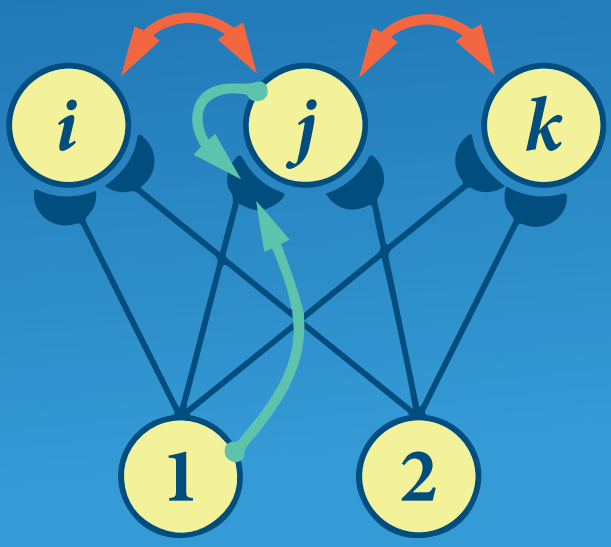
Hanspeter A. Mallot

Computational Neuroscience

An Essential Guide to Membrane Potentials, Receptive Fields, and Neural Networks

Second Edition



Springer

Computational Neuroscience

Hanspeter A. Mallot

Computational Neuroscience

An Essential Guide to Membrane Potentials, Receptive Fields, and Neural Networks

Second Edition

Springer

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

ISBN 978-3-031-75704-4 ISBN 978-3-031-75705-1 (eBook)
https://doi.org/10.1007/978-3-031-75705-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Must we not confess that geometry is the most powerful of all instruments for sharpening the wit and training the mind to think correctly? Was not Plato perfectly right when he wished that his pupils should be first of all well grounded in mathematics?

Galileo Galilei, *Discorsi e dimostrazioni matematiche intorno a due nuove scienze. Giornata seconda.* Leiden 1638. Translated by Henry Crew and Alfonso de Salvio

Preface

It is tempting to define computational neuroscience as the application of the Galilean scientific program—quantification and mathematization—to the study of the brain and its performances. Taking physics as a model, the goal would be to establish a well-ordered edifice of "brain theory" in which the workings of the brain would be described by a set of concise mathematical theorems.

To this day, however, the Galilean challenge is at best partially met: Computational neuroscience mostly deals with the quantification and mathematical description of specific phenomena and mechanisms in a number of more or less unconnected subfields, while overarching "grand laws" are largely missing. In a very coarse classification, extant brain theory may be broken down into four major parts: (i) bioelectricity and membrane biophysics together with the theory of dynamical systems, (ii) stimulus–response behavior of neurons modeled by functional analysis and the theory of signal processing, (iii) neural networks modeled by vector and matrix algebra and again dynamical systems theory, and (iv) the encoding and representation of knowledge together with statistical theories of estimation, information, and decision making. The meaning of the word "computational" varies between these parts: While the first part uses mathematics exclusively for the modeling of physiological and biophysical processes, i.e., as "computation in neuroscience," the latter parts increasingly focus on computation as a performance of the brain, aiming at a "neuroscience of computation." This second meaning of the term "computation" underlies the brain–computer analogy stated already by Norbert Wiener, who, in his 1961 book "Cybernetics," talks of the "essential unity of problems … whether in the machine or in living tissue." Consequently, ideas and modeling approaches have been exchanged in both ways between neuroscience on one side and computer science and robotics on the other. The "neuroscience of computation" links brain theory to behavior and the adaptive value of the nervous system in ecology and evolution. It will therefore be an essential part of a fully developed theory of the brain.

Already in its present state, computational neuroscience has become an indispensable tool for brain science, supporting the quantification of measurements, the structuring of results from different neurons, brain systems, and animal species, and the prediction of the effects of experimental interventions. The book presents the basic theory following the structure given above: Membrane biophysics (Chap. 1), stimulus–response models (Chaps. 2 and 3), neural networks (Chaps. 5 and 6), and

representation and coding (Chap. 7). Chapter 4 is a mathematical detour on Fourier transforms which is helpful for a deeper understanding.

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 neuroscience. This book attempts an inverse approach. It grew out of a lecture series delivered over a period of more than 20 years to graduate students in neuro- and cognitive science, most of which had backgrounds in biology, medicine, or psychology. The mathematical requirements 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. The book also assumes some familiarity with the basic facts of neuroscience; if problems arise, any textbook of neuroscience will help.

At times, 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 that will be useful in such seminars. They are recommended for a deeper study of the subject.

For the second edition, the text and figures have been revised and augmented to improve readability and completeness but without changing the overall character of the book. Mathematical notation has been aligned with the usage in Wikipedia, which is recommended as a general reference. A series of "boxes" has been added that provides background information and links to related topics; they can be read independently of the main text. Key points are highlighted throughout the text and allow the reader to keep track of the central ideas. In short, I have tried to make it the book that I would have loved reading as a student.

Since the publication of the first edition, I again received many questions and comments that have helped improve the text. I am grateful to the publisher for useful suggestions on style and structuring of the text, to Lena Veit who now took over the lecture course at Tübingen University, to Anne Schlecht for reading large parts of the manuscript, and to my students who kept me busy thinking about better ways of presentation.

Tübingen, Germany Hanspeter A. Mallot
September, 2024

Contents

About the Author

Hanspeter A. Mallot received his PhD from the Faculty of Biology, University of Mainz, Germany, in 1986. In the following years, he held postdoctoral and research positions at the Massachusetts Institute of Technology, the Ruhr-University Bochum, the Max-Planck-Institute for Biological Cybernetics in Tübingen, and the Institute for Advanced Study, Berlin. From 2000 to 2023, he headed the cognitive neuroscience unit at the University of Tübingen, where he is currently affiliated as a senior professor. Hanspeter has published more than 100 papers and 3 books on topics of visual perception, spatial cognition, and neural information processing.

Excitable Membranes and Neural Conduction 1

Abstract

Neural information processing is based on three cellular mechanisms, i.e., the excitability of neural membranes, the spatiotemporal 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 like 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 ion 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 that is also discussed in this chapter. Mathematical background is mostly from the theory of dynamical systems.

Learning Objectives

- The molecular basis of bioelectricity and membrane potentials
- Dynamics of voltage dependent channels and how to model them with differential equations
- The Hodgkin–Huxley theory of the action potential
- Simplified models of the action potential including integrate-and-fire and state space analysis
- Passive propagation of membrane potentials and the cable equation
- Active propagation of neural activity

H. A. Mallot, *Computational Neuroscience*,
https://doi.org/10.1007/978-3-031-75705-1_1

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 the resting potential of nerve cells, i.e., the voltage of about $-70\,\mathrm{mV}$ (inside negative) across a neuron's cellular membrane. The resting potential 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., proteins containing a surplus of amino acids such as aspartate and glutamate with carboxyl groups $-COO^-$ in their side chains). 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. Cell membranes void of such channels are almost ideal insulators since charged ions with their hydration shells cannot enter the hydrophobic inner zone of the membrane. The 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.

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. The ion distributions and their dynamic changes generate three types of membrane potentials, which we will discuss in the sequel. These are (i) the equilibrium potential for individual ion types, (ii) the resting potential of the cell as a whole, and (iii) the action potential in which the channel permeabilities quickly change and generate dynamic potential "spikes" lasting for a few milliseconds.

1.1.1 Equilibrium Potentials

If the membrane potential of a neuron does not change over time, this does not mean that no ions are moving across the membrane. This would be the case if the membrane were an ideal insulator. In the living cell, however, there is always ion traffic crossing the membrane. Still, the potential will stay constant if the number of signed charges passing in either direction equals each other and the net current is zero. This situation is called a dynamic equilibrium or a steady state and plays an important role in cell physiology.

[1] The chemical symbol for sodium, Na, refers to the Neo-Latin "Natrium," named after the Wadi El Natrun in Egypt and the mineral natron ($Na_2CO_3 \cdot 10H_2O$) found there.

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

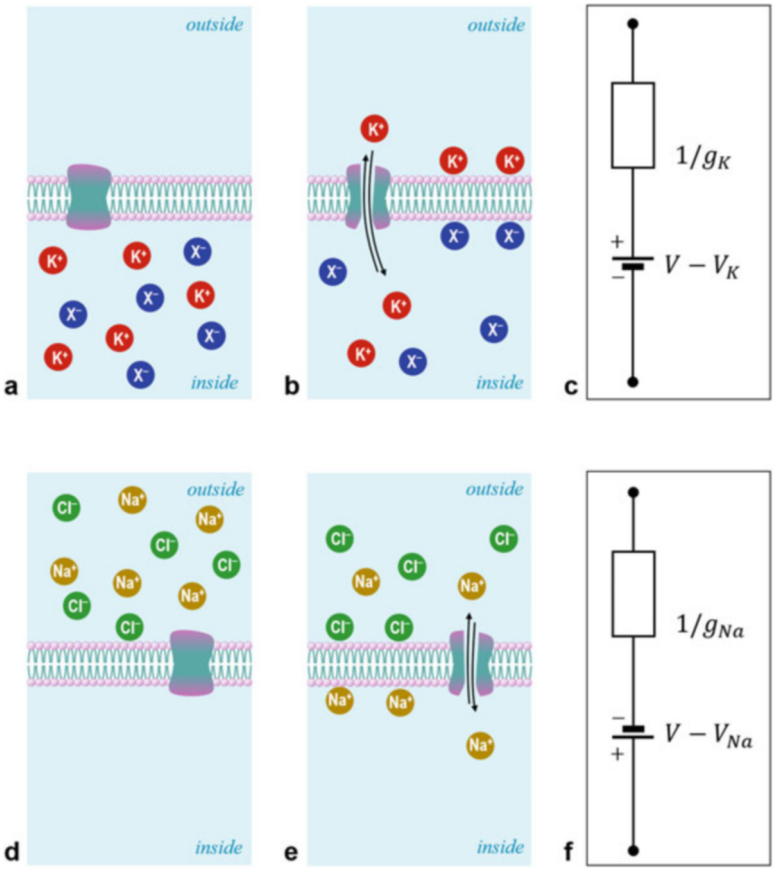


Fig. 1.1 Equilibrium potentials for individual ion sorts in an aqueous medium on two sides of the cell membrane (phospholipid bilayer). A channel is shown as a transmembrane protein. (**a**) Assume an intracellular compartment with a high concentration of K^+ ions, electrically neutralized by negatively charged proteins (anions) X^-. (**b**) If a K^+ channel in the membrane opens up, K^+ ions will move out driven by the osmotic force. Since the anions stay behind, an electric force is build up, pulling the K^+ back in. If both forces are equal, a steady state is reached known as the K^+ equilibrium potential. (**c**) Branch of an equivalent circuit showing the equilibrium potential V_K as a battery and the channel permeability as a resistor; g_K is conductivity, the inverse of resistance. In a closed circuit, the relative position of the two elements is arbitrary. As in standard conventions, current is flowing as positive charge from + to −. The polarity of the battery is such that positive charges move toward the outside as soon as the channel opens. (**d**)–(**f**) Same for Na^+ ions. The channel passes Na^+ but not the negatively charged chloride ions (Cl^-). Since the concentration is high outside the neuron, the overall polarity is reversed

Figure 1.1 shows the emergence of equilibrium, or Donnan[3] potentials for unequal distributions of sodium and potassium ions inside and outside a neuron.

[3] Frederick G. Donnan (1870–1956), British-Irish physical chemist.

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

		Ion concentrations		
		External	Internal	Nernst equilibrium
	Ion	(mmol/L)	(mmol/L)	(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

The crucial element is the "semipermeable" membrane that can be permeated by some ions but not by others. Figure 1.1a shows the situation for potassium ions K^+ of which there are more inside the neuron than outside. If a channel is opened that allows K^+ ions to pass (Fig. 1.1a), some of these will flow out driven by osmotic pressure. Since the large anions (acidic proteins) cannot follow, this leads to a separation of electric charge in the sense that the outside will contain a surplus of positively charged ions, while these are depleted in the interior. The resulting electric field will drive K^+ ions back into the cell. A dynamic equilibrium or steady state is reached if the same number of ions passes the membrane in either direction: that is, the work against the osmotic gradient (which drives ions outward) and the work against the electric field (which drives them back in) have to be equal. The equilibrium is marked by the Nernst[4] equation, $V = RT/(nF) \times \ln(c_o/c_i)$, where V is the potential in volts, c_i and c_o are the ion concentrations in the inner and outer media, n is the charge number (+1 in the case of K^+), and F, R, and T are the Faraday constant, the universal gas constant, and the absolute temperature, respectively. For details and formal derivations, see textbooks of physical chemistry and neurophysiology. The equilibrium potential can be calculated separately for each for the various ion types involved. For potassium, it is negative, since $c_o < c_i$, while for sodium the concentrations are inverted ($c_i < c_o$) resulting in a positive potential. Some typical values for ion concentrations and the resulting Nernst potentials appear in Table 1.1.

Neural membranes are also semipermeable for sodium ions of which there are more in the outside medium than inside the cell. They are electrically neutralized by the presence of chloride ions, which is to say that the intercellular medium contains a fair concentration of common salt, NaCl (Fig. 1.1d, e). If a sodium channel in the membrane opens, Na^+ will flow in, while Cl^- cannot follow. Again, an equilibrium potential results, this time however with inverted polarity, i.e., inside positive.

If a channel is opened in a previously impermeable membrane, i.e., if we go from Fig. 1.1a to b or from d to e, the equilibrium will not be reached instantaneously but as a relaxation process taking a certain amount of time. This is due to two

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

effects: First, the number of ions that can pass through the channel per unit time (i.e., the electric current) is limited. The channel acts as an Ohmic resistor whose conductance is zero if the channel is closed and takes some positive value as it opens. The relevant potential difference driving the current is the deviation from the equilibrium potential; as soon as this is reached, the current will cease. Second, the charges crossing to the other side of the membrane will not distribute evenly in their new compartment but will be electrically attracted across the membrane by the anions they left behind. They will therefore move toward the membrane where they pair up with anions on the other side. As a consequence, the potential difference resulting from a transgression of a given number of charges is not a constant but depends on the membrane area available for this pairing up. In electric circuits, this is modeled as a capacitance; the larger the capacitance is, the more ions are needed to generate a certain potential change. Since the ions move with finite speed, the relaxation process will take the longer, the larger the membrane capacitance is. In typical neurons, it is about 1 μF/cm^2.

Figure 1.1c and f shows the situation as branches of equivalent electrical circuits. The equilibrium potential is treated as a battery or voltage source. The potential generated by this source is defined by Nernst's equation. Over a time span of one or a few spike events (i.e., tens of milliseconds), it can be treated as a constant since the ion currents are small and do not substantially change the concentration in the inside and outside compartments (see below, Box 1.7). The capacitor models the mutual attraction of the ions across the membrane that acts as the insulating gap of the capacitor, while the adjacent inside and outside media are its "plates." In the equilibrium state, the capacitor can be ignored since no net current is flowing. If however, the potential is disturbed by some outside influence, capacitance acts as a buffer that dampens the resulting potential changes.

The Donnan potential is a thermodynamic equilibrium: that is, no energy needs to be supplied to maintain the potential and concentration differences as long as external influences are absent.

1.1.2 Resting Potential

If ions of two or more types are present simultaneously and can cross the membrane, the situation is more complex. The electro-motor force is generated jointly by the unequal ion distributions and will act on both ion types alike, while the osmotic forces will still depend and act on the concentrations of the individual ion types. A thermodynamic equilibrium would therefore require that the concentrations of all involved ions change until their Nernst potentials become equal. In the case of sodium and potassium ions in the nerve cell, this would mean that the ratios of inside and outside concentration would in fact level off. If this happened, the nerve cell could no longer operate.

In the living cell, the resting state is not a thermodynamic equilibrium but maintains the markedly different Nernst potentials for sodium and potassium ions in the order of $V_{\mathrm{Na}} = +60\,\mathrm{mV}$ for Na^+ and $V_{\mathrm{K}} = -80$ to $-100\,\mathrm{mV}$ for K^+. This

requires an active transport of K^+ ions into the neuron and Na^+ in the outward direction. Transport is achieved by the so-called K-Na-pump, a transport protein located in the cell membrane that consumes chemical energy provided as ATP. As a result, the concentrations of sodium and potassium remain roughly constant.[5]

The resulting membrane potential depends on the Nernst equilibrium potentials as well as on the conductances g or "permeabilities" p of the membrane for the ion sorts involved, most notably Na^+, K^+, and Cl^-. If all sodium channels are closed ($g_{Na} = 0$), while some K^+ ions are allowed to pass, the membrane potential will equal V_K. Vice versa, if $g_K = 0$ and $g_{Na} > 0$, the membrane potential will approach V_{Na}. In intermediate cases, it is a mixture of the two extreme potentials, weighted by the membrane permeabilities p of each ion type. The formal relation is given by the Goldman[6] (or Goldman–Hodgkin[7]–Katz[8]) equation of membrane biophysics; for a derivation, see Ermentrout and Terman (2010).[9] The typical value $V_{rest} = -70\,\text{mV}$ is close to the K^+ equilibrium because the conductivity is larger for K^+ than for Na^+.

Figure 1.2 shows a hydraulic analogy of the resting potential where electric potentials are visualized as hydrostatic pressure. The neuron is modeled by the water-filled tank in the lower left. The membrane potential corresponds to the water level in the ascension pipe. The Nernst potentials of Na^+ and K^+ ions are shown by the water levels in the two reservoirs. They are connected to the neuron via valves with openings g_{Na} and g_K, respectively. The membrane potential cannot rise above V_{Na} or fall below V_K. In the resting state, both valves are slightly open such that water is constantly flowing from the upper to the lower reservoir. This is compensated by the pump to the right of the figure.

We will see in the next section that the action potential is based on conductivity changes in the two "valves," which depend on the current membrane potential. This can be modeled by adding a float body in the ascension pipe that controls the opening and closing of the valves.

▶ **Key Point** We distinguish three types of membrane potentials: (i) equilibrium potentials of individual ion sorts, (ii) the resting potential combining effects of different ion sorts, and (iii) the action potential that is a dynamic and transient event.

[5] It is interesting to note that the activity of the brain's K-Na-pumps accounts for about 10% of the basic metabolic rate in humans (Laughlin et al. 1998).

[6] David E. Goldman (1919–1998). United States biophysicist.

[7] Alan Lloyd Hodgkin (1914–1998). English physiologist. Nobel Prize in physiology or medicine 1963.

[8] Sir Bernhard Katz (1911–2003). German-British biophysicist. Nobel Prize in physiology or medicine 1970.

[9] The equation reads: $V = \frac{RT}{F} \ln \frac{p_{Na}[Na^+]_o + p_K[K^+]_o + p_{Cl}[Cl^-]_i}{p_{Na}[Na^+]_i + p_K[K^+]_i + p_{Cl}[Cl^-]_o}$, where p is permeability and $[]_i$ and $[]_o$ denote concentration in the inside and outside media. If only one ion type is considered, it reduces to the Nernst equation.

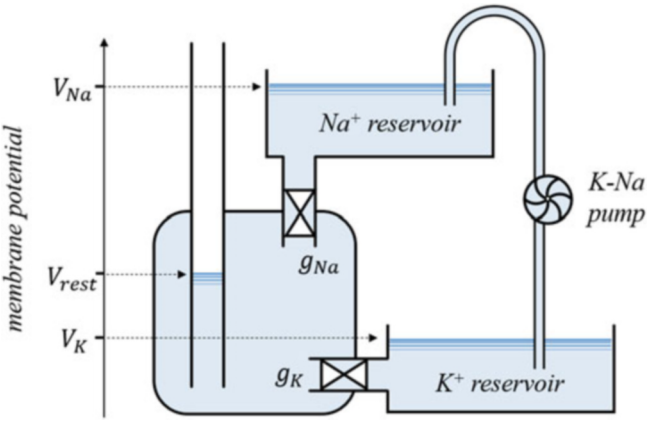


Fig. 1.2 Hydraulic analogy of the resting potential. If the Na^+-valve opens, the water level in the ascension pipe can rise to the level of the Na^+ reservoir. If it closes and the K^+-valve opens, the level in the ascension tube falls to the level of the K^+ reservoir. If both valves are partially open, an intermediate level will result, accompanied by the steady flow from the upper to the lower reservoir that has to be compensated by the pump

Box 1.1 Electricity and Bioelectricity

Electric phenomena are ubiquitous in living cells. They are generated by displaced charges within large organic molecules or by the electrically charged ions into which "salts" dissociate when put in aqueous media. Positively charged ions are called cations, and negatively charged ones are anions.

As in general electricity, the basic quantity is charge, either positive or negative. It is measured in the unit coulomb (C) or in multiples of the elementary charge of $\pm 1.6 \times 10^{-19}$ C carried by individual protons or electrons. Charges of equal or different polarity exert a repelling or attracting force on each other as described by Coulomb's law; they are therefore said to generate an electric field around them. The work energy needed to move a test charge inside this field is given by the product of force and displacement or, more correctly, by the integral of the (changing) force along the path. If we move a test charge from infinity, that is, from outside the effective field, to a given position within the field and divide the required work energy by the value of the test charge, we obtain the electric potential at that position. It is measured in volts (V) where 1 volt equals 1 joule/coulomb. The work needed to move the test charge between different points within the field can be calculated from the difference of the potentials at the start and end points of the movement. This potential difference is called voltage and is again measured in volts.

(continued)

Box 1.1 (continued)

Electric fields exist in the vacuum, inside metallic conductors, and also inside the living cell. If charge carriers such as electrons or ions are present and free to move, electric fields will affect electric currents, i.e., displacements of charges. These will be directed uphill or downhill in the potential landscape, depending on polarity. Current between two points is measured in amperes (A), where 1 ampere equals 1 coulomb per second. A list of the most important electrical quantities appears below.

quantity	symbol	SI unit	relation
charge	Q	coulomb (C)	
voltage	V	volt (V)	$\mathrm{V} = \mathrm{J/C}$
current	I	ampere (A)	$\mathrm{A} = \mathrm{C/s}$
resistance	R	ohm Ω	$\Omega = \mathrm{V/A}$
conductivity	g	siemens (S)	$\mathrm{S} = 1/\Omega = \mathrm{A/V}$
capacitance	C	farad (F)	$\mathrm{F} = \mathrm{C/V}$

While the basic laws of physics apply, bioelectricity differs from electricity in metal conductors in important ways. First of all, the carriers of electric charge are chemical substances with specific properties, not just electrons. The movement of charges, i.e., the electric current, is therefore not only dependent on charge and voltage but also on the chemical properties of the carrier and the environment. Second, ion movement in aqueous media is slowed down by strong effects of friction. Bioelectric currents are therefore generally weak. Third, temporal changes are relatively slow, such that inductance and electromagnetic effects can generally be neglected.

1.1.3 The Action Potential

During an action potential, the steady state maintained in the resting potential is quickly disturbed and then reinstated by dynamic, voltage dependent changes of the channel conductivities, and the corresponding ion flows across the membrane, see Fig. 1.3. The crucial element of the underlying mechanism is the voltage dependence of sodium and potassium channels, which open and close in response to the overall membrane potential. The channel proteins contain differently charged domains that will move in the electric field leading to changes in the three-dimensional configuration of the protein; as a consequence, channels may open or close. The presence of voltage dependent channels is the origin of neural excitability; in dendrites or the internodia of myelinated axons, where no voltage dependent channels are found, no action potentials can therefore occur.

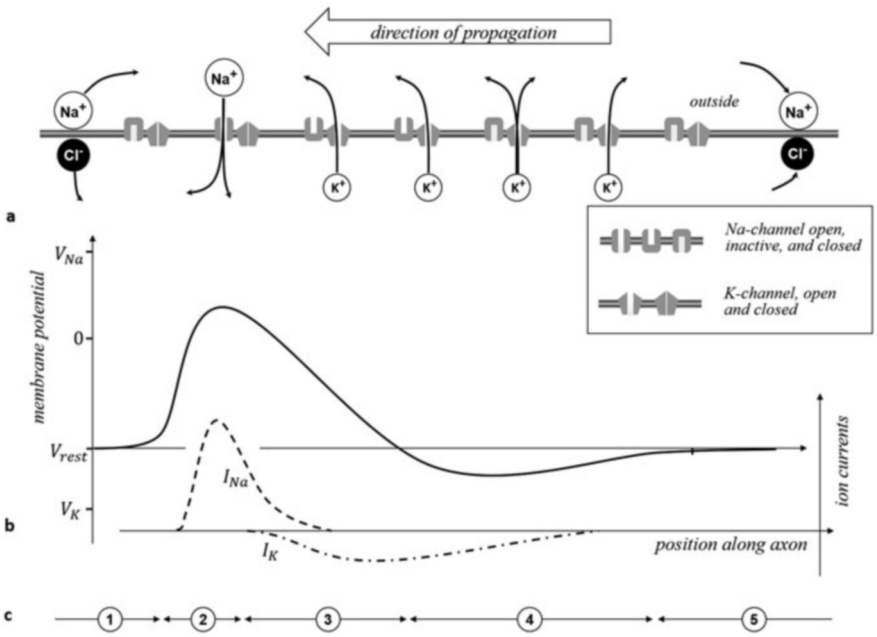


Fig. 1.3 Instantaneous view of an action potential (AP) propagating to the left side of the figure. (**a**) Ion flow and channel states. (**b**) Membrane potential and ion currents. (**c**) Phases of the process corresponding to different zones along the axon for a given instant in time. (1) Passive depolarization. Sodium ions flow from the leading edge of the AP in the forward direction and recharge the membrane capacitor in front of the approaching AP (Na^+ and Cl^- attached to the membrane in part a of the figure). (2) Active depolarization. Sodium channels open and allow more Na^+ ions to flow in. (3) Repolarization. K^+ channels open, while Na^+ channels switch into their "inactive" state. Membrane potential drops to resting level. (4) Hyperpolarization. Membrane potential approaches K^+ equilibrium potential V_K, which causes K^+ channels to close. Resting potential is restored. Na^+ channels switch to their "closed active" state. (5) After the end of the AP, the charging of the membrane capacitor is restored

The action potential is initiated when the membrane is depolarized, i.e., made less negative, for example, by the inflow of positively charged sodium ions at an excitatory synapse or by the spread of sodium ions in front of an approaching action potential in an axon (phase 1 in Fig. 1.3c). This depolarization causes the sodium channels to open and more sodium ions will flow in, driven jointly by the osmotic and electric forces, which in this case act in the same direction. This leads to a still stronger depolarization of the membrane and again to more sodium inflow. Further right in Fig. 1.3c, about 1 mm behind the leading edge, the process has been going on already for about 1 ms.[10] In this region, marked "phase 2" in the figure, the sodium channels inactivate (i.e., close) in a process depending on intrinsic dynamics, not

[10] This corresponds to a propagation speed of 1 m/s, a typical value for nonmyelinated axons.

on membrane potential. This process has been compared to a trap door pushed open from below by the depolarization, but soon afterward falling shut by gravity. Inactive channels cannot open again until the resting potential is restored, i.e., until the ongoing action potential is over. They then switch into their ordinary closed state from which they can again be activated in the next firing event.

The opening of the sodium channels has caused an inflow of positive charges. This is electrically compensated by an outflow of potassium ions, again through a voltage dependent channel, which, however, reacts to the depolarization with a certain delay (phase 3 in Fig. 1.3c). Potassium flow is mostly driven by osmotic force, except during the very peak of the action potential when the inside of the membrane temporarily becomes positive ("overshoot"). Potassium channels stay open until the membrane potential has dropped below the resting level, again with some time delay causing the membrane to polarize even beyond the resting level ("hyperpolarization," phase 4 in Fig. 1.3c). Eventually all channel conductivities are restored and the membrane returns to its resting potential.

When the action potential is over, the ion distributions in the cell and its environment will have slightly changed, due to the involved ion currents. The number of ions passing during each action potential, however, is relatively small; it can be calculated from the membrane capacity, the voltage change, and the elementary charge (see Box 1.7). In a typical axon, less than 0.1% of the ions are exchanged during one single action potential that does not notably affect the Nernst potentials. In the long run of course, these differences have to be compensated by the sodium–potassium pump.

In the hydraulic analogy of Fig. 1.2, membrane capacity can be modeled by adding an elastic bellow to the main volume such that a quick opening or closing of the valves will show in the ascension pipe only with some delay. The action potential would then be initiated by an increase of pressure generated for example by pouring water into the ascension pipe. A float body placed in that pipe could then open the valve of the Na^+ reservoir that will further increase the pressure. The Na^+-valve would need to be constructed in a way that it closes automatically again after a short time. The rising float in the ascension pipe will also open the K^+-valve, albeit with a slower dynamic. This will cause the pressure to sink until the float is lowered to the point where it closes the K^+-valve.

▶ **Key Point** Excitable neural membranes contain voltage dependent ion channels that open and close in dependence of the current membrane potential. The action potential is a dynamic pattern generated by the feedback loop of membrane potential, channel opening and closing, ion flows, and the resulting potential changes.

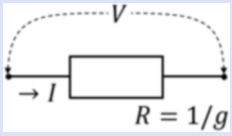
1.2 The Hodgkin–Huxley Theory

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[11] and Huxley[12] (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.

Box 1.2 Basic Rules for Electric Networks

(a) By convention, current flow is from plus to minus. Equal numbers of positive or negative charges moving in opposite directions constitute the same current.

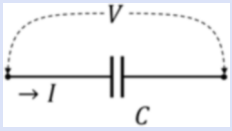
(b) Ohm's law: The current I through a resistor is proportional to the voltage V across the resistor,



$$V = RI.$$

The ratio is called resistance, R, measured in Ω (ohm) = V (volt)/A (ampere). Alternatively, the relation may be expressed by conductivity $g = 1/R$, which is measured in siemens (S), where 1 S equals 1 Ω^{-1}. Ohm's law then reads $I = gV$.

(c) Charge–voltage relation for a capacitor: The voltage across the capacitor is proportional to the amount of charge C stored within the capacitor



$$Q = CV.$$

The ratio C is the capacitance, measured in farad F (farad) = C (coulomb) / V (volt). We will use the temporal derivative of this law, keeping in mind that the change of charge is current:

$$I = \frac{dQ}{dt} = C\frac{dV}{dt}.$$

(continued)

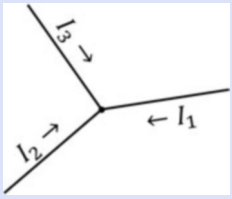
[11] See footnote 7.

[12] Andrew Fielding Huxley (1917–2012). English physiologist. Nobel Prize in physiology or medicine 1963.

Box 1.2 (continued)

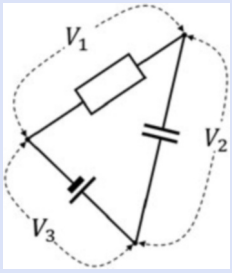
This formulation states that changes of voltage will result in currents flowing in or out of the capacitor.

(d) 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.



$$\sum_i I_i = 0.$$

(e) Kirchhoff's loop rule: The sum of all voltages in a loop adds to zero, if all voltages are measured either in a clockwise or in a counter-clockwise sense.



$$\sum_i V_i = 0.$$

1.2.1 The Total Current Equation

The electrical properties of a small patch of membrane can be modeled by the simple electrical circuit shown in Fig. 1.4. It combines the equivalent circuits for the individual ion equilibria from Fig. 1.1 and adds a third one reflecting the "leakage" of chloride ions across the membrane. The distributions of the sodium, potassium, and chloride ions act as batteries each generating an ion current across the membrane. The conductivity of the membrane for the three ions is denoted by g_{Na}, g_{K}, and g_l, respectively, where the l stands for leakage. The arrows drawn across the resistors indicate that conductivity can change. Voltage dependence: that is, the change of sodium and potassium conductivity as a function of the membrane potential is the basis of the excitability of neural membranes. The leakage channel is not voltage dependent. Finally, the membrane has a capacitance denoted by C_m.

It is interesting to compare the equivalent circuit of Fig. 1.4 with the hydraulic analogy shown in Fig. 1.2. They both illustrate the resting potential, but Fig. 1.4 has a number of additional features. First, it allows for the voltage sensitivity of channel conductance to be explicitly modeled by modifiable resistors. Second, it includes a leakage current that was not shown in Fig. 1.2 but could be easily included. And finally, it shows a capacitor that will become relevant only when the membrane potential changes, i.e., when the resting state is left. In the hydraulic analogy, the

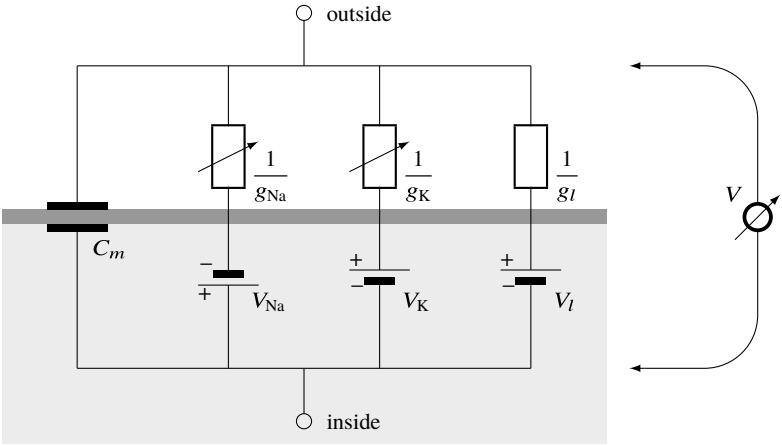


Fig. 1.4 Electrical circuit representing the membrane of a neuron. The right part symbolizes a voltmeter measuring the membrane potential V inside minus outside. Modified from Hodgkin and Huxley (1952)

capacitor corresponds to an additional elastic bellow or simply to the elasticity of the cell membrane that will move out and expand as the pressure increases. A depolarization would correspond to an inflow of water, but this will not lead to an immediate increase in pressure or in the water level of the ascension tube, since part of the incoming water will be stored in the increasing volume of the cell. Likewise, the capacitor in Fig. 1.4 can be thought of as a temporary storage for charges. Capacitive currents are also visible in Fig. 1.3, phases (1) and (5), where the membrane is depolarized or repolarized without any ions actually moving across the membrane, simply by axial movements of Na^+ and Cl^- ions. This causes a reloading of the membrane capacitor modeled as the extra branch in Fig. 1.4.

From the equivalent circuit appearing in Fig. 1.4, we can now derive the main equation of the Hodgkin–Huxley theory. 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. The additivity of the four currents is a consequence of Kirchhoff's node rule (see Box 1.2d): Charge is a conserved quantity, i.e., what goes into a node must also come out. With this consideration, we obtain

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

$$= C_m \frac{dV}{dt} + g_{\mathrm{Na}}(V - V_{\mathrm{Na}}) + g_{\mathrm{K}}(V - V_{\mathrm{K}}) + g_l(V - V_l). \tag{1.2}$$

In the second equation, we have expressed the four currents by their driving forces; they all depend on the membrane potential, which is therefore the variable to

consider. More specifically, the capacitive term I_c depends on the change of the membrane potential, i.e., its derivative dV/dt. The relation $I_c = C\frac{dV}{dt}$ is obtained from the definition of capacitance, $C = Q/V$ (see Box 1.2). This latter equation states that the total charge stored in a capacitor increases linearly with the applied voltage, the ratio being the capacitance C. It is measured in the unit farad that equals coulombs per volt. In a plate capacitor, capacitance depends on the area and thickness of the plates as well as on the width of the gap between them. In the neuron, the gap corresponds to the insulating membrane and maybe myelin sheaths, while the plates are the aqueous media on either side.

For the ion currents, Eq. 1.2 uses Ohm's law (see Box 1.2b). 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) $= 1/\Omega$.[13] V_{Na}, V_{K} and V_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 and no net current will flow in that channel.

For the further development of the Hodgkin–Huxley theory, it is convenient to think of V not as the membrane potential, but as "depolarization," i.e., the deviation of the membrane potential from the resting potential. In Eq. 1.2, this does not change anything, because we simply subtract a constant from all voltages; this does not change the derivative in the capacitive term and cancels out in the resistive terms. Depolarization is zero if the membrane is at rest and becomes positive for excited states. We will switch to this convention from here.

In the sequel, we will discuss the sodium and potassium currents passing through excitable (voltage dependent) channels and present phenomenological models for their dynamics. These phenomenological models will then be combined using Eq. 1.2 to generate a prediction for the time course of the action potential.

1.2.2 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 conductance for potassium is 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.

[13] In the older literature, the unit "mho" (ohm spelled backward) is also used; 1mho = 1S.

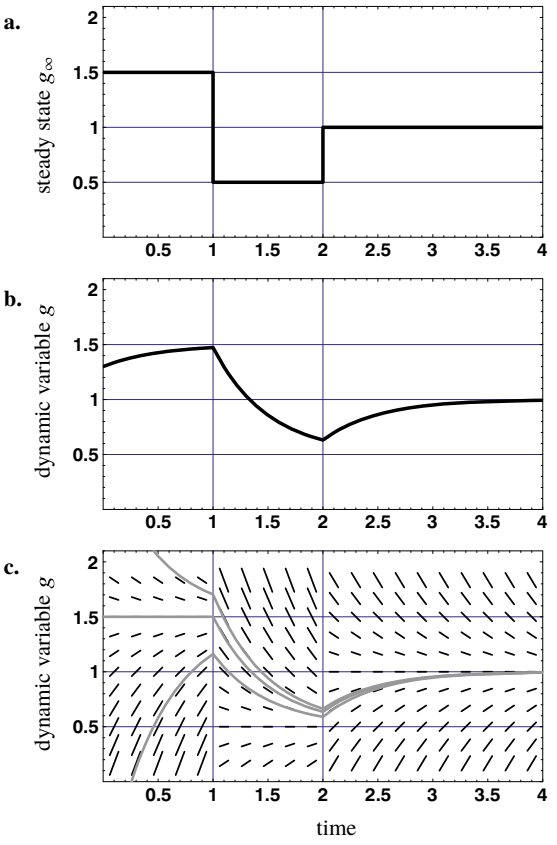


Fig. 1.5 Behavior of a simple differential equation. (**a**) Time course of an outer "driving force," expressed as the steady state eventually obtained by the system. It is switched at discrete times. (**b**) Exponential relaxation to the respective steady states as expressed in Eq. 1.3. The value g 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 1.4. 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 (Fig. 1.5). 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 schematically in Fig. 1.5a. 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 nonzero switching times (finite switching speed) appears in Fig. 1.5b. The curve shows an exponential relaxation; between two stepping times, it might be formally described by

$$g(t) = g_\infty + (g_o - g_\infty) \exp\left\{-\frac{t - t_0}{\tau}\right\}. \tag{1.3}$$

Here, g_o is the initial value of g at time t_o, g_∞ is the steady state of g for the voltage applied during the current interval, and τ is a time constant that is large if the

approach toward the steady state is slow. 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. This can be expressed by saying that τ is the time during which the distance from the steady state is reduced by the fixed proportion $1/e$ or 36.8% of its initial value.

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 temporal derivative of $g(t)$, should be proportional to the current deviation from the steady state given by $g(t) - g_\infty$:

$$g'(t) = -k(g(t) - g_\infty) \quad \text{for some constant } k. \tag{1.4}$$

Here we keep writing g_∞ for the steady state, but keep in mind that it depends on V. Whatever changes $V(t)$—and with it $g_\infty(V(t))$—will undergo over time, Eq. 1.4 remains valid. An illustration of Eq. 1.4 is given in Fig. 1.5c. Each short line marks a local slope $g'(t)$ calculated from Eq. 1.4. Each curve drawn in a way that the local lines in Fig. 1.5c are tangents to the curve is a solution of Eq. 1.4.

Equation 1.4 is completely analogous to Hooke's law of spring motion in elementary mechanics. In this analogy, g is the spring extension and k is the stiffness of the spring. g_∞ is extension at rest that may be assumed zero but may also be altered by attaching a fixed weight to the spring.

The equation is also an example of a *differential equation*, relating the value of a function to its derivative. The solutions of differential equations are functions, not numbers. By taking the derivative of g in Eq. 1.3 and comparing it to g itself, it is easy to show that this function solves Eq. 1.4 if k is set to $1/\tau$. We start by taking the derivative of Eq. 1.3

$$g'(t) = -\frac{1}{\tau}(g_o - g_\infty) \exp\left\{-\frac{t - t_o}{\tau}\right\}. \tag{1.5}$$

Inserting g' and g (again from Eq. 1.3) into Eq. 1.4, we obtain

$$-\frac{1}{\tau}(g_o - g_\infty) \exp\left\{-\frac{t - t_o}{\tau}\right\} \stackrel{!}{=} -k\left(g_\infty + (g_o - g_\infty) \exp\left\{-\frac{t - t_o}{\tau}\right\} - g_\infty\right),$$

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

Equation 1.3 is called an analytical solution of the differential equation 1.4 because it is formulated as a named mathematical function, in this case the exponential. The sketched procedure, that is, 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.

Box 1.3 Ordinary Differential Equations (ODE)
Differential equations relate a function with its first and maybe higher order derivatives. If the function depends on one variable only, the derivatives and the equation are called "ordinary," in contrast to "partial" derivatives and equations occurring in functions of several variables. Equation 1.4 is a first order ODE because it involves only one unknown function (g) and its first order derivative (g').

Solutions of ODEs are functions of the variable with respect to which the derivative is taken. If this is time, the solution describes the evolution of the system governed by the ODE, when released from some starting point. When this starting point or "initial value" is specified, the solution is uniquely determined. In our case, the initial value at time t_o is just a single number $g(t_o)$, but it may be a longer vector $(g(t_o), g'(t_o), g''(t_o), \ldots)$ in higher order ODEs. Finding a solution to an ODE for a given initial value is also called an "initial value problem."

1.2.3 The Potassium Current

The conductances g_{K} and g_{Na} used in Eq. 1.2 are the total effect of the opening and closing of many individual channels. While each channel is either closed or open, the total membrane conductivity is a continuous quantity reflecting the relative number of open and closed channels at any one time. In the classical Hodgkin–Huxley theory, individual channels are not modeled but only the total conductivity of the membrane measured in milli-siemens per square centimeter (mScm^{-2}).

Individual ion currents can be studied in the so-called voltage clamp preparation. The classical experiments were carried out in the giant axon of the squid *Loligo (Dorytheutis) pealeii*. This axon measures about 1 mm in diameter and is more than 15 cm long. It is removed from the animal and placed in seawater or in media with controlled concentrations of sodium and potassium ions. Electrodes are inserted into the axon for measuring the membrane potential and for compensating the observed potential changes by injecting appropriate currents. The voltage clamp thus cuts open the feedback loop formed by (i) membrane current, (ii) the de- or hyperpolarization caused by such currents, (iii) the subsequent gating of voltage dependent ion channels, and (iv) the ion currents resulting from the new potential and conductances. The membrane potential is clamped to a fixed depolarization, and conductances can then be observed under stable conditions. For technical details, see, for example, Aidley (1998).

Figure 1.6 shows the result of a voltage clamp experiment in which the membrane potential is stepped from resting potential to some depolarization at time $t = 0\,\mathrm{ms}$. Conductance rises to a plateau whose level depends on the size of the depolarization V; it is denoted as $g_{\mathrm{K},\infty}(V)$. Conductance change is initially slow and rises most quickly about one millisecond after the onset of depolarization, i.e., the curve is

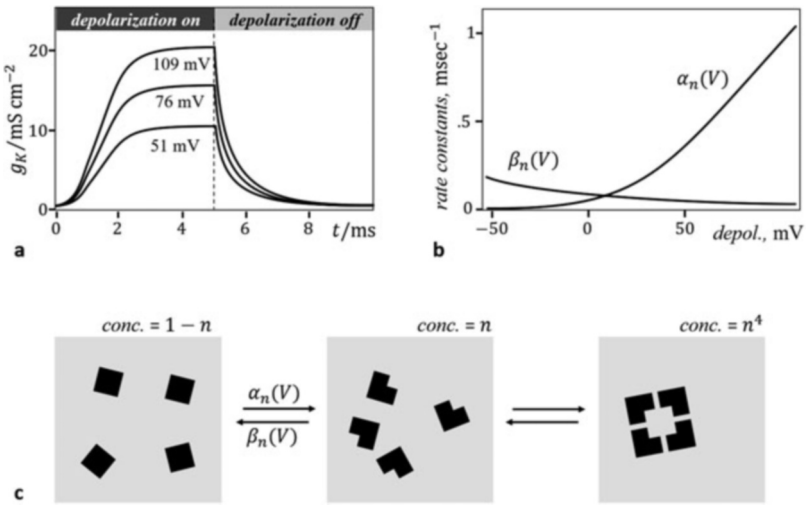


Fig. 1.6 Potassium conductance (g_K) in the voltage clamp experiment. (**a**) Measurement of g_K in milli-siemens per square centimeter when membrane potential is stepped from resting potential by 51, 76, and 109 mV (depolarization) at time $t = 0$ ms and stepped back to resting potential at time $t = 5$ ms. (**b**) Estimates of the rate constants α_n and β_n obtained by fitting the channel model of Eqs. 1.6 and 1.7 to experimental data. (**c**) Kinetics modeled by these equations. Left: pool of subunits in unfavorable state. Middle: subunits in favorable state, portion of subunits in favorable state is n. Right: Channel formed of four subunits. Probability of four subunits joining in one place is n^4. Data for panels a, b from Hodgkin and Huxley (1952)

sigmoidal with an inflection at that point. This is different from the behavior of the simple relaxation system shown in Fig. 1.5b where relaxation speed is highest at the beginning and gradually declines. The inflection point will be important for the modeling of the channel switching behavior. At time $t = 5$ 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 nonlinear.

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

$$g_K(t) = \bar{g}_K \, n(t)^4 \tag{1.6}$$

$$\frac{dn}{dt}(t) = \alpha_n \, (1 - n(t)) - \beta_n n(t). \tag{1.7}$$

Here $\bar{g}_K$ is a constant (about 22 mScm^{-2}) depending neither on time nor on voltage. It represents the maximum conductivity if all potassium channels are open, i.e., the concentration of channel proteins available, and can be estimated from the

maximum K^+ conductivity occurring at large depolarizations. The term $n(t)^4$ in Eq. 1.6 indicates that four channel subunits have to meet in order to form a channel where $n(t)$ is the proportion of subunits available for this process. 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 1.7 assumes that the channels can be in one of two states, one favoring the formation of a channel and one not favoring it. In Fig. 1.6c, the transition from the unfavorable to the favorable state is symbolized by the changing shape of the subunits when going from the left to the middle pool. 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" that depend on voltage, but not on time. They take values between 0 and 1 msec^{-1} and specify the likelihood (the rate) at which the channel subunits switch 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 Fig. 1.6c.

The exponent four in Eq. 1.6 has been chosen to reproduce a subtle feature of the conductivity step response shown in Fig. 1.6a. 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 and Huxley already in 1952.

In order to understand how the rate constants can be estimated from the conductances measured in the voltage clamp, we need to observe that Eq. 1.7 has the same structure as Eq. 1.4 studied above, i.e., $n' = -k(n - n_\infty)$. To see this, we write Eq. 1.7 as

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

from which we read $k = \alpha_n + \beta_n$ and $n_\infty = \alpha_n/(\alpha_n + \beta_n)$. Thus, if k and n_∞ are estimated from the voltage clamp data, we can calculate the rate constants as $\alpha_n = kn_\infty$ and $\beta_n = k(1 - n_\infty)$. Starting from voltage clamp measurements shown in Fig. 1.6a, and many similar curves measured for other depolarizations V, we would thus need to apply the following steps: First, estimate $\bar{g}_K$ and divide the measurements by $\bar{g}_K$ to yield an estimate of $n(t)^4$. The fourth root then gives an estimate of $n(t) = (g_K(t)/\bar{g}_K)^{1/4}$. From these curves, the steady state value n_∞ and the time constant $\tau = 1/k$ are read. Finally, the rate constants $\alpha_n(V)$ and $\beta_n(V)$ are calculated as shown above; the result is plotted in Fig. 1.6b.

Box 1.4 Channel Kinetics and the Law of Mass Action

The Hodgkin–Huxley model of the sodium and potassium channels is basically a chemical reaction kinetic as described by the law of mass action. The idea is that whenever all particles required for a given reaction meet within a small space, the reaction will take place with a fixed probability. The overall speed or rate of the reaction is therefore proportional to the concentrations of the reactants and a factor called the rate constant. The rate constant does not depend on the concentration of the reactants but changes with temperature and, in our case, with membrane potential.

For the K^+ channel, the first step of the reaction involves only one reactant; it can be written as $U \longrightarrow F$, where U and F denote the channel subunits in the unfavorable and favorable states, respectively. The concentrations of the subunits, expressed for example in moles per square centimeter of membrane area, are written as [U], [F]; we observe that $[U]+[F] = const$. Instead of the absolute concentrations, Hodgkin–Huxley theory uses relative concentrations, for the K^+ channel expressed as the auxiliary variables $n = [F]/([U] + [F])$ and $1 - n = [U]/([U] + [F])$.

The law of mass action now states that the proportion of particles undergoing the reaction $U \longrightarrow F$ within one millisecond is $\alpha_n(1-n)$, where α_n is the rate constant from Eq. 1.7. The backward reaction $F \longleftarrow U$ is simultaneously taking place with a rate proportional to the concentration of U, i.e., its speed is $\beta_n n$. The total reaction rate of the two-way reaction $U \rightleftharpoons F$ is the difference of these two directions as stated in Eq. 1.7.

The second step of the channel model assumes that four subunits form a channel, $4F \longrightarrow C$, where C denotes the open channel. According to the law of mass action, the rate of this step should be proportional to the fourth power of [F] and therefore to n^4. In the Hodgkin–Huxley model, the rate constant of this step is assumed to be very large, i.e., the reaction happens instantaneously as soon as four subunits meet on the same spot. It is therefore not modeled as a differential equation but simply by taking the fourth power in Eq. 1.6.

