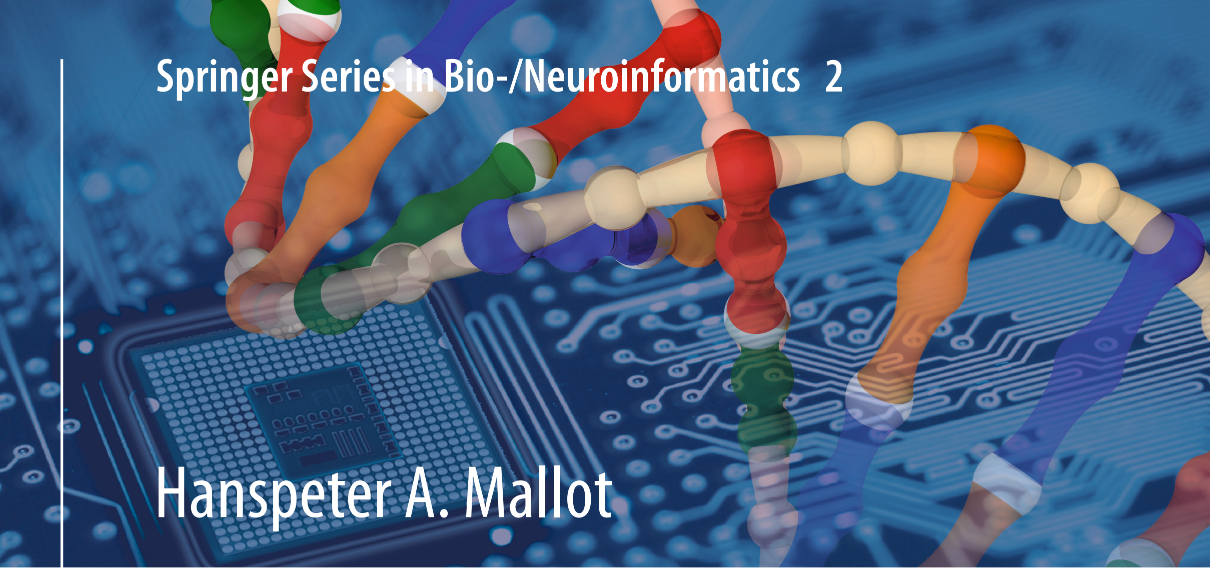


Springer Series in Bio-/Neuroinformatics 2

Hanspeter A. Mallot

Computational Neuroscience

A First Course

Springer

Springer Series in Bio-/Neuroinformatics

Volume 2

Series Editor

N. Kasabov

For further volumes:
http://www.springer.com/series/10088

Hanspeter A. Mallot

Computational Neuroscience

A First Course

Springer

Hanspeter A. Mallot
Department of Biology
University of Tübingen
Tübingen
Germany

ISSN 2193-9349 ISSN 2193-9357 (electronic)
ISBN 978-3-319-00860-8 ISBN 978-3-319-00861-5 (eBook)
DOI 10.1007/978-3-319-00861-5
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013939742

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Computational approaches have been an ever increasing trend in neuroscience research since the publication of Norbert Wiener's seminal book in 1948. Already in its title, *Cybernetics: or control and communication in the animal and the machine*, this book makes a central claim which is valid to this day: Information processing in biology and technology is based on the same principles and procedures, at least if an appropriate level of description is selected. This "essential unity of problems ... whether in the machine or in living tissue" (p. 11 of the 1961 edition) implies two research directions, i.e. the application of mathematics and systems theory to the analysis of neural systems and, reversely, the application of neural procedures to the solution of technical problems. The first approach proved to be very successful. The analysis of sensors and sensoric processing, motor control, and the principles of learning and adaptation in modular systems are now well understood and parameterized, largely based on available mathematical tools. The opposite approach went through ups and downs. In the attempt to exploit the alleged superiority of neural systems, subproblems such as pattern recognition, localization, or learning in neural nets were singled out and their respective neural substrates were analyzed—often with disappointing results: whenever a well conditioned objective function could be defined, optimal realizations by means of technical algorithms and procedures turned out to be more efficient. More recently interest has shifted to machine learning and statistical learning theory, keeping only loose connections to neuroscience and biology. There have, however, been attempts to make neuroscience and computation re-converge, mostly by means of a unified theoretical treatment of the various subproblems mentioned above. The present book is part of this endeavor.

Good scientific books pave a way of thinking which makes concepts understandable, quantifies the facts, and puts them in adequate perspective. Professor Mallot has done just this. Starting with the intricate electrochemical properties of single cells, he proceeds to the connectivity of neurons. Their space-time dependent response properties represent the essence of all neural systems. The learning dependent variations of these nets are also treated as an essential aspect. The associative memory is one example of this property. To understand the coupling of real spatially two-dimensional structures, one has to include some additional aspects like population coding or the geometric and parametric mappings that facilitate computations in two-dimensional representations. The application-oriented claim of cybernetics is treated in a chapter on artificial neural nets. The author, who also published important contributions to the entire theoretical descriptions of neural systems, has the rare talent to put facts and their relations in mathematical terms in a way that is easily understood. Also, the handling of the problems with continuous as well as with discrete methods and the introduction of necessary non-linearities into the description can be captured without special training.

Recent progress in experimental methods has provided exciting new insights into the structure of the brain on all levels of complexity and elucidated the efficiency and the variability of the systems. Theoretical concepts have become more profound and more adequate for dealing with biological constraints. Self-organization and life-long adaptation of the brain proved to determine both its anatomical structures and the way data and knowledge are represented. The underlying processes—and with that the decomposition of tasks—seem to be based on a relatively uniform set of learning procedures and anatomical architectures (such as cortices), both favoring the flexibility of the entire systems. Thus, brains and the information they contain are in a permanent process of adaptation to their current goal. This process of self-organization seems to be the one driving force of both brain development and learning.

The self-organization of neural systems is of course of substantial interest for technical systems with their ever-increasing complexity and diversity. Self-organization of neuron-like structures based on their innate flexibility may lead a way to reduced design efforts. In this sense, the original quest of cybernetics is still alive, albeit with an extended focus on the whole, behaving organism: How do neural systems succeed to autonomously overcome their ignorance and to exploit the predictabilities of their environment for optimal behavior and, indeed, for evolutionary success? Evolution and ecology prove, of course, that solutions to these questions exist.

Not surprisingly, the computational methods used in the neural and behavioral sciences cover the major part of scientific computing at large. The book focuses on the most important approaches and lays the foundations for extended questions and specifications. It shows that it is not only the detailed mathematical formalisms that are crucial but also the broader concepts which allow appropriate mathematizations of neuroscientific phenomena and concepts. This book delivers a clear and understandable introduction into the structure of the problems of a scientific domain that still belongs to the most interesting fields science has to offer.

Bochum and Mainz, April 2013

Werner von Seelen
Institute for Neural Computation
Ruhr-University Bochum

Preface

Mathematical modeling in neuroscience is by now a common tool in many research projects and laboratories; its coverage in neuroscience curricula is therefore increasing. At the same time, computational neuroscience is still largely considered a domain of specialists who have acquired a mathematical background in physics or engineering and subsequently came to apply these ideas to questions of the physiology of neurons and neural networks and of the information processing carried out by these structures.

This book attempts an inverse approach. It grew out of a lecture series delivered over a period of more than ten years to graduate students in neuroscience, most of which had backgrounds in biology, medicine, psychology, or cognitive science. The required mathematical prerequisites for the course were therefore limited to the level of a bachelor degree in science, i.e. to basic calculus and linear algebra. All mathematical issues beyond this level—such as differential equations, convolution, complex numbers, high-dimensional vector spaces, or the statistical information measure—are thoroughly motivated and explained as they arise. I tried to keep these explanations mathematically clean but in cases had to omit subtleties which a full mathematical treatment ought to provide. The text reflects extensive class-room discussions and presents routes of explanation accessible also to mathematical non-experts. This is particularly true for the chapter on Fourier transforms which is the least neuroscientific one in the book. It is, however, instrumental for a deeper understanding of receptive fields and visual information processing.

The book also assumes some familiarity with the basic facts of neuroscience; if problems arise, any textbook of neuroscience will help.

Computational methods are applicable and useful in all fields of neuroscience. In the history of the field, three main roots can be identified, i.e. membrane biophysics, linear and non-linear systems theory, and machine learning. In the biophysics of neural membranes, modeling and experimentation have developed jointly from the beginnings in the 19th century. Membrane potentials are readily treated with the methods of electrodynamics and electrochemistry, and computational neuroscience strongly draws on these disciplines. The theory of receptive fields, developed since the 1940s, uses concepts from two sources, quantum mechanics and "signals and systems" in electrical engineering which may be summarized as systems theory. Since the advent of computer vision, visual neuroscience has in turn contributed important concepts to the computational sciences, including neighborhood operations, feature extraction, and scale space. Artificial neural networks form the most recent addition to theorizing in the neurosciences. They initially applied concepts from linear algebra and multivariate statistics combined with an explorative use of computer algorithms. More recently, artificial neural networks became a part of the boosting field of machine learning, which is not *per se* interested in the neurobiological plausibility of its algorithms. Still, it offers a wealth of theoretical concepts that neuroscientists can make use of. The book focuses on these three domains of computational neuroscience. Higher level topics such as memory and representation, decision making, reasoning, etc., i.e. the whole field of cognitive modeling could not be included in this volume.

The lecture course was accompanied by a journal club in which related classical and recent neuroscience publications were presented and discussed. The "suggested reading" sections refer to a number of papers used in these seminars. They are recommended for a deeper study of the subject.

Finally, I would like to thank my students for the many lively discussions in class and during the preparation of the seminar papers. They have helped me to constantly rethink and improve my presentation.

Tübingen, April 2013 *Hanspeter A. Mallot*

Contents

Chapter 1
Excitable Membranes and Neural Conduction

Abstract. Neural information processing is based on three cellular mechanisms, i.e., the excitability of neural membranes, the spatio-temporal integration of activities on dendritic trees, and synaptic transmission. The basic element of neural activity is the action potential, which is a binary event, being either present or absent, much as the electrical signals in digital circuit technology. In this chapter, we discuss the formation of the action potential as a result of the dynamics of electrical and chemical processes in the neural membrane. In order to infer the closed loop dynamics from the individual processes of voltage sensitive channels and the resulting resistive and capacitive currents, a mathematical theory is needed, known as the Hodgkin-Huxley theory. The propagation of neural signals along axons and dendrites is based on the cable equation which is also discussed in this chapter. Mathematical background is mostly from the theory of dynamical systems.

1.1 Membrane Potentials

In this section, we review the basic facts from membrane neurophysiology as a background to the mathematical models discussed later. As a starting point, we consider

Table 1.1 Ion concentrations and Nernst potentials of three ion types (from Aidley, 1998)

		Ion concentrations		
	Ion	External (mM)	Internal (mM)	Nernst potential (mV)
Frog muscle	K^+	2.25	124	-101
	Na^+	109	10.4	+59
	Cl^-	77.5	1.5	-99
Squid axon	K^+	20	400	-75
	Na^+	440	50	+55
	Cl^-	560	40	-66

H.A. Mallot: *Computational Neuroscience*, SSBN 2, pp. 1–21.
DOI: 10.1007/978-3-319-00861-5_1

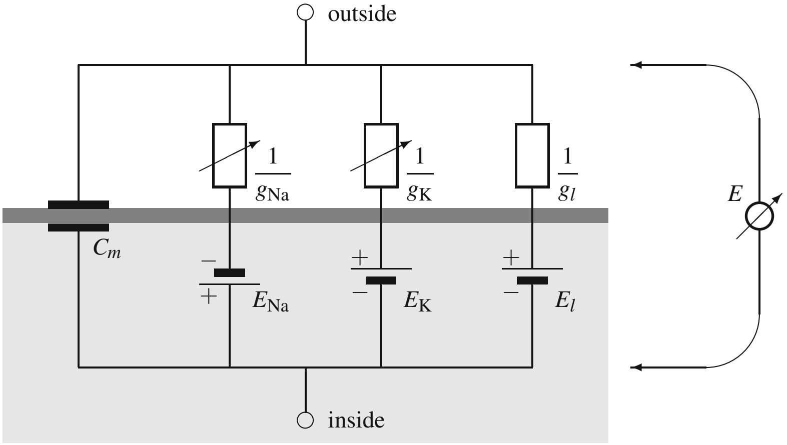


Fig. 1.1 Electrical circuit representing the membrane of a neuron. Redrawn from Hodgkin & Huxley (1952)

the resting potential of nerve cells, i.e. the voltage difference of about -70 mV (inside negative) across a neuron's cellular membrane. This voltage difference is due to an unequal distribution of various ions, including the cations of sodium (Na^+)[1], potassium (K^+)[2], and calcium (Ca^{2+}), as well as the anion chloride (Cl^-) and the anions formed by acidic proteins (i.e. X-COO^-). While the latter cannot permeate the cell membrane, the small inorganic ions can do so to various degrees. In fact, the membrane contains special protein complexes, the channels, which allow the ions to pass. These channels are generally specific for particular ion types and their opening may depend on the presence or absence of specific chemical "ligands" (such as synaptic transmitters), or on the membrane potential. Indeed, voltage dependence of ion channel opening is the key property of neuron membranes from which the excitability of neurons results. It is thus the basis of neural information processing.

In the steady state, i.e. if the neuron is not active, the permeability of the membrane for each ion sort is constant. In this situation the ion distribution across the membrane and the membrane potential are related by the the Nernst[3] and Goldman[4] equations. If only one ion sort is able to permeate the membrane, the Nernst equilibrium is obtained if the osmotic force, resulting from unequal distribution of ions across the membrane, just cancels the electro-motor force, resulting from the unequal distribution of electric charges. Consider a hypothetical membrane separating

[1] The chemical symbol Na refers to the German "Natrium", named after Lake Natron in Tanzania, which is known for its high concentration of Na^+ ions.

[2] The chemical symbol K refers to the German "Kalium". It is derived from the Arab "al-qalya", which refers to plant ashes in which potassium was first discovered.

[3] Walter Nernst (1864 – 1941). German physicist and chemist. Nobel Prize in Chemistry 1920.

[4] David Eliot Goldman (1910 – 1998). United States physicist.

two compartments one of which filled with pure water and the other one filled with the solution of a potassium salt of an acidic protein. If the membrane is permeable to potassium cations, but not to the large anions, potassium cations will cross the membrane following osmotic force, i.e. by diffusion. Since the anions cannot follow, this leads to a depletion of positive charges, i.e. a surplus of negative charges which drives the cations back into their original compartment. The eventual equilibrium where both forces cancel out is given by Nernst's equation. In a realistic neuron, the potassium equilibrium potential is in the order of -100 mV. Similarly, a Nernst potential of +60 mV can be calculated for the Na^+ ions, see Table 1.1. Note that the membrane potential is measured as the difference inside minus outside, -100 mV therefore means that the inside is negative. The Nernst potential marks a thermodynamical equilibrium, i.e. a steady state which is stable without requiring energy for its maintenance. If the membrane potential is changed (e.g. by putting electrodes into the compartments and applying an external voltage), the electro-motor force changes and ions will move until a new equilibrium is reached.

If ions of more than one type can cross the membrane, the equilibrium is more complex, since the electro-motor forces are the same for all ion sorts with the same charge, whereas the osmotic forces depend on the concentrations of each ion sort separately. In effect, a complex mixture of the individual Nernst equilibria is obtained, described by the Goldman equation (see textbooks of physiology for details). The resting potential of a typical neuron is determined by the concentrations of K^+, Na^+, and Cl^- ions and their respective equilibrium (Nernst) potentials, according to Goldman's equation. It is not at a thermodynamical equilibrium; for its maintenance sodium ions are constantly pumped outwards and potassium ions inwards by an active (i.e. energy consuming) process, the sodium-potassium-pump located in the cell membrane.

The electrical properties of a small patch of membrane can be modeled by the simple electrical circuit shown in Figure 1.1. The distributions of the sodium, potassium and chloride ions act as batteries generating an ion current across the membrane. The conductivity of the membrane for the two ions is denoted by g_{Na} and g_{K}, respectively. The arrows drawn across the resistors indicate that conductivity can change. This change of sodium and potassium conductivity as a function of the membrane potential is the basis of the excitability of neural membranes. The third channel is not voltage dependent and is called "leakage"; its current is mostly due to chloride ions. Finally, the membrane has a capacitance denoted by C_m.

The action potential is generated by the different dynamics of the changes in membrane conductivity for the different ions. An initial depolarization causes the sodium channel to open and sodium ions will come in, leading to a still stronger depolarization of the membrane. After about 1 ms, the sodium channel inactivates (i.e. closes) in a process depending on intrinsic dynamics, but not on membrane potential. As an effect of depolarization, the potassium channels will open as well, however, with a slower time course. When they open, potassium will move out and the membrane is repolarized. As an effect of repolarization, the potassium channels close again. When the action potential is over, the ion distributions in the cell and its environment will be slightly less different, due to the involved ion currents.

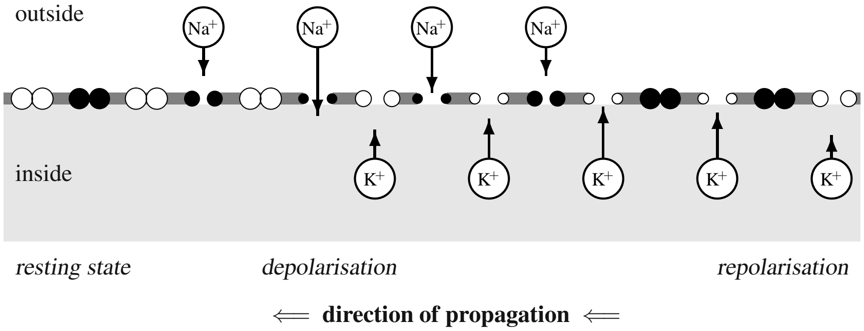


Fig. 1.2 Changes in conductivity of an excitable membrane during the course of an action potential. Initially (left side), the depolarization traveling passively along the axon will lead to an opening of the Na^+-channels (•). This opening is transient and finishes after about 1 ms independent of membrane potential. As a result, the membrane potential will rise quickly (Depolarization). With a certain delay, the potassium channels (K^+; ◦) will open. Due to the reverse distribution patterns of the two ions, this will lead to an outflow of positive charge, i.e. to repolarization. This in turn leads to the closure of the potassium channels. The length of the arrows symbolizes the conductivities g_{Na} and g_{K} (cf. Figure1.8).

The number of ions passing during each action potential can be calculated from the membrane capacity, the voltage change, and the elementary charge. Roughly, about 0.1% of the ions are exchanged during one action potential. In the long run, these differences are compensated for by the sodium-potassium-pump. A schematic picture of the opening and closing events is given in Fig 1.2.

1.2 The Hodgkin-Huxley Theory

One use of mathematical theory in neuroscience is its capacity to predict the results of quantitative measurements from other such measurements. If successful, this prediction is strong evidence for the underlying model. In the case of the action potential, the theory developed by Hodgkin[5] and Huxley[6] (1952) provides compelling evidence for the mechanism sketched out in the previous section. The basic ideas and arguments of this theory are still valid today.

Consider once more the simple circuit presented in Figure 1.1. The total current across the membrane is separated into the ionic currents through voltage dependent channels, i.e. sodium (I_{Na}) and potassium, (I_{K}), a leakage current (I_l) lumping all ions passing through non-voltage dependent channels, and a capacitive current, I_c.

[5] Alan Lloyd Hodgkin (1914 – 1998). English physiologist. Nobel Prize in Physiology or Medicine 1963.

[6] Andrew Fielding Huxley (1917 – 2012). English physiologist. Nobel Prize in Physiology or Medicine 1963.

$$I = I_c + I_{\mathrm{Na}} + I_{\mathrm{K}} + I_l \tag{1.1}$$

$$= C_M \frac{dE}{dt} + g_{\mathrm{Na}}(E - E_{\mathrm{Na}}) + g_{\mathrm{K}}(E - E_{\mathrm{K}}) + g_l(E - E_l) \tag{1.2}$$

For the capacitive current I_c, we have used Coulomb's law of capacitance: The current I flowing into a capacitor is proportional to the change of voltage E across the capacitor, $I = C\frac{dE}{dt}$ (see Figure 1.11b). The factor C is called capacitance and measured in F (Farad) = Coulomb/Volt. Intuitively, capacity describes the fact that ions of opposite polarity attach to the membrane by mutual electric attraction. They thus form a charge reservoir that has to be recharged if the membrane potential changes. Recharging will take the longer, the larger the capacitance is, i.e. the more charges are attached to the membrane.

For the ion currents, Equation 1.2 uses Ohm's law (see Figure 1.11a). Instead of using resistances R, i.e. the ratio of current and voltage measured in Ω (Ohm), the Hodgkin-Huxley theory is generally formulated using the conductivities g, which are simply the inverses of resistance, $g = 1/R$. The unit of conductivity is S (Siemens) $= 1/\Omega$. E_{Na}, E_{K} and E_l denote the Nernst potentials listed in Table 1.1. The ion currents are proportional to the difference between the membrane potential and the ion's Nernst potential. If both potentials are equal, the ion type is at equilibrium.

Instead of the membrane potential E, standard formulations of the Hodgkin-Huxley theory generally use the depolarization V which is the difference between the membrane potential and the resting potential,

$$V = E - E_{resting}. \tag{1.3}$$

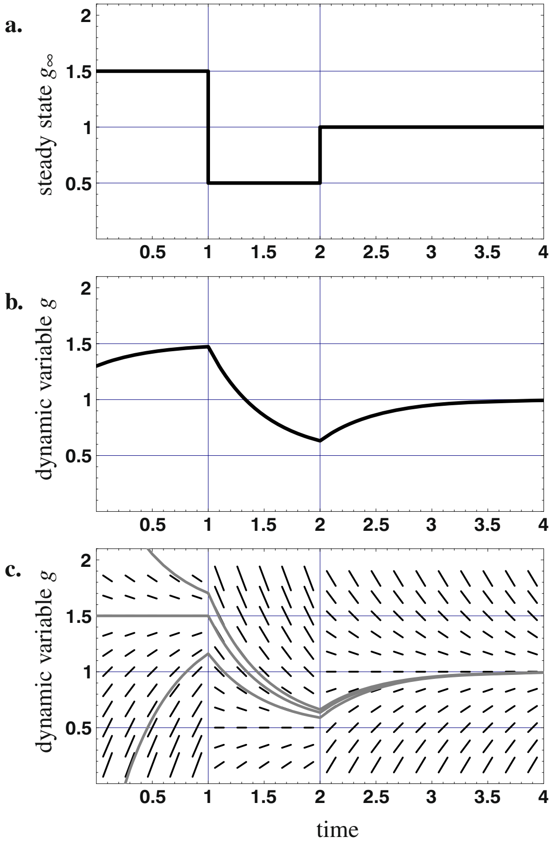
Depolarization is zero if the membrane is at rest and becomes positive for excited states. We will switch to this notion from here.

In the sequel, we will discuss the excitable sodium and potassium channels and present phenomenological models for their dynamics. These phenomenological models will then be combined using Equation 1.2 to generate a prediction for the time course of the action potential.

1.2.1 Modeling Conductance Change with Differential Equations

At the core of the Hodgkin-Huxley theory of the action potential lies the fact that the membrane conductance depends on the voltage or potential across the membrane. For example, if the membrane is depolarized, the conductances for both potassium and sodium are larger than at resting potential. This dependence has two parts: (i), a static dependence $g_\infty = f(V)$ describing the steady state, where the membrane potential is constant over long time intervals, and (ii), a dynamic part describing the time course of conductance change in response to changes in potential. Before we proceed with the potassium channel, we briefly discuss how to model dynamics with differential equations.

Fig. 1.3 Behavior of a simple differential equation. **a.** Time course of an outer "driving force", expressed as the steady state eventually obtained by the system. It is switched at discrete times. **b.** Exponential relaxation to the respective steady states as expressed in Equation 1.4. The value g reached at the time of switching of the driving force acts as initial value for the next relaxation. **c.** Vector field illustrating the differential equation 1.5. The continuous lines show three solutions for different initial values taken at $t = 0$.



Let us assume that the membrane potential changes in a step-like fashion at time $t = 1$ from a value V_1 to V_2 and again at time $t = 2$ from V_2 to V_3 (Figure 1.3). Such controlled membrane potentials and membrane potential changes are studied in the "voltage clamp preparation", where physiological fluctuations of the membrane potential are compensated for by an electronic circuit. The steady-state conductivities $g_\infty(V_i)$ are shown in Figure 1.3a. The actual conductivities $g(t)$ will differ from g_∞ in that they need some time to change between the steady state levels. A plausible time course of conductivity showing the non-zero switching times (finite switching speed) appears in Figure 1.3b. The curve shows an exponential relaxation which between two stepping times might be formally described by

$$g(t) = g_\infty + (g_o - g_\infty)\exp\{-\frac{t - t_0}{\tau}\}. \tag{1.4}$$

Here, g_o is the initial value of g at time t_o, g_∞ is the steady state of the voltage applied during the current interval, and τ is a time constant determining the rate of approach towards the steady state. Note that the approach to the steady state is the faster (the curve is the steeper), the larger the deviation from the steady state is.

A disadvantage of this equation is the fact that it can deal only with step changes of membrane potential occurring at discrete points in time, but not with continuous changes as they occur during the action potential. A formal description that holds for both step changes and continuous changes is derived from the observation that the rate of change, i.e. the slope of $g(t)$ should be proportional to the current deviation from the steady state given by $g(t) - g_\infty$:

$$g'(t) = -k(g(t) - g_\infty) \quad \text{for some constant } k. \tag{1.5}$$

Here we write g_∞ for the steady state, keeping in mind that it depends on V. Whatever changes $V(t)$—and therefore g_∞—undergoes over time, Equation 1.5 keeps valid. An illustration of Equation 1.5 is given in Figure 1.3c. Each short line marks a local slope $g'(t)$ calculated from Equation 1.5. Each curve drawn in a way that the local lines in Figure 1.3c are tangents to the curve is a solution of Equation 1.5.

Equation 1.5 is an example of a *differential equation*, relating the value of a function to its derivative. The solutions of differential equations are functions (or families of functions), not numbers. By taking the derivative of g in Equation 1.4 and comparing it to g itself, it is easy to show that this function solves Equation 1.5 if k is set to $1/\tau$. We start by taking the derivative of Equation 1.4

$$g'(t) = -\frac{1}{\tau}(g_o - g_\infty)\exp\{-\frac{t - t_o}{\tau}\}. \tag{1.6}$$

Inserting g' and g (again from Equation 1.4) into Equation 1.5 we obtain

$$-\frac{1}{\tau}(g_o - g_\infty)\exp\{-\frac{t-t_o}{\tau}\} = -k(g_\infty + (g_o - g_\infty)\exp\{-\frac{t-t_o}{\tau}\} - g_\infty)$$

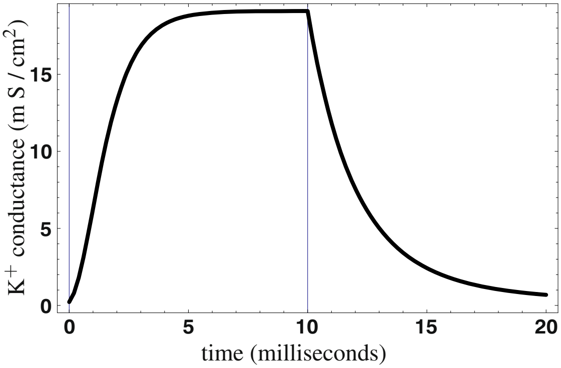
which is satisfied if we set $k = -1/\tau$.

Equation 1.4 is called an analytical solution of the differential equation 1.5 because it is formulated as a named mathematical function, i.e. the exponential. The sketched procedure, i.e., guessing a solution and proving that it satisfies the differential equation if appropriate choices of some variables are made, is a standard way of finding analytical solutions. In contrast, numerical solutions are sequences of functional values (i.e. numbers) obtained at discrete time-steps. We will see below that for the full Hodgkin-Huxley system, only numerical solutions exist.

1.2.2 The Potassium Channel

Individual channels can be studied in voltage clamp preparations using axons in solutions with different concentrations of the sodium and potassium ions. We will not discuss the methods for distinguishing the ions here (see for example Aidley, 1998). Figure 1.4 shows the result of such a measurement where the membrane potential is stepped from resting potential to a depolarization of 109 mV at time

Fig. 1.4 Potassium conductance change (g_K) in the voltage clamp experiment (in milli-Siemens per square centimeter). Membrane potential is stepped from resting potential by 109 mV (depolarization) at time $t = 0$ ms and stepped back to resting potential at time $t = 10$ ms.



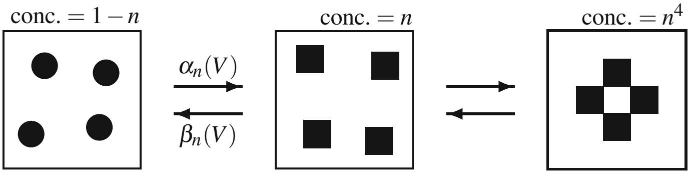


Fig. 1.5 Kinetics modeled by Equations 1.7 and 1.8 (potassium channel). Left: pool of inactivated subunits. Middle: activated subunits, portion of subunits in activated state is n. Right: Channel formed of four subunits. Probability of four subunits joining in one place is n^4.

$t = 0$ ms. Conductance rises to a plateau, with the highest speed of change occurring not at the beginning (as would be the case for the simple relaxation equation studied above), but at some intermediate value. This behavior implies that the solution should have an "inflection point", i.e. the point of highest speed after the onset of the "relaxation". This inflection point will be important for the modeling of the channel switching behavior. At time $t = 10$ ms, the membrane is repolarized and the conductance relaxates to zero, this time without an inflection point. If other depolarizations are chosen, the shape of the curve changes slightly. This is to say that the response is non-linear.

Hodgkin and Huxley (1952) suggested to model this behavior by a two-step process including (i) a simple relaxation process described by an axillary variable n and (ii) a fourth order interaction term needed to model the inflection point found in the voltage clamp response:

$$g_K(t) = \bar{g}_K \, n(t)^4 \tag{1.7}$$

$$\frac{dn}{dt}(t) = \alpha_n \, (1 - n(t)) - \beta_n n(t). \tag{1.8}$$

Here $\bar{g}_K$ is a constant (about 22 milli-Siemens per centimeter squared) depending neither on time nor on voltage. It represents the maximum conductivity if all potassium channels are open, i.e. the number of channels present in the membrane.

Equation 1.7 means that four channel subunits have to meet in order to form a channel. The probability of finding four subunits in one spot is equal to the probability of finding one, raised to the fourth power (as long as the subunits move independently in the membrane). Equation 1.8 assumes that the channels can be in one of two states, one favorable of forming a channel and one unfavorable. The dimensionless number n is the fraction of channel subunits being in the favorable state; it varies between 0 and 1. α_n and β_n are called "rate constants" which depend on voltage, but not on time. They specify the likelihood (the rate) at which the channel subunit switches from the unfavorable state into the favorable state ($\alpha_n(V)$), or back ($\beta_n(V)$). These rate constants thus incorporate the voltage dependence of channel formation. $\alpha_n(V)$ can be expected to grow with V whereas $\beta_n(V)$ should decrease. The equation is illustrated as a chemical reaction kinetic in Figure 1.5.

The exponent four in Equation 1.7 has been chosen to reproduce a subtle feature of the conductivity step response shown in Figure 1.4. While $n(t)$ will show simple exponential relaxation, $n^4(t)$ nicely fits the inflection point occurring in the rising branch but not in the falling one. It is remarkable that molecular biology has since confirmed the involvement of just four subunits in the formation of the potassium channel, as predicted from the voltage clamp measurements by Hodgkin & Huxley (1952).

Observe that Equation 1.8 has the same structure as Equation 1.5 studied above, i.e., $n' = -k(n - n_\infty)$, if we rearrange the terms and replace k by $\alpha_n + \beta_n$ and n_∞ by $\alpha_n/(\alpha_n + \beta_n)$.

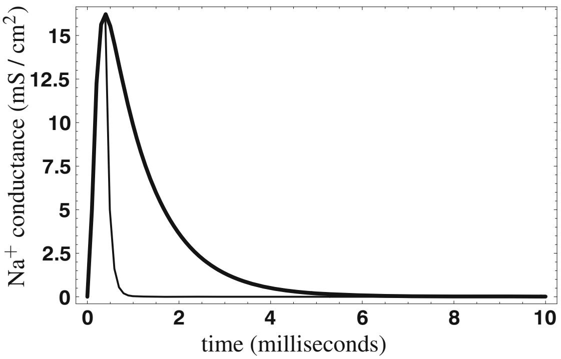
$$\frac{dn}{dt}(t) = -\underbrace{(\alpha_n + \beta_n)}_{k}(n(t) - \underbrace{\frac{\alpha_n}{\alpha_n + \beta_n}}_{n_\infty}). \tag{1.9}$$

This implies that the values of $\alpha_n(V)$ and $\beta_n(V)$ can be measured by performing voltage clamp experiments with different voltage steps. For each voltage step (V) applied, we take the fourth root to go from $g/\bar{g}$ to n; the time constant of the resulting exponential decay curve then corresponds to $\alpha_n(V) + \beta_n(V)$ while the steady-state value corresponds to $\alpha_n(V)/(\alpha_n(V) + \beta_n(V))$. From these two measurements, α_n and β_n themselves can be calculated. As expected, α_n increases with higher voltage (i.e. more channel subunits change into the activated state) while β_n decreases with higher voltage (i.e. more channel subunits change to inactivated state). The detailed dependence of the rate constants on voltage can be found in Hodgkin & Huxley (1952).

1.2.3 The Sodium Channel

The response of the sodium conductance to a step in membrane potential of 109 mV is shown in Figure 1.6 (heavy line). It differs from the potassium response in its quicker rise and in its overall phasic behavior, i.e. the conductance goes back to zero after a few milliseconds of depolarization. If the membrane is repolarized at a time when conductance is still high, the drop in conductance is very steep (thin line in Figure 1.6).

Fig. 1.6 Sodium conductance change (g_{Na}) in the voltage clamp experiment (in milli-Siemens per square centimeter). Membrane potential is stepped from resting potential by 109 mV (depolarization) at time $t = 0$ ms. The thin line shows the conductance if the membrane is repolarized at time $t = 0.39$ ms.



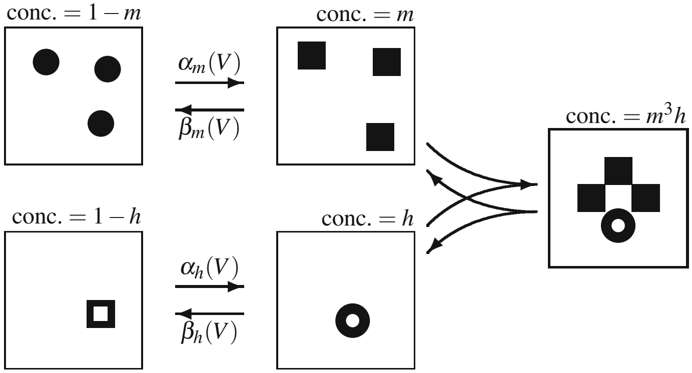


Fig. 1.7 Kinetics modeled by Equation 1.12 (sodium channel). Circles: inactivated "particles"; boxes: activated particles. Filled symbols: activation particles, open symbol: inhibition particles.

The phenomenological model presented for this dynamics by Hodgkin & Huxley (1952) uses two auxiliary variables, one (m) modeling the raise (activation gate) and one (h) modeling the drop (inactivation gate) of conductance:

$$g_{\mathrm{Na}}(t) = \bar{g}_{\mathrm{Na}}\, m(t)^3\, h(t) \tag{1.10}$$

$$\frac{dm}{dt}(t) = \alpha_m\,(1-m(t)) - \beta_m m(t) \tag{1.11}$$

$$\frac{dh}{dt}(t) = \alpha_h\,(1-h(t)) - \beta_h n(t). \tag{1.12}$$

As before, the rate constants α_m, β_m, α_h, and β_h depend on depolarization, but not on time. They can be obtained from voltage clamp measurements, as explained above for the potassium channel.

Figure 1.7 shows a state diagram of the Hodgkin-Huxley kinetics of the sodium channel. The match with the actual molecular mechanisms is not as close as for the potassium channel. Indeed, more complex models of sodium channel kinetics have been suggested. For a recent discussion, see Milescu et al. (2008).

1.2.4 Combining the Conductances in Space Clamp

So far, we have considered how the channel conductivity depends on membrane potential and used the voltage clamp preparation to formalize this relation. Channel conductivity of course affects ion currents across the membrane which in turn will change membrane potential. The whole process is thus a feedback loop of membrane potential, membrane conductivity, membrane current and again membrane potential, which we have artificially disconnected by means of the voltage clamp preparation. In this section, we will now proceed to consider the closed loop situation.

The first step to this end is to look at a complete action potential in the so-called space-clamp situation where the potential is kept constant over a small patch of membrane but may vary over time. This can be achieved by inserting a highly

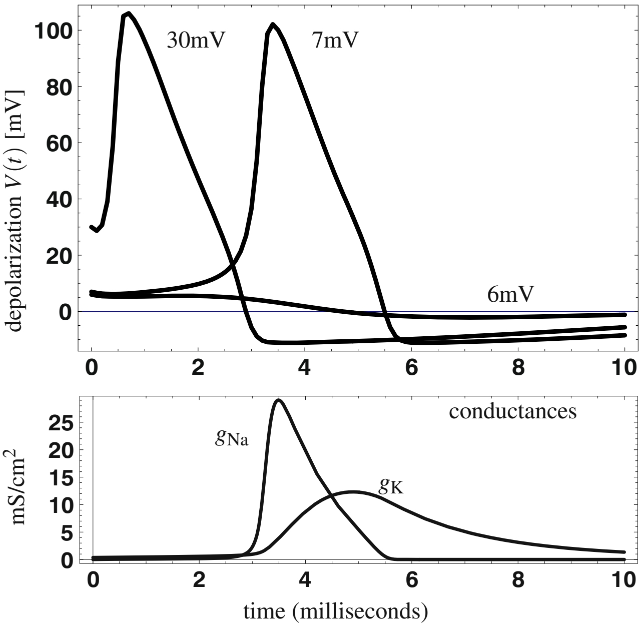


Fig. 1.8 Solutions of the space-clamp equations for external stimuli V_0 occurring at time $t = 0$. Top: Three solutions for the depolarization resulting from external stimuli $V_0 = 6$ mV, 7 mV, and 30 mV above resting potential. The threshold for spike generation is between 6 and 7 mV. Spikes generated by small and large super-threshold stimuli look the same ("all or nothing" rule of spike generation). Bottom: conductance changes resulting from a stimulation with 7 mV.

conducting copper wire into an axon which will equilibrate all differences in membrane potential occurring along the axon. The whole patch will thus exhibit the action potential in synchrony.

In the space-clamp situation, no current will flow in axial direction since potentials are constant over space. Therefore, there can also be no current loops involving an axial component, which in turn excludes net currents across the membrane. We may thus assume $I = 0$ in Equation 1.2 and obtain the following set of four coupled ordinary differential equations:

$$\begin{aligned} -C_M \frac{dV}{dt}(t) &= \bar{g}_{\mathrm{K}}\, n(t)^4\, (V(t) - V_{\mathrm{K}}) + \\ &\qquad \bar{g}_{\mathrm{Na}}\, m(t)^3\, h(t)\, (V(t) - V_{\mathrm{Na}}) + \bar{g}_l\, (V(t) - V_l) \qquad (1.13) \\ \frac{dn}{dt}(t) &= \alpha_n(V(t))\, (1 - n(t)) - \beta_n(V(t))\, n(t) \qquad (1.14) \\ \frac{dm}{dt}(t) &= \alpha_m(V(t))\, (1 - m(t)) - \beta_m(V(t))\, m(t) \qquad (1.15) \\ \frac{dh}{dt}(t) &= \alpha_h(V(t))\, (1 - h(t)) - \beta_h(V(t))\, h(t) \qquad (1.16) \end{aligned}$$

The coupled system of non-linear differential equations 1.13 – 1.16 does not have analytical solutions, i.e. we cannot derive a closed formula for the time-course of the action potential. However, for the overall argument, it suffices to compute numerical solutions, i.e. sampled functions approximating the solution, which can then be compared to the measurements. Numerical solutions are obtained by first specifying “initial values” for each of the four variables V, n, m, and h. For V, the initial value is simply the external stimulus. For the auxiliary n, we observe that dn/dt should be zero if the membrane is at rest. Therefore, we obtain from equation 1.14: $n_{rest} = \alpha_n(0)/(\alpha_n(0) + \beta_n(0))$ where the value 0 in the arguments refers to deplorization. Analogous results are obtained for m and h. The initial values are inserted in the right side of the equations. Each equation then gives a value for the slope of the respective state variable at time 0. With this slope and the initial value, the values at a time t is estimated by linear extrapolation. The resulting new estimates are again inserted in the equations, leading to new slope values and in turn to new estimates of the state variables at time $2t$. This procedure is iterated until the time interval considered is exhausted. A more elaborate version of this scheme, known as Runge-Kutta-integration, was used to produce the plots in this text.

Figure 1.8 shows a numerical solution of the system 1.13 – 1.16 for an external stimulus of $V(0) = 7$ mV. The depolarization V is plotted in the upper part. The shape of the action potential is in very good agreement to the shape measured in space-clamp experiments using squid axons. The corresponding time course of the conductances is shown in the lower part of Figure 1.8.

If different external stimuli $V(0)$ are chosen, two different types of behavior are observed (Figure 1.8). If $V(0)$ is 6 mV or less, the effect is very small; the stimulus is sub-threshold. For stronger stimuli, an action potential is generated which looks more or less equal for all super-threshold stimuli. Stronger stimuli result in earlier

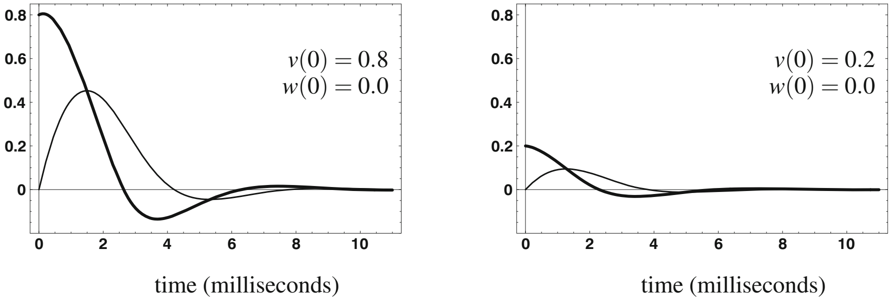


Fig. 1.9 Solutions of the FitzHugh-Nagumo system with $I_a = 0$, $a = 0.3$, $b = 0.8$, $\gamma = 1.0$. Left: start at $v = 0.8$, right: start at $v = 0.2$.

action potentials, not in bigger ones. The Hodgkin-Huxley equations thus capture the all-or-nothing behavior of neural excitation.

1.3 An Analytical Approximation: The FitzHugh-Nagumo Equations[7]

In the Hodgkin-Huxley equations, the sodium conductance g_{Na} has a much shorter time constant than the variables governing g_{K}. In an approximation, one can assume that $dm/dt = 0$. Further, the slow time constant of h might be approximated by assuming $h(t) \equiv h_o$, a constant. With these simplifications, we obtain from Equations 1.13 – 1.16 a simplified (two-dimensional) system whose qualitative properties can be studied in the so-called FitzHugh-Nagumo equations (cf. FitzHugh 1961, Murray 2002):

$$\frac{dv}{dt}(t) = f(v(t)) - w(t) + I_a \tag{1.17}$$

$$\frac{dw}{dt}(t) = b\, v(t) - \gamma\, w(t) \tag{1.18}$$

$$f(v) := v\,(a - v)\,(v - 1) \tag{1.19}$$

I_a is an externally applied current and a, b, γ are constants.

