$. Hence at the 0th step, the 0 th component grows faster.

Next, at the t th step, we assume that $\Delta\widetilde{w}_*^{(t)}(u, \theta_u, \Delta\theta_n)$ can be decomposed as Eqn.(6.6) and the 0 th component grows faster. Then at the $t+1$ th step,

$$\begin{aligned}
&\Delta\widetilde{w}_*^{(t+1)}(u, \theta_u, \Delta\theta_n) \\
&= \widetilde{I}(u, \Delta\theta_n) \sum_{k=0}^{\infty} \widehat{\widetilde{A}}(k) \widehat{\widetilde{w}}_*^{(t+1)}(u, \theta_u, k) e^{ik\Delta\theta_n} \\
&= \widetilde{I}(u, \Delta\theta_n) \sum_{k=0}^{\infty} \widehat{\widetilde{A}}(k) \left(\widehat{\widetilde{w}}_*^{(t)}(u, \theta_u, k) + \Delta\widehat{\widetilde{w}}_*^{(t)}(u, \theta_u, k) \right) e^{ik\Delta\theta_n} \\
&= \Delta\widetilde{w}_*^{(t)}(u, \theta_u, \Delta\theta_n) + \widetilde{I}(u, \Delta\theta_n) \sum_{k=0}^{\infty} \widehat{\widetilde{A}}(k) \Delta\widehat{\widetilde{w}}_*^{(t)}(u, \theta_u, k) e^{ik\Delta\theta_n}.
\end{aligned}$$

Since

$$\begin{aligned}
\Delta\widehat{\widetilde{w}}_*^{(t)}(u, \theta_u, k) &= \int_0^{2\pi} \Delta\widetilde{w}_*^{(t)}(u, \theta_u, \theta) e^{-ik\theta} d\theta \\
&= \int_0^{2\pi} \sum_{l=0}^{\infty} c(l, t) \widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, l) \widetilde{I}(u, \theta) e^{il\theta} e^{-ik\theta} d\theta \\
&= \sum_{l=0}^{\infty} c(l, t) \widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, l) \widehat{\widetilde{I}}(u, k-l),
\end{aligned}$$

where

$$\widehat{\widetilde{I}}(u, k) = 4\pi^2\sigma_I^2 e^{-\sigma_I^2(u^2+\sigma^2)/2} I_k(\sigma_I^2 u\sigma), \tag{6.7}$$

it follows that

$$\begin{aligned}
&\Delta\widetilde{w}_*^{(t+1)}(u, \theta_u, \Delta\theta_n) = \Delta\widetilde{w}_*^{(t)}(u, \theta_u, \Delta\theta_n) \\
&\quad + \widetilde{I}(u, \Delta\theta_n) \sum_{k=0}^{\infty} \widehat{\widetilde{A}}(k) \sum_{l=0}^{\infty} c(l, t) \widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, l) \widehat{\widetilde{I}}(u, k-l) e^{ik\Delta\theta_n}.
\end{aligned} \tag{6.8}$$

From the assumption, the fastest growing component of $\Delta\widetilde{w}_*^{(t)}(u, \theta_u, \Delta\theta_n)$ is the 0 th component. Thus we concentrate on computing the fastest growing component of the second term of the right hand side.

Let $d(u, \theta_u, \Delta\theta_n, t)$ be the second term of the right hand side in Eqn.(6.8) as follows.

$$
\begin{aligned}
& d(u, \theta_u, \Delta\theta_n, t) \\
& \stackrel{\text{def}}{=} \widetilde{I}(u, \Delta\theta_n) \sum_{k=0}^{\infty} \widehat{\widetilde{A}}(k) \sum_{l=0}^{\infty} c(l,t) \widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, l) \widehat{\widetilde{I}}(u, k-l) e^{ik\Delta\theta_n} \\
& = \widetilde{I}(u, \Delta\theta_n) \sum_{l=0}^{\infty} \widehat{\widetilde{A}}(l) \sum_{k=0}^{\infty} c(k,t) \widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, k) \widehat{\widetilde{I}}(u, l-k) e^{il\Delta\theta_n} \\
& = \widetilde{I}(u, \Delta\theta_n) \sum_{l=0}^{\infty} \widehat{\widetilde{A}}(l) \sum_{k=0}^{\infty} c(k,t) \widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, k) \widehat{\widetilde{I}}(u, l-k) e^{ik\Delta\theta_n} e^{i(l-k)\Delta\theta_n} \\
& = \widetilde{I}(u, \Delta\theta_n) \sum_{k=0}^{\infty} c(k,t) \widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, k) e^{ik\Delta\theta_n} \sum_{l=0}^{\infty} \widehat{\widetilde{A}}(l) \widehat{\widetilde{I}}(u, l-k) e^{i(l-k)\Delta\theta_n} \\
& = \widetilde{I}(u, \Delta\theta_n) \sum_{k=0}^{\infty} c(k,t) \widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, k) e^{ik\Delta\theta_n} \sum_{m=0}^{\infty} \widehat{\widetilde{A}}(m+k) \widehat{\widetilde{I}}(u, m) e^{im\Delta\theta_n}.
\end{aligned}
$$

The coefficient of the k th component, which we denote $c'(k,t)$, is as follows.

$$
\begin{aligned}
c'(k,t) & \stackrel{\text{def}}{=} c(k,t) \sum_{m=0}^{\infty} \widehat{\widetilde{A}}(m+k) \widehat{\widetilde{I}}(u, m) e^{im\Delta\theta_n} \\
& < c(k,t) \sum_{m=0}^{\infty} \widehat{\widetilde{A}}(m+k) \widehat{\widetilde{I}}(u, m) \\
& \leq c(k,t) \sum_{m=0}^{\infty} \widehat{\widetilde{A}}(m) \widehat{\widetilde{I}}(u, m).
\end{aligned}
$$

From the assumption, $c(0,t) > c(k,t)$ holds for any k. Hence the fastest growing component of $d(u, \theta_u, \Delta\theta_n, t)$ is also the 0 th component. Therefore at the $t+1$ step, the 0 th component grows faster and finally we conclude that the 0 th component grows faster at every step.

We next show that $c(0,t)$ grows exponentially. This makes sure that the 0th component is the dominant component of $\Delta\widetilde{w}_*^{(t)}(u, \theta_u, \Delta\theta_n)$. A rough calculation shows that

$$
c(0,t) > \widehat{\widetilde{A}}(0)^{t+1} \widehat{\widetilde{I}}(u, 0)^t.
$$

Hence the 0th component is the dominant component of $\Delta\widetilde{w}_*^{(t)}(u, \theta_u, \Delta\theta_n)$ because the 0th component grows faster than the others and the growing speed is exponential with respect to t. As mentioned in the previous section, the 0th component

is also the dominant component of $\widetilde{w}_*^{(t)}(u, \theta_u, \Delta\theta_n)$. Therefore estimating $c(0,t)$ simply as $\widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(u,0)^t$, we obtain

$$\widetilde{w}_*^{(t)}(u, \theta_u, \Delta\theta_n) \approx \widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(u,0)^t\widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, 0)\widetilde{I}(u, \Delta\theta_n).$$

Then it follows

$$\begin{aligned}
&\widetilde{w}_*^{(t)}(u, \theta_u, \Delta\theta_n) \\
&\approx \widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(u,0)^t\widehat{\widetilde{w}}_*^{(0)}(u, \theta_u, 0)\widetilde{I}(u, \Delta\theta_n) \\
&= \widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(u,0)^t \int_0^{2\pi} \widetilde{w_*}^{(0)}(u, \theta_u, \theta)d\theta \cdot \widetilde{I}(u, \Delta\theta_n) \\
&= \widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(u,0)^t \int_0^{2\pi} \widetilde{w}^{(0)}(u, \theta_u, \theta_u - \theta)d\theta \cdot \widetilde{I}(u, \Delta\theta_n) \\
&= \widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(u,0)^t \int_0^{2\pi} \widetilde{w}^{(0)}(u, \theta_u, \theta)d\theta \cdot \widetilde{I}(u, \Delta\theta_n) \\
&= \widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(u,0)^t\widetilde{w}^{(0)}(u, \theta_u)\widetilde{I}(u, \Delta\theta_n),
\end{aligned} \tag{6.9}$$

where

$$\widetilde{w}^{(0)}(u, \theta_u) \stackrel{\text{def}}{=} \int_0^{2\pi} \widetilde{w}^{(0)}(u, \theta_u, \theta)d\theta.$$

Since $\widetilde{w_*}^{(t)}(u, \theta_u, \Delta\theta_n) = \widetilde{w}^{(t)}(u, \theta_u, \theta_u - \Delta\theta_n)$, we can restate Eqn.(6.9) as

$$\widetilde{w}^{(t)}(u, \theta_u, \theta_n) = \widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(u,0)^t\widetilde{w}^{(0)}(u, \theta_u)\widetilde{I}(u, \theta_u - \theta_n).$$

Therefore representing it in rectangular coordinates again, we obtain

$$\widehat{w}^{(t)}(\mathbf{u}, \mathbf{n}) = \widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(|\mathbf{u}|, 0)^t\widehat{w}^{(0)}(\mathbf{u})\widehat{I}(\mathbf{u} - \mathbf{n}),$$

where

$$\widehat{w}^{(0)}(\mathbf{u}) \stackrel{\text{def}}{=} \int_{|\mathbf{m}|=\sigma} \widehat{w}^{(0)}(\mathbf{u}, \mathbf{m})d\mathbf{m}. \tag{6.10}$$

By computing the Fourier inverse transform, we have the dominant component of $w^{(t)}(\mathbf{x}, \Delta\alpha)$ as follows.

$$w^{(t)}(\mathbf{x}, \Delta\alpha) \approx \int_{-\infty}^{\infty}\int_{|\mathbf{n}|=\sigma} \widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(|\mathbf{u}|, 0)^t\widehat{w}^{(0)}(\mathbf{u})\widehat{I}(\mathbf{u} - \mathbf{n})e^{i\mathbf{u}\mathbf{x}}e^{in\Delta\alpha}d\mathbf{n}d\mathbf{u}.$$

Since $\widehat{\widetilde{I}}(|\mathbf{u}|,0)^t$ grows exponentially as t increases, this function extracts a certain $\mathbf{u}$ such that $|\mathbf{u}| = u_{\max}$, where from Eqn.(6.7),

$$u_{\max} = \arg\max_u \widehat{\widetilde{I}}(u,0) = \arg\max_u e^{-\sigma_I^2(u^2+\sigma^2)/2} I_0(\sigma_I^2 u\sigma).$$

This suggests that we can restrict the integral domain of $\mathbf{u}$ from infinite to $|\mathbf{u}| = u_{\max}$ only. We can numerically verify that when σ_I is small $u_{\max} = 0$ while as σ_I increases $u_{\max}$ converges to σ. Therefore, when σ_I is small, the integral domain can be restricted only to $|\mathbf{u}| = 0$. On the other hand, when σ_I is large enough, we can restrict the integral domain only to $|\mathbf{u}| = \sigma$.