1.2.4 The Sodium Current

The response of the sodium conductance to a step change in membrane potential is shown in Fig. 1.7a. It differs from the potassium response in its quicker rise and in its overall phasic behavior, i.e., in the fact that conductance goes back to zero a few milliseconds after the onset of depolarization, even if depolarization lasts.

The phenomenological model presented for this dynamics by Hodgkin and Huxley (1952) uses two types of channel subunits (or "gating particles") that must combine in the stoichiometric ratio 3:1 to form an open channel (Fig. 1.7c). At resting potential, particle type 1 is in its unfavorable state but switches to its

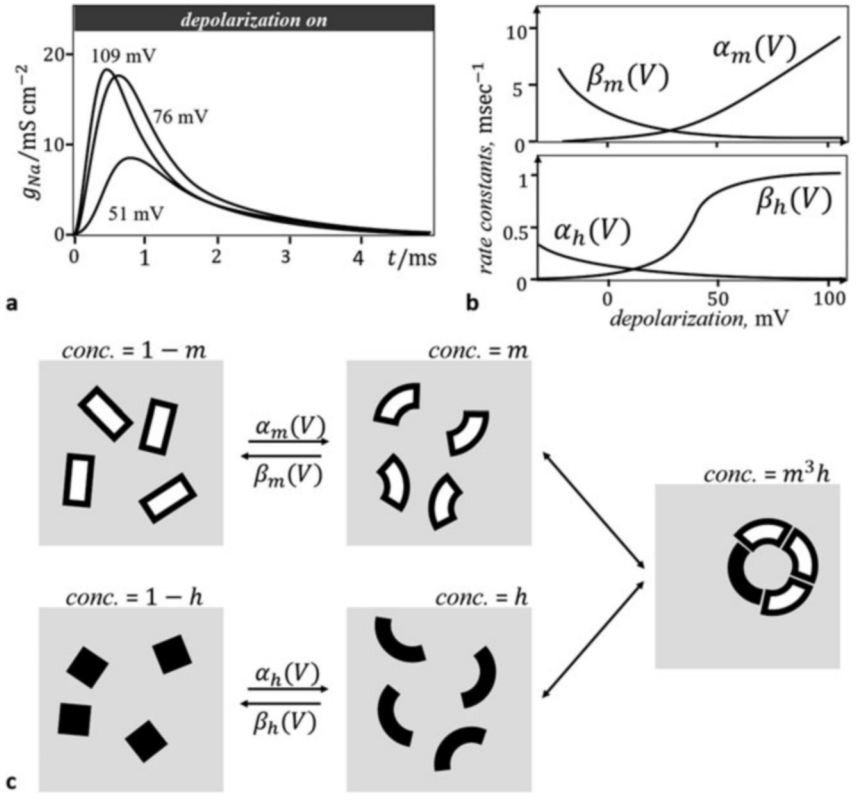


Fig. 1.7 Sodium conductance (g_{Na}) in the voltage clamp experiment. (**a**) Measurement of g_{Na} when membrane potential is stepped from resting potential by 51, 76, and 109 mV (depolarization) at time $t = 0$ ms. Conductance relaxes to zero, while the axon is still depolarized. (**b**) Estimates of the rate constants α_m, β_m, α_h, and β_h obtained by fitting the channel model of Eqs. 1.10 and 1.11 to the data. (**c**) Kinetics modeled by the Hodgkin–Huxley equations with two types of "particles" (channel subunits). Upon depolarization, particles of type 1 (open symbols) change fast from unfavorable to favorable state, while particles of type 2 (filled symbols) change slowly from favorable to unfavorable state (note reversed voltage dependence of rate constants in part b of the figure). An open channel forms when three particles of type 1 and one of type 2 meet in favorable state. This final step is not modeled dynamically but assumed to happen instantaneously. Data for parts a and b from Hodgkin and Huxley (1952)

favorable state as soon as depolarization starts. The proportion of type 1 particles in favorable state is described by the auxiliary variable $m(t)$; at the onset of depolarization, it rises toward a steady state m_∞ following the relaxation model discussed above.

The second particle type shows the reverse behavior: It is initially in its favorable state for opening a channel. Upon depolarization of the membrane, however, it switches into its unfavorable state. Therefore, channels will open immediately after the depolarization step, when favorable type 1 particles become available but

will be inactivated soon afterward when type 2 particles start switching into their unfavorable state. The proportion of type 2 particles in favorable state is modeled by the auxiliary variable $h(t)$ that is initially high but decreases during depolarization. The peaked curves of Fig. 1.7a are thus modeled as a product of two relaxation curves, a quickly rising third-order curve $m^3(t)$ and a slowly decaying first order curve $h(t)$. The opposite effect of depolarization on the two types of gating particles, switching from unfavorable to favorable for type 1 and switching from favorable to unfavorable for type 2, is reflected by the reversed shape of their α and β rate constants as depicted in Fig. 1.7b. Note also the different scales for the rate constants of the m and h processes in Fig. 1.7b. In $m(t)$, the steady state is reached much faster (the time constant is much shorter) than for the h process.

The equations read

$$g_{\mathrm{Na}}(t) = \bar{g}_{\mathrm{Na}}\, m(t)^3\, h(t) \tag{1.9}$$

$$\frac{dm}{dt}(t) = \alpha_m\,(1 - m(t)) - \beta_m m(t) \tag{1.10}$$

$$\frac{dh}{dt}(t) = \alpha_h\,(1 - h(t)) - \beta_h h(t). \tag{1.11}$$

As before, the rate constants α_m, β_m, α_h, and β_h depend on depolarization, but not on time, see Fig. 1.7b. They can be obtained by fitting the voltage clamp measurements, as explained above for the potassium channel. $\bar{g}_{\mathrm{Na}}$ is a constant specifying the maximum possible sodium conductance if all channels would be simultaneously open; it is estimated as $70.7\,\mathrm{mScm}^{-2}$. Note that the maximal obtained g_{Na} in Fig. 1.7a is a little less than $20\,\mathrm{mScm}^{-2}$, which means that even at peak conductance, only a small fraction of the sodium channels is open.

The Hodgkin–Huxley model is a mean-field theory that models the overall current flow across macroscopic patches of membrane such as the squid giant axon, not the behavior of individual channels. The variables $g_{\mathrm{K}}(t)$ and $g_{\mathrm{Na}}(t)$ can be interpreted as the probabilities of channel opening at a given instant in time. The variables n (for the potassium channel) and m, h (for the sodium channel) describe the probabilities of the channel subunits being in their favorable state for forming the channel. From these, the probabilities of the channels themselves being in the open state are derived as n^4 and m^3h, respectively. Other states of the channels, for example, the closed inactivated and closed activatable states of the sodium channel, correspond to multiple combinations of the subunit types and are not explicitly modeled in Hodgkin–Huxley theory. All possible substrate combinations and their transitions are taken into account in the so-called Markov models of channel kinetics, see, for example, Rudy and Silva (2006) and Fink and Noble (2009).

Box 1.5 Four Types of "Clamp" Experiments

Membrane biophysics uses elaborate electrophysiological technologies, several of which have been associated with the term "clamp." The most important paradigms are listed below.

The **voltage clamp** is a circuit for keeping the membrane potential constant even in the presence of membrane currents. This is achieved by injecting compensatory currents and thereby cutting open the loop between membrane current and potential. In addition, individual channel types are blocked by drugs such as tetrodotoxin (TTX) for the Na^+ channel or tetraethylammonium (TEA) for the K^+ channel. If one channel type is blocked, the currents passed by the others can be studied in isolation. Voltage clamp measurements are the basis of the Hodgkin–Huxley theory as discussed in this chapter.

The **space clamp** is basically a copper wire inserted into a giant axon to equalize the membrane potential along the axial direction. As a result, no ion currents can flow along the axon. It is used to study action potentials without propagation. If the axon spikes, all membrane patches will go through the same phases of the action potential in synchrony.

The **patch clamp** is a method for measuring currents passing through individual channels (Sakmann and Neher 1984). It consists of a glass pipette placed on the membrane, a wire electrode inside the pipette, and a voltage clamp to control the overall membrane potential. By slightly lowering the pressure in the electrode, a small patch of membrane is attached to or gently sucked into the pipette, and the interior of the glass electrode is "sealed" against the intercellular medium. This allows to measure currents passing through the (small) number of channels in the attached patch. Patch clamp recordings show that individual channels switch open or close without intermediate states and allow currents in the order of a few picoamperes (10^{-12} amperes).

The **dynamic clamp** extends the idea of the voltage clamp to replace one or several types of channel with a simulation of their behavior (Sharp et al. 1993). One channel type is blocked, and the current expected to pass through it is provided by an electronic circuit. If the expectation is correct, the cell should behave just as it did without the channel blockage.

1.2.5 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 that in turn will change membrane potential. As illustrated in Fig. 1.8, the whole process is a feedback loop of membrane potential, membrane conductivity, membrane current,

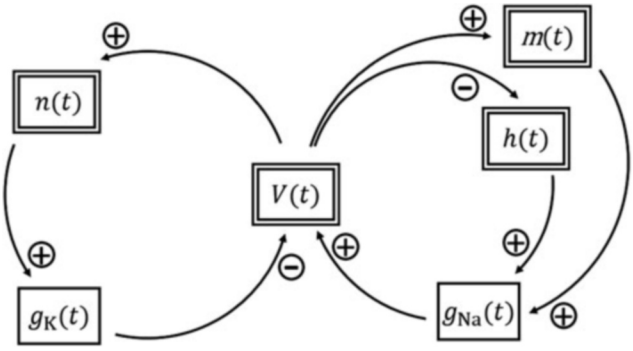


Fig. 1.8 Conductances in closed loop. The double boxed quantities are the state variables of the Hodgkin–Huxley system: V, membrane potential; n, m, and h, relative concentrations of the "gating particles" as used in the channel models. Left loop: depolarization (increase of V) leads to increased potassium conductance g_{K} via Eqs. 1.6 and 1.7. This results in an outflow of potassium ions that decreases the membrane potential. The left loop is therefore a negative feedback loop tending to stabilize the system at resting potential. Right loop: depolarization leads to a quick increase in sodium conductance g_{Na} (Eq. 1.10) and inflow of sodium ions. This increases depolarization further and forms a positive feedback loop. In the long run, however, it is stopped by the inactivation process (Eq. 1.11) such that the negative feedback via the potassium loop drives the system back to the resting state

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 section of the axon but may still vary over time. This can be achieved by inserting a highly conducting copper wire into an axon that will equilibrate all differences in membrane potential occurring along the axon. The whole section will thus exhibit the action potential in synchrony.

In the space-clamp situation, no ion currents 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. That is, ion currents crossing the membrane will be completely counterbalanced by capacitive currents loading and reloading the capacitor. Still, the axon is able to spike. In Eq. 1.2, we can assume $I = 0$ in and obtain the following set of four coupled ordinary differential equations:

$$-C_m \frac{dV}{dt}(t) = \bar{g}_{\mathrm{K}}\, n(t)^4\, (V(t) - V_{\mathrm{K}}) + \bar{g}_{\mathrm{Na}}\, m(t)^3\, h(t)\, (V(t) - V_{\mathrm{Na}}) + \bar{g}_l\, (V(t) - V_l) \tag{1.12}$$

$$\frac{dn}{dt}(t) = \alpha_n(V(t))\, (1 - n(t)) - \beta_n(V(t))\, n(t) \tag{1.13}$$

$$\frac{dm}{dt}(t) = \alpha_m(V(t))\,(1 - m(t)) - \beta_m(V(t))\,m(t) \tag{1.14}$$

$$\frac{dh}{dt}(t) = \alpha_h(V(t))\,(1 - h(t)) - \beta_h(V(t))\,h(t). \tag{1.15}$$

The four state variables of this system are the membrane potential V as well as the three auxiliary variables n, m, and h. These enter the equations in nonlinear ways, first of all by the products and powers n^4 and m^3h, but also via the rate constants that are nonlinear functions of V as shown in Figs. 1.6b and 1.7b.

The coupled system of nonlinear differential equations 1.12–1.15 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 (depolarization is 0). Therefore, we obtain from Eq. 1.13: $n_o = \alpha_n(0)/(\alpha_n(0) + \beta_n(0))$. 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 are 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. The basic scheme sketched out here is known as the forward Euler algorithm with constant step length. Advanced mathematics programming tools provide elaborate routines for solving ODEs based on better extrapolation schemes, error estimation, and variable step lengths. A simple piece of code for solving the Hodgkin–Huxley system in MATLAB appears in Box 1.6.

Box 1.6 MATLAB Code for the Hodgkin–Huxley System in Space Clamp

Declare global variables

```
global vK  vNa  vL  gmaxK  gmaxNa  gL  cMemb
global alpha_n beta_n alpha_m beta_m alpha_h beta_h
```

Set constants according to Hodgkin and Huxley (1952). Unlike the conventions used in the original paper, depolarizations are treated with positive sign.

```
gK  = -12     % potassium equilibrium potential in mV
vNa = +115    % sodium equilibrium potential in mV
vL  =  10.613 % equilibrium potential for leakage. Calibrates Vrest
                        to 0.
gmaxK  = 36   % maximum possible potassium conductance, in mS/cm^2
gmaxNa = 120  % maximum possible sodium conductance, in mS/cm^2
gL =     0.3  % conductance of leakage channels, in mS/cm^2
cMemb  = 1.0  % membrane capacitance, in muF/cm^2
```

(continued)

Box 1.6 (continued)
Define rate constants as MATLAB *anonymous functions (eqs. 12, 13, 20, 21, 23, 24 of Hodgkin and Huxley 1952), again with positive sign for depolarization. Unit is msec*$^{-1}$

```
alpha_n = @(v) 0.01 * (10-v)/(exp((10-v)/10) - 1);
beta_n  = @(v) 0.125 * exp(-v/80);
alpha_m = @(v) 0.1  * (25-v)/(exp((25-v)/10) - 1);
beta_m  = @(v) 4 * exp(-v/20);
alpha_h = @(v) 0.07 * exp(-v/20);
beta_h  = @(v) 1/(exp((30-v)/10) + 1);
```

Set initial conditions and coerce into column vector

```
v0 = +15;                                  % depolarizing stimulus
n0 = alpha_n(0)/(alpha_n(0) + beta_n(0)); % steady states
m0 = alpha_m(0)/(alpha_m(0) + beta_m(0)); % at resting state
h0 = alpha_h(0)/(alpha_h(0) + beta_h(0));
y0 = [v0; n0; m0; h0];
```

Run ODE solver on time interval from 0 to 6 milliseconds and plot voltage

```
[t, y] = ode45(@HoHu,[0 6],y0);
plot(t,y(:,1));
```

Define function HoHu returning the derivatives of V, n, m, and h according to Eqs. 1.12–1.15

```
function dy = HoHu(t,y)
global vK  vNa  vL  gmaxK  gmaxNa  gL cMemb
global alpha_n beta_n alpha_m beta_m alpha_h beta_h

dv = -1/cMemb * (gmaxK * (y(2))^4 * (y(1) - vK) + \ldots
     gmaxNa * (y(3))^3 * y(4) *(y(1) - vNa) + \ldots
     gL * (y(1) - vL));
dn = alpha_n(y(1)) * (1-y(2)) - beta_n(y(1))*y(2);
dm = alpha_m(y(1)) * (1-y(3)) - beta_m(y(1))*y(3);
dh = alpha_h(y(1)) * (1-y(4)) - beta_h(y(1))*y(4);

dy = [dv; dn; dm; dh];     % return derivatives as column vector
end
```

Figure 1.9 shows a numerical solution of the system 1.12–1.15 for external stimuli $V(0)$ of 6, 7, and 30 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 for the 7 mV case in the lower part of Fig. 1.9.

If different external stimuli $V(0)$ are chosen, two different types of behavior are observed (Fig. 1.9). If $V(0)$ is 6 mV or less, no action potential is generated. Rather, the membrane potential quickly returns to the resting state. In this case, the stimulus is said to be 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 action potentials, not in bigger ones. The Hodgkin–Huxley equations thus capture the all-or-nothing behavior of neural excitation. In dynamical systems theory, this is called a "bifurcation" of the system, indicating that the system changes

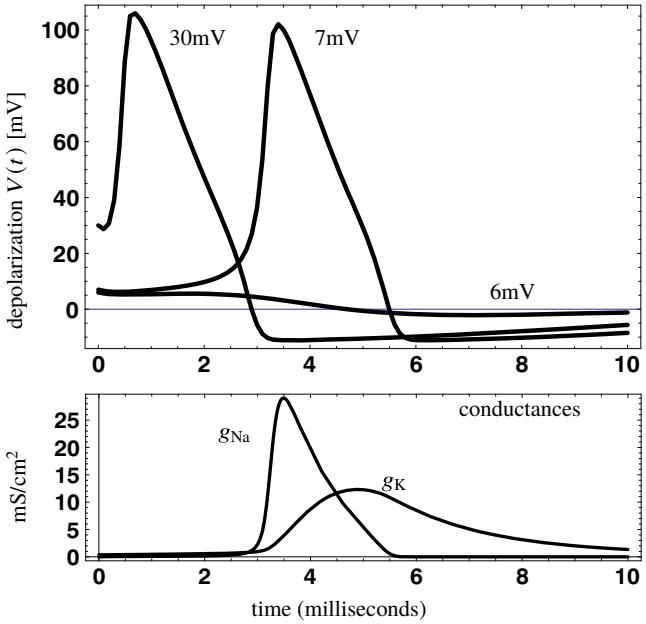


Fig. 1.9 Solutions of the space-clamp equations for external stimuli V_0 occurring at time $t = 0$. Top: Three solutions for the depolarization resulting from external stimuli $V_0 = 6\,\text{mV}$, 7 mV, and 30 mV above resting potential. The threshold for spike generation is between 6 and 7 mV. Spikes 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

its qualitative behavior (spiking or not) upon the variation of a continuous parameter that in this case is the stimulus strength. The bifurcation point or threshold is an emergent property of the system, not another free parameter (see also Sect. 1.3.2).

Box 1.7 Total Number of Ions Moving

How many ions pass the membrane during an action potential? For a rough estimate, we consider a cylindrical piece of axon with diameter $2r = 10\,\mu\text{m}$ and length $l = 1\,\text{mm}$. Its volume v and surface area a can be calculated as

$$v = \pi r^2 l \approx 7.9 \times 10^{-11}\,\text{L}$$
$$a = 2\pi r l \approx 3.1 \times 10^{-4}\,\text{cm}^2.$$

With an intracellular sodium concentration of some $10.4\,\text{mmol}\,L^{-1}$ (Table 1.1), we find that the cylinder will contain

$$n = v \times 10.4 \times 10^{-3}\,\text{mol}\,\text{L}^{-1} \times N_A = 4.9 \times 10^{11}$$

(continued)

Box 1.7 (continued)
Na^+ ions, where $N_A \approx 6 \times 10^{23}\,\mathrm{mol}^{-1}$ is Avogadro's number, i.e., the number of molecules in a mol.

During the action potential, the membrane potential will change by roughly 110 mV from which we calculate the required charge transport according to the equation $Q = CV$ (see Box 1.2c). Assuming a specific membrane capacitance of $C_m = 1\,\mu\mathrm{F/cm}^2$, the total capacitance of our membrane cylinder is

$$C_T = aC_m = 3.1 \times 10^{-4}\,\mathrm{cm}^2 \times 10^{-6}\,\mathrm{F/cm}^2 = 3.1 \times 10^{-10}\,\mathrm{F}.$$

The total charge to be transported is therefore

$$Q = C_T V = 3.1 \times 10^{-10}\,\mathrm{F} \times 1.1 \times 10^{-1}\,\mathrm{V} = 3.4 \times 10^{-11}\,\mathrm{C}.$$

With the elementary charge (the charge of a single Na^+ ion) being 1.6×10^{-19} C, we see that this amounts to a number

$$\Delta n = 3.4 \times 10^{-11}\,\mathrm{C}/(1.6 \times 10^{-19}\,\mathrm{C}) = 2.1 \times 10^8$$

of particles that pass the membrane during an action potential. The change of the Na^+ concentration after an action potential is therefore in the order of

$$\frac{\Delta n}{n} = \frac{2.1 \times 10^8}{4.9 \times 10^{11}} = 4.2 \times 10^{-4} = 0.042\,\%.$$

The change of ion concentrations in one action potential is therefore quite low. Indeed, a neuron can generate tens or even hundreds of action potentials before the ion gradients break down, even if the sodium–potassium pump would not operate.

The Hodgkin–Huxley equations also model the refractory period during which no other action potential can be elicited. In the molecular biology of the sodium channel, the refractory period results from the so-called ball-and-chain mechanism of inactivation that is released only after the channel goes through a phase of hyperpolarization. In the model, the refractory period is a result of the dynamics of the variable h in the model of the sodium channel (Eq. 1.11), which must be large to allow for an action potential to be elicited. As can be seen from the according rate constants α_h and β_h plotted in Fig. 1.7b, h can approach larger values only if depolarization is small or even negative: that is, during the hyperpolarization phase of the action potential.

The sodium and potassium channels discussed in the classical Hodgkin–Huxley theory are but two examples of a large variety of voltage-gated ion channels found

in the nervous system, the heart, or other muscular tissue of various animal species (Bean 2007). These include potassium and sodium channels with other dynamical properties, but also calcium (Ca^{2+}) channels not considered in the classical theory. Detailed time courses of the action potential may indeed vary between different cells, depending on the type and frequency of channels expressed. Hodgkin–Huxley type models based on the general formula $g = \bar{g}m^p h^q$ for small integers p, q can be found for many of these channels and do generally well in simulating the cell's spiking behavior.

One deviation from Hodgkin–Huxley theory results from cooperative gating behavior of channel clusters (Naundorf et al. 2006; Dixon et al. 2022). As pointed out above, the rate Eqs. 1.7, 1.10, and 1.11 are based on the idea that channels open and close independent of each other, with the only interaction resulting from the voltage changes effected by the channel currents. In some cases, however, channels have been shown to form clusters in the cell membrane within which mechanical interaction is also possible. In a study on simple and complex neurons from the cat's visual cortex, Naundorf et al. (2006) showed that the rise of the action potential by Na^+ inflow is even steeper than expected from the Hodgkin–Huxley model. This may indeed be due to cooperative gating in clusters of sodium channels.

▶ **Key Point** The Hodgkin–Huxley equations describe the stimulus dependence and the time course of the action potential as a result of the dynamics of voltage dependent channels, the currents passing through these channels, and the charging and recharging of the membrane.

1.3 Approximations

The Hodgkin–Huxley equations are most useful for the study of the mechanisms of the action potential and the effects of voltage-gated channels in single neurons. For questions involving networks of spiking neurons or the membrane behavior over longer periods of time, approximations have been developed, which are more easily handled in the respective contexts, for overviews see Gerstner and Kistler (2002) and Izhikevich (2004). We will discuss two examples.

1.3.1 Integrate-and-Fire

Consider a neuron receiving a constant depolarizing current delivered either artificially by an intracellular electrode or as an inflow of Na^+ ions through ligand-gated channels at a postsynaptic membrane. The current will depolarize the membrane, initially by reloading the membrane capacitor. As soon as the membrane potential exceeds the firing threshold, the action potential will start and proceed with its standard "all-or-nothing" time course. Afterward the membrane potential will be restored to the resting level, and the depolarization will again start from there. For many applications, the action potential itself can therefore be treated as a fixed and

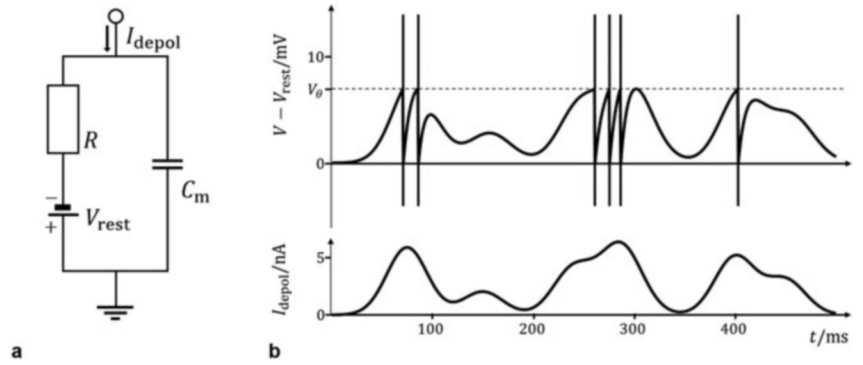


Fig. 1.10 Integrate-and-fire model. (**a**) Simplified Hodgkin–Huxley circuit with all conductances lumped into one constant resistance. Depolarizing current will charge the membrane capacitor until its voltage reaches a steady state V_∞. (**b**) Example with a fluctuating input current I_{depol} (lower panel, in nanoampere, nA, or 10^{-9} ampere) and resulting membrane potential (continuous line in upper panel). If the firing threshold V_Θ is reached, a spike is elicited and the V is reset to the resting state. The vertical bars mark these spiking events

unchangeable event starting as soon as the firing threshold is exceeded and ending in the resting state of the cell. The only remaining component that requires dynamic modeling is the time course of the depolarization process, i.e., the initial reloading of the membrane capacitor.

This idea is known as the "integrate-and-fire" model of neural activity (Abbott 1999) where the word "integration" refers to the accumulation of charge in the capacitor. As in the Hodgkin–Huxley theory, we start by distinguishing the capacitive and resistive components of the membrane current as $I_{\text{depol}} = I_C + I_R$ (see Fig. 1.10a) and express the partial currents in terms of their voltage dependencies,

$$I_{\text{depol}} = C_m \frac{dV}{dt} + \frac{1}{R}(V - V_{\text{rest}}). \tag{1.16}$$

The total membrane resistance R lumps together all channel conductances at resting state, i.e., $R = 1/(g_{\text{K}}(0) + g_{\text{Na}}(0) + g_l)$. With the values listed in Box 1.6, we have $R \approx 1.48\,\text{k}\Omega\text{cm}^2$. Equation 1.16 is a first order ODE for voltage. If we assume a constant depolarizing current I_{depol}, it has the same structure as the relaxation Eq. 1.4. This is a coincidence reflecting the fact that both equations describe relaxation processes for which Eq. 1.4 is the simplest possible form, be it in mechanics, electricity, or reaction kinetics. Solutions are exponential approaches to a steady state V_∞ with time constant τ as explained in Sect. 1.2.2. By comparison with Eq. 1.4 and observing $k = 1/\tau$, we immediately obtain the steady state and the time constant as

$$V_\infty = V_{\text{rest}} + RI_{\text{depol}} \tag{1.17}$$

$$\tau = RC_m. \tag{1.18}$$

Note that the dimension of τ is $\mathrm{k\Omega\,cm^2} \times \mu\mathrm{F\,cm^{-2}} = 10^{-3} \times (\mathrm{V/A}) \times (\mathrm{C/V}) =$ ms, i.e., time, as required. With the values used also for the Hodgkin–Huxley simulation, it evaluates to about 3 ms.

Figure 1.10b shows the situation with a variable input current appearing in the lower panel. The membrane potential (upper panel) follows this trace with a small delay. Whenever it reaches the firing threshold V_Θ, the potential is reset to the resting state and starts again from there. In the figure, these events are marked by the vertical bars that symbolize action potentials.

It is interesting to consider the so-called transfer function of the integrate-and-fire unit, i.e., the firing rate resulting from a constant depolarizing current of a given strength. In the exponential relaxation, the spiking threshold V_Θ is reached if

$$V = RI_{\mathrm{depol}}\left(1 - \exp\left\{-\frac{t}{\tau}\right\}\right) = V_\Theta. \tag{1.19}$$

Here we have set $V_{\mathrm{rest}} = 0$, i.e., we returned to the depolarization notation.

If $I_{\mathrm{depol}} < I_\Theta := V_\Theta/R$, this equation does not have a solution; the current is too week to drive depolarization to threshold. Otherwise, the threshold will be reached at time

$$\Delta t = -\tau \ln\left(1 - \frac{I_\Theta}{I_{\mathrm{depol}}}\right) = \tau \ln\left(\frac{I_{\mathrm{depol}}}{I_{\mathrm{depol}} - I_\Theta}\right). \tag{1.20}$$

Δt is the inter-spike interval; its inverse gives the spike rate resulting from the input current I_{depol}.

Figure 1.11 shows the transfer function in a standardized way, i.e., plotting spikes per time constant over the ratio of input and threshold current. Above threshold, the

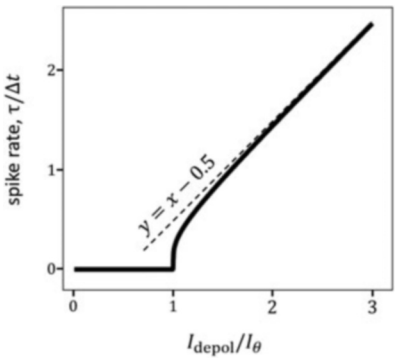


Fig. 1.11 Transfer function of the integrate-and-fire model for a constant input current. The x-axis shows the input current in multiples of the threshold current $I_\Theta = V_\Theta/R$. The y-axis shows the spike rate taking the time constant τ as a unit. The transfer function describes a point nonlinearity similar to the rectifying linear unit ("relu") discussed in Chap. 2. The dashed line shows the asymptotic behavior for large input currents

curve is approximately linear with slope 1. In the context of point nonlinearities (Sect. 2.3.1), this behavior is known as a rectifying linear unit, or "relu." Of course, spike rates in real neurons have an upper limit, determined by the length of the spike itself and the refractory period. Since these take several milliseconds, spike rate is usually limited to some 300 Hz, which roughly corresponds to the value $\tau/\Delta t = 1$ in Fig. 1.11. In the integrate-and-fire model, higher spike rates are possible since the duration of the spiking event is set to zero.

1.3.2 State Space Analysis

One key property of the Hodgkin–Huxley system is the all-or-nothing dynamics of the spike, i.e., the existence of a depolarization threshold below which the system simply relaxes to a steady state, while it generates spikes when stimulated above threshold. Such qualitative and abrupt changes of behavior resulting from the continuous variation of a parameter are called bifurcations. Figure 1.12 shows this behavior for continuous variation of a depolarizing current $-I_{\text{depol}}$ added to the right side of Eq. 1.12. The sign is chosen such that positive currents cause depolarization, i.e., act in the same way as sodium ion inflow.

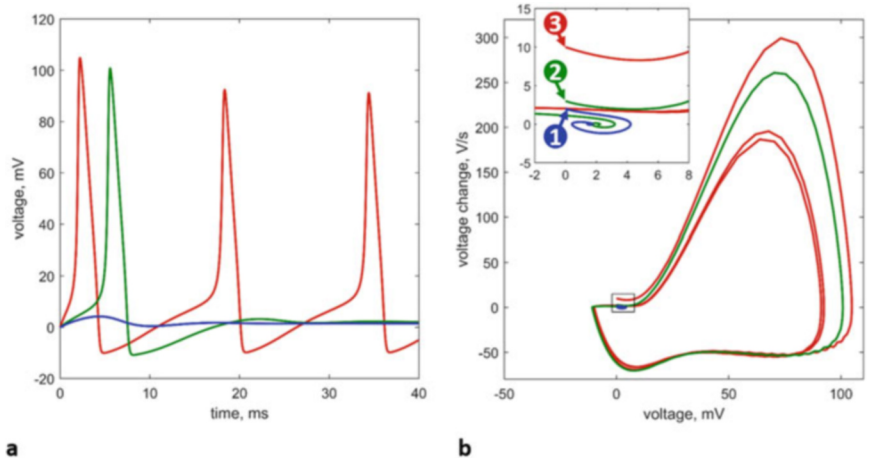


Fig. 1.12 Bifurcation of the Hodgkin–Huxley system in response to various levels of constant input current. Blue: weak current (2 nA), green: 3 nA, red: 10 nA. (**a**) Membrane potential (depolarization) as a function of time. For small stimulation, the system relaxes to a slightly shifted steady state without spiking. For medium stimulation, one spike is generated after which another fixed point is approached. Stronger stimulation results in a periodic spike train. (**b**) System evolution shown as a trajectory in a state space spanned by V and V'. The inset shows the region around the origin with the initial conditions (trajectory starting points) for the three stimulation strengths. The blue and green curves settle for a fixed point, while the red curve runs into a limit cycle

The representation as a movement in state space shown in Fig. 1.12b is a useful tool to describe the qualitative behavior of the dynamic system. Instead of plotting V vs. V', we could also have chosen to plot V vs. g_K or g_{Na} vs. g_K. In these cases, we would see the same "qualitative" behavior, i.e., relaxation to a fixed point or approach of a limit cycle, although the actual shape of the curves would look quite different. The reason for this is that the complete trajectory of the system in the four-dimensional state space spanned by V, n, m, and h also exhibits that same qualitative behavior, i.e., fixed points and limit cycles, and that the projection to a reduced state space such as the one used in Fig. 1.12 captures this overall structure.

The oscillatory behavior shown in the red trace of Fig. 1.12 is also the basis of the pacemaker activity in the sinoatrial node of the heart (Wilders 2007). The depolarizing current is realized by an additional type of channel effecting a long-lasting inflow of cations triggered by hyperpolarization ("funny channel"). In addition, the heart cells have a calcium channel causing the action potential to stay in a depolarized state for an extended period of time, as well as potassium channels with different dynamics. In effect, pacemaker cells of the sinoatrial node generate spikes at a rate of about 1 per second, which trigger the heart beat with its ordinary pace.

Box 1.8 Qualitative Behavior of Dynamical Systems

In the theory of nonlinear dynamical systems, the "qualitative" behavior plays an important role. For example, the system might have a fixed point to which it relaxes upon disturbance. A fixed point of this type would be called stable, because small disturbances will not lead to qualitative changes of the system. Alternatively, a fixed point may be unstable such that the system will move away from the fixed point whenever a disturbance occurs, however small. This is often compared to a ball moving on the surface of a bowl. If the bowl is in standard orientation, the ball will move to the bottom and return there upon small distortions. If, however, the bowl is turned upside down, a ball resting on top will inevitably roll away upon the smallest pushing. Note that the "bowl" on which the ball moves is a surface in the system's state space, spanned by the dynamic variables, i.e., the current position and velocity of the ball. In the case of the action potential, the state space is spanned by the state variables of the Hodgkin–Huxley equations, V, n, m, and h. The resting state of the Hodgkin–Huxley system is a stable fixed point, as is clearly visible in Figs. 1.9a and 1.12 where a disturbance of 6 mV or 2 nA, respectively, does not lead the system qualitatively away from the resting state.

Nonlinear systems may also have periodic solutions: that is, they can be oscillators as is the case for the Hodgkin–Huxley system with a constant depolarizing current. The system will then produce a continuing series of spikes, each with the same time course (Fig. 1.12, red traces). Mathematically speaking, there exists a closed path in state space such that the system keeps

(continued)

Box 1.8 (continued)
orbiting this path. Such closed paths are known as limit cycles or orbits. As stable fixed points, limit cycles can be "attractors" in the sense that trajectories starting off the limit cycle, but in vicinity of it, will approach the limit cycle and return to it after small disturbances. Oscillation or limit cycle behavior can occur in systems of second order or higher.

A final type of qualitative behavior is deterministic chaos. In this case the attractor is not a point or a curve but a volume in state space to which the system will return, albeit with ever changing paths taken inside the attractor. This behavior requires differential equations of order higher than two. Note, however, that the Hodgkin–Huxley system, although of order 4, does not show chaotic oscillations.

Transitions between different types of qualitative behavior occurring with the change of a system parameter are called bifurcations. An example is the transition from a stable fixed point to a limit cycle shown in Figs. 1.12 and 1.14.

1.3.3 The FitzHugh–Nagumo Equations

The choice of two-dimensional projections of the full Hodgkin–Huxley system as in Fig. 1.12b is to some extent arbitrary and reflects the availability of data rather than mathematical requirements. For a rigorous analysis of qualitative behavior, it is therefore better to formulate a reduced system of the differential equations themselves, ideally with just two dimensions, for which the complete state space can be analyzed. One such system is given by the so-called Fitzhugh–Nagumo equations (FitzHugh 1961; Murray 2002):

$$\frac{dv}{dt}(t) = f(v(t)) - w(t) + I_a \tag{1.21}$$

$$\frac{dw}{dt}(t) = b\, v(t) - \gamma\, w(t), \tag{1.22}$$

where

$$f(v) := v\,(a - v)\,(v - 1). \tag{1.23}$$

It has only two state variables that can be thought of as an "activator" v replacing the Hodgkin–Huxley variables V and m, and an inhibitor w reflecting the properties of n and h. Indeed, modeling the dynamics of these variables separately does not add much to the Hodgkin–Huxley model. The nonlinearities are collected in a third-order polynomial function f. Finally, I_a is an externally applied current, and a, b, γ are constants.

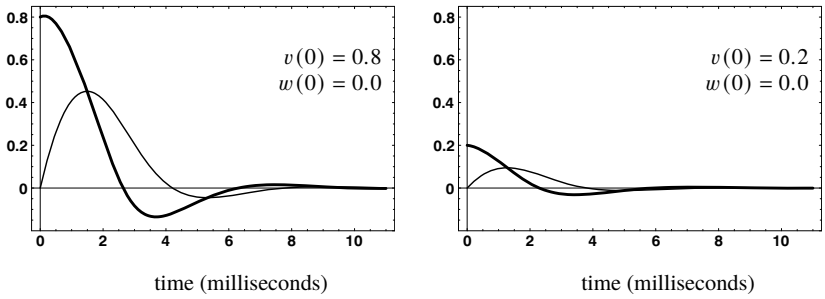


Fig. 1.13 Solutions of the FitzHugh–Nagumo system with $I_a = 0$, $a = 0.3$, $b = 0.8$, $\gamma = 1.0$. Left: start at $v = 0.8$, right: start at $v = 0.2$

The FitzHugh–Nagumo equations are much simpler than the Hodgkin–Huxley equations in that they employ only two (rather than four) coupled variables. Examples of numerical solutions are shown in Fig. 1.13. The qualitative behavior of the FitzHugh–Nagumo system is illustrated in Fig. 1.14 that shows the complete state space together with the vector field that drives the system evolution and the so-called null isoclines discussed below.

We can now look for states (i.e., pairs of numbers (v, w)) for which the system of Eqs. 1.21–1.23 stops moving, i.e., for the fixed points of the system. From Eq. 1.21 we obtain:

$$\frac{dv}{dt} = 0 \quad \Leftrightarrow \quad w = f(v) + I_a \tag{1.24}$$

$$\frac{dw}{dt} = 0 \quad \Leftrightarrow \quad w - \frac{b}{\gamma} v. \tag{1.25}$$

These curves are the null isoclines mentioned above. In the case of Fig. 1.14a, where the current I_a is zero, they intersect at the only fixed point $(v, w) = (0, 0)$. 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 upward. Depending on the slope of the isoclines, there can be either one or three intersection points, i.e., points where both derivatives vanish. Figure 1.14b 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–

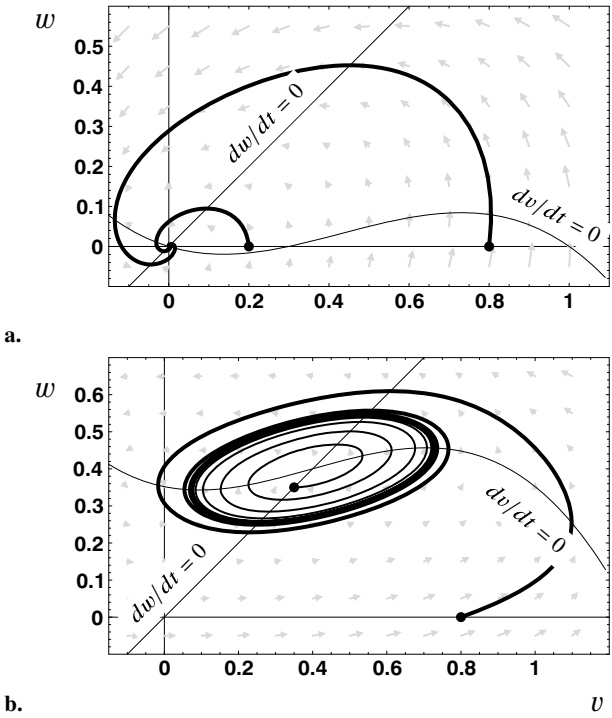


Fig. 1.14 State space and phase trajectories of the FitzHugh–Nagumo system. (**a**) Non-oscillating case. Parameters as in Fig. 1.13. The arrows give the initial movement (direction and speed) of solutions starting from the respective points. The thin lines are the "null isoclines" $dw/dt = 0$ (arrows horizontal) and $dv/dt = 0$ (arrows vertical). The heavy lines show the two solutions from Fig. 1.13. They converge to the intersection of the isoclines that 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 has moved upward. The intersection of the isoclines is no longer a stable point, while a stable limit cycle around the fixed point has emerged. Two solutions are shown approaching a stable limit cycle from within and from outside

Nagumo system can produce oscillating responses when activated by an applied current. The transition from a stable fixed point to a limit cycle, effected by the change of a parameter, is called a Hopf bifurcation.

► **Key Point** The Hodgkin–Huxley equations are best suited to model physiological processes on the level of membranes and single units. More abstract models such as integrate-and-fire neurons or the FitzHugh–Nagumo system have been developed to model spike rates in neural networks or membrane oscillators.

1.4 Passive Conduction

1.4.1 Core Conductors

In the space clamp, we have assumed that the potentials and currents are constant along a section of an axon. In reality, as soon as the copper wire effectuating the space clamp is removed, 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 the action potential to propagate along the fiber. Due to the refractory period of the membrane, the propagation will be directed. In this section, we also use the term "neurite" to denote any type of neural fiber, be it axon or dendrite.

Figure 1.15 shows the relevant currents in a so-called core conductor consisting of a conducting core (the cytoplasm) and a partially insulating membrane, surrounded by an outer medium (the extracellular space) that we will generally assume to be grounded. The main current flows in axial direction within the core; its strength in a segment of length Δx is determined by the voltage difference between the two ends of the segment, $V(x + \Delta x) - V(x)$, and the axial resistance. Total resistance of a segment will be proportional to the segment length. We therefore express axial resistance as a specific resistance r_a, measured in $\Omega\,\mathrm{cm}^{-1}$, and understand that the total resistance of the segment will be $r_a \times \Delta x$.

The second current shown in Fig. 1.15b is the membrane current flowing per segment length (Δx) across the insulating membrane; its dimension is $\mathrm{A\,cm}^{-1}$. This current is composed of two parts, a resistive and a capacitive current, both of which are driven by the local membrane potential V. The resistive current depends on the lumped conductivities g_i of all ion channels that would now be measured in $\mathrm{S\,cm}^{-1}$. Expressed as an Ohmian membrane resistance, we have $r_m = (\sum g_i)^{-1}$

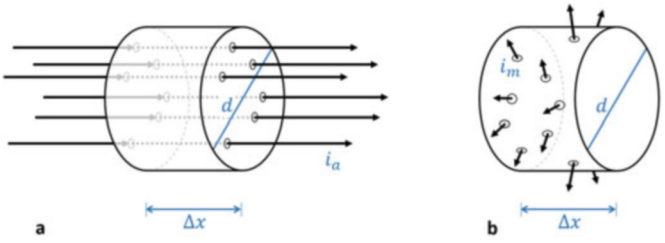


Fig. 1.15 Current flows in passive condition in a cylindrical neurite. d: diameter of cylindrical neurite, Δx length segment. (**a**) Axial flow, i_a, with physical dimension A. (**b**) Membrane current, i_m. This is a "specific current" since the total membrane current in a segment will be proportional to Δx. Membrane current is therefore measured in $\mathrm{A\,cm}^{-1}$

Table 1.2 Quantities involved in the cable equation. Only cylindrical membrane tubes are considered and all lengths are axial. V, i_a, and i_m vary along the neurite (space variable x), while r_a, r_m, and c_m are treated as constants

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

with the dimension $(\mathrm{S\,cm^{-1}})^{-1}$ or $\Omega\,\mathrm{cm}$. Membrane resistance r_m will decrease with segment length, Δx.

The capacitive current describes the attachment of ions to or the detachment of ions from the membrane and depends on membrane capacitance that is proportional to segment length. We describe it by the specific membrane capacitance c_m measured in $\mathrm{F\,cm^{-1}}$. All quantities are summarized in Table 1.2.

It may be confusing at first glance that Ohmian currents in this section are described by resistances R, while we used conductivities g in the Hodgkin–Huxley theory. This seems to have mostly historical reasons. The basic relation of the two quantities is that one is the inverse of the other, $R = 1/g$. Resistances add up when connected in a row (serial circuitry), while conductances add up when combined in parallel.

The hydraulic analogy of a core conductor is a garden watering hose made of some elastic and leaky material. The membrane potential is replaced by the pressure in the hose, while current becomes water flow. The axial current corresponds to the water running inside the hose, while the Ohmian component of membrane current is the water dripping out of the leaks. Capacitive membrane current is modeled by the water streaming in and out of elastic extensions of the hose wall forming when pressure changes. Axial and membrane resistances correspond to cross-sectional area and porosity, while capacitance is replaced by wall elasticity. If we assume that the leaks are realized as a row of little holes on the upper side of the hose, we can imagine what happens if we turn on the water: A row of "fountains" will spring upward from the hose, indicating the local pressure by their height. Along the hose, the tops of the fountains will form a decreasing line, looking somewhat like an exponential decay curve. In the next section, we will derive a mathematical description modeling this and other properties of the core conductor.

1.4.2 The Cable Equation

We will now discuss the theory of core conduction in terms of an electrical equivalent circuit. For this, we will need some basic relations governing the

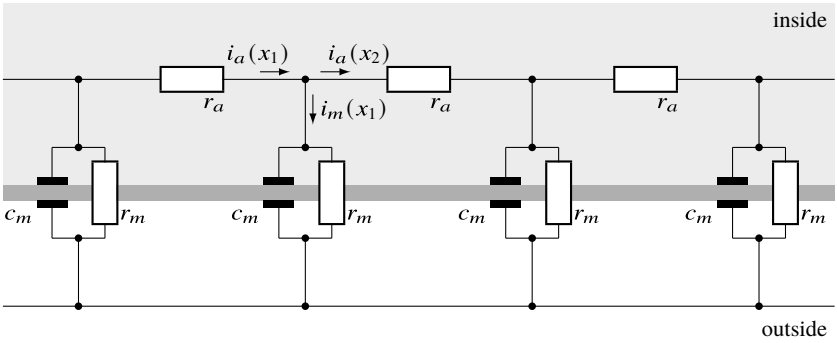


Fig. 1.16 Electrical circuit for modeling the passive propagation of activity along an axon. i_a axial current, i_m membrane current, r_a axial resistance, r_m membrane resistance, c_m membrane capacitance

analysis of electrical circuits, i.e., the relations of potential and current in resistors and capacitors, and Kirchhoff's node and mesh rules (see Box 1.2). Note that Kirchhoff's node rule is a simple consequence of the fact that charges are conserved, i.e., that current flowing into a node must also come out. Likewise, Kirchhoff's mesh rule is a consequence of the definition of the voltage between two nodes in the mesh as the difference between the potentials of these nodes. When going in a loop, these voltage steps add up to zero, just as elevation changes would do when walking a closed path in a mountain range.

Figure 1.16 shows the equivalent circuit for passive conduction with the axial and membrane currents as explained above. The goal of this section is to find an expression for the variation of the membrane potential with time t and position along the neurite x.

The basic relations can be inferred from inspection of Fig. 1.16. 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 Fig. 1.16, the length variable is discretized, but we will use a continuous representation involving spatial and temporal derivatives. That is, we understand that

$$\frac{\partial V}{\partial x}(x,t) \approx \frac{V(x+\Delta x,t) - V(x,t)}{\Delta x}. \tag{1.26}$$

The curly d (∂) denotes partial derivatives that are needed here since V depends on space and time, whereas the derivative is taken only with respect to space. We will not further elaborate on partial derivatives here; for a mathematical definition see Box 5.4.

Following Ohm's law (Box 1.2b), the axial current $i_a(x,t)$ is proportional to the potential change in axial direction, i.e.,

$$i_a(x,t) = -\frac{1}{r_a}\frac{\partial V}{\partial x}(x,t), \tag{1.27}$$

where r_a is the axial resistance measured in $\Omega\,\mathrm{cm}^{-1}$. The minus sign follows from the convention that positive current is positive charge moving from "plus to minus," i.e., downhill on the potential surface.

An axial current originating at some position $x = 0$ will decrease for larger distances x from the origin since part of it will leak across the membrane, thus forming a membrane current i_m. From Kirchhoff's node rule (Box 1.2d) in the discrete case, we see that $i_a(x,t) - i_a(x+\Delta x,t) - i_m(x,t) = 0$, from which we obtain the continuous formulation:

$$-i_m(x,t) = \frac{\partial i_a}{\partial x}(x,t). \tag{1.28}$$

Substituting for i_a from Eq. 1.27, we obtain

$$i_m(x,t) = \frac{1}{r_a}\frac{\partial^2 V}{\partial x^2}(x,t). \tag{1.29}$$

This equation relates the membrane current to the second spatial derivative of membrane potential. If i_m would be zero: that is, in the case of an ideally insulated and capacitance-free conductor, this implies that the potential will drop linearly with distance, in accordance with Ohm's law.

The membrane current i_m has two components, a resistive, or leak current (following Ohm's law, Box 1.2b) and a capacitive current (Box 1.2c). We may write

$$i_m(x,t) = \frac{1}{r_m}V(x,t) + c_m\frac{\partial V}{\partial t}(x,t). \tag{1.30}$$

Note that this equation is completely analogous to the total current equation of the Hodgkin–Huxley theory, Eq. 1.2. The only differences are the consideration of a spatial variable x, the replacement of the individual ion currents by a single one, and the formulation with a resistance rather than with conductivities.