The FitzHugh-Nagumo equations are much simpler than the Hodgkin-Huxley equations in that they employ only two (rather than four) coupled variables; in fact, there exist analytical solutions. Examples of numerical solutions are shown in Figure 1.9. We will now discuss some of the qualitative properties of the dynamics of neural excitation, which can be studied best with the FitzHugh-Nagumo equations.

Figure 1.10a shows the so-called phase portrait of the FitzHugh-Nagumo system. Here, the variables v and w are plotted along the axes (note that this is intuitive only for two-dimensional systems). Time is not explicitly represented in this plot. The

[7] This section may be skipped on first reading.

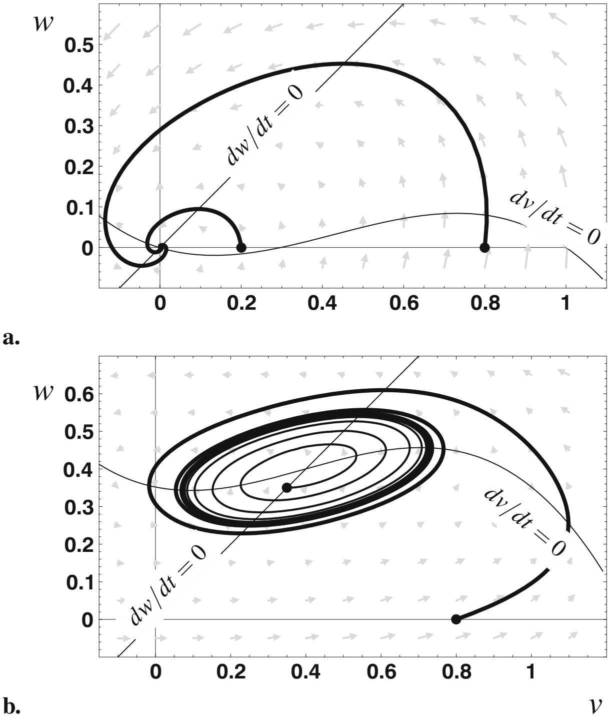


Fig. 1.10 Phase portraits of the FitzHugh-Nagumo system. **a.** Non-oscillating case. Parameters as in Figure 1.9. The thin lines are the "null isoclines" $dw/dt = 0$ and $dv/dt = 0$. The arrows give the initial direction of solutions starting from the respective points. The heavy lines show the two solutions from Figure 1.9. The converge to the intersection of the isoclines which forms a stable fixed point. **b.** Oscillator. Parameters $a = 0.2$, $b = 0.2$, $\gamma = 0.2$, $I_a = 0.352$. Note that the v-isocline have moved upwards. The intersection of the isoclines is no longer a stable point. Two solutions are shown approaching a stable limit cycle from within and from outside.

idea is that each point in the plane represents a state (or phase) of the system and the dynamics is represented by movements in that plane. For example, the solutions shown in Figure 1.9 are represented in Figure 1.10a as *trajectories*, $(v(t), w(t))$.

We can now look for states (i.e. pairs of numbers (v, w)) for which the system of equations 1.17 – 1.19 stops moving. Such points in the phase plane are called *fixed points* or *steady states*. From Equation 1.17 we obtain:

$$\frac{dv}{dt} = 0 \Longleftrightarrow w = f(v) + I_a \tag{1.20}$$

$$\frac{dw}{dt} = 0 \Longleftrightarrow w - \frac{b}{\gamma} v \tag{1.21}$$

These curves are called null-isoclines and are also plotted in Figure 1.10a. They intersect at the only fixed point $(0,0)$. This fixed point is stable in the sense that all arrows in a vicinity of it have a component in the direction of the fixed point. Therefore, the system will approach the fixed point and rest there.

If we add an applied current I_a, the curve $dv/dt = 0$ is shifted upwards. Depending on the slope of both curves ($dv/dt = 0$ and $dw/dt = 0$), there can be either one or three intersection points, i.e. points where both derivatives vanish. Figure 1.10b shows a case where only one such intersection point exists. The parameters are chosen such that it is the inflection point of the curve $dv/dt = 0$. For the parameters chosen, the fixed point is not stable. Trajectories starting in a neighborhood of the fixed point move away from it and enter a *limit cycle* orbiting the fixed point. Trajectories starting further out approach the limit cycle from outside. This result shows that the FitzHugh-Nagumo system can produce oscillating responses when activated by an applied current. The transition from a stable fixed point to an unstable fixed point with limit cycle, effected by the change of a parameter, is called a *Hopf-bifurcation*.

The qualitative behavior described here has been shown to occur for the Hodgkin-Huxley system as well, using numerical simulations. Oscillatory responses of excitable cells are known for example from the pacemaker cells in the sinoatrial and other nodes of the heart; here, the constant incoming current I_a is realized by strong leakage channels carrying a constant sodium current.

1.4 Passive Conduction

In the space-clamp, we have assumed that the potentials and currents are constant along a patch of membrane. In reality, a depolarization at a point x of an axon will spread laterally by means of an axial current i_a, leading to a depolarization of adjacent membrane patches. This process is called passive conduction since it does not involve active (voltage dependent) channels; it occurs in dendritic, somatic and axonal parts of the neuron alike. In dendrites, it is usually the only type of conduction, since voltage dependent channels are generally missing. In axons, passive conduction of depolarization will initiate action potentials in neighboring membrane sections, causing a propagation of the action potential along the fiber. Due to the refractory period of the membrane, the propagation will be directed. In order to derive the equations governing this spread and propagation, we need to remember some basic physics as is summarized in Figure 1.11.

Some of the quantities that we will have to consider differ from the situation in simple circuits in that they are continuous, or distributed in space. For example, the axial, or longitudinal, resistance within an axon or dendrite is the bigger, the longer the considered part of the fiber will be. Therefore, we have to consider a resistance per centimeter. The resistance across the membrane will vary inversely with the considered membrane area or fiber length. Since we consider only cylindrical fibers, we have to use a quantity with the dimension Ohms times centimeter. The current passing the membrane will be proportional to membrane area or fiber length. Finally, membrane potential and axial currents are independent of fiber length. A summary of the quantities involved and their dimensions is given in Table 1.2.

The basic relations can be inferred from inspection of Figure 1.12. We start by noting that the membrane potential will be a function of axon length and time, as are the specific currents involved. We therefore denote them as $V(x,t)$, $i_a(x,t)$, and $i_m(x,t)$, respectively. In Figure 1.12, the length variable is discretized, but we will use a continuous representation involving spatial and temporal derivatives. Following Ohm's law (Figure 1.11a), the axial current $i_a(x,t)$is proportional to the potential change in axial direction, i.e.

$$i_a(x,t) = -\frac{1}{r_a}\frac{\partial V}{\partial x}(x,t), \tag{1.22}$$

where r_a is the axial resistance measured in Ω/cm. The curly d (∂) denotes partial derivatives which are needed here since V depends on space and time, whereas the derivative is taken only with respect to space. We will not elaborate on partial derivatives here; for a mathematical definition see Figure 4.8a and Equations 4.20, 4.21.

The axial current will decrease for larger x since part of it will leak across the membrane, thus forming a membrane current i_m. From Kirchhoff's node rule (Figure 1.11c), we have:

$$-i_m(x,t) = \frac{\partial i_a}{\partial x}(x,t). \tag{1.23}$$

Substituting for i_a from Equation 1.22, we obtain

$$i_m(x,t) = \frac{1}{r_a}\frac{\partial^2 V}{\partial x^2}(x,t). \tag{1.24}$$

The membrane current i_m is composed of two components, a resistive, or leak current (following Ohm's law, Figure 1.11a) and a capacitive current (following Coulomb's law, Figure 1.11b). We can write

Table 1.2 Quantities involved in the cable equation. Only cylindrical membrane tubes are considered and all lengths are axial. See also Figure 1.12.

Symbol	Interpretation	Dimension
V	membrane potential	V (Volt)
i_a	current flow along the fiber axis	A (Ampere)
i_m	current flow across the membrane per fiber length	A / cm
r_a	axial resistance per fiber length	Ω (Ohm) / cm
r_m	membrane resistance times fiber length (1/membrane conductance per fiber length)	Ω cm
c_m	membrane capacitance per fiber length	F (Farad) / cm

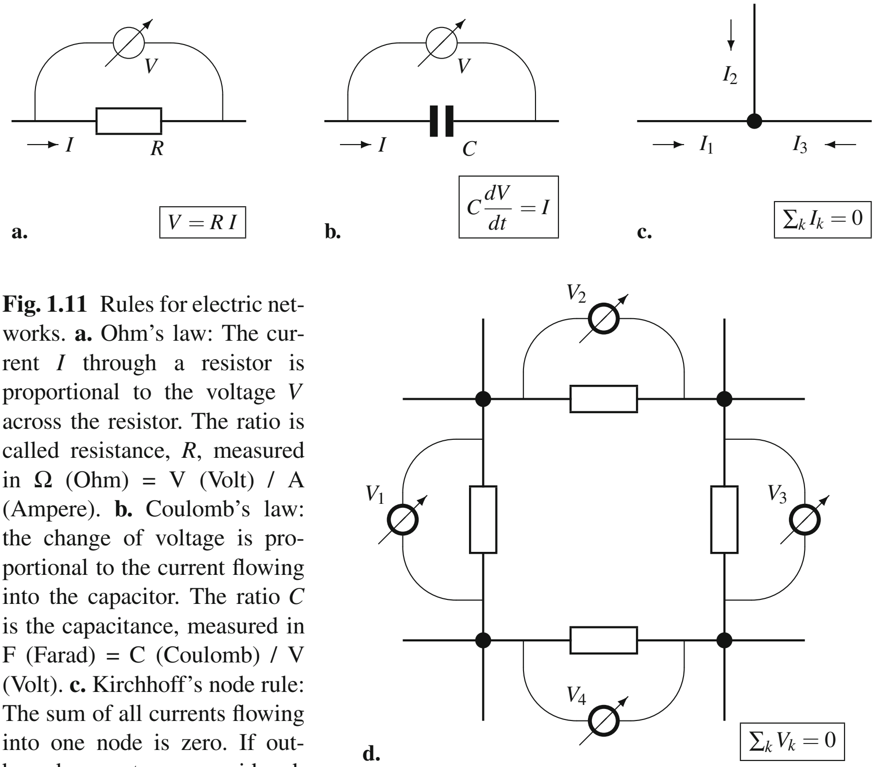


Fig. 1.11 Rules for electric networks. **a.** Ohm's law: The current I through a resistor is proportional to the voltage V across the resistor. The ratio is called resistance, R, measured in Ω (Ohm) = V (Volt) / A (Ampere). **b.** Coulomb's law: the change of voltage is proportional to the current flowing into the capacitor. The ratio C is the capacitance, measured in F (Farad) = C (Coulomb) / V (Volt). **c.** Kirchhoff's node rule: The sum of all currents flowing into one node is zero. If outbound currents are considered, the signs have to be changed accordingly. **d.** Kirchhoff's mesh rule: the sum of all voltages in a mesh adds to zero, if all voltages are measured either in a clockwise or in a counter-clockwise sense.

$$i_m(x,t) = \frac{1}{r_m}V(x,t) + c_m\frac{\partial V}{\partial t}(x,t). \tag{1.25}$$

We can now equate the above two expressions for i_m and obtain

$$V(x,t) = \frac{r_m}{r_a}\frac{\partial^2 V}{\partial x^2}(x,t) - r_m c_m \frac{\partial V}{\partial t}(x,t). \tag{1.26}$$

This is a partial differential equation known as the cable equation. If we assume that the potential is clamped to V_o at location $x = 0$, and consider the steady-state solution (i.e. $\partial V/\partial t = 0$), we obtain the simpler, ordinary differential equation[8]

[8] "Ordinary" is here the opposite of "partial". An ordinary differential equation contains only ordinary derivatives, as opposed to partial ones. The differential equations studied so far were all of the ordinary type.

$$V(x) = \frac{r_m}{r_a}\frac{d^2V}{dx^2}(x) \tag{1.27}$$

which has the solution

$$V(x) = V_o \exp\left(\frac{-x}{\sqrt{r_m/r_a}}\right). \tag{1.28}$$

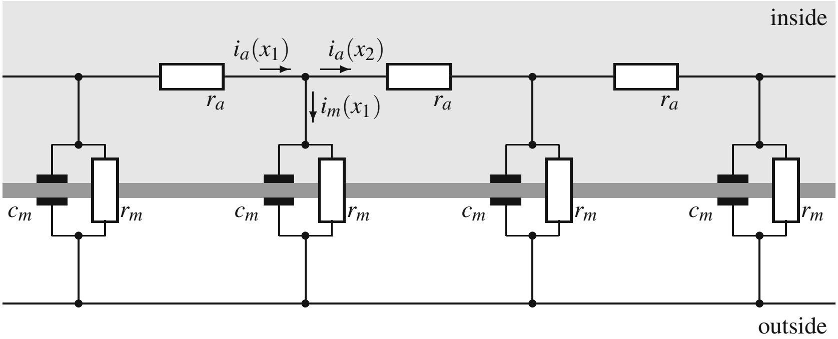


Fig. 1.12 Electrical circuit for modeling the passive propagation of activity along an axon. i_a axial current, i_m membrane current, r_a axial resistance, r_m membrane resistance, c_m membrane capacitance.

Eq: 1.28 describes an exponential decline of the local potential V_o with a "length constant" $\sqrt{r_m/r_a}$. In typical axons, it takes values in the order of a millimeter. Since r_m grows with the circumference of an axon while r_a grows with its cross-sectional area, thicker axons have larger length constants. The length constant determines the range of passive conductance. For example, the distance between the nodes of Ranvier may be the larger, the larger the length constant is. A dendritic tree whose size is well below the length constant may be treated as equipotential in dendritic summation. The influence of dendrite passive conductance on the electric properties of the neuron have been studied by Rall (1962).

In dynamic cases, e.g., in the propagation of action potentials on axons or passive conduction of postsynaptic potentials on dendrites, the capacitor will also play an important role. The speed of recharging the capacitor depends on the time constant $\tau = r_m c_m$ also appearing in Equation 1.26. It is largely independent of cable diameter but can be reduced by reducing the membrane capacitance. Myelin sheets, besides increasing membrane resistance, also have this effect of reducing membrane capacitance, since they increase the distance between charges in the intracellular and extracellular space. Therefore, passive conductance at internodia (i.e. between Ranvier nodes) is faster than in non-myelinated axons of equal thickness.

The spatio-temporal cable equation (1.26) is hard to solve analytically; for a discussion of the issue see Jack et al. (1975) and Tuckwell (1988). The best-studied case concerns a constant membrane current delivered from time $t = 0$ at location $x = 0$. In this case, the membrane potential will approach the steady state described by Equation 1.28. As it turns out, the approach to this steady state is the slower, the further away a given membrane patch is from the stimulation site. Overall, however, the speed is governed by the time constant $\tau = r_m c_m$.

1.5 Propagating Action Potentials

With the cable equation, we can now formulate the Hodgkin-Huxley Equation 1.13 in the space-variant case, i.e. for propagating action potentials. For active membranes, the leak current V/r_m in Equation 1.25 has to be replaced by the membrane current passing through the channels, i.e. $\bar{g}_K n(t)^4 (V - V_\mathrm{K})$ for the potassium channel plus the respective terms for the sodium channel and leakage. We thus obtain

$$\begin{aligned}\frac{1}{r_a}\frac{\partial^2 V}{\partial x^2}(x,t) = c_m \frac{\partial V}{\partial t}(x,t) + \bar{g}_\mathrm{K}\, n(t)^4\, (V(t) - V_\mathrm{K}) + \\ \bar{g}_\mathrm{Na}\, m(t)^3\, h(t)\, (V(t) - V_\mathrm{Na}) + \bar{g}_l\, (V(t) - V_l). \qquad (1.29)\end{aligned}$$

Equations 1.14 – 1.16 remain unchanged, except for the fact that the variables V, n, m, and h now depend additionally on spatial position x. Equation 1.29 is a partial differential equation which would be very hard to solve even numerically. One simplification used by Hodgkin and Huxley (1952) uses the fact that action potentials do not change their shape as they travel along an axon. I.e., if $V(x,t)$ is an action potential traveling with velocity θ, its shifted version after time Δt and distance $\theta \Delta t$ must look the same: $V(x,t) = V(x - \theta \Delta t, t - \Delta t)$. Thus, $V(x,t)$ can be written as a one-dimensional function $f(x - \theta t)$. Taking the second derivatives of f with respect to x and t, it is easy to see that the wave equation

$$\frac{\partial^2 V}{\partial x^2} = \frac{1}{\theta^2}\frac{\partial^2 V}{\partial t^2} \qquad (1.30)$$

must hold. We can therefore reduce Equation 1.29 to the ordinary differential equation

$$\begin{aligned}\frac{1}{r_a \theta^2}\frac{\partial^2 V}{\partial t^2}(t) = C_M \frac{\partial V}{\partial t}(t) + \bar{g}_\mathrm{K}\, n(t)^4\, (V(t) - V_\mathrm{K}) + \\ \bar{g}_\mathrm{Na}\, m(t)^3\, h(t)\, (V(t) - V_\mathrm{Na}) + \bar{g}_l\, (V(t) - V_l). \qquad (1.31)\end{aligned}$$

Numerical solutions of Equation 1.31 can be obtained if the parameter θ is selected in a suitable way. The solutions again reproduce the empirical shape of the action potential.

1.6 Summary and Outlook

In this chapter, we have studied the formation and propagation of action potentials in four steps:

1. Voltage clamp experiments: The switching dynamics of voltage dependent ion channels has been studied in an open loop preparation (voltage clamp), where current flowing through the channels does not affect membrane potential. This resulted in the kinetic models of the potassium and sodium channels.
2. Closing the loop in the space-invariant case: In the "space clamp" preparation, the loop is closed again but the potentials do not propagate in space. The theory of the case shows that standard theory of electric circuits and the models of channel kinetics developed before suffice to explain the time course and "all-or-nothing" property of the action potential.
3. Space dependence is first analyzed for passive membranes (i.e. membranes without voltage dependent channels), resulting in the cable equation (core-conductor theory). The cable equation explains the interaction of axial and cross membrane currents.
4. Propagation of action potentials is modeled by combining the cable equation with the kinetic channel models derived from the voltage clamp experiments.

The work sketched out in the chapter is a basis of a most active field of neuroscience. In different parts of the nervous system as well as in muscle fibers and sensory systems, a large number of different *ion channels* have been identified whose kinetics and switching behavior are still modeled along the lines pioneered by Hodgkin and Huxley. In the *patch clamp* preparation, electrical properties of tiny membrane patches, containing individual channels, can be studied, thus combining in one preparation the much more macroscopic voltage- and space-clamp techniques discussed above. Electrical properties of large neurons with bifurcating dendrites, soma, and bifurcating axons, are studied in *compartmental modeling* in which the neuron's anatomy is broken down into a tree of cylindrical compartments each of which follows the electrical theory presented here (Rall 1962, Hines & Carnevale 1997).

1.7 Suggested Reading

Books

Aidley, D. J. (1998). *The Physiology of Excitable Cells*. 4th edition, Cambridge University Press, Cambridge, UK.

De Schutter, E. (ed.) (2010). *Computational Modeling Methods for Neuroscientists*. The MIT Press, Cambridge, MA.

Gerstner, W. and Kistler, W. (2002). *Spiking Neuron Models. Single Neurons, Populations, Plasticity*, Cambridge University Press, Cambridge, UK.

Jack, J. J. B., Noble, D., and Tsien, R. W. (1975). *Electric Current Flow in Excitable Cells*. Clarendon Press, Oxford.

Murray, J. D. (2002). *Mathematical Biology – An Introduction*. 3rd edition, Springer Verlag, Berlin. (Section 7.5)

Koch, C., and Segev, I. (1998). *Methods in Neuronal Modeling. From Ions to Networks.* 2nd edition, The MIT Press, Cambridge, MA.

Tuckwell, H. (1988). *Introduction to Theoretical Neurobiology (2 Vols.)*. Cambridge University Press, Cambridge.

Original Papers

Hines, M. L. and Carnevale, N. T.(1997) The NEURON simulation environment. *Neural Computation* 9, 1179 – 1209

Important review paper on compartmental modeling, introducing both the basic concepts and the simulation tools.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology (London)*, 117:500 – 544.

Classical account of the mechanism and theory of the action potential, which by and large describes the state of the art to this day. The present chapter closely follows this masterpiece of computational neuroscience.

Milescu, L. S., Yamanishi, T., Ptak, K., Mogri, M. Z., and Smith, J. C. (2008). Real time kinetic modeling of voltage-gated ion channels using dynamic clamp. *Biophysical Journal* 95:66 – 87.

This paper combines the "dynamic clamp" (replacement of specific channels by real-time computed currents) and Markovian modeling of channel switching dynamics.

Naundorf, B. Wolf, F., and Volgushev, M (2006) Unique features of action potential initiation in cortical neurons. *Nature* 440:1060 – 1063.

Action potentials in cortical neurons are shown to rise even steeper than predicted in the Hodgkin-Huxley theory (initially developed for the squid). The authors suggest an additional interaction between adjacent sodium channels not considered in the original theory.

Rall, W. (1962). Electrophysiology of a dendritic neuron model. *Biophysical Journal*, 2:145 – 167.

Early and influential paper on the electrodynamics of (passive) conduction and their dependence on dendritic geometry. Essential for deeper understanding of dendritic summation and compartmental modeling.

Wilders, R. (2007). Computer modelling of the sinoatrial node. *Medical & Biological Engineering & Computing*, 45:189 – 2007

Review of models of neural activity in pacemaker neurons of the heart. Rhythmic activity can occur in Hodgkin-Huxley systems with high leak currents. The paper summarizes and compares specific models of the sinoatrial pacemaker oscillations.

Chapter 2
Receptive Fields and the Specificity of Neuronal Firing

Abstract. Neuronal firing does not occur at random. In the sensory parts of the brain, firing is triggered by properties of various input stimuli, such as the position of a light stimulus in the visual field, the pitch of a tone, or the appearance of a familiar face. In “associative” areas of the brain, specificities for more abstract concepts have been found including cells representing place (e.g., in rodent hippocampus), or numerosity (e.g., in primate prefrontal cortex). In the motor parts of the brain, neurons have been found that fire preferably prior to pointing movements of the arm into a certain direction. That is to say, these neurons are specific for particular motor actions. In the sensory domain, specificities are quantified in terms of the receptive field, which can be defined as the totality of all stimuli driving a given neuron. The receptive field is measured by correlating the activity of a single neuron with externally measurable parameters of the stimulus. This approach is known as reverse correlation, since stimuli will always preceed the neuronal activity. The concept of correlation between neuronal activity and external measurables, however, generalizes easily to the motor system, leading to the concept of the motor field of a neuron. In this sense, visual receptive fields can be considered as an example for neuronal specificity at large. In this chapter, we discuss the basic theory of visual receptive fields which can be extended to similar concepts in other sensory, motor, or associative areas. The theory is closely related to linear systems theory applied to spatio-temporal signals, i.e. image sequences. Mathematically, it rests on integral equations of the convolution type which will be introduced in due course.

2.1 Spatial Summation

2.1.1 Correlation and Linear Spatial Summation

The Superposition Principle

If an electrode is placed in the retina, spikes can be recorded from retinal ganglion cells. Clearly, the spiking activity of these cells will depend on the stimulus, i.e. on the pattern of light shone on the receptor layer of the retina while recording. In the

H.A. Mallot: *Computational Neuroscience*, SSBN 2, pp. 23–55.
DOI: 10.1007/978-3-319-00861-5_2

simplest case, a little spot of light is moved across the retina covering just a few receptors at any one time. Usually, rather than shining the light directly onto the receptor cells, one will have the animal watch a projection screen onto which the stimulus is presented. The receptive field is then defined as the portion of the visual field, from which the firing activity can be modulated by means of the visual stimulus (see Figure 2.1). Mathematically, this is described by a receptive field function $\phi(x,y)$ which for each position of the stimulating spot of light (x,y) specifies the elicited response of the neuron.

What happens if we use larger spots of light, or, more interestingly, patterned stimuli? If we just use two spots of light, the simplest idea is that the activities that are elicited by each of the lights in isolation, are added together. This scheme is called "linear spatial summation" or "linear superposition". Let us describe the stimulus image, i.e. the spatial distribution of light intensity on the screen, by the function $I(x,y)$ which takes values between 0 (no light, black) and 1 (brightest possible light, white). A square light spot with intensity 1 covering the interval from $x=0$ to $x=1$ and $y=0$ to $y=1$ can be described by the function

$$d(x,y) := \begin{cases} 1 & \text{if } 0 \leq x < 1 \text{ and } 0 \leq y < 1 \\ 0 & \text{otherwise} \end{cases} . \tag{2.1}$$

We will call this function the *pixel function* in the sequel.

A spot of light appearing at location (x_o,y_o) would then be given by

$$I(x,y) = d(x-x_o, y-y_o). \tag{2.2}$$

It is worthwhile verifying this equation with some numerical examples. Indeed, the idea that functions can be shifted by subtracting appropriate constants in their argument is important for the understanding of this chapter.

Let us now assume that a light stimulus delivered at location (x_1,y_1) elicits a response ϕ_1 of the neuron, and that a light stimulus delivered at location (x_2,y_2) elicits a response ϕ_2:

$$I(x,y) = d(x-x_1, y-y_1) \;\;\Rightarrow\;\; e = \phi_1 \tag{2.3}$$

$$I(x,y) = d(x-x_2, y-y_2) \;\;\Rightarrow\;\; e = \phi_2 \tag{2.4}$$

where e denotes the excitation of the neuron, measured in spikes per second. In the case of linear spatial summation, presenting both spots together will elicit the sum of the individual responses:

$$I(x,y) = d(x-x_1, y-y_1) + d(x-x_2, y-y_2) \;\Rightarrow\; e = \phi_1 + \phi_2. \tag{2.5}$$

Similarly, if we increase the light intensity of the spot by a factor of $\lambda \in \mathrm{I\!R}$, a linear summation predicts a λ-fold output:

$$I(x,y) = \lambda d(x-x_1, y-y_1) \;\;\Rightarrow\;\; e = \lambda\phi_1. \tag{2.6}$$

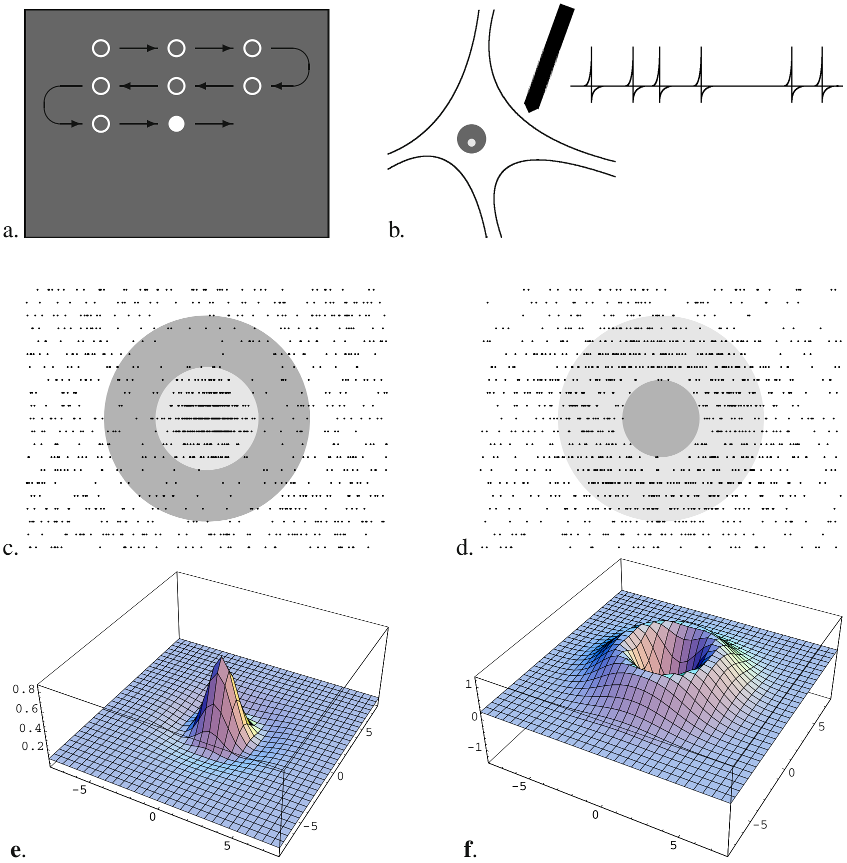


Fig. 2.1 Visual receptive fields defined as an experimental protocol. **a.** The visual field is scanned line by line with a small spot of light. **b.** Simultaneously, neural activity is recorded from a retinal ganglion cell. **c., d.** Each time the neuron fires, a point is plotted at the location of the light spot at the time of firing. **c.** A neuron which is active if the light falls within a central disc and is inhibited if the light falls in the outer annulus (on-center off-surround organization). **d.** A neuron which is inhibited if the light falls within a central disc and is activated if the light falls in the outer annulus (off center on surround organization). The receptive field is the total area, from which the neuron's activity can be modulated. The areas shown in gray are rough estimates of the excitatory and inhibitory parts of the receptive fields. **e., f.** Center-surround receptive field function $\phi(x,y)$ for the two neurons depicted in Figures c, d. The gray areas c, d are delimited by contour lines of ϕ.

Linear spatial summation is a theoretical concept defined by the above two equations. It is useful to describe the behavior of neurons as long as the stimulus intensities are not too large. I.e., if one spot of light already suffices to drive the cell into saturation, additivity will not obtain. Also, since neither light intensities nor spike rates can take negative values, Equation 2.6 becomes meaningless for negative λ. However, even in clear non-linear cases, as will be discussed in later sections of this chapter, linear descriptions are used as first approximations or components of a more comprehensive model.

The Receptive Field Function

Let us now turn back to the problem of full, complex input images. We may think of the image as a set of pixels each of which corresponds to a light spot $d(x-x_i, y-y_j)$ with intensity $I_{ij} = I(x_i, y_j)$ and write down the trivial equation[1]

$$I(x,y) = \sum_{i=1}^{I} \sum_{j=1}^{J} I_{ij}\, d(x-x_i, y-y_j), \tag{2.7}$$

where I and J are the total numbers of pixels in the x- and $y-$ direction.

Assume now that we have measured the neuron's response to an isolated light spot delivered at each of the pixel locations x_i, y_j and denote the result by $\phi(x_i, y_j)$. Then, by the linear superposition principle, we can construct the response as the sum of all responses to the individual pixels and obtain

$$e = \sum_{i=1}^{I} \sum_{j=1}^{J} I(x_i, y_i)\phi(x_i, y_i). \tag{2.8}$$

Finally, if we assume very large numbers of pixels n and very small pixels, we may write

$$e = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x', y')\ \phi(x', y')\ dx'\ dy' \tag{2.9}$$

where the integral is taken over the entire visual field.

Equation 2.9 is called the *correlation* of I with ϕ (or vice versa). The function $\phi(x,y)$ is called the *receptive field function* or *response profile* of the neuron. For each point (x,y) in the visual field, $\phi(x,y)$ is the response of the neuron, elicited from this location. Equation 2.9 is a so-called improper integral which evaluates to a number ($e \in \mathbb{R}$). It can be thought of as the volume under the two-dimensional "surface" $I(x', y')\phi(x', y')$ where x', y' represent the visual field. The visual field is thus modeled as an infinite plane over which the integral is taken. Since the receptive

[1] The sum sign $\sum$ (capital Greek letter sigma) is an abbreviation defined as $\sum_{i=1}^{I} z_i := z_1 + z_2 + \ldots + z_I$ where $I \in \mathbb{N}$ is the number of components of the sum. The double sum is needed here to cover all pixels of a rectangular grid.

field function can be assumed to be zero outside a finite "support" region (the receptive field), the existence of the integral is not a problem.[2] The name "correlation" for Equation 2.9 is based on the idea that for each location (x',y') the two numbers $I(x',y')$ and $\phi(x',y')$ form a data pair. If we omit the subtraction of the means and the normalization with the standard deviations, Equation 2.9 indeed describes the statistical correlation between the variables I and ϕ.

For the receptive field function, further names are in use, including kernel and operator. "Kernel" is a general term for a fixed factor (ϕ) in an integral equation where a second factor (I) is considered a variable input function. An operator is a mapping from a set of input functions (images) to a set of output functions (pattern of neural activity as are studied below). Since all linear operators can be expressed as an integral equation with a suitable kernel (Riesz representation theorem of functional analysis), the term operator is sometimes also used for the kernel.

The transition from the spatially discrete formulation in Equation 2.8 to the continuous formulation in Equation 2.9 can also be applied to Equation 2.7. In this case, the pixel-function $d(x,y)$ has to be replaced by the so-called δ- or Dirac[3] impulse which can be thought of as the limit of a sequence of pixel functions with decreasing pixel size and increasing amplitude, such that the volume remains constantly one. It is mathematically defined by the relation

$$\int_{-\infty}^{\infty} \delta(x) f(x) dx = f(0) \text{ for all functions} f. \tag{2.10}$$

From this definition, it immediately follows

$$f(x) = \int_{-\infty}^{\infty} \delta(x-x') f(x') dx', \tag{2.11}$$

i.e. correlation with $\delta(x-x')$ "cuts out" the function value at position x. In the next section, we will introduce the notion of convolution. Eq. 2.11 then says that convolution with the δ-Pulse does not change the input function f; i.e., it is the neural element of the convolution operation.

Strictly speaking, $\delta(x)$ is not a proper function, because it does not take a finite value at $x = 0$. Mathematically rigorous definitions are provided in the theory of functionals (functional analysis).

Figure 2.1e,f shows the function ϕ for typical retinal ganglion cells. It has circular symmetry and is divided into a center and a surround in which the function ϕ takes different signs. Negative values of ϕ mean that the cell is inhibited when a stimulus is delivered at that location. In measurements, inhibitory regions show up by a reduction of the spontaneous activity (cf. Figure 2.1) or by interactions with

[2] In functional analysis, all functions $f(x)$ for which $\int_{-\infty}^{\infty} f^2(x)dx$ exists (i.e., evaluates to a number) are contained in the *Hilbert space* L^2. Since the visual field is actually not the infinite plane but rather a finite subset of the sphere, all continuous functions defined on the visual field satisfy this condition.

[3] Paul A. M. Dirac (1902 – 1984). English physicist. Nobel Price in Physics (with E. Schrödinger) 1933.

a second, excitatory stimulation. The on-center/off-surround profile shown in Figure 2.1e is also known as "Mexican hat function". It is usually fitted by a difference of two Gaussian[4] functions, as explained in Section 2.2.1.

Visual receptive fields generally do not respond to homogeneous light distributions, i.e. stimuli of the form $I(x,y) \equiv I_o$. In Equation 2.9, this amounts to the constraint

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi(x,y)\; dy\; dy \approx 0. \tag{2.12}$$

This means that for most receptive field functions, the total weights of excitatory and inhibitory regions cancel out.

The receptive field function in the correlation operation can also be considered as a template for a particular sub-pattern or "image feature" for which the neuron is a detector. In the case of an off-center isotropic neuron appearing in Figure 2.1f, the neuron will be maximally responsive if the image depicts a black dot in front of a bright background with the diameter of the dot matching the diameter of the receptive field center. The functional description of the "optimal stimulus" (i.e. the stimulus most strongly driving the neuron) looks just like the receptive field function itself. The correlation operation is therefore also known as a "matched filter" for the image feature described by the neuron's receptive field function. Mathematically, this is an instance of the the "Cauchy-Schwarz-inequality", stating that the correlation equation is maximized if the two functions involved have the same shape. An early account of this idea is given in the famous paper entitled "What the frog's eye tells the frog's brain" (Lettvin et al. 1959) which describes retinal ganglion cells as "bug detectors" signaling to the brain dark dots in front of bright backgrounds, i.e. putative flies.

2.1.2 Lateral Inhibition: Convolution

So far, we have considered a single neuron together with its receptive field. We now turn to the case where a sensory surface (e.g., the retina) projects to an entire layer of neurons each of which has its own receptive field. Clearly, this is the case found in many sensory systems, in particular if representations are organized as topological maps.

Figure 2.2 shows a simple circuitry known as *lateral inhibition*. The sensory input shown in the top part is propagated to a set of neurons in a retinotopic layer. In addition, each connection line branches to make inhibitory inputs to the nearest neighbors of each neuron. If a constant input is given to the network, direct excitatory influences and indirect inhibitory influences cancel and the output layer will be silent. If, however, patterned input is given to the network, intensity edges will be enhanced.

Assume that a point stimulus is delivered to one input location of the lateral inhibition network of Figure 2.2. As a result, a distribution of excitation and inhibition will arise in the output layer which is positive for the neuron directly innervated

[4] Carl Friedrich Gauß (1777 – 1855). German mathematician.

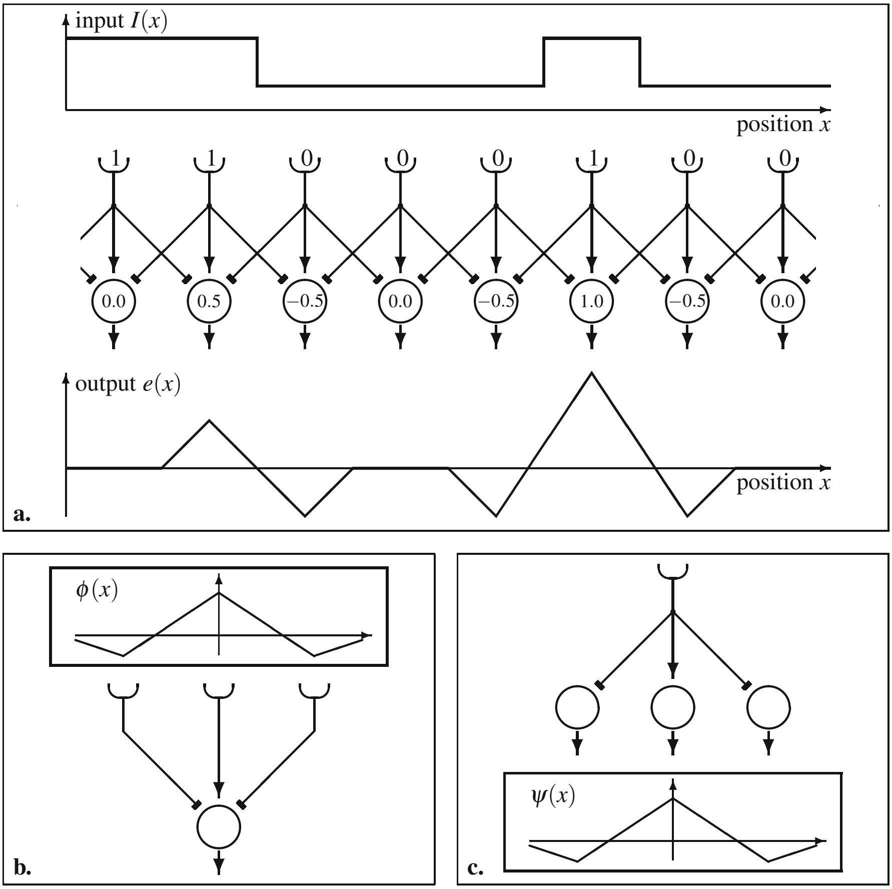


Fig. 2.2 Lateral inhibition. **a.** Input-output relations for a simple example. Links ending in an arrow are excitatory (weight $+1$); links ending in a dash ($\dashv$) symbolize inhibitory connections (weight -0.5). The input shows a step edge (left) and a contour (right) which are enhanced in the output. Constant input is attenuated. **b.** Convergence of activity to one output neuron. The activity of this neuron is characterized by its receptive field function $\phi(x)$. **c.** Divergence of activity from one input site. The resulting distribution of activity over the output layer is the point-spread function $\psi(x)$. The system is shift-invariant, resulting in a close relation between point-spread-function and receptive field function, i.e. $\psi(x) = \phi(-x)$.

from the input site and negative for its neighbors which receive the lateral, inhibitory connections. In general, we will denote such distributions as $e(x,y)$ where (x,y) is the location of a neuron and $e(x,y)$ its activity. The particular distribution resulting from a point-stimulus is called the *point spread function* or *point image*; we will denote it by $\psi(x,y)$. If we have a homogeneous layout of neurons each with the same local connectivity pattern, as is the case in Figure 2.2, the point spread functions for different stimulation sites will be identical up to a shift in position ("translation invariance"). As in the previous section, we may write:

$$I(x,y) = d(x-x_1,y-y_1) \quad \Rightarrow \quad e(x,y) = \psi(x-x_1,y-y_1) \tag{2.13}$$
$$I(x,y) = d(x-x_2,y-y_2) \quad \Rightarrow \quad e(x,y) = \psi(x-x_2,y-y_2), \tag{2.14}$$

where ψ is the point spread function for a point stimulus delivered at position $(x,y) = (0,0)$.

For general images composed of many point stimuli, we obtain:

$$e(x,y) = \sum_{i=1}^{I} \sum_{j=1}^{J} I(x_i,y_j)\,\psi(x-x_i,y-y_j), \tag{2.15}$$

or, in the infinitesimal formulation

$$e(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x',y')\ \psi(x-x',y-y')\ dx'dy'. \tag{2.16}$$

Equation 2.16 is known as the convolution[5] of I with ψ. We also write $e = I * \psi$ or $e(x,y) = (I * \psi)(x,y)$. It is again an improper integral. A space-variant version of Equation 2.15 was first introduced into neuroscience as a model of lateral inhibition in the horseshoe crab (*Limulus*) eye and is therefore also known as the Hartline[6]-Ratliff equation (Hartline & Ratliff 1958).

In a two-layer feed-forward network, convolution describes the divergence of neural activity. Other examples include the blurring of a projected slide (I: slide; ψ: blurring disk) or the superposition of sand hillocks formed by sand trickling through holes in a board (I: the pattern of holes in the board; ψ: sand hillock formed under one individual hole). Interestingly, convolution describes also the probability density function of a sum of two random variables, which is obtained by convolving the two individual density functions.

The variables in Equation 2.16 can be interpreted in the following way: Cortical coordinates (coordinates on the output layer) are given by (x,y); (x',y') parameterize the retina, or input layer, over which the integral is taken. $\psi(x-x',y-y')$ then gives the strength, or weight, with which a stimulus delivered at retinal position (x',y') influences the output at (x,y). Mathematically, it is easy to show that convolution is commutative, i.e. that we can write

$$\begin{aligned} &\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x',y')\ \psi(x-x',y-y')\ dx'dy' \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x-x'',y-y'')\ \psi(x'',y'')\ dx''dy''. \end{aligned} \tag{2.17}$$

This result is achieved by the substitution $x-x' =: x''$ and $y-y' =: y''$. Now, the intuition is slightly different: while (x,y) is still the position in the output layer, (x'',y'') is the stimulus offset relative to the current receptive field center and the integral is taken over the entire receptive field. The input function I has to be evaluated at

[5] The German term “Faltung” is sometimes also found in English texts.

[6] Haldan K. Hartline (1903 – 1983). American biologist. Nobel Price in Physiology or Medicine 1967.