In the following we consider the case of which σ_I is large enough. Then it follows from the above discussion,

$$\begin{aligned}
&w^{(t)}(\mathbf{x},\Delta\alpha)\\
&\approx \int_{|\mathbf{u}|=\sigma}\int_{|\mathbf{n}|=\sigma} \widehat{\widetilde{A}}(0)^{t+1}\widehat{\widetilde{I}}(|\mathbf{u}|,0)^t \widehat{w}^{(0)}(\mathbf{u})\widehat{I}(\mathbf{u}-\mathbf{n})e^{i\mathbf{u}\mathbf{x}}e^{i\mathbf{n}\Delta\alpha}d\mathbf{n}d\mathbf{u}\\
&\propto \int_{|\mathbf{u}|=\sigma}\int_{|\mathbf{n}|=\sigma} \widehat{w}^{(0)}(\mathbf{u})\widehat{I}(\mathbf{u}-\mathbf{n})e^{i\mathbf{u}\mathbf{x}}e^{i\mathbf{n}\Delta\alpha}d\mathbf{n}d\mathbf{u}\\
&= \int_{|\mathbf{u}|=\sigma} \widehat{w}^{(0)}_{\cos}(\mathbf{u})\cos\mathbf{u}\mathbf{x}\int_{|\mathbf{n}|=\sigma}\widehat{I}_{\cos}(\mathbf{u},\mathbf{n})\cos\mathbf{n}\Delta\alpha d\mathbf{n}d\mathbf{u}\\
&\quad + \int_{|\mathbf{u}|=\sigma} \widehat{w}^{(0)}_{\sin}(\mathbf{u})\sin\mathbf{u}\mathbf{x}\int_{|\mathbf{n}|=\sigma}\widehat{I}_{\sin}(\mathbf{u},\mathbf{n})\sin\mathbf{n}\Delta\alpha d\mathbf{n}d\mathbf{u},
\end{aligned} \tag{6.11}$$

where

$$\begin{aligned}
\widehat{I}_{\cos}(\mathbf{u},\mathbf{n}) &\stackrel{\text{def}}{=} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(\mathbf{x})\cos\mathbf{u}\mathbf{x}\cos\mathbf{n}\mathbf{x}d\mathbf{x},\\
\widehat{I}_{\sin}(\mathbf{u},\mathbf{n}) &\stackrel{\text{def}}{=} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(\mathbf{x})\sin\mathbf{u}\mathbf{x}\sin\mathbf{n}\mathbf{x}d\mathbf{x}.
\end{aligned}$$

Now, Eqn. (6.11) explains the detail of the formation of orientation maps, in particular, (1) how the shape of a receptive field for a cell is determined, (2) how the preferred orientation for a cell is determined, and (3) how preferred orientations and phases change on the cortical surface. We examine the structure of Eqn. (6.11) and discuss these points. During the discussion, the roles of functions are clarified and we observe that how those functions contribute to the formation, in other words, how the formation is constrained. Due to the constraints, orientation maps developed by this model obtain some properties. We observe those properties.

First, we consider parts of Eqn. (6.11) defined as follows.

$$\begin{aligned} \mathrm{rf}_{\cos}(\mathbf{u}, \Delta\alpha) &\stackrel{\mathrm{def}}{=} \int_{|\mathbf{n}|=\sigma} \widehat{I}_{\cos}(\mathbf{u}, \mathbf{n}) \cos \mathbf{n}\Delta\alpha d\mathbf{n}, \text{ and} \\ \mathrm{rf}_{\sin}(\mathbf{u}, \Delta\alpha) &\stackrel{\mathrm{def}}{=} \int_{|\mathbf{n}|=\sigma} \widehat{I}_{\sin}(\mathbf{u}, \mathbf{n}) \sin \mathbf{n}\Delta\alpha d\mathbf{n}. \end{aligned} \tag{6.12}$$

When $\mathbf{u}$ is given, these become functions of $\Delta\alpha$. Thus these function determine the shape of receptive fields. $\mathrm{rf}_{\cos}(\mathbf{u}, \Delta\alpha)$ computes a weighted sum of two-dimensional cosine waves of 180 degrees whose spatial frequency is σ, because the integral domain is limited to $|\mathbf{n}| = \sigma$. The coefficient $\widehat{I}_{\cos}(\mathbf{u}, \mathbf{n})$ takes the maximum value when $\mathbf{n} = \mathbf{u}$ and decreases as $\mathbf{n}$ takes apart from $\mathbf{u}$. Thus, the shape of $\mathrm{rf}_{\cos}(\mathbf{u}, \Delta\alpha)$ is tri-lobed with the orientation $\theta = \tan^{-1}(\mathbf{u})$ like upper four figures in Fig.6.2. In a similar way, the shape of $\mathrm{rf}_{\sin}(\mathbf{u}, \Delta\alpha)$ is bi-lobed with the same orientation like lower four figures in Fig.6.2. The sharpness of the orientation tuning is determined by the range of I. Wider I makes the tuning sharper because $\widehat{I}$ becomes narrower. Thus, the preferred orientation is determined by the given $\mathbf{u}$. We then explore how $\mathbf{u}$ is given.

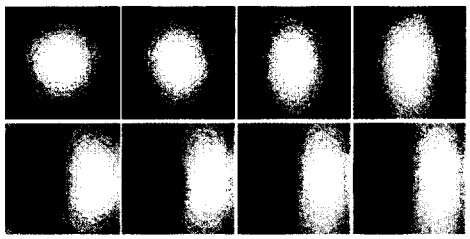

Figure 6.2: Shapes of $\mathrm{rf}_{\cos}(\mathbf{u}, \Delta\alpha)$ and $\mathrm{rf}_{\sin}(\mathbf{u}, \Delta\alpha)$ in Eqn. (6.12). The upper four figures are those of $\mathrm{rf}_{\cos}(\mathbf{u}, \Delta\alpha)$, and from left to right, σ_I^2 values are respectively 0, 10.0, 20.0, 30.0. The lower figures are those of $\mathrm{rf}_{\sin}(\mathbf{u}, \Delta\alpha)$ with the same parameters. $\mathbf{u}$ is taken as a vertical vector, so these figures show the orientation to the vertical direction.

Since σ_I is assumed large enough, $\widehat{I}(\mathbf{u} - \mathbf{n})$ is as a very sharp function, which we may regard as the delta function. From Eqn. (6.11), it follows that

$$\begin{aligned} w^{(t)}(\mathbf{x}, \Delta\alpha) &\propto \int_{|\mathbf{u}|=\sigma} \widehat{w}^{(0)}(\mathbf{u}) e^{i\mathbf{u}(\mathbf{x}+\Delta\alpha)} d\mathbf{u} \\ &= \int_{|\mathbf{u}|=\sigma} \widehat{w}^{(0)}_{\cos}(\mathbf{u}) \cos \mathbf{u}(\mathbf{x} + \Delta\alpha) + \widehat{w}^{(0)}_{\sin}(\mathbf{u}) \sin \mathbf{u}(\mathbf{x} + \Delta\alpha) d\mathbf{u}. \end{aligned} \tag{6.13}$$

This equation is identical to the modification rule of some spectral models of ocular dominance columns [79, 88], and it develops stripes or blobs of the sign of $w^{(t)}(\mathbf{x}, \mathbf{0})$

on the cortical surface, where the sign of $w^{(t)}(\mathbf{x}, \mathbf{0})$ corresponds to the dominance between ON and OFF (Fig.6.3). That is, for any cortical point, the dominant stripe is associated around that point. The orientation of the stripe corresponds to the preferred orientation of the receptive field of the cell on that point, namely, the above $\mathbf{u}$ is given by this stripe.

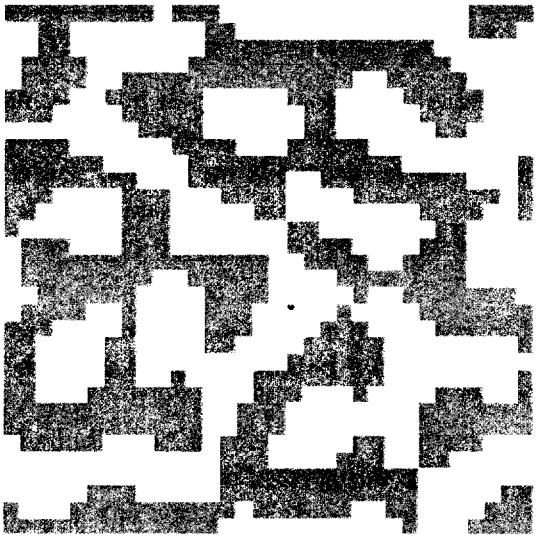

Figure 6.3: Signs of $w^{(t)}(\mathbf{x}, \mathbf{0})$ in Eqn. (6.13). Black and white regions respectively represent negative and positive, which correspond to the dominance between ON and OFF.

Thus for any cortical point $\mathbf{x}$, the preferred orientation of the cell on $\mathbf{x}$ becomes parallel to the orientation of the dominant stripe around $\mathbf{x}$. Moreover, on the cortical surface, the change of the stripes, i.e., the change of $w^{(t)}(\mathbf{x}, \Delta\alpha)$, moving $\mathbf{x}$ and that moving $\Delta\alpha$ in the same direction are identical, so the phase of the receptive field of a cell on $\mathbf{x}$ also becomes identical to the phase of the dominant stripe around $\mathbf{x}$ (Fig.6.4).