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

$$V(x,t) = \frac{r_m}{r_a}\frac{\partial^2 V}{\partial x^2}(x,t) - r_m c_m\frac{\partial V}{\partial t}(x,t). \tag{1.31}$$

This is a partial differential equation for $V(x,t)$ 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

$$V(x) = \frac{r_m}{r_a}\frac{d^2V}{dx^2}(x) \tag{1.32}$$

with boundary conditions $V(0) = V_o$ and $\lim_{x\to\pm\infty} V(x) = 0$. The latter constraint describes the fact that the depolarization of a passive membrane can only decay from x_o. The spread of the potential along the neurite is then described by the according solution of Eq. 1.32,

$$V(x) = V_o \exp\left(\frac{-|x|}{\sqrt{r_m/r_a}}\right); \tag{1.33}$$

a plot appears at the upper curve in the left part of Fig. 1.17 ($t = \infty$).

Equation 1.33 describes an exponential decay of the local potential V_o with a "length constant" $\lambda = \sqrt{r_m/r_a}$. Note that λ has the dimension of a length, since r_m and r_a are measured in $\Omega\,\mathrm{cm}$ and $\Omega\,\mathrm{cm}^{-1}$, respectively, resulting in the dimension $\sqrt{\Omega\mathrm{cm}\Omega^{-1}\mathrm{cm}} = \sqrt{\mathrm{cm}^2} = \mathrm{cm}$. In typical cortical neurons, λ takes values in the order of a millimeter. Since r_m decreases with the circumference of an axon, while r_a decreases with its cross-sectional area, thicker axons have larger length

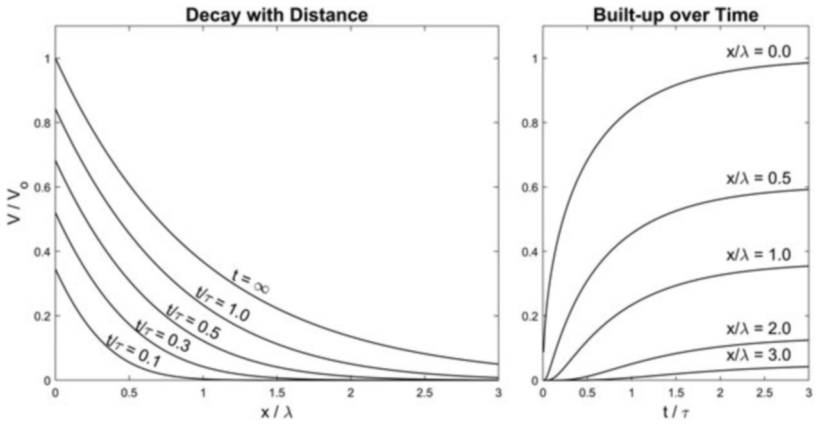


Fig. 1.17 Analytical solution of the cable equation after Hodgkin and Rushton (1946). A constant depolarizing current is inserted into a homogeneous neurite at position $x = 0$ and spreads symmetrically in both directions. V_o is the steady state of the membrane potential reached after prolonged stimulation at position $x = 0$. Lengths and times are given in multiples of the length and time constants, λ and τ. **Left**: Membrane potential as a function of distance from the stimulation site. The top curve ($t = \infty$) is an exponential reached as a steady state. The other curves have slightly different shapes since membrane potential builds up faster in the vicinity of the stimulation site. **Right**: Temporal built-up of the membrane potential at a number of distances from the stimulation site. Note that the curves get more sigmoidal for more distant membrane patches

constants. The length constant determines the range of passive conductance and thereby the speed of conduction. In myelinated axons, 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 has been studied by Rall (1962).

In dynamic cases, such as the propagation of action potentials on axons or the 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 Eq. 1.31. It is largely independent of cable diameter but can be reduced by reducing the membrane capacitance. Myelin sheaths, 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 nonmyelinated axons of equal thickness (see also Sect. 1.5.3).

The spatiotemporal cable Eq. (1.31) is hard to solve analytically; for discussions of the issue see Jack et al. (1975); Tuckwell (1988), and Ermentrout and Terman (2010). The best studied case concerns a constant membrane current delivered from time $t = 0$ at location $x = 0$ as shown in Fig. 1.17 (Hodgkin and Rushton 1946). In this case, the membrane potential will approach the steady state described by Eq. 1.33. As might have been expected from the hydrodynamic analogy discussed above, the approach to the steady state is the slower, the further away a given membrane patch is from the stimulation site.

► **Key Point** Local variations of membrane potential spread along neurites by means of Ohmian currents flowing in axial direction. Since some charges leak across the membrane or are lost in the recharging of the membrane capacitor, the membrane potential decays from a stimulation site with length constant $\lambda = \sqrt{r_m / r_a}$ and time constant $\tau = c_m r_m$.

1.5 Propagating Action Potentials

1.5.1 The Fuse Analogy

As an instructive analogy for the axon with a propagating action potential, consider a powder fuse as used for example in fireworks. Its core contains gunpowder that provides the chemical energy required for the conduction process. When the fuse is lit at one end, the spark will heat up an adjacent section of the fuse that will then ignite and the spark will propagate (i.e., burn) toward the far end of the fuse. The spark is of course the analogy of the action potential, while the heating and subsequent ignition of neighboring portions of the fuse are analogous to passive conduction and the initiation of an action potential.

If the fuse is lit in the middle, sparks will travel in both directions. This is generally also true for axons: If an electrical stimulus is delivered at some distance from the soma, two spikes will be elicited one traveling "upstream" toward the soma and another one traveling further "downstream" toward the synapse. The two travelling directions are called "antidromic" (toward the soma) and "orthodromic," respectively.

This raises the question why a spike traveling in the orthodromic direction, say, does not constantly initiate spikes running backward toward the soma. In the fuse analogy, this is impossible since the powder behind the current position of the spark is already burnt and cannot be lit again since the chemical energy has been dissipated. In the axon, the energy needed for the spiking event is stored in the ion distributions of sodium and potassium and the resulting equilibrium potentials of these ion sorts. As is demonstrated in Box 1.7, a single spike does not change these concentrations substantially, meaning that energy for more spikes is still available.

The reason why propagating spikes on axons do not elicit spikes in the backward direction is of course the refractory period of the sodium channel. This refractory period is transient, and when it ends, new spikes may indeed be initiated. The passing spike, however, which caused the refractory period in the first place, will now be too far away for doing this. Together with the sodium–potassium pump, which constantly replenishes the energy reservoir, the axon thus acts as a self-refilling, reusable fuse.

1.5.2 The Spatiotemporal Theory

Active conduction is the combination of passive conduction, depolarization of adjacent sections of the axon, and the subsequent initiation of action potentials in the depolarized sections. With the cable equation, we can formulate the Hodgkin–Huxley Eq. 1.12 for this case: that is, the spatiotemporal theory. For active membranes, the total leak current V/r_m in Eq. 1.30 has to be replaced by the membrane current passing through the individual ion 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

$$\frac{1}{r_a}\frac{\partial^2 V}{\partial x^2}(x,t) = c_m \frac{\partial V}{\partial t}(x,t) + \bar{g}_\mathrm{K}\, n(t)^4\, (V(t) - V_\mathrm{K}) + \bar{g}_\mathrm{Na}\, m(t)^3\, h(t)\, (V(t) - V_\mathrm{Na}) + \bar{g}_l\, (V(t) - V_l). \quad (1.34)$$

Equations 1.13–1.15 remain unchanged, except for the fact that the variables V, n, m, and h now depend additionally on spatial position x. Equation 1.34 is a partial differential equation that 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; that is to say, propagation is free of dispersion. Dispersion-free propagation cannot be concluded

from Eq. 1.34 itself but is an additional empirical fact that is used to develop the theory. It holds for sections of axons in which the diameter does not change substantially. For a mathematical formulation, consider an action potential $V(x,t)$ traveling with a velocity θ. If its shifted version after time Δt and traveled distance $\theta\Delta t$ has the same shape, we have

$$V(x,t) = V(x-\theta\Delta t, t-\Delta t). \tag{1.35}$$

Thus, $V(x,t)$ can be written as a one-dimensional function $V^*(x-\theta t)$. Taking the second derivatives of V^* with respect to x and t, and dropping the asterisk, it is easy to see that the wave equation

$$\frac{\partial^2 V}{\partial x^2} = \frac{1}{\theta^2}\frac{\partial^2 V}{\partial t^2} \tag{1.36}$$

must hold. Here, θ is the speed of neural conduction. With the wave equation, we can replace the second spatial derivative in Eq. 1.34 by a second temporal derivative and obtain the ordinary differential equation

$$\begin{aligned}\frac{1}{r_a\theta^2}\frac{d^2V}{dt^2}(t) = c_m\frac{dV}{dt}(t) + \bar{g}_{\mathrm{K}}\, n(t)^4\,(V(t)-V_{\mathrm{K}}) \\ +\bar{g}_{\mathrm{Na}}\, m(t)^3\, h(t)\,(V(t)-V_{\mathrm{Na}}) + \bar{g}_l\,(V(t)-V_l).\end{aligned} \tag{1.37}$$

The conduction speed θ is an unknown that is implicitly determined by the requirement that the solutions of Eq. 1.37 should converge to zero for large t. It depends on the length and time constants of passive propagation as well as on the rate constants of the channel opening and closing processes. Hodgkin and Huxley (1952) obtained numerical solutions by initially guessing the parameter θ and then correcting it, if the solution started to diverge (see also Ermentrout and Terman 2010, for a discussion of the numerical procedure). The final estimate of θ is $18.8\,\mathrm{m\,s^{-1}}$, using the other parameters as given in the coding example in Box 1.6. The solutions again reproduce the empirical shape of the action potential.

The dispersion-free propagation property given in Eq. 1.35 also implies that the action potential may be plotted as a function of time at a fixed position (as in Figs. 1.9 and 1.12) or as a function of space (axon length) as in Fig. 1.3.

The wave equation cannot be applied if axons bifurcate or if axon diameter and channel concentrations change along axon length. In these cases, Eq. 1.37 needs to be spatially discretized into the so-called compartments, little cylindric sections within which the electrical parameters are constant. These compartments are connected by nodes for which Kirchhoff's node rule (Box 1.2d) applies. Dendritic compartments are modeled similarly by the discretized cable equation (1.31), see Hines and Carnevale (1997). This approach is known as compartmental modeling and has been successfully used to study the role of complex dendritic and axonal anatomies. Of course, the more compartments are considered, the more free

parameters will have to be set, which also limits the applicability of this approach. For a review, see Almog and Korngreen (2016).

1.5.3 The Speed of Neural Conduction

In the previous section, we have seen that the conduction speed of the Hodgkin–Huxley model of the squid axon is about $18.8\,\mathrm{m\,s^{-1}}$, which is in reasonable agreement with physiological measurements. With all other parameters (such as temperature, ion concentrations, rate constants) being equal, the velocity is determined by the length and time constants of the cable equation (1.31). These constants depend on the diameter of the fiber, as can be seen from the following considerations:

For the axial current (Fig. 1.15a), a wider axon acts as if more conductors had been added in parallel, all with the same specific axial resistance. The total axial resistance therefore scales with the inverse of the cross-sectional area of the axon, or its diameter squared,

$$r_a \propto d^{-2}. \tag{1.38}$$

The membrane resistance and capacitance depend on the surface area of the axon, which is proportional to the diameter (see Fig. 1.15b):

$$r_m \propto d^{-1} \tag{1.39}$$

$$c_m \propto d. \tag{1.40}$$

From these relations, we immediately see that $\lambda \propto \sqrt{d}$, while τ does not depend on the axon diameter at all. Thicker fibers therefore have a higher conduction speed proportional to $\sqrt{d}$, which is of course the reason for having giant fibers in many invertebrates such as the squid or the earthworm. The reported speed of $18.8\,\mathrm{m\,s^{-1}}$ for the squid giant axon is among the highest speeds found in unmyelinated axons.

Achieving higher conduction speed was of course an important issue in the evolution of the nervous system (Castelfranco and Hartline 2016). Myelin sheaths and the saltatory conduction from one node of Ranvier to the next, as are found mostly in vertebrates, allows conduction speeds of $120\,\mathrm{m\,s^{-1}}$ and more. In this case, action potentials occur only at the nodes, where the axon is in direct contact with the intercellular medium, while in the internodial sections, where the axon is covered by the myelin sheath, conduction is purely passive. In the internodia, membrane resistance r_m is much larger than in the Ranvier nodes, both because of a reduced number of channels ($\bar{g}_{\mathrm{Na}}$, $\bar{g}_{\mathrm{K}}$ and g_l in the Hodgkin–Huxley model; see Rasband 2010) and the insulation provided by the myelin sheath. This leads to an increase of both the length and the time constants, which of course have opposite effects on conduction speed. The increase of the time constant is, however, compensated by the reduced membrane capacitance c_m, which is also caused by the

myelin sheath, intuitively by pushing apart the conducting "plates" of the capacitor. In effect, passive conduction in the internodia is therefore much faster than in an unmyelinated axon. For a detailed analysis of the circuits and cable models for saltatory conduction, see Cohen et al. (2020).

Conduction speed in myelinated fibers also increases with axon diameter. Empirical measurements show that this increase is roughly linear, i.e., speed is proportional to diameter, while we saw that it is proportional to the square root of diameter in unmyelinated fibers (Rushton 1951). For axons with a diameter of less than 2μm, no advantage of myelination and saltatory conduction seems to exist.

► **Key Point** Action potentials propagate along the axon by passive spreading of the depolarization and subsequent initiation of active channel opening and closing mechanisms. This is described by a combination of the Hodgkin–Huxley equations with the cable equation.

1.6 Summary and Further Reading

In this chapter, we have studied the formation and propagation of action potentials in five 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. An analysis of this 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. Simpler models can be used for various applications, including the integrate-and-fire neuron for the study of larger networks and the FitzHugh–Nagumo model for qualitative behavior such as oscillations.
4. 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.
5. Propagation of action potentials is modeled by combining the cable equation with the kinetic channel models derived from the voltage clamp experiments.

Texts

Huang (2020): *Latest edition of a classical text on the neurophysiology of single cells, with an eye on mathematical theory.*

Katz (1966): *Though many years out of print, this is still a most useful primer and lucid introduction to the electro-physiology of single cells.*

Ermentrout and Terman (2010): *Comprehensive treatment of neural activity in membranes, single cells, and certain types of neural networks from a dynamical systems point of view.*

Gerstner and Kistler (2002): *Starting with the classical Hodgkin–Huxley theory, this monograph covers a wide variety of single neuron models at various levels of abstraction, as well as simulation approaches to networks of spiking neurons.*

Koch and Segev (1998): *Edited volume covering all aspects of nervous conduction and activity, including also chapters on neural networks and spike train analysis.*

Jack et al. (1975): *In depth mathematical and physical analysis of neural cable theory, dendritic summation, active membranes and many related topics. Available online at https://www.denisnoble.com/wp-content/uploads/2021/01/JNT-2.pdf*

Suggested Original Papers for Classroom Seminars

Hodgkin and Huxley (1952): *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.*

Naundorf et al. (2006): *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.*

Takkala and Prescott (2018): *Application of the dynamic clamp technique to study the mechanism of chronic somatosensory pain.*

Wilders (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.*

Rall (1962): *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.*

Hines and Carnevale (1997): *Important review paper on compartmental modeling, introducing both the basic concepts and the simulation tools.*

London and Häusser (2005): *Overview over computations carried out by dendrites and dendritic summation, i.e., by mechanisms of passive conduction.*

References

Abbott, L. F. 1999. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin* 50: 303–304.

Aidley, D. J. 1998. *The Physiology of Excitable Cells*. 4th ed. Cambridge: Cambridge University Press.

Almog, M., and A. Korngreen. 2016. Is realistic neural modeling realistic? *Journal of Neurophysiology* 116: 2180–2209.

Bean, B. P. 2007. The action potential in mammalian central neurons. *Nature Reviews Neuroscience* 8: 451–465.

Castelfranco, A. M., and D. K. Hartline. 2016. Evolution of rapid nerve conduction. *Brain Research* 1641: 11–33.

Cohen, C. C., M. A. Popovic, J. Klooster, M.-T. Weil, W. Möbius, K.-A. Nave, and M. H. P. Kole. 2020. Saltatory conduction along myelinated axons involves a periaxonal nanocircuit. *Cell* 180: 311–322.

Dixon, R. E., M. F. Navedo, M. D. Binder, and F. Santana, 2022. Mechanisms and physiological implications of cooperative gating of clustered ion channels. *Physiological Reviews* 102: 1159–1210.

Ermentrout, G. B., and D. H. Terman. 2010. *Mathematical Foundations of Neuroscience*. New York: Springer.

Fink, M., and D. Noble. 2009. Markov models for ion channels: Versatility versus identifiability and speed. *Philosophical Transactions of the Royal Society A* 367: 2161–2179.

FitzHugh, R. 1961. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal* 1: 455–461.

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

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

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

Hodgkin, A. L., and W. A. H. Rushton. 1946. The electrical constants of a crustacean nerve fibre. *Proceedings of the Royal Society (London) B* 133: 444–479.

Huang, C. L.-H. 2020. *Keynes & Aidley's Nerve and Muscle*. 5th ed. Cambridge: Cambridge University Press.

Izhikevich, E. M. 2004. Which model to use for cortical spiking models? *IEEE Transactions of Neural Networks* 15: 1063–1070.

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

Katz, B. 1966. *Nerve, Muscle, and Synapse*. New York: McGraw–Hill.

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

Laughlin, S. B., R. R. de Ruyter van Steveninck, and A. C. Anderson. 1998. The metabolic cost of neural information. *Nature Neuroscience* 1: 36–41.

London, M., and M. Häusser. 2005. Dendritic computation. *Annual Review of Neuroscience* 28: 503–532.

Murray, J. D. 2002. *Mathematical Biology—An Introduction*. 3rd ed. Berlin: Springer.

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

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

Rasband, M. N. 2010. The axon initial segment and the maintenance of neural polarity. *Nature Neuroscience* 11: 552–562.

Rudy, Y., and J. R. Silva. 2006. Computational biology in the study of cardiac ion channels and cell electrophysiology. *Quarterly Reviews of Biophysics* 39: 57–116.

Rushton, W. 1951. A theory of the effects of fibre size in medullated nerve. *Journal of Physiology* 115: 101–122.

Sakmann, B., and E. Neher. 1984. Patch clamp techniques for studying ionic channels in excitable membranes. *Annual Review of Physiology* 46: 455–473.

Sharp, A. A., M. B. O'Neil, L. F. Abbott, and E. Marder. 1993. Dynamic clamp: Computer-generated conductances in real neurons. *Journal of Neurophysiology* 69: 992–995.

Takkala, P., and S. A. Prescott. 2018. Using dynamic clamp to quantify pathological changes in the excitability of primary somatosensory neurons. *The Journal of Physiology* 596: 2209–2227.
Tuckwell, H. 1988. *Introduction to Theoretical Neurobiology*. Vol. 1 and 2. Cambridge: Cambridge University Press.
Wilders, R. 2007. Computer modelling of the sinoatrial node. *Medical & Biological Engineering & Computing* 45: 189–207.

Receptive Fields and the Specificity of Neuronal Firing

2

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 visual direction of a light stimulus, the pitch of a tone, or the spot of skin hit by some object. Outside the sensory areas of the brain, specificities for more abstract concepts have been found including cells representing place (the animal's current position) or the numerosity of objects. In the motor parts of the brain, neurons have been found that fire preferably prior to movements of body parts into a certain direction. In the sensory pathways, the specificity of a neuron is quantified by means of its receptive field. Receptive fields are measured by correlating the activity of the neuron with externally recorded parameters of the stimulus; the approach is known as reverse correlation.

In this chapter, we discuss the basic theory of visual receptive fields that can be extended to similar concepts in other sensory, motor, or associative areas. The theory is closely related to linear systems theory applied to spatiotemporal signals such as image sequences. Mathematically, it rests on integral equations of the convolution type which will be introduced in due course.

Learning Objectives

- Receptive fields as descriptions of neuronal specificity and reverse correlation as the corresponding measurement protocol
- Lateral inhibition and the generalization of the receptive field to layers of neurons
- Convolution and linear shift-invariant (LSI) systems

(continued)

H. A. Mallot, *Computational Neuroscience*,
https://doi.org/10.1007/978-3-031-75705-1_2

- Point-spread function and the receptive field function as characteristics of linear shift-invariant systems
- Extension of the spatial theory to temporal and spatiotemporal summation
- Point nonlinearity and other types of nonlinear systems

2.1 Specificity and Reverse Correlation

The action potential as discussed in the previous chapter is the general and ubiquitous neural signal occurring in all sensory and motor systems alike, with only minor variations. This raises the question of how to assess the contents conveyed by a given spiking event: Does a particular spike signal the presence of a familiar face, an unusual sound, a memory recall, or a motor action to be carried out? The answer to this question is that the represented content is a property of the particular neuron which is producing the spike. The spike itself simply indicates that the content associated with the firing neuron is indeed present. This is now generally referred to as the "specificity" of a neuron, following Johannes Müller's[1] famous "law of the specific sense energies" (Müller 1837; Norrsell et al. 1999). In this chapter, we discuss quantitative descriptions of neuronal specificity in terms of the neuron's "receptive field." The question, how content is encoded in a population of neurons with different specificities, will be addressed in the context of population coding in Chap. 7.

The term "receptive field" goes back to the work of Sherrington[2] (1906) on the scratching reflex in dogs. When itched at some spot of its skin, the dog will scratch that spot with the leg able to reach this spot. That is to say, the reflex can be elicited in different forms by itching at various spots on the skin. Sherrington introduces the receptive field as a characteristic of a reflex, in particular the scratching reflex of the left hind leg, and defines it as the area of the dog's skin from which scratching with this leg can be elicited by itching. Sherrington not only describes the receptive field as an area on the skin but also reports a set of threshold values for different stimulation sites on the skin, a notion akin to the "receptive field profile" to be discussed below. He also noted that the scratching reflex is actually a "reflex group," since it involves neurons in different segments of the spinal cord corresponding to different portions of the skin (dermatomes). This leads to the modern notion of the receptive field as a property of single neurons. Hartline[3] (1938) was the first to record from single neurons in the frog retina in response to visual stimuli and charted

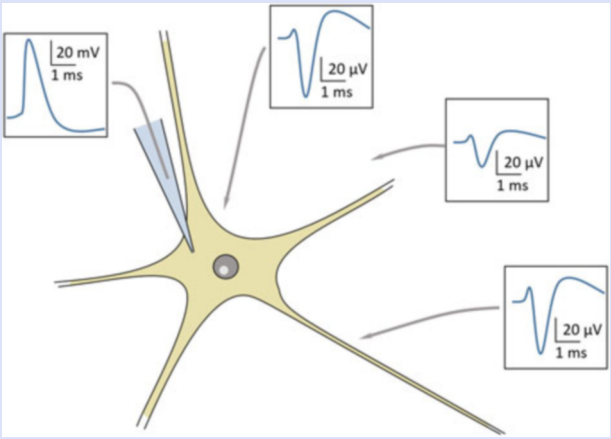
[1] Johannes Peter Müller (1801–1858). German physiologist.

[2] Sir Charles Scott Sherrington (1857–1952). British physiologist. Nobel Prize in physiology or medicine 1932.

[3] Haldan Keffer Hartline (1903–1983). American biologist. Nobel Prize in physiology or medicine 1967.

the position of their receptive fields on the retina. At the same time, he also studied the temporal response characteristics, for example, the time course of responses to the switching on and off of a stimulus. Both aspects, spatial and temporal, will be integrated in the concept of the spatiotemporal receptive field.

Box 2.1 Extracellular Recordings



Action potentials cause ion current around the firing neuron which can be recorded from the outside or "extracellularly." For this, a metal electrode is placed in the vicinity of a cell. The figure shows on the left side an intracellular recording with a "glass electrode" (i.e., a micropipette filled with some electrolyte and a metal wire picking up the current). The recorded action potential has the shape discussed in Chap. 1. The extracellular potentials shown on the right side of the figure show roughly the inverted polarity of the intracellular potentials with the amplitude quickly declining with the distance from the neuron (note scale bars). Amplitude and shape depend also on the exact recording location, i.e., close to the soma, dendritic tree, or axon, and may be further modified by the presence of multiple dendritic branches (Gold et al. 2006). Depending on the input impedance of the electrode, signals may also be recorded from more distant neurons.

The variation in the shape of the extracellular recording allows to distinguish between the signals of multiple cells recorded with the same electrode, even if intracellularly the action potentials would look alike. Statistical methods for identifying individual neurons in multiunit recordings are known as spike sorting (Abeles and Goldstein 1977). If the number of recorded units gets large, which happens if electrodes with low input impedance are used, the total potential is also called the local field potential or LFP.

Figure 2.1 shows a simple measurement protocol for the receptive fields of retinal ganglion cells similar to the one used by Kuffler (1953). An electrode is placed near a ganglion cell (extracellular recording, see Box 2.1) in the retina of a cat, and the visual field is scanned with a small spot of light. As soon as a spike is recorded, a dot is put on a map of the visual field at the position of the scanning stimulus. This procedure allows to measure not only the size and extend of the receptive field but

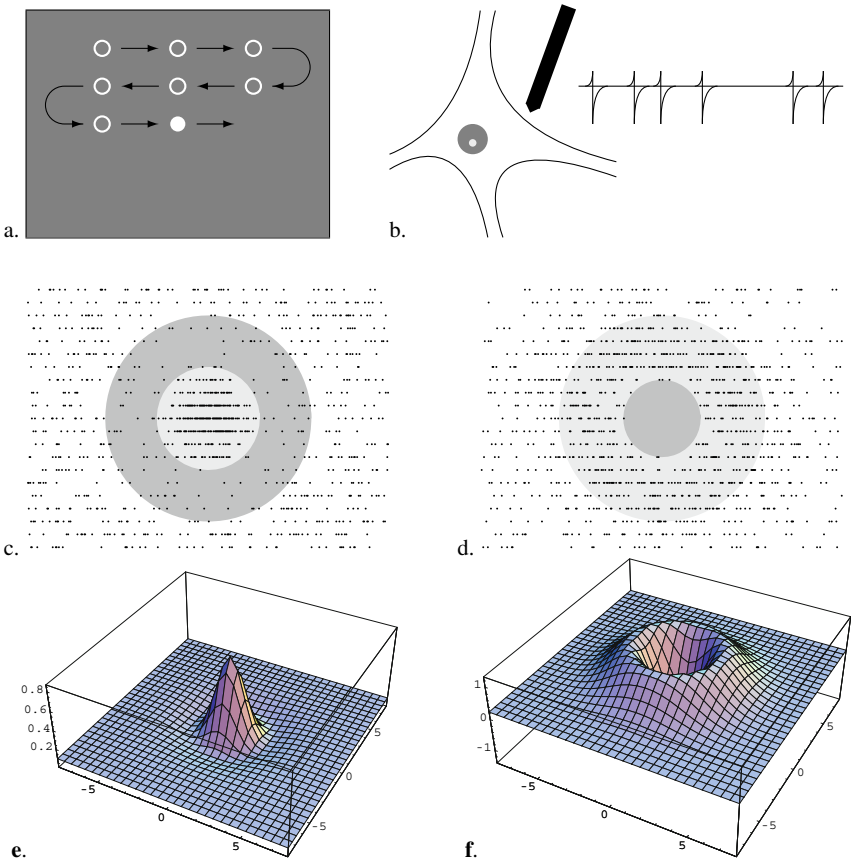


Fig. 2.1 Visual receptive fields defined as an experimental protocol. (**a**) The visual field is scanned line by line with a small spot of light. (**b**) Simultaneously, neural activity is recorded from a retinal ganglion cell. (**c**) and (**d**) Each time the neuron fires, a point is plotted at the location of the light spot at the time of firing. (**c**) Receptive field of a neuron which is active if the light falls within a central disk and is inhibited if the light falls in the outer annulus (ON-center and OFF-surround organization). (**d**) Alternatively, a neuron may be inhibited if the light falls within a central disk and activated if the light falls in the outer annulus (OFF-center and 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**) and (**f**) Center-surround receptive field function $\phi(x, y)$ for the two neurons depicted in figures c and d. The gray areas in figures c and d are delimited by contour lines of ϕ

also its internal structure: As can be seen in Fig. 2.1c, d, different subregions can be defined within the receptive field, from which the neuron may be either activated or inhibited. The special structure shown in the figure is known as center-surround organization, either ON-center, OFF-surround, or vice versa.

Mathematically, the result can be described by a receptive field function which specifies a gain factor for each position in the retina. The gain factor indicates how strongly the neuron can be activated or inhibited from each location; it is negative in inhibitory regions and positive in excitatory regions. Let us denote the excitation of a neuron by the binary variable $e \in \{0, 1\}$, where $e = 1$ means that a spike occurs. We may then represent the measurement described in Fig. 2.1 as the conditional probability of spiking, given that the stimulus was presented at position (x, y). It is called the receptive field function or profile, $\phi(x, y)$:

$$\phi(x, y) = \frac{\mathrm{P}(e = 1|\text{stimulus at position } (x, y)) - p_o}{\text{stimulus strength}}. \tag{2.1}$$

In this formulation, we subtract the probability of spontaneous spiking p_o (i.e., the spike rate in the absence of a stimulus) and normalize the response by the stimulus strength. In our examples so far, the stimulus strength was a constant, since we have been moving an unchanging light spot across the receptive field. Physically, it is the radiant flux arriving at the retina and would be measured in watt (W).[4] In the case of patterned stimuli or images, we will have to consider the radiant flux per retinal area, which is called *irradiance* and measured in $\mathrm{W/m^2}$.

The spiking probability $\mathrm{P}(e = 1)$ in Eq. 2.1 is really a spike rate, i.e., the probability that a spike occurs within a certain time interval; the encoding of information in spike rates is also called rate coding, see Sect. 7.1. We will denote spike rate by the letter a (for activity) from here on; it is measured in spikes per second. Experimentally it can be determined by binning a spike train with a bin width Δt and counting the spikes in each bin. The activity variable $a(t)$ would then be a piecewise constant function that can change its value only at the bin boundaries. Alternatively, one could consider the inter-spike intervals and set $a(t)$ in each interval to the inverse of the interval length. Activity $a(t)$ would then be constant between two subsequent spiking events. In repetitive measurements, where spike times are expressed relative to some trigger, usually the stimulus onset, so-called post- or peristimulus time histograms (PSTHs) are obtained, which give a more reliable estimate of the time course $a(t)$ during or after the stimulus presentation.

[4] The psychophysical analog of radiant flux is luminous flux measured in "lumen" (lm). It takes into account that the sensitivity of the eye varies with wavelength (color). A green monochromatic light with wavelength 555 nm (i.e., at the peak of the luminous efficiency function $V(\lambda)$) and radiant flux of 1 W has luminous flux 683 lm. Other lights perceived as equally bright (i.e., with the same luminous flux) need to have higher radiant flux because the visual system is less sensitive to other wavelengths.

The procedure sketched out in Fig. 2.1 is known as spike-triggered averaging or reverse correlation. It is reverse in the sense that, at least in the sensory pathways, the neural activity is correlated with the stimuli occurring before each spike (Dayan and Abbott 2001; Eggermont et al. 1983). However, similar procedures have also been applied in the motor system (e.g., by Schwartz et al. 1988) and show that neurons in the monkey motor cortex have "motion fields" in the sense that arm movements into a particular portion of grasping space are observed after neuronal firing. Spike-triggered averaging is also used in the recording of hippocampal place cells (O'Keefe and Dostrovsky 1971) which fire with high probability if the rat passes a particular region (the "place field") of its maze (see Chap. 7).

In the remainder of this chapter, we discuss the notion of the receptive field with examples from the visual system, keeping in mind, however, that the theory applies also to other systems. Overlapping receptive fields and tuning curves lead to the concept of population coding which will be discussed in detail in Chap. 7.

► **Key Point: Specificity and the Receptive Field** The informational content of neuronal firing is determined by the specificity of the firing neuron. In the sensory pathways, specificity is operationalized as the neuron's receptive field.

2.2 Linear Shift-Invariant (LSI) Systems

2.2.1 Correlation and Linear Spatial Summation

The Superposition Principle

The reverse correlation procedure discussed in the previous section was based on the scanning of the receptive field with small spots of light. In Eq. 2.1 we were therefore able to represent the stimulus by its position. The receptive field function $\phi(x, y)$ obtained in this way would then tell us how strongly the neuron can be activated or inhibited from each stimulus position.

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, which are elicited by each of the lights in isolation, are added. This scheme is called "linear spatial summation" or "linear superposition." Let us describe the spatial distribution of light impinging on the retina by the image function $I(x, y)$, where (x, y) denotes the retinal coordinates. We treat the retina as a plane, solely for the sake of simplicity, but spherical coordinates can readily be substituted. At every point of the retina, $I(x, y)$ is an irradiance with dimension $\mathrm{W/m^2}$, which may take values between 0 (black) and some upper limit $I_{\max}$. A square light spot with intensity I_o covering the interval from $x = 0$ to $x = 1$ and $y = 0$ to $y = 1$ can be described by the function $I(x, y) = I_o d(x, y)$, where d is the dimensionless function

$$d(x, y) := \begin{cases} 1 & \text{if } 0 \le x < 1 \text{ and } 0 \le y < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

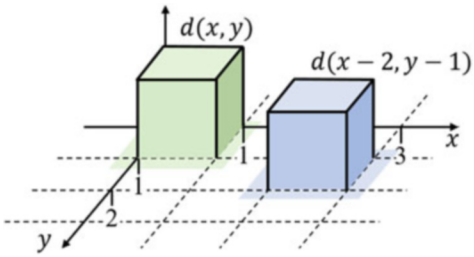


Fig. 2.2 Pixel function as defined in Eq. 2.3. Subtracting the numbers 2 and 1 from the x and y arguments shifts the function in the positive x and y directions. Note that both functions extend to the entire plane, with the functional value 0

see Fig. 2.2. We call this function the *pixel function* because it covers one pixel of our image. The letter d indicates its relation to the δ-function to be introduced below.

Assume now that we shift our light spot to some other location (x_o, y_o). The image function would then be given by

$$I(x, y) = I_o d(x - x_o, y - y_o). \tag{2.3}$$

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 a_1 of the neuron and that a light stimulus delivered at location (x_2, y_2) elicits a response a_2:

$$I(x, y) = I_o d(x - x_1, y - y_1) \;\Rightarrow\; a = a_1 \tag{2.4}$$

$$I(x, y) = I_o d(x - x_2, y - y_2) \;\Rightarrow\; a = a_2. \tag{2.5}$$

As before, a denotes the activation or spike rate 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:

$$I(x, y) = I_o d(x - x_1, y - y_1) + I_o d(x - x_2, y - y_2) \;\Rightarrow\; a = a_1 + a_2. \tag{2.6}$$

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

$$I(x, y) = \lambda I_o d(x - x_1, y - y_1) \;\Rightarrow\; a = \lambda a_1. \tag{2.7}$$

Linear spatial summation is a theoretical concept defined by the above two equations; a system would be called "linear," if it adheres to these conditions. The concept of linearity is useful to describe the behavior of neurons as long as the stimulus intensities are not too large. Clearly, 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, Eq. 2.7 may seem to be meaningless for negative λ. It is meaningful, however, when small signals

are considered on top of stronger base stimuli. For an input $I + \lambda \Delta I$ and the corresponding response $a + \lambda \Delta a$ small negative values of λ make perfect sense. Therefore, even in clear nonlinear cases, as will be discussed in later sections of this chapter, linear descriptions are used as first approximations or as components of more comprehensive models.

The Receptive Field Function

Let us 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

$$I(x, y) \approx \sum_{i=1}^{I} \sum_{j=1}^{J} I_{ij}\, d(x - x_i, y - y_j), \tag{2.8}$$

where I and J are the total numbers of pixels in the x and y directions. Equation 2.8 is an approximation if $I(x, y)$ is not pixelized from the beginning and gets downsampled to $I \times J$ pixels. The pixel size defined by our pixel function d is 1 by 1 but needs to be reduced if better approximations are to be achieved.

Assume now that we have measured the neuron's response to an isolated light spot of size $\Delta x \Delta y$ delivered at each of the pixel locations x_i, y_j and denote the results by a_{ij}. From Eq. 2.1, we obtain[5]

$$\phi(x_i, y_j) = \frac{a_{ij}}{I_{ij} \Delta x \Delta y} \quad \text{or} \quad a_{ij} = \phi(x_i, y_j)\, I_{ij}\, \Delta x \Delta y. \tag{2.9}$$

Here we have assumed that the spontaneous activity (p_o in Eq. 2.1) is zero. Then, by the linear superposition principle, we can construct the response as the sum of all responses to the individual pixels and obtain

$$a = \sum_{i=1}^{I} \sum_{j=1}^{J} a_{ij} = \sum_{i=1}^{I} \sum_{j=1}^{J} \phi(x_i, y_i)\, I_{ij}\, \Delta x \Delta y. \tag{2.10}$$

In order to arrive at a continuous formulation, we now consider a sequence of pixels of size $(\Delta x, \Delta y)$ decreasing to zero. Eventually, if we assume very large numbers of very small pixels, we may write

$$a = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y)\, \phi(x, y)\, dx\, dy = \int_{\mathbb{R}^2} I(\mathbf{x}) \phi(\mathbf{x}) d\mathbf{x}, \tag{2.11}$$

where the integral is taken over the entire visual field.

[5] Here, I_{ij} is an irradiance with the dimension W/m^2. The term $I_{ij} \Delta x \Delta y$ corresponds to the "stimulus strength" in Eq. 2.1, which increases if the stimulus gets larger. Physiologically, this is related to Riccò's law, which states that the contrast threshold for small visual stimuli is inversely proportional to the stimulus area.

Equation 2.11 is called the *correlation* of I with ϕ (or vice versa). The function $\phi(x, y)$ is of course the receptive field function or *response profile* defined above. Equation 2.11 is a so-called improper[6] integral which evaluates to a number ($a \in \mathbb{R}$). It can be thought of as the total signed volume between the surface defined by the two-dimensional function $I(x, y)\phi(x, y)$ and the x, y plane (i.e., portions above the plane are counted positive and portions below negative). The visual field is modeled as the infinite plane $\mathbb{R}^2$ over which the integral is taken. Since the receptive field function can be assumed to be zero outside a finite "support" region (the extend of the receptive field), the existence of the integral is not a problem.[7]

The name "correlation" for Eq. 2.11 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 from the standard definition of correlation the subtraction of the means and the normalization with the standard deviations, Eq. 2.11 indeed describes the statistical correlation between the variables I and ϕ.

Equation 2.11 motivates further names for the receptive field function, including kernel and operator. "Kernel" is a general term for a fixed functional factor (ϕ) in an integral equation where a second factor (I) is considered a variable input or "test" function. An operator is a mapping from a set of input functions (images) to a set of output functions (pattern of neural activity as studied below). Since important classes of operators can be expressed as integral equations with suitable kernels, the term operator is sometimes also used for its kernel (see also Box 2.2).

Box 2.2 Linear Mappings: Functions, Functionals, and Operators

Functions, functionals, and operators are three types of mappings studied in calculus and functional analysis. The simplest case is the *function*, f, which is a mapping between number sets, for example, from the set of real numbers $\mathbb{R}$ into itself. The plot of a linear, one-dimensional function is a straight line passing through the origin; with slope k, we have

$$f : \mathbb{R} \longrightarrow \mathbb{R} \qquad x \mapsto f(x) = kx.$$

Linear functions of two variables describe planes passing through the origin. Writing $(x_1, x_2) = \mathbf{x}$, we can express such functions as

$$f : \mathbb{R}^2 \longrightarrow \mathbb{R} \qquad \mathbf{x} \mapsto f(\mathbf{x}) = (\mathbf{k} \cdot \mathbf{x}) = \sum_{i=1}^{2} k_i x_i.$$

(continued)

[6] An improper integral is the limit of a definite integral with the bounds approaching infinity or some singular point, in our case: $\int_{-\infty}^{\infty} f(x)dx = \lim_{a,b\to\pm\infty} \int_b^a f(x)dx$. It exists if the limit is a number which is independent of the specific paths toward infinity taken by a and b.

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

Box 2.2 (continued)

Here, $\mathbf{k} = (k_1, k_2)$ is the gradient vector describing the direction and the slope of the steepest ascent of the plane; $(\mathbf{k} \cdot \mathbf{x}) = k_1 x_1 + k_2 x_2$ is the dot product.

We may now consider the set of all functions $\mathbb{R}^2 \to \mathbb{R}$, which contains, for example, not only the black and white images $I(x, y)$ but also the receptive field functions $\phi(x, y)$; let us denote it by $\mathbb{F}$. Mappings from the set $\mathbb{F}$ into the real numbers are called *functionals*, $\mathcal{F}$. A linear example is the correlation given in Eq. 2.11, which maps every image to the activity of the neuron with receptive field function ϕ. In the general case, we call the receptive field function a "kernel" $k \in \mathbb{F}$ and $f \in \mathbb{F}$ a test function:

$$\mathcal{F} : \mathbb{F} \to \mathbb{R} \qquad f \mapsto \mathcal{F}(f) = \int_{\mathbb{R}^2} k(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}.$$

Finally, we may consider mappings from the function space into itself as would occur if we relate the image to the entire activity pattern of the retina. Such mappings are called *operators*, O; in the linear case, they are described by kernels taking two vectorial variables:

$$O : \mathbb{F} \to \mathbb{F} \qquad f \mapsto Of; \;\; Of(\mathbf{y}) = \int_{\mathbb{R}^2} k(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) d\mathbf{x}.$$

Here, we have written Of instead of $O(f)$. $Of(\mathbf{y})$ is the value taken by the function Of when evaluated at $\mathbf{y}$. Examples include the convolution operation with kernels of the form $k(\mathbf{x} - \mathbf{y})$ and the Fourier transform with the kernel $k(\mathbf{x}, \mathbf{y}) = \exp\{-i\ (\mathbf{x} \cdot \mathbf{y})\}$.

Note the structural similarity of these equations. The spaces $\mathbb{R}$, $\mathbb{R}^2$, and $\mathbb{F}$ are vector spaces over the "field" of the real numbers, $\mathbb{R}$. In their respective spaces, the first three operations are dot products of the variable with a fixed element of the vector space $(k; \mathbf{k}; k(\cdot))$. The fourth equation can be considered a family of dot products with the vectors $k(\cdot, \mathbf{y})$. The dot product of functions is explained by the general definition of the dot product in a vector space V as a map $V \times V \to \mathbb{R}$ satisfying the following conditions: (i) commutativity: $(a \cdot b) = (b \cdot a)$, (ii) bilinearity or linearity in both components: $(((\lambda a + b) \cdot c) = \lambda(a \cdot c) + (b \cdot c)$ and $(a \cdot (\lambda b + c)) = \lambda(a \cdot b) + (a \cdot c))$, and (iii) positive definiteness: $(a \cdot a) > 0$ for all $a \neq 0$. The fact that linear mappings can be formulated as dot products with a fixed element of the containing space is called the Riesz representation theorem (see also Box 2.5).

The transition from the spatially discrete formulation in Eq. 2.10 to the continuous formulation in Eq. 2.11 can also be applied to Eq. 2.8. In this case, the pixel

function $d(x, y)$ has to be replaced by the so-called δ- or Dirac[8] function 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

$$\int_{-\infty}^{\infty} \delta(x) f(x) dx = f(0) \quad \text{for all functions } f. \tag{2.12}$$

From this definition, it immediately follows that

$$f(x) = \int_{-\infty}^{\infty} \delta(x - x') f(x') dx', \tag{2.13}$$

i.e., correlation with $\delta(x - x')$ "cuts out" the functional value at position x. In the next section, we will introduce the notion of convolution. Equation 2.13 then says that convolution with the δ-pulse does not change the input function f; that is, it is the neutral element of the convolution operation.

Strictly speaking, $\delta(x)$ is not a proper function, because it does not take a finite value at $x = 0$. For mathematically rigorous definitions of δ and other "generalized functions" or "distributions," see textbooks of functional analysis and Box 2.5. For purposes involving sampled data, the "true" δ-function can always be replaced by approximations, i.e., sufficiently narrow versions of the amplified pixel function with integral 1.

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 by a reduction of the spontaneous activity (cf. Fig. 2.1) or by interactions with a second excitatory stimulation. The ON-center, OFF-surround profile shown in Fig. 2.1e is also known as "Mexican hat function." It is usually fitted by a difference of two Gaussian[9] functions, as will be explained in Sect. 3.1.

Visual receptive fields generally do not respond to homogeneous light distributions, i.e., stimuli of the form $I(x, y) \equiv I_o$. In Eq. 2.11, this amounts to the constraint

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi(x, y)\, dy\, dy \approx 0. \tag{2.14}$$

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

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

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

The receptive field function in the correlation operation can also be considered 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 Fig. 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 "Cauchy–Schwarz inequality," stating that the correlation equation is maximized if the two functions involved have the same shape (see Box 5.3). 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." What they signal to the brain is the presence in the neuron's receptive field of a dark dot in front of a bright background: that is, the visual direction toward a putative fly. This information can then be passed on to the motor system to initiate and guide an ejection of the tongue which may result in the catchment of a fly.

► **Key Point: Receptive Field Function** The receptive field function of a visual neuron is the spatial distribution of gain factors linking stimulus strength to neuronal response; it can be measured by scanning the visual field with point stimuli. The correlation operation allows to predict the response to distributed stimuli by linear superposition.

2.2.2 Lateral Inhibition and 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 receptor layer of the retina) projects to an entire layer of neurons (e.g., the layer of retinal ganglion cells), each with its own receptive field. Clearly, this is the case found in many sensory systems, in particular if representations are organized as topological maps (see also Sect. 7.3).

Figure 2.3 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 Fig. 2.3. As a result, a distribution of excitation and inhibition will arise in the output layer which is positive for the neuron directly innervated from the input site and negative for its neighbors which receive the lateral, inhibitory connections. In general, we will denote such distributions as $a(x, y)$, where (x, y) is

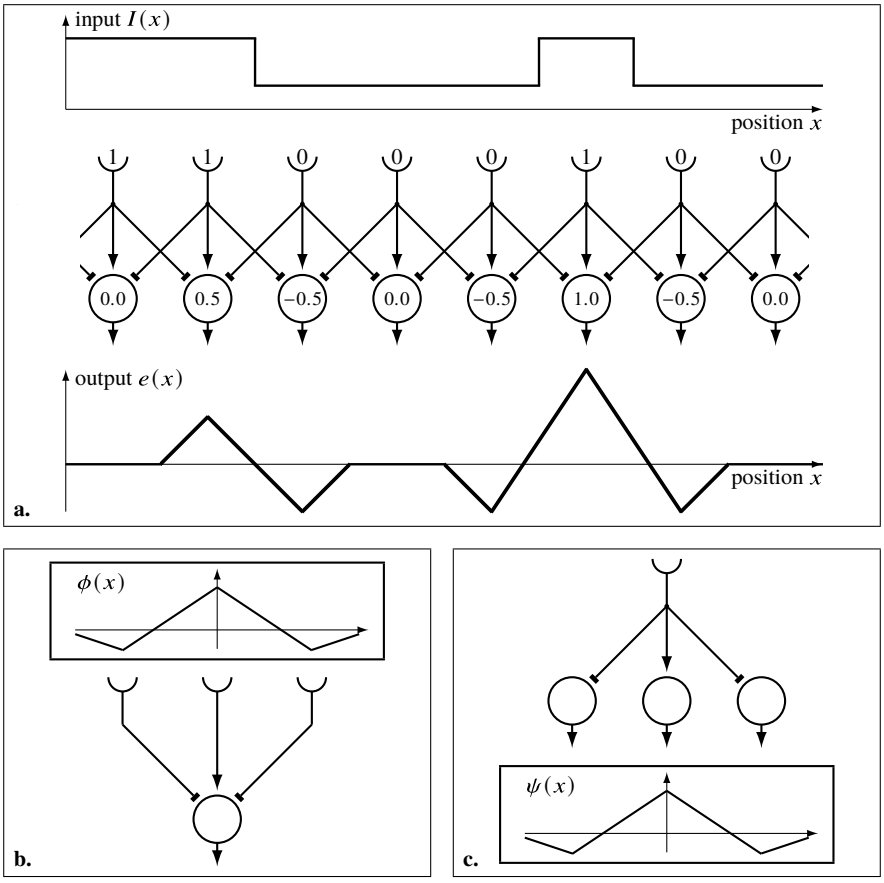


Fig. 2.3 Lateral inhibition. (**a**) Input–output relations for a simple example. Links ending in an arrow are excitatory (weight +1), and links ending in a dash (⊣) symbolize inhibitory connections (weight −0.5). The input shows a step edge (left) and a contour (right) which are enhanced in the output. 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., $\psi(x) = \phi(-x)$

the location of a neuron and $a(x, y)$ its activity. The particular distribution of activity 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 Fig. 2.3, the point-spread

functions for different stimulation sites will be identical up to a shift in position ("shift invariance"). As in the previous section, we may write

$$\begin{aligned} I(x, y) = I_o d(x - x_1, y - y_1) \quad &\Rightarrow \quad a(x, y) = I_o \psi(x - x_1, y - y_1) \\ I(x, y) = I_o d(x - x_2, y - y_2) \quad &\Rightarrow \quad a(x, y) = I_o \psi(x - x_2, y - y_2), \end{aligned} \tag{2.15}$$

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

For general images composed of many point stimuli, we again use the decomposition into pixels given in Eq. 2.8. Assuming linear superposition for all neurons, we may sum the point-spread functions generated by all pixels and obtain

$$a(x, y) = \sum_{i=1}^{I} \sum_{j=1}^{J} I(x_i, y_j) \psi(x - x_i, y - y_j)\, \Delta x \Delta y. \tag{2.16}$$

Here we have added the pixel size $\Delta x \Delta y$ which was assumed to be one in Eq. 2.15. In the infinitesimal formulation, we get

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

Equation 2.17 is known as the *convolution*[10] of I with ψ. We also write $a = I * \psi$ or $a(x, y) = (I * \psi)(x, y)$. It is again an improper integral. A space-variant version of Eq. 2.16 was first introduced into neuroscience as a model of lateral inhibition in the eye of the horseshoe crab (*Limulus*) and is therefore also known as the Hartline[11]–Ratliff equation (Hartline and Ratliff 1958; Ratliff 1965).