$(x-x'', y-y'')$, i.e. the receptive field center in absolute coordinates $((x,y))$ minus the offset. In the end, this is just a difference in the parameterization of the later influences, not a different operation. Generally, convolution operations are characterized by the occurrence of the integration variable in both components, but with opposite signs.

Lateral inhibition was first studied by Ernst Mach[7] in relation to a perceptual phenomenon now known as Mach bands. Continuous intensity changes are perceived as roughly homogeneous areas whereas boundaries between constant intensity areas and intensity ramps are perceived as bright or dark bands, depending on the direction of the kink between the plateau and the ramp. Mach suggested that the retina calculates local second derivatives with respect to space, resulting in an excitation pattern of the form

$$e(x,y) = c\left(\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}\right). \tag{2.18}$$

The term in brackets, i.e. the sum of second partial derivatives, is known as the “Laplacian”[8] of the function $I(x,y)$ and closely related to the convolution with a center surround system. Indeed, the network depicted in Figure 2.2 can be considered as calculating a spatially discretized approximation of a (one-dimensional) second derivative. If the intensity function is equidistantly sampled to values $I_1, I_2, I_3, \ldots$, the local derivatives can be approximated as $I_1' = I_2 - I_1$ and $I_2' = I_3 - I_2$. The second derivative is then approximated as

$$I_1'' = I_2' - I_1' = I_3 - 2I_2 + I_1 \tag{2.19}$$

which is exactly the operation of Figure 2.2 up to a factor $-1/2$. The analogy between the formulations of lateral inhibition by the convolution integral or the partial differential equation (2.18) is reflected by modeling center surround receptive fields as Laplacians of Gaussians.

2.1.3 Correlation and Convolution

Let us now look at the receptive fields of the output neurons in the lateral inhibition network. By inspection, we see that they are identical to the point-spread functions. In order to compare the two types of functions, both of which allow to describe the system behavior, we need to extend Equation 2.9 to layers. We note that $\phi(x',y')$ was the receptive field function of the sole neuron considered in the derivation of Equation 2.9. Let us now assume that this neuron is located at $(x,y) = (0,0)$ in the output layer. In translation invariant systems, the receptive fields of other neurons in that layer will be shifted versions of the original neuron’s receptive field. Specifically, a neuron at location (x,y) will have a receptive field $\phi(x'-x, y'-y)$. We can therefore generalize Equation 2.9 to layered systems:

[7] Ernst Mach (1838 – 1916). Austrian physicist and philosopher.

[8] Pierre-Simon Marquis de Laplace (1749 – 1827). French mathematician.

$$e(x,y) = \iint I(x',y')\phi(x'-x,y'-y)\,dx'dy'. \tag{2.20}$$

By comparing Equation 2.16 with Equation 2.20 for all possible images I, we find that

$$\phi(x',y') = \psi(-x',-y'), \tag{2.21}$$

i.e. the point-spread function and the receptive field function are mirrored versions of each other. Note that this result holds only for translation-invariant systems. In this case, the point-spread function describes the divergence in a network and the receptive field functions the convergence in the same network. Clearly, divergence and convergence have the same origin, i.e. lateral connectivity (cf. Figure 2.2b,c).

If translation-invariance does not obtain, i.e., if receptive fields of neighboring neurons differ in systematic ways, point-spread function and receptive field function will also differ systematically (see Mallot et al. 1990). Figure 2.3 shows a projection between two retinotopically organized neural layers with varying magnification factor. If an input neuron is connected to many output neurons, point-spread functions will be large and receptive fields will be small (upper part of Figure 2.3b). Vice versa, if an input neuron projects only to a few output neurons, point-spread functions will be small and receptive fields large. Similarly, large visual areas such as V1 tend to have small receptive fields and large point-spread functions, whereas small areas such as MT tend to have large receptive fields and small point-spread functions.

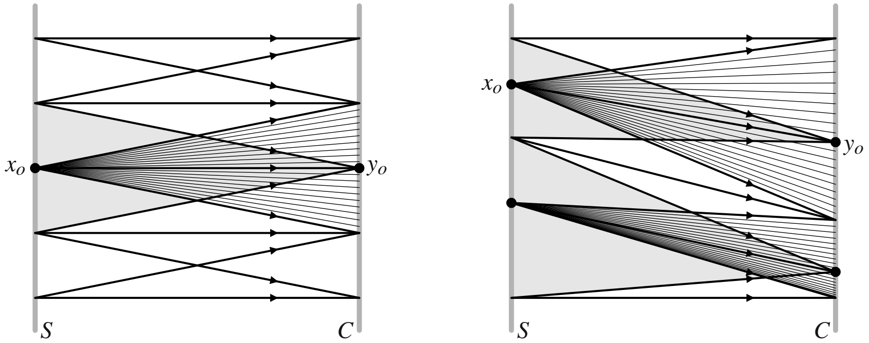


Fig. 2.3 Receptive fields (shaded) and point images (hatched) in space-invariant (left) and space-variant (right) projections between a sensory surface S and a central representation C. The receptive field of a unit $y_o \in C$ is a part of the sensory surface. The point image of a point $x_o \in S$ is a part of the central representation. In the space-invariant case (left; convolution), receptive field and point image do not change over space. In the space-variant case (right) magnified regions (upper part) have large point images and small receptive fields, whereas reduced regions (lower part) have small point images and large receptive fields.

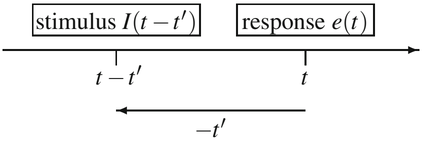


Fig. 2.4 Interpretation of the time variables in Equation 2.22. Note that the naming of time variables can be changed by the transformation $\tau := t-t', t-\tau = t'$. In the equation, the roles of I and w will be exchanged, demonstrating the commutativity of convolution.

2.1.4 Spatio-Temporal Summation

Neurons collect activity not only over space but also over time. To understand how this is modeled in linear summation, we first need to consider spatio-temporal stimuli, i.e. movies. Such stimuli specify an intensity value for each image point and each instant in time and may therefore be represented by a three-dimensional function $I(x,y,t)$. The activity of the neuron at time t will in principle depend on the complete movie up to time t. Just as we divided image space into pixels, we may now conceptualize time as being divided into discrete time steps, as is indeed the case in movies or video clips. The stimulus is now a large, three-dimensional arrangement of individual events, i.e. light flashes, each with the size of one pixel and lasting one time step. Let t' denote the time that passed between the instant of recording from the neuron, t, and the spatio-temporal stimulus event whose influence on the neuron is considered. Clearly, this event took place at $t-t'$.

In this situation, spatio-temporal summation is characterized by a spatio-temporal weighting function or kernel $w(x,y,t')$ specifying for each position (x,y) and each delay t' the sign and strength with which this particular stimulus event influences the output (cf. Figure 2.4). As in spatial summation, we assume translation invariance, which in the temporal domain is also called stationarity. It means that the system will produce the same dynamic response every time the same stimulus is presented. In neural systems, deviations from stationarity may be due to fast changes such as fatigue and adaptation, or to slower processes such as plasticity, growth, or ageing. In the stationary case, the summation of the influences reads:

$$e(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{o}^{\infty} I(x,y,t-t')\; w(x,y,t')\; dt'\; dy\, dx. \tag{2.22}$$

The function $w(x,y,t)$ is called the spatio-temporal receptive field profile.

Note that space and time are treated differently in this equation. Temporally, we have a convolution, reflecting the fact that the neuron will be influenced by the temporal evolution of the stimulus. In space, we simply sum over the entire receptive field. A spatial convolution would be required only if a uniform layer of neurons with identical, but space-shifted receptive fields were to be modeled. The inner integral in Equation 2.22 is taken over positive values of t' only. This is in contrast to spatial convolution, where x' and y' varied between $+$ and $-\infty$. The reason for this restriction is of course that only stimuli occuring before the time of measurement can influence the excitation. This constraint is called "causality". It can also be accounted for by setting the values of $w(x,y,t')$ to 0 for all values $t' < 0$, meaning that an input event cannot have effects in the past. With this convention, the integral may be taken from $-\infty$ to ∞.

An important special case of Equation 2.22 is obtained if we consider a purely spatial summation, followed by a temporal summation process applied only to the result of the spatial process. In this case, the spatio-temporal receptive field function can be split up into two parts, a spatial profile $\phi(x,y)$ and a temporal weighting function $g(t)$:

$$w(x,y,t) = \phi(x,y)g(t). \tag{2.23}$$

The contribution of a stimulus delivered at time $t - t'$ to the neuron's activity at time t is given by a function $g(t')$ where t' specifies how much time has passed between the delivery of the stimulus and the instant t at which response is measured. Generally, $g(t')$ will be small for large delays t' and maximal for small or intermediate values of t'. Spatio-temporal kernels that can be split up in this way into a spatial and a temporal factor are called "separable".

2.1.5 Peri-Stimulus Time Histogram (PSTH) and Tuning Curves

Spatio-temporal receptive field functions provide complete descriptions of a neuron's behavior, at least if linearity and translation-invariance in time is assumed. In practical experiments, however, other descriptors such as the peri-stimulus time histogram and the tuning curve are more commonly used. In this section, we will briefly discuss how these measurements relate to the receptive field function, again for the linear case.

The peri-stimulus time histogram is the average time-dependent rate of action potentials (or spike rate) measured during a stimulus presentation cycle. Let the single spike-train measured during the i-th stimulus presentation be $(t_{i1}, t_{i2}, ..., t_{in_i})$ where n_i is the total number of spikes in that spike-train and t_{ij} denotes the time when the j-th spike in the i-th trial occurred. Then, the spike rate is a piecewise constant function of time; between two subsequent spikes, it evaluates to the inverse of the inter-spike interval:

$$p_i(t) := \frac{1}{t_{i,j+1} - t_{ij}} \quad \text{for} \quad j = 0, ..., n_i - 1. \tag{2.24}$$

The PSTH can be calculated as the pointwise average of this function over all m stimulus presentation cycles,

$$\mathrm{PSTH}(t) = \frac{1}{m} \sum_{i=1}^{m} p_i(t). \tag{2.25}$$

Assume now that the neuron is stimulated by a spatio-temporal pattern $I(x,y,t)$ lasting for some duration T. After each presentation, an inter-stimulus-interval (ISI) T_{ISI} is included during which the presentation screen is left blank. We need to assume that the ISI is long enough for the response to decay to zero, i.e. $w(x,y,t) = 0$

for $t > T_{\text{ISI}}$. In this situation, the time course of the neuronal response $e(t)$ as calculated from Equation 2.22 is the linear prediction of the PSTH.

The PSTH of a given neuron will of course be different for different stimuli. In many experiments, the stimuli used can be thought of as a parameterized family of functions, with a parameter specifying the orientation of a bar or grating, the speed of a moving bar, the length or a bar etc. For each value of the parameter, a PSTH $e_p(t)$ results where p denotes the parameter. From this PSTH, we calculate a characteristic number such as the total number of spikes, the maximal spike rate, etc. The dependence of this number on the parameter is known as the tuning curve of the neuron for the stimulus parameter considered. For the maximal spike rate, we obtain

$$\rho(p) := \max_t e_p(t), \tag{2.26}$$

where the Greek letter ρ (rho) denotes the tuning curve. Alternatively, tuning may be defined as the total number of spikes elicited with each parameter setting,

$$\rho(p) := \int_0^{t_{max}} e_p(t)dt. \tag{2.27}$$

Clearly, peri-stimulus time histogram and tuning curve are measuring protocols which remain to be useful and well-defined if the linear model of the receptive field presented here fails.

2.2 Functional Descriptions of Receptive Fields

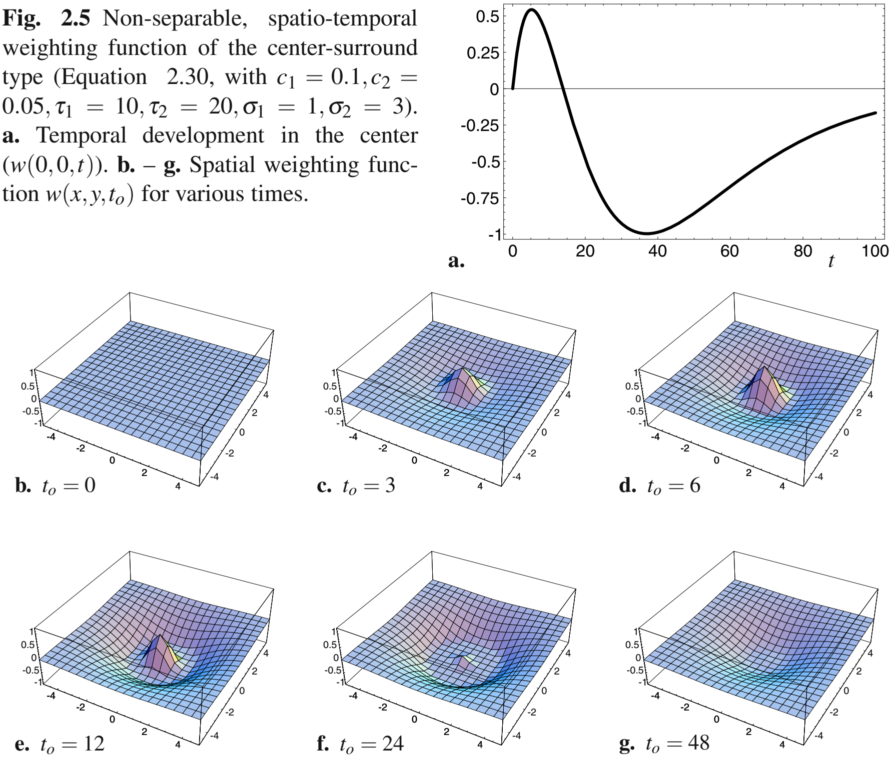
2.2.1 Isotropic Profiles: Gaussians

So far, the examples of receptive field functions were isotropic, or circular symmetric functions, as are found in retinal ganglion cells, and the lateral geniculate nucleus (LGN). They are usually modeled using the well known Gaussian function

$$\phi(x,y) = \frac{1}{2\pi\sigma^2}\exp\left\{-\frac{1}{2\sigma^2}(x^2+y^2)\right\}. \tag{2.28}$$

The factor $1/(2\pi\sigma^2)$ has been chosen such that the integral over the entire function equals unity. The only free variable is the scale σ which determines the width of the bell-shaped surface (see Figure 2.7b). Receptive fields differing in scale will react to stimulus features of different size or granularity. Contour lines of the Gaussian are curves satisfying the relation $(x^2+y^2)/2\sigma^2 = const$, i.e. they are circles. In particular, the contour line at elevation $e^{-1/2}\phi(0,0)$ has radius σ. In one-dimensional versions, i.e. sections along the coordinate axes, the arguments $\pm\sigma$ mark the inflection points of the bell-shaped curve.

Adding a temporal component to the receptive field function, we choose the function $g(t)$ from Equation 2.22 to be



b. $t_o = 0$ **c.** $t_o = 3$ **d.** $t_o = 6$

e. $t_o = 12$ **f.** $t_o = 24$ **g.** $t_o = 48$

Fig. 2.5 Non-separable, spatio-temporal weighting function of the center-surround type (Equation 2.30, with $c_1 = 0.1, c_2 = 0.05, \tau_1 = 10, \tau_2 = 20, \sigma_1 = 1, \sigma_2 = 3$). **a.** Temporal development in the center ($w(0,0,t)$). **b.** – **g.** Spatial weighting function $w(x,y,t_o)$ for various times.

$$g(t) = \frac{t}{\tau}\exp(-\frac{t}{\tau}). \tag{2.29}$$

Here, τ is called a time constant. The function is 0 for $t = 0$, rises to its maximum at $t = \tau$ and declines asymptotically to zero for larger t. If the neuron reacts fast, it will have a short time constant, while slow responses may be modeled by larger time constants.

The standard model of the center-surround type receptive field of retinal ganglion cells can now be formulated:

$$w(x,y,t) = c_1 t e^{-t/\tau_1} \exp\left\{-\frac{x^2+y^2}{2\sigma_1^2}\right\} - c_2 t e^{-t/\tau_2} \exp\left\{-\frac{x^2+y^2}{2\sigma_2^2}\right\}. \tag{2.30}$$

It is a difference of two Gaussians with unequal width, where each Gaussian is multiplied with a temporal summation function. For an on-center, off-surround ganglion cell, we may choose an excitatory mechanism with small width σ_1 and small time constant τ_1 and an inhibitory mechanism with larger width $\sigma_2 > \sigma_1$ and larger time constant $\tau_2 > \tau_1$. A function with these specifications is illustrated in Figure 2.5. Leaving the temporal component out, Equation 2.30 reduces to a so-called "difference of Gaussians" (DoG) or "Mexican hat" function plotted in Figure 2.1e,f.

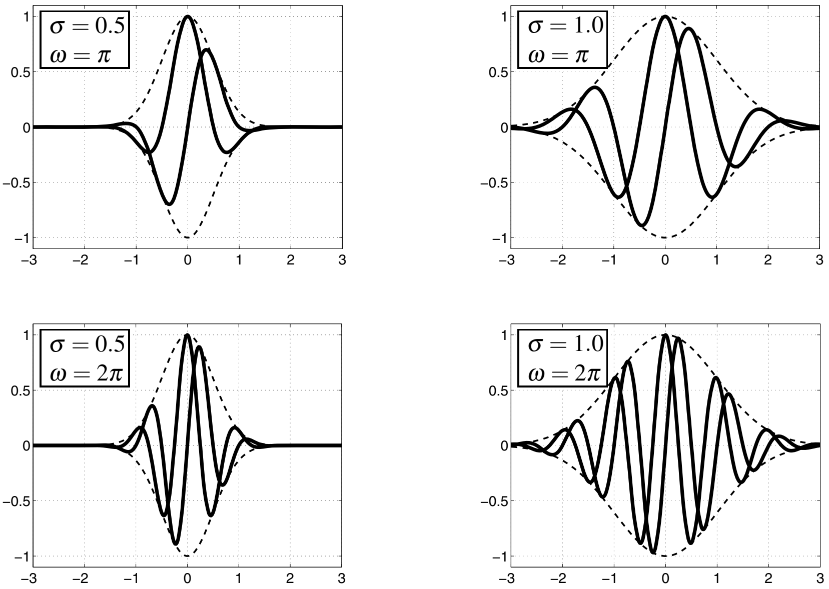


Fig. 2.6 One-dimensional Gabor functions (Equation 2.31, 2.32) for various choices of scale σ and frequency ω. In each plot, the even (left-right mirror symmetric) heavy line is the cosinusoidal Gabor function, the odd (point symmetric heavy line) is the sinusoidal Gabor function. The dashed lines show the enveloping Gaussians, Note that the top right and bottom left plots are identical up to a horizontal compression. This reflects the fact that in these cases the product $\sigma\omega$ is the same. It is proportional to the number of "side lobes" included in the function.

Note that Equation 2.30 is also an example of a non-separable spatio-temporal function, as long as $\tau_1 \neq \tau_2$. Other types on spatio-temporal non-separability include spatio-temporal orientation (i.e. receptive fields shifting over time) or changes in preferred orientation.

2.2.2 Orientation: Gabor Functions

In the visual cortex, most receptive fields are oriented, i.e., their response to elongated stimuli (light bars) changes if these bars are presented vertically, horizontally, or at any other orientation.

Mathematically, there are different ways of generating oriented receptive field functions that can be used to model such behavior. The function type which is now almost exclusively used is the so called Gabor[9] function, generated by multiplying a (one- or two-dimensional) sinusoidal with a Gaussian envelope (see Pollen & Ronner 1983, Jones & Palmer 1987). In one dimension, we have:

[9] Dennis Gabor (1900 – 1979). Hungarian physicist (holography). Nobel Prize in Physics 1971.

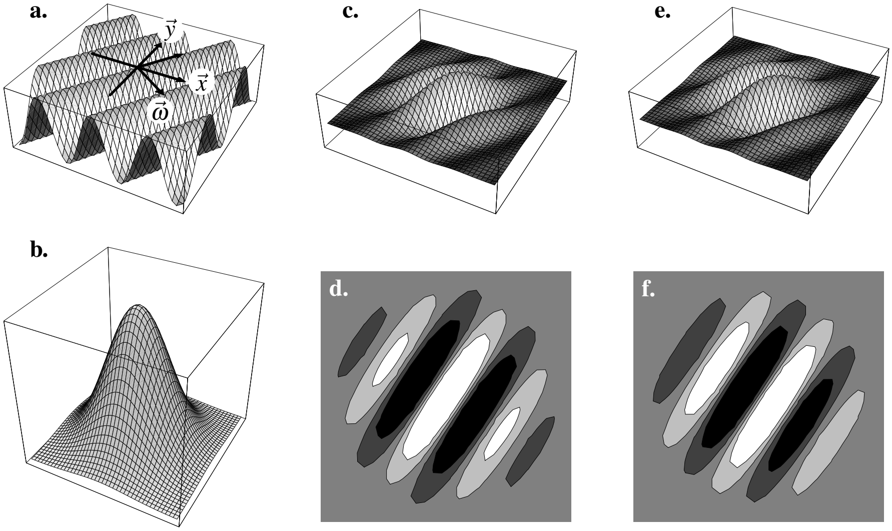


Fig. 2.7 **a.** Plane wave $f(x,y) = \cos(\omega_x x + \omega_y y)$. The vectors on top show the coordinate axes $(\vec{x},\vec{y})$ and the wave direction $\vec{\omega} = (\omega_x, \omega_y)$. The unlabeled vector is the orthogonal direction $(-\omega_y, \omega_x)$ along which f is constant. **b.** Gaussian function $\varphi(x,y) = \exp\{-(x^2 + y^2)/2\sigma^2\}$. **c.** Cosine Gabor function obtained by multiplying f and φ. **d.** Same as c, shown as contour plot. **e.** Sine Gabor function obtained by phase-shifting f by a quarter cycle and multiplying it by φ. **f.** Same as e, shown as contour plot.

$$g_c(x) := \cos(\omega x)\exp\left\{-\frac{x^2}{2\sigma^2}\right\} \tag{2.31}$$

$$g_s(x) := \sin(\omega x)\exp\left\{-\frac{x^2}{2\sigma^2}\right\}. \tag{2.32}$$

The cosinusoidal Gabor function is "even", i.e., it satisfies the condition that $g_c(x) \equiv g_c(-x)$; the sinusoidal Gabor function is odd, $g_s(x) \equiv -g_s(-x)$ (cf. Figure 2.6). As before, σ determines the width of the Gaussian window while ω is the frequency of the underlying sinusoidal; since g_c and g_s are functions of a spatial variable, ω is called a *spatial frequency*. Gabor functions are also sometimes called wavelets, because of their local wave-shaped look.

Orientation is added when the Gabor function is extended to two dimensions. To see this, we consider first the "plane wave"

$$f(x,y) = \cos(\omega_x x + \omega_y y) \tag{2.33}$$

depicted in Figure 2.7a. It describes a corrugated surface similar to a wash board. Sections along the direction $\vec{\omega} = (-\omega_y, \omega_x)$, i.e. along a "wave-front" are constant. All other sections are sinusoidals with various frequencies; in particular, the frequencies of sections along the coordinate axes are w_x and w_y, respectively. The direction with the highest frequency is (ω_x, ω_y), i.e. orthogonal to the wave-front. The spatial frequency of the two-dimensional function $f(x,y)$ is the vector $(\omega_x, \omega_y)^\top$. As in the one-dimensional case, the Gabor function is obtained by multiplying the sinusoidal with a Gaussian, see Figure 2.7b. If the Gaussian is centered on a wave peak or trough, the result will be a symmetric, cosine Gabor function (Figure 2.7c,d); if the Gaussian is centered on a flank, an odd, or sine Gabor function will result (Figure 2.7e,f). In principle, Gabor functions with arbitrary phase relation can be generated by shifting the Gaussian with respect to the plane wave. The equations read:

$$g_c(x,y) := \cos(\omega_x x + \omega_y y) \, \exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\} \tag{2.34}$$

$$g_s(x,y) := \sin(\omega_x x + \omega_y y) \, \exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\}. \tag{2.35}$$

Gabor functions are characterized by the following parameters which model the major specificities found in neurons in the primary visual cortex:

1. In Equation 2.35, the Gabor functions are localized at position $(x,y) = (0,0)$. Shift terms (x_o, y_o) can be added to shift function to arbitrary positions.
2. The *scale* σ determines the overall width of the receptive field.
3. The preferred *orientation* is given by the ratio ω_y/ω_x. If expressed as an angle from the x-axis, orientation becomes $\tan^{-1}(\omega_x/\omega_y)$.
4. The preferred *spatial frequency* is the frequency of a grating at any orientation driving the cell most strongly. This frequency is determined by $\sqrt{\omega_x^2 + \omega_y^2}$. Spatial frequency preference is sometimes referred to as "localization in the spatial frequency domain".
5. The *number of side lobes* is determined by the product of the overall spatial frequency and scale.

2.2.3 Spatio-Temporal Gabor Functions

In the discussion of motion selectivity, we will encounter spatio-temporal Gabor functions. In these, the plane wave of Figure 2.7a is replaced by a propagating wave,

$$g_c(x,y,t) = \cos(\omega_x x + \omega_y y - \nu t). \tag{2.36}$$

Here, ν is a temporal frequency (dimension $1/\text{sec}$) and the the speed of movement in the (ω_x, ω_y)-direction is $\nu / \sqrt{\omega_x^2 + \omega_y^2}$.

The Gaussian window function also has to be replaced by some spatio-temporal window function. Clearly, in time, this function can only extend into the past, since the future stimuli are unknown ("causality"). A sensible choice for the spatio-temporal window function uses a logarithmically distorted time axis. In order to apply the scale variable σ to both the space and the time domain, we need to make the space and time variables dimensionless, i.e. divide them by their respective units.

We call the new variables ξ, ζ, θ (read xi, zeta, and theta) and obtain (Koenderink 1988):

$$w(\xi,\zeta,\theta) = \exp\left\{ \frac{(\xi - \xi_o)^2 + (\zeta - \zeta_0)^2 + (\ln(-\theta) - \ln(-\theta_o))^2}{2\sigma^2} \right\} \quad \text{for} \;\; \theta, \theta_o < 0. \tag{2.37}$$

$\theta_o < 0$ is the moment in the past, onto which the temporal window is centered. The temporal window will be asymmetric, compressed in the forward direction (between θ_o and 0) and expanded in the backward direction. Its overall size will shrink the closer θ_o is to 0, i.e., the present moment.

The drifting grating of Equation 2.36 is then multiplied with the spatio-temporal window function to generate a spatio-temporal Gabor function.

2.2.4 *Why Gaussians?*

Why are Gaussians and Gabor functions used to describe receptive fields? In principle, the choice of a particular mathematical function as a model of a receptive field will be based on the ability of the function to fit the receptive field profile. Gaussians and Gabor functions, however, while well suited to fit experimental data, are also chosen for mathematical reasons. The Gaussian is special in that convolving a Gaussian with another Gaussian yields again a Gaussian ("central limit theorem"). Also, the Fourier transform of a Gaussian (see below) is again a Gaussian. The Gabor function is optimally localized both in space and in spatial frequency. Indeed, in the 1970s, neuroscientists controversially discussed the question whether cortical neurons are tuned to (localized in) space or spatial frequency, assuming that both ideas were mutually exclusive. The introduction of the Gabor function settled this question. The Fourier transform of the Gabor function is a Gaussian centered at a non-zero frequency.

2.3 Non-linearities in Receptive Fields

Linear neurons are a theoretical abstraction. They are special in that they are completely defined by one spatio-temporal receptive field function which predicts the neuron's response to all possible stimuli by means of convolution. In contrast, non-linearity is not a well-defined concept in itself, but simply the absence of linearity, which can occur in many ways. Before discussing some important deviations from linearity, we start with a general definition.

2.3.1 *Linearity Defined: The Superposition Principle*

In the previous sections, we have modeled simple spatial receptive fields by a weighting function ϕ. A stimulus occurring at position (x,y) is multiplied by the local weight $\phi(x,y)$ and the weighted contribution is added to the neuron's response. More generally, we will now introduce the notation

$$e = \Phi(I), \tag{2.38}$$

where $I = I(x,y)$ is the stimulus (an image) and Φ is a general *mapping* from the set of all images to the set of possible excitations e of the neuron. In mathematics, a mapping from a function set into a number set is called a *functional*, the corresponding discipline of mathematics is called *functional analysis*[10]. A functional Φ is said to be *linear* if the following two requirements are satisfied (see also Equation 2.5, 2.6):

1. The response to the sum of two stimuli $(I(x,y), J(x,y)$ is the sum of the individual responses:
$$\Phi(I+J) = \Phi(I) + \phi(J). \tag{2.39}$$
Inserting from Equation 2.9, we obtain
$$\begin{aligned}\Phi(I+J) &= \int\!\!\int \phi(x,y)\ (I(x,y)+J(x,y))\ dx\,dy \\ &= \int\!\!\int \phi(x,y)\ I(x,y)\ dx\,dy + \int\!\!\int \phi(x,y)\ J(x,y)\ dx\,dy \\ &= \Phi(I) + \phi(J),\end{aligned} \tag{2.40}$$
i.e. this condition is obtained from the distributive law and the linearity of the integral.
2. The response to a multiple of the stimulus is the multiple of the response:
$$\Phi(\lambda\ I) = \lambda\ \Phi(I) \ \text{ for all } \ \lambda \in \mathrm{I\!R}. \tag{2.41}$$
Again, we verify this condition from Equation 2.9:
$$\begin{aligned}\Phi(\lambda\ I) &= \int\!\!\int \phi(x,y)\ (\lambda\ I(x,y))\ dx\,dy \\ &= \lambda\ \int\!\!\int \phi(x,y)\ I(x,y)\ dx\,dy\ =\ \lambda\ \phi(I),\end{aligned} \tag{2.42}$$
i.e. this condition holds due to the associative law of multiplication and the linearity of the integral.

These two properties taken together are known as the superposition principle. In words, it states that in linear systems, responses to individual stimuli superimpose, or add up, if the stimuli are presented together.

If I is taken to be the image intensity, i.e. the physical power (irradiance) impinging per unit area on the retinal receptors, the response of a neuron can never be strictly linear. Consider for example a stimulus leading to some excitation e. Linearity requires that the neuron would react to the same stimulus, amplified by a factor of

[10] If e is also considered a function, as was the case in the discussion of lateral inhibition, Φ in Equation 2.38 becomes a mapping from a set of functions (input images) into another set of functions (pattern of neural excitation). Such mappings are called *operators*. The definition of linearity is extended to operators in a straight-forward way.

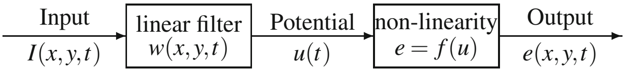


Fig. 2.8 An important class of non-linear system can be described as a cascade of a linear system, followed by a static non-linearity. In neuroscience, L–NL-cascades are considered as a coarse model of dendritic summation followed by spike initiation at the axon hillock.

λ, with a λ-fold activity λe. Clearly, this will work only for small positive values of λ, since the activity of the neuron is limited and cannot become negative. Linearity, therefore, is often a good model as long as small stimuli (or small deviations from some standard stimulus) are considered but usually fails for large signal amplitudes. Even then, however, the linear case is always considered as a first approximation of the problem.

Note that non-linearity and translation-invariance (see Figure 2.3) are independent concepts. A system may be linear but not translation invariant, or vice versa. In either case, however, it cannot be described by convolution. Indeed, it can be proven that all linear, translation-invariant systems are convolution systems.

2.3.2 Static Non-linearity

The simplest type of non-linearity is described by terms like thresholding, saturation, or compression and can be modeled as a sequence, or cascade of a linear system and a so-called static non-linearity, or transfer function. In neural networks, neurons are often considered as linear–non-linear (L–NL) cascades, where the (approximately) linear part is dendritic summation, whereas spike generation at the axon-hillock is non-linear, but depends only on the instantaneous value of the generator potential. While potential can be below or above resting potential (hyper- vs. depolarized soma), spike rates as described in Equation 2.24 cannot be negative. The static non-linearity will therefore be a mapping from the real numbers $\mathbb{R}$ (the potentials) to the non-negative reals $\mathbb{R}_o^+$. In summary, the cascade system can be written as

$$e(t) = f(u(t)) \tag{2.43}$$

$$u(t) = \iiint I(x,y,t-t')\; w(x,y,t')\; dx\, dy\, dt' \tag{2.44}$$

(cf. Figure 2.8).

The function f is called a static non-linearity, because it acts on individual points and instances of $u(x,y,t)$, irrespective of the values at adjacent points in space or instances in time. Typical examples of static non-linearities are shown in Figure 2.9. Half-wave rectification simply sets negative values to zero:

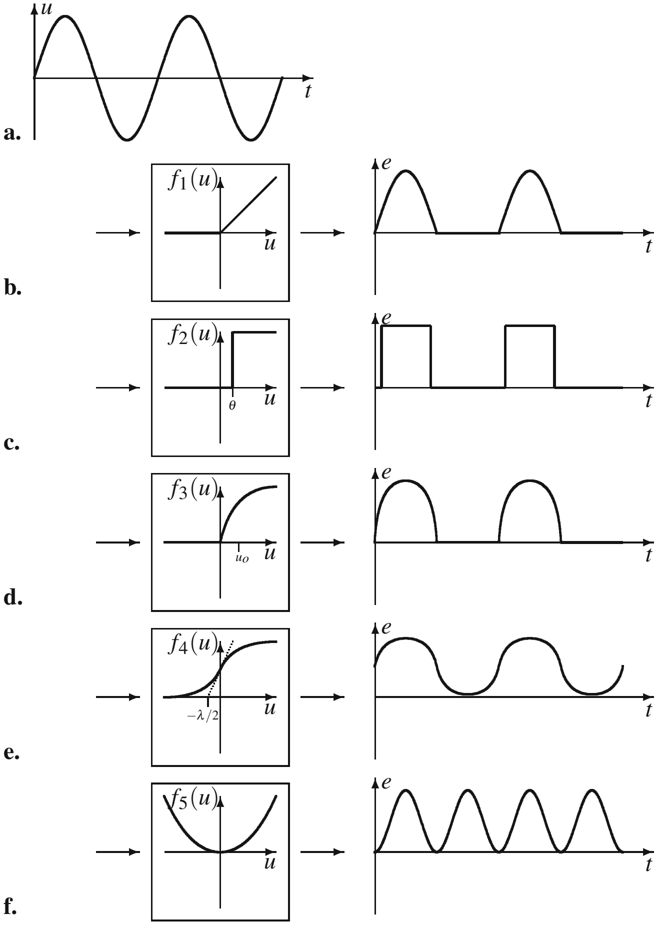


Fig. 2.9 Static non-linearity applied to a sinusoidal. **a.** Sinusoidal used as an input in all cases. **b.** Half-wave rectification. The boxed plot shows the static non-linearity function $f(u)$, the right part the effect on the sinusoidal input, i.e. $f(\sin u)$. Half-wave rectification deletes all negative values and passes the positive values unchanged. **c.** Binary switching function (Heaviside function) with threshold θ. **d.** Saturation function with half-saturation value u_o. **e.** Sigmoidal with slope λ. The value $f(0) = 1/2$ is the spontaneous activity of a neuron. **f.** Squaring non-linearity doubles the frequency of a sinusoidal input (since $\sin^2 t = 0.5 - 0.5\cos 2t$). It plays an important role in so-called energy models of cortical complex cells (see below).

$$f_1(u) := \begin{cases} 0 \text{ if } u \leq 0 \\ u \text{ if } u > 0 \end{cases}, \tag{2.45}$$

it appears in Figure 2.9b.

The binary switching function (Heaviside function[11]) is defined as

$$f_2(u) := \begin{cases} 0 \text{ if } u \leq 0 \\ 1 \text{ if } u > 0 \end{cases} . \tag{2.46}$$

Figure 2.9c shows the function $f_2(u - \theta)$ where θ is the *threshold*. Note that the threshold is modeled by a subtraction in the argument of f. This can be done with all static non-linearities. In the case of a sinusoidal input, the threshold modulates the 'duty-cycle' of the binary output, i.e. the relative length of zero and non-zero sections.

Saturation is often modeled by the equation

$$f_3(u) := \begin{cases} 0 \text{ if } u \leq 0 \\ \frac{u}{u+u_o} \text{ if } u > 0 \end{cases} \tag{2.47}$$

(Figure 2.9d). The constant u_o determines the slope of the function; it can be thought of as the input value where f_3 generates the output value 0.5. Equation 2.47 has been used to model the non-linearity of photoreceptors. In this case, u_o can be used to adjust the curve to the state of light adaptation ("Naka-Rushton non-linearity").

The static non-linearity used most frequently in neural network modeling is the sigmoidal (Figure 2.9e). It can be formalized in various ways, for example:

$$f_4(u) = \frac{e^{\lambda u}}{e^{\lambda u} + e^{-\lambda u}}, \quad \lambda > 0. \tag{2.48}$$

The parameter λ is the slope of the sigmoidal at $u = 0$. The value $f_4(0)$ models the spontaneous activity of the neuron. In the formulation given, $f_4(0) = 1/2$ for all λ. If other spontaneous activities are desired, the function can be shifted along the u-axis.

At first glance, the squaring non-linearity

$$f_5(u) = u^2 \tag{2.49}$$

does not seem particularly plausible as a model of neural transmission. It turns out, however, to be very useful in the so-called energy models of complex cells which will be discussed below. There, it can be thought of as a rectifying square (i.e., $f(u) = 0$ for $u < 0$ and $f(u) = u^2$ for $u \geq 0$) applied separately to an on and an off channel processing the same signal. The sum of the two outputs can be calculated faster by applying the two-sided squaring function to the linearly filtered input.

2.3.3 Non-linearity as Interaction: Volterra Kernels

In convolution, a stimulus can be thought to be decomposed into a set of small, instantaneous stimuli ("pixels" and "pulses") each of which elicits a response

[11] Oliver Heaviside (1850 – 1924). British mathematician and physicist.

described by the spatio-temporal weighting function w. By the superposition principle, the total response is the sum of the effects of all individual pulses.

One interesting type of non-linearity is to extend the superposition principle by assuming that there is an interaction between "pulses" occurring at different points and instants, and that the responses to such pairs of pulses superimpose. To keep the notations simple, we consider a purely temporal system, i.e. spatial summation is assumed to be absent. In this case, the cell might be driven by the co-occurrence of two stimuli at times t' and t'' an example of which is given in Figure 2.13a below. Let us denote the contribution of this stimulus pair to the cell's activity at time 0 by $K(-t', -t'')$. Assuming further that the contributions from all such stimulus pairs superimpose, we get

$$e(t) = \int\int K(t-t', t-t'')I(t')I(t'')dt'dt''. \tag{2.50}$$

The function K in Equation 2.50 is called a Volterra[12]-kernel of second order, sometimes, the term "2-way interaction kernel" is also used. It describes a general multiplicative non-linearity. For example, if K equals the Dirac impulse $\delta(t'-t'')$, e becomes the integral over the squared input function, i.e. the total signal power presented. Higher order interactions can be modeled analogously, i.e. with an n-dimensional kernel K and the product of the stimulus values at n different times (n-way interactions). The first-order Volterra equation would be (linear) convolution.

General non-linearities can be modeled by sums of Volterra-integrals of increasing order (Volterra series). This approach has in fact been used to fit neurophysiological data from complex receptive fields. The advantage of this method is that it gives a general means for the identification of non-linearities. The disadvantage lies in its huge number of unknown variables. For a spatio-temporal Volterra kernel of order n, a function of $3n$ variables has to be measured.

2.3.4 Energy-Type Non-linearity

Figure 2.10 shows four edges with the same orientation, but differing in two parameters called polarity and phase. Figure 2.10a,b shows "step edges" where the intensity function is anti-symmetric (odd) with respect to the location of the edge. Figure 2.10c,d shows "contour edges" whose intensity function is symmetric (even) with respect to edge location. In analogy to the difference between the symmetric cosine function and the anti-symmetric sine-function, the difference between step and contour edges is called a *phase difference*. Optimal filters for step edges are odd, i.e., sine-Gabor functions whereas optimal filters for contour edges are even, i.e., cosine-Gabor functions. For each phase, the edges depicted differ also in polarity, leading to the following four cases: dark-to-bright (rightward increasing step

[12] Vito Volterra (1860 – 1940). Italian mathematician.

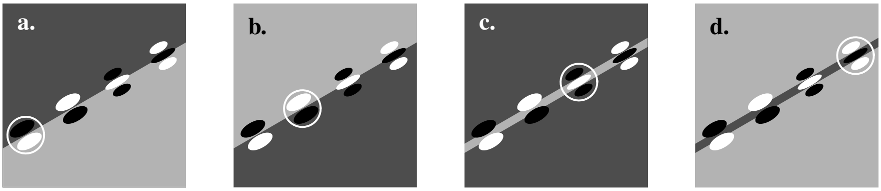


Fig. 2.10 Different types of edges and receptive fields. In each frame, four receptive fields of the simple type are shown. Black ellipses indicate inhibitory regions, white ellipses excitatory regions. The two fields to the left have odd symmetry, the two to the right even symmetry. For each symmetry type, the two possible polarities are also shown. The receptive field responsive in each case is highlighted by a white circle. **a.**, **b.**: Step edges. Only odd symmetric fields of the appropriate polarity will respond. **c.**, **d.**: Contour edges. Only even symmetric fields of the appropriate polarity will respond.

edge), bright-to-dark (rightward decreasing step edge), bright-on-dark (incremental contour), and dark-on-bright (decremental contour).

Also shown in Figure 2.10 are four receptive fields modeled as Gabor functions which can be characterized as (i) increasing, odd (g_s in Equation 2.35), (ii) decreasing odd ($-g_s$), (iii) incremental even (g_c), and (iv) decremental even ($-g_c$). All four profiles are shown for each edge and the one profile yielding the strongest response is circled. All these receptive fields are oriented and linear, i.e. they model cortical simple cells.

In contrast to simple cells, cortical complex cells respond to edges of all polarities and phase, as long as the edges have the preferred orientation (and scale), i.e. they behave invariant with respect to polarity and scale. Polarity invariance is an intrinsically non-linear property as can be seen from the following consideration:

Let $I_+(x)$ be a bright-on-dark ("incremental") contour edge as depicted in Figure 2.10c and the first column of Figure 2.11. Let the response of some linear receptive field to this stimulus be denoted by e_+. The decremental, dark-on-bright contour can then be written as $I_-(x) = I_o - I_+(x)$ where I_o is a constant. Due to the assumed linearity, the response to this stimulus must be the difference between the response to the constant intensity I_o and e_+. Since neurons will generally not respond to constant stimuli at all ($\int \phi(x,y)dxdy \approx 0$), we obtain $e_- = -e_+$ which can be satisfied only if the neuron is not responding to the edge at all. This proves that polarity-invariant neurons must be non-linear.