Here we summarize the formation of orientation maps and clarify the roles of functions A, I, C to the formation. First, on the cortical surface, stripes are associated. This stripes are developed by function I if σ_I is large enough, and elongate in irregular directions because the initial weights are assigned uniformly at random. Next, for any cortical point, the cell on that point develops its receptive field whose preferred orientation and phase obeys the dominant stripe around that point, as implied by Eqn. (6.13). Finally, the combination of A and I works to develop oriented receptive fields. Both functions extract the DC component of $w^{(0)}(\mathbf{x}, \Delta\alpha)$ with respect of $\Delta\alpha$, what we called the 0th component, and I modifies

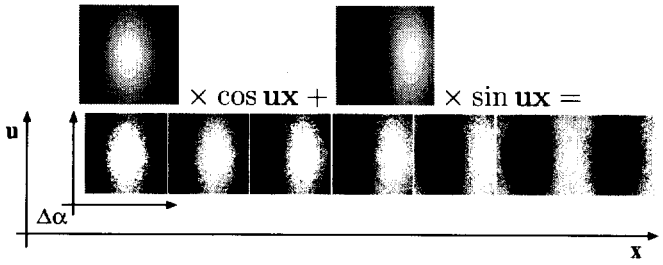

Figure 6.4: The change of the shapes of receptive fields. For a vertical unit vector **u** (i.e., stripes run vertically), the shapes of receptive fields arranged in parallel to the x-axis change, which depend on the position on the x-axis. Phases of these receptive fields also depend on the position.

the DC component to the direction of given **u** in the way of Eqn.(6.12). The spatial frequency of receptive fields is approximated by σ, which is the peak frequency of $\widehat{C}$.

These observations indicate that developed orientation maps obtain the following properties. Since stripes elongate in irregular directions, the preferred orientations smoothly change on the cortical surface, however the change is not periodic. It follows that singular points appear irregularly on the cortical surface. On the other hand, the change of phases becomes periodic.

Finally, it should be noted that this development appears similar to that obtained by Miyashita and Tanaka's thermodynamic model [64]. In their model, all cells on the visual cortex share a layer called "synaptic terminal point", and synaptic terminals on the layer are arranged similar to ocular dominance columns or blobs. Cortical cells are assumed to exist over the layer while keeping the topography and each cell has its "receptive field" on the layer defined by clipping a small region by an isotropic function, e.g., a Gaussian function.

6.3 Computer Simulations

We justify our mathematical analysis by some computer simulations.

We consider 32×32 presynaptic cells and the same number of postsynaptic cells. Each postsynaptic cell receives synaptic connections from 13×13 presynaptic cells with keeping the retinotopy. We make use of Eqn. (6.2) as the modification rule, where parameters are taken from Erwin, Obermayer, & Shulten [23] as follows: $\sigma_A^2 = 4.5$, $\sigma_I^2 = 16.0$, $C_1 = 1.00$, $C_2 = 1/9$, $\sigma_{C1}^2 = 0.61$, and $\sigma_{C2}^2 = 5.45$. On the other hand, we take σ_I^2 large enough as discussed in the previous section. Fig. 2 shows the shapes of $C(\alpha)$ and $\widehat{C}(1)$, and with this setting, σ, the peak frequency

of $\widehat{C}$, is approximately 0.953. Initial synaptic weights are assigned uniformly at random from the range $[-0.4, 0.4]$, and the modification rule is applied 80 times. To avoid finite size effects, the periodic boundary condition is adopted.

Figure6.5 shows the developed orientation map. As can be seen, receptive fields become oriented as shown in Fig.6.2 and the change of phases of receptive fields is similar to Fig.6.4

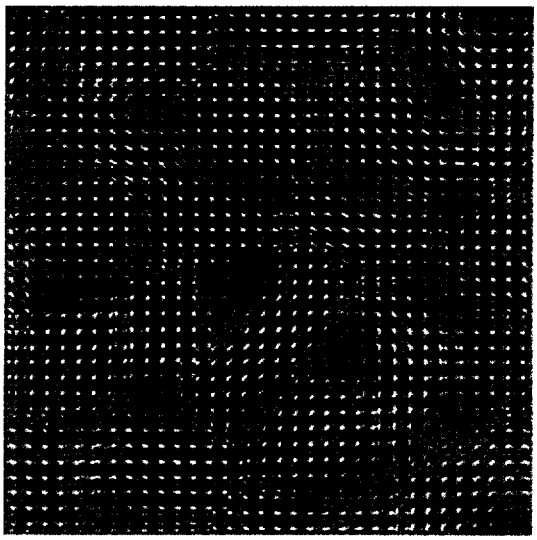

Figure 6.5: Orientation map obtained from a computer simulation. Receptive fields corresponding to each cell are arranged on the cortical surface.

One can calculate the spatial frequency of receptive fields. A calculation shows that the mean and the variance are respectively 0.962 and 0.03. This is a good approximation of $\sigma \approx 0.953$ (Recall that σ is the peak frequency of $\widehat{C}$).

Figure6.6 shows the stripes of $w^{(t)}(\mathbf{x}, \mathbf{0})$ computed by Eqn.(6.13) and preferred orientations extracted from Fig.6.5. As can be observed, preferred orientations become parallel to the stripes and smoothly change on the cortical surface, while the change is not periodic.

6.4 Conclusions

We considered a correlation-based model for the orientation map formation proposed by Miller [60]. We performed the Fourier transform and computed the principal component of the model mathematically. Then using the principal component, we studied how the oriented receptive fields and orientation columns were

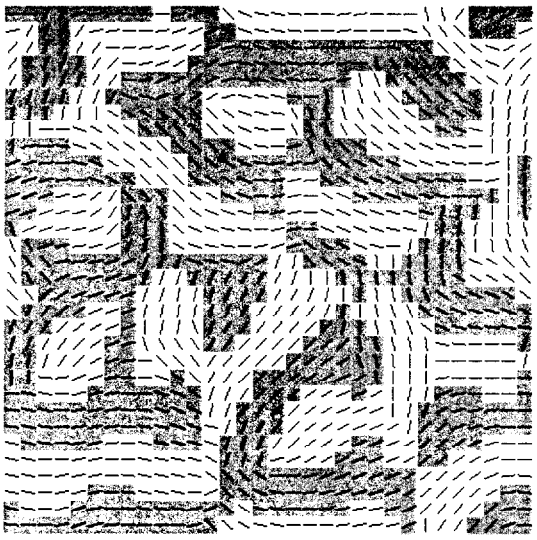

Figure 6.6: Stripes of $w^{(t)}(\mathbf{x}, \mathbf{0})$ computed by Eqn.(6.2) and the preferred orientations extracted from Fig.6.5. A square is associated for each cell position, and an oriented line in each square represents the preferred orientation of the cell at that position while the background color corresponds to the sign of stripes at that position. Grey and white respectively corresponds to negative and positive.

developed. With our analysis, we clarified the roles of functions as follows. (1) A Mexican-hat-type correlation function determines the spatial frequencies of receptive fields; (2) The combination of a Gaussian arbor function and a Gaussian intracortical function develops oriented receptive fields; (3) The intracortical function develops stripes similar to ocular dominance columns on a layer on which each cell has its receptive field if the range is sufficiently wide; (4) The preferred orientations and the phases of receptive fields for each cell are determined by the stripes within the receptive fields. These observations inditicated that this model develops orientation maps with the following properties. (1) It develops oriented receptive fields; (2) Preferred orientations smoothly change on the cortical surface; (3) The periodicity of preferred orientations does not appear; (4) The periodicity of phases appears; (5) Singular points appear irregularly on the cortical surface. Finally our analytical results were justified by computer simulations.

Chapter 7

Details of Computer Simulation

Although the author has performed a number of computer simulations, only few of these appear in this thesis. In particular, there is no result of computer simulations for the development of oriented receptive fields in this thesis. In this chapter we see the details of computer simulations.

First we review the computer environment in which simulations are performed. In the next section, we discuss how to make a computer simulation running faster. We consider two techniques, one is a coding technique using a table while the other is a parallel programming using MPI. Then, in the following two sections, the details of computer simulations for the development of oriented receptive fields and that for the orientation map formation are reviewed. In these sections, we consider the data structure and the algorithm of the synaptic modification in the simulation programs, how to visualize the numerical data, and how simulations are performed. Finally we confirm that the results of computer simulations support our analytical results.

7.1 Environment for Computer Simulations

Almost all computer simulations throughout this thesis were performed on a beowulf cluster [86] developed by the author. The cluster consists of 8 PCs equipped with single Intel Pentium III 1GHz Processor and 512MB Memory and these PCs are connected via 100Base-TX Ethernet in a star topology. The operating system of PCs are Debian GNU/Linux [17]. All simulation programs are written in C using `gcc` [26], and we make use of MPICH library [67] in simulations of the orientation map formation to parallelize the computation. MT [56] is also used to generate pseudo random numbers. Numerical data of synaptic weights are converted into XPM files [109] by the original program written in `perl` [105], and these files are visualized by the original program written in C++ using `g++` and Qt toolkit [13].

The rest of the simulations were performed on computers of several operating systems and architectures including GNU/Linux on DEC Alpha, SunOS 5.6 on SUN Ultra Sparc, GNU/Linux on Intel Pentium and Celeron.

7.2 Techniques for faster computation

In model studies, a mathematical model is proposed and is justified by computer simulations, that is, the plausibility and capability of the model are investigated by computer simulations. Hence a computer simulation plays an important role in model studies. However, along with the model becomes large and complicated, its computer simulation also becomes so large and complicated that it spends much computer resources, in particular, computing time. Thus, it is important to investigate a way to make a computer simulation running faster. In this section, we investigate some general techniques to make a computer simulation program running faster and evaluate these numerically. First we focus on code tuning, in particular, reducing the number of calling math functions in our simulation program of orientation map formation. Then we observe that the running time is drastically improved. Next we expand our programming paradigm to parallel computation. We introduce a parallel programming library called MPI and then we make use of it in the same simulation program. After that we confirm that the running time decreases as the number of available computers increases.

7.2.1 Faster computation via code tuning

A computer program often drastically improves its running time by a couple of changes in its source code. There are several literatures which explains how we can modify a program in algorithmic aspects [3, 4]. However, there are several cases that even a simple transformation is enough. In this subsection, we investigate a technique of reducing the number of floating point arithmetics in the source code. Our computer simulations heavily depend on floating point arithmetics. It is widely known that a floating point arithmetic is much slower that an integer arithmetic [57]. Thus, we can expect that a computer simulation runs much faster by reducing the number of calling math functions in the source code.

This subsection is organized as follows. First we show the code fragment of our simulation program and then we discuss how we can reduce the number of calling math functions. We consider two techniques. The first one is expanding values of a math function into a table. The second one is introducing an additional variable to keep the value of an element of the table in a loop. Next we evaluate these techniques numerically and observe that the first techniques drastically improves the running time while the second techniques also improves slightly more.