Box 2.3 Convolution in Other Contexts

Convolution is a standard operation occurring not only in the theory of lateral inhibition and receptive fields but also in many other fields of scientific computing.

Numerical Analysis For an equidistantly sampled function with values $(x_1, \ldots, x_n)$ and sampling width h, approximate derivatives are calculated as

$$x_i' \approx (x_{i+1} - x_i)/h.$$

(continued)

[10] The German term "Faltung" is sometimes also found in English texts.

[11] See footnote 3.

Box 2.3 (continued)
This can be written as a discrete convolution

$$x_i' \approx x_{i+1}\psi_{-1} + x_i\psi_0 = \sum_{j=-1}^{0} x_{i-j}\psi_j,$$

where (ψ_{-1}, ψ_0) takes the values $(1, -1)/h$. In numerical analysis, ψ would be called a two-point "stencil." Better approximations can be obtained with three- or five-point stencils of the form $(-1, 0, 1)/2h$ or $(1, -8, 0, 8, -1)/12h$, in which case the sum must be taken from $j = -1$ to $+1$ or from $j = -2$ to $+2$, respectively.

Statistics In order to calculate the probability distribution for the sum of two die, we observe that the probability for each ordered number pair (i, j), where i and j are the outcomes for the first and second die, is $\frac{1}{36}$. A given outcome, for example, $i + j = 8$, can be obtained by any of the number pairs $(6, 2)$, $(5, 3)$, $(4, 4)$, $(3, 5)$, or $(2, 6)$. We can write these pairs as $(8 - j, j)$, where $j = 2, \ldots, 6$ is the number on the second die. For the probability of the sum to take value k (between 2 and 12), we thus obtain

$$\mathrm{P}(i + j = k) = \sum_{j=1}^{6} \mathrm{P}(k - j)\mathrm{P}(j) = \frac{6 - |k - 7|}{36}.$$

Here we have assumed that $\mathrm{P}(i) = 0$ for $i < 1$ or $i > 6$. This example illustrates a general rule: For two independent random variables X and Y with densities $p(x)$ and $q(y)$, the density of the sum $Z = X + Y$ is given by $r(z) = \int p(z - y)q(y)dy = \int p(x)q(z - x)dx$, i.e., by the convolution of the density functions of X and Y.

Differential Equations Consider the linear ODE $f'(t) = -kf(t)$. It is called "homogeneous" because all terms contain the unknown variable f or its derivative. The initial value problem $f(0) = f_o$ has the solution $f(t) = f_o e^{-kt}$. If we now add an "inhomogeneity" $h(t)$

$$f'(t) = -kf(t) + h(t),$$

we can calculate the solution from the temporal convolution of h with the homogeneous solution:

$$f(t) = f_o \int_{-\infty}^{t} h(t') \exp\{-k(t - t')\}dt'.$$

The homogeneous solution $\exp\{-kt\}$ would be called a "Green's function."

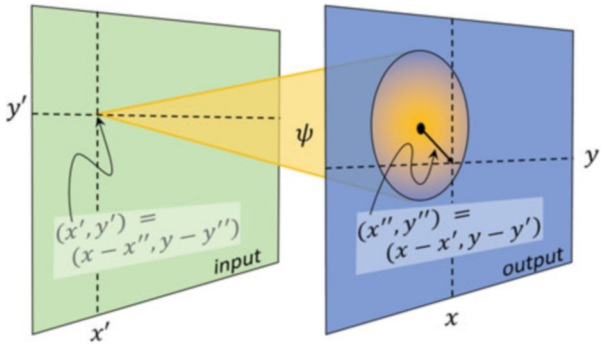


Fig. 2.4 Illustration of the coordinates of convolution as used in Eq. 2.18. (x, y): a point in the output layer, at which the resultant activity $a(x, y)$ is considered. (x', y'): a point in the input layer at which a stimulus is delivered and the point-spread function ψ is activated. (x'', y''): a point within and relative to the domain of the point-spread function specifying the transmission weight $\psi(x'', y'')$. In the first line of Eq. 2.18, the integral is taken over the input layer; for each input point (x', y'), the point-spread function is therefore evaluated at $(x - x', y - y')$. In the second formulation of Eq. 2.18, we start by picking a position (x'', y'') within the domain of the point-spread function and find the connection from the input layer going through this point. The integral is then taken over the domain of the point-spread function. The equivalence of the two formulations amounts to saying that the convolution operation is commutative

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 (see Box 2.3).

The variables in Eq. 2.17 can be interpreted in the following way (see also Fig. 2.4): Coordinates of the output layer (e.g., the retinal ganglion cells) are given by the (x, y) coordinates; they are the argument of the activity function $a(x, y)$. The primed coordinates (x', y') parameterize the input layer, for example, the receptor cell layer of the retina or the projection screen in Fig. 2.1a. For a stimulus delivered at (x', y'), the value $\psi(x - x', y - y')$ gives the strength, or weight, with which it influences the output at (x, y). Mathematically, it is easy to show that convolution is commutative, i.e., that we may write

$$\begin{aligned} a(x, y) = (I * \psi)(x, y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x', y')\, \psi(x - x', y - y')\, dx' dy' \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \psi(x'', y'')\, I(x - x'', y - y'')\, dx'' dy'' = (\psi * I)(x, y). \end{aligned} \tag{2.18}$$

This result is achieved by the substitution $x - x' =: x''$ and $y - y' =: y''$. With the (x'', y'') coordinates, 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 output site and the integral is taken over these offsets. The input function I has to be evaluated at $(x - x'', y - y'')$, i.e., the output site minus the offset. In the end, this is just a difference in the parameterization of the lateral 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 is but one example of a neural point-spread function realized by neural wiring. Besides the center-surround type illustrated in Fig. 2.1, other functions exist which will be discussed in Chap. 3.

► **Key Point: Convolution and Lateral Inhibition** A layer of linear neurons with identical receptive fields, shifted to be centered on each neuron's position, can be described by the convolution operation. Divergent and convergent connectivity is described by the receptive field and point-spread functions, respectively. A simple example is lateral inhibition.

2.2.3 A Formulation with a Differential Operator

Lateral inhibition was first studied by Ernst Mach[12] in relation to a perceptual phenomenon now known as Mach bands (Mach 1865; Ratliff 1965). Mach observed that continuous intensity gradients in images are perceived as roughly homogeneous areas, whereas boundaries between constant intensity areas and intensity ramps appear as bright or dark bands, depending on the direction of the kink between the plateau and the ramp (Fig. 2.5). Mach observed that the bands gain more contrast as the kinks get sharper, just as if the perceived contrast depended on the second derivative of intensity with respect to space. He therefore suggested that the retina calculates local second derivatives with respect to space, resulting in an excitation pattern of the form

$$a(x, y) = c \left(\frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \right) = c \Delta I(x, y), \tag{2.19}$$

where c is some unknown constant and Δ is an abbreviation for taking the sum of the second partial derivatives; it is known as the Laplace[13] operator. $\Delta I(x, y)$ is called the Laplacian of the function $I(x, y)$. Equation 2.19 is also known as Poisson's equation and is closely related to the convolution with a center-surround system. Indeed, the network depicted in Fig. 2.3 can be considered as calculating a spatially discretized approximation of a (one-dimensional) second derivative. If the intensity

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

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

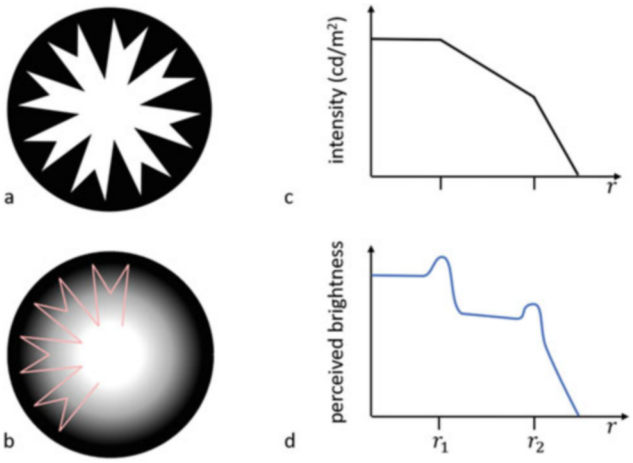


Fig. 2.5 Mach bands. (**a**) Patterned disk as used by Ernst Mach. (**b**) As the disk is quickly spinning, it will generate a homogeneous intensity profile. The red lines mark the black–white boundary in the resting disk. (**c**) Section along the intensity profile from b, taken along a radius. It shows a white area in the center, a ring of linearly decreasing intensity between r_1 and r_2, and a ring with steeper decrease of intensity outside of r_2. (**d**) Perceived brightness as it appears to the observer. At the kinks of the intensity profile (positions r_1 and r_2) bright rings are seen, which are now known as Mach bands. These bands get brighter, if the kink is sharper. The intensity decrease between r_1 and r_2 is barely perceived. Adapted from Mach (1865)

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 $I_2' = I_3 - I_2$ (using the two-point stencil with $h = 1$ in Box 2.3a). Using the same stencil again to calculate the second derivative, we obtain

$$I_1'' = I_2' - I_1' = I_3 - 2I_2 + I_1, \tag{2.20}$$

which is exactly the operation of Fig. 2.3 up to a factor $-1/2$. In the sense of Box 2.3a, it can be written as a three-point stencil $(1, -2, 1)$. The analogy between the formulations of lateral inhibition by the convolution integral or the partial differential equation (2.19) is reflected by modeling center-surround receptive fields as Laplacians of Gaussians (see Sect. 3.1.1).

2.2.4 Correlation and Convolution

Let us now look at the receptive fields of the output neurons in the lateral inhibition network as shown in Fig. 2.3a,b. 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 Eq. 2.11 to layers. We recall that $\phi(x', y')$ was the receptive field function of the sole neuron considered in

the derivation of Eq. 2.11. Let us now assume that this neuron is located at position $(x, y) = (0, 0)$ in the output layer. In shift-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 Eq. 2.11 to layered systems:

$$a(x, y) = \iint I(x', y')\phi(x' - x, y' - y)\, dx'dy'. \tag{2.21}$$

By comparing Eq. 2.17 with Eq. 2.21 for all possible images I, we find that

$$\phi(x', y') = \psi(-x', -y'), \tag{2.22}$$

i.e., the point-spread function and the receptive field function are mirrored versions of each other. Note that this result holds only for shift-invariant systems. In general, 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. Fig. 2.3b,c).

Box 2.4 Chaining LSI-Operations

Convolution ($*$) and correlation ($\otimes$) reflect the divergence and convergence of neural connectivity and are therefore two sides of the same coin; either operation is suited to describe a linear shift-invariant (LSI) system. The two formulations differ, however, in two respects that make convolution generally more convenient:

First, convolution is commutative, while correlation is not:

$$(f * g)(\mathbf{x}) = (g * f)(\mathbf{x}) \quad \text{while} \quad (f \otimes g)(\mathbf{x}) \neq (g \otimes f)(\mathbf{x})$$

Instead of commutativity, correlation satisfies the relation $(f \otimes g)(\mathbf{x}) = (g \otimes f)(-\mathbf{x})$.

Second, convolution satisfies the associative law, whereas correlation does not:

$$(f * g) * h = f * (g * h) \quad \text{while} \quad (f \otimes g) \otimes h \neq f \otimes (g \otimes h).$$

Instead, we have $(f \otimes g) \otimes h = f \otimes (g * h)$ and $f \otimes (g \otimes h) = (f \otimes g) * h$. The relations are best understood in the light of the Fourier convolution and correlation theorems, Eqs. 4.71 and 4.74; see also Borsellino and Poggio (1973).

As a consequence, a chain of n LSI operations with point-spread functions $\psi_1, \ldots, \psi_n$ can be described by a single "master" point-spread function

(continued)

Box 2.4 (continued)

$$\Psi = \psi_1 * \ldots * \psi_n$$

while the analogous "master" receptive field function requires the observation of nested brackets. For example, with $n = 5$, we have

$$\Phi = (((\phi_1 \otimes \phi_2) \otimes \phi_3) \otimes \phi_4) \otimes \phi_5.$$

If shift 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 2.6 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 Fig. 2.6b). Vice versa, if an input neuron projects only to a few output neurons, point-spread functions will be small and receptive fields large. Similarly, neurons in large areas of the visual cortex such as V1 tend to have small receptive fields and large point-spread functions, whereas neurons in small areas such as the medial temporal area MT tend to have large receptive fields and small point-spread functions.

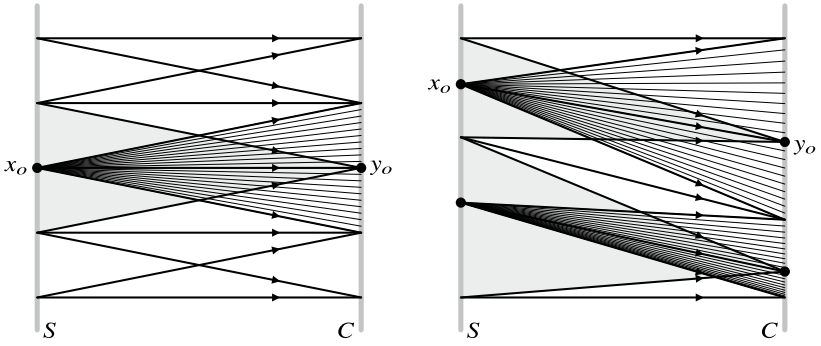


Fig. 2.6 Receptive fields (shaded) and point images (hatched) in space-invariant (left) and space-variant (right) projections between a sensory surface S and a central representation C. The receptive field of a unit $y_o \in C$ is a part of the sensory surface. The point image of a point $x_o \in S$ 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

2.2.5 Convolution and Linear Shift-Invariant (LSI) Systems

We can now formulate the notion of a linear shift-invariant system and its relation to convolution in a mathematically proper way:

Consider a set of functions containing the input images $I(x, y)$ and the activity patterns $a(x, y)$ of a neural layer. We call this set $\mathbb{F}$. Let $O : \mathbb{F} \to \mathbb{F}$ denote an operator mapping input distributions to output distributions, i.e., $(OI)(x, y) = a(x, y)$. The mapping O is called a linear system, if it satisfies the following two conditions:

$$\text{additivity:} \qquad O(I_1 + I_2) = OI_1 + OI_2 = a_1 + a_2 \tag{2.23}$$

$$\text{homogeneity :} \qquad O(\lambda I) = \lambda OI = \lambda a \text{ for all } \lambda \in \mathbb{R}. \tag{2.24}$$

Next, we consider the shift operator $\mathcal{S}$ mapping a function $f \in \mathbb{F}$ to its shifted version $(\mathcal{S}f)(\mathbf{x}) = f(\mathbf{x} - \mathbf{s})$ for some fixed $\mathbf{s}$; it is easy to see that the shift operator is itself linear. An operator O is called shift-invariant, if it satisfies:

$$\text{shift invariance:} \qquad O(\mathcal{S}(I)) = \mathcal{S}(O(I)) = \mathcal{S}(a). \tag{2.25}$$

that is, if the concatenation of O and $\mathcal{S}$ is commutative.

Linearity and shift invariance are independent concepts: an operator may be shift-invariant but nonlinear, or shift-variant and linear. Examples for the former combination are the cascades of LSI-systems followed by the point nonlinearity as discussed below. An example for linear but shift-variant operation would be a slide projector with lens distortions, such that shifting the slide produces an output image that is both shifted and distorted. Another example for a linear operation not satisfying shift invariance is the Fourier transform to be discussed in Chap. 4.

With our derivation of the convolution operation, we have already shown that all convolution systems are LSI. The reverse statement is also true: All LSI systems can be written as convolutions, at least if we allow for point-spread functions such as the δ-function, which are called "generalized functions" or distributions, see Box 2.5.

► **Key Point: Linear Shift-Invariant Systems** A system is linear and shift-invariant if and only if it can be written as a convolution with some kernel ψ, which may be a proper or a generalized function.

Box 2.5 The δ "Function" and its Relatives

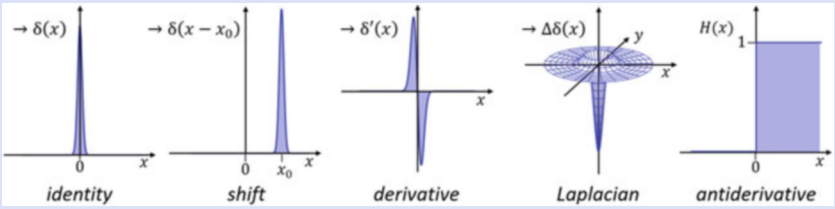
Linear shift-invariant (LSI) systems can always be expressed as convolutions with a suitable kernel, at least if we allow for generalized functions such as the delta function (Riesz representation theorem). Some LSI operations requiring generalized functions as kernels are listed here.

(continued)

Box 2.5 (continued)

identity	$f \mapsto g_i = f;$	$g_i(x) = f(x)$
shift operator	$f \mapsto g_s = \mathcal{S}f;$	$g_s(x) = f(x - x_o)$
differentiation	$f \mapsto g_d = \mathcal{D}f;$	$g_d(x) = f'(x)$
Laplace operator	$f \mapsto g_l = \Delta f;$	$g_l(x, y) = \frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y)$

Approximately, they can be realized as convolutions with proper functions such as our pixel function (Eq. 2.2) or smoothed versions thereof, its derivatives and Laplacians:



In the limit case, when the kernels are replaced by the generalized functions, the convolutions read:

identity	$g_i(x) = \int \delta(x - x') f(x') dx'$
shift operator	$g_s(x) = \int \delta(x - x_o - x') f(x') dx'$
differentiation	$g_d(x) = \int \delta'(x - x') f(x') dx'$
Laplace operator	$g_l(x, y) = \iint \Delta\delta(x - x', y - y') f(x', y') dx' dy'$

The equation for the identity operation was already given in Eq. 2.13. Shift can be written as convolution with a shifted δ, derivation as convolution with δ', and so forth. Also shown in the figure is the antiderivative, $f \mapsto g_a$ with $g_a(x) = \int_{-\infty}^{x} f(x')dx'$ for which the kernel is an ordinary function, namely the antiderivative of the δ function, which is identical to the Heaviside function given in Eq. 2.27.

$$H(x) = \int_{-\infty}^{x} \delta(x')dx' \quad \text{and} \quad g_a(x) = \int_{-\infty}^{\infty} H(x - x') f(x')dx'.$$

2.2.6 Temporal and Spatiotemporal 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 spatiotemporal stimuli, i.e., image sequences or movies. Such stimuli specify an intensity value for each image point and each instant in time and may therefore be represented by three-dimensional functions $I(x, y, t)$. 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 in Eq. 2.8, 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 of recording from the neuron and t' the time that passed since a given light flash was presented. Clearly, this event took place at time $t - t'$.

Figure 2.7a shows the situation for a single pixel or small spatial spot within which a temporal signal $I(t)$ is delivered. The response to a single light flash is called the *impulse response* and is analogous to the point-spread function used in the spatial domain; it is denoted as $g(t)$. As for the spatial case, we assume linear superposition (i.e., the additivity of individual responses) as well as shift invariance, which in the temporal domain means that the responses to impulses delivered at

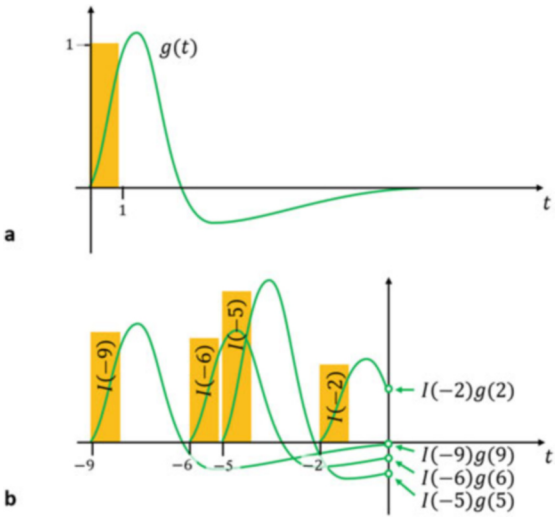


Fig. 2.7 Convolution in time. (**a**) A brief light flash of unit height (yellow rectangle) elicits an impulse response $g(t)$, in the example a biphasic response of activation and subsequent inhibition. (**b**) A general temporal signal $I(t)$ can be decomposed into a sequence of flashes or "pulses" of varying amplitude. Each pulse elicits an impulse response (green lines) scaled with the pulse amplitude. At time $t = 0$, a response that started at time $-t'$ will persist with strength $I(-t')g(t')$. The eventual output of the system is obtained by summing the contributions from all inputs t', i.e., the integral given in Eq. 2.26 for $t = 0$

different times are equal up to the temporal shift. In Fig. 2.7b, shift invariance means that the green curves are shifted versions of each other, up to the vertical scaling which reflects the intensity of the light flash that elicited the response. In this situation, the response to a general temporal input function is given by the temporal convolution

$$a(t) = \int_0^\infty I(t - t')g(t')dt', \tag{2.26}$$

where t' is the time elapsed between the stimulus and the time when the output is measured. Note that the integral in Eq. 2.26 is taken from 0 to ∞, not from $-\infty$ to ∞. The reason for this that negative values of t' would correspond to stimuli delivered only in the future, which are unknown and obviously cannot affect the output. This choice of the integration boundaries can be relaxed, if we assume that $g(t) = 0$ for all $t < 0$. Impulse responses satisfying this condition are called "causal."

Experimentally, the impulse response can be recorded as the peristimulus time histogram discussed in Sect. 2.1. Since the stimulus is just a brief temporal flash, it is also called the poststimulus time histogram in this case.

In practical applications, it is often convenient to describe dynamic behavior not by impulse responses but by step responses. For example, a stimulus may be switched on at time $t = 0$ and then maintained at a constant level for some extended period of time. This is of course more practical than delivering a very short, very strong stimulus approximating a δ-pulse. The relation between the impulse response and the step response is again given by convolution: Let us describe the step stimulus by the so-called Heaviside[14]-function

$$H(t) = \begin{cases} 1 \text{ for } t \geq 0 \\ 0 \text{ for } t < 0 \end{cases}; \tag{2.27}$$

it can be considered the antiderivative of the δ-pulse:

$$H(t) = \int_{-\infty}^t \delta(t')dt'. \tag{2.28}$$

From Eq. 2.26 we then obtain the step response $s(t)$,

$$s(t) = \int_0^\infty H(t - t')g(t')dt' = \int_0^t g(t')dt'. \tag{2.29}$$

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

Thus, the step response is simply the (indefinite[15]) integral or "antiderivative" of the impulse response, just as the step function is the antiderivative of the δ-pulse. If we switch off the stimulus after some duration Δt, the stimulus will be $H(t)-H(t-\Delta t)$ and the response will be $s(t) - s(t - \Delta t)$, due to linearity and shift invariance.[16]

The notion of shift invariance in time is completely analogous to shift invariance in space. It is sometimes also called "time invariance" (as in "linear time-invariant" or LTI systems). Keep in mind, however, that the system is still time dependent, but that this dependence itself does not change over time. This property is also described as being "stationary," as opposed to "static." Stationarity is violated if the impulse response changes over time, for example by adaptation or neuronal fatigue but also by more long-term processes such as growth, plasticity, learning, or aging.

Equation 2.26 is formulated for just one spatial position at which temporally changing stimuli are delivered. In general, when the receptive field covers larger areas and is structured into excitatory and inhibitory subfields, a spatiotemporal weighting function $w(x, y, t')$ needs to be considered. For a fixed delay t' between stimulation and recording, it is a receptive field function just as $\phi(x, y)$ in Eq. 2.11, which will, however change with elapsed time t'. For examples of spatial receptive field functions changing with the time after stimulus presentation, see DeAngelis et al. (1995) and Fig. 3.3 in the next chapter. Alternatively, the three-dimensional kernel can be thought of as an impulse response for every stimulation site within the receptive field. For the activity of a single neuron, the spatiotemporal summation becomes

$$e(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{o}^{\infty} I(x, y, t - t')\, w(x, y, t')\, dt'\, dy\, dx. \tag{2.30}$$

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

Note that space and time are treated differently in Eq. 2.30. 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.

An intuition of spatiotemporal convolution can be obtained from the following example: Consider a pond with a mirror smooth water surface. If we drop a pebble into the pond, it will cause ripples forming a spatiotemporal pattern on the surface. If we drop two pebbles, the two ripple patterns will superimpose and this superposition would be roughly linear as long as the pebbles are not too large. The pebble dropping is of course the equivalent of a spatiotemporal impulse and the circular ripple pattern

[15] An indefinite integral is taken over a variable interval; here, it is expressed as a function of the upper limit while the lower limit is assumed constant. It is also called the antiderivative because, by the fundamental theorem of calculus, $f(x) = \int_0^x f'(x')dx' + f(0)$.

[16] ON and OFF step responses have also been used in the voltage clamp experiments discussed in Chap. 1.

is the kernel $w(x, y, t)$. If we now drop many pebbles at various times, the overall surface pattern will become the convolution of the pattern of pebbles dropped and the ripples caused by each individual pebble.

An important special case of Eq. 2.30 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 spatiotemporal receptive field function can be split up into two components, a spatial profile $\phi(x, y)$ and a temporal weighting function $g(t)$:

$$w(x, y, t) = \phi(x, y)g(t). \tag{2.31}$$

Spatiotemporal kernels that can be split up in this way into a spatial and a temporal factor are called "separable." Separability is violated if different parts of the receptive field have different dynamics, for example if inhibition is slower than activation.

Spatiotemporal separability implies that the profile $w(x, y, t)$ will be zero whenever $g(t) = 0$ or $\phi(x, y) = 0$. When displayed as a space-time density plot (e.g., Fig. 3.3), zeros of the profile must therefore appear as horizontal or vertical lines.

▶ **Key Point: Spatiotemporal Systems** Linear shift-invariant systems in time are described by convolution with the impulse-response function. It differs from the spatial point-spread function by its "causality," i.e., the fact that no response can occur before the stimulus is delivered. Space and time combine in spatiotemporal kernels and receptive fields, which are functions of three variables: x, y, and t.

2.3 Nonlinearities in Receptive Fields

Linear neurons are a theoretical abstraction. They are special in that they are completely defined by one spatiotemporal receptive field function which predicts the neuron's response to all possible stimuli by means of correlation. In contrast, nonlinearity is not a well-defined concept in itself, but simply the absence of linearity, which can occur in many ways.

A formal definition of a linear system (function, mapping) was given in the previous section in terms of the requirements of additivity and homogeneity (Eqs. 2.23 and 2.24). Additivity states that the response to a sum input should be the sum of the individual responses while homogeneity requires that a scaled input (even with a negative scaling factor) should lead to the accordingly scaled output. Both requirements are satisfied in the convolution and correlation systems discussed

above. They cannot, however, hold true in the nervous system for the following reasons:

1. The range of neural activity is limited to the interval between 0 and some 100 spikes per second. Added or amplified stimuli will therefore drive the neuron into saturation, in which case additional stimuli will not lead to stronger answers.
2. Neurons have firing thresholds below which they do not respond at all. Strongly attenuated stimuli will therefore elicit zero response, not the scaled response to the original input.
3. There are no negative spike rates, so homogeneity with negative factors cannot obtain.

The points listed above apply mostly to spikes and spike rates, not so much to dendritic summation, where scaling and even negative signals (hyperpolarization) do occur. This leads to the cascade model of linear summation followed by the point nonlinearity as discussed below.

Recall that linearity and shift invariance are independent concepts. A system may be linear but shift-variant or nonlinear but shift-invariant. In either case, however, it cannot be described by convolution alone. Keep in mind that all linear, shift-invariant systems are convolution systems.

2.3.1 Point Nonlinearity

The simplest examples of nonlinear response characteristics include thresholding, half-wave rectification, saturation, sigmoidal compression, etc. They occur in "cascaded" systems, i.e., systems composed of a linear part followed by a so-called point nonlinearity. In neural networks, neurons are often considered linear–nonlinear (L–NL) cascades, where the (approximately) linear part is dendritic summation, whereas spike generation at the axon hillock is nonlinear, depending only on the instantaneous value of the generator potential $u(t)$. While potential can be below or above resting potential (hyper- vs. depolarized soma), spike rates cannot be negative. The point nonlinearity will therefore be a simple function $f : \mathbb{R} \to [0, a_{\max}]$ mapping the intracellular potential u (a real number) to the spike rate a (a nonnegative real number not exceeding $a_{\max}$). In summary, the cascade system can be written as

$$\begin{aligned} u(t) &= \iiint I(x, y, t - t')\, w(x, y, t')\, dx\, dy\, dt' \\ a(t) &= f(u(t)) \end{aligned} \tag{2.32}$$

(cf. Fig. 2.8). Note that f acts instantaneously on the generator potential u as its single input and does not depend on previous values of u or details of the input

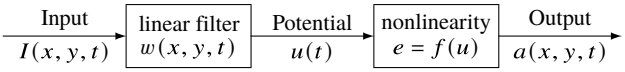


Fig. 2.8 An important class of nonlinear system can be described as a cascade of a linear system, followed by the point nonlinearity. u is the "generator potential" at the axon hillock. In neuroscience, L–NL cascades are used as approximate models of dendritic summation followed by spike initiation at the axon hillock

$I(x, y, t)$ that led to the current value of the potential u. This is what is meant by the terms "point" or "static" nonlinearity.

Typical examples of static nonlinearities are shown in Fig. 2.9. Half-wave rectification simply sets negative values to zero and leaves positive ones unchanged:

$$f_1(u) := \max(0, u) = \begin{cases} 0 \text{ if } u \leq 0 \\ u \text{ if } u > 0, \end{cases} \tag{2.33}$$

and it appears in Fig. 2.9b. This nonlinearity is also known as rectifying linear unit or "relu." In Sect. 1.3.1, we have seen that it is approximately realized by the integrate-and-fire model of the neuron.

The binary switching function (Heaviside function) is defined as

$$f_2(u) := \begin{cases} 0 \text{ if } u < 0 \\ 1 \text{ if } u \geq 0. \end{cases} \tag{2.34}$$

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. In the same way, thresholds can be added to any of the static nonlinearities discussed in this section. 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 nonzero sections.

Saturation is often modeled by the equation

$$f_3(u) := \begin{cases} 0 & \text{if } u \leq 0 \\ u/(u + u_o) & \text{if } u > 0 \end{cases} \tag{2.35}$$

(Fig. 2.9d). The semisaturation constant $u_o > 0$ determines the slope of the function; it can be thought of as the input value for which f_3 generates the output value 0.5. Equation 2.35 has been used to model the nonlinearity of photoreceptors (Naka and Rushton 1966). In this case, u_o can be used to adjust the curve to the state of light adaptation.

The static nonlinearity used most frequently in neural network modeling is the sigmoidal (Fig. 2.9e). It can be formalized in various ways, for example,

$$f_4(u) = \frac{e^{\lambda u}}{e^{\lambda u} + e^{-\lambda u}}, \quad \lambda > 0. \tag{2.36}$$

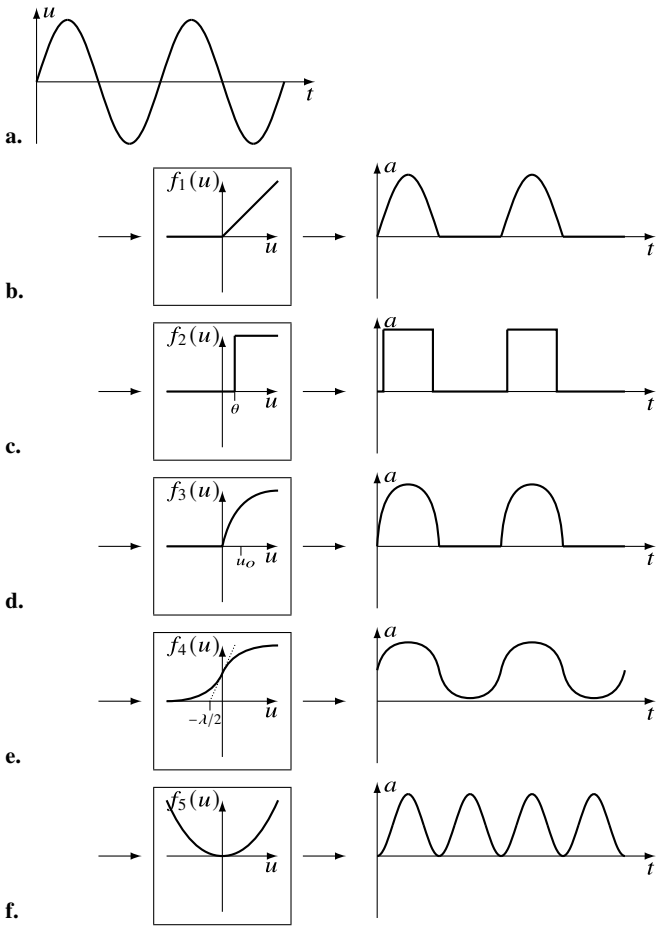


Fig. 2.9 Static nonlinearity applied to a sinusoidal function. (**a**) Sinusoidal used as an input $u(t)$ in all cases. (**b**) Half-wave rectification. The boxed plot shows the static nonlinearity function $f(u)$ and the right part the effect on the sinusoidal input, i.e., $f(\sin u)$. Half-wave rectification 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 nonlinearity 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 Sect. 3.3)

The parameter λ is the slope of the sigmoidal at $u = 0$ shown as a dashed line in the figure. 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.

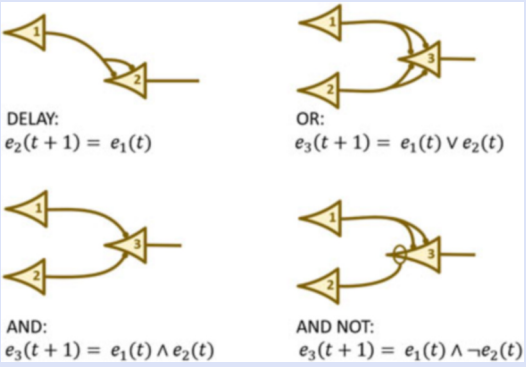
At first glance, the squaring nonlinearity

$$f_5(u) = u^2 \tag{2.37}$$

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 in the next chapter. 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.

Box 2.6 Logical Neurons

The threshold function, Eq. 2.34, discretizes the neural activity to the values 0 or 1. If we also discretize the synaptic transmission weights to the values -1, 0, or 1 and introduce discrete time steps, we arrive at the so-called logical neuron model of McCulloch and Pitts (1943).



Networks of logical neurons can implement simple logical operations, or "gates," such as the AND and OR operations shown in the figure (redrawn from the original paper). The threshold is assumed to be 1, such that two inputs are needed to drive a neuron; the loop around the tip of the output neuron in the AND NOT network symbolizes inhibition, i.e., a synaptic weight of -1.

McCulloch and Pitts (1943) show that networks of logical neurons can implement the central processing unit (CPU) of a Turing machine. This implies that neural networks can solve all problems that an ordinary computer can solve, provided there is sufficient storage capacity that can serve as the Turing machine's "tape."

(continued)

Box 2.6 (continued)
The logical neuron model has not received much attention in neuroscience but was important in machine theory and logic design. The notion of the finite automaton and the implementation of the instructions of a Turing machine by logical gates, be they built of neurons or integrated circuits, go back to this work (see Piccinini 2004). In comparison to real neurons, the model appears overly simplistic. Indeed, there is little experimental support for the idea that neural networks implement logical predicates. More importantly, a discrete time frame with synchronous steps does not exist in the brain. Rather, inter-spike intervals may take arbitrary values in continuous time. The brain is therefore not a finite automaton but may assume an infinite number of states or spatiotemporal activity patterns.

2.3.2 Nonlinearity as Interaction

Volterra Kernels

In convolution, a stimulus is thought of as a composition of small, instantaneous stimuli ("pixels" and "pulses"), each of which elicits a response described by the spatiotemporal weighting function w. By the superposition principle, the total response is the sum of the effects of all individual pulses.

One interesting type of nonlinearity extends the superposition principle to the interactions between pairs of pulses. Each pair of pulses occurring at different points and instants generates an interactive response, and these responses are assumed to 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 Fig. 3.9a in the next chapter. 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

$$a(t) = \iint K(t - t', t - t'')I(t')I(t'')dt'dt''. \tag{2.38}$$

The function K in Eq. 2.38 is called a Volterra[17] kernel of second order or "two-way interaction kernel"; it describes a general multiplicative nonlinearity. For example, if K equals the Dirac impulse $\delta(t' - t'')$, a becomes the integral over the squared input function, i.e., the total signal power presented. Higher order interactions can be modeled analogously, using an n-dimensional kernel K and the

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

product of the stimulus values at n different times (n-way interactions). The first-order Volterra equation would be (linear) convolution.

Sums of Volterra integrals of increasing order are called Volterra series. They can be used to approximate general nonlinear systems in much the same way as a Taylor series approximates a nonlinear function (Schetzen 1980). This approach has in fact been used to fit neurophysiological data from complex receptive fields (see, e.g., Rapela et al. 2006). The advantage of this method is that it gives a general means for the identification of nonlinearities. The disadvantage lies in its huge number of unknown variables. For a spatiotemporal Volterra kernel of order n, a function of $3n$ variables has to be measured.

Gain Control

The activation of a neuron may also be defined in relation to the potentials of other neurons in a group $(u_1, \ldots, u_n)$. Its output would then be kept constant even if the overall input level to the group is varying. This may be useful in the visual system in the case of changing illumination conditions, say, when the available dynamic range of the neuron is to be optimally used in all cases. The mechanism is called gain control and can be achieved by divisive normalization (Heeger 1992):

$$f_i(u_1, \ldots, u_n) = \frac{u_i}{u_o + \sum_{j=1}^{n} u_j} \quad \text{for } u_i \geq 0. \tag{2.39}$$

Divisive normalization is reminiscent of the saturation nonlinearity defined in Eq. 2.35. As before, u_o is a semisaturation constant. If the sum of all inputs equals u_o, the normalization will halve the output. It differs from simple saturation in that many inputs are considered.

Two other types of relational nonlinearities are the maximum and the winner-take-all operations. The maximum nonlinearity can be written as

$$f(u_1, \ldots u_n) = \max_i u_i. \tag{2.40}$$

The winner-take-all operation acts simultaneously on a group of inputs, keeping the strongest one and setting all others to zero. These operations play important roles in artificial neural networks and machine learning, see Sect. 6.3.3.

► **Key Point: Nonlinearity in Neural Systems** Nonlinearities in neural systems arise mainly from thresholding and spike-rate saturation; they are modeled as point nonlinearities applied after linear spatiotemporal summation. Additional types of nonlinearity include multipoint interactions (Volterra kernels) and relational nonlinearities such as gain control, maximum, or winner-take-all.

2.4 Summary and Further Reading

The specificity of neuronal firing in the sensory systems of the brain is described by the receptive field of each neuron. The underlying measurement protocol is reverse correlation or, more generally, the time-dependent correlation of neuronal firing with the respective stimulus parameter. The theory therefore applies not just to vision, where the relevant stimulus parameter is visual field position, but, with the appropriate changes, also to other sensory modalities, multimodal integration, or the motor pathways. The main points are:

1. Receptive fields are described by a function or profile specifying a gain factor for each point on a sensory surface (retina, skin, and basilar membrane). If linearity can be assumed, the overall response of the neuron is the sum of stimuli (conceived of as points on the sensory surface), multiplied with the respective gain factor. This operation is called correlation.
2. If multiple neurons are arranged in a layer and receive orderly input from the sensory surface, the receptive fields describe the convergence of input on each neuron. When looking from a stimulation site, the same connectivity leads to a divergence of activity which is described as the point-spread function. The relation of receptive field and point-spread function was discussed for an important example, i.e., lateral inhibition.
3. If the point-spread functions for different stimulation sites are identical up to the shift between the two sites, the system is called shift-invariant. Linear shift-invariant systems are described by the convolution operation. The point-spread function is simply the mirror image of the receptive field function.
4. Neuronal response will also depend on time, and this dependence is again described as a convolution. Temporal convolution kernels are called impulse responses. In the full spatiotemporal case, the convolution kernel is a three-dimensional function depending on two spatial coordinates and time.
5. Nonlinearity generated by thresholding or saturation is modeled by point nonlinearities applied subsequently to spatial summation (L–NL cascade). Other nonlinear operations are multipoint interaction (Volterra kernels) or gain control.

Overall, the theory presented in this chapter is an application of the theory of linear functionals and operators in functional analysis and of signal theory as a field of scientific computing.

Texts

Dayan and Abbott (2001): *Chap. 2 discusses reverse correlation and linear and nonlinear components of the receptive field.*

Oppenheim et al. (1997): *Standard text on signal processing including a chapter on linear time-invariant systems and convolution. In-depth discussion of convolution systems in the purely temporal case.*

Gonzalez and Wood (2018): *Chap. 3 discusses discrete spatial convolution as a tool in image processing.*

Rudin (1991): *This is a classical textbook of functional analysis covering the mathematical foundations of convolution and other mappings between functions; it requires some mathematical background.*

Suggested Original Papers for Classroom Seminars

Fantana et al. (2008): *This paper explores the role of lateral inhibition in the olfactory system, more specifically among mitral cells, which are the principle cell type in the glomeruli of the olfactory bulb. In contrast to lateral inhibition in the retina, connectivity is sparse and shift invariance does not obtain.*

Mach (1865): *This century-old paper is still recommended reading for its clear and clever argument and for its general discussion of psychophysics and the relation between neural mechanisms and perceptual experiences. It introduces the effect now known as Mach bands and its explanation as lateral inhibition.*

Srinivasan et al. (1982): *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 autocorrelation function of the visual input.*

Ringach and Shapley (2004): *Modern account of reverse correlation and receptive field theory which examples from the visual system.*

Kay et al. (2008): *This paper extends the notion of the receptive field to voxels, i.e., small volumes of the brain, whose activity can be resolved by noninvasive brain scanning. The measured receptive fields are then used to decode the neural activity. By this approach, it is possible to "predict" from the recorded neural activity which of a number of known pictures the subject was looking at.*

References

Abeles, M., and M. H. Goldstein. 1977. Multispike train analysis. *Proceedings of the IEEE* 65: 762–773.

Borsellino, A., and T. Poggio. 1973. Convolution and correlation algebras. *Kybernetik* 13: 113–122.

Dayan, P., and L. F. Abbott. 2001. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: The MIT Press.

DeAngelis, G. C., I. Ohzawa, and R. D. Freeman. 1995. Receptive-field dynamics in the central visual pathway. *Trends in Neurosciences* 18, 451–458.

Eggermont, J. J., P. I. M. Johannesma, and A. M. H. J. Aertsen. 1983. Reverse-correlation methods in auditory research. *Quarterly Review of Biophysics* 16, 341–414.

Fantana, A. L., E. R. Soucy, and M. Meister. 2008. Rat olfactory bulb mitral cells receive sparse glomerular inputs. *Neuron* 59: 802–814.

Gold, C., D. A. Henze, C. Koch, and G. Buzsáki. 2006. On the origin of the extracellular action potential waveform: A modeling study. *Journal of Neurophysiology* 95: 3113–3128.

Gonzalez, R. C., and R. E. Wood. 2018. *Digital Image Processing*. 4th ed. New York: Pearson.

Hartline, H. K. 1938. The response of single optic nerve fibers of the vertebrate eye to illumination of the retina. *American Journal of Physiology* 121: 400–415.

Hartline, H. K., and F. Ratliff. 1958. Spatial summation of inhibitory influences in the eye of Limulus, and the mutual interaction of receptor units. *Journal of General Physiology* 41: 1049–1066.

Heeger, D. J. 1992. Normalization of cell responses in cat striate cortex. *Visual Neuroscience* 9: 181–197.

Kay, K. N., T. Naselaris, R. J. Prenger, and J. L. Gallant. 2008. Identifying natural images from human brain activity. *Nature* 452: 352–355.

Kuffler, S. W. 1953. Discharge patterns and functional organization of mammalian retina. *Journal of Neurophysiology* 16: 37–68.

Lettvin, J. Y., H. R. Maturana, W. S. McCulloch, and W. H. Pitts. 1959. What the frog's eye tells the frog's brain. *Proceedings of the Institute of Radio Engineers* 47: 1950–1961.

Mach, E. 1865. Über die Wirkung der räumlichen Vertheilung des Lichtreizes auf die Netzhaut. In *Sitzungsberichte der mathematisch-naturwissenschaftlichen Classe der kaiserlichen Akademie der Wissenschaften Wien*, vol. 52/2, 303–322. English translation in Ratliff (1965).

Mallot, H. A., W. von Seelen, and F. Giannakopoulos. 1990. Neural mapping and space–variant image processing. *Neural Networks* 3, 245–263.

McCulloch, W. S., and W. Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5: 115–133.

Müller, J. P. 1837. *Handbuch der Physiologie des Menschen für Vorlesungen*. Vol. II. Coblenz: J. Hölscher.

Naka, K. I., and W. A. H. Rushton. 1966. S-potentials from luminosity units in the retina of fish (Cyprinidae). *Journal of Physiology* 185: 587–599.

Norrsell, U., S. Finger, and C. Lajonchere. 1999. Cutaneous sensory spots and the "law of specific nerve energies": History and development of ideas. *Brain Research Bulletin* 48, 457–465.

O'Keefe, J., and J. Dostrovsky. 1971. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research* 34: 171–175.

Oppenheim, A. V., A. S. Willsky, and S. H. Nawab. 1997. *Signals & Systems*. 2nd ed. Upper Saddle River, New Jersey: Prentice Hall.

Piccinini, G. 2004. The first computational theory of mind and brain: A close look at McCulloch and Pitt's 'Logical calculus of ideas immanent in nervous activity'. *Synthese* 141: 175–215.

Rapela, J., J. M. Mendel, and N. M. Grzywacz. 2006. Estimating nonlinear receptive fields from natural images. *Journal of Vision* 6: 441–474.

Ratliff, F. 1965. *Mach Bands: Quantitative Studies on Neural Networks in the Retina*. San Francisco, London: Holden–Day.

Ringach, D., and R. Shapley. 2004. Reverse correlation in neurophysiology. *Cognitive Science* 28: 147–166.

Rudin, W. 1991. *Functional Analysis*. 2nd ed. Singapore: McGraw–Hill.

Schetzen, M. 1980. *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley.

Schwartz, A. B., R. E. Kettner, and A. P. Georgopoulos. 1988. Primate motor cortex and free arm movements to visual target in three-dimensional space. I. Relations between single cell discharge and direction of movement. *The Journal of Neuroscience* 8: 2913–2927.

Sherrington, C. S. 1906. Observations on the scratch-reflex in the spinal dog. *The Journal of Physiology* 34: 1–50.

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

Functional Models of Receptive Fields

3

Abstract

Receptive fields are crucial elements in the brain's processing of information. They filter out meaningful components from the stream of incoming data and define higher order features. They allow for the storage and further processing of such features both in individual sensory modalities and in multimodal integration. On the output side, motor fields similarly support the generation of patterns of coordinated movement. In this chapter, we study specific examples from the visual system that are paradigmatic for receptive fields also in other parts of the brain. Center-surround processing in the retina and lateral geniculate nucleus (LGN) enhance intensity and color contrast and remove irrelevant information from the images. The standard model here is based on Gaussians and convolution. Neurons in the visual cortex are characterized by their selectivity for edge orientation that is modeled by Gabor functions, convolutions, and point nonlinearities. As an example of a more derived feature, we study the detection of visual motion; it is based on spatiotemporal orientation selective receptive fields with a squaring nonlinearity.

Learning Objectives

- Gaussians and Gabor functions in two dimensions
- Spatiotemporal center-surround organization of retinal ganglion cells
- Orientation specificity and specificities for other parameters in cortical simple cells
- Energy model of cortical complex cells
- Neural processing of visual motion and three-dimensional (spatiotemporal) Gabor functions

H. A. Mallot, *Computational Neuroscience*,
https://doi.org/10.1007/978-3-031-75705-1_3

3.1 Retinal Ganglion Cells: Isotropic Center-Surround Organization

The retina is a complex, multilayered part of the brain, located in the fundus of the eye, and specialized to the transduction of light stimuli into neural signals, their initial processing, and the eventual transmission to other parts of the brain via the optic nerve (Wässle and Boycott 1991; Masland 2001). In short, it consists of five main types of neurons of which many subtypes have been described. The outermost layer contains the receptor cells—cones and rods—in which the transduction of the light stimulus into a membrane potential takes place. They connect via bipolar cells directly to the innermost layer, in which the retinal ganglion cells reside. The axons of the ganglion cells leave the eye and form the optic nerve. In addition to the direct pathway from the receptors via the bipolar cells to the ganglion cells, two lateral connectivity systems exist: Horizontal cells connect receptors with laterally offset bipolar cells, while amacrine cells establish lateral connections between ganglion cells. Of all these cell types, only the ganglion cells produce action potentials.

As pointed out in the previous chapter, the retina and specifically the retinal ganglion cells have played an important role in the discovery of neural specificity and the notion of the receptive field. We discuss them here for two reasons: their historical relevance for the development of the field and their biological relevance as the initial stage of visual information processing.

3.1.1 Difference of Gaussians

If a stimulus is moving over the receptive field of a retinal ganglion cell, the response will not depend on the direction of movement, be it left–right, up–down, or oblique.[1] Likewise, if an oriented light rectangle (a "bar") crossing the receptive field center is flashed on and off, the response does not depend on the orientation of the bar. This behavior is called "isotropic" and implies that the receptive field function is rotationally symmetric.