How, then, can we construct a model of a complex cell reacting equally well to step and contour edges of either polarity? The simplest way to achieve this is to take all four simple receptive fields shown in Figure 2.10, pass them through a rectifying non-linearity (i.e. neglect negative outputs) and take the sum of the four input lines.

$$\begin{aligned} e_{cpx} &= f(I \otimes g_s) + f(I \otimes (-g_s)) + f(I \otimes g_c) + f(I \otimes (-g_c)) \\ f(u) &= \begin{cases} u^2 \text{ for } u \geq 0 \\ 0 \quad \text{for } u < 0 \end{cases} \end{aligned} \tag{2.51}$$

where e_{cpx} is the excitation of the complex neuron, g_c and g_s are the even and odd Gabor functions, and $\otimes$ denotes the correlation (Equation 2.9). The static non-linearity f could also be modeled as simple half-wave rectification, but the squaring gives better results. Instead of writing down the on- and off-Gabor fields $\pm g_c$ and $\pm g_s$ separately, we observe $I \otimes (-g_{s,c}) = -(I \otimes g_{s,c})$ and obtain

$$e_{cpx} = (I \otimes g_s)^2 + (I \otimes g_c)^2. \tag{2.52}$$

Equation 2.52 is called the *energy model* of the cortical complex cell (Adelson & Bergen, 1985), for a schematic depiction see Figure 2.12c. The term “energy” is not to be taken literally, but simply reflects an analogy from the theory of alternate current, where the sum of squared sine and cosine components is an electrical energy. It is also related to the notion of a “signal power” introduced in the following chapter.

Although in Equation 2.52, only two receptive field functions are considered, the formulation of Equation 2.51 is biologically more plausible, since it takes into account the non-negativity of neuronal spike rates. In the final model, on- and off-profiles which individually are subject to a rectifying non-linearity are combined to a channel which behaves completely linear.

Figure 2.11 shows the energy model in more detail. The top row shows the four edge types (two phases, two polarities). These edges are meant to appear on top of some constant background, such that negative values are no problem. As before, we assume that no responses to constant inputs occur. The left column shows the receptive field profiles of the four types of simple cells, $\pm g_c(x)$, $\pm g_s(x)$. Each bar plot in the center shows the simple cell activity $\pm g_{s,c} \otimes \Delta I$, where the center bar refers to a neuron exactly centered at the edge, while the flanking bars refer to receptive fields somewhat beside the edge. The bottom row shows the activity of an energy neuron. Note that the response is the same for all four edge types.

The removal of phase and polarity information does not depend on the use of Gabor functions but can be obtained with any so-called quadrature pair of receptive field functions. A general quadrature pair consists of an odd function f_o and an even function f_e and satisfies the condition that the complex function $f_e + i f_o$, where $i = \sqrt{-1}$ is the complex unit, must be analytical, i.e. differentiable in the complex plane. Or, to state this another way, the Hilbert transform must convert the even function into the odd and vice versa. In fact, the odd/even Gabor functions only approximately satisfy this condition.

2.3.5 Summary: Receptive Fields in the Primary Visual Pathway

- Isotropic or rotationally symmetric receptive fields are found in retinal ganglion cells and the LGN. They are usually modeled as differences of Gaussians. Isotropic fields vary mostly with respect to their polarity (on-center vs. off-center) and size (scale) (Figure 2.12 a, b).
- Orientation specific receptive fields of the “simple” type are found throughout the visual cortex. In addition to polarity and scale, they vary with respect to orientation, phase (odd vs. even) and spatial frequency (Figure 2.12 c, d).

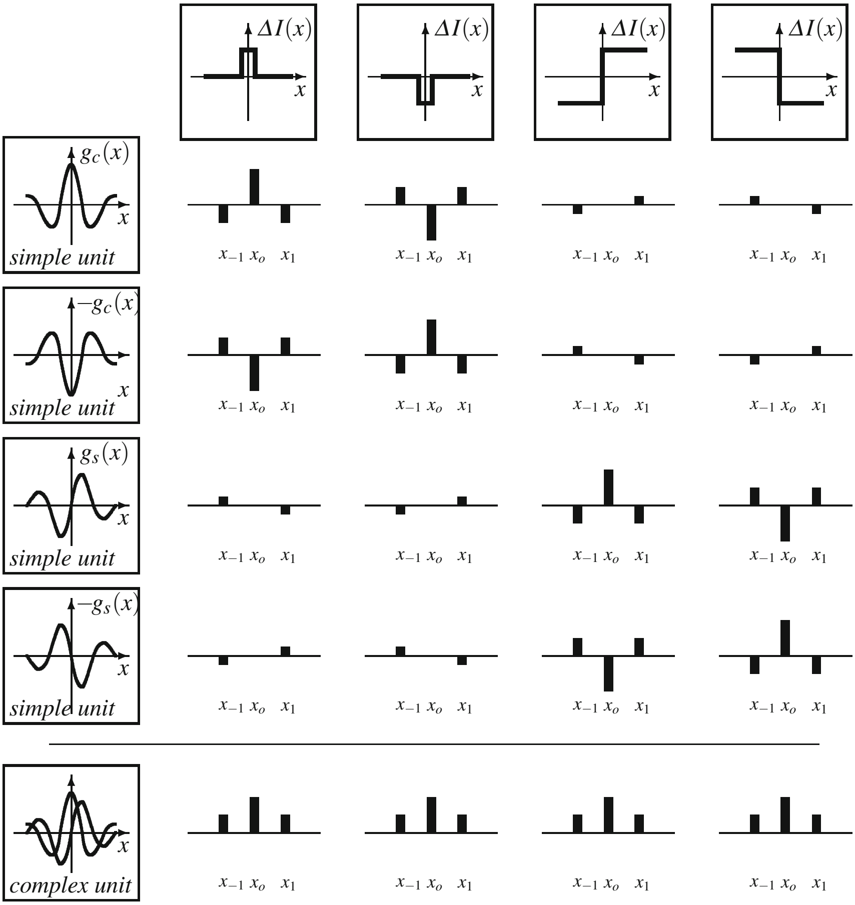


Fig. 2.11 Energy model of visual neurons of the "complex" type. The top row shows from left to right four local intensity profiles corresponding to a bright contour on dark ground, a dark contour on bright ground, a step edge from a dark to a bright image region, and a step edge from a bright to a darker image region. The leftmost column shows receptive field functions of four simple cells modeled as Gabor function with even or odd phase (sine or cosine) and positive or negative polarity. The histograms above the bar show the responses of three such cells with receptive fields centered at x_{-1}, x_o, and x_1, respectively. The row below the bar shows a complex cell summing up the squared outputs of each of the simple cells. Note that the histograms for all four input profiles are identical, indicating the complex cell's invariance to edge polarity and phase.

- Cortical neurons of the "complex" type are invariant to phase and polarity, but share with simple cells the specificities for orientation, spatial frequency, and scale. The invariance requires a non-linearity which is modeled by the energy model (Figure 2.12 e).

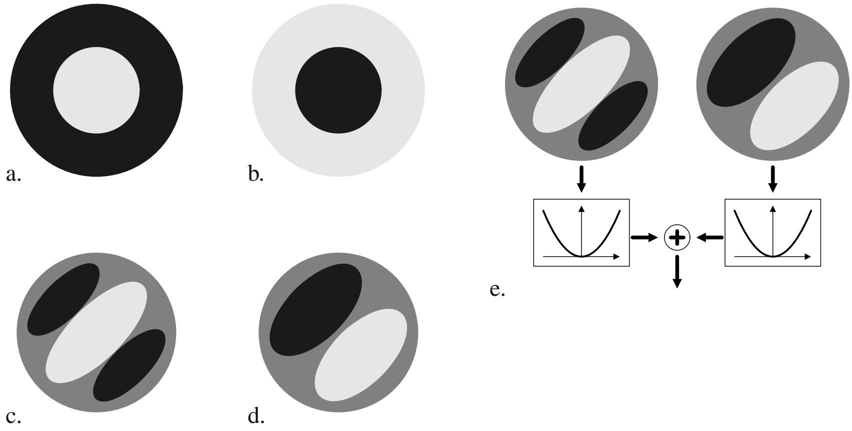


Fig. 2.12 Visual receptive fields. *a, b.* Isotropic (rotationally symmetric) types found in retina, LGN. *a.* On-center off-surround. *b.* Off-center on-surround. *c, d.* Oriented simple cells of the visual cortex *c.* Even (cosine) Gabor-function *d.* Odd (sine) Gabor-function *e.* Cortical complex neuron responsive to "contrast energy".

2.4 Motion Detection

2.4.1 Motion and Flicker

Visual motion is the most important stimulus for many neurons throughout the visual system. On the input side, motion can be defined as the coherent displacement of an image patch or an image feature over time, where the amount and direction of the displacement form the motion vector. Visual motion is thus characterized by two quantities, i.e., length (speed) and direction, or the x- and y-components of the motion vector. If we denote the local motion vector by $\vec{v}(x,y,t) = (v_x(x,y,t), v_y(x,y,t))^\top$, image change due to visual motion can be expressed by the equation

$$I(x,y,t+dt) = I(x - v_x(x,y,t)dt, y - v_y(x,y,t)dt, t) \qquad (2.53)$$

where $\vec{v}dt$ is the motion displacement in the time interval dt.

Note that not every change in an image is a motion. For example, if the light in a room is switched on and off, all pixel intensities change, but there is no displacement of image patches and therefore no motion. Likewise, the dynamic noise pattern appearing on a poorly tuned television set is not a motion stimulus despite its dynamic, i.e. ever changing structure. Image change which cannot be described as image motion is called flicker. It is a scalar quantity without a direction. In analogy to Equation 2.53, we write

$$I(x,y,t+dt) = f(x,y,t)\ dt + I(x,y,t), \qquad (2.54)$$

where f denotes the flicker.

By inspection of Equations 2.53, 2.54, it is clear that the measurement of visual motion amounts to the estimating two variables per pixel (v_x and v_y) whereas only one variable is to be measured in flicker (f). We will not pursue the role of flicker but continue with visual motion.

The fact that visual motion is a vector quantity implies that the tuning curve of a motion-selective neuron depends on two stimulus parameters (speed and direction), not just on one. With sharp tuning, the neuron operates as a detector for a particular motion, i.e., it signals whether or not its preferred motion is present. This type of coding is known as labeled line coding; it is different from a (hypothetical) motion estimator, where the output would be a continuous estimate of the two-dimensional motion vector. Clearly, no such estimator can be realized by a single neuron.

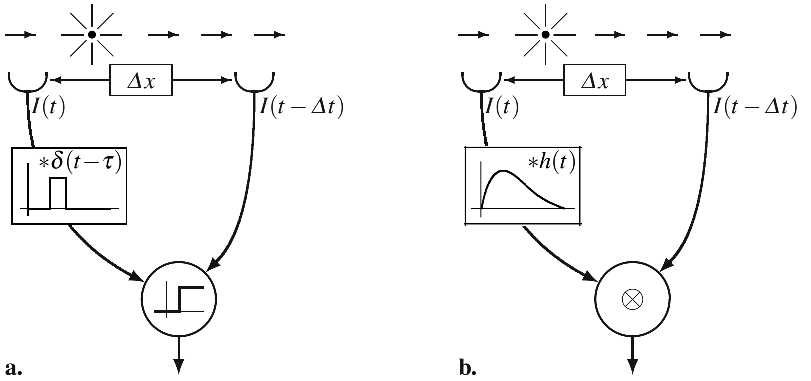


Fig. 2.13 **a.** Simple motion detector consisting of two input lines of which one is delayed by the time lag τ, and a threshold. The delay is shown as a temporal convolution with a displaced δ-pulse. The stimulus is moving from left to right with a velocity $v = \Delta x/\Delta t$, leading to the inputs $I(t)$ and $I(t-\Delta t)$ at the two receptors. If the delay τ matches Δt, the activations in both lines will reach the threshold unit simultaneously. The threshold has to be adjusted such that it is passed only by coincident inputs on both lines. The detector is specific for direction and speed. **b.** In the correlation, or Reichardt detector, the delay is replaced by a temporal lowpass filter and the threshold is replaced by a multiplication and subsequent averaging over time. The latter two steps realize a cross-correlation between the direct and the low-passed input line with offset zero (symbol $\otimes$). The complete Reichardt detector consists of two mirror symmetric circuits of the type shown, whose outputs are subtracted to give signed motion output.

2.4.2 Coincidence Detector

The most obvious approach to motion detection is to combine a thresholding operation with a delay in one of its input lines (Figure 2.13a). If a signal, such as a flash of light, appears first at the delayed input line and later at the non-delayed line, and if the time of travel matches the built-in delay, both inputs will arrive simultaneously and the threshold may be passed. If, however, the stimulus moves in the opposite direction or with the wrong speed, both inputs arrive at different times and the threshold will not be passed. This principle has been suggested as a model of motion selectivity in rabbit retinal ganglion cells (Barlow & Levick 1965).

In order to prevent the circuit from responding to constant white input on both lines, an appropriate preprocessing has to added such as a differentiation in time and maybe a Difference-of-Gaussians operator in space.

Note that the delay operation is a linear, shift-invariant operation which can be expressed as a convolution. Its impulse response is a delayed impulse, $\delta(t-\Delta t)$. Using the definition of the δ pulse, Equation 2.10, we may write

$$\int I(t')\delta(t-\Delta t-t')\,dt' = I(t-\Delta t). \tag{2.55}$$

In Figure 2.13a, the delay operation is illustrated by its impulse response.

The delay-threshold detector is tuned sharply to the velocity $\Delta x/\Delta t$ where δx is the distance between the receptive fields of its input lines and Δt is the delay included. Its velocity tuning curve thus looks just like the impulse response of the delay process.

2.4.3 Correlation Detector

An alternative design is shown in Figure 2.13b, where the delay is replaced by a temporal low-pass filter and the threshold is replaced by a correlation (correlation- or Reichardt[13]-detector, Eichner et al. (2011)). At any instant in time, the output of the temporal low-pass is a mixture of its inputs at the preceeding times, weighted by the temporal impulse response function $h(t)$ depicted in Figure 2.13b. It thus acts like the superposition of a whole set of weighted delay lines. Therefore, the comparison operation in the subsequent neuron will yield response not just to one velocity, but to many. As in the pure delay case, the tuning curve looks like the temporal impulse response function in the low-passed input line.

The comparison operation used in this case is somewhat more elaborate than simple thresholding. It is based on the comparison of the time dependent signals during a extended period of time. To understand this, we have to briefly introduce the notion of auto- and cross-correlation of functions, see Figure 2.14. In statistics, sample correlation is defined for a set of paired data $(x_i, y_i)_{i\in 1,..,n}$ as

$$\mathrm{cor}(x,y) := \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}} = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sqrt{\mathrm{var}(x)\ \mathrm{var}(y)}}, \tag{2.56}$$

where $\bar{x}$ and $\bar{y}$ are the average values of x and y, and $\mathrm{var}(x)$ and $\mathrm{var}(y)$ are the respective variances. In signal theory, it is customary to neglect the averaged values (i.e.,

[13] Werner E. Reichardt (1924 – 1992). German biologist and physicist.

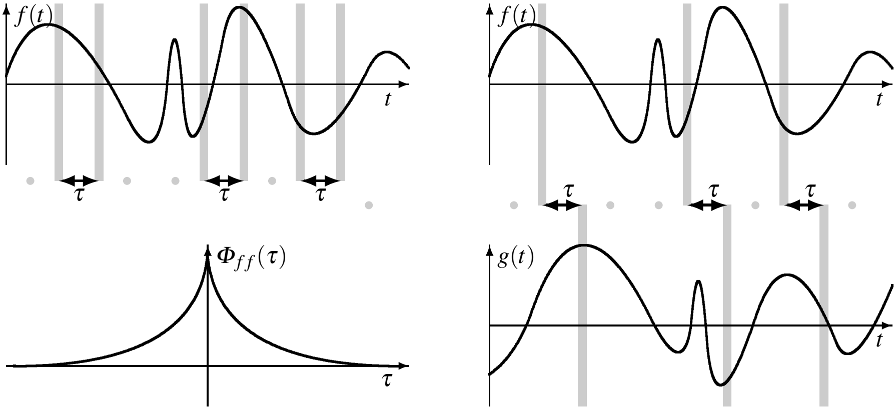


Fig. 2.14 Left: The auto-correlation for a function f and an offset τ is calculated by evaluating the function at all pairs of points with separation τ. Of the resulting pairs $(f(t), f(t+\tau)$ for all possible positions t), "correlation" is calculated by multiplying the values in each pair and summing the results. Since this procedure can be repeated for each interval τ, a so-called auto-correlation *function* results which is denoted $\Phi_{ff}(\tau)$. A typical auto-correlation function is shown in the lower part of the picture. **Right:** The same procedure can be applied to two different functions, in which case the result is called cross-correlation, $\Phi_{fg}(\tau)$.

assume $\bar{x} = \bar{y} = 0$) and omit the division by the variances.[14] For a time dependent signal $f(t)$, we can then consider the correlation of a value occurring at time t with a value occurring at time $t + \tau$ for some fixed "offset" τ. For various times t, the data pairs $(f(t), f(t+\tau))$ form a set of paired variables, for which correlation can be calculated. Since we have an infinite set of data pairs indexed by the variable t, the sum in Equation 2.56 is to be replaced by an integral and we obtain

$$\Phi_{ff}(\tau) = \int_{-\infty}^{\infty} f(t) f(t+\tau) dt. \tag{2.57}$$

In the theory of random processes, the replacement of a population average by a temporal average requires a property of the random process called ergodicity. We will assume here that ergodicity obtains. Auto-correlation as defined in Equation 2.57 can be calculated for each offset τ; thus Φ_{ff} becomes a function of the offset. It is easy to see that $\Phi_{ff}(\tau)$ takes its maximal value at $\tau = 0$, and that $\Phi_{ff}(-\tau) = \Phi_{ff}(\tau)$.

The cross-correlation function is defined by the same logic, only that the two values of the paired variables are now taken from different functions; see Fig. 2.14. For two functions f, g, it is defined as

[14] Without the normalization, the operation is actually more like a *covariance*, defined as $\mathrm{cov}(x,y) = (1/n)\sum(x_i - \bar{x})(y_i - \bar{y})$. In some texts, the auto- and cross-correlation functions are therefore called auto- and cross-covariance, respectively. We will not adopt this terminology here.

$$\Phi_{fg}(\tau) = \int_{-\infty}^{\infty} f(t)g(t+\tau)dt. \tag{2.58}$$

Cross-correlation is a means to detect delays of shifts between two related functions. For example, if $g(t) = f(t+\Delta t)$, the cross-correlation function takes its maximum at $\tau = \Delta(t)$, which can be used to estimate such delays from longer sequences of data.

In the correlation-type motion detector, the output neuron multiplies its two inputs and accumulates this product over time, i.e.

$$e = \int \underbrace{\int h(t')I(t-t')dt'}_{h*I} I(t-\Delta t)dt \tag{2.59}$$

$$= (\Phi_{II} * h)(\Delta t). \tag{2.60}$$

This is to say, the output of the motion detector is given by convolving the auto-correlation function of the input with the impulse response h and evaluating the result at τ. If the image is a white noise pattern, i.e. a noise pattern where adjacent pixels are uncorrelated, its auto-correlation function is a δ-pulse, $\Phi_{II}(\tau) = \delta(\tau)$ and the tuning curve of the motion detector will be $\rho(v) = h(\Delta x/v)$.

2.4.4 Motion as Orientation in Space-Time

The standard model of motion selectivity in the mammalian visual cortex (Adelson & Bergen 1985) can be derived from the “bi-local” detectors shown in Figure 2.13a, b by assuming that the detecting neuron receives not just two input lines, but many, each with a specific delay, and that these inputs are summed up at the soma (see Figure 2.15). The delays are adjusted such that for a particular speed of a stimulus passing by the receptors, all inputs will reach the output neuron at the same time. As before, this speed is the preferred motion speed for the neuron. Figure 2.15b shows an alternative depiction of the same idea where the stimulus is now presented in a space-time diagram with the spatial coordinate x as a horizontal axis and time as the vertical axis. The diagram thus shows the temporal development of one image line; the second spatial dimension has been left out for simplicity. In this diagram, the stimulus “optimally” driving the neuron is a row of light spots lined up along the line $t = x/v$. The same output would be generated by a continuously moving stimulus, moving with the same speed, which in Figure 2.15b would show as a bright bar covering the three spots. Velocity, in the space-time diagram, corresponds to slope, or spatio-temporal orientation.

The space-time diagram of Figure 2.15b can also be interpreted as a spatio-temporal receptive fieldfunction as discussed already in Section 2.2.3. If we allow the input lines from each spatial position to have full-fledged temporal impulse responses, rather than just a delay element, a full spatio-temporal receptive field results as is shown in Figure 2.15c. It depicts the spatio-temporal Gabor function defined by Eqs. 2.36 and 2.37, without the logarithmic compression of the time axis. As

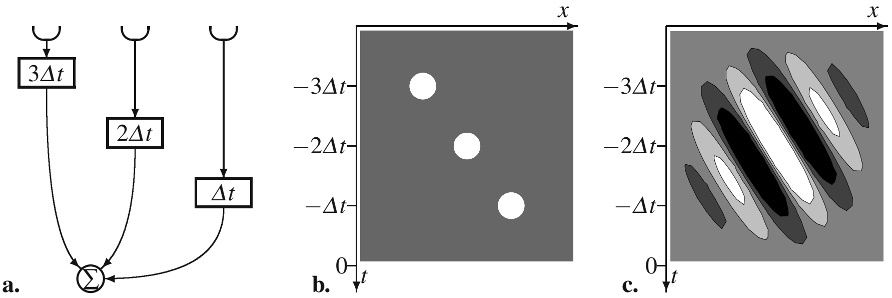


Fig. 2.15 Motion selective receptive fields. **a.** A hypothetical neuron with three equally spaced input sites, each with a different temporal delay (rectangular boxes). **b.** (x,t)-diagram of a stimulus flashing subsequently at the three input positions of the neuron. With the shown timing and the delays shown in a., all activities arrive simultaneously at the unit. **c.** (x,t)-diagram of a spatio-temporal Gabor function (Equation 2.35 modeling the receptive field of a motion selective simple cell. The orientation in the space-time plot corresponds to the preferred velocity.

a receptive field function, it describes the response behavior of a simple cell tuned for motion (rightwards), scale, spatial frequency, location, phase (cosinusoidal), and polarity. Spatial orientation cannot be shown in the figure due to the restriction to two dimensions, but is also included in the mathematical model.

For motion selective complex cells, the energy model (Section 2.3.4) is applied analogously. It is built on two simple cells, one with sinusoidal and one with cosinusoidal phase, but with identical value for all other parameters. The outputs are squared and summed up as in shown in Figure 2.12d above.

2.5 Suggested Reading

Books

De Valois, R. L. and De Valois, K. K. (1988). *Spatial vision*. Oxford Psychology Series No. 14. Oxford University Press, Oxford, UK.

Frisby, J. and Stone, J. V. (2010). *Seeing: The computational approach to biological vision*. The MIT Press, Cambridge, MA.

Mallot, H. A. (2000). *Computational Vision. Information Processing in Perception and Visual Behavior.* The MIT Press, Cambridge, MA.

Marr, D. (1982). *Vision. A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, San Francisco.

Rolls, E. T. and Deco, G. (2002). *Computational Neuroscience of Vision*. Oxford University Press, Oxford, UK.

Original Papers

Adelson, E.H. and Bergen, J.R. (1985). Spatiotemporal energy models for the detection of motion. *Journal of the Optical Society of America A* 2:284-299

Fundamental account of visual motion as spatio-temporal orientation. Derives suitable spatio-temporal filters for motion detection and suggests biologically plausible version constructed from spatio-temporally separable filters.

Itti, L. and Koch, C. (2000) A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research* 40:1489 – 1506

Saliency of visual field locations is defined using the overall output of a large battery of visual filters corresponding to the so-called perceptual dimensions orientation, granularity, motion, and color. The model is thus an application of the receptive field theory explained in this chapter. It is still one of the standard approaches to focal attention.

Mach, E. (1865) Über die Wirkung der räumlichen Vertheilung des Lichtreizes auf die Netzhaut. *Sitzungsberichte der mathematisch-naturwissenschaftlichen Classe der kaiserlichen Akademie der Wissenschaften Wien* 52/2, p.303 – 322. An English translation (On the effect of the spatial distribution of the light stimulus on the retina) appears in F. Ratliff, *Mach Bands: Quantitative Studies on Neural Networks in the Retina*, Holden-Day, San Francisco, 1965.

This century-old paper is still recommended reading for its clear and clever argument as well as for its general discussion of psychophysics and the relation between neural mechanisms and experienced perceptions. It introduces the effect now known as Mach bands.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society (London) B*, 207:187 – 217.

This paper intuitively shows how center-surround mechanisms support early visual processes. At the same time it gives a comprehensive account of the Difference-of Gaussian and Laplacian-of-Gaussians operators. Together with a series of similar treatments of other visual processing steps covered in D. Marr's book (see above), it belong to the foundations of the field of computational vision.

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

This paper shows how receptive field organization in the visual cortex can be derived from optimality criteria introduced on the statistical considerations such as sparseness of coding, redundancy reduction etc.

Srinivasan, M. V., Laughlin, S. B., and Dubs, A. (1982) Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society London B*, 216:427 – 459.

Center-surround receptive field organization is shown to be optimal in the sense that redundancies in visual coding are removed. Diameters of center and surround parts are derived from the average auto-correlation function of the visual input.

Chapter 3
Fourier Analysis for Neuroscientists

Abstract. In this Chapter, we introduce a piece of mathematical theory that is of importance in many different fields of theoretical neurobiology, and, indeed, for scientific computing in general. It is included here not so much because it is a genuine part of computational neuroscience, but because computational and systems neuroscience make extensive use of it. It is closely related to systems theory as introduced in the previous chapter but is also useful in the analysis of local field potentials, EEGs or other brain scanning data, in the generation of psychophysical stimuli in computational vision and of course in analyzing the auditory system. After some instructive examples, the major results of Fourier theory will be addressed in two steps:

- Sinusoidal inputs to linear, translation-invariant systems yield sinusoidal outputs, differing from the input only in amplitude and phase but not in frequency or overall shape. Sinusoidals are therefore said to be the “eigen-functions” of linear shift invariant systems. Responses to sinusoidal inputs or combinations thereof are thus reduced to simple multiplications and phase shifts. This is the mathematical reason for the prominent role of sinusoidals in scientific computing.
- The second idea of this chapter is that any continuous function (and also some non-continuous functions) can be represented as linear combinations of sine and cosine functions of various frequencies. Alternatively to the use of sine- and cosine functions, one may also use sinusoidals with a phase value for each frequency, or complex exponentials from the theory of complex numbers.

Both ideas combine in the convolution theorem, stating that the convolution of two functions can also be expressed as the simple product of the respective Fourier transforms. This is also the reason why linear shift-invariant systems are often considered “filters” removing some frequency components from a signal and passing others.

H.A. Mallot: *Computational Neuroscience*, SSBN 2, pp. 57–81.
DOI: 10.1007/978-3-319-00861-5_3

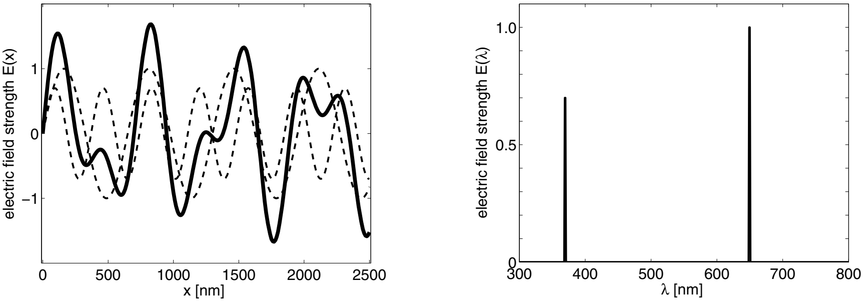


Fig. 3.1 **a.** Distribution of electric field strength in two primary colors (blue and red, dashed lines) and in the resulting mixture (magenta, solid line). **b.** Spectra of the pure colors red and blue. The spectrum of the mixture includes both lines.

3.1 Examples

3.1.1 Light Spectra

The notion of a spectrum in the physics of light is closely related to the ideas of Fourier[1] transform. Light is an electromagnetic wave which can be described by the electric and magnetic fields as a function of space and time. Consider a beam of coherent light in direction x. At a given instant in time, the corresponding fields are smooth functions of spatial position x. In pure, or "spectral" colors, both the electric and the magnetic field strength oscillate according to a sinusoidal function of x, generally specified by its wavelength λ. Figure 3.1a shows the electric field distribution of a blue (short wavelength, high frequency) and a red (long wavelength, low frequency) light (dashed lines). If we superimpose both lights, we will experience the color magenta, or purple. The electric field distribution is the sum of the according distributions of the two basic colors red and blue (Figure 3.1a, solid line). Note that this distribution is no longer a sinusoidal; it need not be periodic at all. Still, it can be thought of as being composed of two sinusoidal components with different frequencies. Figure 3.1b shows the spectral representation of the same lights. Here, for each wavelength λ the amplitude of the sinusoidal with this particular wavelength is shown. For the primaries red and blue, these are line-spectra, i.e. the spectrum is zero except for the wavelengths $\lambda_r = 650$ nm and $\lambda_b = 370$ nm. The spectrum of the purple light is the sum of the two line spectra.

3.1.2 Acoustics

In acoustics, time-dependent sound pressure is measured by a microphone and can be visualized with an oscilloscope. For example, when playing the tone e' on a treble recorder, the oscilloscope will show a fairly clean sinusoidal wave with frequency

[1] Jean Baptiste Joseph Fourier (1768 – 1830). French mathematician and physicist.

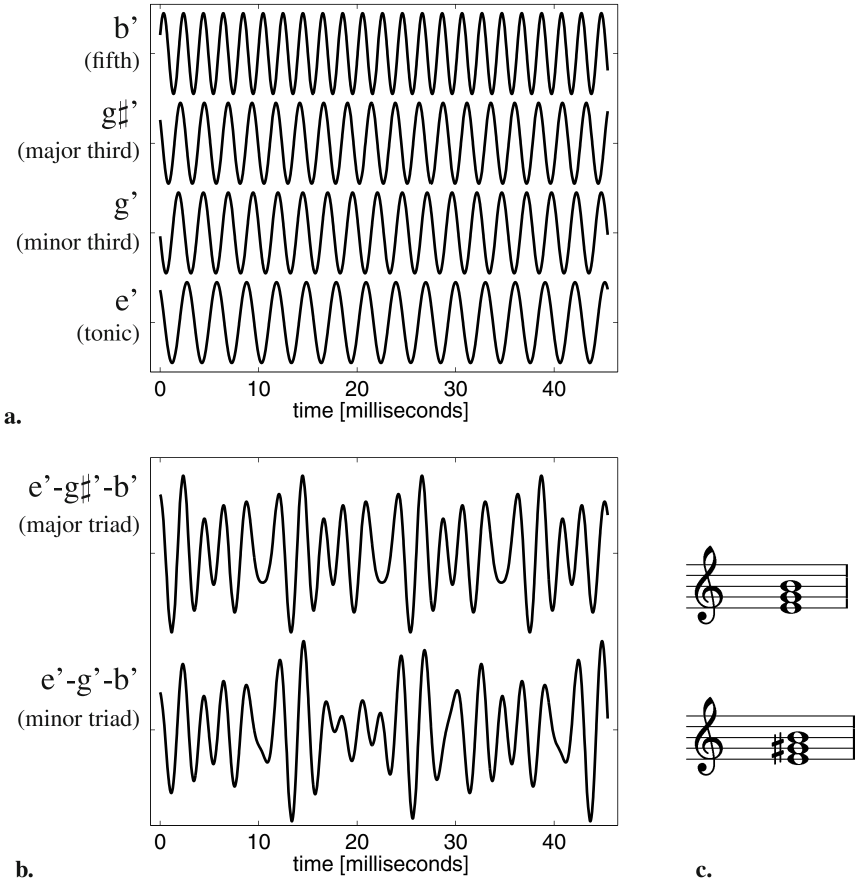


Fig. 3.2 Time- and frequency signals in music. **a.** Sound pressure of a pure tone at $\omega_o =$ 330Hz, i.e. e' (tonic), the minor third g' ($\omega/\omega_o = 6/5$), the major third g♯' ($\omega/\omega_o = 5/4$), and the fifth b' ($\omega/\omega_o = 3/2$). The frequency ratios correspond to the just, or pure intonation. The phases of the signals have been randomized. **b.** Sound pressure for the e-minor and e-major triads e'-g'-b' and e'-g♯'-b'. **c.** Musical notation for the same two triads. The staff and clef define a reference frame for pitch which is akin to a logarithmic frequency scale.

330 Hz. Figure 3.2a shows the time-dependent signal for the tone e' together with the signals for the tones g', g♯', and b', all of which are sinusoidal waves differing in frequency. If these tones are played together, the sound pressure signals add up to the pattern shown in Figure 3.2b. In music, these pattern correspond to two well-known chords, called the e-minor and e-major triads. When such chords are played and reach the ear of a listener, the cochlea of the inner ear will decompose the complex time signals of Figure 3.2b into the individual tones they are composed of. Mathematically, this operation corresponds to a Fourier decomposition of the

compound signal resulting in an acoustic spectrum with peaks at the respective frequencies. The musical notation shown in Figure 3.2c represents this spectrum by the individual note symbols.

When playing the same tone e' on a piano, the microphone signal will again be a periodic function with frequency 330 Hz, but the shape of the wave will be less sinusoidal. Representing the sound pressure function as a frequency spectrum will now result in a peak at 330 Hz plus additional peaks at some or all integer multiples of 330 Hz. These multiples are known as harmonics. The pattern of harmonics generates the "timbre" of the sound, i.e. the differences between productions of the same tone by various musical instruments or the voices of different singers. Complex pattern of harmonics known as "formants" make the differences between vowels such as an /a/ or an /ee/ sung with the same pitch. The pitch of the tone corresponds to its fundamental frequency, of which the harmonics are multiples.

Complex acoustical events such as uttered speech or bird song are not easily decomposed into tones, but may be represented as spectrograms (or sonograms). Consider first a tune, i.e. a sequence of tones over time. In the spectrogram, the signal will be decomposed into temporal sections (with some overlap defined by a window function). Next, the Fourier transform within each temporal window is computed and the result is plotted as a column of gray-values in a coordinate system spanned by time (the center time of each window) and frequency. As a result, a line will appear in the sonogram going up for high pitches and down as pitch is lowered. In speech or bird song, many frequencies will be present simultaneously, resulting in more complicated sonograms.

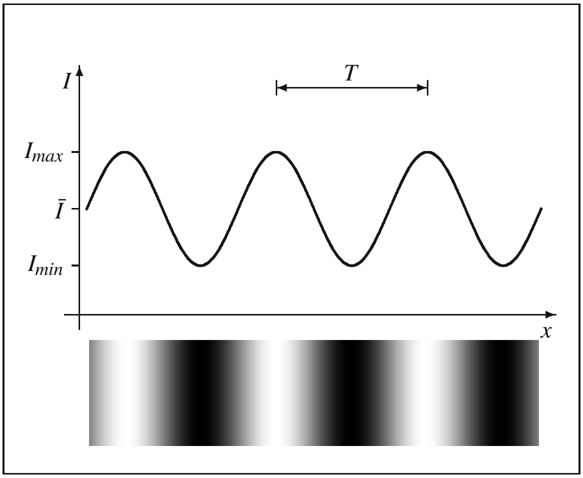


Fig. 3.3 Sinusoidal grating of the form $I(x,y) = \bar{I} + 0.5\Delta I \, \sin(2\pi x/T)$, where $\bar{I} = 0.5(I_{max} + I_{min})$. Such gratings with various period lengths T and contrasts are used for measuring of the acuity of vision.

3.1.3 Vision

The representation of signals by sinusoidals and frequencies is rather intuitive in the cases of colors and tones. In images, i.e. two-dimensional distributions of intensity values, frequency representations seem to be much less intuitive at first glance. However, properties such as resolution, acuity, or granularity of an image are closely related to a frequency variable. Indeed, the well-known JPEG-encoding of images rests on the discrete cosine transform (DCT), a close relative of the Fourier transform. In this representation, homogeneous image regions with little intensity variation are represented by a coarse grating and thus need less storage space than image regions with fine-grain contrasts.

Figure 3.3 shows a sinusoidal intensity variation over an image coordinate x. Gratings are characterized by a wavelength, or alternatively by a (spatial) frequency. Since image intensities cannot be negative, gratings will always be offset from zero on the intensity axis. Instead of characterizing mean and amplitude, the strength of modulation is often defined as the so-called Michelson[2] contrast,

$$\text{contrast} := \frac{I_{max} - I_{min}}{I_{max} + I_{min}}. \tag{3.1}$$

In the two-dimensional case, the sinusoidals become plane waves, i.e. functions with a sinusoidal modulation in one direction and zero variation in the orthogonal direction (cf. Figure 2.7a). Images may then be decomposed into two-dimensional gratings of various frequencies. Table 3.1 shows icons of two-dimensional gratings with various combinations of spatial frequencies. Superposition of gratings amounts to a pixelwise summation of intensities. Fourier theory posits that any image can be generated by superimposing gratings with variable amplitude and phase (i.e., positional shift). Since the set of gratings is a two-dimensional manifold, the spectrum becomes a two-dimensional function specifying the amplitude for each component grating.

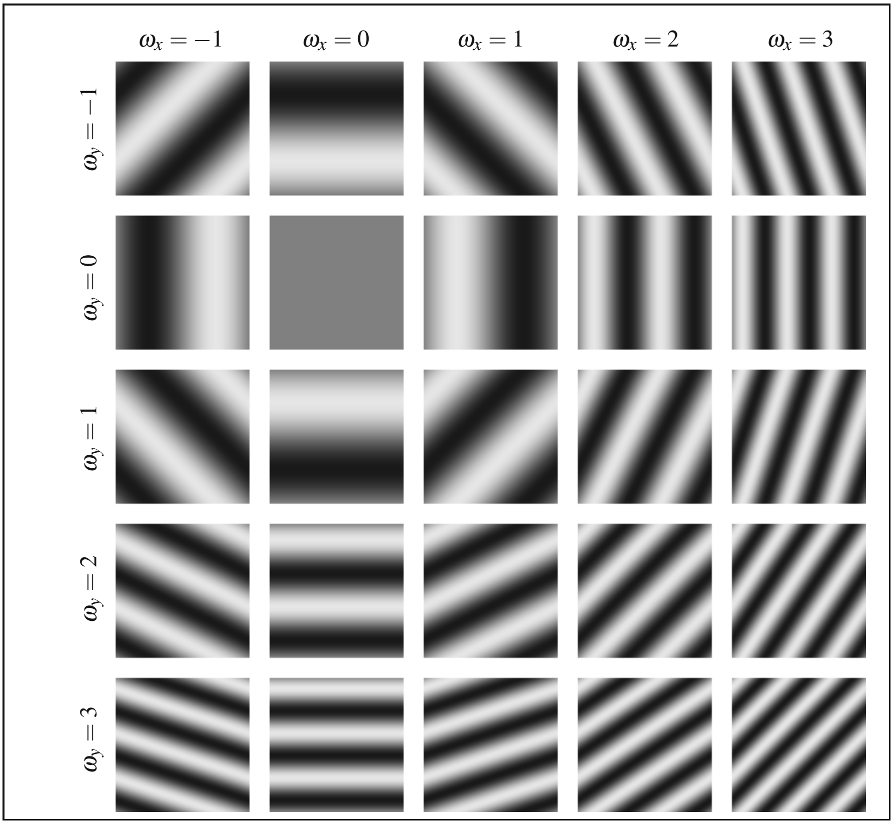
3.1.4 Magnetic Resonance Tomography

In nuclear magnetic resonance imaging, a slice image of a volume such as the head of a patient is generated based on the local density of hydrogen atomic nuclei (protons). The basic underlying effect is nuclear magnetic resonance (NMR). Protons placed in a strong static magnetic field (e.g., $H_o = 1.5$ Tesla) can be "activated" by applying a second, high frequency alternating electromagnetic field. Once the protons are activated, they "resonate", i.e. they emit an electromagnetic wave with a specific frequency. The frequencies for both activation and response are proportional to the basic field strength H_o and the gyro-magnetic ratio of the atom considered.

In the imaging process, a slice is defined in the volume by adding an axial gradient field G to the basic field H_o. If z denotes the axial coordinate, the field strength of the static field takes the form $H_o + Gz$. If an activating frequency is now applied,

[2] Albert A. Michelson (1852 – 1931). American physicist. Nobel price in Physics 1907.

Table 3.1 Two-dimensional gratings of the form $I(x,y) = \sin(2\pi(\omega_x x + \omega_y y))$ for some integer values of ω_x and ω_y.



only the protons at a particular z-location will satisfy the resonance condition. These protons fill a plane perpendicular to the axial direction. After the activation, another DC gradient field is applied, say along the x-coordinate. This field component is called a read-out gradient. The resonance signals emitted by the activated protons have frequencies proportional to the total locally "perceived" field strength. Signals emitted from voxels with different x-coordinate will therefore have different frequencies. Since all signals are overlayed in the recording, the different frequency components will have to be isolated in order to find the contributions from voxels at a particular x-position. This is done with the Fourier transform. The remaining problem, then, is that all voxels with a given x-coordinate, corresponding to one Fourier component of the signal, form a line in the slice, extending along the y-coordinate. To localize signals also in the y-direction, measurements with other read-out gradients must be performed and combined. Also in this step, the Fourier transform can be useful.

3.2 Why Are Sinusoidals Special?

The examples given above make it clear that sinusoidal functions are used in many contexts, but the mathematical reason for this needs further explanation. In this section, we will present an important relation between linear, shift-invariant systems and sinusoidals. A system is considered a "mapping", assigning an output function to each input function. Extending the concept of a "functional" from Equation 2.38 we write

$$E = \Phi(I) \tag{3.2}$$

where I and E are functions such as an input image and an activity distribution on a layer of neurons; the arguments of these functions are omitted. Mappings such as Φ, which assign functions to functions, are also called *operators*. An important characteristic of the operator Φ is its set of eigenfunctions,[3] i.e. functions satisfying the equation

$$\Phi(f) = \lambda f \tag{3.3}$$

where λ is a number called the eigenvalue associated with the eigenfunction f.

Before we give a formal account of the problem, we note that the eigenfunctions of convolution (i.e., linear shift-invariant) operators are likely to be periodic functions, as is illustrated in Figure 3.4.

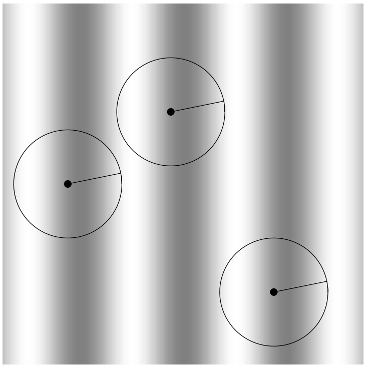


Fig. 3.4 Convolution of a periodic function. The periodic function is shown in the background. At three locations, the local neighborhood for the convolution is symbolized by a circle. Clearly, the images within the circles are identical. Therefore, the result of convolution at all three positions must also be identical. It follows that the convolution of the periodic function with an arbitrary kernel yields again a periodic function. This results is an immediate consequence of the translation-invariance of convolution.

3.2.1 *The Eigenfunctions of Convolution: Real Notation*

Let $f(x)$ be a convolution kernel. We write the convolution operation applied to a sine function with frequency ω as $(f * sin_\omega)$ and note that this is a function (the output function of the system) which can be evaluated at variable values x. We may then write the convolution with the sine function $\sin(\omega x)$ as

[3] The term "eigenfunction" is based on the German word "eigen" which means "own". It expresses the fact that eigenfunctions and eigenvalues characterize the operator.

$$(f * \sin_\omega)(x) = \int_{-\infty}^{\infty} f(x') \sin(\omega(x - x'))\, dx' \tag{3.4}$$

where x' is an auxiliary variable which cancels out as the integral is computed (cf. Equation 2.16). Although in Equation 3.4 we use the spatial variable x, the argument works just as well in the temporal domain if causality is taken care of by setting $f(t') = 0$ for $t < 0$.