Reducing the number of calling math functions

The most time-consumptive part in our simulation program is the following code fragment.

```
for x = (x1, x2) in (1,1) .. (32,32)
  for α = (α1, α2) in (1,1) .. (32,32)
    Δw[x][α] = 0;
    for y = (y1, y2) in (1,1) .. (32,32)
      for β = (β1, β2) in (1,1) .. (32,32)
        Δw[x1][x2][α1][α2] +=
          A(|x − α|)×I(|x − y|)×C(|x − y|)×w[y1][y2][β1][β2]
      done
    done
  done
done
```

As can be seen, functions `A`, `I`, and `C` are called repeatedly, and these are Gaussians and the difference of Gaussians. `exp(3)` is called to compute a Gaussian function and the total number of calls in this code is 32^4. On the other hand, we have noticed that `A`, `I`, `C` are functions of the distance between two points, that is, those values can be computed immediately when two points are given. This suggests that we can expand those values into a table beforehand, and each table consists of a two-dimensional matrix.

From this observation, we can modify the above code as follows.

```
Expand_A(); // Store the value of A into a matrix A.
            // The value of an element A[|x − α|] is that of
            // A(|x − α|)
Expand_I();
Expand_C();
:
for x = (x1, x2) in (1,1) .. (32,32)
  for α = (α1, α2) in (1,1) .. (32,32)
    Δw[x][α] = 0;
    for y = (y1, y2) in (1,1) .. (32,32)
      for β = (β1, β2) in (1,1) .. (32,32)
        Δw[x1][x2][α1][α2] +=
          A[|x − α|]×I[|x − y|]×C[|x − y|]×w[y1][y2][β1][β2]
      done
    done
  done
done
```

Furthermore, we can improve the running time slightly more. It is time-consuming to access an element of a matrix when the dimension of the matrix is greater than 2 [57]. In the above code fragment the value of $\mathtt{A[|x-\alpha|]}$ does not change in the loops of $\mathbf{y}$ and α. In a similar way, the value of $\mathtt{I[|x-y|]}$ does not change in the loop of β. Thus, introducing two variables, we can expect to decrease the number of accesses to a matrix. The modified code is the following.

```
Expand_A(); // Store the value of A into a matrix A.
            // The value of an element A[|x − α|] is that of
            // A(|x − α|)
Expand_I();
Expand_C();
weight valueA, valueI;
:
for x = (x1, x2) in (1,1) .. (32,32)
  for α = (α1, α2) in (1,1) .. (32,32)
    Δw[x][α] = 0;
    valueA = A[|x − α|]
    for y = (y1, y2) in (1,1) .. (32,32)
      valueI = I[|x − y|]
      for β = (β1, β2) in (1,1) .. (32,32)
        Δw[x1][x2][α1][α2] +=
          valueA×valueI×C[|x − y|]×w[y1][y2][β1][β2]
      done
    done
  done
done
```

Numerical Evaluation

We evaluate how these modifications make the simulation faster. We run the simulation program on our beowulf cluster described in the previous chapter five times and measure its average running time of the first five steps including its initialization and output of numerical data. In particular, we measure the user time of the output of UNIX `time(1)`. The following table shows how the running time was improved by the first technique.

the original code	the tuned code	reduced rate
2192s	334s	85%

The original code spent 2192 seconds during the simulation while the tuned code as explained the above spent only 334 seconds. Thus, the tuned code spent only 15%

of the running time of the original code. Therefore we conclude that the running time can be greatly improved by reducing the number of calling math functions.

Then we evaluate the second technique in the similar way explained above. The following table shows the result.

original	1st tuned	2nd tuned	1st reduced rate	2nd reduced rate
2192s	334s	324s	85%	3%

As can be seen, we obtain further improvement of the running speed. Thus, we conclude that this modification also works efficiently although it achieves a small improvement.

7.2.2 Faster computation via parallel programming

Further improvements can be achieved by parallel programming, that is, dividing a program into several components and dealing these with several computers simultaneously. However, we do not want to get into the detail of parallelization. Our concern is to make a computer simulation running faster, not to study a parallel programming. Thus, we want a library or a toolkit which enables us writing a program running in parallel with minimal knowledge of parallel programming.

A solution is using OpenMP [9, 75]. OpenMP automatically parallelize a program without knowledge of parallelization. For example, we consider to parallelize a program which contains the following dual loop.

```
for(i = 0; i < n; i++){
  for(j = 0; j < m; j++){
      x = i / n;
      y = j / m;
      weight[i][j] = updateWeight(x, y);
   }
}
```

This is achieved by simply inserting one pragma (i.e., command) into the head of the loop as follows.

```
#pragma omp parallel for private(j, x, y)
for(i = 0; i < n; i++){
  for(j = 0; j < m; j++){
      x = i / n;
      y = j / m;
      weight[i][j] = updateWeight(x, y);
   }
}
```

Then by compiling the source code, the first loop is expanded so that the program can work in parallel.

Thus, with OpenMP, we do not need to know the detail of parallelization. All the difficult (and standard) procedures to make the program parallelized are automatically performed. However, OpenMP works only on a SMP machine, that is, a computer equipped with several CPUs. Thus, we need a SMP machine to use OpenMP. A cluster with several machines which have single CPU cannot receive benefits of OpenMP.

The other choices are PVM [27,77] and MPI [33,66,76,83]. These libraries work on a cluster of single CPU machines as well as on SMP machines. Although using these requires knowledge of parallel programming and parallelizing a program by hand, the knowledge required is not so much and it is not difficult to parallelize a program if the task is a simple loop decomposition as the above example. Thus this subsection concentrates on the simple loop decomposition using MPI. Although one may imagine which is better between PVM and MPI, they are completely different [34]. For out simulations, we make use of MPI. This subsection is organized as follows. First we explain how to write a program using MPI is by looking a simple sample program called "Hello World". Next we consider a loop decomposition of the simulation program for orientation map formation used to obtain the result in the previous chapter. Then we evaluate the running time with respect to the number of processors. As a result, we make sure that parallelization efficiently works in the case of a simple loop decomposition.

Overview of MPI

MPI is the acronym of "Message Passing Interfaces". The message passing model posits a set of processes that have only local memory but are able to communicate with other processes by sending and receiving messages. It is a defining feature of the message passing model that data transfer from the local memory of one process to the local memory of another requires operations to be performed by both processes.

All processes execute the same program. When starting the execution, an unique number is given to each process which we call "rank", while the toral number of processors are also given. Using these numbers, each process change its behavior in the program. In Fig.7.1, we show a schematic network of processes executing `a.out`.

Hello world

Before getting into the main topic, we first explain how programming with MPI is using a simple sample program "Hello World". The program is shown in Fig. 7.2.

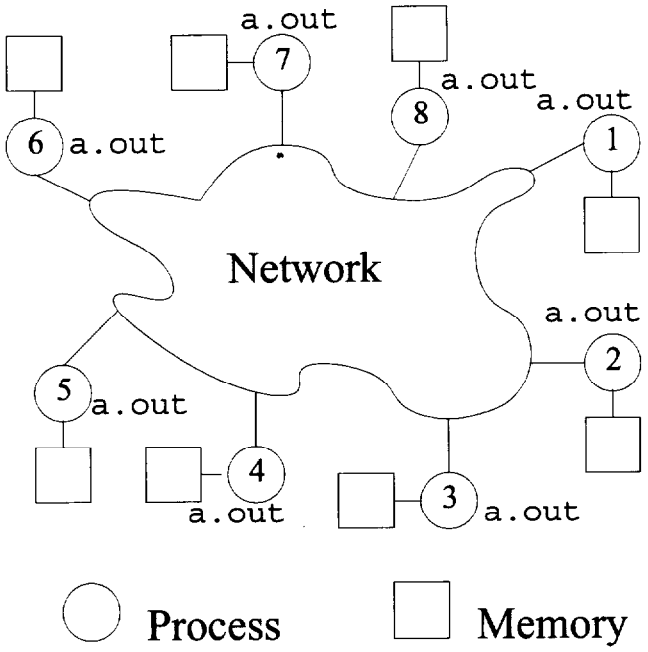

Figure 7.1: Schematic network of processes in the message passing model. A number in a circle represents a rank.

```
#include<stdio.h>
#include<string.h>
#include<mpi.h>

int main(int argc, char *argv[])
{
  int rank, size, node, tag = 0;
  int namelen;
  char name[MPI_MAX_PROCESSOR_NAME];
  char greeting[sizeof(name)+100];
  MPI_Status status;

  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &size);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  MPI_Get_processor_name(name, &namelen);

  sprintf(greeting, "Hello world: rank %d of %d running \
    on node %s.\n", rank, size, name);

  if (rank == 0){
    fputs(greeting, stdout);
    for(node = 1; node < size; node++){
      MPI_Recv(greeting, sizeof(greeting), MPI_BYTE,
           node, tag, MPI_COMM_WORLD, &status);
      fputs(greeting, stdout);
    }
  }
  else{
    MPI_Send(greeting, strlen(greeting)+1, MPI_BYTE,
         0, tag, MPI_COMM_WORLD);
  }

  MPI_Finalize();
  exit(0);
}
```

Figure 7.2: MPI version of "Hello World" program in C.

First of all, all MPI programs must include `mpi.h`. This is the header file of MPI which contains definitions of functions and constants. Then, all MPI programs must be initialized via calling `MPI_Init`. `MPI_Init` must be called first before calling any other functions. The total number of nodes and the rank of the current node can be obtained via `MPI_Comm_size` and `MPI_Comm_rank`. In this example, the total number of nodes is stored in `size` while the rank of the current node is in `rank`. Here `MPI_COMM_WORLD` is so-called a "communicator", which determines a collection of nodes that can send messages to each other. `MPI_Get_processor_name` enables the current node to get the hostname. The hostname and its length are stored respectively in `name` and `namelen`. So far all nodes execute the common stuff.

Then, a node changes its behavior with respect to its rank. All nodes except the one whose rank is 0 sends a message to the node of rank 0 via calling `MPI_Send`. `MPI_Send` requires six arguments, they are, 1. the message body (`greeting`), 2. the length of the message, 3. the type of the message (`MPI_BYTE` represents a byte sequence), 4. the rank of the node to send the message, 5. an integer called "tag" which is associated with the message, and 6. a communicator. On the other hand, the node of rank 0 works as follows. It receives messages from other nodes via calling `MPI_Recv` and output the messages to `stdout`. `MPI_Recv` requires seven arguments, they are, 1. the message body, 2. the length of the message, 3. the type of the message, 4. the rank of the node from which this node receives the message, 5. the tag number associated with the message (i.e., the same value used in `MPI_Recv`), 6. a communicator, and 7. a variable which contains status of the message passing.