Isotropic center-surround organization as shown in Fig. 2.1 is usually modeled as the combination of two Gaussian functions, see Box 3.1. Center and surround are represented by two separate Gaussians with different width (scale factors σ_c, σ_s) and amplitudes (m_c, m_s). We first discuss the purely spatial case, in which stimulus variations over time are absent. We may then write:

$$\begin{aligned} DoG(x, y) &= \phi_c(x, y) - \phi_s(x, y) \\ &= m_c \exp\left\{-\frac{x^2 + y^2}{2\sigma_c^2}\right\} - m_s \exp\left\{-\frac{x^2 + y^2}{2\sigma_m^2}\right\}, \end{aligned} \tag{3.1}$$

[1] This statement is generally true for primates. In rabbits and frogs, direction-selective ganglion cells have also been found.

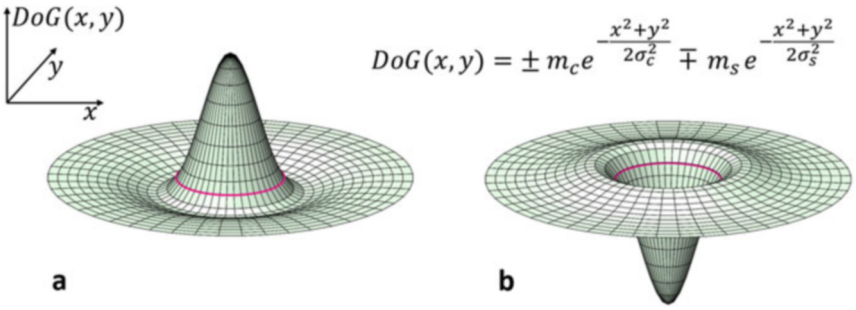
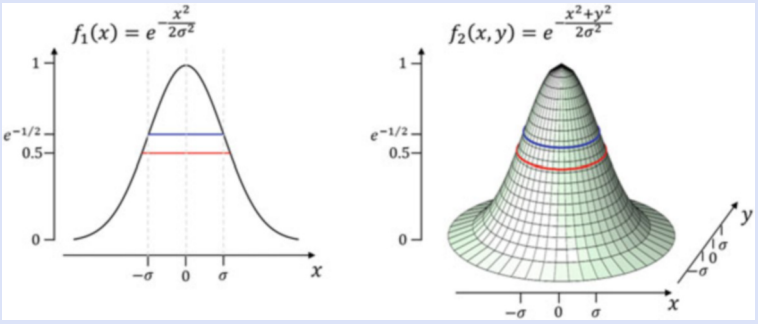


Fig. 3.1 Difference of Gaussians. The red line marks the zeros of the function, i.e., the loci (x, y) where $DoG(x, y) = 0$. Center and surround components have amplitudes m_c and m_s and widths σ_c and σ_s, respectively. In the figure we chose $\sigma_s/\sigma_c = 1.6$ and $m_c/m_s = -\sigma_s^2/\sigma_m^2$ such that the total integral over the DoG function vanishes. (**a**) ON-center OFF-surround; (**b**) OFF-center ON-surround

see Fig. 3.1. This function is also known as the Difference of Gaussians or DoG function and was first proposed as a model of the retinal ganglion cell receptive field by Rodieck and Stone (1965). It models lateral inhibition in the sense that the three-point stencil $(-0.5, 1, -0.5)$ shown in Fig. 2.3b can be considered a coarsely sampled version of a (one-dimensional) DoG. In addition, it has more degrees of freedom than shown in Fig. 2.3, in that the amplitudes and spatial extend of activation and inhibition may vary. Spatial extend is modeled by the factors σ_c and σ_s that define the size of the retinal receptive field. While lateral inhibition is modeled by the ON-center, OFF-surround DoG (Fig. 3.1a), an equal amount of retinal ganglion cells show "lateral activation," or OFF-center, ON-surround organization (Fig. 3.1b).

Box 3.1 The Gaussian Function and its Total Mass



(continued)

Box 3.1 (continued)
The Gaussian function is the most commonly used model of a "bump" or localized distribution of some quantity. It is bell shaped and centered at the origin with inflection points (zeros of the second derivative) at $\pm\sigma$. The figure shows the width at the inflection point ($\pm\sigma$) as a blue line and the width at half height as a red line. In the two-dimensional case, the value depends only on $x^2 + y^2$, which means that the "bell" is rotationally symmetric about the vertical axis and all contour lines are circles. All vertical sections are again (one-dimensional) Gaussians and so are the marginal integrals of the form $g(y) = \int f_2(x, y)dx$, etc.

The indefinite integral of the Gaussian (with some suitable normalizations) is called the error function; it has no analytical expression. However, we can calculate the improper integral in the two-dimensional case (i.e., the total volume of the bell) by the substitution $(x, y) = r(\cos\phi, \sin\phi)$:

$$V = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\} dxdy = \int_{-\pi}^{\pi}\int_{0}^{\infty} \exp\left\{-\frac{r^2}{2\sigma^2}\right\} rdrd\phi.$$

The inner integral on the right side evaluates to σ^2, yielding $V = 2\pi\sigma^2$.

The integral of the one-dimensional Gaussian is derived from this by observing that $f_2(x, y) = f_1(x)f_1(y)$. We can therefore separate the double integral as

$$\begin{aligned} V &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_2(x, y)dxdy = \int_{-\infty}^{\infty} f_1(x)dx \int_{-\infty}^{\infty} f_1(y)dy \\ &= \left(\int_{-\infty}^{\infty} f_1(x)dx\right)^2 \end{aligned}$$

and obtain

$$\int_{-\infty}^{\infty} f_1(x)dx = \sqrt{V} = \sigma\sqrt{2\pi}.$$

For Gaussians in n dimensions, $f_n(\mathbf{x}) = f_1(x_1) \times \ldots \times f_1(x_n) = f_1(\|\mathbf{x}\|)$, this result generalizes to

$$\int_{\mathbb{R}^n} f_n(\mathbf{x})d\mathbf{x} = \sigma^n(2\pi)^{n/2}.$$

In statistics (normal distribution) and other applications, these factors are used to normalize Gaussians to a total mass of 1.

In primates, the receptive field sizes of retinal ganglion cells vary between about 1 minute of arc[2] in the fovea and several degrees in the periphery of the visual field (Croner and Kaplan 1995). Ganglion cells can be subdivided into a small ("parvocellular," P) and a large ("magnocellular," M) type that cover overlapping portions of the visual field and have small or large receptive fields, as implied by their names. They thereby respond to patterns of different granularity or coarseness, such as individual letters in a text vs. the layout of the print area on a book page. The idea of processing images simultaneously in multiple scales has been picked up in technical image processing, where it is known as "scale space" or "resolution pyramid." It is realized by choosing different values of the scale[3] factor σ (see, Marr and Hildreth 1980).

Another function that is sometimes also used to model retinal ganglion cells is the Laplacian-of-a-Gaussian, or LoG function

$$\mathrm{LoG}(x, y) = \Delta f_2(x, y) = \frac{\partial^2}{\partial x^2} f_2(x, y) + \frac{\partial^2}{\partial y^2} f_2(x, y), \tag{3.2}$$

where Δ denotes the Laplace operator and f_2 is the two-dimensional Gaussian as defined in Box 3.1. We set $r^2 = x^2 + y^2$ and obtain

$$\mathrm{LoG}(r) = (r^2 - 2) \exp\left\{-\frac{r^2}{2\sigma^2}\right\}. \tag{3.3}$$

It is motivated by Mach's theory of lateral inhibition (Eq. 2.19) and treats center-surround processing as a cascade of spatial averaging with a Gaussian window and subsequent application of the Laplace operator. Note that the Laplacian is itself a linear shift-invariant operation that satisfies the associative law; its convolution kernel is $\Delta\delta$ as explained in Box 2.5. If we denote by G and I the Gaussian and the input image, we may therefore write $\Delta(G * I) = (\Delta G) * I$, where ΔG and LoG are the same. In the first version, $\Delta(G * I)$, we may think of the LoG operation as the Laplacian of a smoothed image, while the second version, $(\Delta G) * I$, describes a smoothing lateral inhibition applied to the original image. The LoG function closely resembles the OFF-center DoG function with $\sigma_s/\sigma_c = 1.6$ (Marr and Hildreth 1980) but does not allow to independently vary the ranges and amplitudes of center and surround, as would be needed to fit experimental receptive field profiles.

[2] The sizes of visual receptive fields are given in degrees of visual angle, which can be converted into millimeters on the fundus of the eye by multiplication of the angle in radians with the radius of the eyeball. A minute of arc is one sixtieth of a degree of arc, i.e., 0.01667°.

[3] In receptive field theory, the letter σ can therefore be taken as mnemonic for "scale," rather than for "standard deviation," which would make little sense in this context.

3.1.2 Dynamic Model

The ON and OFF ganglion cells with the receptive fields depicted in Fig. 3.1a, b are parts of two parallel processing streams originating already at the first retinal synapse (Werblin and Dowling 1969). The receptor cells feed into two types of bipolar cells that in turn form the input to the ganglion cells. Upon illumination, ON-bipolar cells are activated (i.e., depolarized), while OFF-bipolar cells get inhibited (hyperpolarized). For an input image $I(x, y)$, the signals in the ON and OFF channels can therefore be approximated as $I(x, y)$ and $1 - I(x, y)$, respectively. The constant 1 is assumed to represent the highest possible intensity and is added to avoid negative intensities (see also Fig. 3.2).

The ON and OFF ganglion cells receive input from the respective bipolar cells and process it further by their center-surround connectivity. While the center is mostly driven by direct inputs, inhibitory surrounds are realized by additional neurons, known as horizontal and amacrine cells. This involves additional synapses that affect the dynamics of the surround part and render it slower than the receptive field center. Note that the "surround" mechanism is not interrupted in the center region but overlaid by the stronger center mechanism; it does show in the dynamics of the system. For example, in the ON-center, OFF-surround case, central stimulation leads to an initial strong response mediated by the center mechanism that is reduced as soon as the response of the surround mechanism sets in.

The full spatiotemporal model of retinal ganglion cells appears in Fig. 3.2 (cf. Rodieck 1965). Spatially, center and surround are modeled by Gaussians of different width, as already used in the static model above. Dynamics is added as impulse responses of the form

$$g_c(t) = \frac{t}{\tau_c^2} e^{-t/\tau_c} \quad \text{and} \quad g_s(t) = \frac{t}{\tau_s^2} e^{-t/\tau_s} \tag{3.4}$$

separately for the center and the surround. The time constants τ control the speed of processing; we have $\tau_s > \tau_c$. The functions are normalized to a total integral of 1. For more elaborate modeling of retinal impulse responses, see, for example, Enroth-Cugell and Shapley (1973) and Chichilnisky and Kalmar (2002). With the notations from Eqs. 3.1 and 3.4, we can now write the full spatiotemporal kernel applied to the bipolar cell signal as follows:

$$w(x, y, t) = g_c(t)\phi_c(x, y) - g_s(t)\phi_s(x, y). \tag{3.5}$$

Rodieck (1965) considers the "response functions" $I * w$ for the ON channel and $(1-I)*w$ for the OFF channel, where $*$ denotes convolution. They can take negative values and are inverted versions of each other. Indeed, we have $-I*w = (1-I)*w$, at least if we choose the model parameters such that the total integral over w is zero, so that the cells do not respond to constant and uniform stimuli. Of course, the spike rates a_{ON} and a_{OFF} measurable from the retinal ganglion cells will only show

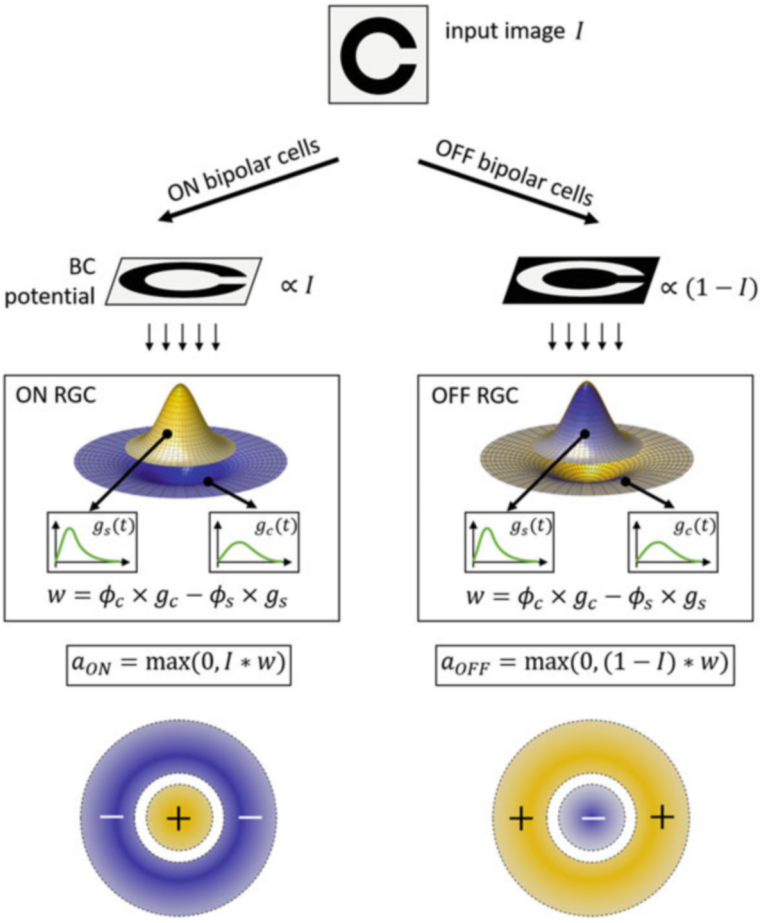


Fig. 3.2 Standard model of retinal ganglion cell (RGC) receptive fields after Rodieck (1965). The input image is transduced to a membrane potential in the receptors cells of the outer retinal layer and passed on in parallel to two types of bipolar cells (BC), known as ON and OFF BCs. The ON type is depolarized upon illumination, and the OFF type is hyperpolarized. The RGCs receive input over a center and a surround mechanism. Spatially, these are modeled as Gaussians ϕ_c and ϕ_s with different scales, amplitudes, and signs. The dynamics are modeled as low-passes g_c and g_s differing in their time constants. The RGCs generate spikes if their total input is positive. Note that color vision is not included in this figure. The yellow and blue colors symbolize activation and inhibition, respectively

the positive values of this function, as is indicated by the rectifying nonlinearity $(\max(0, w * I))$ in the figure.

Figure 3.3 shows the so-called space-time plot of the response profile w. The spatial dependence is shown only for the line $y = 0$ passing horizontally through the receptive field center. Each horizontal line in the plot shows the spatial dependence for one instance in time (temporal offset between stimulus and response), and each

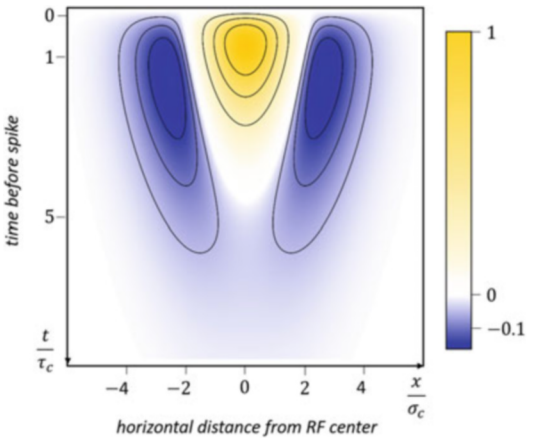


Fig. 3.3 Space-time plot of the spatiotemporal kernel $w(x, y, t)$ from Eq. 3.5 for $y \equiv 0$. The time constants are set to $\tau_s = 1.5\tau_c$ where $\tau_c = 30\,\text{ms}$ is a typical value. The color bar indicates the value of w in arbitrary units, and contour lines show 75, 50, and 25% of maximal activation and inhibition. Note that this is an example of a "non-separable" spatiotemporal profile since the time dependence in Eq. 3.5 cannot be bracketed out (unless $\tau_c = \tau_s$), see Eq. 2.31. In the figure, this shows in the boundary between the excitatory (yellow) and inhibitory (blue) parts, i.e., the zeros: In separable profiles, these should be straight lines parallel to one of the axes

vertical line shows the temporal response to a point stimulus (spatiotemporal δ-pulse) delivered at $(x, 0)$. Note that responses to near central stimuli will be biphasic, i.e., activation followed by inhibition, since the activating center and the inhibitory surround act with different speeds.

3.1.3 Why ON–OFF Channels?

In visual perception, an intensity increment is generally just as important as a decrement. We can write text in black letters on white paper, which will mostly activate the OFF channel, or in white letters on black background, thereby stimulating mostly the ON channel. Increments and decrements should clearly be treated in equivalent ways. In truly linear systems, this would be possible by allowing positive and negative signals for increments and decrements, but this is impossible in spiking neurons that always show rectification and saturation-type nonlinearity. The parallel ON and OFF channels allow to respond to both increments and decrements with equal neural activities. In the completely symmetric model described above, it even reinstates linearity, as can be seen from the following consideration: In the linear case, the response functions $w * I$ would carry the complete information, given that it can take negative values. While the negative parts are cut away by the rectification,

the OFF channel with its response function $-w * I$ provides just these missing parts, such that the complete response function can be reconstructed:

$$w * I = a_{ON} - a_{OFF}. \tag{3.6}$$

In a sense, the parallel processing in an ON and an OFF pathway can therefore be seen as a mechanism to overcome the disadvantages of nonlinearity in early visual processing. In further processing steps, nonlinearities make important contributions, as will be explained in the sections on motion detection and pattern classification.

▶ **Key Point: Difference of Gaussians (DoG)** The DoG function models lateral inhibition or center-surround organization in the early stages of the visual system, most notably in retinal ganglion cells. Parallel ON/OFF channels allow for an equal representation of incremental (light on dark) and decremental (dark on light) image structures.

3.2 Primary Visual Cortex: Edge Orientation

3.2.1 Orientation Specificity

In the mammalian primary visual cortex (area V1) and also in areas further down the stream of visual processing, neurons will only weakly respond to circular spots of retinal illumination, while elongated rectangles of light ("bars") shone on the retina elicit brisk and vivid responses. This behavior, first observed by Hubel[4] and Wiesel[5] (1959) in the cat, shows how the processing of visual information proceeds from the retina to the cortex: Cortical neurons detect features, that is: salient subpatterns in images such as oriented bars, moving bars, or binocular pairs of bars with various disparities corresponding to different positions in depth. These features are characterized not just by their position in the retinal image, but also by one or more stimulus parameters such as bar orientation, the speed and direction of motion, or binocular disparity. While these parameters differ from neuron to neuron, the specificity for the orientation of edges and bars as such is the common hallmark of cortical receptive fields.

Like the center-surround receptive fields studied so far, oriented receptive fields have excitatory and inhibitory subregions, which, however, are not concentric rings but form elongated stripes laid out side by side. Hubel and Wiesel (1962) suggested that this may be achieved by rows of isotropic receptive fields converging into a single cortical neuron. The mathematical model would thus be a sum of DoG

[4] David Hunter Hubel, 1926–2013, Canadian neurophysiologist. Nobel Prize in physiology or medicine 1981.

[5] Torsten Nils Wiesel, born 1924, Swedish neurophysiologist. Nobel Prize in physiology or medicine 1981.

functions centered at different positions along a straight line. A theoretical analysis by Daugman (1980) showed that such alignments of isotropic fields predict different orientation preferences for bar and grating[6] stimuli and that orientation preferences may even change with the spatial frequency of the grating. Such dependencies, however, have not been reported experimentally.

The Hubel and Wiesel model of a row of aligned isotropic fields forming an oriented receptive field by feed-forward connectivity is still found in many textbooks, probably because it illustrates the fact that neurons with isotropic fields in the lateral geniculate nucleus (LGN) do provide the input for orientation selective neurons in V1. The model is misleading; however, in that it ignores the role of intracortical connectivity, including feedback and inhibition (Sillito 1975; Ferster and Miller 2000). As pointed out before, it also fails to explain how equal orientations preferences are achieved for bar and grating stimuli.

Receptive field functions with well-defined orientation preferences are now generally modeled using the so-called Gabor[7] function, generated by multiplying a (one- or two-dimensional) sinusoidal with a Gaussian envelope (see Daugman 1980; Marčelja 1980; Jones and Palmer 1987). The Gabor function is mathematically simpler than the aligned DoG model, provides a satisfactory description of orientation specificity, and is simultaneously localized in the space and frequency domains (a property that can be fully appreciated only after reading Chap. 4). Note, however, that the Gabor model of orientation selectivity is purely phenomenological: that is, it does not explain the neural circuitry underlying the receptive field structure. Some models for the formation of this circuitry will be addressed in Sect. 6.3.3.

Gabor functions are now generally used in technical vision systems and image compression and occur as a result of the deep learning algorithm in the input layers of pattern recognition networks. For us, however, the most important property is their superior fitting ability for the response profiles of cortical orientation selective neurons (Jones and Palmer 1987).

3.2.2 Gabor Function in One and Two Dimensions

The Gabor function is a sinusoidal multiplied with a Gaussian envelope or window function. In one dimension, we have

$$g_c(x) := \cos(\omega x)\ \exp\left\{-\frac{x^2}{2\sigma^2}\right\} \tag{3.7}$$

[6] A visual grating is a regular pattern of stripes, either black and white or with sinusoidal intensity modulation. Its spatial frequency is the number of stripes (or cycles) per degree of visual angle. See Fig. 3.7.

[7] Dennis Gábor (1900–1979). Hungarian physicist. Nobel Prize in Physics 1971.

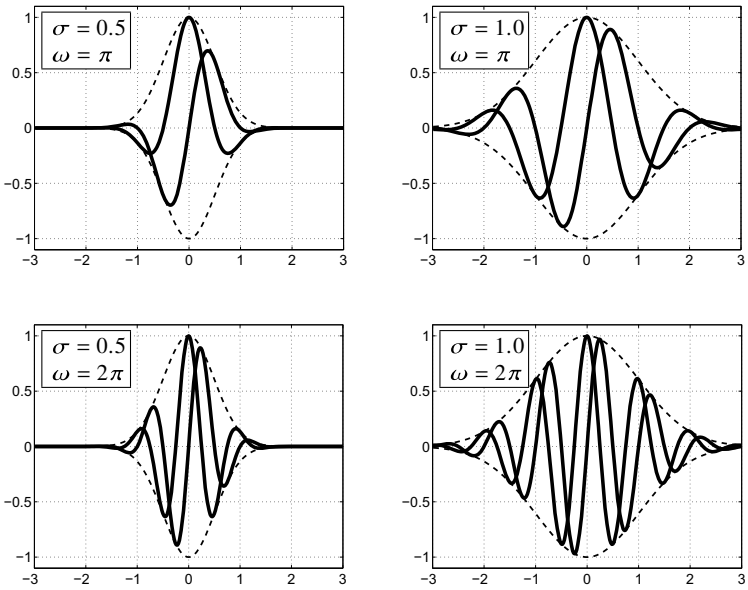


Fig. 3.4 One-dimensional Gabor functions (Eqs. 3.7, and 3.8) for various choices of scale σ and frequency ω. In each plot, the even (left–right mirror symmetric) heavy line is the cosinusoidal Gabor function, and the odd (point symmetric heavy line) is the sinusoidal Gabor function. The dashed lines show the enveloping Gaussians. Note that the top-right and bottom-left plots are identical up to a horizontal compression. This reflects that fact that in these cases the product $\sigma\omega$ is the same. It is proportional to the number of "side lobes" fitting under the Gaussian envelope

$$g_s(x) := \sin(\omega x)\ \exp\left\{-\frac{x^2}{2\sigma^2}\right\}. \tag{3.8}$$

Like the cosine itself, the cosinusoidal Gabor function is "even," i.e., it satisfies the condition that $g_c(x) \equiv g_c(-x)$, while the sinusoidal Gabor function, like the sine, is odd, $g_s(x) \equiv -g_s(-x)$ (cf. Fig. 3.4). 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"

$$f(x, y) = \cos(\omega_x x + \omega_y y) \tag{3.9}$$

depicted in Fig. 3.5a. It describes a corrugated surface similar to a wash board. Sections along the direction $\boldsymbol{\omega} = (-\omega_y, \omega_x)$, i.e., along a "wavefront" are constant. All other sections are sinusoidals with various frequencies; in particular, the

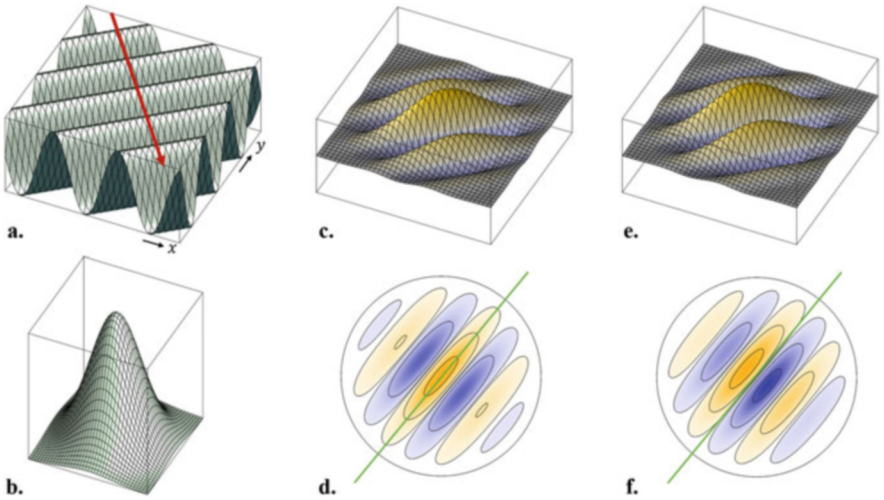


Fig. 3.5 Construction of two-dimensional Gabor functions. (**a**) Plane wave $f(x, y) = \cos(\omega_x x + \omega_y y)$. The red line marks the direction of wave "propagation," given by the vector $\boldsymbol{\omega} = (\omega_x, \omega_y)$. The orientation is the direction of the wavefronts that are orthogonal to the red line. (**b**) Gaussian function $f(x, y) = \exp\{-(x^2 + y^2)/2\sigma^2\}$. (**c**) Cosine Gabor function obtained by multiplying the plane wave and the Gaussian. (**d**) Same as c, shown as contour plot. The green line is the axis of symmetry showing the "even" structure of the cosine Gabor function. (**e**) Sine Gabor function obtained by phase shifting the plane wave by a quarter cycle and multiplying it with the unmoved Gaussian. (**f**) Same as e, shown as contour plot. The function is antisymmetric (odd) with respect to the green line

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 wavefront; it is marked in the figure by a red arrow. The spatial frequency of the two-dimensional wave $f(x, y)$ is the vector $(\omega_x, \omega_y)^\top$; it describes both the frequencies and the orientation of the wave. As in the one-dimensional case, the Gabor function is obtained by multiplying the sinusoidal with a Gaussian, see Fig. 3.5b. If the Gaussian is centered on a wave peak or trough, the result will be a symmetric, cosine Gabor function (Fig. 3.5c,d); if the Gaussian is centered on a flank, an odd or sine Gabor function will result (Fig. 3.5e,f). The equations read

$$g_c(x, y) := \cos(\omega_x x + \omega_y y) \ \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\} \tag{3.10}$$

$$g_s(x, y) := \sin(\omega_x x + \omega_y y) \ \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\}. \tag{3.11}$$

Receptive fields of real neurons rarely show the pure sinusoidal or cosinusoidal symmetry. Rather, intermediate cases are found, which can be modeled by shifting the plane wave relative to the Gaussian envelope by a phase factor φ that may take

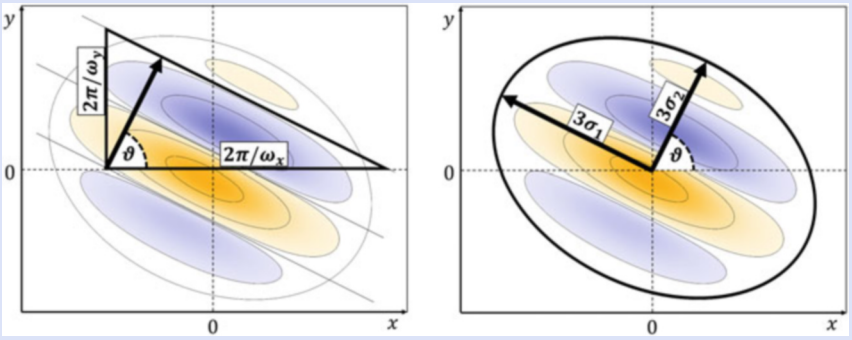
arbitrary values. The wave part of the equation then becomes $\cos(\omega_x x + \omega_y y - \varphi)$. For $\varphi = 0$ and $\varphi = \pi/2$, the pure cosine and sine functions are reproduced.

Box 3.2 summarizes the Gabor model used by Jones and Palmer (1987) for fitting cortical receptive fields. It allows also for Gabor functions elongated in one direction (typically the wavefronts) and compressed in the opposite direction. In this case, the Gaussian has two scale factors, σ_1 and σ_2, one for each direction.

To sum up, Gabor functions are characterized by the following parameters that model some of the major specificities found in neurons in the primary visual cortex. These are:

1. The receptive field *center* ($\mathbf{x}_o$ in Box 3.2).
2. The *size* of the receptive field, specified by the *scale* variables σ_1 and σ_2 explained in Box 3.2. If $\sigma_1 = \sigma_2$, the outline is circular. If not, the receptive field is elongated by the aspect ratio σ_1/σ_2. Elongation is generally in the direction of the wavefronts. The field size is proportional to $\sigma_1\sigma_2$.
3. The preferred *spatial frequency* is the frequency of a grating at any orientation driving the cell most strongly. This frequency is determined by $\sqrt{\omega_x^2 + \omega_y^2} = \|\boldsymbol{\omega}\|$. Spatial frequency preference is sometimes referred to as "localization in the spatial frequency domain."
4. The preferred *orientation* depends on the ratio ω_y/ω_x. If expressed as an angle from the x-axis, orientation becomes $\vartheta = \tan^{-1}(\omega_y/\omega_x)$, see Box 3.2. The variables ϑ, ω_x, ω_y, and $\|\boldsymbol{\omega}\|$ are related by the equations $\omega_x = \|\boldsymbol{\omega}\| \cos\vartheta$ and $\omega_y = \|\boldsymbol{\omega}\| \sin\vartheta$.
5. The *phase* φ controls the odd or even symmetry of the Gabor function.
6. The *number of side lobes* is a derived parameter determined by the product of the overall spatial frequency and scale, $\|\boldsymbol{\omega}\|\sigma_2$.

Box 3.2 Parameters of the Two-Dimensional Gabor Function



(continued)

Box 3.2 (continued)
Here we summarize the parameters used in the function fits by Jones and Palmer (1987). The general two-dimensional Gabor function with elliptical outline and an arbitrary phase is given by

$$g(\mathbf{x}) = A\cos((\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}_o)) - \varphi)\ \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_o)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\mathbf{x}_o)\right\}.$$

Here, we have used vectorial notation with $\mathbf{x} = (x, y)^\top$ and $\boldsymbol{\omega} = (\omega_x, \omega_y)^\top$ and the dot product $(\boldsymbol{\omega}\cdot\mathbf{x}) = \omega_x x + \omega_y y$; $\mathbf{x}_o$ is the receptive field center and A is an amplitude factor. $\boldsymbol{\Sigma}$ is a 2×2 matrix known in statistics as covariance matrix (see also Box 6.2); it depends on ω_x, ω_y, and two scale variables σ_1 and σ_2 and determines the shape and orientation of the elliptic envelope.

The figures show the geometric interpretation of the parameters. The oblique lines in the left figure are the "wavefronts" satisfying $x\omega_x + y\omega_y - \varphi = n\pi$ for some integer $n \in \mathbb{Z}$. The black triangle spans one wavelength in the x and y directions; its sides are therefore $2\pi/\omega_x$ and $2\pi/\omega_y$. From this we can calculate the angle ϑ as $\tan\vartheta = \omega_y/\omega_x$; the wavefront orientation is orthogonal to ϑ, i.e., $\vartheta^\perp = -\tan^{-1}(\omega_x/\omega_y)$. The right image shows an elliptic outline produced by the quadratic form in the argument of the Gaussian. We first observe that a (counterclockwise) rotation by the angle ϑ is given by the rotation matrix

$$\mathbf{R} = \begin{pmatrix} \cos\vartheta & -\sin\vartheta \\ \sin\vartheta & \cos\vartheta \end{pmatrix} = (w_x^2+\omega_y^2)^{-\frac{1}{2}} \begin{pmatrix} \omega_y & \omega_x \\ -\omega_x & \omega_y \end{pmatrix}.$$

A suitable matrix $\boldsymbol{\Sigma}$ describing an ellipse with half axes σ_1 in the direction of the wavefronts and σ_2 in the orthogonal direction is then obtained as

$$\boldsymbol{\Sigma} = \mathbf{R}^\top \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \mathbf{R}.$$

The ellipse shown in the figure is the contour line of a Gaussian with $\sigma_1/\sigma_2 = 1.4$ at level $g(\mathbf{x}) = \exp\{-3^2/2\}$. Its half axes are $3\sigma_1$ and $3\sigma_2$.

The free parameters for fitting are $\mathbf{x}_o$, A, ω_x, ω_y, φ, σ_1, and σ_2. Instead of ω_x, ω_y, one might also use $\|\boldsymbol{\omega}\| = (\omega_x^2+\omega_y^2)^{1/2}$ and ϑ.

▶ **Key Point: Gabor Function** The two-dimensional Gabor function is the standard model of orientation selective receptive fields in the visual cortex. It consists of a sinusoidal wave multiplied with a Gaussian envelope. Excitatory and inhibitory subregions correspond to the positive and negative wave regions and form a pattern of oriented stripes within a circular or elliptical window.

3.3 Simple and Complex Cells: The "Energy" Model

3.3.1 Response Properties

Hubel and Wiesel (1962) distinguished between two types of cortical cells specifically responding to bars of a preferred orientation, which they called "simple" and "complex," respectively. Simple cells have well-defined subregions, which may be modeled as the excitatory and inhibitory lobes of a Gabor function. From these, the overall response to patterned stimuli can be predicted by the linear correlation operation (Eq. 2.11) followed by a point nonlinearity such as half-wave rectification. If the stimulus, say an elongated bar, is slightly moved away from its preferred position, it might fall on an inhibitory subregion of the receptive field and the response will drop, even though the orientation of the bar is unchanged. This is to say, the neuron is tuned not only to orientation, but also to position.

In contrast, complex cell responses are invariant to small positional shifts of a bar stimulus while at the same time keeping the sharp tuning for orientation. This behavior cannot be explained by simply assuming coarser receptive fields, since these would lose specificity for position and orientation alike. In fact, position-invariant orientation tuning cannot be explained by linear receptive field correlations with a subsequent point nonlinearity but requires some sort of nonlinear interaction. Hubel and Wiesel (1962) suggest a simple feed-forward circuit in which simple cells with the same preferred orientation but slightly offset field centers converge to a complex cell. If the stimulus moves out of the excitatory subfield of one simple cell, it enters the field of another one, which now drives the complex cell. The nonlinearity needed in this model is the maximum operation taken over the various simple-cell inputs, see Eq. 2.40. If the stimulus now falls on the inhibitory region of the first cell, this signal will not inhibit the input from the second cell, since it does not pass the nonlinearity.

The simple/complex distinction can also be studied with drifting gratings, i.e., stimuli of the form

$$I(x, y, t) = I_o + \frac{A}{2}(1 + \cos(\omega_x x + \omega_y y - \nu t)), \tag{3.12}$$

see Fig. 3.6. The orientation of the grating is $\vartheta^{\perp} = -\tan^{-1}(\omega_x/\omega_y)$, as explained in Box 3.2. The temporal frequency ν (Greek letter nu) is the modulation frequency seen at a fixed point, while the grating is passing over it. The grating is drifting in direction $\vartheta = \tan^{-1}(\omega_y/\omega_x)$ with velocity $\nu/\|\boldsymbol{\omega}\|$. Both simple and complex cells will be tuned to the orientation and the spatial frequency (bar width) of the grating. Over time, simple cells show the expected response, which is roughly a half-wave rectified (see Eq. 2.33) sinusoidal modulation with frequency ν (Fig. 3.7). Complex cell responses, however, will be completely unmodulated or at least show an unmodulated response component while still being tuned to a preferred orientation and spatial frequency (Movshon et al. 1978).

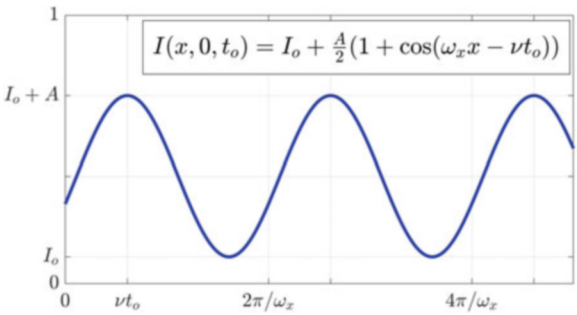


Fig. 3.6 Section through the drifting cosinusoidal grating, Eq. 3.12 with $y = 0$ and $t = t_o$. I_o, baseline intensity; A, amplitude; ω, spatial frequency; νt_o phase offset. The average value $I_o + A/2$ is a so-called DC (for direct current) component that prevents the function from taking negative values

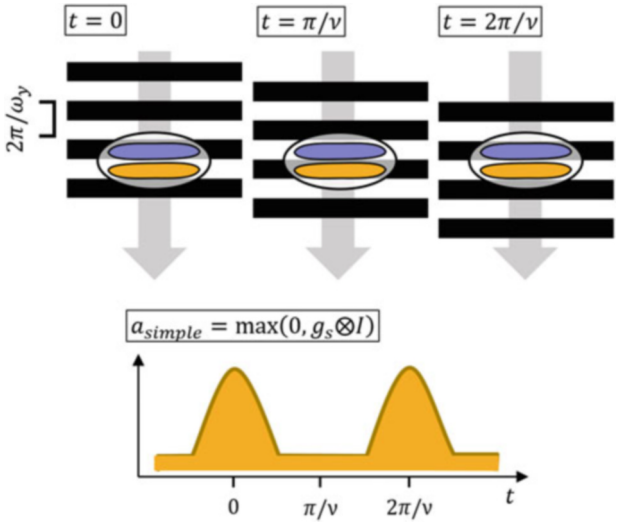


Fig. 3.7 *Top:* Three time steps of a box-grating drifting downward ($\omega_x = 0, \nu/\omega_y < 0$) and stimulating a simple cell with an odd symmetric Gabor-shaped receptive field. Note that the receptive field is fixed in space. As the grating moves by, the input seen by excitatory (yellow) and inhibitory (blue) subfields varies. *Bottom:* Response (spike rate a) of the simple cell after half-wave rectification. The response is modulated with frequency ν

This result leads to a slightly different view of the complex cell property: Complex neurons, like simple ones, are tuned to the orientation and spatial frequency of grating stimuli but (largely) ignore their phase (Pollen and Ronner 1981). This is not in conflict with the original idea of partial positional invariance, since in gratings, a phase shift is also a shift in position. As we will see in the next section, it allows a mathematically consistent modeling of complex cell receptive fields and their special type of nonlinear behavior using the notions of signal energy (or power) and quadrature pairs.

3.3.2 Model

How can we obtain the unmodulated grating response of complex cells from the sinusoidal modulations found in simple cells? Mathematically, the answer is found in a formulation of the Pythagoras theorem with sinusoidals,

$$\sin^2(\nu t) + \cos^2(\nu t) = 1. \tag{3.13}$$

If a grating is presented to two simple cells with receptive fields centered at the same position in the visual field, one with an even (cosine) symmetry and another one with an odd (sine) symmetry, the outputs of these cells prior to passing the point nonlinearity (i.e., the generator potential u in the sense of Eq. 2.32) will indeed be two sinusoidal modulations offset by a quarter cycle: that is, a sine and a cosine wave. The idea is illustrated in Fig. 3.8 that also includes a rectifying nonlinearity. In this case, four simple cells with phase offsets of zero, one quarter, one half, and three quarters of a cycle are needed. Their outputs sum up to an unmodulated signal.

For a mathematical proof we assume $\omega_x = 0$ (as in Fig. 3.7) and the same ω_y for both the Gabor functions and the grating. The cells are thus stimulated with

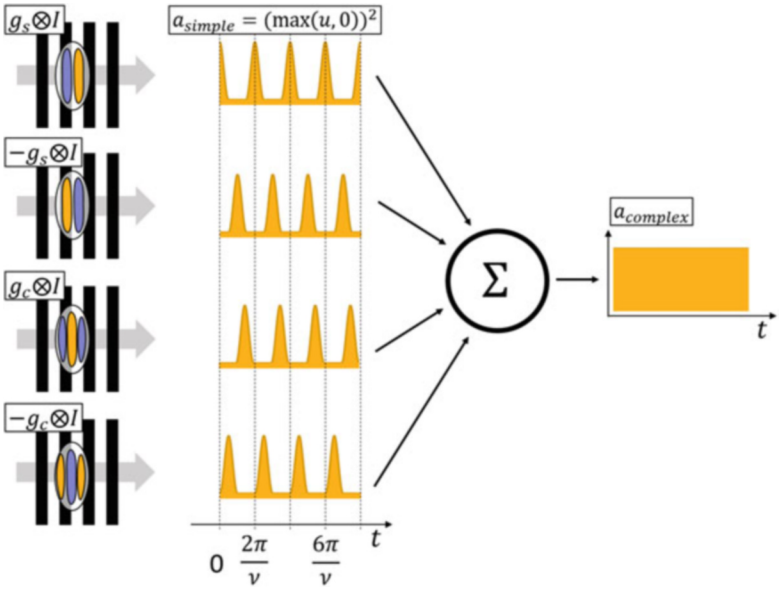


Fig. 3.8 Energy model of the cortical complex cell. On the left side, four simple receptive fields are shown with odd or even phase and positive or negative polarity; g_s and g_c as in Eq. 3.11. The outputs $u = g \otimes I$ are passed through the rectifying and squaring nonlinearities and show the expected modulations. If the four signals are summed up, the modulation goes away and a constant signal is constructed as is observed in complex cells. The complex cell will be tuned to the same orientation and spatial frequency as its inputs, but not to phase

their preferred orientation and spatial frequency. The response prior to the point nonlinearity is

$$u_s(t) = \iint g_s(x, y) I(x, y, t)\, dxdy \tag{3.14}$$
$$= \frac{A}{2}\left[\frac{1}{\sqrt{2\pi}\sigma}\int \sin(\omega_y y)\exp\left\{\frac{-y^2}{2\sigma^2}\right\}\cos(\omega_y y - \nu t)dy + \iint g(x, y)dxdy\right].$$

Here we have inserted g_s and I from Eqs. 3.11 and 3.12 with $\omega_x = 0$. The double integral in the second row results from the constant ("DC," for direct current) component in Eq. 3.12. It will be close to zero and can be neglected.[8] The factor $1/\sqrt{2\pi}\sigma$ results from the x-integral that was bracketed out in the first term. We lump all factors in one constant c, replace the Gaussian window by a box function over the interval $[-\pi, \pi]$, and apply the addition theorem $\cos(\alpha - \beta) = \cos\alpha\cos\beta + \sin\alpha\sin\beta$:

$$\begin{aligned} u_s(t) &\approx c\int_{-\pi}^{\pi} \sin(\omega_y y)\cos(\omega_y y - \nu t)dy \\ &= c\left[\cos\nu t\int_{-\pi}^{\pi}\sin(\omega_y y)\cos(\omega_y y)dy + \sin\nu t\int_{-\pi}^{\pi}\sin^2(\omega_y y)dy\right] \\ &= c\pi\sin\nu t. \end{aligned} \tag{3.15}$$

The first integral in the second row evaluates to zero, due to the orthogonality rules for sinusoidals (Eq. 4.50), while the second one evaluates to π. In conclusion, we have shown that the odd Gabor function stimulated with its preferred cosinusoidal drifting grating responds with a sinusoidal modulation. Its half-wave rectified version is shown by the yellow profile in the lower part of Fig. 3.7. Likewise, an even Gabor function g_c will respond with a cosinusoidal modulation (yellow profiles in the middle part of Fig. 3.8). We leave this proof to the reader.

A pair of two filters capturing the sine and cosine components of a modulated signal and combining the squared components as in Eq. 3.13 is called a "quadrature pair." Pollen and Ronner (1981) observed that adjacent simple cells, which had been recorded from visual cortex with the same electrode placement, usually share the same specificity for orientation and spatial frequency but that their response modulations are usually offset by a quarter cycle. This means that they can be combined into a quadrature pair as is illustrated in Fig. 3.8. Since all simple-cell outputs will be passed through a rectifying nonlinearity, two cells of opposite

[8] This is reminiscent of the use of "Morlet wavelets" in signal theory, which are normalized Gabor functions with zero DC components. Electrophysiological measurements of receptive field functions may show small nonzero DC responses.

polarity (i.e., $\pm g_s$ and $\pm g_c$) are needed for both the odd and even channels. The complex cell model becomes

$$a_{\text{complex}} = f(I \otimes g_s) + f(I \otimes (-g_s)) + f(I \otimes g_c) + f(I \otimes (-g_c))$$
$$f(u) = (\max(0, u))^2, \tag{3.16}$$

where a_{complex} is the firing rate of the complex neuron, g_c and g_s are the even and odd Gabor functions, and $\otimes$ denotes the correlation (Eq. 2.11). The Gabor functions $\pm g_c$ and $\pm g_s$ differ only in their polarity; instead of writing them down separately, we observe that $f(u) + f(-u) = u^2$ and obtain

$$a_{\text{complex}} = (I \otimes g_s)^2 + (I \otimes g_c)^2. \tag{3.17}$$

In the terminology of signal theory, the complex cell output as formulated in Eq. 3.17 is a "signal power" in a certain band of spatial frequency defined by the parameters ω_x, ω_y, and σ of the Gabor functions g_s and g_c. We will come back to this notion in Chap. 4. The term "energy model" was suggested by Adelson and Bergen (1985) to stress the importance of spatiotemporal receptive fields in the processing of visual motion.

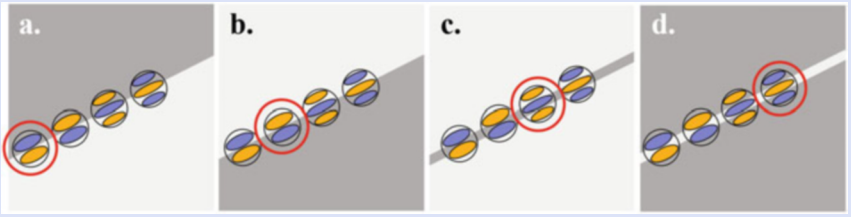
The energy model as depicted in Fig. 3.8 requires a highly regular connectivity in that four Gabor channels with phase offsets zero, a quarter, one half, and three quarters of a cycle are co-localized on the same visual field position and converge to one complex cell. Indeed, these requirements are needed to generate completely flat complex cell outputs. If, however, small variations in simple-cell phase or the exact shape of the nonlinearity are allowed, or if larger populations of simple cells converge on one complex cell, the model does not break down but produces complex cell outputs with small modulations superimposed on a flat baseline. This is indeed closer to the original findings by Movshon et al. (1978) who reported such combinations of modulated and unmodulated components in the complex cell responses. A population-based model of complex cells in the context of disparity tuning was presented by Burge and Geisler (2014).

Box 3.3 Edge Detection with Complex Cells

Consider four edges in an image, each with the same orientation, but differing in polarity and phase. Figures a and b show "step edges" where the intensity function is antisymmetric (odd) with respect to the location of the edge. Figures c, d show "contour edges" whose intensity function is symmetric (even) with respect to the edge location. In analogy to the difference between the symmetric cosine function and the antisymmetric 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

(continued)

Box 3.3 (continued)
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: (a) dark-to-bright (rightward increasing step edge), (b) bright-to-dark (rightward decreasing step edge), (c) dark-on-bright (decremental contour), and (d) bright-on-dark (incremental contour).



Also shown in the figures are four simple receptive fields modeled as Gabor functions that can be characterized as (i) increasing, odd (g_s in Eq. 3.11), (ii) decreasing odd ($-g_s$), (iii) decremental even ($-g_c$), and (iv) incremental even (g_c). All four profiles are shown for each edge, and the one profile yielding the strongest response is circled.

Complex cells combining the input of simple cells with odd and even receptive fields as shown in Fig. 3.8 will respond to all four edge types alike, while still keeping the specificity of orientation. That is to say, their reaction is invariant with respect to edge phase and polarity.

It is interesting to note that polarity invariance cannot be achieved by a linear filter. Rather, it is an intrinsically nonlinear property as can be seen from the following consideration: Let $I_+(x)$ be a bright-on-dark ("incremental") contour edge as depicted in figure d. Let the response of some linear receptive field to this stimulus be denoted by a_+. The decremental, dark-on-bright contour can then be written as $I_-(x) = I_o - I_+(x)$, where I_o is a constant; the unit's response to this stimulus is called a_-. Due to the assumed linearity, this response must be the difference between the responses to the constant intensity I_o and to I_+. Since neurons will generally not respond to constant stimuli at all ($\int \phi(x, y)dxdy \approx 0$), we obtain $a_- = -a_+$. With nonnegative activities a, this can be satisfied only if the neuron is not responding at all: $a_+ = a_- = 0$. Therefore, polarity-invariant neurons must be nonlinear.

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 $f_e + if_o$, where $i = \sqrt{-1}$ is the imaginary 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 one and vice versa. In fact, the odd and even Gabor functions only approximately satisfy this condition.

▶ **Key Point: Energy Model of the Complex Cell** Complex cells in the visual cortex are tuned to edge orientation and scale, but not to phase and polarity. They thus generate an invariance for exact position and treat incremental and decremental step and contour edges alike. This is achieved by combining the squared output of two simple cells with a phase offset of $\pi/2$.