It is important for the following argument, that the difference $x - x'$ appears in the sine-term, even though the integral remains the same if we exchange the roles of f and the sine (commutativity of convolution).

Applying the addition theorem $\sin(\alpha - \beta) = \sin\alpha\, \cos\beta - \cos\alpha\, \sin\beta$, we obtain:

$$(f * \sin_\omega)(x) = \int f(x')(\sin\omega x\, \cos\omega x' - \cos\omega x\, \sin\omega x')\, dx'. \tag{3.5}$$

We may now split up the integral in two (distributive law) and move the terms not depending on x' out of the integrals, which are taken over x':

$$\begin{aligned}(f * \sin_\omega)(x) &= \sin\omega x \int f(x')\, \cos\omega x'\, dx' - \cos\omega x \int f(x')\, \sin\omega x'\, dx' \\ &= \tilde{f}_c \sin\omega x - \tilde{f}_s \cos\omega x. \qquad (3.6)\end{aligned}$$

After moving the variable x out of the integrals, the remaining integrals evaluate to constants, for which we introduced the names

$$\tilde{f}_c := \int f(x) \cos\omega x\, dx \tag{3.7}$$

$$\tilde{f}_s := \int f(x) \sin\omega x\, dx. \tag{3.8}$$

Thus, we have shown that the convolution of a sine with frequency ω with any mask is a weighted sum of a sine and a cosine with the same frequency ω. A linear, translation-invariant system cannot change the frequency of a signal.

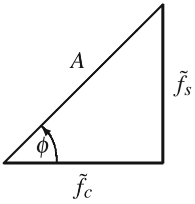
Equation 3.6 is not yet the full answer to the eigenfunction problem, since the response to a sine is a weighted sum of a sine and a cosine. We can get one step closer if we observe that such sums of a sine and a cosine can be written as a sine with a phase shift. This is in fact a special case of the addition theorem stated above. We introduce the new variables A and ϕ, called amplitude and phase:

$$A := \sqrt{\tilde{f}_c^2 + \tilde{f}_s^2}, \quad \phi := \tan^{-1} \frac{\tilde{f}_s}{\tilde{f}_c}. \tag{3.9}$$

A geometrical interpretation of the quantities $\tilde{f}_c$, $\tilde{f}_s$, A, and ϕ is given in Figure 3.5. With these variables, we obtain

$$\begin{aligned}(f * \sin_\omega)(x) &= \sqrt{\tilde{f}_c^2 + \tilde{f}_s^2}\, (\cos\phi\, \sin\omega x - \sin\phi \cos\omega x) \qquad (3.10) \\ &= A \sin(\omega x - \phi). \qquad (3.11)\end{aligned}$$

Fig. 3.5 Geometrical interpretation of the relation of the quantities $\tilde{f}_s$, $\tilde{f}_c$ and A, ϕ



Equation 3.11 is as close as we can get to the sought eigenfunction equation: when passed through a linear translation-invariant system, sinusoidals are attenuated ($|A| < 1$) or amplified ($|A| > 1$, only for energy consuming systems) and shifted in phase. Both amplitude modulation and phase shift depend on the frequency of the sinusoidal which cannot be changed by the linear translation-invariant system. Both effects can be formally combined in one factor if complex number theory is used. We will explain the complex notation in the next section.

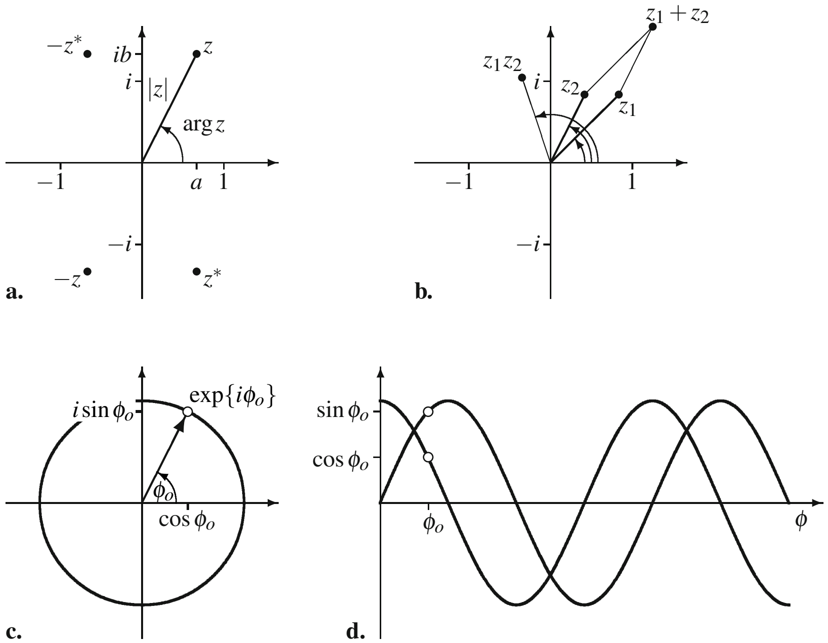


Fig. 3.6 Complex numbers. **a.** The complex plane is spanned by the real and imaginary axes with units 1 and $i = \sqrt{-1}$. A complex number can be expressed through its real and imaginary parts (a,b), or by its "modulus" $|z|$ and "argument", or phase, $\arg z$. **b.** Two complex numbers z_1, z_2 can be summed by adding their respective real and complex parts. Multiplication can be thought of as a rotation and scaling, i.e. it adds up the arguments and multiplies the moduli. **c.** The complex exponential with the purely imaginary argument $i\phi$ describes a unit circle in the complex plane. **d.** The real and the imaginary parts of the complex exponential are sinusoidal functions of ϕ (Euler's formula).

3.2.2 Complex Numbers

Complex numbers arise from solutions of algebraic equations. For example, the equation $x^2 - 1 = 0$, has the solutions $x = \pm 1$. In the set of real numbers, however, the equation $x^2 + 1 = 0$ has no solution. Formally, one may say that the "number" $\sqrt{-1}$ solves the above equation, even though $\sqrt{-1}$ is clearly not a real number. We introduce the notation

$$i = \sqrt{-1} \tag{3.12}$$

and call i the imaginary unit. For any pair of real numbers, (a,b), we call

$$z := a + ib \tag{3.13}$$

a complex number with real part a and imaginary part b (cf Figure 3.6a). It can be shown that every algebraic equation (i.e. equation of the form $\sum_{k=0}^{n} a_k x^k = 0$) has exactly n solutions in the set of complex numbers, where multiple solutions are counted accordingly. In the set of real numbers, it may have anything between 0 and n solutions.

We consider a few basic properties of complex numbers (see Figure 3.6). For a complex number $z = a + ib$, the real number

$$|z| := \sqrt{a^2 + b^2} = \sqrt{(a+ib)(a-ib)} \tag{3.14}$$

is called the absolute value or modulus of z. For a complex number $z = a + ib$, the number $z^* := a - ib$ is called its *complex conjugate*. With this notation, we may write Equation 3.14 as $|z|^2 = zz^*$.

The counterclockwise angle between the real axis and the "vector" z is called the argument, of phase of z:

$$\arg z := \begin{cases} \tan^{-1} b/a & \text{for} \quad a > 0 \\ \pi + \tan^{-1} b/a & \text{for} \quad a < 0, b \geq 0 \\ -\pi + \tan^{-1} b/a & \text{for} \quad a < 0, b < 0 \end{cases} . \tag{3.15}$$

For $a = b = 0$, the argument function is undefined. Strictly speaking, Equation 3.15 is only the so-called principle value of the arg-function, to which integer multiples of 2π may freely be added to get the true arg function. We will not further pursue this issue here.

The complex numbers form a plane which is spanned by the real axis with unit 1 and the complex axis with unit i. Calculations on the set of complex numbers are defined in a straight forward way (Figure 3.6b). For example, addition of two complex numbers is accomplished by adding the real and imaginary parts. Here, complex numbers behave like two-dimensional vectors. They differ from 2D vectors by their multiplication rule,

$$(a_1 + ib_1)(a_2 + ib_2) := a_1a_2 + i(a_2b_1) + i(a_1b_2) + i^2(b_1b_2) \tag{3.16}$$

$$= \underbrace{a_1a_2 - b_1b_2}_{\text{real part}} + i \underbrace{a_2b_1 + a_1b_2}_{\text{imaginary part}} . \tag{3.17}$$

With some computations, it can be shown that $\arg(z_1z_2) = \arg z_1 + \arg z_2$ and $|z_1z_2| = |z_1||z_2|$ (cf Figure 3.6b). This is to say, multiplication with a number z amounts to a rotation by $\arg z$ and a scaling by $|z|$.

This latter result, i.e. expressing multiplication as an addition of the arg-values and a multiplication of the moduli, is reminiscent of a property of the exponential function, i.e. $e^xe^y = e^{x+y}$. Indeed, a complex exponential function can be defined based on the famous Euler[4] formula:

$$e^{i\varphi} = \cos\varphi + i\sin\varphi \tag{3.18}$$

$$e^{a+ib} = e^a\cos b + ie^a\sin b \tag{3.19}$$

Two illustrations of Euler's formula are given in Figure 3.6c, d. Figure 3.6c is called the polar (or "phasor") representation of a complex number, i.e. its representation by modulus and phase:

$$z = |z|\exp\{i\arg z\} \tag{3.20}$$

Figure 3.6d illustrates the relation of the complex exponentials to sinusoidals. As the phasor rotates about the origin in the complex plane (i.e. as the phase ϕ increases), the real and imaginary parts of the complex number describe a cosine and a sine wave.

As a final remark on complex numbers, we note two inverted versions of Euler's formula that allow to go back from the complex to the real notation:

$$\cos\varphi = \frac{1}{2}(e^{i\varphi} + e^{-i\varphi}) \tag{3.21}$$

$$\sin\varphi = \frac{1}{2i}(e^{i\varphi} - e^{-i\varphi}). \tag{3.22}$$

3.2.3 The Eigenfunctions of Convolution: Complex Notation

With the complex notation, we gain two things: first, we can calculate convolutions with complex exponentials without applying the addition formulae for trigonometric functions, thus simplifying the calculations considerably. Second, we can combine the amplification and phase shift into one complex number, thus obtaining a true eigenfunction. For the calculation, we insert the complex exponential function $\exp\{i\omega x\}$ into the convolution equation:

[4] Leonhard Euler (1707 – 1783). Swiss mathematician.

$$(f * \exp_\omega)(x) = \int f(x')\exp\{i\omega(x-x')\}dx' \tag{3.23}$$

$$= \exp\{i\omega x\}\underbrace{\int f(x')\exp\{-i\omega x'\}dx'}_{\tilde{f}(\omega)}. \tag{3.24}$$

The eigenvalue associated with the eigenfunction $\exp\{i\omega x\}$ is, then, $\tilde{f}(\omega)$. The values $\tilde{f}_c$ and $\tilde{f}_s$ from Section 3.2.1 are the real and imaginary parts of $\tilde{f}$,

$$\tilde{f} = \tilde{f}_c + i\tilde{f}_s = Ae^{i\phi}, \tag{3.25}$$

where $A = |\tilde{f}|$.

The function $\tilde{f}(\omega)$ describes for each frequency ω the action that the system exerts on an input signal $\exp i\omega x$. In the output, the frequency itself is not changed, but the amplitude of each frequency component is multiplied by $|\tilde{f}(\omega)|$ and its phase is shifted by $\arg\tilde{f}(\omega)$. The function $\tilde{f}(\omega)$ is therefore called the *modulation transfer function* (MTF) of the convolution system. As we shall see later, its relation to the point-spread function f in Equation 3.24 is already the definition of the Fourier transform.

3.2.4 Gaussian Convolution Kernels

As an example, consider the convolution with a Gaussian. In the two-dimensional case, this example describes the projection of a slide with a de-focused slide projector. The point-spread function for this system is the image of a small light spot, i.e. a blurring "disk". It is mathematically described as a Gaussian where the width σ describes the amount of blur,

$$f(x) = \exp\{-\frac{x^2}{2\sigma^2}\}. \tag{3.26}$$

The modulation transfer function $\tilde{f}$ for this system is:

$$\tilde{f}(\omega) = \int \exp\{-\frac{x^2}{2\sigma^2}\}\exp\{-i\omega x\}\,dx. \tag{3.27}$$

Next, we collect all exponents in one exponential function and add the term $-\sigma^4\omega^2 + \sigma^4\omega^2 = 0$ in the exponent. We can then move a term not depending on x outside the integral and are left inside with a completed square expression to which the binomial rule can be applied:

$$\tilde{f}(\omega) = \int \exp\{-\frac{1}{2\sigma^2}(x^2 - 2i\sigma^2\omega x - \sigma^4\omega^2 + \sigma^4\omega^2)\}\,dx \tag{3.28}$$

$$= \exp\{-\frac{1}{2}\sigma^2\omega^2\}\int \exp\{-\frac{1}{2\sigma^2}(x - i\sigma^2\omega)^2\}\,dx. \tag{3.29}$$

Next, we substitute $y = x - i\sigma^2\omega$ in the integral and note that $\int \exp\{-y^2\}dy = \sqrt{\pi}$ even for complex arguments[5]. We thus obtain

$$\tilde{f}(\omega) = \exp\{-\frac{1}{2}\sigma^2\omega^2\} \int \exp\{-\frac{y^2}{2\sigma^2}\}\, dy \tag{3.30}$$

$$= \sqrt{2\pi}\sigma \exp\{-\frac{1}{2}\sigma^2\omega^2\}. \tag{3.31}$$

This is a Gaussian with width $1/\sigma$. This result is also known as "uncertainty relation": the product of the width values of the point spread function and the MTF, σ and $1/\sigma$, is a constant.

We now use Euler's formula in the form of Equation 3.21 to write the result in real notation. Note that this is the standard way to interpret complex number results:

$$(f * \cos_\omega)(x) = \frac{1}{2}\left((f * \exp_\omega)(x) + (f * \exp_{-\omega})(x)\right) \tag{3.32}$$

$$= \frac{1}{2}(\tilde{f}(\omega)e^{i\omega x} + \tilde{f}(-\omega)e^{-i\omega x}) \tag{3.33}$$

$$= \exp\{-\frac{1}{2}\sigma^2\omega^2\}\cos\omega x. \tag{3.34}$$

The latter equality is due to the fact that the MTF is real and symmetric. Thus, convolution with a Gaussian does not change the phase of the signal. This result is true for all real, symmetric MTFs. For the sine-function, we obtain analogously

$$(f * \sin_\omega)(x) = \exp\{-\frac{1}{2}\sigma^2\omega^2\}\sin\omega x. \tag{3.35}$$

Equations 3.31 to 3.35 show three things. First, the value of $\tilde{f}(\omega)$ for a Gaussian kernel is real and symmetric. Therefore, convolving with a Gaussian does not involve a phase shift. In our blurred projection example, this means that the blur leads to a symmetric smear of the point input, but not to a displacement. Second, the amplification factor decreases as the frequency ω of the filtered pattern increases. That is to say, the system will leave low spatial frequencies (coarse pattern) largely unchanged, but will remove high spatial frequencies (fine detail). This behavior is called *low-pass*. Of course, this behavior is very intuitive for the blurred projection example given above. If spatial gratings of the type shown in Table 3.1 are used, blur will leave the coarse pattern unchanged, but wash out the fine pattern. Third, the width of the MTF, i.e. the "speed" with which $\tilde{f}(\omega)$ decreases for higher frequencies, depends on the width of the filter, σ. The wider the filter mask, the faster does the amplification factor decrease, i.e. the stronger or more pronounced is the low-pass property.

[5] This result rests on the path independence of complex line integrals.

3.3 Fourier Decomposition: Basic Theory

Sinewave gratings are rare patterns in the real world. The relevance of the above theory therefore rests strongly on the fact that most functions (and virtually all naturally occurring signals) can be expressed as a linear superposition of sine and cosine waves with characteristic weight functions. We will not prove this result rigorously here but rather give an extended motivation. For a deeper mathematical treatment, see the texts cited at the end of the chapter.

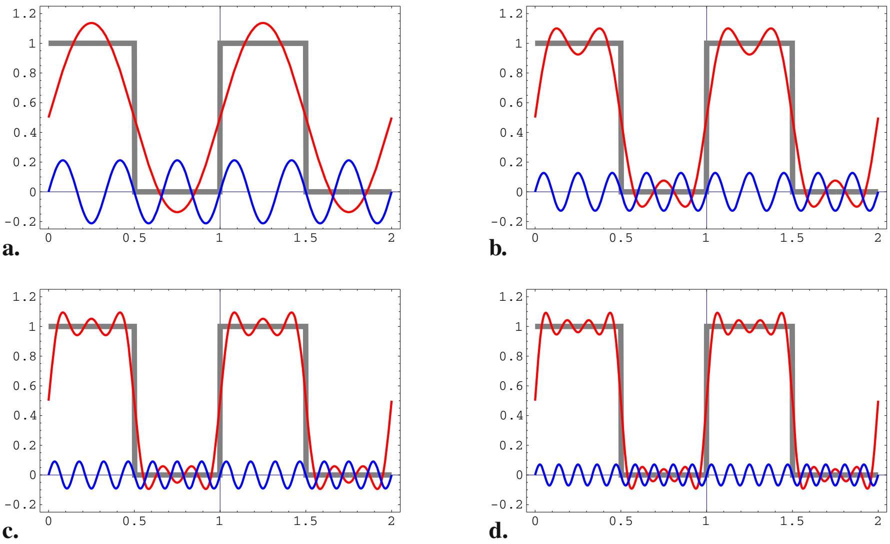


Fig. 3.7 Approximation of the "box function" (Equation 3.36 with $\omega = 1$) by a Fourier-sine-series (Equation 3.39). The box function is shown as a heavy gray line. In each panel, the two thin lines show the current approximation $g_k(x)$ and the next correction term $a_{k+2}\sin(k+2)\nu x$ for $k = 1,3,5,7$. For further explanations see text.

Table 3.2 Wavelength, frequency, and wave number of typical functions

	$\sin x$	$\sin \omega x$	$\sin 2\pi x$	$\sin 2\pi\omega x$
wavelength T	2π	$2\pi/\omega$	1	$1/\omega$
frequency $\omega = 1/T$	$1/(2\pi)$	$\omega/(2\pi)$	1	ω
wave-number $k = 2\pi/T$	1	ω	2π	$2\pi\omega$

3.3.1 Periodic Functions

We consider first functions with the property $f(x+T) = f(x)$ for some T. An example is the sine wave with $T = 2\pi$. Such functions are called periodic with wavelength (or period) T. The inverse of the wavelength T is called the frequency, $\omega = 1/T$. Sometimes, we will also use the "wave number" $k = 2\pi/T$ to characterize a periodic function. The relations are summarized in Table 3.2.

As an example, consider the "box-function",

$$b(x) := \begin{cases} 1 \text{ if } \mathrm{mod}\,(x,T) < T/2 \\ 0 \text{ else} \end{cases} . \tag{3.36}$$

Figure 3.7 shows this box function (heavy gray line) together with two sinusoidals. The first one has the same frequency as the box function itself; we call this frequency the fundamental frequency $\omega_f = 1/T$. In the sequel, we will use the "wave number" $\nu = 2\pi/T$ for the sake of simplicity. The sinusoidal thus has the form

$$g_1(x) = 1/2 + b_1 \sin(\nu x) \tag{3.37}$$

where the amplitude $b_1$is chosen such that the quadratic difference between the box function and the approximation is minimized. The factor $a_o = 1/2$ is called the DC, or direct current part; it is the average of the box function, $a_o = 1/T \int_o^T b(x)dx$. If we consider the difference between $b(x)$ and $g_1(x)$, we note that it changes sign three times as often as the fitting sine function. We can therefore arrive at a better approximation of the box function by adding a second sine with frequency 3ν; it appears in the lower part of figure. (The fact that the frequency 2ν does not contribute to better approximation is an accidental property of the box function.) Figure 3.7b shows the least square approximation of the box function by the function $g_3(x) = g_1(x) + b_3 \sin(3\nu x)$. The approximation is better, but again deviates from the desired shape by an oscillatory function, this time with frequency 5ν. We can repeat this procedure of stepwise improvements and eventually obtain

$$g_n(x) = a_o + b_1 \sin \nu x + b_3 \sin 3\nu x + \ldots + b_n \sin n\nu x \tag{3.38}$$

$$= a_o + \sum_{k=1}^{n} b_k \sin k\nu x. \tag{3.39}$$

The coefficients b_k can be shown to take the values

$$b_k = \begin{cases} \frac{2}{k\pi} & \text{for} \quad i \in \{1,3,5,\ldots\} \\ 0 & \text{for} \quad i \in \{2,4,6,\ldots\} \end{cases} ; \tag{3.40}$$

a procedure for determining the coefficients for arbitrary functions will be presented below.

We call series of the type shown in Equation 3.39 (including also cosine values) "trigonometric polynomials" or *Fourier series* (see also Equation 3.46 below). Figure 3.7 shows the first steps of that series, i.e. the functions $g_1(x)$

(Figure 3.7a) through $g_7(x)$ (Figure 3.7d). For each x satisfying $\bmod(x,T) \neq 0.5$ and $\bmod(x,T) \neq 1$, i.e. for each x where $b(x)$ is continuous, the series converges[6] towards the true functional value:

$$\lim_{n\to\infty} g_n(x) = b(x). \tag{3.41}$$

Figure 3.8 shows a more general case where the sinusoidals needed to reconstruct the signal have different phases. This can be seen by checking the value at $x = 1$ of the correction term (lower sinusoid in each panel); while this is zero for all frequencies in Figure 3.7, the value now changes from panel to panel. In the equation, the phase shift is accounted for by adding a sine and a cosine term for each frequency, each with its own coefficient a_k and b_k, respectively:

$$g_n(x) = a_o \;+\; \sum_{k=1}^{n} a_k \cos k\nu x \;+\; \sum_{k=1}^{n} b_k \sin k\nu x. \tag{3.42}$$

3.3.2 The Convolution Theorem; Low-Pass and High-Pass

In Section 3.2.3, we have shown that the convolution of a sine or cosine function with some kernel amounts to multiplying the sine or cosine with an according amplification factor and introducing a phase shift. This result generalizes nicely to Fourier series. Since convolution is a linear operation, we may convolve a Fourier series with a given kernel function by convolving each sine and cosine in the series individually with that same kernel and adding up the results afterwards. Since the frequencies of the sines and cosines do not change, this amounts to a multiplication of the according coefficients with some frequency-dependent factor.

As an example, we consider the convolution of the box function (Equation 3.36) with a Gaussian (Equation 3.26). Recall that we already discussed this convolution as modeling a de-focused slide projector taking a stripe pattern as its input and producing a blurred image of the stripes. We write the box function as its Fourier series

$$b(x) = \frac{1}{2} + \sum_{k=1}^{\infty} b_k \sin k\nu x \tag{3.43}$$

Using the results from Equation 3.34 and 3.35 together with the linearity of convolution, we obtain

$$(b * f)(x) = \frac{1}{2}(\cos_0 * f)(x) + \sum b_k (\sin_{k\nu} * f)(x) \tag{3.44}$$

$$= \frac{1}{2} + \sum_k b_k \exp\{-\frac{1}{2}\sigma^2 (k\nu)^2\} \sin k\nu x. \tag{3.45}$$

[6] For the box function, convergence is pointwise, but not uniform. In the vicinity of the discontinuities of the box-function, the oscillations are not damped, but move closer and closer to the edge while the overshot remains constant. This effect is known as Gibbs phenomenon. It occurs only for discontinuous functions, which are of little relevance for our purposes.

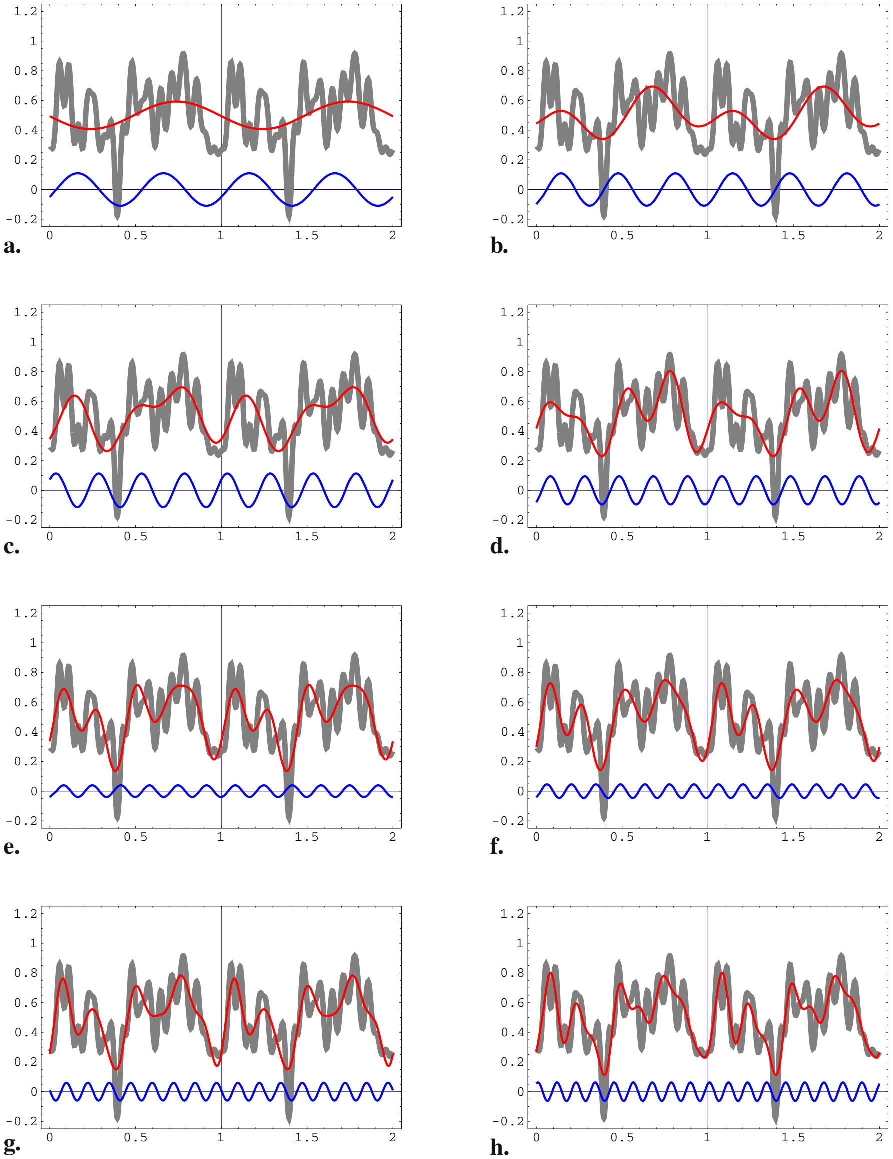


Fig. 3.8 Approximation of an arbitrary periodic function by a Fourier-series. In each panel, repetitions of the function are shown as heavy gray line. The thin lines show the current approximation $g_k(x)$ and the next correction term $a_{k+1}\cos(k+1)\nu x + b_{k+1}\sin(k+1)\nu x$ for $k = 1\dots 8$. For further explanations see text.

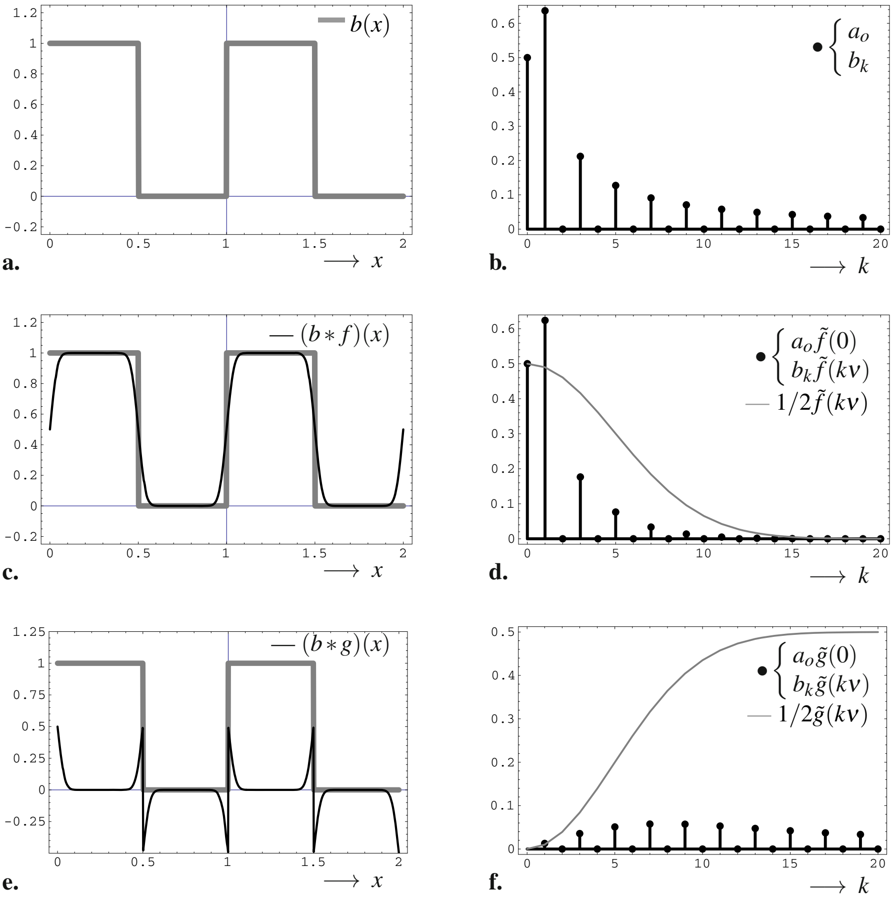


Fig. 3.9 Illustration of the convolution theorem. **a.** Box function (Equation 3.36 with $\omega = 1$) **b.** Coefficients of the according Fourier-series (Equation 3.39). **c.** Low-pass filtered version of the box-function, using a Gaussian filter kernel$(b * f)$. **d.** Fourier-coefficients for the filtered signal. The amplification factor $\tilde{f}(\omega)$ (Fourier transform of the Gaussian kernel) is shown in the background (not to scale). The coefficients are obtained by multiplying the original coefficients (Figure **b.**) with the Gaussian. **e** High-pass filtered version of the box function obtained by convolution with $g := \delta - f$. **f** Fourier coefficients of the high-pass filtered signal. The Fourier-transform of the filter kernel ($\tilde{g} = 1 - \tilde{f}$) is shown in gray (not to scale).

This is a new Fourier series with the unchanged DC component $1/2$ and the new coefficients $b_k^* = b_k \tilde{f}(k\nu)$. This equation also illustrates why $\tilde{f}(\omega)$ is called a modulation transfer function: For each "modulation (i.e. sinusiodal), it specifies a factor with which this sinusoidal is transferred through the linear translation-invariant system. Figure 3.9d illustrates this result for the box function with $T = 1$ and a Gaussian blur with $\sigma = 0.2$.

Figure 3.9 shows that by convolving the box function with a Gaussian, its high frequency components are strongly attenuated while the low frequency components are passed largely unchanged. This behavior is known as low-pass. It amounts to a smoothing, or blur of the input function. An ideal low-pass is a step function (Heaviside function) in the frequency domain, which however is hard to realize as a convolution in the space- or time- domains.

If the smoothed box function is subtracted from the original one, the high spatial frequency components that were removed in smoothing, stand out in the difference (Figure 3.9e). Since the box function itself can be thought of as the convolution of the dot function with the Dirac δ impulse (Equation 2.11), we can formally write this difference as a convolution with a new kernel g defined as $g := \delta - f$. It will pass the high frequencies and remove the low ones, and is therefore called a high-pass. In the frequency domain, this particular high-pass is characterized by the coefficients $\tilde{g}(k\nu) = 1 - \tilde{f}(k\nu)$ (Figure 3.9f).

This example is an instant of the convolution theorem, the central result of this chapter. It states that the convolution of a signal with a point spread function can be replaced in the Fourier domain by the multiplication of the Fourier transform of the signal and the modulation transfer function, and that this MTF is the Fourier transform of the point-spread function (see also Figure 3.11) below.

3.3.3 Finding the Coefficients

So far, we have seen that trigonometric polynomials can approximate continuous functions. We write the general form of such polynomials as

$$p_n(x) := \sum_{k=o}^{n} a_k \cos k\nu x + \sum_{k=1}^{n} b_k \sin k\nu x. \tag{3.46}$$

Note that the DC-component has been incorporated into the first sum, keeping in mind that $\cos(0\nu x) = 1$ for all ν and x. In Equation 3.46 ν is again the fundamental frequency of the signal, i.e. p_n repeats itself with a wave-length of $T = 2\pi/\nu$. Note that this is true even if $a_1 = b_1 = 0$, i.e. if the fundamental frequency itself is missing in the signal. An example of such a "missing fundamental" stimulus is obtained from the box function by subtracting its first (fundamental) Fourier component.

How can we find the coefficients a_k, b_k? If we assume that every continuous periodic function can in fact be written as a trigonometric polynomial,[7] we can find the coefficients by exploiting the so-called orthogonality relations of sinusoids which hold for all $k, l > 0$:

[7] The prove of this assumption is the mathematically hard part of Fourier theory. We do not prove it here.

$$\int_0^{2\pi} \sin kx \, \sin lx \, dx = \begin{cases} \pi \text{ if } k = l \\ 0 \text{ if } k \neq l \end{cases} \tag{3.47}$$

$$\int_0^{2\pi} \cos kx \, \cos lx \, dx = \begin{cases} \pi \text{ if } k = l \\ 0 \text{ if } k \neq l \end{cases} \tag{3.48}$$

$$\int_0^{2\pi} \sin kx \, \cos lx \, dx = 0. \tag{3.49}$$

Geometrical motivations for these relations can be found by plotting the involved functions for selected values of k and l[8]. With these relations, we obtain:

$$a_k = \frac{2}{T} \int_0^T p(x) \cos k\nu x dx; \;\; k \in \{0, 1, 2, \ldots\} \tag{3.50}$$

$$b_k = \frac{2}{T} \int_0^T p(x) \sin k\nu x dx; \;\; k \in \{1, 2, 3, \ldots\}. \tag{3.51}$$

These latter results are proven by substituting for $p(x)$ from Equation 3.46 and applying the distributive law until the orthogonality terms appear for each element of the sum. All of these vanish except for the one where $k = l$ which evaluates to π.

We call a_k the Fourier-sine coefficient for the frequency $k\nu$ and b_k the Fourier-cosine coefficient for the frequency $k\nu$.

In the complex notation, we proceed in the same way, using the orthogonality relation

$$\int_0^{2\pi} \exp\{ikx\} \exp\{-ilx\} dx = \begin{cases} 2\pi \text{ if } k = l \\ 0 \;\; \text{if } k \neq l \end{cases}. \tag{3.52}$$

The coefficients are obtained from

$$c_k = \frac{2}{T} \int_0^T g(x) \exp\{-ik\nu x\} dx; \;\; k \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}. \tag{3.53}$$

Note that this time, we have to consider coefficients with negative index number. The resulting series reads

$$p_n(x) := \sum_{k=-n}^{n} c_k \exp\{ik\nu x\}. \tag{3.54}$$

Clearly, the coefficients of the real and complex notations are related. The equation can be calculated from Euler's formula. They read

$$c_j = \begin{cases} \frac{1}{2}(a_j - ib_j) & \text{for } \; j > 0 \\ \frac{1}{2}(a_j + ib_j) & \text{for } \; j < 0 \\ a_o & \text{for } \; j = 0 \end{cases}. \tag{3.55}$$

[8] Mathematically, it can be shown that eigenfunctions of a linear operator must be orthogonal as long as the eigenvalues differ. This consideration is consistent with the elementary result.

3.4 Fourier Decomposition: Generalizations

We have now achieved the basic results of Fourier theory for the special case known as Fourier series. Fourier series apply to periodic functions or functions with finite domain which can be made periodic by simply repeating the finite domain over and over again. An important example is given by functions defined on the circle. Their Fourier representation is a discrete set of coefficients associated with so-called harmonic frequencies, i.e. frequencies that are integer multiples of the fundamental. Spectra of such functions, as introduced in Figure 3.1 are line spectra, with each line corresponding to a discrete frequency component.

We now discuss two extensions of the Fourier series concept.

3.4.1 Non-periodic Functions

The generalization to non-periodic functions is mathematically difficult, but intuitively quite easy, if we consider functions of increasing period length T. For a given T, for example $T = 2\pi$, we have coefficients at the multiples of the fundamental frequency $\nu = 2\pi/T = 1$,

$$\omega = k\nu = \frac{2k\pi}{T} \in \{1,2,3,4,5,\ldots\}. \tag{3.56}$$

If the period is twice as long, $T = 4\pi$, we obtain $\nu = 1/2$ and

$$\omega = k\nu = \frac{2k\pi}{T} \in \{\frac{1}{2},1,\frac{3}{2},2,\frac{5}{2},\ldots\}. \tag{3.57}$$

In the end, if the period is infinite (i.e. if the function is no more periodic at all), we get a "coefficient" for every value of ω, i.e. a function of frequency. Switching back to the complex notation, we thus obtain the Fourier transform:

$$\tilde{g}(\omega) := \int_{-\infty}^{\infty} g(x)\exp\{-i\omega x\}dx. \tag{3.58}$$

By the same token, the trigonometric series becomes:

$$g(x) := \frac{1}{2\pi}\int_{-\infty}^{\infty} \tilde{g}(\omega)\exp\{i\omega x\}d\omega. \tag{3.59}$$

Equation 3.58 is called the Fourier forward transform and Equation 3.59 the Fourier backward transform. Applying both in a sequence reconstructs the original function as long as this was continuous and its square was integrable.

Note that the equations for Fourier forward and backward transform are almost identical, up to a sign in the exponential and a normalizing factor, which can be split symmetrically between the two transform equations. Applying the forward transform twice results in a mirror image of the original function, applying it four times reproduces the original.

The Fourier transform of a periodic function does not produce the discrete coefficients of the Fourier series, but rather a sum of δ-functions at the locations of the multiples of the fundamental frequencies, weighted by the coefficients. In particular, the Fourier transform of a sine of frequency ω_o is $(\delta(\omega-\omega_o)-\delta(\omega+\omega_o))/2i$. Thus, Fourier series and Fourier transform are two different operations. In practical applications, where numerical versions of the Fourier transform are used, this difference is of minor relevance.

The function $\tilde{g}(\omega)$ in Equation 3.58 is a complex function of the real variable ω (cf. Figure 3.10). By Euler's formula (Equation 3.19) the complex number $\tilde{g}(\omega_o) = \tilde{g}_c(\omega_o) + i\tilde{g}_s(\omega_o)$ for each ω_o gives the amplitude and phase of the component with spatial frequency ω_o. If only the spatial frequencies present in a pattern are to be considered, one often uses the so-called power spectrum of g, i.e. the square of the absolute value (modulus) of $\tilde{\omega}$. A famous theorem in Fourier theory (Wiener[9]-Khinchin[10] theorem) states that the power spectrum equals the Fourier transform of the auto-correlation function introduced in Equation 2.57. In formal notation, we may write

$$\tilde{\Phi}_{gg}(\omega) = |\tilde{g}(\omega)|^2 = \tilde{g}\tilde{g}^*. \tag{3.60}$$

3.4.2 Fourier-Transforms in Two and More Dimensions

The Fourier transform generalizes to functions of two or more variables, such as images or spatio-temporal intensity distributions. The sinusoidal must in this case be replaced by a plane wave, e.g.

$$\sin(\omega_x x + \omega_y y) = \sin(\vec{\omega}\cdot\vec{x}) \tag{3.61}$$

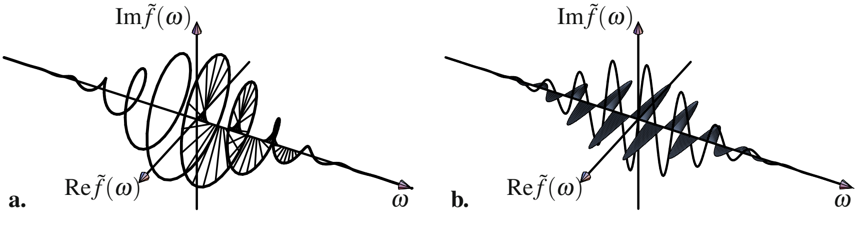


Fig. 3.10 Complex Fourier transform of a displaced Gaussian, $\exp\{-(x-x_o)^2\}$. **a.** 3D plot showing the complex functional values of each frequency ω as "vectors", or pointers in the complex plane. **b.** Real and imaginary parts of the same function shown separately. The lengths of the pointers in Figure a correspond to the power of the signal (Fourier transform of autocorrelation). The angle of the pointer in the complex plane is the Fourier phase.

[9] Norbert Wiener (1894 – 1964). United States mathematician.

[10] Aleksandr Y. Khinchin (1894 – 1959) Soviet mathematician

as are shown in Table 3.1. The argument of the sine function is an inner product (see below, Section 4.1.3) of a frequency vector $(\omega_x, \omega_y, ...)$ and a vector of the original coordinates $(x, y, ...)$. Time and temporal frequency may be treated as just another component of these vectors. In two dimensions, these plane waves look like corrugated surfaces or wash-boards whose contour lines form a set of parallel straight lines. The orientation of these contour lines is orthogonal to the vector (ω_x, ω_y), the separation of wave peaks (wave length) is $2\pi/\sqrt{\omega_x^2 + \omega_y^2}$ (cf. Figure 2.7a).

The Fourier transform then becomes a complex function of two or more real frequency variables:

$$\tilde{g}(\omega_x, \omega_y) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x,y) \exp\{i(\omega_x x + \omega_y y)\} dx dy. \tag{3.62}$$

Each point in the frequency plane (ω_x, ω_y) corresponds to one plane wave. An intuition for this frequency plane may be obtained from Table 3.1: each cell in this table represents a two-dimensional frequency vector for which the associated grating is shown.

3.5 Summary: Facts on Fourier Transforms

1. Every (sufficiently) continuous function g can be unambiguously and reversibly represented by its Fourier transform $\tilde{g}$:

$$\text{forward:} \quad \tilde{g}(\omega) := \int g(x) \exp\{-i\omega x\} dx, \tag{3.63}$$

$$\text{backward:} \quad g(x) := \frac{1}{2\pi} \int \tilde{g}(\omega) \exp\{i\omega x\} d\omega. \tag{3.64}$$

   $\tilde{g}(\omega$ is a complex number which can be decomposed into a sine and a cosine component via Euler's formula. These components are called Fourier sine and Fourier cosine components, respectively. Intuitively, Equation 3.63 therefore means that every continuous function can be represented as the sum of sine and cosine functions.

2. In the set of functions, an infinite-dimensional coordinate system can be introduced by the basis "functions" $\delta(x - y)$ for each value of y. If the function is sampled and the list of values is treated as a vector, the canonical basis (i.e., basis vectors $(0, ..., 0, 1, 0..., 0)^\top$) is an approximation of this basis. The Fourier transform can then be considered a coordinate transform with the new basis functions $\exp\{i\omega x\}$. The orthogonality constraint guarantees that this new basis is orthonormal. Since the length of a vector does not depend on the coordinate system used, the relation $\int f^2(x)dx = (1/2\pi)\int \tilde{f}(\omega)d\omega$ holds (Parseval's identity).