Finally, all MPI programs must be finalized via calling `MPI_Finalize`.

The result of running this program is as Fig. 7.3. As can be seen, all nodes say "Hello world" with its hostname and its rank.

```
[tyam@p1 hello_world]$ mpirun -np 8 hello_world
Hello world: rank 0 of 8 running on node p1.
Hello world: rank 1 of 8 running on node p2.
Hello world: rank 2 of 8 running on node p3.
Hello world: rank 3 of 8 running on node p4.
Hello world: rank 4 of 8 running on node p5.
Hello world: rank 5 of 8 running on node p6.
Hello world: rank 6 of 8 running on node p7.
Hello world: rank 7 of 8 running on node p8.
```

Figure 7.3: Result of running "Hello World" program.

Simple loop decomposition

In the simulation of orientation map formation, the most power-consumptive part is updating synaptic weights, that is, computing the following equation.

$$\Delta w^{(t)}(\mathbf{x}, \alpha) = \sum_{\mathbf{y},\beta} A(\mathbf{x}, \alpha) I(\mathbf{x}, \mathbf{y}) C(\alpha, \beta) w^{(t)}(\mathbf{y}, \beta).$$

An implementation of this equation is as follows. As can be seen, it consists of quadruple loop.

```
for x = (x1, x2) in (1,1) .. (32,32)
  for α = (α1, α2) in (1,1) .. (32,32)
    Δw[x][α] = 0;
    for y = (y1, y2) in (1,1) .. (32,32)
      for β = (β1, β2) in (1,1) .. (32,32)
        Δw[x1][x2][α1][α2] +=
          A(|x − α|)×I(|x − y|)×C(|x − y|)×w[y1][y2][β1][β2]
      done
    done
  done
done
```

It is the most effective to decompose the most outside loop. This can be done by performing the same computation on all nodes. At the initialize section, the node of rank 0 reads the seed of a pseudo random number generator from `stdin` and broadcasts the seed to the other nodes. This is enabled via calling `MPI_Bcast` as follows.

```
MPI_Bcast(&seed, 1, MPI_INT, 0, MPI_COMM_WORLD)
```

where arguments are 1. an integer variable which contains the value of the seed, 2. the size of the variable, 3. the type of the seed (`MPI_INT`), 4. the rank of the node broadcasts the seed, and 5. a communicator. The other nodes receives the seed and initialize itself using the seed. Since two copies of a pseudo random number generator with the same seed generate the same sequence of numbers, configurations of all nodes at this point are identical. The seed can be broadcasted via

Then, all nodes move to the loop section. Let `size` and `rank` be the total number of nodes nd the rank of the current node. Then, we decompose the above loop as follows (here we assume that `rank` starts from 1) so that each node computes different $\Delta w[x_1][x_2][\alpha_1][\alpha_2]$.

```
for x = (x1, x2) in (rank, 1) .. (32, 32) skip (size, 1)
  for α = (α1, α2) in (1, 1) .. (32, 32)
    Δw[x][α] = 0;
    for y = (y1, y2) in (1, 1) .. (32, 32)
      for β = (β1, β2) in (1, 1) .. (32, 32)
        Δw[x1][x2][α1][α2] +=
          A(|x − α|)×I(|x − y|)×C(|x − y|)×w[y1][y2][β1][β2]
      done
    done
  done
done
```

For example, assume that the value of `size` is 8. Then, the node of rank 1 computes $\Delta w[1][x_2][\alpha_1][\alpha_2]$, $\Delta w[9][x_2][\alpha_1][\alpha_2]$, $\Delta w[17][x_2][\alpha_1][\alpha_2]$, the node of rank 2 computes $\Delta w[2][x_2][\alpha_1][\alpha_2]$, $\Delta w[10][x_2][\alpha_1][\alpha_2]$, $\Delta w[18][x_2][\alpha_1][\alpha_2]$, In general, the node of rank k computes $\Delta w[k][x_2][\alpha_1][\alpha_2]$, $\Delta w[k+8][x_2][\alpha_1][\alpha_2]$, $\Delta w[k+16][x_2][\alpha_1][\alpha_2]$, By this decomposition, the total number of iterations of this quadruple loop is 1/`size` of the original.

After finishing the loop, all nodes have to exchange its computed values of $\Delta w[x_1][x_2][\alpha_1][\alpha_2]$ with the other nodes and combine these. There is a MPI command which enables such exchange, that is, `MPI_Allreduce`. For the above example, we may use this function as follows.

```
MPI_Allreduce(&Δw, &Δw_reduced, sizeof(Δw), \
              MPI_DOUBLE, MPI_SUM, MPI_COMM_WORLD);
```

Thus, the contents of Δw are combined and stored into Δw`_reduced`. Since Δw is a sparse matrix, this combination can be realized by summing up all Δw. So `MPI_SUM` is specified. Finally, storing the contents of Δw`_reduced` back into Δw, all nodes get computed Δw.

Numerical Evaluation

Here we evaluate how efficient of this loop decomposition. We consider our program for orientation map formation used in the previous chapter and parallelize it as explained so far. Then, we run the program and measure the running time in the same environment described in the previous subsection. While several implementation of MPI are available, we make use of `MPICH` [67] in this evaluation because this implementation is somehow standard. We consider cases of which the number of nodes are 1, 2, 4, and 8. For each case, we run the program five times with different seeds and compute the average running time.

The following table shows the measured running times and the result is plotted as shown in Fig. 7.4.

1 node	2 nodes	4 nodes	8 nodes
2634s	1321s	662s	335s

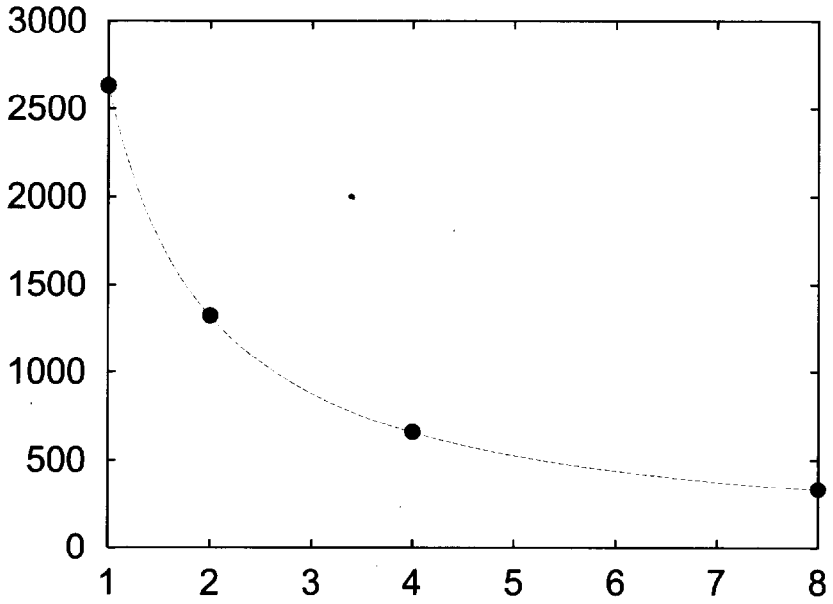

Figure 7.4: Running time of the simulation with respect to the number of nodes.

The x-axis and y-axis respectively represent the number of nodes and the running time of the simulation. Dots represent the running times when the number of nodes are 1, 2, 4, and 8, while the line represents $y = 2634/x$, where 2634 is the user time when one node is available. As can be seen, the user time decreases as the number of nodes increases and the change follows the inverse of the number of nodes almost exactly. In general, the user time does not decrease like this because a parallelization must involve overheads. Thus this result indicates that the original program is so straightforward that it does not show its full potential. We can expect the program running much faster by further modifications. Nevertheless, this result clearly shows that this parallelization works effectively.

7.3 Simulation program for the receptive field development

7.3.1 Data structure

In this simulation we consider one postsynaptic neuron and 13×13 presynaptic cells. Each presynaptic cell projects a synaptic connection to the post synaptic cell and the synaptic weights are modified following Eqn.(5.4) in steps. The synaptic weights are implemented as 13×13 two-dimensional matrix of `double` type since presynaptic cells are arranged on a two-dimensional grid. Thus an element of the matrix represents the synaptic weight of the presynaptic weight on that position on the two-dimensional grid. Therefore let $w[x_1][x_2]$ and $\Delta w[x_1][x_2]$ be two-dimensional matrices which represents $w(\mathbf{x})$ and $\Delta w(\mathbf{x})$ where (x_1, x_2) stands for $\mathbf{x}$, a position on the presynaptic layer.

7.3.2 Algorithm of the synaptic modification

Our algorithm of the synaptic modification is as follows.

```
Input:
  The seed for the pseudo random number generator.

Output:
  The values of w[x1][x2]

Initialize:
  for x = (x1,x2) in (1,1) .. (13,13)
    w[x1][x2] = a number chosen uniformly at random from [-1.0,1.0].
  done

Main loop:
  for t in 1 .. # of simulation steps
    for x = (x1,x2) in (1,1) .. (13,13)
      Δw[x1][x2] = 0
      for y = (y1,y2) in (1,1) .. (13,13)
        Δw[x1][x2] += A(|x|)×C(|x - y|)×w[y1][y2]
      done
    done
    for (x1,x2) in (1,1) .. (13,13)
      w[x1][x2] += Δw[x1][x2]
    done
  done
```

This program is written in C using gcc and random numbers are generated using MT pseudo random number generator in the initialize section.

7.3.3 Visualization of numerical data

After executing the main loop, we obtain the numerical data of the configuration of synaptic weights. This data are then processed by a perl script written by the author and converted to a XPM file. In the perl script, the data are read and values are stored and normalized (i.e., divided by the absolute value of the maximal value). Then the range from -1.0 to 1.0 are separated into 16 subranges and each value is classified into one of these subranges. After that, with respect to its subrange, the color and the contract of the dot corresponding to a value are determined. The color represents the sign of the value (white for positive, black for negative) and the contrast does its absolute value (thick for large, pale for small). Finally, these dots are expressed in XPM format and written in a file. Thus the figure consists of 13×13 grayscale pixels of 16 degrees.

For example, the numerical data in Table.7.1 is converted and the figure in Fig.7.5 is obtained.