3.4 Motion Detection

3.4.1 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 parameterized as speed and direction or as the x- and y-components of the motion vector. If we denote the local motion vector by $\mathbf{v} = (v_x, v_y)^\top$, image change due to visual motion can be expressed by the equation

$$I(x, y, t + dt) = I(x - v_x dt, y - v_y dt, t), \tag{3.18}$$

where $\mathbf{v}dt$ is the motion displacement in the time interval dt. The motion vector is defined for every pixel and may vary with time; it thus constitutes a vector field $\mathbf{v}(x, y, t) = (v_x(x, y, t), v_y(x, y, t))^\top$.

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 Eq. 3.18, we write

$$I(x, y, t + dt) = I(x, y, t) + f(x, y, t)dt, \tag{3.19}$$

where f denotes the flicker; it is again defined for every pixel and instant in time. In fact, $f(x, y, t)$ is the partial derivative of I with respect to time.

By inspection of Eqs. 3.18 and 3.19, it is clear that the measurement of visual motion amounts to 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 the two stimulus parameters speed and

direction, rather than solely 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. In order to represent different motion events, this type of coding requires the presence of many neurons tuned to different velocities and motion directions; it is known as labeled line coding and will be discussed further in Chap. 7. Motion detection is different from a (hypothetical) motion estimator, or two-dimensional speedometer, 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.

3.4.2 Coincidence Detector

The most obvious approach to motion detection is to combine a thresholding operation with a delay in one of the two input lines of the detector (Fig. 3.9a). 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 at the output neuron 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

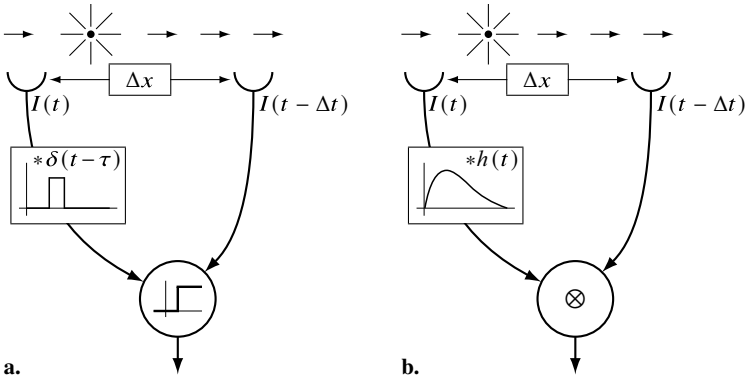


Fig. 3.9 (**a**) Simple motion detector consisting of two input lines of which one is delayed by the time lag τ, and a threshold. The delay is shown as a temporal convolution with a displaced δ-pulse. The stimulus is moving from left to right with a velocity $v = \Delta x/\Delta t$, leading to the inputs $I(t)$ 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 low-pass 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 a signed motion output

has been suggested as a model of motion selectivity in rabbit retinal ganglion cells by Barlow and Levick (1965).

In order to prevent the circuit from responding to constant white input on both lines, an appropriate preprocessing has to be 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 that can be expressed as a convolution (see Box 2.5). Its impulse response is a delayed delta impulse, $\delta(t - \Delta t)$. Using the definition of the δ pulse, Eq. 2.12, we may write

$$\int I(t')\delta(t - \Delta t - t')\, dt' = I(t - \Delta t). \tag{3.20}$$

In Fig. 3.9a, the delay operation is illustrated by its impulse response.

The delayed coincidence 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.

Coincidence detection is a general element of neural processing found also in other sensory systems. An example not related to the detection of motion but to directional hearing is the delay-line system in the barn owl auditory system (Carr and Konishi 1990). Signals from the two ears converge on populations of brainstem neurons via axons of various delays. A sound arriving from a given direction will be received by the two ears with an interaural delay depending on the direction of arrival, the distance between the two ears, and the speed of sound. As the direction changes, exact coincidence of the signal from the two ears will move from neuron to neuron in the brainstem system. With this mechanism, the barn owl can determine interaural time differences in the order of tens of microseconds (10^{-5} s) and locate rustling prey even in the dark.

3.4.3 Correlation Detector

An alternative design for visual motion detection is shown in Fig. 3.9b, where the delay is replaced by a temporal low-pass filter and the threshold is replaced by a correlation; it is known as the correlation or Reichardt[9] detector, (see Borst and Groschner 2023). At any instant in time, the output of the temporal low-pass is a mixture of its inputs at the preceding times, weighted by the temporal impulse response function $h(t)$ depicted in Fig. 3.9b. It thus acts like the superposition of a whole set of weighted delay lines. Therefore, the comparison operation in the subsequent neuron will respond 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.

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

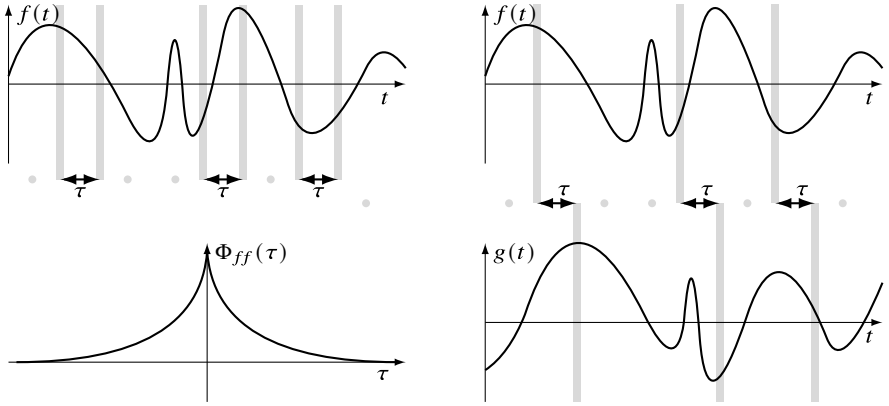


Fig. 3.10 **Left:** The autocorrelation of a function f with offset or lag τ is calculated by evaluating the function at all pairs of points with separation τ. Of the resulting pairs $(f(t), f(t+\tau)$ for all possible positions t), "correlation" is calculated by multiplying the values in each pair and summing the results. Since this procedure can be repeated for each lag τ, a so-called autocorrelation *function* results, which is denoted $\Phi_{ff}(\tau)$. A typical autocorrelation 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, $\Phi_{fg}(\tau)$

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 an extended period of time. To understand this, we have to briefly introduce the notion of auto- and cross correlation of functions, see Fig. 3.10 and Box 3.4. In statistics, sample correlation for a set of paired data $(x_i, y_i)_{i\in 1,..,n}$ is defined as

$$\mathrm{cor}(x, y) := \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sqrt{\mathrm{var}(x)\ \mathrm{var}(y)}}, \qquad (3.21)$$

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., assume $\bar{x} = \bar{y} = 0$) and omit the division by the variances; what remains is simply the sum of the products of the data pairs.[10] 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 or "lag" τ. For multiple samples 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,

[10] 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.

the sum in Eq. 3.21 is to be replaced by an integral, and we obtain the autocorrelation of f for time lag τ as

$$\Phi_{ff}(\tau) = \frac{1}{2T}\int_{-T}^{T} f(t)\,f(t+\tau)dt. \tag{3.22}$$

The interval $[-T, T]$ is normally chosen to include the available sample of data. Values of $f(t+\tau)$ where $t+\tau$ falls outside this interval are patched with zeros. For some applications, we can also consider the limit for $T \to \infty$ in which case the integral will become improper.

In comparison to the statistical notion of correlation, we have replaced a population average by a temporal average in Eq. 3.22. Random processes for which the two averages are indeed the same are called "ergodic." We will assume here that ergodicity obtains.

Autocorrelation as defined in Eq. 3.22 can be calculated for each lag value $-T \leq \tau \leq T$; thus Φ_{ff} becomes a function of the lag. It is easy to see that $\Phi_{ff}(\tau)$ takes its maximum value at $\tau = 0$ and that $\Phi_{ff}(-\tau) = \Phi_{ff}(\tau)$.

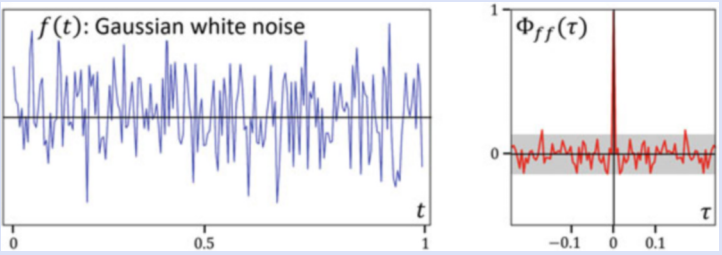
The cross correlation function is defined by the same logic, except that the two values of the paired variables are now taken from different functions; see Fig. 3.10. For two functions f, g, it is defined as

$$\Phi_{fg}(\tau) = \frac{1}{2T}\int_{-T}^{T} f(t)\,g(t+\tau)dt. \tag{3.23}$$

Cross correlation is a means to detect delays or 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. The position of this peak does not depend on the normalizations and the zero-mean assumption, which justifies the simplified definitions of Eqs. 3.22 and 3.23.

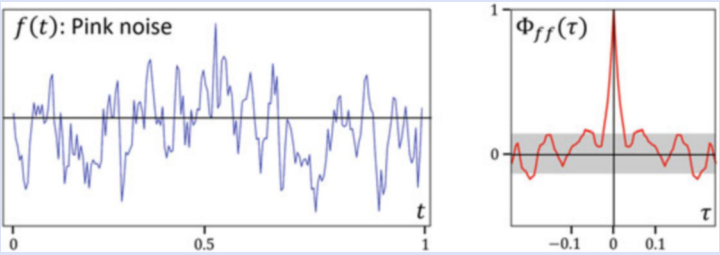
Box 3.4 Autocorrelation Examples

Here we consider a few examples for autocorrelation functions of typical signals. All autocorrelation functions shown here are normalized to $\Phi_{ff}(0) = 1$.

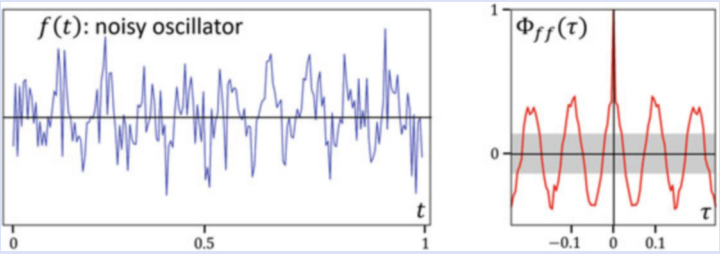


(continued)

Box 3.4 (continued)
The Gaussian white noise signal is obtained by randomly drawing at each time step from a normal distribution (blue curve). Its autocorrelation function has a sharp peak at lag zero since for all nonzero lags τ, the data pairs $f(t)$, $f(t+\tau)$ are uncorrelated and the expected correlation will be zero. The gray band shows the margin within which deviations from zero are not statistically significant.



A "pink noise" process can be generated by the rule $f(t+1) = \alpha f(t) + (1-\alpha)\mathbf{n}$, where α is a decay factor (0.75 in the example) and $\mathbf{n}$ is again drawn randomly from a normal distribution. Nearby values of f keep some correlation, which leads to a wider peak in Φ_{ff}.



Oscillations hidden in noise can be made visible by autocorrelation. For lags that are integer multiples of the cycle length, Φ_{ff} will show peaks. The smallest positive lag at which a peak occurs (0.1 in the example) is the cycle length of the oscillation.

We now turn back to the problem of motion detection. In the correlation-type motion detector, the output neuron multiplies its two inputs and accumulates this product over time, i.e.,

$$a = \int \underbrace{\int h(t')I(t-t')dt'}_{(h*I)(t)} \; I(t-\Delta t)dt. \tag{3.24}$$

It thus evaluates the similarity between the two signals that is generated by low-pass filtering for larger time delays. Formally, we can summarize the response of the correlation detector to a stimulus moving with velocity v as

$$a_v = (\Phi_{II} * h)\left(\frac{\Delta x}{v}\right), \tag{3.25}$$

where Δx is the separation between the two input lines as shown in Fig. 3.9b and $\Delta t = \Delta x/v$. This is to say, the output of the motion detector is given by convolving the autocorrelation function of the input with the impulse response h and evaluating the result at Δt. If the image is a white noise pattern, i.e., a noise pattern where adjacent pixels are uncorrelated, its autocorrelation function is a δ-pulse, $\Phi_{II}(\tau) = \delta(\tau)$, see Box 3.4. The tuning curve for velocity as defined in Eq. 7.1 will then evaluate to $f(v) = h(\Delta x/v)$.

Auto- and cross correlation are used not only in models of motion detection but also in other contexts. For example, Simmons (1979) suggested that bats determine the time lag between a call and the returning echo by a cross correlation technique. This allows them to sense changes in echo delays as short as 0.5 µs, which corresponds to distance variations of less than a millimeter. In the human auditory system, autocorrelation is thought to underlie the perception of the pitch of a complex sound: The perceived pitch frequency corresponds to the first peak (the peak with the smallest positive time lag) in the autocorrelation function. In vision, the autocorrelation of the image function can be used to determine the optimal range of lateral interaction in receptive fields (Srinivasan et al. 1982). As an example from data analysis, we mention the study by Hafting et al. (2005) who used the spatial autocorrelation to demonstrate the regular spacing of grid cell firing fields.

3.4.4 Motion as Orientation in Space-Time

The standard model of motion selectivity in the mammalian visual cortex (Adelson and Bergen 1985) can be derived from the "bi-local" detectors shown in Fig. 3.9 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 Fig. 3.11). 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 of the neuron. Figure 3.11b shows an alternative depiction of the same idea where the circuit of the triple coincidence detector is now presented as a receptive field function in a space-time diagram. Space-time representations of spatiotemporal receptive fields have already been used in Fig. 3.3; as in this diagram, the spatial coordinate x is shown as a horizontal axis and time as the vertical axis. It extends backward from the moment of measurement that is defined as $t = 0$. 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 Fig. 3.11b

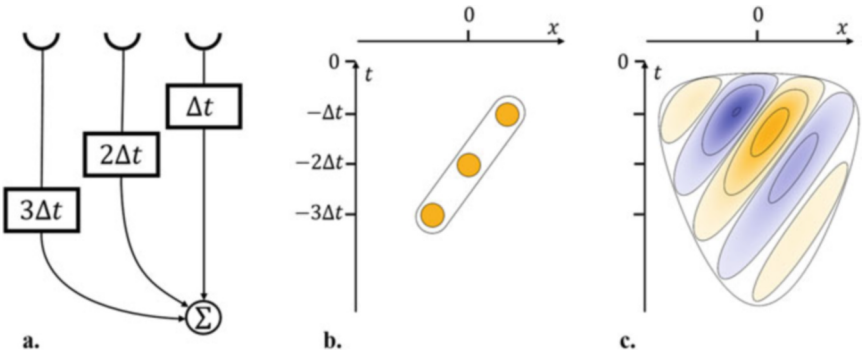


Fig. 3.11 Motion selective receptive fields. (**a**) A hypothetical neuron (triple coincidence detector) with three equally spaced input sites (spacing Δx), each with a different temporal delay (rectangular boxes). (**b**) (x, t)-diagram of the spatiotemporal receptive field of this neuron. The potential u will be maximal if a stimulus is jumping along the three input sites of the neuron, or moving with speed $\Delta x/\Delta t$. (**c**) (x, t)-diagram of a spatiotemporal Gabor function combined of a spatiotemporal wave (Eq. 3.26) and the causal window function from Eq. 3.27. It models the receptive field of a motion selective simple cell. The orientation in the space-time plot corresponds to the preferred velocity

would show as a bright bar covering the three spots. Velocity in the space-time diagram thus corresponds to slope, or spatiotemporal orientation.

The space-time diagram of Fig. 3.11b can also be interpreted as a spatiotemporal receptive field function as discussed already in Sect. 2.2.6. 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 spatiotemporal receptive field results as is shown in Fig. 3.11c.

The profile shown in Fig. 3.11c is a spatiotemporal Gabor function defined pretty much like the two-dimensional Gabor function in Eq. 3.11. The wave component is given by

$$f(x, y, t) = \cos(\omega_x x + \omega_y y - \nu t) \tag{3.26}$$

just as the drifting grating given by Eq. 3.12. Again, the preferred velocity would be given by $\nu(\omega_x^2 + \omega_y^2)^{-1/2}$. The envelope is more difficult to model since a Gaussian window in time will always violate causality. Here, we follow a suggestion by Koenderink (1988) of using a logarithmic transformation to expand the half axis of past time, $(-\infty, 0]$, to the full real axis. A spatiotemporal receptive field with a peak response delay of $t_d > 0$ will then extend asymmetrically in the two directions of time but never leave the range of causality. For a receptive field centered at $(x, y, t) = (0, 0, t_d)$ the window function becomes

$$w(x, y, t) = \exp\left\{-\frac{1}{2}\left(\frac{x^2 + y^2}{\sigma_s^2} + \frac{(\log(-t) - \log(-t_d))^2}{\sigma_t^2}\right)\right\}, \tag{3.27}$$

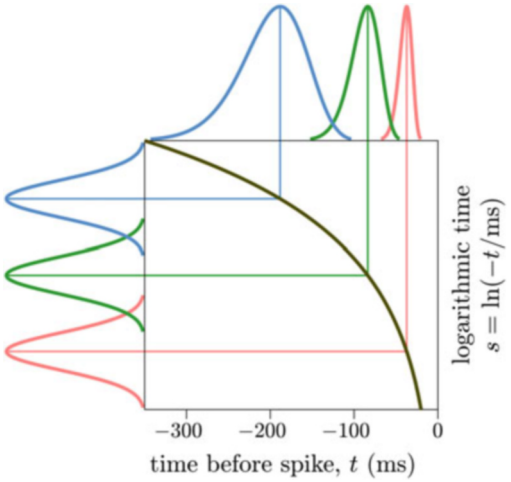


Fig. 3.12 Scale-time model of causal spatiotemporal receptive fields. The top curves show the temporal components $w(0, 0, t)$ of the window function from Eq. 3.27 for three values of t_d. They are generated by ordinary Gaussians with width σ_t in logarithmically compressed time (left curves). The asymmetric window function is used in the full spatiotemporal receptive field shown in Fig. 3.11c

see Fig. 3.12. The variables σ_s and σ_t are spatial and temporal scales, i.e., they control the width of the envelope. From the scale-time model, the full spatiotemporal receptive field function is obtained by multiplication with the plane-wave component, $g = f \times w$; an example appears in Fig. 3.11c. It describes the response behavior of a simple cell tuned for motion (rightward), scale, spatial frequency, location, phase (cosinusoidal), and polarity. Preferred speed is slower, if the slope of the contour lines is steeper. Visual motion is therefore said to correspond to spatiotemporal orientation. 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 (Sect. 3.3) 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 Fig. 3.13d. As a result, the response will be invariant to phase and polarity, while the specificities for velocity, spatial frequency, orientation, and position remain.

► **Key Point: Visual Motion Detection** The simplest detector for visual motion is the delayed coincidence detector. If many input lines with spatiotemporal receptive fields are considered, it generalizes to a detector for oriented edges in space-time. This can be realized by spatiotemporal Gabor filters and combined in the energy scheme.

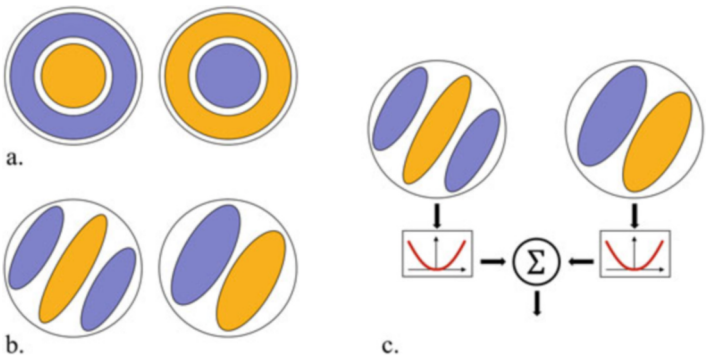


Fig. 3.13 Visual receptive fields. (**a**) Isotropic (rotationally symmetric) types of the retina and LGN modeled as Difference of Gaussians. (**b**) Oriented simple cells of the visual cortex modeled as even (cosine) and odd (sine) Gabor function. (**c**) Cortical complex neuron responsive to "contrast energy"

3.5 Summary and Further Reading

1. 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 or scale (Fig. 3.13a).
2. 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 (Fig. 3.13b).
3. 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 nonlinearity that is described by the energy model (Fig. 3.13c).
4. Visual motion is among the strongest stimuli to which cortical neurons react. Receptive fields of motion (direction) selective neurons are oriented in space-time. They may be of the simple or the complex type.

Texts

Frisby and Stone (2010): *Modern textbook of visual perception covering the computational topics largely on an informal level*

DeValois and DeValois (1988): *Treatise of receptive field theory as lateral interaction. Relies heavily on Fourier theory*

Mallot (2000): *Describes the computational theory of early vision, including edge detection, shape-from-shading, color, motion, and stereo*

Marr (1982): *Foundational text on the computational theory of vision*

Rolls and Deco (2002): *Computational account of neural networks in the visual system with a focus on visual attention*

Suggested Original Papers for Classroom Seminars

Adelson and Bergen (1985): *Fundamental account of visual motion as spatiotemporal orientation. Derives suitable spatiotemporal filters for motion detection and suggests biologically plausible versions constructed from spatiotemporally separable filters.*

Itti and Koch (2000): *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.*

Jones and Palmer (1987): *The receptive field profiles of cortical simple cells are measured and fitted with Gabor functions. Results show the high quality of the fits if parameters for spatial frequency, orientation, scale, aspect ratio, and phase are optimized.*

Marr and Hildreth (1980): *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 belongs to the foundations of the field of computational vision.*

Ohzawa (1998): *Extension of the energy model to the detection of binocular disparity. For a more advanced example of this approach, see also Burge and Geisler (2014).*

References

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

Barlow, H. B., and R. W. Levick. 1965. The mechanism of directional selectivity in the rabbit's retina. *Journal of Physiology* 173: 477–504.

Borst, A., and L. N. Groschner. 2023. How flies see motion. *Annual Review of Neuroscience* 46: 17–37.

Burge, J., and W. S. Geisler. 2014. Optimal disparity estimation in natural stereo images. *Journal of Vision* 14(2): 1.

Carr, C. E., and M. Konishi. 1990. A circuit for detection of interaural time differences in the brain stem of the barn owl. *Journal of Neuroscience* 10: 3227–3246.

Chichilnisky, E. J., and R. S. Kalmar. 2002. Functional asymmetries in ON and OFF ganglion cells of primate retina. *The Journal of Neuroscience* 22: 2727–2747.

Croner, L. J., and E. Kaplan. 1995. Receptive fields of P and M ganglion cells across the primate retina. *Vision Research* 35: 7–24.

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

DeValois, R. L., and K. K. DeValois. 1988. *Spatial Vision*. Number 14 in Oxford Psychology Series. Oxford: Oxford University Press.

Enroth-Cugell, C., and R. M. Shapley. 1973. Adaptation and dynamics of cat retinal ganglion cells. *Journal of Physiology* 233: 271–309.

Ferster, D., and K. D. Miller. 2000. Neural mechanisms of orientation selectivity in the visual cortex. *Annual Review of Neuroscience* 23: 441–471.

Frisby, J. P., and J. V. Stone. 2010. *Seeing: The Computational Approach to Biological Vision*. 2nd ed. Cambridge, MA: The MIT Press.

Hafting, T., M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser. 2005. Microstructure of a spatial map in the entorhinal cortex. *Nature* 436: 801–806.

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

Hubel, D. H., and T. N. Wiesel. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology* 160: 106–154.

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

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

Koenderink, J. J. 1988. Scale-time. *Biological Cybernetics* 58: 159–162.

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

Marčelja, S. 1980. Mathematical description of the responses of simple cortical cells. *Journal of the Optical Society of America* 70: 1297–1300.

Marr, D. 1982. *Vision. A Computational Investigation into the Human Representation and Processing of Visual Information*. New York: W. H. Freeman.

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

Masland, R. H. 2001. The fundamental plan of the retina. *Nature Neuroscience* 4: 877–886.

Movshon, J. A., J. D. Thompson, and D. J. Tolhurst. 1978. Receptive field organization of complex cells in the cats striate cortex. *Journal of Physiology* 283: 79–99.

Ohzawa, I. 1998. Mechanisms of stereoscopic vision: The disparity energy model. *Current Opinion in Neurobiology* 8: 509–515.

Pollen, D. A., and S. F. Ronner. 1981. Phase relationships between adjacent simple cells in the visual cortex. *Science* 212: 1409–1411.

Rodieck, R. W. 1965. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research* 5: 583–601.

Rodieck, R. W., and J. Stone. 1965. Analysis of receptive fields of the cat retinal ganglion cells. *Journal of Neurophysiology* 28: 833–849.

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

Sillito, A. M. 1975. The contribution of inhibitory mechanisms to the receptive field properties of neurons in the striate cortex of the cat. *Journal of Physiology* 250: 305–329.

Simmons, J. A. 1979. Perception of echo phase information in bat sonar. *Science* 204: 1336–1338.

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

Wässle, H., and B. B. Boycott. 1991. Functional architecture of the mammalian retina. *Physiological Reviews* 71: 447–480.

Werblin, F. S., and J. E. Dowling. 1969. Organization of the retina of the mudpuppy *Necturus maculosus*. II. Intracellular recording. *Journal of Neurophysiology* 32: 339–355.

Fourier Analysis for Neuroscientists

4

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 chapters 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 presented in three steps: In the first step, we will demonstrate that sinusoidal inputs to linear, shift-invariant (LSI) 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 "eigenfunctions" of LSI systems. In the second step, we show that most functions can be represented as linear combinations of sine and cosine functions of various frequencies; these may be conveniently written as complex exponentials. Both ideas combine in the third step: that is, the convolution theorem, which states that the convolution of two functions can also be expressed as the simple multiplication of the respective Fourier transforms. This is also the reason why linear shift-invariant systems are often described as "filters" removing some frequency components from a signal and passing others.

H. A. Mallot, *Computational Neuroscience*,
https://doi.org/10.1007/978-3-031-75705-1_4

Learning Objectives

- Applications of Fourier theory in various fields of science
- The eigenfunctions of linear shift-invariant (LSI) systems
- Fourier decomposition and transform pairs
- Complex exponentials as a convenient representation of sinusoidals
- The convolution theorem
- Fourier theory to two and more dimensions

4.1 Examples

4.1.1 Light Spectra

The notion of a spectrum in the physics of light is closely related to the ideas of the Fourier[1] transform. Light is an electromagnetic wave that can be described by the electric and magnetic fields as a function of space and time. Consider a beam of coherent light traveling in direction **x**. At a given instant in time, the corresponding fields are smooth functions of spatial position along that direction, x. In pure, or "spectral" colors, the field strengths oscillate according to a sinusoidal function of x, generally specified by its wavelength, λ, or by its frequency, ω. Figure 4.1a shows as thin lines the electric field distributions of a blue (short wavelength, high frequency) and a red (long wavelength, low frequency) light. 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 (Fig. 4.1a, heavy line). Note that this distribution is no longer a sinusoidal;

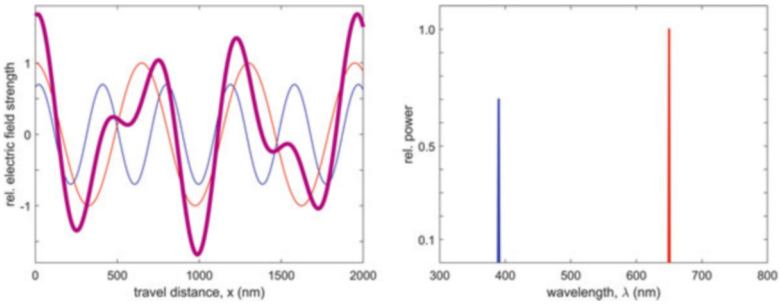


Fig. 4.1 (**a**) Distribution of electric field strength in two spectral light waves (thin blue and red sinusoidal curves) shown with random phase. The heavy magenta lines show their mixture or superposition. (**b**) Spectrum of the mixed color (magenta) consisting of two lines for the pure colors red and blue

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

it need not be periodic at all. Still, it can be thought of as being composed of two sinusoidal components with different frequencies. Figure 4.1b 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 pure colors red and blue, these are line spectra, i.e., the spectrum is zero except for the wavelengths $\lambda_r = 650$ nm or $\lambda_b = 390$ nm, respectively. The spectrum of the purple light is the combination of these two lines.

Figure 4.1b shows the spectrum as a function of wavelength. Alternatively, we could plot it as a function of frequency, which relates to the wavelength via the speed of wave propagation, in this case the speed of light (3×10^8 m/s). For the wavelengths used in the example, we obtain the frequencies 7.7 and 4.6×10^{14} Hz (blue and red, respectively). Plotting the spectrum as a function of frequency rather than of wavelength inverts the horizontal axis, while the pure color components are still represented as lines. The relation between the wave picture (Fig. 4.1a) and the spectral line picture (Fig. 4.1b) is an instance of the Fourier transform.

4.1.2 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 330 Hz. Figure 4.2a 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 Fig. 4.2b. In music, these patterns 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 Fig. 4.2b into the individual tones they are composed of. This is achieved by the complex hydrodynamics of the cochlea generating a so-called tonotopic pattern of stimulation in which high pitch activates basal parts of the cochlea (close to the oval window), while low pitch activates parts further up toward the helicotrema. In complex sounds, each pure (or "partial") tone, that is, each frequency of sound, stimulates is own section of the basilar membrane. Mathematically, this operation corresponds to a Fourier decomposition of the compound signal and results in an acoustic spectrum with peaks at the respective positions along the basilar membrane. The frequency axis realized by the basilar membrane approximates a logarithmic scale.[2]

The spectral structure is also reflected in the musical notation shown in Fig. 4.2c: The lines of the staff roughly represent logarithmic steps on a frequency axis running

[2] The fact that complex sounds are perceived as combinations of pure (or "partial") tones is also known as Ohm's second law, his first law being the relation of voltage and current at a resistor.

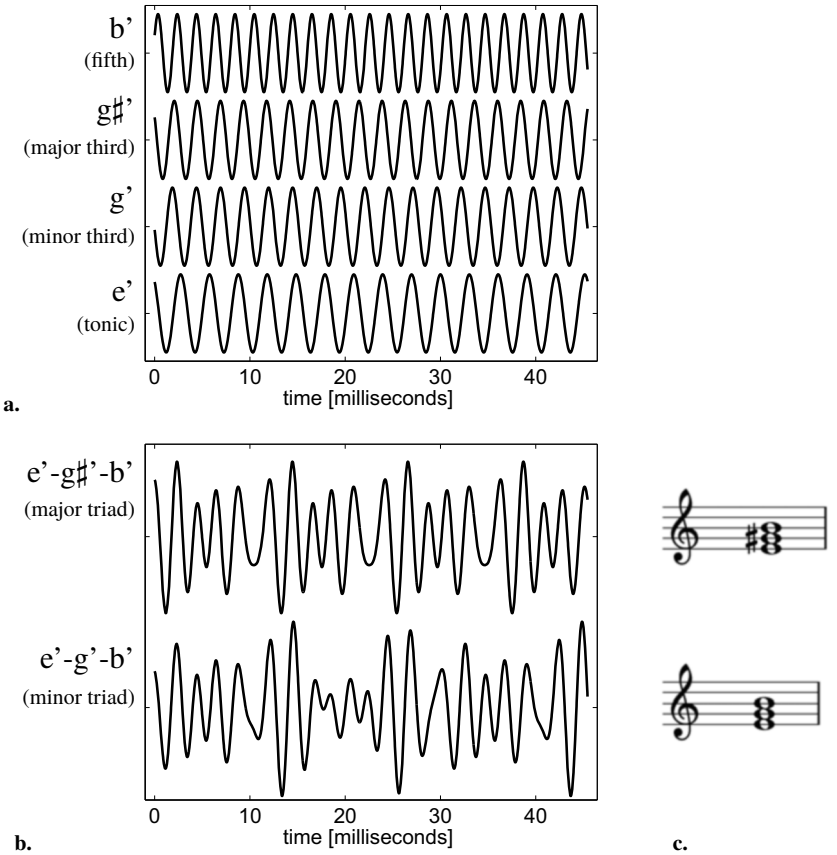


Fig. 4.2 Time and frequency signals in music. (**a**) Sound pressure of pure tone e' ($\omega_o = 330$Hz), the minor third g' ($\omega/\omega_o = 6/5$), the major third g♯' ($\omega/\omega_o = 5/4$), and the fifth b' ($\omega/\omega_o = 3/2$). The frequency ratios correspond to the just, or pure intonation. 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 that is akin to a logarithmic frequency scale

from bottom to top with the line circled by the clef as a reference frequency (g' = 392 Hz). The individual note symbols mark the spectral components.

When playing the same tone as before, e', on a piano rather than on a treble recorder, 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 a number of integer multiples of 330 Hz. These multiples are known as harmonics. They are not generally perceived as separate tones (a violation of Ohm's second law) but generate the "timbre" of the sound, i.e., the differences between productions of the same tone by various musical instruments or human

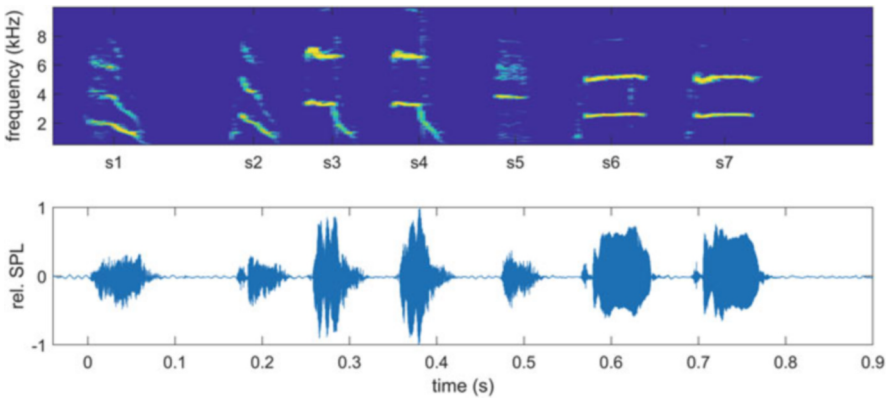


Fig. 4.3 Spectrogram of a song sequence of the Bengalese finch (*Lonchura striata domestica*). The bottom plot shows the logarithmic microphone signal (SPL = sound pressure level). Individual oscillations of the sound wave are in the order of several kHz and are not resolved in the plot. The upper part shows the spectrogram as a density plot where warm colors indicate high signal power. The sequence is composed of seven syllables marked s1 to s7. The syllables s6 and s7 show two main frequency components at about 2.8 and 5.6 kHz, i.e., in frequency ratio 1 : 2. In syllables s1 and s2, three such frequency bands are visible, again in integer ratio 1 : 2 : 3. Over time, the fundamental frequency moves to lower pitch. This is even clearer in syllables s3 and s4 that show a strong frequency modulation downward in their lower band. The wide rectangular structure in the upper part of syllable s5 is not tonal, but a noise component. Song data and plots kindly provided by Lena Veit, Tübingen

voices. Complex patterns 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.[3]

Complex acoustical events such as uttered speech or bird song are not easily decomposed into tones but may be represented as spectrograms, see Fig. 4.3. To compute a spectrogram, the time-dependent signal is decomposed into short, overlapping sections, which are obtained by multiplying the original signal with a Gaussian window function centered at discrete time steps. Within each window, the Fourier transform is computed, and the power of each frequency component is plotted as a column of color-coded intensity values in a coordinate system spanned by time (the center time of each window) and frequency. The result is a time–frequency picture of sound intensity. For a simple tune, it will show a sequence

[3] Pitch perception itself is not based on Fourier decomposition. If the fundamental frequency is removed from a tone with rich harmonics (missing fundamental stimulus), the perceived pitch is unchanged. Perceived pitch then corresponds to the periodicity of the sound that can be determined using the autocorrelation function described in the previous chapter (see Box 3.4). Neurons responsive to pitch in missing fundamental stimuli have been reported in the monkey auditory cortex by Bendor and Wang (2005).

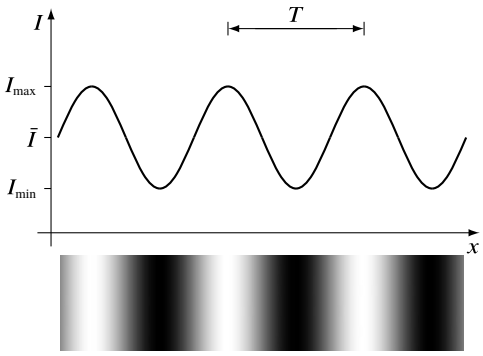


Fig. 4.4 Sinusoidal grating of the form $I(x, y) = \bar{I} + 0.5\,(I_{max} - I_{min})\sin(2\pi x/T)$, where $\bar{I} = 0.5\,(I_{max} + I_{min})$. Such gratings with various period lengths T and contrasts are used for measuring of the acuity of vision

of high-intensity lines corresponding to the frequency and duration of each tone. In speech or bird song, many frequencies will be present simultaneously, resulting in more complicated spectrograms.

4.1.3 Spatial Vision

The representation of signals by sinusoidals and frequencies is rather intuitive in the cases of colors and tones. In images, conceptualized as 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 coarse gratings and thus need less storage space than image regions with fine-grain contrasts.

Figure 4.4 shows a one-dimensional sinusoidal intensity variation over an image coordinate x. It is characterized by a wavelength, or alternatively by a (spatial) frequency measured in cycles per degree of visual angle. Since image intensities cannot be negative, gratings will always be offset above zero on the intensity axis. Instead of characterizing mean and amplitude, the strength of modulation is often defined using the so-called Michelson[4] contrast,

$$\text{contrast} := \frac{I_{max} - I_{min}}{I_{max} + I_{min}}. \tag{4.1}$$

Contrast is a dimensionless quantity taking values in the interval [0, 1]. Contrast 0 means that $I_{max} = I_{min}$, irrespective of the absolute intensity values. Contrast is 1 if $I_{min} = 0$, which is, however, hard to achieve. The intensity halfway between I_{min}

[4] Albert A. Michelson (1852–1931), American physicist. Nobel prize in Physics 1907.

and I_{max} in terms of contrast is the geometric mean $\sqrt{I_{min} I_{max}}$. This relation underlies the well-known γ-correction for video screens, in which the gray levels (usually expressed as integers from 0 to 255) are equidistant in contrast, not in intensity. Contrast, not intensity, is also the quantity driving visual cortical neurons (Albrecht et al., 2002).

An important characteristic of the visual system is resolution measured as the sensitivity to gratings of varying spatial frequency. Contrast sensitivity is defined as the inverse of the minimal image contrast needed to perceive a grating of a given spatial frequency. In humans, contrast sensitivity has a maximum at about 5–10 cycles per degree (cpd) depending on absolute light intensity and drops for both lower and higher frequencies (Campbell and Green, 1965). The value 5–10 cpd marks the coarseness at which image structure is best perceived. The plot of contrast sensitivity versus spatial frequency is now known as the Campbell[5] curve. It can be considered the absolute value of the modulation transfer function (MTF, see Sect. 4.2.2) of the early visual system.

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. Fig. 3.5a). Images may then be decomposed into two-dimensional gratings of various frequencies. Figure 4.5 shows patches 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 different gratings result for each choice of the variables ω_x and ω_y, the spectrum becomes a two-dimensional function of the variables ω_x, ω_y specifying the amplitude for each component grating.

4.1.4 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., $H_o = 1.5\,\mathrm{T}$) 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, i.e., the frequency at which resonance occurs, are proportional to the basic field strength H_o experienced by each proton and a material constant called the gyromagnetic ratio, γ. For protons, γ lies in the range of 43 MHz/T.

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

[5] Fergus W. Campbell (1924–1993), British vision scientist.

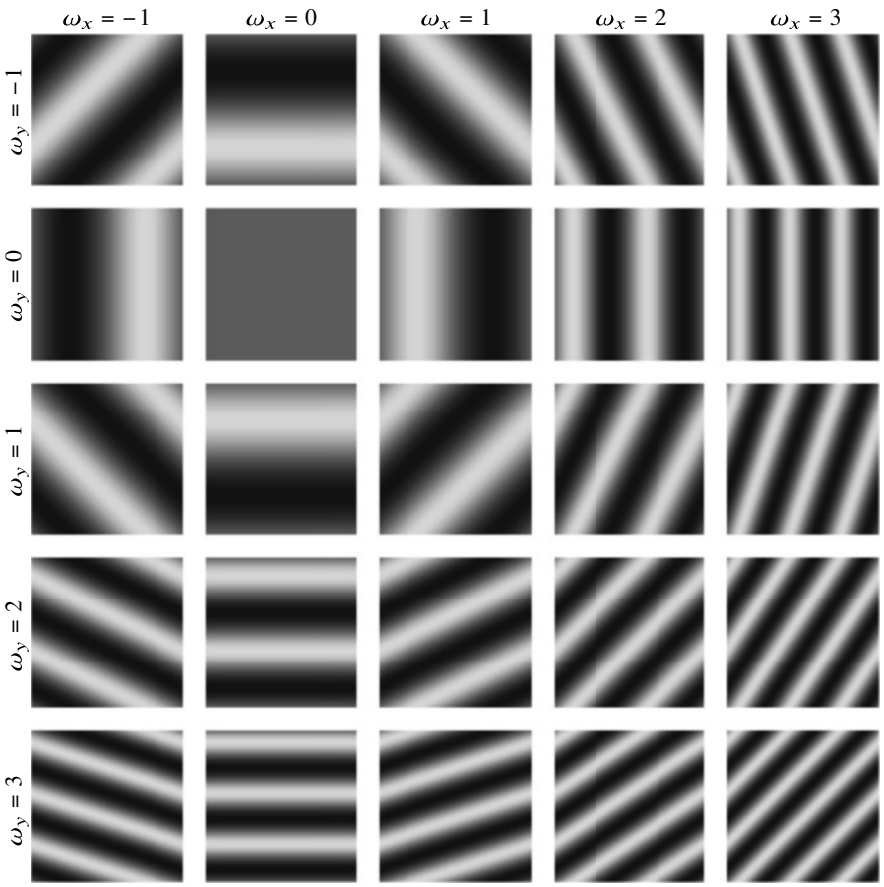


Fig. 4.5 Two-dimensional gratings of the form $I(x, y) = \sin(2\pi(\omega_x x + \omega_y y))$ for some integer values of ω_x and ω_y. Each square shows the patch $(x, y) \in [0, 1] \times [0, 1]$

now applied, only the protons at a particular z-location will satisfy the resonance condition. These protons fill a plane perpendicular to the axial direction, i.e., a slice of the investigated volume. After the activation, another 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 in the slice have frequencies proportional to the total locally "perceived" field strength. Signals emitted from voxels with different x-coordinates 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.

▶ **Key Point: Fourier Analysis** In Fourier analysis, functions of space or time are reexpressed as (transformed into) functions of spatial or temporal frequency. The latter functions are said to reside in the "Fourier domain," while the former ones are in the spatial or temporal "domain," respectively. Fourier transforms are also called spectra, after the decomposition of light into "spectral" or single-wavelength colors.

4.2 Why Are Sinusoidals Special?

4.2.1 Eigenfunctions

The examples given above make it clear that sinusoidal[6] functions are used in many contexts, but the underlying mathematical rational needs further explanation. In this section, we will present an important relation between linear shift-invariant (LSI) systems and sinusoidals. A system is conceptualized as a mapping, which assigns an output function to each input function. Using the notation already introduced in Box 2.2, we write

$$h = \mathcal{M}(f), \tag{4.2}$$

where h and f are functions such as the activity distribution on a layer of neurons and input image and the retinal input image. The arguments of these functions are omitted. Mappings such as $\mathcal{M}$, which assign functions to functions, are also called operators (see Box 2.2). An important characteristic of the operator $\mathcal{M}$ is its set of eigenfunctions.[7] These are the functions satisfying the equation

$$\mathcal{M}f = \lambda f, \tag{4.3}$$

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

Assume now that $\mathcal{M}$ is a convolution or linear shift-invariant (LSI) system as introduced in Chap. 2. We will show in this section that the sinusoidal functions are eigenfunctions of LSI systems. Before we give a formal account of the problem, we note that an LSI system will always keep the periodicity of its input in the produced output. If the input to a system repeats with a shift, or wavelength, $(\Delta x, \Delta y)$,

[6] The term "sinusoidal" is used here as an adjective or noun to denote sines, cosines, or sines with arbitrary phase, wavelength, and amplitude alike.

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

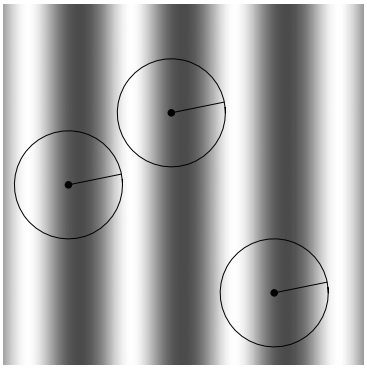


Fig. 4.6 Convolution of a periodic image function with a circular receptive field function. The periodic image function is shown in the background. At three locations, the local kernels for the convolution are symbolized by the circles. Clearly, the images within the circles are identical. Therefore, 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 result is an immediate consequence of the shift invariance of convolution

accordingly spaced kernels applied to the input will see the same input and will produce equal output due to shift invariance. The situation is illustrated in Fig. 4.6. Sinusoidals are therefore obvious candidates for the eigenfunctions of convolution.

4.2.2 The Eigenfunctions of Convolution: Real Notation

We will develop the mathematical argument for functions of only one real variable. Results for functions with two and more variables will be presented later. Let $g(x)$ be a point-spread function or convolution kernel of an LSI system. We write the convolution operation applied to a sine function with frequency ω as $(g * \sin_\omega)$ and note that this is a function (the output function of the system) that can be evaluated at variable values x. From the definition of convolution, Eq. 2.17, we have

$$(g * \sin_\omega)(x) = \int_{-\infty}^{\infty} g(x') \sin(\omega(x - x'))\, dx', \tag{4.4}$$

where x' is an auxiliary variable that cancels out as the integral is computed. Although in Eq. 4.4 we use the spatial variable x, the argument works just as well in the temporal domain.

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 g and the sine (commutativity of convolution, see Eq. 2.18). In the present form, we may apply the addition theorem $\sin(\alpha - \beta) = \sin\alpha\ \cos\beta - \cos\alpha\ \sin\beta$ and obtain:

$$(g * \sin_\omega)(x) = \int g(x')(\sin\omega x\ \cos\omega x' - \cos\omega x\ \sin\omega x')\, dx'. \tag{4.5}$$

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

$$(g * \sin_\omega)(x) = \sin \omega x \int g(x') \cos \omega x' \, dx' - \cos \omega x \int g(x') \sin \omega x' \, dx'$$
$$= \tilde{g}_c \sin \omega x - \tilde{g}_s \cos \omega x. \tag{4.6}$$

Here we have used the fact that after moving the variable x out of the integrals, the remainders evaluate to constants, for which we introduced the names

$$\tilde{g}_c := \int g(x) \cos \omega x \, dx \tag{4.7}$$

$$\tilde{g}_s := \int g(x) \sin \omega x \, dx. \tag{4.8}$$

Similarly, one can show that

$$(g * \cos_\omega)(x) = \tilde{g}_s \sin \omega x + \tilde{g}_c \cos \omega x; \tag{4.9}$$

we leave this proof to the reader.

Taken together, the results show that the convolution of a sine with any kernel g is a weighted sum of a sine and a cosine with the frequency of the input signal. A linear shift-invariant system cannot change the frequency of a sinusoidal signal.

Equation 4.6 is not yet the full answer to the eigenfunction problem, since the response to a sine is not again a sine but 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:

$$A := \sqrt{\tilde{g}_c^2 + \tilde{g}_s^2}, \quad \cos \phi = \frac{\tilde{g}_c}{A}, \quad \sin \phi = \frac{\tilde{g}_s}{A}, \tag{4.10}$$

and obtain

$$(g * \sin_\omega)(x) = \sqrt{\tilde{g}_c^2 + \tilde{g}_s^2} \, (\cos \phi \, \sin \omega x - \sin \phi \cos \omega x) \tag{4.11}$$
$$= A \sin(\omega x - \phi). \tag{4.12}$$

A geometrical interpretation of the relations between the quantities $\tilde{g}_c$, $\tilde{g}_s$, A, and ϕ is given in Fig. 4.7.

Equation 4.12 is as close as we can get to the sought eigenfunction equation, at least with our present mathematical devices: When passed through a linear shift-invariant system, sinusoidals are attenuated ($|A| < 1$) or amplified ($|A| > 1$, for energy-consuming systems only) and shifted in phase. Both amplitude modulation

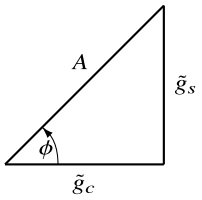


Fig. 4.7 Geometrical interpretation of the quantities $\tilde{g}_s$, $\tilde{g}_c$, A, ϕ, and their relations

and phase shift depend on the frequency of the sinusoidal, but the frequency itself cannot be changed. Both effects, attenuation and phase shift, can be formally combined in one factor or eigenvalue if complex number theory is used. We will explain the complex notation in the next section.

4.2.3 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 contrast, the similar equation $x^2 + 1 = 0$ has no solution inside the set of real numbers. If we wanted to construct a "number" set, in which a solution exists, it would have to contain the elements $\sqrt{-1}$ and $-\sqrt{-1}$ that solve the equation $x^2 + 1 = 0$ if the ordinary rules of calculation are formally applied. We introduce the notation

$$i = \sqrt{-1} \tag{4.13}$$

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

$$z := a + ib \tag{4.14}$$

a complex number with real part $\Re(z) = a$ and imaginary part $\Im(z) = b$. The "fundamental theorem of algebra" states that every algebraic equation (i.e., every equation of the form $\sum_{k=0}^{n} a_k x^k = 0$) has exactly n solutions (or "roots") in the set of complex numbers, where multiple roots are counted by their multiplicity. In the set of real numbers, it may have anything between 0 and n solutions.