3. Convolution theorem: If the original function is replaced by its Fourier transform, then convolution is replaced by multiplication:

$$(g * h)^{\sim}(\omega) = \tilde{g}(\omega)\, \tilde{h}(\omega) \tag{3.65}$$

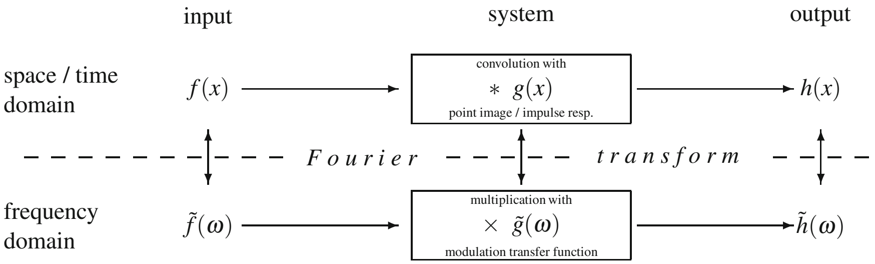


Fig. 3.11 Summary of the relation of Fourier transform and linear systems theory, i.e. the convolution theorem.

(see Figure 3.11). The commutativity and associativity of convolution, follow directly from this theorem.

4. Correlation or Wiener-Khinchin theorem: The Fourier transform of the cross-correlation function of two functions f, g,

$$\Phi_{fg}(y) := \int f(x)g(x+y)dx \tag{3.66}$$

is given by the equation

$$\tilde{\Phi}_{fg}(\omega) = \tilde{f}(\omega)\tilde{g}^*(\omega) \tag{3.67}$$

where $z^* := \mathrm{Re}\, z - i\, \mathrm{Im}\, z$ is the complex conjugate of z.

5. The modulus (complex absolute value) of the Fourier transform of a function f is known as the power spectrum of f. As a special case of the correlation theorem, we note that the power spectrum equals the Fourier transform of the auto-correlation function of f (Equation 3.60).

6. Shift theorem: Let $g(x)$ be a function with Fourier transform $\tilde{g}(\omega)$ and $s \in \mathbb{R}$ a number specifying a shift of g. The shifted version of g, $g_s(x) := g(x+s)$ has the Fourier transform

$$\tilde{g}_s(\omega) = \exp\{-i\omega s\}\tilde{g}(\omega) \tag{3.68}$$

Due to the symmetry of the Fourier transform, this implies that the Fourier transform of a Gabor function is a displaced (shifted) Gaussian.

7. Scale theorem: Let $g(x)$ be a function with Fourier transform $\tilde{g}(\omega)$ and $a \in \mathbb{R}$ a scaling factor. The scaled version of the function, $g_a(x) := g(ax)$ has the Fourier transform

$$\tilde{g}_a(\omega) = \frac{1}{a}\tilde{g}(\frac{\omega}{a}). \tag{3.69}$$

The uncertainty relation studied in Equation 3.31 is a special case of the scale theorem.

3.6 Suggested Reading

Books

Bracewell, R. N. (2003). *Fourier Analysis and Imaging*. Kluwer/Plenum, New York.

Butz, T. (2005). *Fourier transformation for pedestrians (Fourier series).* Berlin: Springer Verlag, Berlin.

James, J. F. (2011). *A student's guide to Fourier transforms. With applications in physics and engineering*. 3. edition, Cambridge University Press

Tolstov, G. P. (1962). *Fourier series*. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Original Papers

Daugman, J. (1980) Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research* 20:847 – 856

Extends spatial frequency models of the receptive field to the two-dimensional case and gives clean definitions of orientation and frequency specificity.

Hendrikson, L., Nurminen, L., Hyvärinen, L., and Vanni, S. (2008) Spatial frequency tuning in human retinotopic visual areas. *Journal of Vision* 8(10):5,1-13

Spatial frequency tuning in human visual cortex is studied based on fMRI data. The paper shows differences between different areas and gives a useful overview of earlier findings on spatial frequency tuning.

Jones, J. P. and Palmer, L. A. (1987) An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology* 58:1233 – 1258

Describes detailed fits of cortical receptive field profiles by Gabor function with suitable parameters. Additionally, spatial frequency specificities are measured with grating stimuli (see next chapter) and the results are compared to predictions derived from the Gabor fits.

Chapter 4
Artificial Neural Networks

Abstract. In models of large networks of neurons, the behavior of individual neurons is treated much simpler than in the Hodgkin-Huxley theory presented in Chapter 1: activity is usually represented by a binary variable (1 = firing; 0 = silent) and time is modeled by a discrete sequence of time steps running in synchrony for all neurons in the net. Besides activity, the most interesting state variable of such networks is synaptic strength, or weight, which determines the influence of one neuron on its neighbors in the network. Synaptic weights may change according to so-called "learning rules", that allow to find network connectivities optimized for the performance of various tasks. The networks are thus characterized by two state variables, a vector of neuron activities per time step and a matrix of neuron-to-neuron transmission weights describing the connectivity, which also depends on time. In this chapter, we will discuss the basic approach and a number of important network architectures for tasks such as pattern recognition, learning of input-output associations, or the self-organization of representations that are optimal in a certain, well-defined sense. The mathematical treatment is largely based on linear algebra (vectors and matrices) and, as in the other chapters, will be explained "on the fly".

4.1 Elements of Neural Networks

Artificial neural networks are built of three major building blocks which will be discussed in detail in the sequel. These building blocks are:

1. The activity (excitation) of a neuron together with its activation dynamics. As already discussed in the theory of receptive fields, activity is generally modeled not as a membrane potential, but as a number reflecting instantaneous spike rate. In neural network models, it is normally discretized into separate time steps.
2. The synaptic weights and learning rules governing the flow of activation in the network. This flow is also called the activation dynamics. Change of synaptic weights ("weight dynamics") as a result of previous activation pattern is studied as a model of learning.

H.A. Mallot: *Computational Neuroscience*, SSBN 2, pp. 83–112.
DOI: 10.1007/978-3-319-00861-5_4

3. The topology of the network is the pattern of connectivity between the neurons involved. It is generally described by a weight matrix, but higher level descriptions such as feed-forward vs. feed-back or layered vs. completely connected are also used. We will assume the overall topology fixed, but dynamic changes such as the creation of new neurons are also studied in the literature ("growing networks").

4.1.1 Activity and the States of a Neural Network

We start by numbering the neurons in a net with positive integers $i \in \mathbb{N}$. For each neuron i, the number e_i is the current state of excitation or activity. In "logical" neurons, e can take only the values 0 or 1. In other models, the activity may be considered a continuous variable, either in the interval $[0,1]$ or in the real numbers $\mathbb{R}$.

If we consider a network with I neurons, we can write the activity states of all neurons as an ordered list, or vector

$$\vec{e} := \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_I \end{pmatrix} = (e_1, e_2, ..., e_I)^\top . \tag{4.1}$$

If $e_i \in \mathbb{R}$, the set of all possible activity vectors forms a vector space with dimension I, $\mathbb{R}^I$. Note that I may be much larger than three, rendering geometric interpretations of this vector space difficult. In the sequel, we will occasionally give geometric interpretations for $I = 2$ or $I = 3$. For the general case, $\vec{e}$ should be thought of as an ordered list without trying to imagine I-dimensional spaces. Note also that, following standard conventions, vectors are always considered to be columns of numbers. If we need to consider rows, we use the "transposition" symbol ($\vec{e}^\top$ or $\vec{e}'$) which works in both directions, i.e., $(\vec{e}^\top)^\top = \vec{e}$.

The vector $\vec{e}$ is also called a state-vector of the neural network since it contains the information about the current activity state. In time-discrete models, an upper index t is attached to the state vector which is then written as $\vec{e}^t$.

If the neurons have spatial coordinates, as in the cases studied in previous chapters, $\vec{e}$ becomes a function e of space and time and each "neuron" i is identified with a point (x_i, y_i), leading to the interpretation

$$\vec{e}^t = (e(x_1, y_1, t), e(x_2, y_2, t), ..., e(x_I, y_I, t))^\top . \tag{4.2}$$

Equation 4.2 relates discrete neural network theory to layers of continuous activity functions as were studied in convolution systems, Equation 2.16.

4.1.2 Activation Function and Synaptic Weights

To model the dynamic development of neural activity in the network, each neuron is given an *activation function* (also called transfer function)

$$\alpha_i : \vec{e}^t \to e_i^{t+1}, \tag{4.3}$$

which takes the complete activity vector of the network at time t as its input and the activity of neuron i at time $t+1$ as its output. The activation function is usually described by *synaptic weights* w_{ij} and a static non-linearity f:

$$\alpha_i(\vec{e}) = f(\sum_{j=1}^{J} w_{ij} e_j). \tag{4.4}$$

Examples of static non-linearities which are also used in neural networks have been given in Section 2.3.2. The weighted sum forming the argument of the static non-linearity, $\sum w_{ij} e_j$, is called the "potential" of cell i and denoted by u_i.

Here, we use the convention that the weights w_{ij} are indexed with the number of the post-synaptic cell (i) and the pre-synaptic cell (j). In more detail, one might read "weight of input received by i from j." If a unit k does not receive input from unit l, the weight w_{kl} is set to zero. Thus, the set of all weights also determines the connectivity pattern or topology of the network.

Note that except for the non-linearity f, Equation 4.4 is analogous to the correlation Equation 2.9 if we discretize the input plane (x_j, y_j) and consider the image intensities $I(x_j, y_j)$ as the presynaptic inputs e_j (e.g., activities of receptor cells). The corresponding values of the receptive field function $\phi(x_j, y_j)$ become the weights w_{oj} of the sole output unit e_o. The differences between the two equations are (i) that in Equation 2.9 we have identified neurons with their spatial coordinates (x', y') which implies a dense, continuous layout and hence an integration operation over space (cf. Equation 4.2), and (ii) that the receptive field function ϕ in Equation 2.9

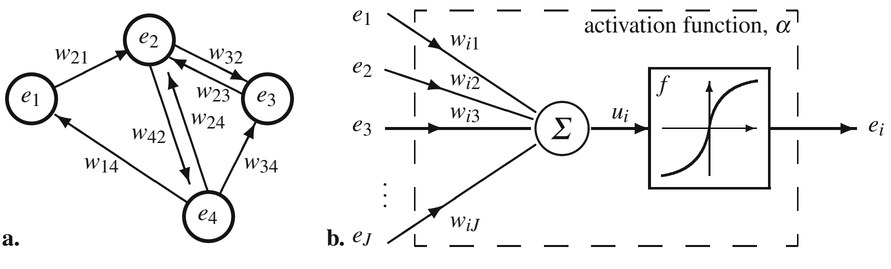


Fig. 4.1 a. A simple neural network consisting of four units ("neurons") with activities $e_1, \ldots, e_4$ and links with transmission weights w_{ij}. For explanations see text. **b.** Simple model neuron for artificial neural networks. e_j: Input activities, w_{ij} synaptic weights, Σ summation, u_i potential ($u_i = \Sigma_{i=1}^{J} w_{ij} s_j$), f static non-linearity, e Output activity ($e_i = f(u_i)$).

is not modeling just one synaptic layer, but describes the entire preprocessing from the retina to the neuron in question.

4.1.3 *The Dot Product*

For a given postsynaptic neuron, the index i in Equation 4.4 is fixed and the set of weights can be considered a vector of the same dimension as the input vector. We denote the vector of input weights of a given neuron i as $\vec{w}_i$ where i is the first ("postsynaptic") index and note that $\vec{w}_i := (w_{i1}, w_{i2}, \dots, w_{iJ})^\top$. The potential u_i is then obtained as the "dot product" of the vectors $\vec{w}_i$ and $\vec{e}$,

$$u_i = (\vec{w}_i \cdot \vec{e}) := \sum_{j=i}^{J} w_{ij} e_j. \tag{4.5}$$

The dot product is also known as the inner product, in which case it is often written as $\vec{w}_i^\top \vec{e}$, or as scalar product to indicate that the result is a scalar (i.e. a number as opposed to a vector). It has various geometric interpretations, which are helpful to understand neural network theory.

Norm of a Vector

First, the length, or "norm" of a vector can be defined as

$$||\vec{x}|| := \sqrt{\sum_i x_i^2} = \sqrt{(\vec{x} \cdot \vec{x})}, \tag{4.6}$$

which in two dimensions is simply Pythagoras' theorem. Clearly, the length of a vector is well defined in any number of dimensions. A vector with norm 1 is called a unit vector. For any vector (except $(0,0,...)$), a unit vector with the same direction can be generated by dividing the vector by its norm. This process is called a normalization.

Orthogonality and Angles

Second, two vectors with vanishing dot product, $(\vec{x} \cdot \vec{y}) = 0$, are said to be orthogonal, or uncorrelated. Orthogonality is a geometrical concept which, like length, easily generalizes to multiple dimensions. Consider two non-collinear vectors in three-dimensional space; they span a plane, i.e. a two-dimensional subspace, in which the intuitive notion of an angle applies. The same is true also for higher dimensional spaces. The dot product can therefore be used to define angles; in this sense, the dot product of two unit vectors equals the cosine of the angle included by these vectors:

$$(\vec{x} \cdot \vec{y}) = ||\vec{x}|| \; ||\vec{y}|| \; \cos \angle \vec{x}, \vec{y}. \tag{4.7}$$

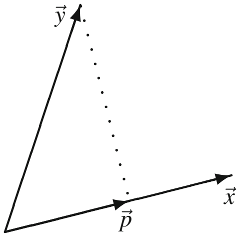


Fig. 4.2 Dot product as projection. For two vectors $\vec{x}$, $\vec{y}$, the projection of $\vec{y}$ to $\vec{x}$ is the vector $\vec{p}$. Its length is given via the dot product as $||\vec{p}|| = (\vec{x} \cdot \vec{y})/||\vec{x}|| = (\vec{x}^o \cdot \vec{y})$ where $\vec{x}^o = \vec{x}/||\vec{x}||$ is the unit vector in direction of $\vec{x}$. For the vector $\vec{p}$ itself, we thus obtain $\vec{p} = (\vec{x}^o \cdot \vec{y})\, \vec{x}^o$. To prove this relations, verify that the projection line $\vec{y} - \vec{p}$ is orthogonal to $\vec{x}$.

Thus, if the dot product of two vectors is zero, the angle included by the vectors is ±90°. The relation of the dot product to correlation will be explained later (see Equation 4.16).

Projection

Finally, the dot product is related to the notion of a projection. If an arbitrary vector $(x_1, x_2, ...x_n)$ is multiplied with a coordinate vector $(0, \dots, 0, 1, 0, \dots 0)$, where the only 1 appears at position j, the result is x_j, i.e., the component of the vector in direction of the j-th coordinate vector. In general, the dot product of two vectors can be considered as the length of the component of one vector in the direction of the other (see Fig. 4.2). Clearly, for orthogonal vectors, this length is 0.

4.1.4 Matrix Operations

If we neglect the static non-linearity in Equation 4.4 (i.e. if we assume $f(u) = u$ for all u), we obtain for the example given in Figure 4.1a the activation update rule:

$$\begin{array}{rl} e_1^{t+1} &= \quad 0e_1^t + \quad 0e_2^t + \quad 0e_3^t + w_{14}e_4^t \\ e_2^{t+1} &= w_{21}e_1^t + \quad 0e_2^t + w_{23}e_3^t + w_{24}e_4^t \\ e_3^{t+1} &= \quad 0e_1^t + w_{32}e_2^t + \quad 0e_3^t + w_{34}e_4^t \\ e_4^{t+1} &= \quad 0e_1^t + w_{42}e_2^t + \quad 0e_3^t + \quad 0e_4^t \end{array}. \tag{4.8}$$

In matrix notation, the same equation reads:

$$\begin{pmatrix} e_1^{t+1} \\ e_2^{t+1} \\ \vdots \\ e_I^{t+1} \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1I} \\ w_{21} & w_{22} & \dots & w_{2I} \\ \vdots & \vdots & & \vdots \\ w_{I1} & w_{I2} & \dots & w_{II} \end{pmatrix} \begin{pmatrix} e_1^t \\ e_2^t \\ \vdots \\ e_1^t \end{pmatrix}. \tag{4.9}$$

Note that each component of the output vector e_j^{t+1} is a dot product of the j-th row of the matrix and the input vector $\vec{e}^t$. Weights such as w_{12} which are missing in the figure are simply set to zero. The matrix of all weights is a square matrix and will be denoted by W. We may then write Equation 4.9 shorter as:

$$\vec{e}^{t+1} = W\,\vec{e}^{t}. \tag{4.10}$$

4.1.5 Weight Dynamics ("Learning Rules")

At the core of neural network theory is the introduction of synaptic learning rules, i.e. rules for the change of synaptic weights as a function of the network's state. We briefly summarize here the most important learning rules. Further explanations will be given in the course of the presentation. The detailed relation between the physiological mechanisms of synaptic plasticity and the resulting learning mechanisms is a topic of active research.

1. *Unsupervised* learning is usually modeled by variants of the Hebb[1] rule. This rule states that the synaptic weight is increased if the pre- and post-synaptic cells are active in subsequent time steps, i.e. if the pre-synaptic cell fires and the synapse is "successful" in the sense that the post-synaptic cell fires as well. The Hebb-rule models classical conditioning (see Fig. 4.3). A possible physiological mechanism of Hebbian learning is long term potentiation.
2. In *competitive learning*, the weight change is determined not only by the activities of the pre- and post-synaptic neuron, but also by the activities of other, "competing" neurons. Usually, only the most active neuron in a group (the "winner") will be allowed to learn. Competitive learning is used in models of self-organization such as Kohonen's self-organizing feature map. Physiologically, competition may be about resources needed for synaptic growth.
3. In *reinforcement learning*, the weight change is determined by the activities of the pre- and post-synaptic neuron and by a "pay-off" signal carrying information about the overall performance of the network. The pay-off signal is one global variable transmitted to all units in the network, telling them whether the last action was good or bad. It does not say what should have been done instead. As possible physiological mechanisms, neuro-modulation has been discussed.
4. In *supervised learning*, the weight change is determined by a specific teacher signal telling each neuron how it should have reacted to its last input. The most popular scheme for supervised learning is *back-propagation*. The physiological basis of a teacher signal is hard to imagine.

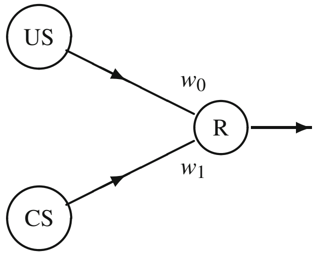


Fig. 4.3 Classical conditioning and the Hebb synapse. CS: *conditioned stimulus*, US: *unconditioned stimulus*, R: *Response*, w_0, w_1: synaptic weights. Before learning, a response R is elicited only by the unconditioned stimulus US. If CS and US occur together, w_1 will be reinforced since CS and R are both active. Eventually, CS can drive unit R by its own.

[1] Donald O. Hebb (1904 – 1985). Canadian psychologist.

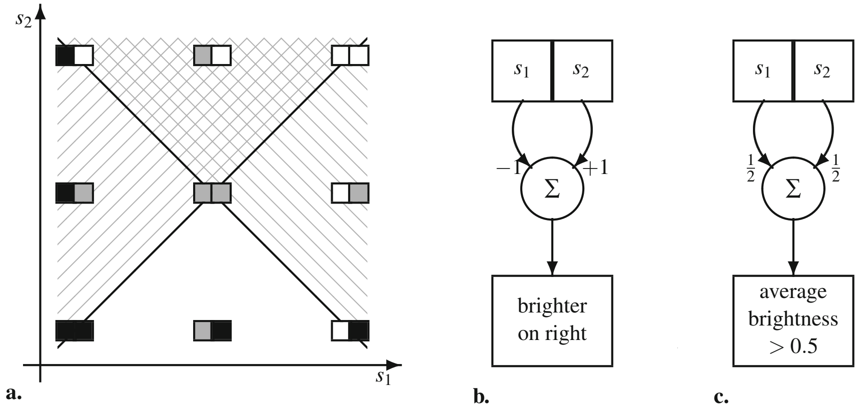


Fig. 4.4 Simple perceptrons with only two input lines. **a.** Feature space. **b.** Perceptron detecting patterns with the characteristic "brightness increasing to the right". **c.** Perceptron detecting patterns whose average intensity is above 0.5.

4.2 Classification

4.2.1 The Perceptron

As already pointed out in Section 2.1.1, the function of neurons is often seen in recognizing or detecting things, or "pattern". Since all inputs to a neuron arrive through its synapses, such pattern are eventually vectors of input activities. The idea is that neurons implement logical predicates (i.e. statements that are either true or false) about this stimulus vector. The value "TRUE" is signaled by the neuron taking the activity 1, the value "FALSE" by the activity 0. A simple example is shown in Fig. 4.4. Consider a neuron with just two inputs, let's say from light sensitive receptors. How can we build a neuron that responds if the right receptor receives more light than the left receptor? Simply enough, we can use an excitatory connection from the right receptor and an inhibitory connection from the left (Fig. 4.4b). If we apply a step non-linearity to the weighted input sum, the cell will be active if and only if the right sensor receives more light.

For the understanding of the perceptron, and indeed for pattern recognition in general, the notion of a feature space is crucial. In our example, the feature space consists of all possible pairs of light intensities (s_1, s_2) that might be delivered to the two sensors. This is equivalent to the positive quadrant of the real two-dimensional plane (Fig. 4.4a). Each possible pattern corresponds to a point in feature space and the perceptron will react with activity 1 to patterns taken from a certain subset of the feature space; other patterns will elicit no response. In the example of Figure 4.4b, all light intensity pairs where the right pixel is brighter fall in a triangular area above and left of the diagonal $s_1 = s_2$. Figure 4.4c shows a second example of a perceptron with two inputs calculating the predicate "average brightness above

0.5." All pattern satisfying this condition are right and above the line $s_1 + s_2 > 1$ also shown in Figure 4.4a.

In general, feature spaces have a separate dimension for each input line of the perceptron. If one considers the retina as the input layer, the number of dimensions becomes the number of ganglion cells which in humans is in the order of 10^6.

In terms of our neural network models, we can model a perceptron as a single unit with inputs $s_1, ..., s_N$, forming an input vector $\vec{s}$, and with weights $w_1, ..., w_N$, forming a weight vector $\vec{w}$. The weighted sum of the inputs is passed through a step non-linearity with threshold θ

$$u = \sum_{j=1}^{J} w_j s_j \; = \; (\vec{w} \cdot \vec{s}) \tag{4.11}$$

$$e = f(u) := \begin{cases} 0 \text{ if } u \leq \theta \\ 1 \text{ if } u > \theta \end{cases} . \tag{4.12}$$

The perceptrons in Fig. 4.4b, c have $\vec{w} = (-1,1)^\top$ with $\theta = 0$, and $\vec{w} = (0.5, 0.5)^\top$ with $\theta = 0.5$, respectively. Note that the weight vector $\vec{w}$ has the same dimension as the stimulus vector and can therefore be interpreted as a vector in feature space.

4.2.2 Linear Classification

Decision Boundary

We now ask the question of which subsets of a feature space can be recognized by a perceptron. This question is usually answered in terms of a "decision boundary", i.e. a line in feature space separating stimuli that the perceptron does or does not respond to. Crossing this boundary will mean that the potential u changes sign; the boundary is therefore determined by setting $u = 0$ in Equation 4.12. In the resulting equation $(\vec{w} \cdot \vec{s}) = \theta$, the weight vector $\vec{w}$ is a constant and the set of all points $\vec{s}$ satisfying the equation is a straight line orthogonal to $\vec{w}$, passing the origin with distance $\theta / \|\vec{w}\|$ (Fig. 4.5). The decision boundary of a perceptron with two inputs is therefore a straight line in feature space and cannot be curved. The perceptron is thus an example of a "linear classifier" where the term linear refers to the nature of the decision boundary, as well as to the summation part of the activation function (Equation 4.11).

Another way to think of linear classification relates to the projection property of the dot product in Equation 4.12 (cf. Fig. 4.2), which implies that any input vector $\vec{s}$ will be projected orthogonally onto the line defined by the weight vector. All points on lines orthogonal to the weight vector will therefore end up at the same location and will be put in the same category in the classification step.

In perceptrons with three input lines ($J = 3$), the feature space will have three dimensions. The weight vector still defines a straight line in that space. At a distance $\theta / \|\vec{w}\|$ along that line, the potential will take the value zero. The decision boundary then becomes a plane orthogonal to the weight vector and passing through the

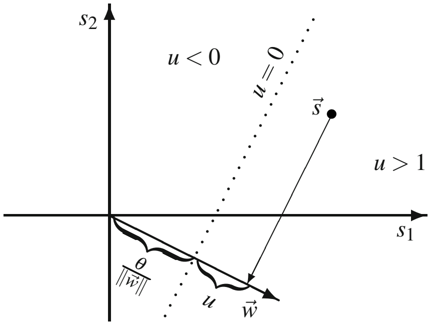


Fig. 4.5 The perceptron classifies feature vectors to one side of the linear hyperplane (dotted line) $(\vec{w} \cdot \vec{s}) = \theta$ in feature space. The distance at which this hyperplane passes the origin is $\theta/||\vec{w}||$. An input vector $\vec{s}$ is projected orthogonally onto the weight vector direction, yielding the potential u.

$u = 0$-point on the weight vector line. Mathematically, this plane is still defined by the equation $(\vec{w} \cdot \vec{s}) = \theta$ which is also known as the normal form of a plane equation. Indeed, this logic generalizes to an arbitrary number of dimensions. The decision boundary in a J-dimensional feature space will always be a linear subspace with dimension $J - 1$, cutting the feature space into two parts. Such subspaces are called “hyperplanes”.

Optimal Stimulus

On inspection of Fig. 4.4 b, it can be seen that the perceptron detecting a brightness increase to the right has a weight vector $\vec{w} = (-1, 1)^\top$ which, when interpreted as an input image, also corresponds to a rightwards increase in brightness. This observation reflects a general principle, i.e. that perceptrons are “matched filters” detecting patterns similar to their weight vector. In the same logic, we discussed center-surround receptive fields with an inhibitory center and excitatory surround as “bug detector” in Section 2.1.1 above.

Mathematically, we can ask which pattern will lead to the strongest potential in a perceptron. Since the calculation of the potential is a linear operation and therefore is doubled by doubling input intensity, this question is only meaningful if we normalize the input pattern, i.e. consider only stimuli $\vec{s}$ satisfying $||\vec{s}|| = 1$. In this case, we call

$$\vec{s}^* := \underset{\vec{s}}{\operatorname{argmax}}(\vec{w} \cdot \vec{s}) \tag{4.13}$$

the “optimal stimulus” of the perceptron. From the interpretation of the dot product as a projection, it is clear that the maximum is reached if $\vec{s}$ and $\vec{w}$ point in the same direction,

$$\vec{s}^* = \frac{\vec{w}}{||\vec{w}||}. \tag{4.14}$$

Formally, this result is proven in the so-called Cauchy-Schwarz-inequality, $(\vec{a} \cdot \vec{b}) \leq ||\vec{a}|| \, ||\vec{b}||$. The situation is completely analogous to the discussion of optimal stimuli and matched filters in Section 2.1.1. In fact, if we consider continuous layers of neurons (Equation 4.2), the resulting feature space will have an infinite number of dimensions, one for every point of the continuous plane. In this situation, the dot-product is no longer a discrete sum, but an integral, and in fact the correlation integral defined in Equation 2.9.

The idea of optimal stimuli can also be related to the statistical notion of covariance. Formally, we can calculate the covariance of $\vec{s}$ and $\vec{w}$ by considering the value pairs $(s_1,w_1),(s_2,w_2),\dots,(s_J,w_j)$. If we denote the means as

$$\bar{s} := \frac{1}{J}\sum_{j=1}^{J} s_j \text{ and } \bar{w} := \frac{1}{J}\sum_{j=1}^{J} w_j, \tag{4.15}$$

we obtain

$$\mathrm{cov}(\vec{s},\vec{w}) = \frac{1}{J}\sum_{j=1}^{J}(s_j-\bar{s})(w_j-\bar{w}) = \frac{1}{J}(\vec{s}\cdot\vec{w}) - \bar{s}\bar{w}. \tag{4.16}$$

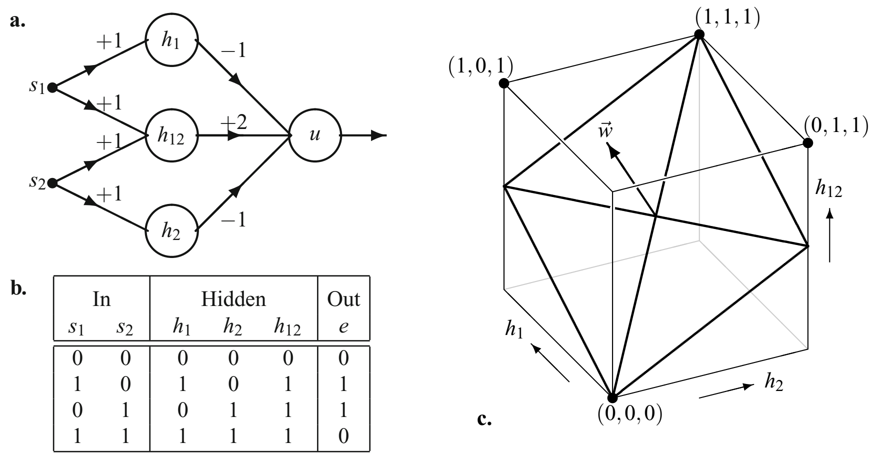
That is to say, the dot product equals the covariance up to a constant scaling factor $1/J$ and an additive constant which vanishes if the means are zero. Like the dot product, covariance is maximal if $\vec{s}$ and $\vec{w}$ are aligned.

4.2.3 Limitations

If the perceptron is a good model of neural computation, it should be able to do more interesting things than the examples given in Fig. 4.4. This is indeed the case if the dimension of feature space is large. However, in the development of neural network theory, two limitations of the perceptron have received much attention, the so-called “XOR-problem” and the locality of the perceptron.

Suppose we want to construct a two-pixel perceptron that responds with $e = 1$ if one and only one of its inputs is active. In logic, the related predicate is called “exclusive or” or XOR. If we plot this in two-dimensional feature space, the four stimulus vectors, $\vec{s} = (0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$, form a square. Clearly, there can be no line (hyperplane) such that the points $(0,1)$ and $(1,0)$ fall on one side of the plane and $(0,0)$ and $(1,1)$ fall on the other side. If classification boundaries are linear hyperplanes, this problem can only be solved by using a cascade of perceptrons (multi-layer perceptron, MLP), where the outputs of three initial perceptrons form the input of a fourth, higher level perceptron (Fig. 4.6). The output units of the initial perceptrons are called “hidden” because their activities do not explicitly show up in the classification procedure, i.e. are neither input nor output. Here, we denote the activity of the hidden units by the letter h. The response of the resulting three-layer perceptron, using the weights shown in Figure 4.6, is illustrated by the table in Fig. 4.6b.

If we allow for multiple layers, perceptrons can thus do more than just linear classification. By means of the hidden units, the classification problem is embedded in a higher dimensional space. Figure 4.6c shows the position of the resulting $\vec{h}$ vectors for the XOR-problem as corners of a unit cube in three-dimensional space. In this embedding, it is possible to separate the “true” and “false”-cases by a linear decision boundary, i.e. the plane $-h_1 + 2h_{12} - h_2 = 0$ also shown in the figure. The



In		Hidden			Out
s_1	s_2	h_1	h_2	h_{12}	e
0	0	0	0	0	0
1	0	1	0	1	1
0	1	0	1	1	1
1	1	1	1	1	0

Fig. 4.6 Three-layer perceptron implementing an exclusive or. **a.** Network topology with two input lines and three hidden units. The thresholds are set to $\theta = 0$. **b.** Truth table. **c.** 3D-feature space for the final unit, taking the hidden units as input lines. Possible input vectors are marked by black dots on the cube. In the 3D feature space, linear separation of "true" and "false"-cases with a plane is possible. $\vec{w}$ denotes the weight vector, $\vec{w} = (-1,-1,2)^\top$, which is normal to the decision plane. The intersection between the decision plane and the unit cube in $\vec{h}$-space is indicated by a bold line.

"true" cases $(0,1,0)$ and $(1,0,0)$ fall above this plane and yield the output 1; the "false" cases $(0,0,0)$ and $(1,1,1)$ lay in the decision plane and yield the output 0. When projected back into the s_1, s_2-plane, the decision boundary is no longer linear (i.e., a line), but a polygon with the desired separation properties.

A second limitation becomes apparent if a generalization of the XOR-problem is considered which is known as the parity problem: is it possible to build a perceptron that responds if and only if the number of active inputs is odd? For just two inputs, this is the XOR-problem. Again, such perceptrons are possible if three layers are used. One can show, however, that each such perceptron will need at least one hidden unit looking at all inputs simultaneously. Similarly, if the "connectedness" of complex geometrical line patterns such as intertwined spirals is considered, one hidden unit is required receiving input from the entire input layer. This is in conflict with the idea of breaking down vision into local bits and pieces, each processed by a unit with a small receptive field. The famous book by Minsky & Papert (1988) discusses this problem in detail. It has led many researchers to believe that neural networks cannot explain perception. More recently, however, it has been recognized that the parity and connectedness problems, interesting as they are from a computational point of view, may not be the central problems in visual perception and visually guided behavior. In fact, judging the connectedness of intertwined spirals, say, is not an easy task for human observers either.

4.2.4 Supervised Learning and Error Minimization

Until now, we have assumed that the weights are given, or set be the modeler, which of course raises the question how this can be achieved. Mathematically, there exists an elegant solution which, for multi-layer perceptrons, is known as back-propagation, an example of supervised learning. It can be used to check whether a MLP for a given problem exists, but is hard to interpret as a physiological mechanism.

Consider a perceptron whose task is to classify a pattern presented on its input layer. For example, the perceptron should respond with activity 1 if the pattern is a capital letter "A", and it should stay silent in all other cases. The desired performance of the perceptron may then be described by a function T mapping the input set (feature space) into the set of activity values of the perceptron. For example, $T(\vec{s}) = 1$ if $\vec{s}$ is the image of the capital letter "A"; and $T(\vec{s}) = 0$ if it is not. $T(\vec{s})$ is called the teacher signal. How can we determine the optimal weights in this case?

Error Function for Two-Layer Perceptrons

The original learning rule for perceptrons was derived from a simple heuristics. Assume a unit's output is higher than the teacher signal. In this case, the input weights which have been carrying an activation should be reduced. Likewise, if the output is below the teacher signal, the weights of active input lines have to be increased. Weights of non-active input lines are left unchanged in both cases. Formally, this rule can be written as

$$\Delta w_j = \lambda (T - e) s_j, \tag{4.17}$$

which evaluates to a positive change if the output was too low and a negative change if the output was too high. $\lambda \in \mathrm{I\!R}^+$ is a "learning rate"; it can be used to enforce convergence of the learning process by gradually decreasing it to zero.

This simple idea assumes that it is known which weights are "responsible" for a given output. However, this is often not the case in more complex network topologies. We therefore consider a more systematic approach. To this end, we define the

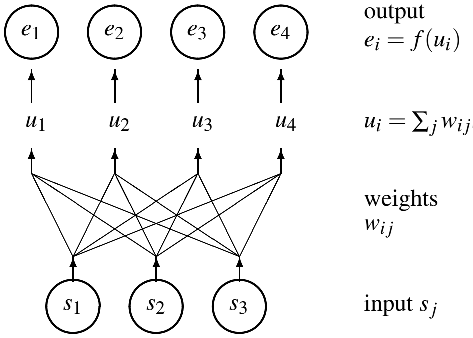


Fig. 4.7 Set of four two-layer perceptrons $e_1, \ldots, e_4$ with common input vector $s_1, \ldots s_3$. The topology is a complete feed-forward connectivity between the input and output layer, just as in the hetero-associator shown in Fig. 4.11.

performance error of the network by comparing actual and desired outputs. This is formulated in terms of an error function depending on the network weights. Learning then becomes a minimization problem, i.e. it amounts to finding a set of weights such that the error is minimized.

We start by rewriting the activity of a simple perceptron as in Equation 4.12:

$$e(\vec{s}) = f(\sum_{j=1}^{J} s_j w_j) = f((\vec{s} \cdot \vec{w})) \tag{4.18}$$

where f is a suitable non-linearity including a possible threshold. In this section, we will assume that f is differentiable. Instead of the step non-linearity, we might therefore use a sigmoidal function.

Since we know what the answer of the perceptron to a stimulus $\vec{s}$ should be—it is defined by the teacher signal $T(\vec{s})$—we can now define a performance of the perceptron via the squared error E :

$$E(\vec{w},\vec{s}) := [e(\vec{s}) - T(\vec{s})]^2 = \left[f\left(\sum_{j=1}^{J} s_j w_j \right) - T(\vec{s}) \right]^2 . \tag{4.19}$$

Clearly, this is the error for an individual stimulus presentation. Eventually, we would want to calculate the average error over all stimuli and minimize this by choosing the best possible weight vector. The average error for a given training set of stimuli is given as the average of $E(\vec{w},\vec{s}^{\,t})$ over all stimuli $\vec{s}^{\,t}$ in the training set. We can therefore drop the argument $\vec{s}$ and obtain an error function depending solely on $\vec{w}$. The optimal weight set is the one minimizing the error function $E(\vec{w})$.

Gradient Descent

Minimization of functions of several variables is a general and important topic of scientific computing. If the error function (also known as cost function or objective function) is differentiable, the following logic is generally used:

The error function represents a "surface" over the space spanned by its variables, in our case $w_1,...,w_J$ (see Fig. 4.8a). Note that this space can be identified with the feature space. Every error value is an "elevation" in that landscape and the optimum (minimum) corresponds to the deepest "depression" or trough in that landscape.

For finding this minimum we need to consider the partial derivatives of E with respect to each coordinate w_j. Fig. 4.8a shows a point (a,b) in the (w_1,w_2)-plane and its surface elevation $E(w_1,w_2)$. We now consider a vertical section through the landscape passing through (a,b) and running in parallel to the coordinate axis w_1. Within this section, the error function reduces to $E(w_1,b)$, i.e. a function of just one variable w_1 since b is constant. The derivative of this function at $w_1 = a$ is called the partial derivative of E at (a,b) in direction w_1:

$$E_{w_1}(a,b) = \frac{\partial E}{\partial w_1}(a,b) := \lim_{h \to 0} \frac{E(a+h,b) - E(a,b)}{h}. \tag{4.20}$$

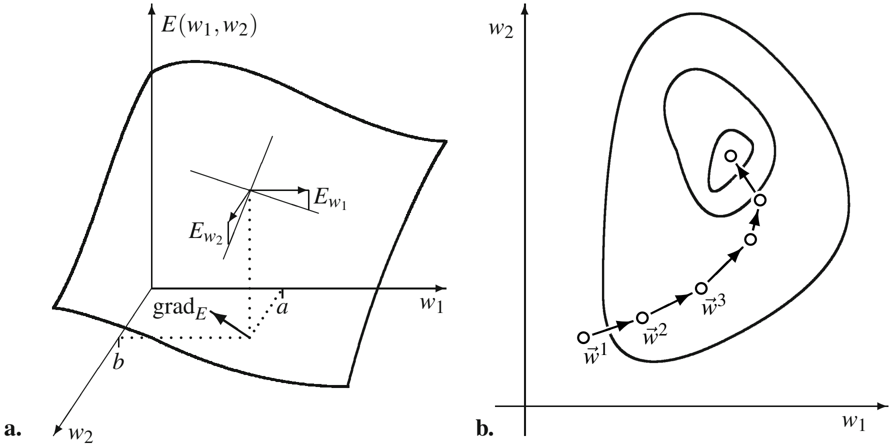


Fig. 4.8 Learning as error minimization for a two-dimensional example. **a.** The error function is shown as a surface over the weight space. At a point (a,b) in weight space, the partial derivatives are defined via tangents to the surface in the coordinate directions. The gradient is the direction of steeped ascent. **b.** The error function is shown as a contour plot. The optimum corresponds to the deepest trough in the error-surface. Numerical optimization searches this trough by starting at a random location $\vec{w}^1$, and calculating the local gradient, i.e. the steepest upward direction. It then moves the current weight estimate by a certain step in the negative gradient direction, i.e. downhill, and iterates. If a minimum is reached, the gradient evaluates to zero and the procedure stops. Note that this approach can be caught by "local minima", since it cannot take into account distant troughs.

Figure 4.8a shows for the surface point $(a,b,E(a,b))$ the tangent line in the sectional plane together with a unit vector in the w_1 direction. The difference at the tip of this vector is the partial derivative. The same logic can be applied to the w_2 direction,

$$E_{w_2}(a,b) = \frac{\partial E}{\partial w_2}(a,b) := \lim_{h \to 0} \frac{E(a,b+h) - E(a,b)}{h}, \tag{4.21}$$

and indeed for each individual dimension in the multi-dimensional case.

For a differentiable function of J variables, J such partial derivatives exist. They are combined into a vector called the "gradient" of the function, $grad_E$ or ∇E (read nabla E):

$$\mathrm{grad}_E(a,b) = \left(\frac{\partial E}{\partial w_1}(a,b), \frac{\partial E}{\partial w_2}(a,b) \right). \tag{4.22}$$

The gradient direction is the direction of steepest ascent on the surface; i.e. it is always orthogonal to the surface contour lines, pointing uphill. The length of the gradient is local slope. The gradient is defined at every point in the domain of the function, i.e. it establishes a vector field.

In gradient descent, minimization starts at a random initial position, marked as $\vec{w}^1$ in Fig. 4.8b. The gradient is calculated at this position and the current weight estimate is shifted in the opposite (downhill) direction. In standard numerical analysis, the optimal length of this shift is often determined by additional computations which are generally not used in perceptron learning. In any case, the shift will lead to a new estimate $\vec{w}^2$ of the weight vector with a reduced error. By iterating this procedure, a local minimum of the error surface can be reached. The process is stopped if the gradient is close to the zero vector, i.e. if there is no remaining local slope.

The δ-Rule

In neural networks, as in biological learning, training is usually sequential. That is to say, one stimulus is presented and the weight vector is adjusted. Then another stimulus is presented and again a small learning step is carried out. For one such step, we can therefore treat $\vec{s}$ in Equation 4.19 as a constant and reduce E by an appropriate shift of $\vec{w}$. We therefore calculate the partial derivatives of $E(\vec{w})$ with respect to the weights, using the chain rule:

$$\frac{\partial E}{\partial w_k}(w_1,...,w_J) = 2\left[f\left(\sum_{j=1}^{J} s_j w_j\right) - T(\vec{s})\right] \frac{\partial}{\partial w_k} f\left(\sum_{j=1}^{J} s_j w_j\right). \tag{4.23}$$

Next, we substitute from Equation 4.18 and apply the chain rule a second time:

$$\frac{\partial E}{\partial w_k}(w_1,...,w_I) = 2\left[e(\vec{s}) - T(\vec{s})\right] f'\left(\sum_{j=1}^{J} s_j w_j\right) \frac{\partial}{\partial w_k} \sum_{j=1}^{J} s_j w_j. \tag{4.24}$$

As in the previous section, we write $u := \sum s_j w_j$. The derivative of the sum is simply the coefficient s_k since all other terms in the sum do not depend on w_k. We obtain:

$$\frac{\partial E}{\partial w_k}(w_1,...,w_J) = 2\left[e(\vec{s}) - T(\vec{s})\right]\ f'(u)\ s_k. \tag{4.25}$$

Equation 4.25 gives the gradient direction in the error landscape E. By comparison with Equation 4.17, we see that this rule which was based on a simple heuristic, does indeed move the weight vector in the negative gradient direction, i.e. downhill.