-0.06	-0.00	0.06	0.13	0.16	0.15	0.10	0.03	-0.02	-0.04	-0.03	-0.00	0.03
-0.01	0.10	0.22	0.32	0.35	0.30	0.19	0.05	-0.06	-0.12	-0.11	-0.06	0.00
0.08	0.25	0.43	0.55	0.57	0.46	0.26	0.03	-0.15	-0.24	-0.23	-0.15	-0.05
0.18	0.41	0.64	0.78	0.76	0.58	0.29	-0.03	-0.29	-0.42	-0.40	-0.28	-0.13
0.27	0.55	0.81	0.94	0.88	0.63	0.25	-0.15	-0.47	-0.62	-0.59	-0.43	-0.22
0.32	0.62	0.89	1.00	0.90	0.59	0.15	-0.30	-0.65	-0.81	-0.76	-0.55	-0.29
0.33	0.62	0.87	0.95	0.82	0.48	0.01	-0.45	-0.80	-0.94	-0.86	-0.62	-0.33
0.29	0.55	0.76	0.82	0.67	0.32	-0.13	-0.56	-0.88	-0.99	-0.88	-0.62	-0.32
0.21	0.42	0.59	0.63	0.48	0.17	-0.23	-0.61	-0.86	-0.93	-0.80	-0.55	-0.27
0.13	0.28	0.40	0.42	0.30	0.04	-0.27	-0.57	-0.75	-0.77	-0.64	-0.41	-0.18
0.05	0.15	0.23	0.24	0.15	-0.03	-0.25	-0.45	-0.57	-0.55	-0.43	-0.25	-0.08
-0.00	0.05	0.10	0.11	0.06	-0.05	-0.18	-0.30	-0.35	-0.32	-0.22	-0.10	0.00
-0.03	-0.00	0.03	0.04	0.02	-0.04	-0.10	-0.15	-0.16	-0.13	-0.07	0.00	0.05

Table 7.1: Example of numerical data.

Figure 7.5: Computed receptive field from the data in Table7.1.

7.3.4 Simulation settings

Since we consider 13 × 13 presynaptic cells, we thus fix the arbor radius 6 in the Step arbor function case. On the other hand, we make use of several σ_A in the case of Gaussian arbor function. We change the setting of the correlation function, namely, the peak spatial frequency σ. For each setting of the correlation function, We perform computer simulations several (five, typically) times with different random seeds. The number of simulation steps is 80.

We consider several settings of the correlation function, some of these are described as follows. When $\sigma = 0.0$ with $K_1 = 1.0$ and $K_2 = 0.0$ (i.e., Gaussian correlation function), a centre-surround receptive field is developed (Fig.7.6 left). When $\sigma = 0.43$ with $K_1 = 1.0$, $K_2 = 0.1111$, $\sigma_1 = 3.0$, and $\sigma_2 = 27.0$, a bi-lobed receptive is developed (Fig.7.6 center). When $\sigma = 1.0$ with $K_1 = 1.0$, $K_2 = 0.1111$, $\sigma_1 = 0.5$, and $\sigma_2 = 5.0$, a quadre-lobed receptive field is developed (Fig.7.6 right). These results support our analytical results.

Figure 7.6: Examples of developed receptive fields under some settings.

7.4 Simulation program for the orientation map formation

7.4.1 Data structure

This time we consider 32 × 32 cortical neurons and the same number of LGN cells. We assume that all LGN cells project synaptic connections to each cortical neuron for simplifying the data structure, although only connections from LGN cells within its arbor are modified in fact. We thus represent synaptic weights of all cortical neurons by a four-dimensional matrix $w[x_1][x_2][\alpha_1][\alpha_2]$, where $\mathbf{x} = (x_1, x_2)$ stands for the position of the cortical layer where $\alpha = (\alpha_1, \alpha_2)$ does for that of the LGN layer. This is apparently memory-consuming, however, we do not care of it because our computers are equipped with sufficient amount of the memory. In a similar way, let $\Delta w[x_1][x_2][\alpha_1][\alpha_2]$ be the matrix which contains $\Delta w(\mathbf{x}, \alpha)$ at each step in the simulation.

7.4.2 Algorithm of the synaptic modification

The algorithm of the synaptic modification is similar to that for the receptive field development. One different point is that one more loop for each cortical neuron is included.

```
Input:
  The seed for the pseudo random number generator.

Output:
  The values of w[x1][x2][α1][α2]

Initialize:
  for x = (x1, x2) in (1, 1) .. (32, 32)
    for α = (α1, α2) in (1, 1) .. (32, 32)
      w[x1][x2][α1][α2] = A(|x − α|)
        × a number chosen uniformly at random from [−1.0, 1.0].
      done
    done

Main loop:
  for t in 1 .. # of simulation steps
    for x = (x1, x2) in (1, 1) .. (32, 32)
      for α = (α1, α2) in (1, 1) .. (32, 32)
        Δw[x1][x2][α1][α2] = 0
        for y = (y1, y2) in (1, 1) .. (32, 32)
          for β = (β1, β2) in (1, 1) .. (32, 32)
            Δw[x1][x2][α1][α2] +=
              A(|x − α|)×I(|x − y|)×C(|x − y|)×w[y1][y2][β1][β2]
          done
        done
      done
    done
    for x = (x1, x2) in (1, 1) .. (32, 32)
      for α = (α1, α2) in (1, 1) .. (32, 32)
        w[x1][x2][α1][α2] += Δw[x1][x2][α1][α2]
      done
    done
  done
```

This program is also written in C using `gcc` and random numbers are generated using MT pseudo random number generator in the initialize section.

7.4.3 Visualization of numerical data

When the execution of the main loop is finished, we obtain 32 × 32 numerical data of the configurations of synaptic weights which represent receptive fields of the cortical neurons. For each data of each cortical neuron, we run the same perl script used in the simulation of the receptive field development and we obtain 32 × 32 XPM files. Here it should be noted that we do not visualize all 32 × 32 synaptic weights for each cortical neuron. We extract only 13×13 synaptic weights on the center from the numerical data and visualize these. Thus the each figure of a receptive field consists of 13 × 13 pixels as the same as the previous simulation.

After obtaining all 32 × 32 figures, these are arranged on a two-dimensional window and displayed on X Window System [108]. We thus obtain the figure of the orientation map, i.e., the arrangement of oriented receptive fields as shown in Fig.6.5. The author wrote a program which displays the window in C++ using g++ and Qt toolkit [13]. Qt is a C++ toolkit which enables us to easily write a GUI program on X Window System as well as on Microsoft Windows.

Meanwhile from the numerical data, we extract preferred orientations of receptive fields. For each cortical neuron, its preferred orientation is extracted from its configuration of synaptic weights and visualized. The extraction is performed by the original program written by the author in C, which performs the two-dimensional Fourier transform. On the other hand the visualization is done by the perl script. After the all extractions, figures are arranged on a two-dimensional window and displayed in the similar way. Fig.6.6 shows an example.

7.4.4 Simulation settings

A typical setting of this simulation is $\sigma_A^2 = 4.5$, $\sigma_I^2 = 16.0$, $C_1 = 1.00$, $C_2 = 1/9$, $\sigma_{C1}^2 = 0.61$, and $\sigma_{C2}^2 = 5.45$, as described in the previous chapter. In addition to this setting, the author have tried several settings.

First the author tried narrower cortical interaction function defined by $\sigma_I^2 = 4.0$. The author found that the stripes on the cortical layer does not appear and therefore almost cortical neurons developed center-surround receptive fields.

The author also investigated wider correlation function defined by $C_1 = 1.00$, $C_2 = 0.11111$, $\sigma_{C1}^2 = 3.0$, and $\sigma_{C2}^2 = 27.0$. We found that; (1) the width of stripes in receptive fields became wider; and (2) the width of stripes on the cortical layer also became wider. These results suggest our analytical results. Under the above setting of the correlation function, the peak frequency of $\widehat{C}$ is about 0.43, while that under the typical setting is about 0.98. Since our analysis shows that the peak frequency represents the spatial frequency and lower spatial frequency develops wider stripes, the width of stripes in receptive fields become wider. On the other hand, since out analysis suggests that the value of the spatial frequency also converges to that of

the peak frequency, the width of stripes on the cortical layer also become wider.

Moreover, the author have studied the wider arbor function defined by $\sigma_A = 8.0$. We observed that the result did not change. It thus suggests that the change of the arbor function does not affect the formation of the orientation map, which is consistent with out analysis.

Finally for each parameter setting, we performed four or eight simulations with different random seeds.

7.5 Conclusions

In this chapter we focused on computer simulation settings performed throughout this thesis. First we reviewed the hardware specification in which all simulations were performed, and we discussed some general coding techniques for faster computation, they are, the table expansion technique and the parallel programming. Next we presented the details of computer simulations for the development of oriented receptive fields and that for the orientation map formation. In these sections, we introduced the data structure and the algorithm of the synaptic modification in the simulation programs, how to visualize the numerical data, and how simulations were performed. Finally we confirmed that the results of computer simulations supported our analytical results.

Chapter 8

Conclusions

We briefly summarize the results presented in this thesis with some future works. We mathematically analyzed some neural network models to reveal its underlying mechanisms and processes of the self-organization which are necessary for the emergence of orientation selectivity.

In Chapter 4, we considered a stochastic process of the self-organization in von der Malsburg's model. In particular, we focused on a constraint for the synaptic weight modification and studied its role for the stability of the self-organization. We proposed a probabilistic game called the monopolist game which represents the stochastic process of the weight modification. The monopolist game consisted of players each corresponding to an orientation and a player was initially given an amount of money corresponding the strength of the synaptic connection. The amount of money changed by following a game rule, and a player lost all money was ruled out. The goal of this game was to survive with much money until the other players were ruled out, which corresponded to that a preferred orientation was determined with a sufficient connection strength. Thus the dynamics of Malsburg's model could be expressed in terms of random walks. Analyses of the random walks stated that a Malsburg-type normalization surely enabled that the monopolist emerged. Besides, we also calculated the average number of steps needed until the monopolist emerged.

In Chapter 5, we studied how a correlation-based mechanism enabled emergence of orientation selectivity for one single neuron. We mathematically analyzed Linsker's model, in particular, its development in a higher layer, and could obtained the solution in terms of the summation of the Fourier components. Then we investigated the fastest growing component. We could clarify the role of the correlation function, specifically it determined the spatial frequency of the receptive field. Moreover, we focused on the importance of the arbor function, no attention had been caught to the arbor function so far. We showed that the arbor function determined the structure of the receptive field, in particular, a Gaussian

arbor function developed a center-surround receptive field while a Step arbor function enabled the development of a multi-lobed receptive field in addition to the center-surround one.