Some basic properties of complex numbers are illustrated in Fig. 4.8. For a complex number $z = a + ib$, the real number

$$|z| := \sqrt{a^2 + b^2} = \sqrt{(a + ib)(a - ib)} \tag{4.15}$$

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 Eq. 4.15 as $|z|^2 = zz^*$.

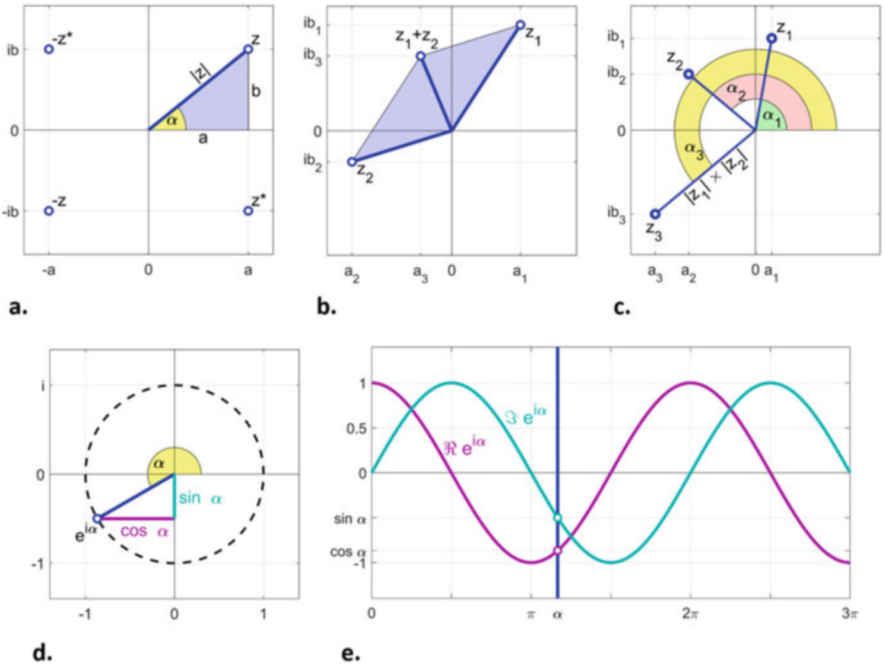


Fig. 4.8 Complex numbers. (**a**) The complex plane is spanned by the real and imaginary axes with units 1 and $i = \sqrt{-1}$. A complex number can be expressed through its real and imaginary parts ($z = a + ib$), or by its "modulus" $|z|$ and "argument," or phase, $\alpha = \arg z$. (**b**) Two complex numbers z_1, z_2 can be summed by adding their respective real and complex parts. (**c**) Multiplication can be thought of as a rotation and scaling, i.e., it adds up the arguments and multiplies the moduli. (**d**) The complex exponential with the purely imaginary argument $i\alpha$ describes a unit circle in the complex plane. (**e**) The real and imaginary parts of the complex exponential are sinusoidal functions of α (Euler's formula)

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

$$\arg z := \begin{cases} \tan^{-1} b/a & \text{for} \quad a > 0 \\ \pi + \tan^{-1} b/a & \text{for} \quad a < 0,\ b \geq 0 \\ -\pi + \tan^{-1} b/a & \text{for} \quad a < 0,\ b < 0 \end{cases} . \tag{4.16}$$

For $a = b = 0$, the argument function is undefined. Strictly speaking, Eq. 4.16 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 arg function as defined in Eq. 4.16 is also known as the two-argument inverse tangent or arctangent, atan2, see Fig. 4.8.

The complex numbers form a plane that is spanned by the real axis with unit 1 and the imaginary axis with unit i (Fig. 4.8a). Calculations on the set of complex numbers are defined in a straightforward way, simply by treating i as an ordinary

number and observing $i^2 = -1$. For example, the sum of two complex numbers is calculated by adding the real and imaginary parts separately, as if complex numbers were two-dimensional vectors. The vector analogy fails, however, in multiplication, for which we obtain

$$\begin{aligned} z_1 z_2 &= (a_1 + ib_1)(a_2 + ib_2) \\ &= a_1 a_2 + i(a_2 b_1) + i(a_1 b_2) + i^2(b_1 b_2) \\ &= \underbrace{a_1 a_2 - b_1 b_2}_{\text{real part}} + i \underbrace{a_2 b_1 + a_1 b_2}_{\text{imaginary part}} . \end{aligned} \tag{4.17}$$

With some computations, it can be shown that $\arg(z_1 z_2) = \arg z_1 + \arg z_2$ and $|z_1 z_2| = |z_1||z_2|$ (cf. Fig. 4.8b, c). This is to say, multiplication of two complex numbers, z_1 and z_2, can be visualized by thinking of z_1 as a "vector" in the complex plane that is rotated by the angle $\arg z_2$ and stretched by the amount $|z_2|$. Of course, due to the commutativity of multiplication, the roles of z_1 and z_2 in this rotational stretching can be exchanged.

In Fourier analysis, complex numbers are used because they simplify the calculations with trigonometric functions, especially with respect to the addition theorems. In fact, sine and cosine functions can be replaced by the complex exponential defined by Euler's[8] formula

$$e^{i\alpha} = \cos\alpha + i \sin\alpha. \tag{4.18}$$

While this relation between the exponential and the sinusoidals may look surprising at first glance, it conforms with—and indeed arises from—the standard rules of calculation. For example, we have $(e^{i\alpha})' = ie^{i\alpha}$ in agreement with $(\cos\alpha + i\sin\alpha)' = -\sin\alpha + i\cos\alpha = i(\cos\alpha + i\sin\alpha)$. Also, it is easy to show that Taylor expansions of the left and right sides of Eq. 4.18 yield equal results.

In the standard version of Euler's formula, Eq. 4.18, the exponent is purely imaginary, which entails $|z| = |e^{i\alpha}| = 1$ for all α. For a general complex number $z = a + ib$, Euler's formula becomes

$$e^z = e^{a+ib} = e^a(\cos b + i \sin b). \tag{4.19}$$

Reversely, we may express a complex number z by its modulus and phase:

$$z = |z| e^{i \arg z}. \tag{4.20}$$

This latter version of Euler's formula nicely illustrates the stretching and rotation property of multiplication in the complex plane: $z_1 z_2 = |z_1||z_2| \exp\{i(\arg z_1 +$

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

$\arg z_2)\}$, see Fig. 4.8c. It is also called the polar or "phasor" representation of a complex number. Figure 4.8d,e 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 α increases), the real and imaginary parts of the complex number describe a cosine and a sine wave, respectively.

We conclude this very brief introduction of complex numbers with the remark that imaginary numbers got their name for a reason. They are indeed unreal in the sense that they cannot be used for measuring physical quantities by combining them with units such as meters or seconds. Complex numbers are used to simplify computations, especially computations involving sinusoidals. When it comes to the interpretation of the obtained results, we have to go back to real numbers, for example, by using Euler's formula "backwards." This reads

$$\cos\alpha = \frac{1}{2}(e^{i\alpha} + e^{-i\alpha}) \tag{4.21}$$

$$\sin\alpha = \frac{1}{2i}(e^{i\alpha} - e^{-i\alpha}). \tag{4.22}$$

4.2.4 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 with a complex eigenvalue. For the calculation, we insert the complex exponential function $\exp\{i\omega x\}$ into the convolution equation:

$$\begin{aligned}(g * \exp_\omega)(x) &= \int g(x')\exp\{i\omega(x-x')\}dx' \\ &= \exp\{i\omega x\}\underbrace{\int g(x')\exp\{-i\omega x'\}dx'}_{\tilde{g}(\omega)}.\end{aligned} \tag{4.23}$$

The eigenvalue associated with the eigenfunction $\exp\{i\omega x\}$ is therefore $\tilde{g}(\omega)$. The values $\tilde{g}_c$ and $\tilde{g}_s$ from Sect. 4.2.2 are the real and imaginary parts of $\tilde{g}$,

$$\tilde{g} = \tilde{g}_c + i\tilde{g}_s = Ae^{i\phi}, \tag{4.24}$$

where $A = |\tilde{g}|$ and $\phi = \arg\tilde{g}$.

For fixed ω, $\tilde{g}$ is a constant; it is, however, defined for all values of ω. We may therefore consider a function $\tilde{g}(\omega)$ that 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 is multiplied by $|\tilde{g}(\omega)|$ and its phase is shifted by $\arg \tilde{g}(\omega)$. The function $\tilde{g}(\omega)$ is therefore called the *modulation transfer function* (MTF) of the convolution system. As we shall see later, its relation to the point-spread function or convolution kernel g in Eq. 4.23 is already the definition of the Fourier transform.

▶ **Key Point: The Eigenfunctions of Convolution** LSI systems can change the amplitude and phase of a sinusoidal input, but not its frequency or shape. This is expressed by saying that the complex exponentials are the eigenfunctions of convolution. The eigenvalue as a function of frequency is called the modulation transfer function (MTF).

Box 4.1 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, and a number of alternative functions have been discussed in the literature. 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, a property that also underlies the central limit theorem in statistics. If the scales of the original Gaussians are σ_1 and σ_2, the scale of the resulting Gaussian is $\sqrt{\sigma_1^2 + \sigma_2^2}$. Also, the Fourier transform of a Gaussian with scale σ is again a Gaussian, however with scale $1/\sigma$.

The Gabor function is closely related to the Gaussian since its Fourier transform is a displaced Gaussian, i.e., a Gaussian shifted away from the origin (see Fig. 4.13 and the accompanying text). The Gabor function is therefore 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: It is simultaneously localized in both domains, albeit to various degrees. A wide Gabor function allows good frequency tuning but is poorly localized, while a narrow Gabor function is nicely localized, but frequency tuning will be poor. This is as good as it can get; tuning in the two dimensions is yoked by the uncertainty relation discussed in connection with Eq. 4.30. Among all possible functions, the Gabor function minimizes the combined uncertainty of space and frequency localization.

Approximations of Gaussian and Gabor-shaped kernels are also found in the early layers of artificial neural networks trained for image processing (see Chaps. 5 and 6). They therefore seem to reflect fundamental properties of natural image statistics.

4.2.5 Example: 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 defocused slide projector. The point-spread function for this system is the image of a small spot of light, i.e., a blurring "disk." It is mathematically described as a Gaussian where the width σ describes the amount of blur,

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{x^2}{2\sigma^2}\right\}. \tag{4.25}$$

The kernel is normalized to a total mass of one; the total light intensity in the blurring disk therefore equals the intensity passing through that particular pixel of the slide.

We know already from the previous sections that the eigenfunctions of the convolution with this (or any other) kernel are sinusoidals. That is to say, if we put a slide showing a sinusoidal grating into a slide projector, the resulting image on the screen will again be a sinusoidal grating with the same spatial frequency (relative to the image diameter) but with reduced amplitude. Since we are talking about blur, it is not surprising that attenuation will be stronger for gratings with higher spatial frequency than for those with lower spatial frequency. If the blurring disk is circular symmetric, no phase shift will occur. From Eq. 4.23, we obtain the modulation transfer function (MTF) $\tilde{g}$ for this system:

$$\tilde{g}(\omega) = \frac{1}{\sqrt{2\pi}\sigma} \int \exp\left\{-\frac{x^2}{2\sigma^2}\right\} \exp\{-i\omega x\}\, dx. \tag{4.26}$$

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:

$$\tilde{g}(\omega) = \frac{1}{\sqrt{2\pi}\sigma} \int \exp\left\{-\frac{1}{2\sigma^2}(x^2 - 2i\sigma^2\omega x - \sigma^4\omega^2 + \sigma^4\omega^2)\right\} dx \tag{4.27}$$

$$= \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\sigma^2\omega^2\right\} \int \exp\left\{-\frac{1}{2\sigma^2}(x - i\sigma^2\omega)^2\right\} dx. \tag{4.28}$$

Next, we substitute $y = x - i\sigma^2\omega$ in the integral and note that $\int \exp\{-y^2/2\sigma^2)\}\, dy = \sqrt{2\pi}\sigma$ (see Box 3.1), even if the integral is taken along a line in the complex plane.[9] We thus obtain the MTF for a Gaussian blurring disk as

$$\tilde{g}(\omega) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\sigma^2\omega^2\right\} \int \exp\left\{-\frac{y^2}{2\sigma^2}\right\}\, dy \tag{4.29}$$

$$= \exp\left\{-\frac{1}{2}\sigma^2\omega^2\right\}. \tag{4.30}$$

This is a Gaussian with width $1/\sigma$. In the example, it means that for large blurring disks (σ large) the frequency range of weakly attenuated gratings is narrow (namely $1/\sigma$), whereas sharp systems with small blurring disks pass also finer gratings. The formal 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 Eq. 4.21 to write the result in real notation. As noted above, this is the standard way to interpret complex number results:

$$(g * \cos_\omega)(x) = \frac{1}{2}\left((g * \exp_\omega)(x) + (g * \exp_{-\omega})(x)\right) \tag{4.31}$$

$$= \frac{1}{2}(\tilde{g}(\omega)e^{i\omega x} + \tilde{g}(-\omega)e^{-i\omega x}) \tag{4.32}$$

$$= \exp\left\{-\frac{1}{2}\sigma^2\omega^2\right\} \cos\omega x. \tag{4.33}$$

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 (see also Box 4.3).

For the sine function, we obtain analogously

$$(g * \sin_\omega)(x) = \exp\left\{-\frac{1}{2}\sigma^2\omega^2\right\} \sin\omega x. \tag{4.34}$$

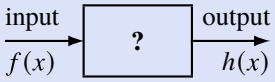
Box 4.2 System Identification

In the "black box" approach, systems are described by the relations between their input and output only, without caring about the internal mechanism. The input–output relations of LSI systems can be measured and described in various ways:

(continued)

[9] This result rests on the path independence of complex line integrals (Cauchy integral theorem).

Box 4.2 (continued)



- The *impulse response* or point-spread function $g(x)$ is the output elicited by a δ pulse. From g, the output to an arbitrary input $f(x)$ can be computed by convolution,

$$h(x) = \int_{\mathbb{R}} g(x - x')f(x')dx'.$$

  In physical systems, very narrow impulses with very large amplitudes can only be realized approximately and will often damage the system in question.
- The *modulation transfer function* $\tilde{g}(\omega)$ describes the relation between sinusoidal input and output. If this is obtained for all frequencies, the system is also completely identified. If we denote the Fourier transforms of input and output as $\tilde{f}$ and $\tilde{h}$, the mapping is given by

$$\tilde{h}(\omega) = \tilde{g}(\omega)\tilde{f}(\omega),$$

  i.e., by simple pointwise multiplication. The two approaches are related by the fact that the MTF, $\tilde{g}(w)$, is the Fourier transform of the impulse response $g(x)$. This relation is also known as the convolution theorem.
- LSI systems can also be identified with *white noise* inputs. This is based on the fact that the cross-correlation function (see Sect. 3.4.3) of the input and output of an LSI system equals the convolution of the impulse response with the autocorrelation function of the input:

$$\Phi_{fh} = \Phi_{f,g*f} = \Phi_{ff} * g.$$

  Marmarelis and Marmarelis (1978). If the input f is a white noise process, its autocorrelation function is $\delta(x)$, see Box 3.4. Its convolution with $g(x)$ therefore reproduces $g(x)$ itself. This method may be the least harmful but requires long measurement times to obtain good estimates of the correlation function.
- A general way to think about system identification is *deconvolution*. From the above equation $\tilde{h}(\omega) = \tilde{g}(\omega)\tilde{f}(\omega)$ we immediately obtain

$$\tilde{g}(\omega) = \frac{\tilde{h}(\omega)}{\tilde{f}(\omega)} \quad \text{for all } \omega \text{ with } \tilde{f}(\omega) \neq 0.$$

(continued)

Box 4.2 (continued)
With deconvolution, some values of $\tilde{g}(\omega)$ can be determined from any known input–output relation. If g is known, as in the blurring disk example, deconvolution can also be used to recover the unblurred input f from the output h.

Equations 4.30 to 4.34 show three things. First, the MTF of 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 attenuate or stop high spatial frequencies (fine detail). This behavior is called *low-pass*; it is very intuitive for the blurred projection example. Third, the width of the MTF, i.e., the "speed" with which $\tilde{g}(\omega)$ decreases for higher frequencies, depends on the width of the filter, σ. The wider the filter mask, the faster does the MTF decrease, i.e., the stronger or more pronounced is the low-pass property.

▶ **Key Point: Gaussian Filters** The modulation transfer function of an LSI system with a Gaussian kernel with scale factor σ is a Gaussian in the frequency domain with scale factor $1/\sigma$. That is to say, filters with narrow Gaussian pass a wide range of frequencies and vice versa.

4.3 Fourier Decomposition

4.3.1 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 linear superpositions 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.

Periodic Functions: Fourier Series

We start by considering 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. Note that $f(x) = f(x + T)$ implies $f(x) = f(x + nT)$ for $n \in \mathbb{Z}$. We therefore assume throughout that T is the smallest positive number satisfying $f(T + x) = f(x)$; it is then also called the fundamental wavelength. In time-dependent signals, T is measured in seconds, while in images and other space-

Table 4.1 Wavelength, frequency, and angular frequency of typical functions

Name	Symbol	$\sin x$	$\sin \omega x$	$\sin 2\pi x$	$\sin 2\pi \omega x$
Wavelength	T	2π	$2\pi/\omega$	1	$1/\omega$
Frequency	$\omega = 1/T$	$1/(2\pi)$	$\omega/(2\pi)$	1	ω
Angular frequency	$\nu = 2\pi/T$	1	ω	2π	$2\pi\omega$

dependent signals, the unit of T may be millimeter, pixels, or degrees of visual angle. The inverse of the wavelength T is called the (linear) frequency, $\omega = 1/T$. Sometimes, we will also use the "angular frequency" $\nu = 2\pi/T$ to characterize a periodic function. It can be thought of as the speed of phase progression with the variable x. The relations are summarized in Table 4.1.

Fourier decomposition is best explained with periodic functions. As an example, consider the "square wave,"

$$s(x) := \begin{cases} 1 \text{ if } \mod(x, T) < T/2 \\ 0 \text{ else} \end{cases} . \tag{4.35}$$

Figure 4.9a shows this square wave function (heavy gray line) together with two sinusoidals. The first one has the same frequency as the square wave itself; we call this frequency the fundamental frequency $\omega_f = 1/T$. In the sequel, we will use the

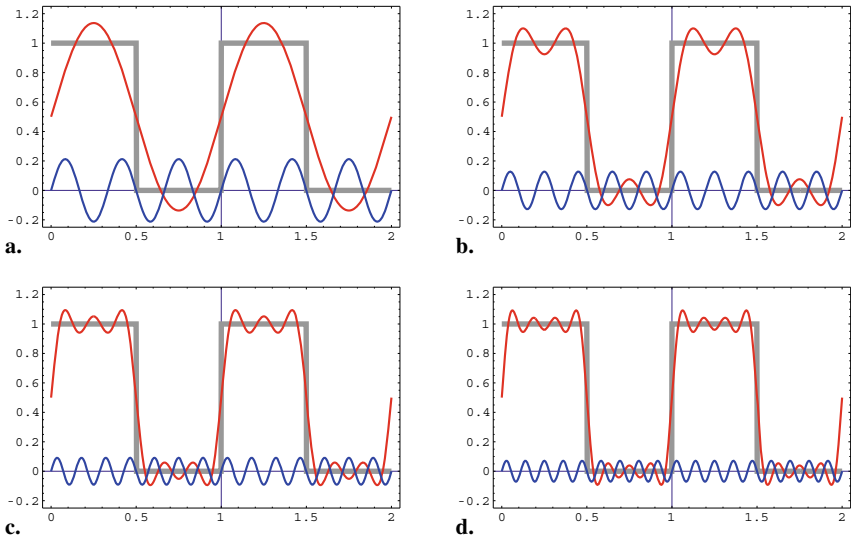


Fig. 4.9 Approximation of the square wave (Eq. 4.35 with $T = 1$) by a Fourier sine series (Eq. 4.38). The square wave is shown as a heavy gray line. In each panel, the two thin lines show the current approximation $p_k(x)$ and the next correction term $b_{k+2} \sin(k+2)\nu x$ for $k = 1, 3, 5, 7$. For further explanations, see text

angular frequency $\nu = 2\pi/T$ for the sake of simplicity. The sinusoidal thus has the form

$$p_1(x) = a_o + b_1 \sin(\nu x), \tag{4.36}$$

where the amplitude b_1 is chosen such that the squared difference between the square wave and the approximation is minimized. The factor a_o is called the DC, or direct current part. In the example, it takes the value 0.5; in general, it is the average of the square wave taken over one complete period. If we consider the difference between $s(x)$ and the first order approximation $p_1(x)$, 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 the Fig. 4.9a. (The fact that the frequency 2ν does not contribute to a better approximation is an accidental property of the square wave.) Figure 4.9b shows the least square approximation of the square wave by the function $p_3(x) = p_1(x) + b_3 \sin(3\nu x)$. 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

$$p_n(x) = a_o + b_1 \sin \nu x + b_3 \sin 3\nu x + \cdots + b_n \sin n\nu x \tag{4.37}$$

$$= a_o + \sum_{k=1}^{n} b_k \sin k\nu x. \tag{4.38}$$

The coefficients b_k can be shown to take the values

$$b_k = \begin{cases} 2/(k\pi) & \text{for} \quad i \in \{1, 3, 5, \ldots\} \\ 0 & \text{for} \quad i \in \{2, 4, 6, \ldots\} \end{cases}; \tag{4.39}$$

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

Series of the type shown in Eq. 4.38 are called "trigonometric polynomials" or *Fourier series*; a more complete definition including also cosine terms appears in Eq. 4.41. Figure 4.9 shows the first steps of that series, i.e., the functions $p_1(x)$ through $p_7(x)$. Except for the discontinuities of the square wave s, which occur at all integer multiples of $T/2$, the series converges[10] toward the true functional value:

$$\lim_{n\to\infty} p_n(x) = s(x) \quad \text{for all } x \text{ with } 2x \bmod T \neq 0; \tag{4.40}$$

[10] For the square wave and other functions with step discontinuities, convergence is pointwise, but not uniform. In the vicinity of the discontinuities of the square wave, 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.

at the discontinuities, $p_n(x)$ takes the value $1/2$, i.e., the mean of the one-sided limits.

Figure 4.10 shows a more general case where the function is irregular within the interval $[0, T]$ but repeats this irregular course to infinity. In this case, 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 Fig. 4.9, 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 coefficients a_k and b_k:

$$p_n(x) = a_o + \sum_{k=1}^{n} a_k \cos k\nu x + \sum_{k=1}^{n} b_k \sin k\nu x. \tag{4.41}$$

The DC component a_o can be included into the first sum with $k = 0$ since $\cos(0x) = 1$ for all x.

▶ **Key Point: Fourier Series** Each piecewise continuous, periodic function with wavelength T can be expressed as a sum of cosine and sine components with angular frequencies $2k\pi/T$ and the coefficients a_k, b_k. The components with $k > 1$ (i.e., whose frequencies are integer multiples of the fundamental frequency) are called harmonics.

Gaussian High- and Low-Pass: A Preview of the Convolution Theorem

In Sect. 4.2.4, we have shown that the convolution of a sine or cosine function with some kernel amounts to multiplying the sine or cosine with a gain 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 and adding up the results afterward:

$$(\textstyle\sum_k f_k) * g = \sum_k (f_k * g). \tag{4.42}$$

The frequencies of the sines and cosines do not change; the convolution therefore amounts to a simple multiplication of the according coefficients with some frequency-dependent factor.

As an example, we consider the convolution of the square wave (Eq. 4.35) with a Gaussian (Eq. 4.25), see Fig. 4.11. Recall that we already discussed this convolution as a model of a defocused slide projector taking a stripe pattern as its input and producing a blurred image of the stripes (Sect. 4.2.5). We express the square wave as its Fourier series

$$s(x) = \frac{1}{2} + \sum_{k=1}^{\infty} b_k \sin k\nu x. \tag{4.43}$$

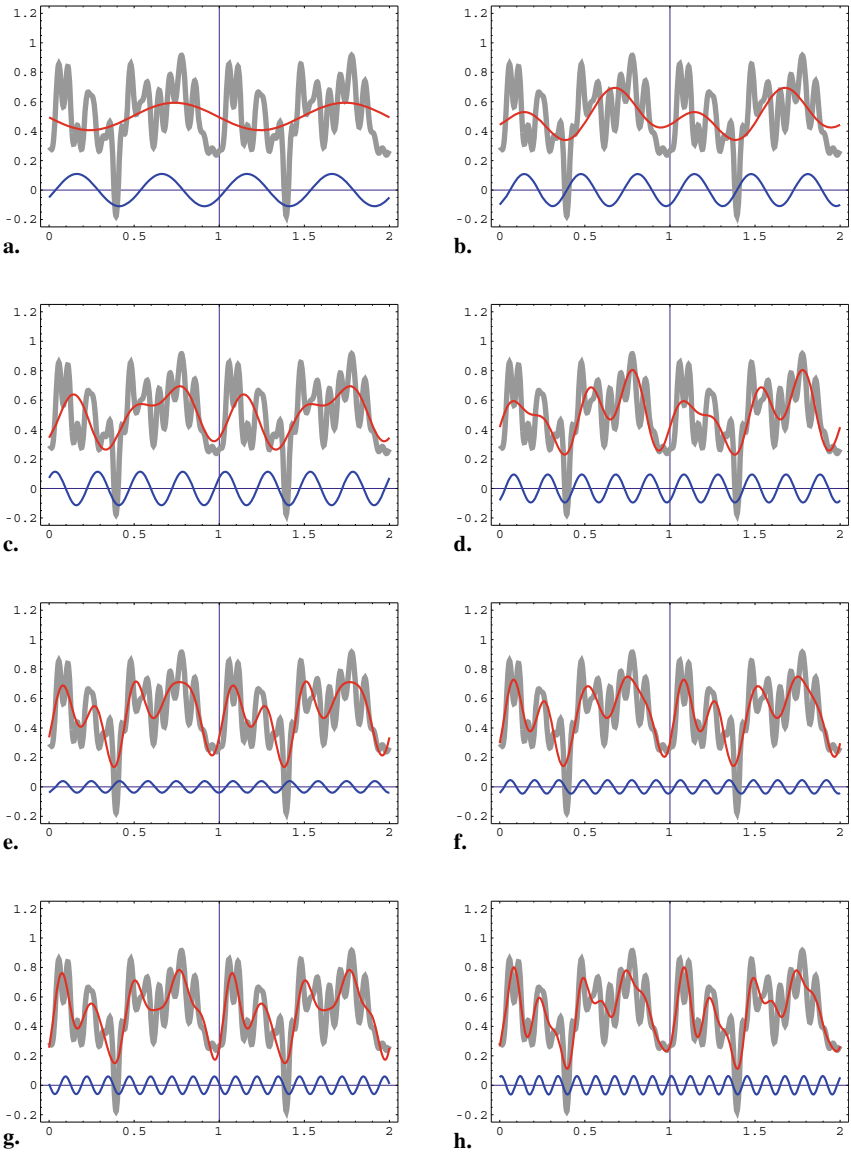


Fig. 4.10 Approximation of an arbitrary periodic function by a Fourier series. In each panel, the function is shown as heavy gray line. The thin lines show the current approximation $p_k(x)$ (red) and the next correction term $a_{k+1}\cos(k+1)\nu x + b_{k+1}\sin(k+1)\nu x$ (blue) for $k = 1\ldots 8$. For further explanations see text

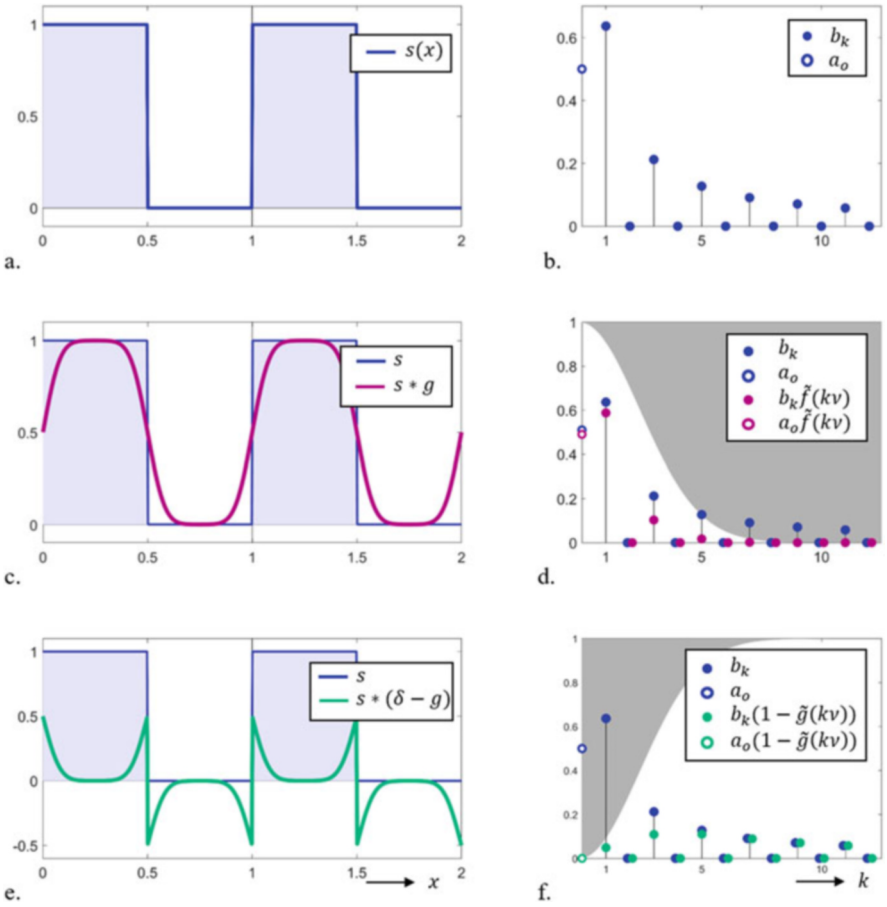


Fig. 4.11 Gaussian high- and low-passes. Blue: original function and its Fourier coefficients; purple and cyan: filtered functions and their Fourier coefficients. (**a**) Square wave (Eq. 4.35 with $T = 1$). (**b**) Coefficients of the according Fourier series (Eq. 4.38). (**c**) Low-pass filtered version $(s * g)$ of the square wave, using a Gaussian filter kernel with $\sigma = 0.4$. (**d**) Fourier coefficients for the filtered signal. The modulation transfer function $\tilde{g}(k\nu)$ is shown in the background. The coefficients are obtained by multiplying the original coefficients (blue dots from part **b**) with the MTF. (**e**) High-pass filtered version of the square wave obtained by convolution with $\delta - g$. (**f**) Fourier coefficients of the high-pass filtered signal. In this case, the MTF is $\mathbf{1} - \tilde{g}$; it appears in the background

As a consequence of the linearity of convolution, we obtain

$$(s * g)(x) = \frac{1}{2}\int g(x)dx + \sum b_k(\sin_{k\nu} * g)(x). \tag{4.44}$$

The integral results from the "convolution" of g with the DC component; in the example, we have chosen g with total mass 1.

Using the results from Eqs. 4.33 and 4.34, we may now replace the convolution of a sine wave with angular frequency $k\nu$ by a multiplication with the according value of the modulation transfer function of the Gaussian, $\tilde{g}$:

$$(s * g)(x) = \frac{1}{2} + \sum_k b_k \tilde{g}(k\nu) \sin k\nu x \tag{4.45}$$

$$= \frac{1}{2} + \sum_k b_k \exp\left\{-\frac{1}{2}\sigma^2(k\nu)^2\right\} \sin k\nu x. \tag{4.46}$$

This is a new Fourier series with the unchanged DC component 1/2 and the new coefficients $b'_k = b_k \tilde{g}(k\nu)$; they are shown as pink dots in Fig. 4.11d. By comparing the blue and pink dots, it can be seen that attenuation is absent or weak for the low frequencies ($k = 0, 1$) but gets increasingly more pronounced for higher values of k. The ratio between the amplitudes marked by the pink and blue dots equals the value of the MTF for each particular frequency; it appears in the background of the figure. The overall behavior is described as low-pass, because low frequencies are passed, while higher frequencies are stopped.

If the smoothed square wave function is subtracted from the original one, the high-spatial-frequency components that were removed in smoothing now stand out in the difference (Fig. 4.11e). Since the square wave—like any function—can be thought of as the convolution of itself with the Dirac δ impulse (Eq. 2.13), we can formally write this difference as a convolution with a new kernel g' defined as $g' := \delta - g$. It will pass the high frequencies and stop the low ones and is therefore called a high-pass. In the frequency domain, this particular high-pass is characterized by the MTF $1 - \tilde{g}(k\nu)$.

This example is an instance of the convolution theorem, see Sect. 4.4. 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.

▶ **Key Point: The Modulation Transfer Function (MFT)** The modulation transfer function of an LSI system specifies the input–output relation for sinusoidal inputs of each frequency. It is called low-pass if low frequencies get through, while high frequencies are stopped or attenuated, and high-pass in the reverse case.

Box 4.3 Symmetry Relations: Odd and Even, Real and Imaginary

(a) One important difference between the sine and cosine functions is about their symmetries. The cosine function is *even*, while the sine function is

(continued)

Box 4.3 (continued)
odd: that is, they satisfy the relations

$$\cos(-x) = \cos(x) \quad \text{and} \quad \sin(-x) = -\sin(x).$$

As we have seen in Chap. 3, these symmetry relations are shared by the sine and cosine Gabor functions, while the Gaussian envelope itself is even. Note that continuous odd functions need to satisfy $f(0) = 0$.

Arbitrary functions can be split up into an even and an odd part by setting

$$f_e(x) = \frac{1}{2}(f(x) + f(-x))$$
$$f_o(x) = \frac{1}{2}(f(x) - f(-x)).$$

Clearly, we have $f_e(x) = f_e(-x)$, $f_o(-x) = -f_o(x)$, and $f(x) = f_e(x) + f_o(x)$.

(b) In Fourier series expansions, the sine coefficients of even functions vanish ($b_k = 0$ for all k), as do the cosine coefficients (a_k) of odd functions. In the complex notation, Fourier transforms of even functions are real, whereas Fourier transforms of odd functions are imaginary. Since the Fourier transform is itself a linear operation, we may write $\tilde{f} = (f_e)^\sim + (f_o)^\sim$. Then, the following relations hold:

$$(f_e)^\sim(\omega) = \Re\tilde{f}(\omega) \quad \text{and} \quad (f_o)^\sim(\omega) = i\Im\tilde{f}(\omega),$$

where $\Re$ and $\Im$ denote the real and imaginary parts of a complex number.

(c) The functions considered here in the space and time domains are all real-valued. In this case, the real part of the Fourier transform is even, while the imaginary part is odd. Using the notion of the complex conjugate, $z^* = \Re z - i\Im z$, we may write

$$\tilde{f}(-\omega) = \tilde{f}^*(\omega),$$

in agreement with Eq. 4.57. When plotting Fourier transforms, it therefore suffices to use the positive half of the frequency axis. The phase will be an odd function of frequency: $\arg(\tilde{f}(-\omega)) = -\arg(\tilde{f}(\omega))$. Two-dimensional Fourier transforms of real functions are completely determined by their values on a half plane.

(continued)

Box 4.3 (continued)
(d) If a causal function, i.e., a function satisfying $g(t) = 0$ for $t < 0$, is split up into an even and an odd part, we obtain $g_o(t) = \mathrm{sgn}(t) g_e(t)$ where sgn is the sign function. In this case, $\Re\tilde{g}$ and $\Im\tilde{g}$ are a transform pair of the Hilbert transform.

Finding the Coefficients

So far, we have seen that trigonometric polynomials can approximate continuous functions; they are then called Fourier series. We write the general form as

$$p_n(x) := \sum_{k=o}^{n} a_k \cos k\nu x + \sum_{k=1}^{n} b_k \sin k\nu x, \tag{4.47}$$

where the DC component has been incorporated into the first sum. In Eq. 4.47, $\nu/2\pi$ is the fundamental frequency of the signal, i.e., p_n repeats itself with a wavelength 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. It looks much like the cyan curve in Fig. 4.11e.

How can we find the coefficients a_k, b_k such that p_n will approximate a given function f? If we assume that every continuous periodic function can in fact be written as a trigonometric polynomial or Fourier series, $f = p_\infty$,[11] we can find the coefficients by exploiting the so-called orthogonality relations of sinusoidals that hold for all $k, l > 0$:

$$\frac{1}{\pi}\int_0^{2\pi} \sin kx \ \sin lx \ dx = \begin{cases} 1 & \text{if} \quad k = l \\ 0 & \text{if} \quad k \neq l \end{cases} \tag{4.48}$$

$$\frac{1}{\pi}\int_0^{2\pi} \cos kx \ \cos lx \ dx = \begin{cases} 1 & \text{if} \quad k = l \\ 0 & \text{if} \quad k \neq l \end{cases} \tag{4.49}$$

$$\int_0^{2\pi} \sin kx \ \cos lx \ dx = 0 \ \text{ for all } \ k, l \in \mathbb{Z}. \tag{4.50}$$

Geometrical motivations for these relations can be found by plotting the involved functions for selected values of k and l. Mathematically, it can be shown that

[11] The proof of this assumption is the mathematically hard part of Fourier theory. It includes the problem of the completeness of the sinusoidals as a basis in function space and the convergence of the series. For details, see textbooks of functional analysis.

eigenfunctions of a linear operator must be orthogonal as long as the eigenvalues differ. With the orthogonality relations, we obtain:

$$a_o = \frac{1}{T}\int_0^T f(x)dx \tag{4.51}$$

$$a_k = \frac{2}{T}\int_0^T f(x)\cos k\nu x\, dx; \quad k \in \{1, 2, 3, \ldots\} \tag{4.52}$$

$$b_k = \frac{2}{T}\int_0^T f(x)\sin k\nu x\, dx; \quad k \in \{1, 2, 3, \ldots\}. \tag{4.53}$$

These latter results are proven by substituting for $f(x) = p_\infty(x)$ from Eq. 4.47 and applying the distributive law until the orthogonality terms appear separately for each element of the sum. All of these vanish except for the one where $k = l$, which evaluates to $T/2$. We call a_k and b_k the Fourier cosine and sine coefficients for the k-th harmonic.

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

$$\frac{1}{2\pi}\int_0^{2\pi} \exp\{ikx\}\exp\{-ilx\}\, dx = \begin{cases} 1 & \text{if} \quad k = l \\ 0 & \text{if} \quad k \neq l \end{cases}. \tag{4.54}$$

The coefficients are obtained from

$$c_k = \frac{1}{T}\int_0^T g(x)\exp\{-ik\nu x\}dx; \quad k \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}. \tag{4.55}$$

If we allow for coefficients with negative indices, we may write the Fourier series as

$$p_n(x) := \sum_{k=-n}^{n} c_k \exp\{ik\nu x\}. \tag{4.56}$$

Instead of having two coefficients for each frequency $k\nu$ (a_k, b_k for sine and cosine), we now have coefficients for positive and negative frequencies (c_k and c_{-k}) that are complex conjugates of each other: $c_{-k} = c_k^*$ (see also Box 4.3). The relations between the real and complex coefficients follow from Euler's formula (Eqs. 4.21, 4.22). They read

$$c_k = \begin{cases} \frac{1}{2}(a_k - ib_k) & \text{for} \quad k > 0 \\ \frac{1}{2}(a_k + ib_k) & \text{for} \quad k < 0 \\ a_o & \text{for} \quad k = 0 \end{cases}. \tag{4.57}$$

Determining the coefficients from the orthogonality relations is in fact equivalent to the least square fitting approach used in the context of Figs. 4.9 and 4.10. In the space of functions, each finite set of complex exponentials $\exp\{ikx\}$ forms the orthonormal basis of a linear subspace. The Fourier series expansion of the square wave or any other function projects this function on the linear subspace of trigonometric polynomials. The "distance," or residual, between the original function and its trigonometric approximation is minimized, if and only if this projection is orthogonal.

In numerical analysis, functions are sampled and represented as discrete lists or vectors of numbers. In this case, the "discrete Fourier transform" is applied, which transforms a sampled input function into a list of Fourier coefficients. An efficient algorithm for this computation is known as fast Fourier transform, or FFT.

▶ **Key Point: Fourier Coefficients** The complex exponentials provide an orthonormal basis in function space. The Fourier coefficients are the coordinate values of a function with respect to this new basis. A numerical algorithm for calculating the transform is called FFT.

4.3.2 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 that can be made periodic by simply repeating the finite domain over and over again. An important example for this latter group is given by functions defined on the circle. Their Fourier representation is a discrete set of coefficients associated with the harmonics, i.e., frequencies that are integer multiples of the fundamental frequency. Spectra of such functions, as introduced in Fig. 4.1, are line spectra, with each line corresponding to a discrete frequency component. We now discuss two extensions of the Fourier series concept.

Nonperiodic Functions

The generalization to nonperiodic functions is mathematically difficult, but intuitively quite easy, if we consider functions of increasing period length T (see Fig. 4.12). For a given T, for example $T = 1$, we have coefficients at the multiples of the fundamental frequency $\omega_o = 1/T = 1$,

$$\omega = k\omega_o = \frac{k}{T} \in \{1, 2, 3, 4, 5, \ldots\}. \tag{4.58}$$

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

$$\omega = k\omega_o = \frac{k}{T} \in \left\{\frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, \ldots\right\}. \tag{4.59}$$

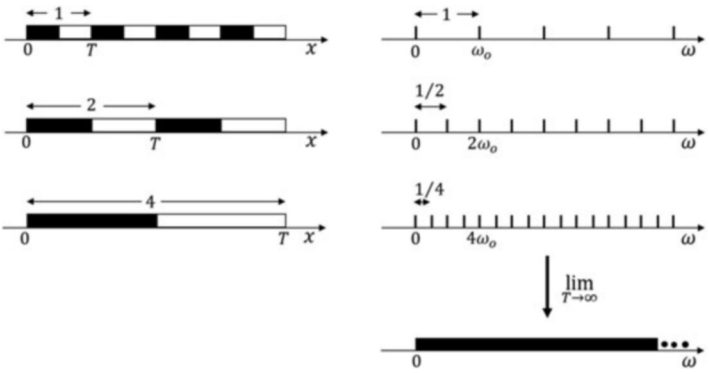


Fig. 4.12 Motivation of continuous Fourier transforms. Periodic functions with wavelength T (left) have Fourier components at the integer multiples of the fundamental frequency $\omega_o = 1/T$ (right). As T increases, the separation of the components decreases. Nonperiodic functions can be considered periodic functions with wavelength infinity. In this case, we get Fourier components for all frequencies in a continuum

In the limit of an infinite period, i.e., if the function is not periodic at all, we get a "coefficient" for every value of ω; that is to say: we get a function of a continuous frequency variable. Switching back to the complex notation, we thus obtain the Fourier transform of a function $g(x)$ as

$$\tilde{g}(\omega) := \int_{-\infty}^{\infty} g(x) \exp\{-i\omega x\} dx. \tag{4.60}$$

Note that its equation is identical to the underbraced part of Eq. 4.23, which implies that the modulation transfer function is the Fourier transform of the convolution kernel.

Equation 4.60 is the analytic side of Fourier transform, and it describes how we can identify the sinusoidals contained in a function defined in the spatial or temporal domain. On the constructive side, where we rebuild arbitrary functions as trigonometric series, the continuous case becomes

$$g(x) := \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{g}(\omega) \exp\{i\omega x\} d\omega. \tag{4.61}$$

Equation 4.60 is called the Fourier forward transform and Eq. 4.61 the Fourier backward transform. Applying both in a sequence reconstructs the original function as long as this was continuous and all integrals exist.

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 may also

be put in front of the forward equation[12] or split symmetrically between the two transform directions. Applying the forward transform twice results in a mirror image of the original function (i.e., $\tilde{\tilde{g}}(x) = g(-x)$), and applying it four times in a row reproduces the original.

The motivation of continuous Fourier transforms given here and in Fig. 4.12 should not mislead us to think that the Fourier transform of a periodic function and the discrete coefficients of the Fourier series are simply the same. Rather, the Fourier transform of a periodic function is a "comb" or "train" of equidistant Dirac pulses, $\delta(\omega - k\omega_o)$, multiplied with the corresponding Fourier coefficient. In particular, the Fourier transform of a sine of frequency ω_o is $(\delta(\omega - \omega_o) - \delta(\omega + \omega_o))/2i$. Strictly speaking, Fourier series and Fourier transform are therefore two different operations. However, in practical applications, where numerical versions of the Fourier transform are used, this difference is of minor relevance.

The function $\tilde{g}(\omega)$ in Eq. 4.60 is a complex-valued function of the real variable ω, for an example see Fig. 4.13. In the space spanned by the frequency axis and the complex plane, the graph of the function spirals around the ω-axis. It can be expressed by its real and imaginary parts (red and green projections in Fig. 4.13) or, alternatively, by its power and phase. The power of a particular frequency is the squared absolute value of the complex number $\tilde{g}(\omega)$; in Fig. 4.13, it corresponds to the squared distance of the curve from the frequency axis, see also Sect. 4.2.3. In the figure, the phase angle is best seen as the angle between the real axis and the brownish rays on the left.

The example chosen in Fig. 4.13 is the Fourier transform of the displaced Gaussian, $\exp\{-\frac{1}{2}(x - x_o)^2\}$, where x_o is the displacement. For the Gaussian itself, we have seen in Sect. 4.2.5 that its Fourier transform is again a Gaussian, however with inverted scale ($1/\sigma$ instead of σ); the displacement results in an additional factor $\exp\{-i\omega x_o\}$ that is 1 if $x_o = 0$ (Fourier shift theorem). The Fourier transform of a displaced Gaussian is therefore a (complex) Gabor function where the frequency of the sinusoidal component increases with the amount of the displacement. Displacement affects only the phase of the Fourier transform, and its amplitude remains unchanged.

▶ **Key Point: Continuous Fourier Transforms** For nonperiodic functions, the Fourier coefficients are replaced by a complex-valued function depending on a continuous real frequency variable.

Fourier Transforms in Two and More Dimensions

The Fourier transform generalizes to functions of two or more variables, such as images or spatiotemporal intensity distributions. In this case, the sinusoidal basis functions must be replaced by plane waves, e.g.,

$$\sin(\omega_x x + \omega_y y) = \sin(\boldsymbol{\omega} \cdot \mathbf{x}) \tag{4.62}$$

[12] This would have been more intuitive since the forward equation is basically a generalization of Eq. 4.55. It is, however, not used in the standard definitions.

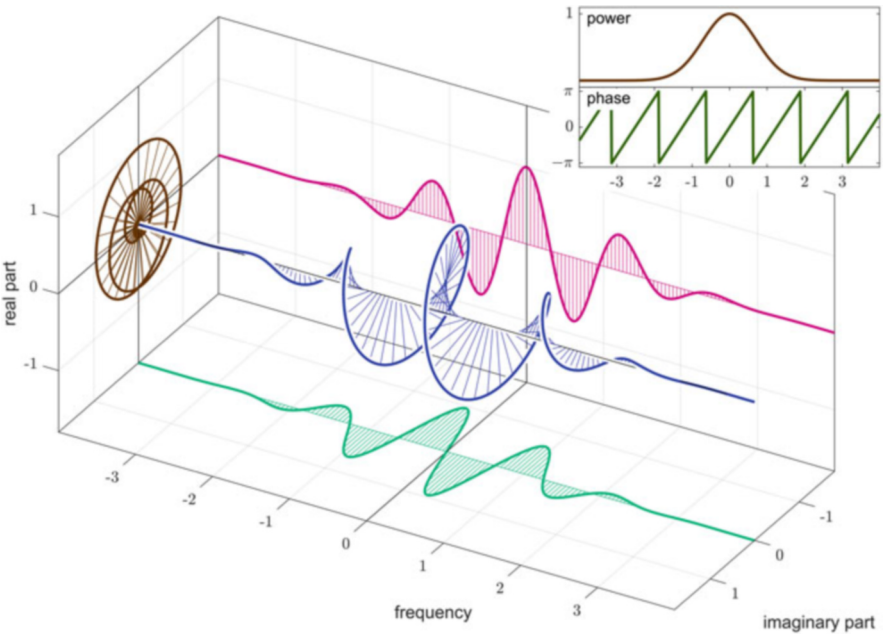


Fig. 4.13 Complex Fourier transform of a displaced Gaussian. The blue curve shows the function $\tilde{f}(\omega) = \exp\{-\omega^2\sigma^2/2\}\exp\{i\omega x_o\}$, i.e., the Fourier transform of $f(x) = \exp\{-(x - x_o)^2/(2\sigma^2)\}$ with $\sigma = 1$ and $x_o = -5$. The complex functional value of each frequency ω is also shown as a "vector," or pointer in the complex plane. The red and green curves are the real and imaginary parts of the same function shown as projections on the ω-real and ω-imaginary planes, respectively. The brown curve in the left is a projection on the complex plane itself. The squared lengths of the blue pointers correspond to the power of the signal. The angle of the pointers in the complex plane is the Fourier phase. The inset shows the power, $|\tilde{f}(\omega)|^2$, and the phase angle, $\arg(\tilde{f}(\omega))$

as are shown in Figs. 3.5a and 4.5. The argument of the sine function is a dot product (see below, Sect. 5.1.3) of a frequency vector $(\omega_x, \omega_y, \ldots)$ and a vector of the original coordinates $(x, y, \ldots)$ defined as $(\boldsymbol{\omega} \cdot \mathbf{x}) = \sum_i \omega_i x_i$. Time and temporal frequency may be treated as just another component of these vectors. The orientation of the wavefronts is orthogonal to the vector (ω_x, ω_y), and the separation of wave peaks (wavelength) is $2\pi(\omega_x^2 + \omega_y^2)^{-1/2}$.