We now introduce the notation

$$\delta := (T(\vec{s}) - e(\vec{s}))\ f'(u) \tag{4.26}$$

and obtain the so-called δ or Widrow–Hoff learning rule:

$$\Delta w_k = \lambda\ \delta\ s_k. \tag{4.27}$$

In our iterative learning process, we start with some (maybe random) weight vector $\vec{w}^1$ where the upper index denotes the current classification trial (Fig. 4.8b). We now present a stimulus and obtain a response from which we calculate δ and apply the learning rule. The new weight set will generate a reduced error if the last stimulus is presented again. However, we now proceed to present the next stimulus and again calculate δ. Strictly speaking, the optimization in each step therefore refers to a different error function, namely the one for the current stimulus. In perceptron learning, this somewhat surprising approach works better than a more systematic approach, where each weight vector would be tested with all stimuli before performing the next descent step. This is at least partially due to the problem of overlearning which occurs for fixed training sets of stimuli. Optimizing to changing stimuli introduces an element of randomness that renders the eventual result more robust.

For sets of perceptrons with shared input, i.e. a layer of perceptrons as depicted in Fig. 4.7, we can consider each output neuron individually and obtain

$$\Delta w_{ij} := \lambda \delta_i s_j = \lambda (T_i - e_i) s_j. \tag{4.28}$$

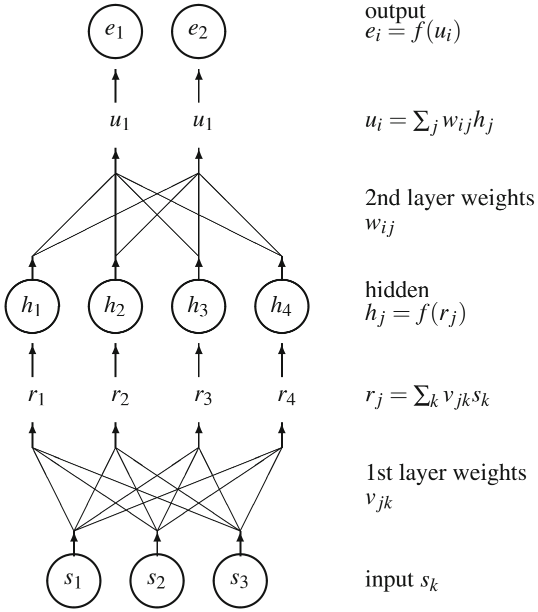


Fig. 4.9 Multi-layer perceptron with $K = 3$ input units, $J = 4$ hidden units and $I = 2$ output units. Note that the input and hidden layers are identical to the two-layer perceptrons depicted in Fig. 4.7. s_k input unit activities, v_{jk} input-to-hidden weights, r_j potentials of hidden units, h_j hidden unit activities, w_{ij} hidden-to-output weights, u_i potentials of output units, e_i output unit activity, f static non-linearity.

Multi-layer Perceptrons: Back-Propagation

In multi-layer perceptrons, the response of an output neuron is determined not only by its own weight vector but also by the weight vectors of all hidden neurons. Still, the overall approach used so far can be applied: we can formulate the error as a function of all weights and than perform a gradient descent in the resulting error landscape. With some calculation, it is straight forward to derive the following error minimization algorithm known as back-propagation learning (Rumelhart et al., 1986). As before, we present a stimulus and compare the output with the teacher signal. From this the output-layer corrections δ_{ij} can be obtained using Eq. 4.28

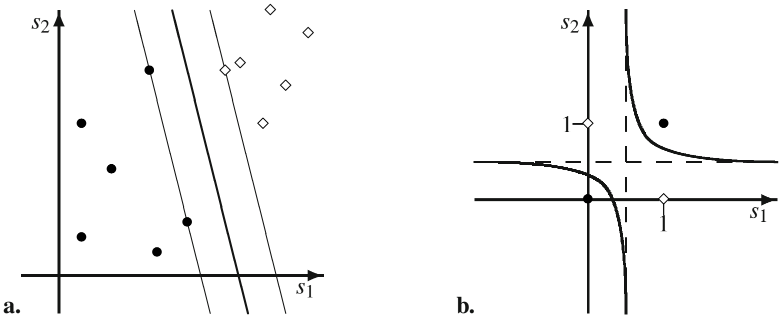


Fig. 4.10 Support vector machine. **a.** Between the two data clusters, the linear decision boundary (heavy line) is chosen such that the margin between the two clusters is maximized. In the two-dimensional case depicted here, this is obtained by choosing three points (two from one cluster and one from the other) defining two parallel lines between which no data points fall (thin lines). The midline between these parallels is the decision boundary. **b.** Solution of the XOR-problem using the kernel $(s_1, s_2, s_1 s_2)$ (Eq. 4.30) and the weights $(w_1, w_2, w_3) = (-1, -1, 2)$ in Eq. 4.31.

and the according weight update $\Delta w_{ij} = \lambda \delta_i h_j$ is applied. In addition, hidden-layer corrections ε_j can be obtained from the output-layer corrections via

$$\varepsilon_j = f'(r_j) \sum_i \delta_i w_{ij}, \tag{4.29}$$

from which the hidden layer weights are updated as $\Delta v_{jk} = \lambda \varepsilon_j s_k$. Eq. 4.29 is called "back-propagation" because the summation is over the first index of the weights, as if the correction signals would run backwards from the output units to the hidden units and were summed there according to the hidden-to-output weights. Of course, this is just an elegant mathematical result of the gradient calculation in the multi-layer case, not an actual neurobiological process.

4.2.5 Support Vector Machines

The back-propagation and δ-rules are derived from the requirement to minimize the overall classification error in a training sample. One general problem with this approach is overlearning and generalization, i.e. the performance of the classifier with novel data. In this situation, it often turns out that back-propagation has picked up on a feature difference which happened to work in the training set, but leads to false classifications in novel data.

One systematic approach to avoid this problem is to look for classification boundaries maximizing the decision margin, i.e. the distance in feature space between the boundary and the closest data point. This approach will give the highest possible robustness in generalization, since novel data will differ from the training set in statistical ways. Fig. 4.10a illustrates the idea in the two-dimensional case. The data-points defining the boundary are the "support vectors" after which the approach has been named. In the n-dimensional case, $n+1$ such vectors will be needed to define a $(n-1)$-dimensional hyperplane and the margin. Algorithms for calculating the support vectors and decision boundaries from a set of data can be found in textbooks of machine learning, e.g., Schölkopf & Smola (2002), Haykin (2008). These algorithms can be relaxed to allow for a small number of data points to fall within the margin, which is necessary if the two data clusters overlap.

So far, the support vector machine is yet another example of linear classification, since the boundaries are again hyperplanes. Curved decision boundaries (non-linear classification) can be achieved by the so-called kernel-approach in which the original data vector, in the two-dimensional case $\vec{s} = (s_1, s_2)^\top$, is replaced by a higher-dimensional function $\Phi(\vec{s})$ such as

$$\Phi(\vec{s}) := (s_1, s_2, s_1 s_2)^\top. \tag{4.30}$$

This embedding in a high-dimensional space is reminiscent of the multilayer-perceptron approach, where the input stimulus is re-coded by a number of hidden units. In the example given in Equation 4.30, the original data points are mapped to a three-dimensional space. In practical applications, embeddings to higher dimensional spaces are used. The idea is, that linear separation in a higher dimensional space is generally more powerful than in lower-dimensional spaces. In any case, the optimal linear decision boundary in the high dimensional space is determined according to the maximal margin procedure sketched in Fig. 4.10a. It leads to a hyperplane given by the equation

$$(\Phi(\vec{s}) \cdot \vec{w}) - \theta = 0 \tag{4.31}$$

where $\vec{w}$ is the normal vector of the hyperplane and θ specifies its position. In the example of Equation 4.30, Equation 4.31 can be rewritten as

$$w_1 s_1 + w_2 s_2 + w_3 s_1 s_2 - \theta = 0. \tag{4.32}$$

Solving this equation for s_2 yields a functional expression for curved decision surface in the original (s_1, s_2) feature space:

$$s_2 = \frac{\theta - w_1 s_1}{w_2 + w_3 s_1}. \tag{4.33}$$

This equation describes a hyperbola with a pole at $s_1 = -w_2/w_3$ and the asymptote $s_2 = -w_1/w_3$ approached as s_1 goes to $\pm\infty$. This decision surface could for example be used to solve the XOR-problem (see Fig. 4.10b). If other maps Φ are used, other classes of decision functions can be obtained.

4.3 Associative Memory

Associative networks compute mappings between an input vector and an output vector. This is a common task in the brain. For example, an input vector coding some image recorded in the retina may be associated with a motor output vector to the larynx and vocal tract, causing the utterance of the name of a person depicted in this image. In the vestibulo-ocular reflex, input data obtained from the semicircular canals my be associated with (i.e. transformed to) a motor command for stabilizing posture. In technical applications, mappings from multi-electrode recordings of the motor cortex to the joint angles of the arm have been learned and used to control the movement of arm prostheses with neural commands. The important question for neural network theory is to find mechanisms, by which such association rules can be learned.

The distinction between classification and association is not completely clear-cut, especially if "perceptrons" with multiple output neurons are considered as in Figs. 4.7 and 4.9. In this section, we will mostly consider linear neurons, i.e. identify potential and excitation, and study the correlations between the various inputs and outputs.

4.3.1 Topology: The Feed-Forward Associator

Consider a network composed of two subsets of units (layers), called input and output layer (Fig. 4.11). The activity vectors are denoted by $\vec{s} = (s_1, ..., s_J)^\top$ for the input layer and $\vec{e} = (e_1, ..., e_I)^\top$ for the output. Within each layer, there are no lateral connections; however, each cell in the output layer receives input from each cell in the input layer, described by a weight c_{ij}. Note that these conventions differ from our previous definition (Equation 4.9): c_{ii} is not the coupling of a unit with itself, but rather the coupling of two units—one in the input layer and one in output layer—which happen to have the same index number. For a linear activation transfer function (i.e. without non-linearity), we have:

$$e_i = \sum_{j=1}^{J} c_{ij} s_j \quad \text{or} \quad \vec{e} = C\vec{s}. \tag{4.34}$$

If (in agreement with Equation 4.9) we denote by W the connectivity matrix of the whole set of neurons, i.e. $(s_1, s_2, ..., s_J; e_1, e_2, ...e_I)^\top$, the weight matrix of the combined set becomes

$$W = \left(\begin{array}{c|c} 0_{JJ} & 0_{JI} \\ \hline C & 0_{II} \end{array} \right), \tag{4.35}$$

where 0_{IJ} is the $I \times J$-matrix with all zero coefficients. W describes a complete feed-forward connectivity. Note also that this topology is identical to the two-layer perceptron with multiple outputs (Fig. 4.7) although the graphical representation in Fig. 4.11 looks quite different.

4.3.2 Example: A 2×3 Associator

Consider the associator shown in Figure 4.11. Assume that we want to implement in this associator the input-output pair ($\vec{s}^1 = (1,0)^\top, \vec{e}^1 = (1,0,1)^\top$). As before, the upper index marks a presentation number, or a presentation made at a certain time step.

It is easy to see that this association is implemented by the weight matrix

$$C^1 = \begin{pmatrix} 1\,0 \\ 0\,0 \\ 1\,0 \end{pmatrix}, \tag{4.36}$$

since

$$\vec{e}^1 = C^1 \vec{s}^1 = \begin{pmatrix} 1\,0 \\ 0\,0 \\ 1\,0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}. \tag{4.37}$$

The weight matrix C^1 was found by setting to the value of 1 all weights with indices i, j for which $s_i = 1$ and $e_j = 1$. In our example, where all activities are either 0 or 1, this is equivalent to saying

$$c_{ij} = s_j \; e_i. \tag{4.38}$$

The activities s_j and e_i are the pre- and postsynaptic activities with respect to the connection c_{ij}. The product of these activities is a standard component in formalizations with the Hebb learning rule mentioned above. It can be thought of as some kind of "one-shot" learning where the weight is set and fixed after just one presentation of stimulus and desired output. In matrix notation, we obtain

$$C := \begin{pmatrix} c_{11}, c_{12} \\ c_{21}, c_{22} \\ c_{31}, c_{22} \end{pmatrix} = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} (s_1, s_2) = \vec{e} \, \vec{s}^\top. \tag{4.39}$$

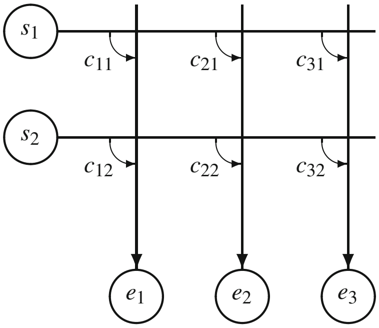


Fig. 4.11 A 2×3 associator. Each unit in the input layer (s_1, s_2) is connected to each neuron in the output layer (e_1, e_2, e_3). By suitable choices of the connecting weights, the network can be made to store "associations", i.e. transform known input pattern into arbitrary output pattern.

The multiplication of a column vector with a row vector, treating them as $J \times 1$ and $1 \times I$ matrices, respectively, is known as the *outer product*[2] of the two vectors. Equation 4.39 is therefore called the outer product rule. It states that associations can be stored by calculating the outer product of output and input vector and using it as a connectivity matrix. We will give a formally correct account of this idea below.

Of course, storing just one association pair does not lead very far. Let us therefore assume that we want to store a second pair, e.g., $(\vec{s}^2 = (0,1)^\top, \vec{e}^2 = (0,0,1)^\top)$. From the outer product rule, we obtain

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \;\Rightarrow\; C^2 = \vec{e}^2 \vec{s}^{2\top} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} (0,1) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}. \tag{4.40}$$

It turns out that, at least in this example, the two connectivity matrices C^1 and C^2 can simply be added up to obtain an associator that works for both pairs simultaneously.

$$C = C^{(1)} + C^{(2)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}. \tag{4.41}$$

Easy calculation proves the result: $\vec{e}^1 = C\vec{s}^1$ and $\vec{e}^2 = C\vec{s}^2$. We will see below that is is always true if the input vectors are orthogonal to each other.

4.3.3 Associative Memory and Covariance Matrices

Let now $\vec{s}$ be a general stimulus vector, which we want to associate with an activity vector $\vec{e}$ on the output layer. We need to find a weight matrix C satisfying the equation

$$\vec{e} = C\vec{s}. \tag{4.42}$$

Extrapolating from the example, we might expect that a matrix with this property can be found by considering the outer product of input and output vector:

$$C = \vec{e} \cdot \vec{s}^\top = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_I \end{pmatrix} \cdot (s_1, s_2, \dots, s_J) = \begin{pmatrix} e_1 s_1 & e_1 s_2 & \dots & e_1 s_J \\ e_2 s_1 & e_2 s_2 & \dots & e_2 s_J \\ \vdots & \vdots & & \vdots \\ e_I s_1 & e_I s_2 & \dots & e_I s_J \end{pmatrix}. \tag{4.43}$$

As the response of our network, we obtain:

$$C\vec{s} = \left(\vec{e} \cdot \vec{s}^\top\right) \vec{s} = \vec{e} \left(\vec{s}^\top \vec{s}\right) = \vec{e} \|\vec{s}\|^2. \tag{4.44}$$

[2] The outer product is also known as the tensor product. Its result is a matrix. Recall for comparison that the dot product discussed in Section 4.1.3, $c = \vec{x}^\top \vec{y}$, evaluates to a number. In contrast to the outer product, the dot product is also called the *inner product*.

This is the desired output, up to a factor $||\vec{s}||^2$. We can incorporate this factor in the weight matrix by choosing the coefficients $c_{ij} = e_i s_j ||\vec{s}||^{-2}$.

In general, an associative network should store not just one association, but many. We denote (as before) the various input and output vectors by raised indices, i.e. $\vec{s}^p$ is the p-th input vector. Here, p is mnemonic for presentation. For each presentation (or association pair), we can calculate the connectivity matrix by the outer product rule. As in the example, we now simply add up these outer products and obtain:

$$C = \sum_q \vec{e}^q \cdot \vec{s}^{q\top} = \begin{pmatrix} \sum_q e_1^q s_1^q & \sum_q e_1^q s_2^q & \dots & \sum_q e_1^q s_J^q \\ \sum_q e_2^q s_1^q & \sum_q e_2^q s_2^q & \dots & \sum_q e_2^q s_J^q \\ \vdots & \vdots & & \vdots \\ \sum_q e_I^q s_1^q & \sum_q e_I^q s_2^q & \dots & \sum_q e_I^q s_J^q \end{pmatrix}. \tag{4.45}$$

This matrix is essentially the matrix of the covariances that can be computed between all input units and all output units, up to their respective means (cf. Equation 4.16). Note that in Equation 4.16 the covariance was calculated over the components of a stimulus vector, whereas here, the sum runs over the presentations. Mathematically, this is no problem, but the interpretation is not the same.

C in Equation 4.45 is a “mixed” covariance matrix of $\vec{e}$ and $\vec{s}$. Later we will also encounter the (standard) covariance matrix of $\vec{s}$, $(1/P)\sum_P \vec{s}^p \vec{s}^{p\top}$.

In order to test whether the matrix C has the desired property of realizing the complete set of associations, we apply it to the p-th input pattern:

$$C\vec{s}^p = \left(\sum_q \vec{e}^q \cdot \vec{s}^{q\top}\right) \vec{s}^p = \sum_q \vec{e}^q \left(\vec{s}^{q\top} \vec{s}^p\right). \tag{4.46}$$

The term $(\vec{s}^{q\top}\vec{s}^p)$ is the dot product of the input vectors for the p-th and q-th association pair. In order to produce the correct associations, this expression has to evaluate to zero if $p \neq q$ and to one if $p = q$. This is to say that the input vectors must be pairwise orthogonal and of unit length. Therefore, if the associator has J input lines, only up to J association pairs can be stored since no more than J pairwise orthogonal vectors can be placed in the J-dimensional feature space. For larger numbers of P in Eq. 4.45, mixtures of outputs will occur.

4.3.4 General Least Square Solution

If the number of association pairs exceeds the number of input lines, one may still ask for the optimal weight set, reproducing the desired outputs as closely as possible. This question can be answered in the following way.

Let P be the number of associations. We introduce the matrices

$$S := [\vec{s}^1; \vec{s}^2; ...; \vec{s}^P] = \begin{pmatrix} s_1^1 & s_1^2 & \dots & s_1^P \\ s_2^1 & s_2^2 & \dots & s_2^P \\ \vdots & \vdots & & \vdots \\ s_J^1 & s_J^2 & \dots & s_J^P \end{pmatrix} \tag{4.47}$$

and

$$E := [\vec{e}^1; \vec{e}^2; ...; \vec{e}^P] = \begin{pmatrix} e_1^1 & e_1^2 & \dots & e_1^P \\ e_2^1 & e_2^2 & \dots & e_2^P \\ \vdots & \vdots & & \vdots \\ e_I^1 & e_I^2 & \dots & e_I^P \end{pmatrix}. \tag{4.48}$$

These are simply obtained by writing the column vectors $\vec{s}^{(p)}$ and $\vec{e}^{(p)}$ one after the other. Instead of Eq. 4.42, we may now write the relation for all presentations p jointly as

$$E = CS, \tag{4.49}$$

where E is a $I \times P$, C a $I \times J$, and S a $J \times P$ matrix.

If there are only J input pattern which are pairwise orthogonal, the matrix S will be square and unitary, i.e. $S^{-1} = S^\top$. We therefore obtain C simply from $C = ES^\top$. In the general case, i.e. if the input vectors are not orthogonal, one can show that the best possible weight matrix for the desired associations is

$$C^* = ES^\top (SS^\top)^{-1}. \tag{4.50}$$

This matrix minimizes the squared error between the desired outputs E and the actual outputs C^*. The matrix $S^\top (SS^\top)^{-1}$ is called the *Moore-Penrose pseudo inverse* of S. It also occurs in regression analysis, in the solution of general linear problems.

4.3.5 Applications

In this section, we have discussed very general ideas of associative memory, which—on the level of our presentation—amounts basically to linear regression. This idea is simple, but powerful, especially if large numbers of neurons, i.e. high dimensional feature vectors are considered. We briefly discuss three cases.

Memory

Associative networks store the associations of an input pattern, or stimulus, and an output pattern, or recall. The memory is *content-addressable* in the sense that it retrieves items not just by reference number, but by some sort of meaning. For example, one could code words (strings) of n characters into n groups of 26 input neurons (one for each letter of the alphabet), and code $k \times l$-pixel images into the activities of kl output neurons. In a training session, each image is associated with a written name and the network learns the covariances between the input neuron activities and the output neuron activities. If a string is presented at the input, the associated image will thus form at the output.

Associative memory is also *distributed* in the sense that deleting a number of internal connections will not delete specific memory items, since the output is stored in all covariances. Therefore, the result of partial deletion will be an unspecific deterioration of the output image, not the loss of individual images or memory items. By the same token, miss-spellings of the input string can be corrected. Memories with this property are also called "holographic".

Both properties, content-addressability and distributedness, are highly plausible for models of biological memories. For a classical reference, see Kohonen et al. (1976).

Attractor Neural Networks

The simple associator of Fig. 4.11 can be turned into a network with complete connectivity by feeding the output back to the "dendrites" in the upper part of the figure. Since this feedback involves some time delay, it can be considered as an iteration of an input being passed through the same connectivity matrix over and over again. In the linear case, such "auto-associators" have been used to improve a degraded pattern. In the non-linear case, so-called "attractors" will arise, i.e. special patterns of activity onto one of which the global activity state will eventually converge. Since the particular attractor reached depends on the input, attractor dynamics can be considered a mechanism of pattern classification (Hopfield 1982) .

Neuroprostheses

Robotic arms have been controlled by signals recorded from large number of neurons in the motor- and parietal cortices of a monkey. Simultaneously with this recordings, movements of the monkey's are also recorded and encoded in terms of the angles and angular velocities occurring at the shoulder and elbow joints. The "association" between the neural activities and the arm movement can be learned along the lines discussed above; indeed, this amounts to a linear regression analysis of the arm movement as a function of neural activity. Once the covariance matrix is known, new neural activities can be used to predict the "intended" arm movement and drive a robotic arm with this signal. The results show substantial agreement between the monkey's arm movements and the movements of the robot arm (Wessberg et al. 2000).

4.4 Self-organization and Competitive Learning

For a synapse with pre- and postsynaptic activities s_j and e_i, respectively, the Hebb learning rule is usually formulated as

$$w_{ij}^{t+1} = w_{ij}^{t} + \lambda \; e_i s_j \quad \text{or} \quad W^{t+1} = W^t + \lambda \vec{e}\,\vec{s}^{\top} \tag{4.51}$$

where $\lambda \geq 0$ is again a learning rate and t and $t+1$ are time steps. The left equation applies to an individual synaptic weight while the right is formulated for the entire weight matrix. In the unsupervised case considered here, e_i is simply the result of the network activity, i.e. $\vec{e} = f(W\vec{s})$ where f is the static non-linearity introduced in Equation 4.4. If we substitute this activation dynamics into Equation 4.51 we obtain a difference equation for W depending on the input pattern:

$$\Delta W = \lambda \vec{e}\,\vec{s}^{\top} = \lambda f(W\vec{s})\vec{s}^{\top}. \tag{4.52}$$

If, as a first approximation, the activation function is assumed linear, we can omit the function f and obtain $\Delta W = \lambda W\vec{s}\,\vec{s}^{\top}$. Since the sum of outer products $\vec{s}\vec{s}^{\top}$ taken over time is the covariance matrix of the input set, we can expect that the weight matrix eventually learned by the system will reflect the statistical properties of the stimulus set. Indeed, in competitive learning, it can be shown that neural networks have the ability to form statistically optimal representations by means of self-organization.

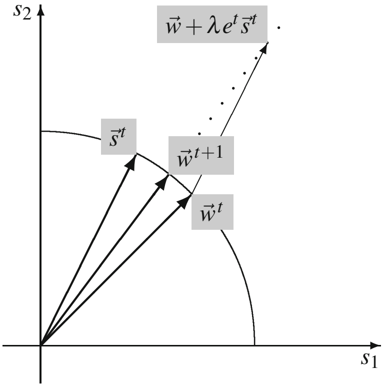


Fig. 4.12 Geometric interpretation of Oja's learning rule, Eq. 4.54. At time t, the weight vector is $\vec{w}(t)$. Assume now that a stimulus $\vec{s}^t$ is delivered. The new weight vector $\vec{w}^{t+1}$ is obtained by adding a certain multiple of the stimulus vector to the old weight vector and normalizing the sum. The tip of the weight vector moves on a circle (due to the normalization) towards the side where the stimulus occurred. If the system reaches a steady state, the weight vector will be pointing to the center of the cloud of input vectors delivered to the system during training.

4.4.1 The Oja Learning Rule

One problem with the Hebb learning rule as formulated above is that the weights will grow without limit if appropriate stimuli are presented. To avoid this, one may postulate that the norm of the weight vectors, $\sum_j w_{ij}^2$, is kept constant, say 1, during the entire learning process, so that if one weight is increased, all others will be decreased in an unspecific way.

Here, we consider a version of this idea that leads to mathematically straightforward results. This rule was suggested by Oja (1982) for the input weights of just one neuron without non-linearity. The "network" topology is that of a simple linear neuron or two-layer perceptron where the weight matrix reduces to the weight vector of this neuron. The situation is modeled by the activation function

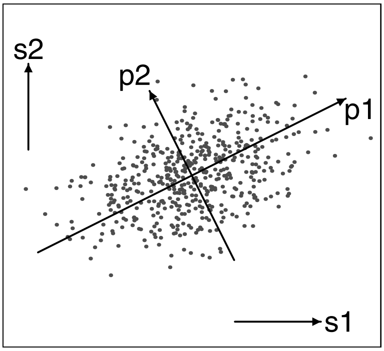


Fig. 4.13 Principal component analysis (PCA). A sample of input vectors $\vec{s}^p$ for a network with N input units forms a cloud of dots in the N-dimensional feature space. In many applications, it is useful to recode these vectors in a new coordinate system whose first axis is the axis of longest elongation of the cloud of dots. The further axes are determined as being orthogonal to the already defined axes and having the largest elongation subject to the orthogonality constraint. Mathematically, these axes are computed as the eigenvectors of the covariance matrix of the data set. They are called principal components of the data set.

$$e^t := \sum_{j=1}^{J} w_j s_j^t = \vec{w}^\top \vec{s}^{\,t} \tag{4.53}$$

and a set of input vectors presented as a temporal sequence $\vec{s}^{\,t} := (s_1^t, ..., s_J^t)^\top$. We now introduce the normalizing Hebbian rule (also known as Oja's learning rule):

$$w_j^{t+1} = \frac{w_j^t + \lambda e^t s_j^t}{\sqrt{\sum_{j=1}^{J}(w_j^t + \lambda e^t s_j^t)^2}}; \quad \vec{w}^{t+1} = \frac{\vec{w}^t + \lambda e^t \vec{s}^t}{\|\vec{w}^t + \lambda e^t \vec{s}^t\|}. \tag{4.54}$$

The denominator guarantees that $\|\vec{w}\| = 1$ at all times.

The resulting development of the weight vector is illustrated in Fig. 4.12 for a neuron with just two input lines. Due to the normalization, the total length of the weight vector is constant; that is to say, during learning, all the weight vector can do is move with its tip on a circle, or (hyper-)sphere. The changes of the weight vector, $\Delta\vec{w} = \vec{w}^{t+1} - \vec{w}^t$ will therefore always be orthogonal to $\vec{w}$, i.e. tangent to the sphere of which $\vec{w}^t$ is a radius.

We now use this property to reformulate the learning rule Equation 4.54, thereby dropping the time argument. Without the normalization, $\Delta\vec{w}$ consists only of the Hebbian term $e\vec{s}$, multiplied with the learning rate λ. The normalization can be approximated by subtracting a vector from $\lambda e\vec{s}$ which lies in a plane with $\vec{w}$ and $\vec{s}$ and makes the total weight change orthogonal to $\vec{w}$. It is easy to prove that these conditions are satisfied by

$$\Delta\vec{w} = \lambda e[\vec{s} - e\vec{w}] \tag{4.55}$$

since $\|\vec{w}\| = 1$. Equation 4.55 can be formally derived from Equation 4.54 by a Taylor expansion of $\Delta\vec{w}$ about $\lambda = 0$. Alternatively, $\Delta\vec{w}$ in Equation 4.55 can be considered the result of a Gram-Schmidt orthogonalization of $\vec{w}$ and $\vec{s}$. For our discussion, it suffices to take it as an approximation of Equation 4.54.

Next, we substitute from Equation 4.53, keeping in mind that $\vec{w}^\top \vec{s} = \vec{s}^\top \vec{w}$ due to the commutativity of the dot product. This yields

$$\Delta\vec{w} = \lambda[\vec{s}\underbrace{\vec{s}^\top\vec{w}}_{e} - \underbrace{\vec{w}^\top\vec{s}}_{e}\underbrace{\vec{s}^\top\vec{w}}_{e}\vec{w}]. \tag{4.56}$$

Due to the associativity of matrix multiplication, we may bracket the outer products $\vec{s}\vec{s}^\top$. In the temporal average, i.e. for many stimulus presentations, these outer products become the covariance matrix of the training set, $C := (1/t_{max})\sum_t \vec{s}\vec{s}^\top$. We obtain:

$$\Delta\vec{w} = \lambda(C\vec{w} - (\vec{w}^\top C\vec{w})\vec{w}). \tag{4.57}$$

The weight vector will converge when $\Delta\vec{w} = 0$. For the resulting weight set, we obtain

$$C\vec{w} = (\vec{w}^\top C\vec{w})\vec{w}. \tag{4.58}$$

The term in braces, $(\vec{w}^\top C\vec{w})$, is a number, let's call it μ. Equation 4.58 therefore becomes the eigenvalue[3] equation $C\vec{w} = \mu\vec{w}$, i.e. the weight vector will converge to an eigenvector of C. Moreover, Oja (1982) has shown that $\vec{w}$ will converge to the eigenvector with the largest eigenvalue, i.e. the perceptron will evaluate the loading (coefficient) of the first principal component of the data set.

This result means that by the normalizing Hebb rule, a perceptron will automatically adjust its weight vector to the first principle component of the data set. If we think of the set of input data as a cloud of dots in feature space, the first principle component is the longest axis of this cloud (Figure 4.13). This result was to be expected from our graphical consideration in Fig. 4.12 since the weight vector is attracted by all stimulus vectors in cloud and thus ends up in the center of that cloud. Since the activation function of the perceptron performs a projection on the weight vector (Section 4.2.2), this result therefore means that the Oja rule automatically finds the axis representing the largest possible variability of the data set in just one number. Thus, the information conveyed is maximized.

Note that learning only occurs if the output e is positive. This means that the perceptron needs to start with a weight vector not too distant from the stimulus. Also, if the data set consists of two distinct and sufficiently distant clouds of dots, the weight vector will pick up on the one which is closer to its start position.

4.4.2 Self-organizing Feature Map (Kohonen[4] Map)

There are various ways to generalize these ideas to networks with many neurons. Simply adding another output neuron does not help, because in this case both weight vectors might converge to the same direction in feature space, which is clearly not an optimal representation.

[3] The notion of an eigenvalue in matrix theory is completely analogous to its usage in Fourier theory, Section 3.2. The matrix is considered a mapping between vectors. For a vector and a number satisfying $M\vec{v} = \lambda\vec{v}$, $\vec{v}$ is an eigenvector and λ is the corresponding eigenvalue.

[4] Teuvo Kohonen (born 1934). Finnish mathematician and computer scientist

Let us consider a linear hetero-associator (Fig. 4.14a), i.e. a feed-forward projection of an input layer with stimuli $\vec{s} = (s_1, s_2, ..., s_J)^\top$ to an output layer $\vec{e} = (e_1, e_2, ..., e_I)^\top$. Inside the output layer, there are no lateral connections, but the output neurons are arranged in a grid, such that a distance between any two output neurons is defined. This distance function is a novel feature which was not considered in previously discussed neural network models. It may be used to define a neighborhood $\mathcal{N}_k$ of each unit k as the set of units whose distance to unit k is less than a certain threshold. As in Section 4.3.1, the activation dynamics is given by an input-output weight matrix $C = \{c_{ij}\}_{i \leq I, j \leq J}$:

$$e_i = \sum_{j=1}^{J} c_{ij} s_j := \vec{c}_i^\top \vec{s}. \tag{4.59}$$

Here, $\vec{c}_i = (c_{i1}, c_{i2}, ..., c_{iJ})^\top$ is again the vector of all input weights of unit i, i.e. the receptive field of the unit. The so-called Kohonen map uses the following competitive weight dynamics. Let

$$k := \operatorname*{argmax}_{1 \leq i \leq I} \{e_i\} \tag{4.60}$$

be the index number of the output unit generating the strongest excitation after a given stimulus presentation. This unit will be the one whose weight vector most closely resembles the input vector; it is usually the "winner"-unit. After each stimulus presentation, the winner unit is selected and subjected to a learning step which will make it react even stronger when the same stimulus occurs the next time. Learning is also applied to the units in the neighborhood of the winner, i.e. these units as well will react stronger if the last stimulus re-occurs:

$$\text{for all } i \in \mathcal{N}_k\text{:} \quad \vec{c}_{i\cdot}^{\,t+1} := \frac{\vec{c}_{i\cdot}^{\,t} + \lambda^t \vec{s}^t}{\|\vec{c}_{i\cdot}^{\,t} + \lambda^t \vec{s}^t\|}; \quad \lambda^t := \frac{\lambda^o}{t} \in \mathrm{IR}. \tag{4.61}$$

The learning rule itself is again a normalizing Hebb rule. Thus, there are two points where competition occurs, first in the selection of a "winner neuron" and its neighborhood which are eligible for learning, and second among the various input weights of these neurons since weight increase at one input line results in unspecific reduction of all other weights. The time-dependence of the learning rate is introduced to enforce convergence of the procedure. This type of enforced convergence is also known as "annealing".

Fig. 4.14b,c illustrate the learning process in a self-organizing feature map. The coordinate system plotted represents the unit-hypersphere on which the weight vectors $\vec{c}_i$ move. In the three-dimensional case, the coordinates $\sigma_{1,2}$ on this hypersphere can be thought of as elevation and azimuth. The shaded area shows the region in feature space where stimuli are likely to occur. The network starts with a random initialization (Fig.4.14b) where the (σ_1, σ_2)-coordinates of each output unit are independent of the unit's respective position in the neighborhood grid, the map therefore looks scrambled. During learning, the weight vectors move into the gray area because at each learning step, the winner and its neighbors are attracted by a

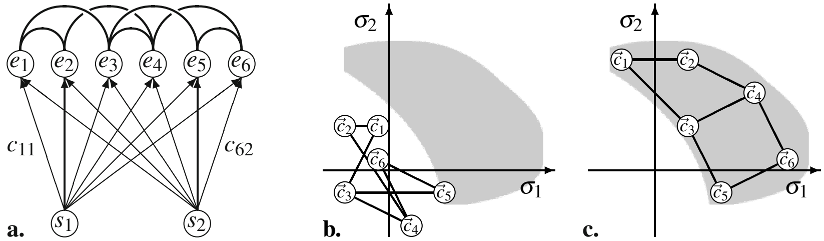


Fig. 4.14 Self-organizing feature map. **a.** Network topology with input layer (s_j), input weights (c_{ij}), map layer (e_i) and map layer adjacencies, in this example with the topology of a 2 × 3 grid. **b.** Unit-hypersphere in the input space, with coordinates σ_1, σ_2. For each map unit e_i, its input weight vector $(c_{i1}, c_{i2})^\top$ is marked as c_i. The heavy black lines show the map layer adjacencies. The weights are initialized at random, leading to an unordered arrangement of weight vectors. The gray area marks the region in feature space where input probability is high. **c.** After learning, the weight vector grid has "crawled" into the interesting area of feature space. It also has "unfolded" such that adjacent units now have similar weight vectors (i.e., receptive fields).

stimulus vector from within that range (Fig. 4.14c). At the same time, the map unfolds and ends up in a "neighborhood-preserving" (or continuous) fashion where adjacent output neurons have similar weight vectors. This continuity is a result of the shared learning in the neighborhoods. Input pattern with high frequency of occurrence will be represented by more units (larger grid regions) than rare pattern.

Kohonen maps or similar forms of competitive learning underly a large class of models for self-organization of representations, for example for orientation columns in the visual cortex or for representational plasticity in the somatosensory cortex. For examples, see von der Malsburg (1973) or Antolik & Benard (2011).

4.5 Suggested Reading

Books

Arbib, M. (ed.), (2002). *The Handbook of Brain Theory and Neural Networks*. 2nd edition, Cambridge, MA: The MIT Press

Haykin, S. (2008). *Neural Networks and Learning Machines*. 3rd edition, Upper Saddle River, NJ: Pearson Prentice Hall.

Minsky, M. L. and Papert, S. A. (1988). *Perceptrons, expanded edition*. Cambridge, MA: The MIT Press.

Shepherd, G., and Grillner, S. (eds.), (2010) *Handbook of Brain Microcircuits*. Oxford University Press

Original Papers

Antolik, J., Bednar, J.A. (2011) Development of maps of simple and complex cells in the primary visual cortex. *Frontiers in Computational Neuroscience* 5 — 17 — 1-19

Recent example of a self-organization model for cortical orientation selectivity, taking into account differences between cortical layers and cell-types.

Buchsbaum, G., Gottschalk, A. (1983) Trichromacy, opponent colours coding and optimum colour information transmission in the retina. *Proceedings of the Royal Society (London) B* 220:89 – 113.

Shows that color opponency can be interpreted as a recoding of the cone receptor signals via principle component analysis. One of the most compelling examples of an application of this theory in neuroscience.

Carreira-Perpiñán, A. Á., Lister, R. J., Goodhill, G. J. (2005) A computational model for the development of multiple maps in primary visual cortex. *Cerebral Cortex* 15:1222 – 1233

This paper uses a slightly different approach to self-organization to model the interlaced structured of cortical maps for various neuronal specificities. The results are in good agreement with neurophysiological data.

von der Malsburg, C. (1973). Self–organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85 – 100.

One of the first papers demonstrating that the formation of cortical orientation columns might be a result of competitive learning.

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

Modern classificator model based on the multi-layer perceptron. The layers are interpreted as areas of the visual and temporal cortices. Pattern classification abilities are tested with real images.

J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. A. L. Nicolelis. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408:361 – 365, 2000.

Large numbers of neurons are recorded from the cortex of a monkey and the activities are correlated to joint angles and joint velocities of the monkey's forearm. By statistical methods akin to those described in this chapter, arm movements can be predicted from the neural activities. The quality of these predictions is assessed by using them to control a robot arm and compare the movements of robot and monkey arm.

Chapter 5
Coding and Representation

Abstract. Neuronal activity does generally not in itself contain information about the stimulus. Spikes elicited by different stimuli look quite the same, but do generally occur in different cells. The information is contained in the specificity, or tuning curve of the cell generating the spike. This may be considered a modern account of the notion of the "specific sense energies" formulated by Johannes Müller[1] already in 1826. In effect, any stimulation of the eye (or visual pathways) is perceived as visual and any stimulation of the ear (or auditory pathways) as auditory. The information encoded by each neuron is described by its tuning curve. Often, neurons are tuned simultaneously to different parameters such as position in visual space, edge orientation, and color, albeit to various extends (i.e., with sharper or coarser tuning curves). Tuning curves of different neurons overlap, leading to population coding where each stimulus is represented by the activities of a group, or population, of neurons. The first part of this chapter explores the consequences of population coding for neural information processing. In the second section, we study the fact that neighboring neurons in the cortical surface tend to have similar receptive fields and tuning curves. This similarity is defined for a combination of many parameters including position in the visual field as well as the well-known "perceptual dimensions" orientation, spatial frequency, color, motion, and depth. It is particularly evident for the tuning to visual field position, i.e. in retinotopic mapping of the visual field onto the cortical surface in the centimeter range.

5.1 Population Code

5.1.1 Types of Neural Codes

Action potentials do not in their shape encode information about the stimulus. Rather, neural coding is based on the temporal pattern of neuronal firing which, for a single neuron, is represented as a spike train. As explained in Section 2.1.5, we may derive from the spike train a time-dependent spike rate which we identify

[1] Johannes Peter Müller (1801 – 1858). German physiologist.

H.A. Mallot: *Computational Neuroscience*, SSBN 2, pp. 113–129.
DOI: 10.1007/978-3-319-00861-5_5

with our excitation variable $e(t)$. In this section, we assume that spike rate carries the major part of neural information, as is generally also assumed in single-cell neurophysiology. A related variable is the local field potential measured by low-resistance extracellular electrodes which approximately is a local spatial average of spike rate. In addition to spike rates and local field potentials, higher-order pattern in spike trains have been studied which are revealed by auto- and cross-correlation (cf. Equation 2.57) or other techniques from time series analysis.

Consider a continuous variable taking values in the interval $(0,1)$. In principle, there are at least three ways of coding such variables in neural activity, or spike rate:

1. Intensity code: The activity of the neuron is proportional to (or a monotonic function of) the coded parameter. Image contrast is coded in this way, leading to stronger activities if contrast is increased. One problem of this coding scheme is that the dynamic range of the neuron (0 to $<$ 500 spikes per second) determines the resolution of the code.
2. Channel coding or labeled line code without overlap (Fig. 5.2a): The set of all possible stimulus values—in our example the interval $(0,1)$—is partitioned into n parts, each of length $1/n$. For each part i, a detector neuron exists which is active if and only if the stimulus falls within the interval $(\frac{i-1}{n}, \frac{i}{n})$. The tuning curve of each detector neuron is one on this interval and zero elsewhere.
3. Population code or labeled line code with overlap (Fig. 5.2b): Each stimulus is coded by a set or population of neurons each of which is tuned to a section of the coded stimulus. The tuning curves may overlap. This is the standard way of neural coding.

Population codes need more than one neuron to encode a parameter value, whereas in simple channel coding, one neuron may suffice. Despite this apparent disadvantage, it can be shown that population codes are superior to non-overlapping channels in many respects. We will now turn to the question of information content.

5.1.2 Information Content of Population Codes

In information theory (e.g. Cover and Thomas 1991), the information of a message is defined as the average number of binary questions (i.e. questions that can be answered either "yes" or "no") needed to reconstruct that message. The unit of information, i.e. the number of binary questions needed, is the "bit". For example, if the message is a string of zeros and ones, both numbers occurring equally likely, then each number contains one bit of information. In this case, the question to ask is "is the symbol a one" (or "is the symbol a zero"). Now consider the case that the probability of a one is only $q_1 = 0.25$ while the probability of a zero is $q_0 = 0.75$ (see also Fig. 5.1). In this case, one might start by asking "are the next two digits 00?" The probability that this is true is $p_1 = 0.75^2 = 0.5625$. If the answer is "no", the next question could be "are the next two digits 01?". The probability that this is true is $p_2 = 0.25 \times 0.75 = 0.1875$. If the answer is "no", the third question would be "are the next two digits 01?". Since any answer to the third question will make the result clear, the probability of being done after exactly three questions is

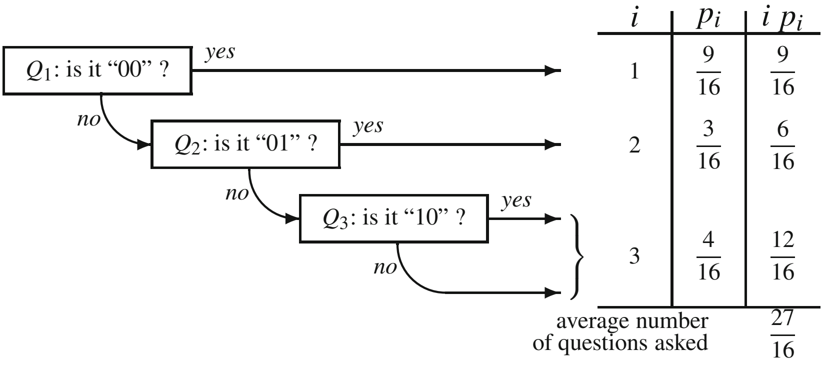


Fig. 5.1 Questioning scheme for recovering a sequence of the characters "0" and "1" appearing with probability $3/4$ and $1/4$ respectively. i, number of questions asked; p_i, probability of finding the answer after the i-th question. Each cycle recovers two digits by asking on average $27/16 = 1.6875$ questions, i.e. roughly 0.84 binary questions per digit.