In Chapter 6, we studied a correlation-based model for many neurons with lateral connections proposed by Miller for the orientation map formation. We also attempted to obtain its mathematical solution in terms of the Fourier components, and could obtain the description of the fastest growing component. Investigating the description revealed the formation of the orientation map. We could clarify the role of the lateral connections. It developed stripes or blobs on the cortical surface like ocular dominance columns and cortical neurons determined preferred orientations to becoming parallel to the stripes around them. Furthermore, we found that the lateral connections elongated the shape of a receptive field, thus a Gaussian arbor function could develop oriented receptive fields in this model, although it developed unoriented receptive fields in Linsker's model.

In this thesis, we concentrated on correlation-based models. An important future work is to investigate a model based on another mechanism as shown in Table 3.1. Another, and more important, future work is to clarify the relationships between models based on different mechanisms.

For example, Erwin et al reported that low-dimensional SOM successfully reproduce both orientation columns and ocular dominance columns simultaneously [23], in particular, the resulting joint maps are more similar to the biologically observed maps than those obtained by correlation-based models. Thus, if we analyze the SOM models and can obtain the essential mechanism why SOM works so effectively, then we may include the mechanism to correlation-based models so that they also develop better joint maps. Since SOM is a stochastic model, we need a mathematical tool to analyze the stochastic process. Our monopolist game and the demonstrated techniques for the analysis may work for this issue.

For another example, high-dimensional SOM is a model between low-dimensional SOM and correlation-based models, because while high-dimensional SOM is based on Winner-Take-All mechanism, the inputs are similar to the one used by correlation-based models. Thus by a similar analytical technique demonstrated for Miller's model, we may analyze high-dimensional SOM. Then it may clarify the difference between low-dimensional SOM and correlation-based models.

In summary, mathematical analysis should help us developing an essentially new and better mathematical model and give us a comprehensive perspective for model study. Moreover it may enable us to feedback theoretical results to physiology.

Bibliography

[1] ALBUS, K. 14c-deoxyglucose mapping of orientation subunits in the cat's visual cortical areas. *Experimental Brain Research 24* (1979), 181–202.

[2] ARBIB, M. A., ET AL. *Neural Organization: structure, function, and dynamics.* A Bradford Book, 1998, p. 201.

[3] BENTLEY, J. *Programming Pearls.* Addison-Wesley, Reading, Mass., 1986.

[4] BENTLEY, J. *More Programming Pearls: Confessions of a Coder.* Addison-Wesley, Reading, Mass., 1988.

[5] BLAKEMORE, C., AND COOPER, G. F. Development of the brain depends on the visual environment. *Nature 228* (1970), 477–478.

[6] BLASDEL, G. G., OBERMAYER, K., AND KIORPES, L. Organization of ocular dominance and orientation columnsin the striate cortex of neonatal macaque monkeys. *Visual Neuroscience 12* (1995), 589–603.

[7] BLASDEL, G. G., AND SALAMA, G. Voltage sensitive dyes reveal a modular organization in monkey striate cortex. *Nature 321* (1986), 579–585.

[8] BONHOEFFER, T., AND GRINVALD, A. Orientation columns in cat are organized in pin-wheel like pattern. *Nature 364* (1991), 144–146.

[9] CHANDRA, R., ET AL. *Parallel Programming in OpenMP.* Morgan Kaufmann Publishers, 2001.

[10] CHAPMAN, B., ZAHS, K. R., AND STRYKER, M. P. Relation of cortical cell orientation selectivity to alignment of receptive fields of the geniculocortical afferents that arborize within a single orientation column in feret visual cortex. *Journal of Neuroscience 13* (1991), 1347 1358.

[11] CHURCHLAND, P. S., AND SEJNOWSKI, T. J. *The computational brain.* The MIT Press, 1994.

[12] CLAIR, M. C., GILLESPIE, D. C., AND STRYKER, M. P. The role of visual experience in the development of columns in cat visual cortex. *Science 279* (1998), 566–570.

[13] DALHEIMER, M. K. *Programming Qt*, 2nd ed. O'reilly and Associates, 2002. http://www.trolltech.com/.

[14] DAUGMAN, J. G. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research 20* (1980), 847–856.

[15] DAUGMAN, J. G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A/2* (1985), 1160–1169.

[16] DEANGELIS, G., GHOSE, G., OHZAWA, I., AND FREEMAN, R. Functional micro-organization of primary visual cortex: Receptive field analysis of nearby neurons. *Journal of Neuroscience 19*, 10 (1999), 4046–4064.

[17] Debian GNU/linux - the universal operationg system. http://www.debian.org/.

[18] DIAO, Y., ET AL. Functional organization of the cortical 17/18 border region in the cat. *Experimental Brain Research 79* (1990), 271–282.

[19] DOMINGO, C., WATANABE, O., AND YAMAZAKI, T. A role of constraint in self-organization. In *Randomization and Approximation Techniques in Computer Science* (1998), M. Luby et al., Eds., vol. 1518 of *Lecture Notes in Computer Science*, pp. 307–318.

[20] DOWLING, J. E. *The retina: an approachable part of the brain.* The Belknap Press of Harvard University Press, 1987.

[21] DURBIN, R., AND MITCHISON, G. A dimension reduction framework for understanding cortical maps. *Nature 343* (1990), 341–344.

[22] ERWIN, E., AND MILLER, K. D. Correlation-based development of ocularly-matched orientation and ocular dominance maps: Determination of required input activities. *Journal of Neuroscience 18*, 23 (1998), 9870–9895.

[23] ERWIN, E., OBERMAYER, K., AND SHULTEN, K. Models of orientation and ocular dominance columns in the visual cortex: a critical comparison. *Neural Computation 7* (1995), 425–468.

[24] FELLER, W. *An Introduction to Probability Theory and its Applications*, third ed., vol. I. John Wiley, 1968.

[25] FENG, J., PAN, H., AND ROYCHOWDHURY, V. P. Linsker-type hebbian learning: A qualitative analysis on the parameter space. *Neural Networks 10*, 4 (1997), 705–720.

[26] GCC home page. http://gcc.gnu.org/.

[27] GEIST, A., ET AL. PVM: Parallel virtual machine. a users' guide and tutorial for networked parallel computing, 1994.

[28] GÖDECKE, I., AND BONHOEFFER, T. Development of identical orientation maps for two eyes without common visual experience. *Nature 379*, 18 (1996), 251–254.

[29] GOLDBERG, D. H., SHOUVAL, H., AND COOPER, L. N. Lateral connectivity as a scaffold for developing orientation preference maps. *Neurocomputing 26* (1999), 381–387.

[30] GOODHILL, G. J., AND CIMPONERIU, A. Analysis of the elastic net model applied to the formation of ocular dominance and orientation columns. *Network 11* (2000), 153–168.

[31] GOODHILL, G. J., AND WILLSHAW, D. J. Application of the elastic net algorithm to the formation of ocular dominance stripes. *Network 1* (1990), 41–59.

[32] GRADSHTEYN, I. S., RYZHIK, I. M., AND JEFFREY, A., Eds. *Table of Integrals, Series, and Products*, 6th ed. Academic Pr, 2000.

[33] GROPP, W., ET AL. *MPI — The Complete Reference*, vol. 2, The MPI-2 Extensions. The MIT Press, 1998.

[34] GROPP, W., AND LUSK, E. L. Why are PVM and MPI so different? In *PVM/MPI* (1997), pp. 3–10.

[35] HUBEL, D. H., WIESEL, T. H., AND LEVAY, S. Plasticity of ocular dominance columns in monkey striate cortex. *Philosofy of Transactions Royal Society of London 278* (1977), 377–409.

[36] HUBEL, D. H., AND WIESEL, T. N. Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology 148* (1959), 574–591.

[37] HUBEL, D. H., AND WIESEL, T. N. Integrative action in the cat's lateral geniculate body. *Journal of Physiology 155* (1961), 385–398.

[38] HUBEL, D. H., AND WIESEL, T. N. Receptive fields, biocular interaction, and functional architecture of the cat striate cortex. *Journal of Physiology 160* (1962), 106–154.

[39] HUBEL, D. H., AND WIESEL, T. N. Receptive fields of cells in striate cortex of very young, visually inexperienced kittens. *Journal of Neurophysiology 26* (1963), 994–1002.

[40] HUBEL, D. H., AND WIESEL, T. N. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology 195* (1968), 215–243.

[41] HUBEL, D. H., AND WIESEL, T. N. Sequence regularity and geometry of orientation columns in the macaque striate cortex. *Journal of Computational Neurology 158* (1974), 267–294.

[42] HUBEL, D. H., AND WIESEL, T. N. Functional architecture of macaque monkey visual cortex. In *Proceeding of Royal Society of London* (1977), vol. 198 of *B*, pp. 1–59.

[43] JAPANESE STANDARDS ASSOCIATION, Ed. *Statistical Tables and Formulas with Computer Applications*, vol. JSA-1972. Japanese Standards Association, 1972.

[44] JONES, J. P., STEPNOSKI, A., AND PALMER, L. A. The two-dimensional spectral structure of simple receptive fields in cat striate cortex. *Journal of Neurophysiology 58*, 6 (1987), 1212–1232.

[45] KAMMEN, D. M., AND YUILLE, A. L. Spontaneous symmetry-breaking energy functions and the emergence of orientation selective cortical cells. *Biological Cybernetics 59* (1988), 23–31.

[46] KOHONEN, T., HUANG, T. S., AND SCHROEDER, M. R., Eds. *Self-Organizing Maps*, 3rd ed., vol. 30 of *Springer Series in Information Science*. Springer, 2001.

[47] LEVAY, S., ET AL. The complete pattern of ocular dominance stripes in the striate cortex and visual field of the macaque monkey. *Journal of Neuroscience 5* (1985), 486–501.

[48] LEVAY, S., WIESEL, T. N., AND HUBEL, D. H. The development of ocular dominance columns in normal and visually deprived monkeys. *Journal of Computational Neurology 191* (1980), 1–51.

[49] LINSKER, R. From basic network principles to neural architecture: Emergence of orientation columns. In *Proceedings of the National Academy of Science USA* (1986), vol. 83, pp. 8779–8783.

[50] LINSKER, R. From basic network principles to neural architecture: Emergence of orientation-selective cells. In *Proceedings of the National Academy of Science USA* (1986), vol. 83, pp. 8390–8394.