$p_3 = 0.25 \times 0.75 + 0.25 \times 0.25 = 0.25$. With this question answered, we have reconstructed the next two digits. The average number of questions to be asked in this example is

$$I = p_1 + 2p_2 + 3p_3 \tag{5.1}$$

$$= \frac{9}{16} + 2\frac{3}{16} + 3\frac{4}{16} = \frac{27}{16} = 1.6875. \tag{5.2}$$

Per digit, this message therefore contains only $1.688/2 = 0.844$ bits.

The optimal way of asking questions is such that each question comes out "yes" or "no" about equally likely. By asking n such questions, one arrives at alternatives with probability 0.5^n. Put differently, if p_i is the probability of a given alternative, the number of questions asked to get its outcome in the optimal case is $\log_{0.5} p_i = -\log_2 p_i$. The average number of questions asked in an optimal scheme is therefore

$$H := -\sum_{i=1}^{n} p_i \log_2 p_i. \tag{5.3}$$

H is called the Shannon[2] entropy, or information content, of the distribution $(p_1, ..., p_n)$. It is assumed that the digits of the message occur statistically independent of each other.

We may now apply these ideas to different neural coding schemes. We will assume that the stimulus parameter q is uniformly distributed on the interval $(0,1)$. Consider first the labeled line code without overlap and n equally spaced channels (Fig. 5.2a). Each channel has the characteristic, or tuning curve

[2] Claude E. Shannon (1916 – 2001). United States mathematician and engineer.

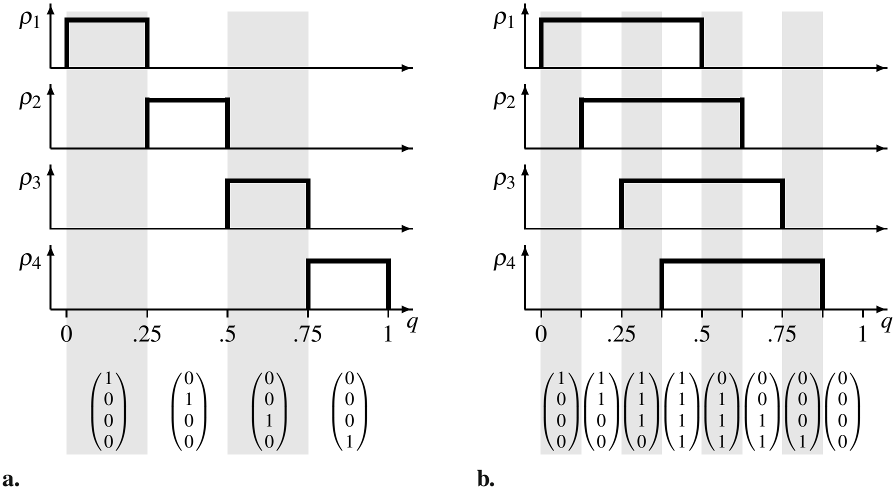


Fig. 5.2 Information transmission in channel-coded systems without overlap (**a.**) and with overlap (**b.**) With overlap, more information can be transmitted. The vertically printed vectors are the channel activities $(\rho_1(q), \rho_2(q), \rho_3(q), \rho_4(q))$ for each q-Interval. For further explanation see text.

$$\rho_i(q) = \begin{cases} 1 \text{ if } \frac{i-1}{n} < q \leq \frac{i}{n} \\ 0 \text{ otherwise} \end{cases} . \tag{5.4}$$

By coding the signal in this scheme, it is digitized to steps of $1/n$, each encoded by one of the activity patterns $(1,0,..,0), (0,1,...,0), ..., (0,0,...,1)$. Smaller differences cannot be resolved.

The probability of each of these n activity patterns to occur is $1/n$, since we assumed that the parameter q is equally distributed in the interval $(0,1)$. With n equally likely activity patterns, we may calculate the entropy

$$H = -\sum_{i=1}^{n} \frac{1}{n} \log_2 \frac{1}{n} = \log_2 n. \tag{5.5}$$

Information content increases with the number of channels, but only very slowly; information content per channel decreases as the number of channels increases.

Consider now a coding scheme with overlapping channels as shown in Fig. 5.2b. The width of each tuning curve is $1/2$, and the tuning curves of the n channels are shifted by $1/(2n)$:

$$\rho_i(q) = \begin{cases} 1 \text{ if } \frac{i-1}{2n} < q < \frac{i-1}{2n} + \frac{1}{2} \\ 0 \text{ otherwise} \end{cases} . \tag{5.6}$$

As can be seen from Figure 5.2b, there are are $2n$ different activity distributions (or code words) in this case. Therefore, the entropy is:

$$H = -\sum_{i=1}^{2n} \frac{1}{2n} \log_2 \frac{1}{2n} = 1 + \log_2 n. \tag{5.7}$$

As compared to the case without overlap, information content is increased by one bit. Put differently, carrying the same information in the interval code would require twice as many neurons. Further increase can be obtained by allowing for channels with graded activities. For example, the three channels in color vision (short, middle, and long wave-length cones) allow to distinguish between more than a million colors. In contrast, graded activities do not add any information in the non-overlapping case.

5.1.3 *Reading a Population Code: The Center of Gravity Estimator*

For the nervous system, having the information about a stimulus represented in an activity pattern over a population of neurons is perfectly okay. For the experimenter, however, it would be helpful to recover the original signal from the population activity, in order to know what is actually represented in a given activity pattern.

Assume we are encoding a parameter p, where p can be anything, from the frequency of a light wave to the retinal position of a stimulus, or, indeed, the direction of a planned arm movement. The parameter is represented by a population of n neurons, each with a preferred parameter value $\hat{p}_i$ (read: p hat sub i). The preferred parameter value is the one for which the neuron's tuning curve takes its maximum. In color vision, these are the peak positions of the quantum catch functions of the cone receptors. The preferred stimulus values must be known to the experimenter in advance, e.g., from an independent measurement of tuning curves. We now present a particular stimulus value, say q, and take the resulting activities of each neuron, $e_i(q)$, as a "vote" for the neuron's preferred stimulus. The votes are weighted with activity strength and summed together. Mathematically, we can describe the procedure by

$$q^* = \frac{\sum_i \hat{p}_i e_i(q)}{\sum_i e_i(q)} \tag{5.8}$$

where q^* is the estimate for q. If only two neurons are considered with $\hat{p}_{1/2} = \pm 1$, Equation 5.8 describes a balance with two weights corresponding to the activities of the neurons and q^* marks the point along the arm of the balance where it needs to be supported in order not to tilt to either side. The right side of Equation 5.8 is therefore called the "center of gravity"-estimator.

Of course, it would be nice to prove in Equation 5.8 that the estimate q^* is indeed equal to the original parameter value q. This, however, depends on the detailed shape of the tuning curves $\rho_i(q)$; for example, with cosinusoidal tuning curves, the center-of-gravity estimator is generally biased. The question of the exactness of the center-of-gravity estimator can be considered as an instance of the general problem of function approximation with the tuning curves as "basis functions". We will not elaborate on this, but study a simple example illustrated in Fig. 5.3.

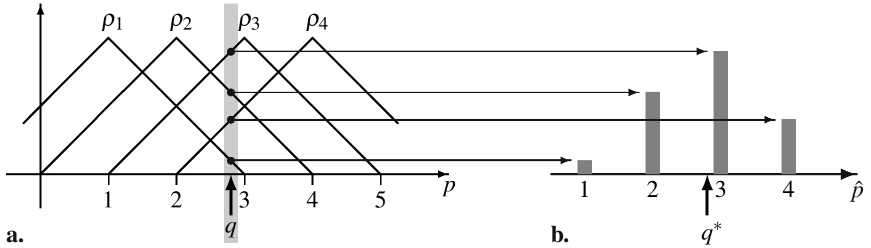


Fig. 5.3 a. An array of equidistant triangular tuning curves (Equation 5.9). Also shown is a stimulus value q which excites all four channels according to their tuning curves. **b.** The activities generated in these four channels. Their center of gravity equals the original stimulus.

Suppose a population of neurons with triangular tuning curves $\rho_i(p)$ centered at the preferred stimulus $\hat{p}_i$,

$$\rho_i(p) = \begin{cases} 1 - \frac{1}{2}|p - \hat{p}_i| & \text{for } |p - i| < 1 \\ 0 & \text{otherwise} \end{cases} . \tag{5.9}$$

Fig. 5.3a shows a set of four such tuning curves where the preferred stimuli of each neuron have been identified with the neuron's number, i.e. $\hat{p}_i = i$.

Now, consider a stimulus with parameter value q; in Figure 5.3a it appears between the preferred stimuli of neurons 2 and 3. From Equation 5.9 it is easy to calculate the activities in the four channels. They are shown in Figure 5.3b as gray columns. As explained before, we consider these columns to be loads placed on the beam of a balance. Since we now have more than two channels, each load is placed at the position corresponding to the channel's preferred stimulus. The definition of the center-of-gravity as support-point of the beam remains the same. In the case of triangular tuning curves, it is easy to show that the center-of-gravity estimator actually recovers the encoded parameter value, $q = q^*$ in Equation 5.8.

If the equality $q = q^*$ does not strictly hold, the center-of-gravity estimator is still useful as an approximation; for an interesting example, see Kay et al. (2008). In addition, more sophisticated methods of estimation theory have been applied to the problem, most notably the idea of maximum likelihood estimation. It interprets tuning curves as conditional probabilities for neuronal firing given that a particular stimulus was presented. For a recorded pattern of population activity, one can then calculate for each possible stimulus the probability of the recorded pattern to occur. This probability is called the likelihood[3] of the stimulus. Finally, one picks the one stimulus generating the largest likelihood.

[3] Note that the likelihoods of all stimuli need not sum to unity. This is why the term "likelihood" is introduced, to distinguish the quantity from a true probability.

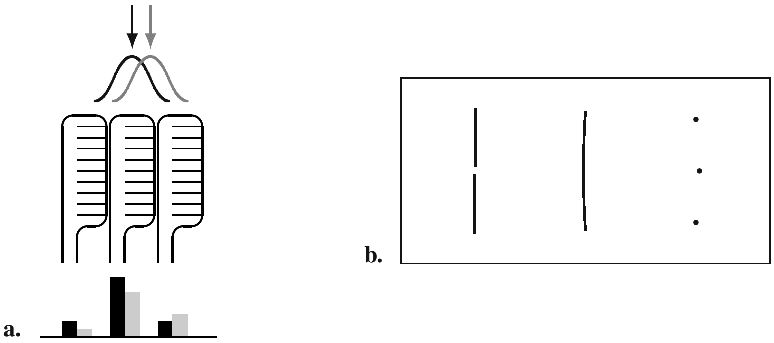


Fig. 5.4 Hyperacuity (or sub-pixel resolution) in a population code for visual position. **a.** The arrows mark visual positions for two ideal point lights. By the imaging properties of the eye, they are washed out into two blurring disks on the retina (shown as Gaussians). The width of blurring under ideal conditions is about 0.5 minutes of arc. These pattern excites the cone photoreceptors which in the fovea have a diameter of again 0.5 minutes of arc. Each blurring disc will therefore excite a small population of adjacent cone receptors. The activity pattern over these group differs even if the light positions are less than 0.5 minutes of arc apart. **b.** Test patterns for hyperacuity: left, vernier, middle, arc, right, row of dots. Well trained subjects correctly judge offset directions for offsets below 10 seconds of arc, i.e. less than one third of the cone diameter.

5.1.4 Examples, and Further Properties

Hyperacuity (Sub-Pixel Resolution)

If visual position is considered as a stimulus parameter, the receptive field functions as were studied in Chapter 2 can be identified with the tuning-curves for the parameter "space". The perception of visual position from overlapping receptive fields is thus an example of population coding, since the position of a point stimulus is encoded not by the activity of just one neuron, but by the population activity of all neurons whose receptive field includes the position in question.

Fig. 5.4 shows population coding of visual position already on the level of retinal cone receptors. The set of all visual locations from which a given cone receptor can be stimulated is determined by the width of the cone receptor (about 30 seconds of arc in terms of visual angle or 1.7 μm in terms of retinal distance) and the width of the blurring disk of a light beam arriving at the cornea. For ideal imaging conditions (iris light adapted, exact accommodation of lens), this is again in the order of 30 seconds of arc. Fig. 5.4a shows this situation for two locations less than 30 seconds of arc apart. In either case, three adjacent cone receptors will be activated, but the amount of activation differs between the two locations. The population activity, or the center-of-gravity estimator calculated from it, maintains the information about the separation between the two stimuli (cf. Poggio et al. 1992). In a labeled line code without overlap, stimulus locations can only be told apart if they exceed two cone widths.

Fig. 5.4b shows some patterns which may be used to measure visual resolution. The versions of the patterns depicted, or their mirror images, are shown to the test subject. Then the subject is asked whether the upper line is to the left or the right of the lower one (example at the left); in which direction the arc is bent (middle example), or whether the middle of the three points is to the left or the right of the line connecting the two other points (right example). All of these experiments reveal perceptual thresholds on the order of 10 seconds of arc or less, i.e. well below the resolution of the cone mosaic.

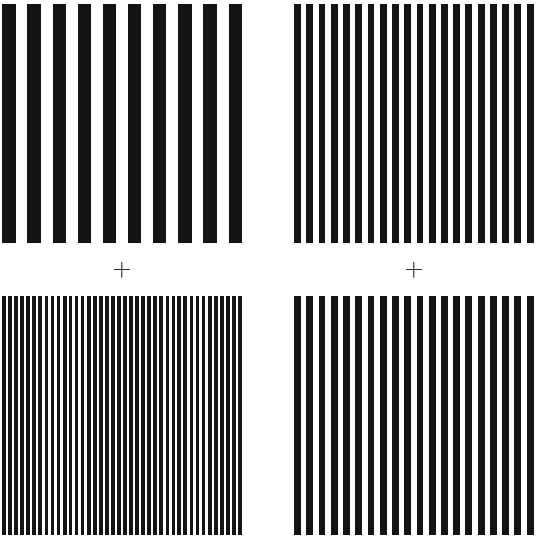


Fig. 5.5 Demonstration of the shift in perceived spatial frequency after adaptation. After closing one eye and looking at the small cross at the left for at least one minute, then looking at the cross on the right, the two striped patterns at the right appear to have different spatial frequencies (Adapted from Blakemore & Sutton 1969).

After-Effects and Work-Range Adjustment

Sampling a parameter space with overlapping tuning-curves allows to adjust the sensitivity of parameter contrasts to the variability or statistics of the input signal. On a relatively short time scale (minutes), this is evident from so-called after-effects which are an ubiquitous phenomenon in perception. Fig. 5.5 shows an example from the perception of the spatial frequency, or granularity, of a grating. After monocularly fixating the left fixation cross for a minute or so, the upper and lower parts of the right display appear to differ in spatial frequency. The visual system seems to adapt to the spatial frequency on the left side and after adaptation conveys the new spatial frequencies in relation to the ones adapted to. Similarly, after adapting to the downward movement in a waterfall, still objects appear to move upward; after adapting to a red pattern, the same pattern shown only by a black outline on a white background appears greenish, etc.

Neurophysiologically recoded tuning curves support the idea that many sensory parameters such as orientation and spatial frequency of visual textures, motion,

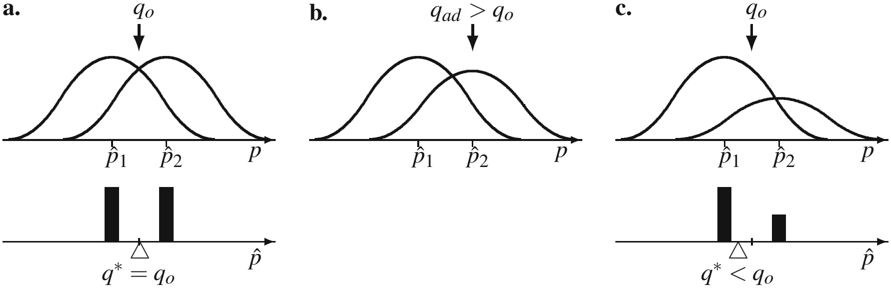


Fig. 5.6 After-effects are a common phenomenon in the perception of motion, color, texture granularity, etc. A standard explanation assumes that perception is based on the population activity of channels with overlapping tuning curves. **a.** A stimulus q_o is presented intermediate between the preferred stimuli $\hat{p}_1$ and $\hat{p}_2$ of two channels. The resulting population activity shows two equal excitations and the center of gravity estimator q^* truly reproduces the original stimulus q_o. **b.** The system now adapts to a new stimulus, q_{ad}. Adaption is modeled as a reduction of the sensitivity of active channels. **c.** If the original stimulus is again presented to the adapted system, the activity of the previously active channel will be reduced. The center-of-gravity estimator is therefore displaced away from the adapting stimulus, i.e. to the left.

color, or depth, are represented in a population code. In this coding scheme, adaptation is modeled as an overall decrease of sensitivity caused by strong and sustained activity of a given channel in the adaptation phase. This decrease affects the channels most closely tuned to the adapting stimulus and may be due to fatigue, learning processes or other mechanisms. In any case, the sensitivity of the channels tuned to the adapting stimulus will be reduced in the test phase of the experiment. Thus, the population code will be biased away from the adapting stimulus (Fig. 5.6). In ongoing perception, channel adaptation will lead to an improved detection of changes, or of deviations from the average. The perception of constant stimuli will be suppressed while more sensitivity will be assigned to parameter ranges with higher variability.

Vector-Valued Parameters and Interpolation

The theory of population coding is not restricted to stimuli varying in just one parameter. As an example, we consider the encoding of pointing movements of the arm, which can be described in a two-dimensional parameter space of azimuth and elevation from the shoulder (Georgopoulos et al. 1993). Cells in the motor cortex are active prior to arm movements into certain portions of the grasp space. One can define the motor field of a cell as the probability of each movement given that the cell was active. These motor fields are analogous to tuning curves in the sensory systems. Different cells in the motor cortex have different, but overlapping motor fields. One may now ask the question whether it is possible to predict the motor action based on the activities of a large number of motor cells. To do this, the motor fields are first determined and for each cell, the preferred motion vector is identified. Then, if a

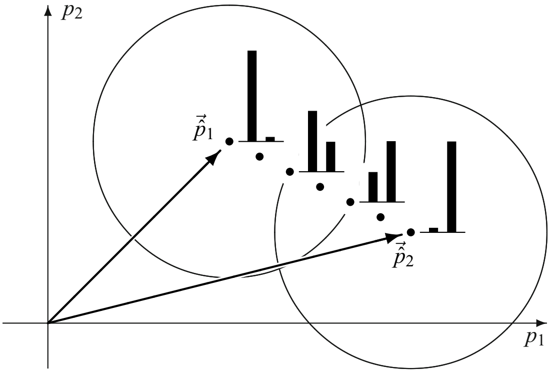


Fig. 5.7 Population coding for multi-dimensional parameters. In a two-dimensional parameter space p_1, p_2, two units (channels) are marked with preferred stimuli $\vec{p}_{1,2}$. In this case, the tuning curves become two-dimensional functions $\rho_i(p_1, p_2)$ which peak at $(p_1, p_2) = \vec{p}_i$. Pattern of population activity are shown for a number of intermediate stimuli $q_\lambda := \lambda\vec{p}_1 + (1-\lambda)\vec{p}_2$, with $\lambda = 0, 0.33, 0.67, 1$.

pattern of activity is given on the motor cell population, one may calculate the center of gravity estimator (Equation 5.8) where it is understood that the scalar "preferred stimuli" $\hat{p}_i$ of each neuron are to be replaced by vectorial "preferred movements" $\vec{p}_i$. Note that Equation 5.8 works just as well with vectorial quantities, since the activities e_i always remain scalars. Georgopoulos et al. (1993) use the term "population vector" for the center-of-gravity estimator of vectorial quantities. It turns out that by means of this population vector, the actual arm movement is nicely predicted. If the population vector is monitored over time while the monkey is planning a movement, it can be seen anticipating this movement ("mental rotation").

Fig. 5.7 shows the general situation for a two-dimensional parameter space. The tuning curves (or motor fields) are bell-shaped curves in that space, centered at the cell's preferred parameter value $\vec{p}_i \in \mathbb{R}^2$. The interpolation between two such preferred parameter vectors is realized by decreasing activity in one channel and increasing activity in the other. By means of the center-of-gravity calculation, this amounts to an interpolation of the population estimate. If the two preferred stimuli are different pointing directions, the described shift of activity from one channel to another amounts of a rotation of the population vector.

Two- or higher-dimensional tuning curves are are generally modeled as "radial basis functions" (RBF):

$$\rho_i(\vec{p}) := f(\|\vec{p} - \vec{p}_i\|) \tag{5.10}$$

where f is a monotonically decreasing function on $\mathbb{R}_o^+$ such as the Gaussian (Poggio & Girosi 1990). In the figure, the tuning curves are symbolized by circles. If the preferred stimuli (RBF-centers) are suitably distributed in parameter space, continuous variation of the parameter values can be represented by the relative activities in the population.

5.1.5 Summary

Population coding is an ubiquitous scheme of neural coding which has a number of characteristic properties, which play also a role in neural information processing. Some of these properties are:

- *Information content*: Population codes can convey more information than non-overlapping labeled lines. The most compelling example is color vision where three channels can code for some two million colors, differentiated in hue, saturation and brightness.
- *Hyperacuity (sub-pixel resolution)*: Population codes inherently allow for a resolution better than the channel spacing. Hyperacuity in visual line localization is a clear example.
- *After-effects and the adjustment of working-ranges*: Since individual parameter values are coded by the balance of activity in overlapping channels, adaptation of individual channels leads to distorted percepts. Overall, aftereffects tend to emphasize contrasts, rather than static situations. If the average value of a sensory parameter changes, the system can adjust its working-range.
- *Interpolation and mental rotation* : Reading population codes by the center of gravity estimator implies an interpolation operation. Examples include the coding of arm movements in cortical motor neurons, eye-movements to multiple targets, both in the fronto-parallel plane and in depth, and the hippocampal place-field code to location. The different phenomena described as mental rotation (correlation of reaction time and turning angle in same-different judgments, neural representation of arm movements) can be interpreted as direct consequences of the interpolation property of population codes.

5.2 Retinotopic Mapping

In the visual cortex, information from the different parts of the retina is systematically ordered, resulting in what is called a retinotopic map or retinotopic representation. This map can be charted by recording from cortical neurons and then determining which location on the retina delivers peak stimulation to each cortical location. The cortex of primates has a large number (> 20) of areas each one of which contains a more or less retinotopic image of the retina.

Here we will briefly discuss models for retinotopic maps given in terms of mappings or coordinate transforms from the retina to the visual cortex. We denote such mapping by curly capital letters:

$$\mathscr{R} : \mathbb{R}^2 \to \mathbb{R}^2, \quad \mathscr{R}(x,y) := (u,v). \tag{5.11}$$

The retinal and cortical coordinates are given by the two-dimensional vector (x,y) and (u,v), respectively.

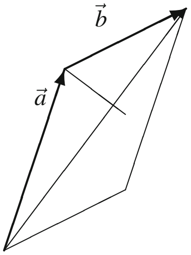


Fig. 5.8 The area of a parallelogram. A parallelogram may be described by the vectors $\vec{a} = (a_1, a_2)^\top$ and $\vec{b} = (b_1, b_2)$. After dividing it into two triangles along the diagonal $\vec{a} + \vec{b}$, the surface area may be calculated as the base times the height. The length of the base is $\|\vec{a} + \vec{b}\|$. The normal vector in the direction of the height is $(-a_2 - b_2, a_1 + b_1)^\top / \|\vec{a} + \vec{b}\|$. The height may be obtained by projecting $\vec{a}$ onto this normal. Then the surface area may be calculated: $A = |a_1 b_2 - a_2 b_1|$.

5.2.1 Areal Magnification

An important issue in retinotopic mapping is areal magnification, i.e. the cortical representational area devoted to some patch of visual field or retina. In this section, we derive an expression for the magnification of a transformation between two-dimensional areas.

We first note that the surface area of a parallelogram defined by the vectors $(a_1, a_2)^\top$ and $(b_1, b_2)^\top$ (cf. Fig. 5.8) is

$$A = |a_1 b_2 - b_1 a_2| = \left| \det \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \right| . \tag{5.12}$$

The areal magnification of a linear mapping L,

$$L \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_{11}x + a_{12}y \\ a_{21}x + a_{22}y \end{pmatrix},$$

can be determined by examining the square defined by the vectors $\vec{a} = (1, 0)^\top$ and $\vec{b} = (0, 1)^\top$ in the domain of the mapping. The area of this square is 1. Its image is the parallelogram defined by the vectors $L(\vec{a}) = (a_{11}, a_{21})^\top$ and $L(\vec{b}) = (a_{12}, a_{22})^\top$. According to the rule discussed above, its area is $|a_{11}a_{22} - a_{12}a_{21}|$. Since the area of the original square was 1, the new area is also the areal magnification of the linear mapping. For the linear mapping, this magnification is the same everywhere.

For a general (differentiable) mapping $\mathscr{M} : \mathbb{R}^2 \to \mathbb{R}^2$, areal magnification can be determined by first finding a local linear approximation and than calculating the magnification of this linear mapping (cf. Fig. 5.9). A linear approximation (gray grid in Fig. 5.9) can be obtained from the matrix of the partial derivatives of the two components of the mapping $\mathscr{M}$, that is, from the Jacobian matrix of $\mathscr{M}$ (see, for example Rudin 1976):

$$J_{\mathscr{M}}(x, y) := \begin{pmatrix} \dfrac{\partial \mathscr{M}_1}{\partial x} & \dfrac{\partial \mathscr{M}_1}{\partial y} \\ \dfrac{\partial \mathscr{M}_2}{\partial x} & \dfrac{\partial \mathscr{M}_2}{\partial y} \end{pmatrix} . \tag{5.13}$$

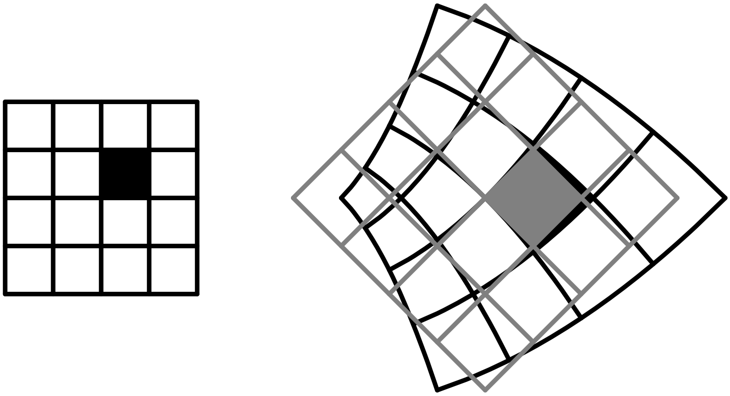


Fig. 5.9 The definition of areal magnification of a general transformation T. Left: the domain in which T is defined. Right: The image of the grid as transformed by T (black) and the local approximation of T in the vicinity of the center of the grid.

The linear approximation of a multidimensional mapping by means of the Jacobi matrix is equivalent to the approximation of a one-dimensional function by a tangent whose slope is given by the local derivative of the function. The determinant of the Jacobi matrix is the local areal magnification produced by the mapping $\mathscr{M}$.

$$|\det J_{\mathscr{M}}(x,y)| = \left| \frac{\partial \mathscr{M}_1}{\partial x} \frac{\partial \mathscr{M}_2}{\partial y} - \frac{\partial \mathscr{M}_1}{\partial y} \frac{\partial \mathscr{M}_2}{\partial x} \right| \tag{5.14}$$

The reason for having different areal magnifications in the cortical representation of the visual field is that the density of retinal ganglion cells is not constant. We can assume that, at least approximately, each retinal ganglion cell maps to the same number of cortical neurons covering a constant area of the cortical surface. In the primate retina, the distribution of ganglion cells has a peak around the fovea and declines towards the periphery in a more or less isotropical way. In order to obtain equal representation in the cortex, the mapping must therefore be distorted, with an expansion (high areal magnification) of the central retina and a compression (low areal magnification) of the periphery. If $d_g(x,y)$ denotes the density of ganglion cells and $\mathscr{R}$ represents the imaging function describing the retinotopic map, equal cortical representation of all ganglion cells amounts to

$$d_g(x,y) = c|\det J_{\mathscr{R}}(x,y)|, \tag{5.15}$$

where c is a constant and $|\det J_{\mathscr{R}}|$ is the areal magnification factor.

Equation 5.15 is a partial differential equation related to the so-called eikonal equation of mathematical physics. It does not have a unique solution. Indeed, multiple solutions of Equation 5.15 have been used to model multiple retinotopic maps in cortical areas V1, V2, and V3 (Mallot 1985).

5.2.2 Conformal Maps

The analysis can be simplified for a class of mathematical mapping functions known as conformal maps. These functions are characterized by the property that small circles are mapped to circles of variable size but not to ellipses. A related property is that lines intersecting at right angles in the domain of the function will be mapped to curves again intersecting at a locally orthogonal angle. Cortical retinotopic maps are not generally conformal; however, conformality may be used as a first approximation.

Mathematically, conformal mapping satisfies a pair of partial differential equations known as the Cauchy-Riemann equations. If $\mathscr{C}_1(x,y)$ and $\mathscr{C}_2(x,y)$ denote the components (cortical coordinates) of a mapping function $\mathscr{C}$, the Cauchy-Riemann equations read

$$\frac{\partial \mathscr{C}_1}{\partial x} = \frac{\partial \mathscr{C}_2}{\partial y} \quad \text{and} \quad \frac{\partial \mathscr{C}_1}{\partial y} = -\frac{\partial \mathscr{C}_2}{\partial x}. \tag{5.16}$$

It follows immediately that the Jacobian of a conformal function describes a pure rotation combined with a scaling, but no shear, in accordance with the mapping properties described above.

$$J_{\mathscr{C}}(x,y) = \begin{pmatrix} A & B \\ -B & A \end{pmatrix} \tag{5.17}$$

with

$$A = \frac{\partial \mathscr{C}_1}{\partial x} = \frac{\partial \mathscr{C}_2}{\partial y} \quad \text{and} \quad B = \frac{\partial \mathscr{C}_1}{\partial y} = -\frac{\partial \mathscr{C}_2}{\partial x}. \tag{5.18}$$

For cortical magnification we obtain

$$|\det J_{\mathscr{C}}| = A^2 + B^2. \tag{5.19}$$

Conformal maps can also be thought of as one-dimensional complex differentiable functions, where the real and imaginary parts of domain and range are the two-dimensional coordinates of retina and cortex, respectively. In this case, areal magnification is simply the squared absolute value (modulus) of the complex derivative. Since formally, complex derivatives behave just as real derivatives, the mapping-magnification equation can now be turned into an ordinary differential equation. As a result of conformality, the areal magnification will also be equal to the square of the length magnification of a line segment (the "linear" magnification). In monkeys, areal magnification has been shown to approximately equal $1/r^2$ where $r = \sqrt{x^2 + y^2}$ is retinal eccentricity. Eq. 5.15 then becomes:

$$\frac{1}{r^2} = c(\mathscr{R}'(r))^2 \tag{5.20}$$

for some constant c. From this, we immediately compute (see Fischer 1973):

$$\mathscr{R}(r) = \frac{1}{c} \log r. \tag{5.21}$$

This function will be discussed in the following section.

5.2.3 Log-Polar Mapping

Following the conformality assumption, we can replace r in Equation 5.21 by the complex number, $x+iy = re^{i\phi}$, where $r = \sqrt{x^2+y^2}$ and $\phi = \arg(x+iy)$.

With Euler's formula (Eq. 3.19) we then obtain:

$$\begin{aligned} \mathscr{P}(x+iy) &= u+iv \\ &= \log(|x+iy|)\, e^{i \arg(x+iy)} = \log r + i\phi. \end{aligned} \tag{5.22}$$

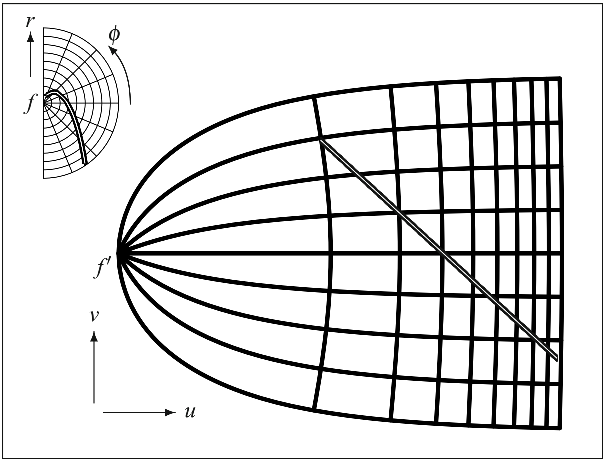


Fig. 5.10 Log-polar mapping of the visual hemifield with offset. The inset on the left shows the contralateral (right) visual hemifield with $f = (0,0)$ marking the center. The big figure shows the image of the polar grid under the log-polar map $(u,v) = \mathscr{P}_c(x,y)$ where $f' = \mathscr{P}_c(f)$ is the foveal representation. The representation of the vertical meridian delimiting the visual hemifield to the left, is the $\subset$-shaped curve. The semicircular perimeter maps to the straight line to the right. Close to the fovea, the radial structure of the retinal grid is preserved. Further to the periphery, the visual field radii become parallel lines and the iso-eccentricity circles approximate straight verticals. This effect is increasingly pronounced if the value of c is reduced. The straight line in the map, running from top left to bottom right is the image of the logarithmic spiral marked in the visual field.

The cortical u-coordinate thus corresponds to the logarithmically compressed retinal eccentricity, while the cortical v-coordinate represents the angle from the horizontal meridian.

Equation 5.22 is known as the log-polar mapping. In the fovea itself, i.e. $r = 0$, it is undefined since $\log(r)$ approaches negative infinity. Biologically, this reflects our assumption $d_g(r) = r^{-2}$ which unrealistically implies that ganglion cell density and therefore areal magnification in the retina be infinite. This can be mended by adding a small constant c in the denominator, i.e. $d_g(x,y) = 1/((x+c)^2 + y^2)$ which will prevent d_g from growing beyond $1/c$. We replace the complex numbers by two-dimensional vectors (x,y) and (u,v), respectively, and obtain

$$\mathscr{P}_c(x,y) = \begin{pmatrix} \frac{1}{2}\log((x+c)^2 + y^2) \\ \arctan \frac{y}{x+c} \end{pmatrix}. \tag{5.23}$$

The retinotopic map $\mathscr{P}_c$ is shown in Fig. 5.10 by means of a polar grid in the right visual field and the image of this grid in the left visual cortex, area V1. The plot nicely models neurophysiological measurements of visual retinotopic maps in monkeys. Also shown is a straight line in the cortical representation together with a curve in the visual field that will be mapped to this straight line under the log-polar mapping. Easy calculation proves that this curve is a logarithmic spiral. This is thought to be the reason for the spiral structure of visual hallucination patterns known for example from migraine "aura" events. In migraine aura events, a wave front known as spreading depression is moving across the visual cortex, inducing a temporary scotoma in the visual field. While the depression wave front is approximately a straight line, the subjects perceive a spirally shaped scotoma, since they (as always) interpret the visual field position of cortical activity via the inverse of the retinotopic map (see Bressloff et al. 2001.

5.3 Suggested Reading

Books

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, Inc., New York.

Kriegeskorte, N. and Kreiman, G. (eds.) (2012) *Visual Population Codes. Toward a Common Multivariate Framework for Cell Recording and Functional Imaging*. The MIT Press, Cambridge, MA.

Rieke, F., Warland, D., de Ruyter van Steveninck, R., Bialek, W. (1997) *Spikes. Exploring the Neural Code*. The MIT Press, Cambridge, MA.

Original Papers

Bressloff, P. C., and Cowan, J. D., and Golubitsky, M., and Thomas, P. J. and Wiener, M. C. (2001) Geometric visual hallucinations, Euclidean symmetry and the functional architecture of striate cortex. *Philosophical Transactions of the Royal Society London (B)* 356:299 – 330

Hallucination pattern are explained as simple waves of activity on the cortex, interpreted by the observer by "backward projection" with the retinotopic mapping function. The idea is extended to the pattern of cortical hypercolumns which is hypothesized to give rise to pattern such as the zig-zag in migraine "fortification" hallucinations.

Fischer, B. (1973). Overlap of receptive field centers and representation of the visual field in the cat's optic tract. *Vision Research*, 13:2113 – 2120.

Derivation of the logarithmic structure of retinotopic maps from the density distribution of the retinal ganglion cells.

Georgopoulos, A. P., Taira, M., and Lukashin, A. (1993). Cognitive neurophysiology of the motor cortex. *Science*, 260:47 – 52.

Demonstration of population coding in the motor cortex. This paper popularized the ideas of population coding and the center-of-gravity estimator, for which the authors introduced the term "population vector".

Kay, K. N., Naselaris, T., Prenger, R. J., and Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature* 452:352 – 356.

This paper defines a receptive field function for cortical voxels measured with functional brain imaging. Using the methods of population coding, it then shows to what extent visually presented images can be recovered from simultaneously recorded brain activity.

Poggio, T., Fahle, M., and Edelman, S. (1992). Fast perceptual learning in visual hyperacuity. *Science*, 256:1018 – 1021.

This papers presents a model of visual hyperacuity along the lines of population coding and radial basis function networks. It then proceeds to show that perceptual learning in human subjects can be simulated with the same model.

Polimeni, J. R., Balasubramanian, M., and Schwartz, E. L. (2006). Multi-area visuotopic map complexes in macaque striate and extra-striate cortex. *Vision Research* 46: 3336 – 3359

The idea of log-polar mapping the primary visual cortex area V1 is extended to adjacent areas which can be modeled by similar functions, i.e. the negative branch of the complex logarithm.

References

Adelson, E.H., Bergen, J.R.: Spatiotemporal energy models for the perception of motion. Journal of the Optical Society of America A 2, 284–299 (1985)

Antolik, J., Bednar, J.A.: Development of maps of simple and complex cells in the primary visual cortex. Frontiers in Computational Neuroscience 5(17) (2011)

Aidley, D.J.: The Physiology of Excitable Cells, 4th edn. Cambridge University Press, Cambridge (1998)

Bressloff, P.C., Cowan, J.D., Golubitsky, M., Thomas, P.J., Wiener, M.C.: Geometric visual hallucinations, Euclidean symmetry and the function architecture of striate cortex. Philosophical Transactions of the Royal Society (London) B 356, 299–330 (2001)

Barlow, H.B., Levick, R.W.: The mechanism of directional selectivity in the rabbit's retina. Journal of Physiology 173, 477–504 (1965)

Blakemore, C., Sutton, P.: Size adaptation: a new aftereffect. Science 166, 245–247 (1969)

Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, New York (1991)

Eichner, H., Joesch, M., Schnell, B., Reiff, D.F., Borst, A.: Internal structure of the fly elementary motion detector. Neuron 70, 1155–1164 (2011)

Fischer, B.: Overlap of receptive field centers and representation of the visual field in the cat's optic tract. Vision Research 13, 2113–2120 (1973)

FitzHugh, R.: Impulses and physiological states in theoretical models of nerve membrane. Biophysical Journal 1, 445–466 (1961)

Georgopoulos, A.P., Taira, M., Lukashin, A.: Cognitive neurophysiology of the motor cortex. Science 260, 47–52 (1993)

Haykin, S.: Neural Networks and Learning Machines, 3rd edn. Pearson Prentice Hall, New York (2008)

Hines, M.L., Carnevale, N.T.: The NEURON simulation environment. Neural Computation 9, 1179–1209 (1997)

Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. Journal of Physiology 117, 500–544 (1952)

Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences 79, 2554–2558 (1982)

Hartline, H.K., Ratliff, F.: Spatial summation of inhibitory influences in the eye of *limulus*, and mutual interaction of receptor units. Journal of General Physiology 41, 1049–1066 (1958)

Itti, L., Koch, C.: A saliency-based search mechanism for overt and covert shifts of visual attention. Vision Research 40, 1489–1506 (2000)

Jack, J.J.B., Noble, D., Tsien, R.W.: Electric current flow in excitable cells. Clarendon Press, Oxford (1975)

Jones, J.P., Palmer, L.A.: An evaluation of the two-dimensional Gabor filter model of simple receptive-fields in the cat striate cortex. Journal of Neurophysiology 58, 1233–1258 (1987)

Kay, K.N., Naselaris, T., Prenger, R.J., Gallant, J.L.: Identifying natural images from human brain activity. Nature 452, 352–355 (2008)

Koenderink, J.J.: Scale-time. Biological Cybernetics 58, 159–162 (1988)

Kohonen, T., Reuhkala, E., Mäkisara, K., Vainio, L.: Associative recall of images. Biological Cybernetics 22, 159–168 (1976)

Lettvin, J.Y., Maturana, H.R., McCulloch, W.S., Pitts, W.H.: What the frog's eye tells the frog's brain. Proceedings of the Institute of Radio Engineers 47, 1950–1961 (1959)

von der Malsburg, C.: Self–organization of orientation sensitive cells in the striate cortex. Kybernetik 14, 85–100 (1973)

Mallot, H.A.: An overall description of retinotopic mapping in the cat's visual cortex areas 17, 18, and 19. Biological Cybernetics (1985)

Minsky, M.L., Papert, S.A.: Perceptrons, expanded edition. The MIT Press, Cambridge (1988)

Murray, J.D.: Mathematical Biology. I. An introduction, 3rd edn. Springer, Berlin (2002)

Mallot, H.A., von Seelen, W., Giannakopoulos, F.: Neural mapping and space–variant image processing. Neural Networks 3, 245–263 (1990)

Milescu, L.S., Yamanishi, T., Ptak, K., Smith, J.C., Mogri, M.Z.: Real time kinetic modeling of voltage-gated ion channels using dynamic clamp. Biophysical Journal 95, 66–87 (2008)

Oja, E.: A simplified neuron model as a principal component analyzer. Journal of Mathematical Biology 15, 267–273 (1982)

Poggio, T., Girosi, F.: Regularization algorithms for learning that are equivalent to multilayer networks. Science 247, 978–982 (1990)

Pollen, D.A., Ronner, S.F.: Visual cortical neurons as localized spatial frequency filters. IEEE Transactions on Systems, Man, and Cybernetics 13, 907–916 (1983)

Rall, W.: Electrophysiology of a dendritic neuron model. Biophysical Journal 2, 145–167 (1962)

Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back–propagating errors. Nature 323, 533–536 (1986)

Rudin, W.: Principles of mathematical analysis, 3rd edn. McGraw-Hill, New York (1976)

Schölkopf, B., Smola, A.: Learning with Kernels. Support Vector Machines, Regularization, Optimization and Beyond. The MIT Press, Cambridge (2002)

Tuckwell, H.: Introduction to theoretical neurobiology (2 Vols.). Cambridge University Press, Cambridge (1988)

Index