[51] LINSKER, R. From basic network principles to neural architecture: Emergence of spatial-opponent cells. In *Proceedings of the National Academy of Science USA* (1986), vol. 83, pp. 7508–7512.

[52] LINSKER, R. Self-organization in a perceptual system: How network models and information theory may shed light on neural organization. In *Connectionist Modeling and Brain Function*, S. Hanson and C. Olson, Eds. The MIT Press, 1990, pp. 351–392.

[53] LÖWEL, S., FREEMAN, B., AND SINGER, W. Topographic organization of the orientation column system in large flat-mounts of the cat visual cortex: a 2-deoxyglucose study. *Journal of Computational Neurology 255* (1987), 401–415.

[54] MACKAY, D. J. C., AND MILLER, K. D. Analysis of linsker's application of hebbian rules to linear networks. *Network 1* (1990), 257–297.

[55] MARČELJA, S. Mathematical description of the responce of simple cortical cells. *Journal of the Optical Society of America 70*, 11 (1980), 1297–1300.

[56] MATSUMOTO, M., AND NISHIMURA, T. Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation 8*, 3 (1998), 3–30. http://www.math.keio.ac.jp/~matumoto/emt.html.

[57] MCCONNELL, S. C. *Code Complete : A Practical Handbook of Software Construction.* Microsoft Press, 1993.

[58] MILLER, K. Development of orientation columns via competition between on- and off-center inputs. *Neuro Report 3* (1992), 73–79.

[59] MILLER, K. D. Derivation of linear hebbian equations from a nonlinear hebbian model of synaptic plasticity. *Neural Computation 2* (1990), 321–333.

[60] MILLER, K. D. A model for the development of simple cell receptive fields and the ordered arrangement of orientation columns through activity-dependent competition between on- and off-center inputs. *Journal of Neuroscience 14*, 1 (1994), 409–441.

[61] MILLER, K. D., KELLER, J. B., AND STRYKER, M. P. Ocular dominance column development: Analysis and simulation. *Science 245* (1989), 605–615.

[62] MILLER, K. D., AND MACKAY, D. J. C. The role of constraints in hebbian learning. *Neural Computation 6* (1994), 100–126.

[63] MIYASHITA, M., KIM, D.-S., AND TANAKA, S. Cortical direction selectivity without directional experience. *Neuro Report 8*, 5 (1997), 1187–1191.

[64] MIYASHITA, M., AND TANAKA, S. A mathematical model for the self-organization of orientation columns in visual cortex. *NeuroReport 3* (1992), 69–72.

[65] MOWER, G. D., ET AL. Dark rearing prolongs physiological but not anatomical plasticity of the cat visual cortex. *Journal of Computational Neurology 235* (1985), 448–466.

[66] Message passing interface (MPI) forum home page. http://www.mpi-forum.org/.

[67] MPICH - a portable implementation. http://www-unix.mcs.anl.gov/mpi/mpich/.

[68] NAKAGAMA, H. *Study on the self-organization of neural networks in the visual cortex.* PhD thesis, University of Tokyo, 2001. (in Japanese).

[69] NAKAGAMA, H., SAITO, T., AND TANAKA, S. Effect of imbalance in activities between on- and off-center lgn cells on orientation map formation. *Biological Cybernetics 83* (2000), 85–92.

[70] NICHOLLS, J. G., MARTIN, A. R., AND WALLACE, B. G. *From neuron to brain*, third ed. Synauer Associates, 1992.

[71] OBERMAYER, K., AND BLASDEL, G. G. Geometry of orientation and ocular dominance columns in monkey striate cortex. *Journal of Neuroscience 13* (1993), 4114–4129.

[72] OBERMAYER, K., AND BLASDEL, G. G. Singularities in primate orientation maps. *Neural Computation 9* (1997), 555–575.

[73] OBERMAYER, K., BLASDEL, G. G., AND SCHULTEN, K. Statistical-mechanical analysis of self-organization and pattern formation during the development of visual maps. *Physical Review A 45*, 10 (1992), 7568–7589.

[74] OBERMAYER, K., RITTER, H., AND SCHULTEN, K. J. A model for the development of the spatial structure of retinotopic maps and orientation columns. *IEICE Trans. Fundamentals E75-A*, 5 (1992), 537–545.

[75] OpenMP specifications. http://www.openmp.org/specs/.

[76] PACHECO, P. S. *Parallel Programming with MPI.* Morgan Kaufmann Publishers, 1997.

[77] PVM: Parallel virtual machine. http://www.epm.ornl.gov/pvm/pvm_home.html.

[78] RIESENHUBER, M., ET AL. Breaking rotational symmetry in a self-organizing map model for orientation map development. *Neural Computation 10* (1998), 717–730.

[79] ROJER, A. S., AND SCHWARTZ, E. L. Cat and monkey cortical columnar patters modeled by bandpass-filtered 2d white noise. *Biological Cybernetics 62* (1990), 381–391.

[80] SHATZ, C. J., AND STRYKER, M. P. Ocular dominance columns in layer iv of the cat's visual cortex and the effects of monocular deprivation. *Journal of Physiolosy 281* (1978), 267–283.

[81] SHOUVAL, H., AND COOPER, L. N. Organization of receptive fields in networks with hebbian learning: the connection between synaptic and phenomenological models. *Biological Cybernetics 74* (1996), 439–447.

[82] SILLITO, A. M. The contribution of inhibitory mechanisms to the receptive field properties of neurones inthe striate cortex of the cat. *Journal of Physiology (London) 250* (1975), 305–329.

[83] SNIR, M., ET AL. *MPI — The Complete Reference*, 2nd ed., vol. 1, The Core. The MIT Press, 1998.

[84] SOMERS, D. C., NELSON, S. B., AND SUR, M. An emergent model of orientation selectivity in cat visual cortical cells. *Journal of Neuroscience 15*, 8 (1995), 5448–5465.

[85] SOMPOLINSKY, H., AND SHAPLEY, R. New perspectives on the mechanisms for orientation selectivity. *Current Opinion in Neurobiology 7* (1997), 514–522.

[86] STEERING, T. L., ET AL. *How to Build a Beowulf (Scientific and Engineering Computation).* The MIT Press, 1999.

[87] STRYKER, M. P., AND HARRIS, W. A. Binocular impulse blockade prevents the formation of ocular dominance columns in cat visual cortex. *Journal of Neuroscience 6* (1986), 2117–2133.

[88] SWINDALE, N. V. A model for the formation of ocular dominance stripes. In *Proceedings of the Royal Society of London* (1980), vol. 208 of *B*, pp. 243–264.

[89] SWINDALE, N. V. Absence of ocular dominance patches in dark reared cats. *Nature 290* (1981), 332–333.

[90] SWINDALE, N. V. A model for the formation of orientation columns. In *Proceedings of the Royal Society of London* (1982), vol. 215 of *B*, pp. 211–230.

[91] SWINDALE, N. V. Role of visual experience in promoting segregation of eye dominance patches in the visual cortex of the cat. *Journal of Computational Neurology 267* (1988), 472–488.

[92] SWINDALE, N. V. A model for the coordinated development of columnar systems in primary visual cortex. *Biological Cybernetics 66* (1992), 217–230.

[93] SWINDALE, N. V. A development of topography in the visual cortex: a review of models. *Network 7* (1996), 161–247.

[94] TANAKA, K. Cross-correlation analysis of geniculostriate neuronal relationships in cats. *Journal of Physiology 49* (1983), 1303–1318.

[95] TANAKA, S. Theory of self-organization of cortical maps: mathematical framework. *Neural Networks 3* (1990), 625–640.

[96] TANAKA, S. Phase transition theory for abnormal ocular dominance column formation. *Biological Cybernetics 65* (1991), 91–98.

[97] TANAKA, S. Theory of ocular dominance column formations: Mathematical basis and computer simulation. *Biological Cybernetics 64* (1991), 263–272.

[98] TANAKA, S. Information representation and self-organization of the primary visual cortex. In *Neural & artificial parallel computation: proceedings of the fifth NEC research symposium* (1996), D. L. Waltz, Ed., pp. 93–126.

[99] TANAKA, S., AND SHINBATA, H. A mathematical model for neuronal responce properties and modular organization in the motion-processing area of the primate visual cortex. *NEC Research and Development 35*, 1 (January 1993), 1–11.

[100] TANG, D. S. Information-theoretic solutions to early visual information processing: Analytic results. *Physical Review A 40*, 11 (1989), 6626–6635.

[101] TSUMOTO, T., ECKART, W., AND CREUTZFELDT, O. D. Modification of orientation sensitivity of cat visual cortex neurons by removal of GABA-mediated inhibition. *Experimental Brain Research 34* (1979), 351–363.

[102] VIDYASAGAR, T. R., PEI, X., AND VOLGUSHEV, M. Multiple mechanisms underlying the orientation selectivity of visual cortical neurones. *Trends in Neuroscience 19* (1996), 272–277.

[103] VON DER MALSBURG, C. Self-organization of orientation sensitivity cells in the striate cortex. *Kybernetik 14* (1973), 85–100.

[104] VON DER MALSBURG, C., AND WILLSHAW, D. J. A mechanism for producing continuous neural mappings: ocularity dominance stripes and ordered retino-tectal projections. *Experimental Brain Research 1* (1976), 463–469.

[105] WALL, L., CHRISTIANSEN, T., AND ORWANT, J. *Programming Perl*, 3rd ed. O'reilly and Associates, 2000. http://www.perl.com/.

[106] WILLSHAW, D. J., AND VON DER MALSBURG, C. How patterned neural connections can be set up by self-organization. In *Proceeding of Royal Society of London* (1976), vol. 194 of *B*, pp. 431–445.

[107] WIMBAUER, S., GERSTNER, W., AND VAN HEMMEN, J. L. Analysis of a correlation-based model for the development of orientation-selective receptive fields in the visual cortex. *Network 9* (1998), 449–466.

[108] The XFree86 project. http://www.xfree86.org/.

[109] The XPM format and library. http://koala.ilog.fr/lehors/xpm.html.

[110] YAMAZAKI, T. Mathematical analysis of a correlation-based model for orientation map formation. In *International Conference on Artificial Neural Networks* (2001), G. Dorffner et al., Eds., vol. 2130 of *Lecture Notes in Computer Science*, pp. 1121–1128.

[111] YAMAZAKI, T. Mathematical analysis of a correlation-based model for orientation map formation. *submitted* (2001).

[112] YAMAZAKI, T. A mathematical analysis of the development of oriented receptive fields in Linsker's model. *Neural Networks 15*, 2 (2002), 125–131.