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

$$\tilde{g}(\omega_x, \omega_y) := \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x, y)\exp\{-i(\omega_x x + \omega_y y)\}dxdy. \tag{4.63}$$

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 Fig. 4.5: Each cell in this figure represents a two-dimensional frequency vector for which the associated grating is shown.

Note that if $g(x, y)$ can be written as a product, $g(x, y) = g_1(x)g_2(y)$, then, by Fubini's theorem, the Fourier transform is obtained as the product of the one-dimensional transforms $\tilde{g}_1(\omega_x)$, $\tilde{g}_2(\omega_y)$. As a special case, consider the two-dimensional Gaussian $\exp\{-(x^2 + y^2)/2\} = \exp\{-x^2/2\}\exp\{-y^2/2\}$. Its Fourier transform is the two-dimensional Gaussian $\exp\{-(\omega_x^2 + \omega_y^2)/2\}$.

▶ **Key Point: Two-Dimensional Fourier Transform** In two-dimensional functions with variables (x, y), the complex exponentials used as basis functions generalize to plane waves defined via the dot product of (x, y) and a two-dimensional frequency (ω_x, ω_y). The transform is then a complex-valued function of two real variables.

4.4 The Convolution Theorem

We are now ready to discuss the central result of this chapter: that is, the convolution theorem. Intuitively, this theorem states that when we go from the ordinary domain of spatial and/or temporal coordinates to the frequency domain, convolution is replaced by pointwise multiplication. This is illustrated in Fig. 4.14 for the most important application of the theorem: that is, the analysis of LSI systems. The functions f and g are the input and point-spread function of an LSI system. The Fourier transform of g is the modulation transfer function. The operation of the system, i.e., the convolution of f and g, can then be realized by expressing f by its Fourier components (i.e., as $\tilde{f}$), passing each component separately through the system such that it gets multiplied with the respective value of the modulation transfer function $\tilde{g}$, and reconstructing the output $h(x)$ by summing all frequency components according to their new weights $\tilde{h}(\omega) = \tilde{f}(\omega)\tilde{g}(\omega)$. The theorem can thus be summarized in the phrase: The Fourier transform of the convolution $f * g$ equals the product of the Fourier transform $\tilde{f} \times \tilde{g}$:

$$(f * g)^{\sim} = \tilde{f}\,\tilde{g}. \tag{4.64}$$

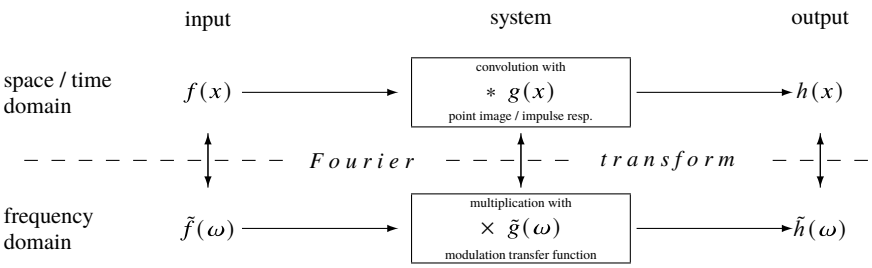


Fig. 4.14 Summary of the relation of Fourier transform and linear systems theory, i.e., the convolution theorem

An example of this relation was already given in Fig. 4.11. Here we present a formal proof for completeness. We start by writing down the definition of convolution

$$h(x) = \int f(x')g(x - x')dx \tag{4.65}$$

and its Fourier transform

$$\tilde{h}(\omega) = \int h(x)e^{-i\omega x}dx = \iint f(x')g(x - x')dx'\, e^{-i\omega x}dx. \tag{4.66}$$

If all occurring integrals exist, we may, by Fubini's theorem, take the exponential into the inner integral and exchange the inner and outer integrals.

$$\tilde{h}(\omega) = \iint f(x')g(x - x')e^{-i\omega x}dxdx'. \tag{4.67}$$

We substitute $(x', x - x') \mapsto (u, v)$ and observe $x = u + v$ and $\exp\{-i\omega x\} = \exp\{-i\omega u\}\exp\{-i\omega v\}$. Then, another application of Fubini's theorem yields

$$\tilde{h}(\omega) = \int f(u)e^{-i\omega u}du \int g(v)e^{-i\omega v}dv = \tilde{f}(\omega)\,\tilde{g}(\omega), \tag{4.68}$$

which completes the proof.

The convolution theorem works also for the inverse Fourier transform: If we multiply two functions in the space-time domain, the Fourier transform of the product equals the frequency-domain convolution of the individual Fourier transforms.

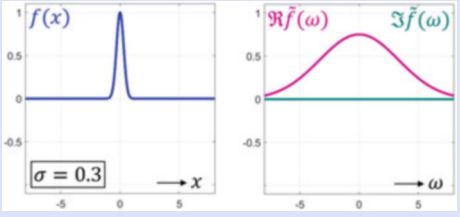
Box 4.4 Fourier Transform Pairs

This box shows a number of important Fourier transform pairs. Note that it can be read in both directions. For example, the Fourier transform of a wide Gaussian is a narrow Gaussian (case a.). For the sake of simplicity, we use the notation $\varphi(x) := \exp\{-x^2/2\}$ throughout:

(a) Narrow Gaussian $\longleftrightarrow$ Wide Gaussian

$$f(x) = \varphi\left(\frac{x}{\sigma}\right)$$

$$\tilde{f}(\omega) = \sqrt{2\pi}\sigma\,\varphi(\sigma\omega)$$



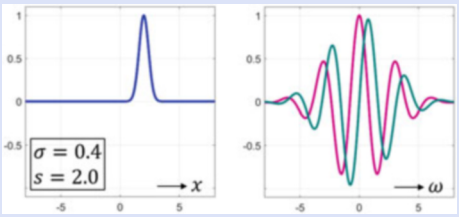
(continued)

Box 4.4 (continued)

(b) Displaced Gaussian $\longleftrightarrow$ Gabor function

$$f(x) = \varphi\left(\frac{x-s}{\sigma}\right)$$

$$\tilde{f}(\omega) = \sqrt{2\pi}\sigma\ \varphi(\sigma^2\omega^2) \times \exp\{-i\omega s\}$$

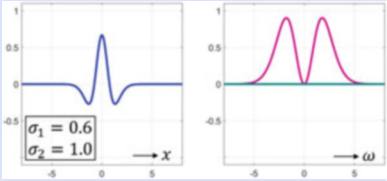


The Fourier transforms of the sine and cosine Gabor functions can be inferred from this. They are pairs of displaced Gaussians at $\pm s$ with width $1/\sigma$. Both Gaussians have real positive amplitudes for the cosine case, and imaginary amplitudes with opposite signs for the sine case.

(c) Difference of Gaussians $\longleftrightarrow$ Band-Pass

$$f(x) = \frac{1}{\sigma_1}\varphi\left(\frac{x}{\sigma_1}\right) - \frac{1}{\sigma_2}\varphi\left(\frac{x}{\sigma_2}\right)$$

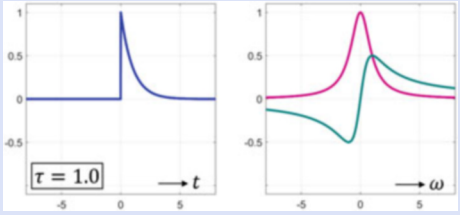
$$\tilde{f}(\omega) = \sqrt{2\pi}\,[\varphi(\sigma_1\omega) - \varphi(\sigma_2\omega)]$$



(d) Exponential decay $\longleftrightarrow$ First order low-pass

$$f(t) = \begin{cases} 0 & \text{for } t \le 0 \\ e^{-t/\tau} & \text{for } t > 0 \end{cases}$$

$$\tilde{f}(\omega) = \tau\frac{1+i\omega}{1+\tau^2\omega^2}$$



4.5 Facts on Fourier Transforms

So far, we have tried to develop the central results of Fourier theory in relation to LSI systems with a minimum of mathematical rigor. In this section, we summarize the most important facts for reference.

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

$$\text{forward:} \qquad \tilde{g}(\omega) := \int g(x)\exp\{-i\omega x\}dx, \tag{4.69}$$

$$\text{backward:} \qquad g(x) := \frac{1}{2\pi}\int \tilde{g}(\omega)\exp\{i\omega x\}d\omega. \tag{4.70}$$

$\tilde{g}(\omega)$ is a complex number that 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, Eq. 4.69 therefore means that every continuous function can be represented as the sum of sine and cosine functions.

Multiple Dimensions Fourier transforms in multiple dimensions are obtained by replacing the ordinary products ωx by the dot product ($\boldsymbol{\omega} \cdot \mathbf{x}$), where $\boldsymbol{\omega}$ and $\mathbf{x}$ are vectorial quantities in the frequency and in the space/time domains, respectively. The complex exponentials then become plane waves, for example sinusoidal gratings in the two-dimensional case. Fourier transforms of images are usually plotted as two two-dimensional functions, one for the real and one for the imaginary part. Alternatively, one may plot the power (squared absolute value) and the phase.

Linearity The Fourier transform is a linear mapping from the space of square integrable functions into itself. Therefore a multiple of a function transforms to the according multiple of the transform. Also, sums of functions transform to the sum of the individual transforms. This property, together with the scale and shift theorems discussed below, can be used to get an intuition of how specific transforms look, even if the proper calculation is often difficult. For examples, see Box 4.4.

Fourier Transform as Change of Coordinates 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 $(0, \ldots, 0, 1, 0 \ldots, 0)^\top$) 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 Fourier basis is complete, the length of a vector does not depend on the coordinate system used. Therefore, the relation $\int f^2(x)dx = 1/(2\pi) \int |\tilde{f}(\omega)|^2 d\omega$ holds (Parseval's identity).

Convolution Theorem If the original functions are replaced by their Fourier transforms, then convolution is replaced by multiplication:

$$(f * g)\tilde{}(\omega) = \tilde{f}(\omega)\, \tilde{g}(\omega) \tag{4.71}$$

(see Fig. 4.14). For an LTI system with impulse response g, this implies that the modulation transfer functions equal the Fourier transform of the impulse response. The commutativity and associativity of convolution follow directly from this theorem.

Filters: High-Pass, Low-Pass, Band-Pass, Notch As a consequence of the convolution theorem, LSI systems can be described by their modulation transfer function that specifies an amplitude factor ($|\tilde{g}(\omega)| > 1$: amplification; $|\tilde{g}(\omega)| < 1$:

attenuation) and a phase shift ($\arg(\tilde{g}(\omega))$) applied to every frequency component in the input. Frequencies with $|\tilde{g}(\omega)| \approx 0$ are removed, or "stopped," while others are "passed." Filters are roughly classified into four groups: low-passes pass low frequencies only (e.g., Gaussian smoothing filter); high-passes pass high frequencies only (e.g., differentiation); band-passes pass frequencies in a middle "pass band," while lower and higher frequencies are stopped (e.g., difference of Gaussian filter); and notch filters stop frequencies in a middle "stop band," while lower and higher frequencies are passed.

Power Spectrum The squared modulus (complex absolute value) of a Fourier transform $\tilde{f}(\omega)$ is known as the power spectrum of f,

$$|\tilde{f}(\omega)|^2 = \tilde{f}(\omega)\tilde{f}^*(\omega), \tag{4.72}$$

where the superscript $*$ marks the complex conjugate, $z^* = \Re z - i\Im z$. The power spectrum is a real-valued function that specifies for each frequency the (squared) amplitude of the sinusoidal contained in the function, irrespective of phase. For two Fourier transforms $\tilde{f}$, $\tilde{g}$, we call $\tilde{f}\tilde{g}^*$ the cross spectrum of f and g. The cross spectrum is not commutative and may take complex values.

Correlation Theorem The correlation (or Wiener[13]–Khinchin[14]) theorem states that the Fourier transform of the cross-correlation function of two functions, f and g,

$$\Phi_{fg}(y) := \int f(x)g(x+y)dx \tag{4.73}$$

is given by the cross spectrum defined above,

$$\widetilde{\Phi_{fg}}(\omega) = \tilde{f}(\omega)\tilde{g}^*(\omega). \tag{4.74}$$

If $f = g$, this theorem implies that the Fourier transform of the autocorrelation function equals the power spectrum. It reduces the correlation operation to a product of two Fourier transforms (one as complex conjugate). Note the analogy to the convolution theorem that reduced the convolution operation to a multiplication of two (standard) Fourier transforms.

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

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

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

$$\tilde{g}_s(\omega) = \exp\{i\omega s\}\tilde{g}(\omega). \tag{4.75}$$

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

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

$$\tilde{g}_a(\omega) = \frac{1}{a}\tilde{g}\left(\frac{\omega}{a}\right). \tag{4.76}$$

The uncertainty relation studied in Eq. 4.30 is a special case of the scale theorem.

4.6 Summary and Further Reading

1. The complex exponentials are the eigenfunctions of convolution, or LSI systems.
2. In the Fourier transform, complex exponentials are used as a basis system to express functions that are originally defined in the space or time domain ("original function") as functions of spatial or temporal frequencies ("transforms").
3. For a given LSI system, the eigenvalues associated with each complex exponential (i.e., with each frequency) form the modulation transfer function characterizing the system. It is the Fourier transform of the impulse response or point-spread function.
4. The convolution theorem states that the Fourier transform of the convolution of two functions equals the pointwise product of the two Fourier transforms.

Texts

Bracewell (1999): *Classical textbook on Fourier transforms focusing on parts of theory relevant for applications without overly indulging in mathematical theory.*

Gonzalez and Wood (2018): *Standard text on image processing including image filtering and two-dimensional Fourier analysis.*

Oppenheim et al. (1997): *Standard text on signal processing with full coverage of Fourier analysis.*

Robinson (2020): *Modern text on functional analysis that gives rigorous proofs of many claims that have been made in this chapter.*

Tolstov (2012): *Reprint of a classical textbook originally published in 1962. Fourier theory is developed for series which captures the main points but avoids the conceptual problems of Fourier integrals.*

Suggested Original Papers for Classroom Seminars

Bendor and Wang (2005): *Pitch perception in the auditory cortex is studied with missing fundamental stimuli. The authors present neurons tuned to the periodicity of sound stimuli irrespective of the presence or absence of the fundamental frequency. Pitch or periodicity perception may be based on the autocorrelation function.*

Daugman (1980): *Extends spatial frequency models of the receptive field to the two-dimensional case and gives clean definitions of orientation and frequency specificity.*

Diehl (2008): *The source-filter theory of vowel production is an instructive example of Fourier theory and LSI systems. The glottal sound has a dense and extended spectrum that is filtered by the vocal tract. The MTF of this filtering can be changed by movements of the tongue and lips, i.e., articulation. The paper reviews this theory and discusses its current status.*

Henriksson et al. (2008): *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.*

Yang et al. (2024): *The spatiotemporal frequency content of visual stimuli is analyzed in the presence and absence of visual blinks. The paper shows that short blinks enhance the signal power especially for the low-spatial-frequency contents. A psychophysical study demonstrates that blinking improves the perception of coarse patterns.*

References

Albrecht, D. G., W. S. Geisler, R. A. Frazor, and A. M. Crane. 2002. Visual cortex neurons of monkeys and cats: Temporal dynamics of the contrast response function. *Journal of Neurophysiology* 88: 888–913.

Bendor, D., and X. Wang. 2005. The neuronal representation of pitch in primate auditory cortex. *Nature* 436: 1161–1165.

Bracewell, R.N. 1999. *The Fourier transform and its applications*. 3rd ed. New York: McGraw–Hill.

Campbell, F. W., and D. G. Green. 1965. Optical and retinal factors affecting visual resolution. *Journal of Physiology* 181: 576–593.

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

Diehl, R.L. 2008. Acoustic and auditory phonetics: the adaptive design of speech sound systems. *Proceedings of the Royal Society (London) B* 363: 965–978.

Gonzalez, R.C., and R.E. Wood. 2018. *Digital Image Processing*. 4th ed. New York: Pearson.

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

Marmarelis, P.Z., and V.Z. Marmarelis. 1978. *Analysis of Physiological Systems*. New York: Plenum Press.

Oppenheim, A.V., A.S. Willsky, and S.H. Nawab. 1997. *Signals & Systems*. 2nd ed. Upper Saddle River: Prentice Hall.

Robinson, J.C. 2020. *An Introduction to Functional Analysis*. Cambridge: Cambridge University Press.

Tolstov, G.P. 2012. *Fourier Series*. Garden City: Dover Publications.

Yang, B., J. Intoy, and M. Rucci. 2024. Eye blinks as a visual processing stage. *Proceedings of the National Academy of Sciences* 121: e2310291121.

Artificial Neural Networks and Classification 5

Abstract

In models of large networks of neurons, the behavior of individual neurons is treated in much simpler ways than in the Hodgkin–Huxley theory presented in Chap. 1. Activity is usually represented by a binary variable (1 = firing and 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 each neuron on its neighbors in the network. Synaptic weights may change according to so-called learning rules, which create 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 apply it to an important network architecture used for pattern recognition tasks. Other problems of neural information processing will be addressed in Chap. 6. The mathematical treatment is largely based on linear algebra (vectors and matrices) and, as in the other chapters, will be explained "on the fly."

Learning Objectives

- Elements of artificial neural networks, including activity vectors, weight matrices, activation functions, learning rules, and weight dynamics in discrete time.
- The perceptron as a simple neural classifier and how it can be trained in supervised learning.

(continued)

H. A. Mallot, *Computational Neuroscience*,
https://doi.org/10.1007/978-3-031-75705-1_5

- Mathematical background of artificial neural networks, in particular the dot product, the feature space, and partial derivatives and their role in weight optimization by error minimization.
- Similarities and differences of neural and technical pattern recognition systems.

5.1 Elements of Neural Networks

5.1.1 Background

The idea that the brain is a huge network of threshold gates or switches, each collecting some tens of thousands of inputs and spreading its output again to tens of thousands of other units, has been extremely fruitful in neuroscience, machine learning, and artificial intelligence. It emerged in the first half of the last century as a consequence of a number of neurobiological findings. The first of these is the discovery by Ramón y Cajal[1] that the brain consists of individual neurons that are separated from each other by their cell membranes. Using a staining method developed earlier by Golgi,[2] Cajal was able to fill individual cells completely with silver chromate microcrystals which appear dark in the light microscope. Since only some neurons are filled in each specimen, these neurons stand out clearly against the background of densely packed tissue. Cajal published detailed anatomical drawings of neurons and networks such as the retina, hippocampus, or cerebellum, which are still standard textbook illustrations today (Ramón y Cajal 1899–1904). He therewith became the founder of the "neuron doctrine," that is, the theory that neurons, despite of their many, extended, and delicate fiber arborizations, are completely covered by a cell membrane and that the cytoplasm of adjacent cells remains unconnected.[3] At the same time, he identified different anatomical types of neurons and showed that they form complex networks and circuits. Today, the connectivity of neural networks is studied in the field of connectomics (see, e.g., Helmstaedter et al. 2013).

The second development is the discovery of the all-or-nothing law of the action potential, that is, the fact that action potentials are initiated when the sum of passively conducted potentials arriving at the initial section of the axon (the "axon hillock") exceeds a threshold. The mechanism of the action potential has been discussed in detail in Chap. 1; thresholding behavior is a consequence of

[1] Santiago Ramón y Cajal (1852–1934), Spanish neuroanatomist. Nobel Prize in physiology or medicine 1906.

[2] Camillo Golgi (1843–1926), Italian neuroanatomist. Nobel Prize in physiology or medicine 1906.

[3] An exception to this rule is the electrical synapse, in which charged ions can pass in either direction directly from one neuron to another. We will not discuss electrical synapses any further in this text.

the bifurcation of the membrane (or Hodgkin–Huxley) oscillator illustrated in Fig. 1.12b. All-or-nothing behavior had been observed early on in muscle fibers and efferent[4] nerves. Adrian[5] and Forbes (1922) showed that the same behavior is also present in afferent fibers, in particular somatosensory nerves from the stretch receptors of skeletal muscles (muscle spindles). The all-or-nothing rule is now considered to apply throughout the nervous system, with very few exceptions. One of these is the transmission and processing of the receptor signal in the outer layers of the retina, which are exclusively based on passive potentials with the first action potentials occurring only in the retinal ganglion cells (see also Sect. 3.1.2). The importance of thresholding for information processing was emphasized by McCulloch and Pitts (1943), who suggested that it implements logic gates on which the central processing unit of a Turing machine can be built, see Box 2.6.

The third element of neural network modeling is the weighting of the inputs and their summation into the "generator potential" whose rising above the threshold will then elicit the action potential. This is built after the mechanisms of neural transmission by chemical synapses as investigated by Eccles[6] (1964). In short, synapses are composed of parts of two neurons: a presynaptic neuron emitting "transmitter" molecules and a postsynaptic neuron exhibiting receptor proteins in its membrane to which the transmitter may bind. The two membranes are separated by a narrow extracellular space called the synaptic cleft; it is crossed by the transmitter by way of diffusion. In the simplest case, the receptors are "ligand-gated" ion channels, that is, upon binding of the ligand (in this case the transmitter) they open for ions such as Na^+, K^+, Ca^{2+}, Cl^-, or for combinations thereof. The resulting ion flow between the postsynaptic neuron and the extracellular medium (the synaptic cleft) generates postsynaptic membrane potentials (PSPs) which are passively conducted to the soma and contribute to the generator potential at the axon hillock. The significance of chemical transmission for artificial neural networks is threefold. First, it establishes a polarity of the processing stream: Transmission in a given synapse is always from the pre- to the postsynaptic neuron. Second, it allows for synaptic weighting in spite of the all-or-nothing rule of the action potential: The amplitude of the postsynaptic potential depends on the total quantity of transmitter emitted upon the arrival of an action potential at the presynaptic terminal and also on the quantity of postsynaptic channels. These parameters are also called the synaptic strength. Third, chemical transmission allows to change the sign of the signal. If a Na^+ or Ca^{2+} channel is opened, the effect will be a depolarization which acts as an activation of the postsynaptic neuron. If, however, a K^+ or Cl^- channel is opened,

[4] Efferent nerves conduct effector signals in the "outbound" direction, i.e., from the central nervous system to end organs such as muscles or glands. In contrast, afferent nerves conduct sensory signals back to the central nervous system.

[5] Edgar Douglas Adrian (1889–1977), English physiologist. Nobel Prize in physiology or medicine, 1932.

[6] John Carew Eccles (1903–1997), Australian neurophysiologist. Nobel Prize in physiology or medicine 1963.

the result will be a hyperpolarization, acting as inhibition. Synaptic transmission is therefore modeled in neural network theory by multiplication with a signed number, the synaptic weight.[7]

The final element of neural network modeling is plasticity, most notably synaptic plasticity modeled as the change of transmission weights by usage. The most influential idea is the Hebb[8] synapse in which the simultaneous activation of the pre- and the postsynaptic neuron leads to synaptic growth and an increased synaptic weight (Hebb 1949). It was postulated as a mechanism for the generation of memory "engrams," that is, physical traces of stored memories, and is able to explain simple forms of learning such as classical conditioning in the Pavlovian[9] reflex. The physiology of Hebbian learning is often seen in long-term potentiation (LTP), but other mechanisms may be involved; for reviews see Kandel et al. (2014) and Minatohara et al. (2016). Plasticity may even involve adult neurogenesis, i.e., the differentiation of new neurons from stem cells and their incorporation into the existing network (Kempermann 2019). Figure 5.1 summarizes the key elements of polar neuronal transmission from postsynaptic inputs to presynaptic outputs.

5.1.2 Model

Artificial neural networks are built of three major building blocks inspired by the principles discussed above. These building blocks are:

1. The activity (excitation) of a neuron together with its dynamic evolution over time. As already discussed in the theory of receptive fields, activity is generally modeled not as a membrane potential, but as a number related to the current spike rate (see below). It is discretized into separate time steps where one time step corresponds to the passage of activity from the postsynaptic receptors to the presynaptic terminal of one neuron and to the postsynaptic receptors of the next neuron, which is in the order of 10 ms. Time steps are assumed to be synchronous throughout the entire network.[10]

[7] Note that most neurons release the same transmitters at all their synaptic terminals ("Dale's principle"). Therefore, a given neuron will generally cause either excitation or inhibition in all its postsynaptic cells. All synaptic output weights of a given neuron should therefore have the same sign which cannot be changed by learning. Switching from excitation to inhibition therefore requires an extra interneuron. In artificial neural networks, this rule is usually ignored.

[8] Donald Olding Hebb (1904–1985), Canadian psychologist.

[9] Ivan Petrovich Pavlov (1849–1936), Russian physiologist. Nobel Prize in physiology or medicine 1904.

[10] Global synchrony is probably the single most problematic simplification used in neural network modeling. It implies that artificial neural networks are finite-state machines in the sense of automata theory. In contrast, the fact that interspike intervals are continuous variables renders the set of possible brain states infinite. See also Box 2.6.

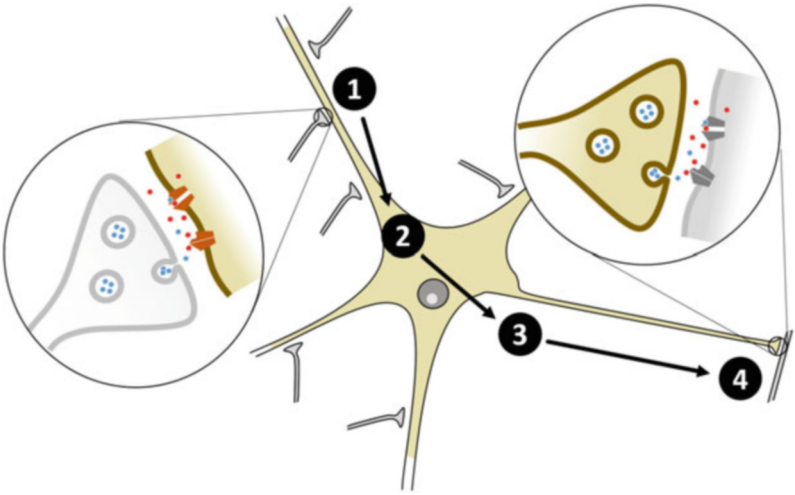


Fig. 5.1 Processing stream of a neuron. The main neuron is shown in color, and components of other neurons are shown in gray. (1) Input is received from a "presynaptic" neuron at a chemical synapse. Synapses can weight the signal by the amount of transmitter emitted upon each arriving spike and the number and efficiency of postsynaptic receptors. The sign of the postsynaptic potential (excitatory or inhibitory) depends on the receptor type and the ions it allows to pass through the membrane. (2) Postsynaptic potentials are propagated along the dendrite and sum up to form the generator potential in the soma. (3) If the generator potential exceeds a threshold, an action potential is initiated at the axon hillock and travels down the axon. (4) At the axon terminals, the electrical signal is transformed back into a chemical signal transmitted to other ("postsynaptic") cells. Blue dots show neurotransmitters and red dots charged ions such as Na^x, K^+, Ca^{2+}, or Cl^-

2. The synaptic weights and learning rules governing the flow of activation in the network. The change of synaptic weights ("weight dynamics") as a result of previous activation pattern is studied as a model of learning. The weight changes also occur in synchrony with the overall time step.
3. The topology of the network is the pattern of connectivity between the individual neurons. It is generally described by a weight matrix, but higher level descriptions such as feedforward versus feedback or layered versus completely connected are also used. We will assume the overall topology fixed, but dynamic changes such as the creation of new neurons in adult neurogenesis are also studied in the literature ("growing networks").

5.1.3 Activation Dynamics

Activation Vector

The neurons in a net are numbered with positive integers $i \in \mathbb{N}$. For each neuron i, the number a_i is the current state of excitation or activity. In "logical" neurons, a_i can take only value 0 or 1. In other models, the activity may be considered a

continuous variable, either in the interval [0, 1] (spiking probability, see Eq. 2.1) or in the real numbers $\mathbb{R}$. If we consider a network of n neurons, we can write the activity states of all neurons as an ordered list or vector

$$\mathbf{a} := \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_I \end{pmatrix} = (a_1, a_2, ..., a_n)^\top. \tag{5.1}$$

If $a_i \in \mathbb{R}$, the set of all possible activity vectors forms a vector space with dimension n; it is denoted $\mathbb{R}^n$. Note that n may be much larger than 3, rendering geometric interpretations of this vector space difficult. In the sequel, we will occasionally give geometric interpretations for $n = 2$ or $n = 3$. For the general case, the activity vector $\mathbf{a}$ should be thought of as an ordered list without trying to imagine n-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 ($\mathbf{a}^\top$ or $\mathbf{a}'$) which works in both directions, i.e., $(\mathbf{a}^\top)^\top = \mathbf{a}$.

The vector $\mathbf{a}$ is also called a state variable of the neural network since it contains the information about the current activity state; in the time-discrete approach, an upper index t, for time, is attached to the state vector,[11] which is then written as $\mathbf{a}^t = (a_1^t, \ldots, a_I^t)^\top$. The upper index will also be used to identify the various patterns of a training set used in neural network learning.

In neural network theory, neurons are identified by their index number and usually do not have a specified location in space. This is different from the situation studied, for example, in Eq. 2.17, where neural layers were modeled as continuous fields and each coordinate point (x, y) corresponded to a neuron. The analogy between the activity vector $\mathbf{a}^t$ and the activity function $a(x, y, t)$ from Chap. 3 can be explicitly stated if we assume that each neuron i has a position (x_i, y_i) within some layer. We then obtain

$$\mathbf{a}^t = \begin{pmatrix} a_1^t \\ \vdots \\ a_n^t \end{pmatrix} = \begin{pmatrix} a(x_1, y_1, t) \\ \vdots \\ a(x_n, y_n, t) \end{pmatrix}. \tag{5.2}$$

Note that on the left side of this equation, the letter a denotes the components of the activity vector $\mathbf{a}$, while on the right side, it denotes the sampled values of the continuous spatiotemporal activity function of a neural layer.

[11] In cases where upper indices can be confused with powers, they are often written with brackets, as in $\mathbf{a}^{(t)}$. We will occasionally make use of this convention.

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)

$$\alpha_i : \mathbf{a}^{t-1} \mapsto a_i^t, \tag{5.3}$$

which takes the complete activity vector of the network at time $t-1$ as its input and produces the activity of neuron i at time t as its output, see Fig. 5.2b. The activation function is usually described by *synaptic weights* w_{ij} and a point nonlinearity f:

$$\alpha_i(\mathbf{a}^{t-1}) = a_i^t = f\left(\sum_{j=1}^{n} w_{ij} a_j^{t-1}\right). \tag{5.4}$$

Examples of static nonlinearities which are also used in neural networks have been given in Sect. 2.3.1. While "logical units" would use the step nonlinearity of Fig. 2.9c, learning in neural networks requires differentiable nonlinearities such as the sigmoidal shown in Figs. 2.9e and 5.2b. The weighted sum forming the argument of the nonlinearity, $\sum w_{ij} a_j^{t-1}$, is called the "generator potential" or simply the potential of cell i and is denoted by u_i^t.

In Eq. 5.4 and throughout this text we use the convention that the weights w_{ij} are indexed with the number of the postsynaptic cell (i) followed by the number of the presynaptic cell (j); 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.

Except for the nonlinearity f, Eq. 5.4 is analogous to the correlation Eq. 2.11, if we identify neurons with points in the plane, as we did before in Eq. 5.2. The sampled image intensities $I(x_j, y_j)$ then become the presynaptic inputs a_j, and the sampled values of the receptive field function $\phi(x_j, y_j)$ become the weights w_{ij} of the sole output unit i. Note, however, that the receptive field function differs from the weight vector in that it models effects of the entire pathway from the retina to the output neuron, not just the transition from the last preceding layer.

Box 5.1 Geometric Properties of the Dot Product

Definition For two vectors of equal dimension n, $\mathbf{a} = (a_1, ..., a_n)^\top$ and $\mathbf{b} = (b_1, ..., b_n)^\top$, the standard dot product is defined as

$$(\mathbf{a} \cdot \mathbf{b}) = (\mathbf{b} \cdot \mathbf{a}) := \sum_{i=1}^{n} a_i b_i.$$

Norm of a Vector The length or "norm" of a vector is given by Pythogoras theorem. It is well defined in any number of dimensions:

$$\|\mathbf{a}\| := \sqrt{\sum_i a_i^2} = \sqrt{(\mathbf{a} \cdot \mathbf{a})}.$$

(continued)

Box 5.1 (continued)
Unit Vector A vector with norm 1 is called a unit vector. For any vector $\mathbf{a} \neq 0$, a unit vector $\hat{\mathbf{a}}$ with the same direction is given by

$$\hat{\mathbf{a}} = \frac{\mathbf{a}}{\|\mathbf{a}\|} = \left(\frac{a_1}{\|\mathbf{a}\|}, \ldots, \frac{a_n}{\|\mathbf{a}\|} \right)^\top .$$

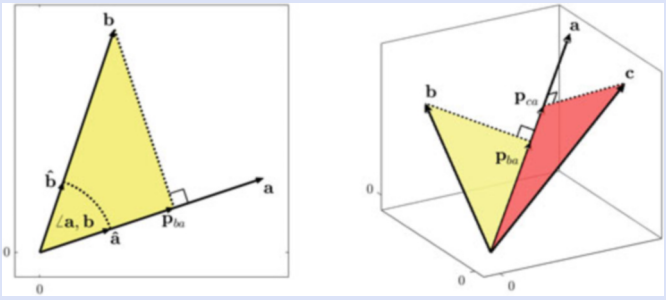
Orthogonality Two vectors $\mathbf{a}, \mathbf{b} \neq 0$ are said to be orthogonal if their dot product vanishes

$$(\mathbf{a} \cdot \mathbf{b}) = 0 \quad \Leftrightarrow \quad \angle \mathbf{a}, \mathbf{b} = \pm 90^\circ .$$

Orthogonality is well defined in any number of dimensions: It suffices to consider the subspace spanned by the two vectors involved, which is always contained in a plane. For vectors including arbitrary angles, the orthogonality rule generalizes to

$$(\mathbf{a} \cdot \mathbf{b}) = ||\mathbf{a}||\ ||\mathbf{b}||\ \cos \angle \mathbf{a}, \mathbf{b}.$$

This relation can also be used as a definition of the cosine function.



Projection The projection of a vector $\mathbf{b}$ on a second vector $\mathbf{a}$ is the point where a perpendicular dropped from the tip of vector $\mathbf{b}$ will hit the line passing through vector $\mathbf{a}$; it is marked $\mathbf{p}_{ba}$ in the figure. From the facts that the perpendicular $\mathbf{p}_{ba} - \mathbf{b}$ and vector $\mathbf{a}$ are orthogonal and $\mathbf{p}_{ba} = \lambda \mathbf{a}$ for some $\lambda \in \mathbb{R}$, it is easy to show that

$$\|\mathbf{p}_{ba}\| = (\mathbf{b} \cdot \hat{\mathbf{a}}) = \frac{(\mathbf{b} \cdot \mathbf{a})}{\|\mathbf{a}\|}.$$

The right part of the figure shows that the same relation holds also in three-dimensional space.

The Activation Function as a Measure of Similarity

The activation function of a given neuron (neuron i, say) involves two vectors, the inputs $\mathbf{a}^{t-1} = (a_1^{t-1}, \ldots, a_n^{t-1})^\top$ and the weights $\mathbf{w}_i = (w_{i1}, \ldots, w_{in})^\top$. Since every input line has its own weight, the dimension of these two vectors is always the same. In the summation step of the activation function, they are combined in a weighted sum, which can be considered a dot product in the sense of vector algebra:

$$u_i = \sum_{j=i}^{n} w_{ij} a_j^{t-1} =: (\mathbf{w}_i \cdot \mathbf{a}^{t-1}). \tag{5.5}$$

The dot product is also known as the inner product, in which case it is often written in matrix notation, $\mathbf{w}_i^\top \mathbf{a}$, 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 in understanding neural network theory, see Box 5.1. The most important one of these is projection: The dot product calculates a measure of the similarity between the incoming pattern $\mathbf{a}$ and the neuron's preferred pattern $\mathbf{w}_i$. In the geometric interpretation in the vector space $\mathbb{R}^n$, similarity relates to the length of the projection of the input vector $\mathbf{a}$ on the weight vector $\mathbf{w}$. If both have unit length, the projection equals the cosine of the angle between the vectors $\mathbf{a}$ and $\mathbf{w}_i$, which will be maximal if the angle is zero (see also Box 5.3).

▶ **Key Point: Activation Function** The activation function of a neuron in an artificial neural network assesses the similarity between an incoming pattern and the neuron's weight vector. Its output is transmitted to other neurons and becomes their input in the next time step.

The Weight Matrix

If we consider the dynamics of multiple units simultaneously, we can collect the linear parts of the activation functions into a matrix equation. For the network shown in Fig. 5.2a, we have

$$\begin{aligned}
u_1^t &= 0\, a_1^{t-1} + 0\, a_2^{t-1} + 0\, a_3^{t-1} + w_{14}\, a_4^{t-1} \\
u_2^t &= w_{21}\, a_1^{t-1} + w_{22}\, a_2^{t-1} + w_{23}\, a_3^{t-1} + w_{24}\, a_4^{t-1} \\
u_3^t &= 0\, a_1^{t-1} + w_{32}\, a_2^{t-1} + 0\, a_3^{t-1} + w_{34}\, a_4^{t-1} \\
u_4^t &= 0\, a_1^{t-1} + w_{42}\, a_2^{t-1} + 0\, a_3^{t-1} + 0\, a_4^{t-1}
\end{aligned} \tag{5.6}$$

In matrix notation, this set of equations can be written as

$$\begin{pmatrix} u_1^t \\ u_2^t \\ \vdots \\ u_n^t \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & \ldots & w_{1n} \\ w_{21} & w_{22} & \ldots & w_{2n} \\ \vdots & \vdots & & \vdots \\ w_{n1} & w_{n2} & \ldots & w_{nn} \end{pmatrix} \begin{pmatrix} a_1^{t-1} \\ a_2^{t-1} \\ \vdots \\ a_n^{t-1} \end{pmatrix} \tag{5.7}$$

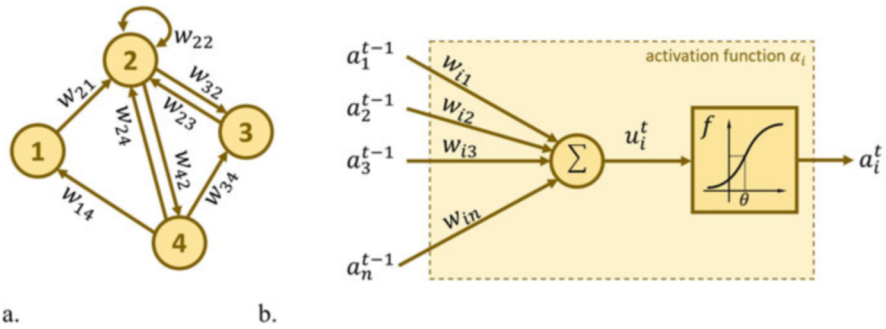


Fig. 5.2 (**a**) A simple neural network consisting of four units ("neurons") with activities $a_1, \ldots, a_4$ and links with transmission weights w_{ij}. For explanations see text. (**b**) Simple model neuron for artificial neural networks. a_j^{t-1}: Input activities, w_{ij} synaptic weights, Σ summation, u_i^t potential ($u_i = \Sigma_{j=1}^n w_{ij} a_j^{t-1}$), f static nonlinearity, and a_i^t output activity ($a_i^t = f(u_i^t)$). Note the analogies with Fig. 5.1

or shorter

$$\mathbf{u}^t = \mathbf{W}\mathbf{a}^{t-1}. \tag{5.8}$$

The matrix $\mathbf{W}$ is called the weight matrix. For each neuron i, row i of the weight matrix is the vector of input weights of this neuron, while column i is the vector of output weights. Note also that the i-th component of the potential vector, u_i^t, is a dot product of the i-th row of the matrix and the input vector $\mathbf{a}^{t-1}$. Weights such as w_{12} which are missing in the figure are simply set to zero.

The weight matrix is square ($n \times n$) unless network topologies with restricted connectivity are considered. The diagonal of the weight matrix contains the weights of each neuron onto itself. The weight matrix is generally not symmetric, since connectivity need not be reciprocal, i.e., the weights w_{ij} and w_{ji} may well be different.

The nonlinearity would then be applied to each component of the potential vector $\mathbf{u}^t$, i.e., $\mathbf{a}^t = (f(u_1^t), \ldots, f(u_n^t))^\top$; for short, we write

$$\mathbf{a}^t = f(\mathbf{u}^t) = f(\mathbf{W}\mathbf{a}^{t-1}). \tag{5.9}$$

The activity vector $\mathbf{a}$ and the weight matrix are the state variables of the network. As for the activity vector, we will use an upper index t for time to study its dynamics.

Box 5.2 Matrix Basics

A $n \times m$ matrix $\mathbf{A}$ is an array of (real or complex) numbers with n rows and m columns. Here, we only consider real matrices. The coefficient in row i and column j is called a_{ij}. Column and row vectors can be considered $n \times 1$ or $1 \times m$ matrices, respectively.

Multiplication An $n \times m$ matrix $\mathbf{A}$ can be multiplied with an $m \times l$ matrix $\mathbf{B}$; the resulting matrix $\mathbf{C} = \mathbf{AB}$ is $n \times l$. The multiplication rule ("row times column") reads

$$\mathbf{C} = \mathbf{AB}; \quad c_{ik} = \sum_{j=1}^{m} a_{ij} b_{jk}.$$

Matrix multiplication is associative: $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$, but not commutative.

Distributive Law Matrices of equal dimensions can be added by adding their components. The distributive law applies: $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$ and $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$.

Linear Mappings The class of real $n \times m$ matrices $\mathbf{A}$ is equivalent to the class of all linear mappings from $\mathbb{R}^m \to \mathbb{R}^n$. They are given as $\mathbf{y} = \mathbf{Ax}$.

Transposition An $n \times m$ matrix $\mathbf{A}$ is said to be the transpose of an $m \times n$ matrix $\mathbf{B}$ if $a_{ij} = b_{ji}$. We write $\mathbf{B} = \mathbf{A}^\top$. Important relations are $(\mathbf{A}^\top)^\top = \mathbf{A}$ and $(\mathbf{AB})^\top = \mathbf{B}^\top \mathbf{A}^\top$.

Squareness and Symmetry A matrix with an equal number of rows and columns ($n = m$) is called square. A square matrix satisfying $\mathbf{A}^\top = \mathbf{A}$ is called symmetric. Its coefficients satisfy $a_{ij} = a_{ji}$.

Identity Matrix A square ($n \times n$) matrix $\mathbf{I}$ with coefficients $a_{ii} = 1$ for all i and $a_{ij} = 0$ for $i \neq j$ is called the identity matrix. It is the neutral element of matrix multiplication, i.e., $\mathbf{IA} = \mathbf{A}$ for all matrices $\mathbf{A}$ with n rows and $\mathbf{BI} = \mathbf{B}$ for all matrices $\mathbf{B}$ with n columns.

Rank The rank of a matrix is the number of linear independent column vectors contained or, equivalently, the dimension of the vector space spanned by the column vectors. The rank of the identity matrix is n.

Matrix Inversion Square matrices with full rank (i.e., rank n) are invertible; that is, there is a matrix $\mathbf{A}^{-1}$ with $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$.

Orthonormal Matrix A square matrix $\mathbf{Q}$ whose column vectors have unit length and are pairwise orthogonal (have dot product zero) is called orthonormal. It will satisfy the equation $\mathbf{QQ}^\top = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$. This implies that the row vectors will also be orthonormal; it also implies $\mathbf{Q}^{-1} = \mathbf{Q}^\top$.

Diagonal Matrices A square matrix with $a_{ij} = 0$ for all $i \neq j$ is called diagonal. If the diagonal elements do not vanish ($a_{ii} \neq 0$ for all i), it is invertible; the inverse is again diagonal with coefficients $1/a_{ii}$.

Eigenvector and Eigenvalue A vector $\mathbf{x}$ whose multiplication with a square matrix $\mathbf{A}$ results in a multiple of itself is called an eigenvector of $\mathbf{A}$. It satisfies the eigenvalue equation $\mathbf{Ax} - \lambda\mathbf{x} = 0$. The number $\lambda \in \mathbb{C}$ is

(continued)

Box 5.2 (continued)
the corresponding eigenvalue. The notions are completely analogous to their usage in Fourier theory, Sect. 4.2, if functions are considered "vectors" in an infinitely dimensional space.

5.1.4 Weight Dynamics ("Learning Rules")

At the core of neural network theory is the introduction of synaptic learning rules: that is, rules for the change of synaptic weights as a function of the network's state and maybe external performance measures. We briefly summarize here the most important learning rules. Further explanations and formal accounts will be given in the course of the presentation.

Hebbian or *unsupervised* learning (Fig. 5.3a) is a completely local learning rule relying on the activities of the directly involved neurons only. The Hebb rule

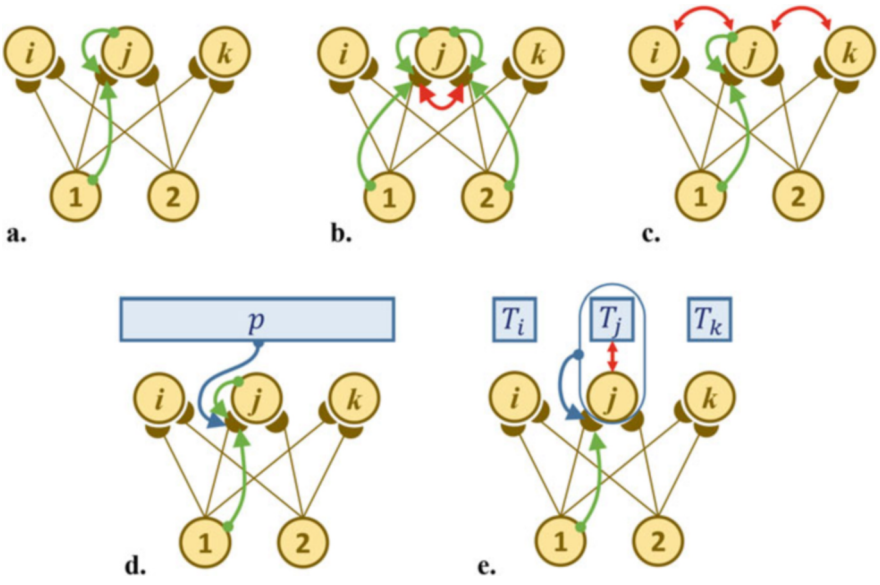


Fig. 5.3 Overview of learning rules. (**a**) Hebbian learning. The change of synaptic weights depends on the coincidence of pre- and postsynaptic signals (a_1^{t-1}, a_j^t, green arrows). (**b**), (**c**) Competitive learning. In addition to the Hebbian component, inhibition (red arrows) prevents neurons from learning the same weights. In b, competition is between the input weights of each neuron, and in c, it is between neighboring output units (red arrows). (**d**) Reinforcement learning. A global payoff signal p^t is available for each output pattern (blue bar and arrow) and acts on all synapses. (**e**) Supervised learning. An external teacher signal T_i^t is available for each neuron and each input pattern (blue boxes). The learning rule then calculates the difference between the desired and the actual output ($\delta_j = T_j^t - a_j^t$, blue arrow)

states that the synaptic weight is increased if the pre- and postsynaptic cells are active in subsequent time steps, i.e., if the presynaptic cell fires and the synapse is "successful" in the sense that the postsynaptic cell fires as well. The Hebb rule thus responds to correlations, as is also expressed by the phrase "fire together, wire together" coined by Shatz (1992). For the initiation of synaptic growth, it does not matter whether the activation of the postsynaptic neuron is actually caused by the presynaptic or by some other signal. For example, in classical (Pavlovian) conditioning, the response neuron ("R", e.g., the motor neuron effecting the secretion of saliva) is initially driven by the unconditioned stimulus ("US", e.g., the sight of a piece of meat). At the beginning of the experiment, the presentation of the conditioned stimulus ("CS", e.g., the ringing of a bell) has no effect. By the coincident activity of CS and R, however, a synapse connecting CS and R (if it at all exists) will see both pre- and postsynaptic activities, the former one caused by presentation of the CS and the latter one by that of the US. It will therefore grow such that eventually the CS will be able to elicit the response on its own. Hebbian correlation of pre- and postsynaptic activities is also present in the learning rules discussed below, which, however, take additional factors into account.

In *competitive learning*, the weight change is determined not only by the activities of the pre- and postsynaptic neuron but also by the activities of other "competing" neurons. Competition may be between the synaptic inputs of a given neuron (Fig. 5.3b), between neighboring neurons (Fig. 5.3c), or both. Competitive learning is used in models of the self-organization of representations or the unsupervised discovery of regularities in data sets. Physiologically, competition may be realized by inhibition between output neurons ("winner-take-all" networks) or via the limitation of resources needed for synaptic growth.

In *reinforcement learning*, the weight change is determined by the activities of the pre- and postsynaptic neurons and by a "payoff" or "reward" signal carrying information about the overall performance of the network (Fig. 5.3d). The payoff signal is one global variable transmitted to all units in the network, telling them whether the last action was successful or not; it does not say what should have been done instead. Reinforcement learning is also known as trial-and-error learning or operant conditioning. In physiology, the payoff signal is thought to correspond to the activity of dopaminergic neurons in the reward system of the brain (Schultz et al. 1997).

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 (Fig. 5.3e). The network is trained with a "labeled" data set providing this teacher signal for each input. The most popular schemes for supervised learning, *backpropagation* and *deep learning*, are based on the iterative minimization of the deviation between actual output and teacher signal. Physiological correlates of the teacher signal do not seem to exist. The significance of supervised learning for computational neuroscience lies not so much in its ability to model the actual mechanisms of learning, but in the existence proof of a network for a given problem.

► **Key Point: Learning Rules** Learning rules describe the update of synaptic weights based on neural activities of the pre- and postsynaptic neurons and external factors such as payoff and teacher signals. They constitute the weight dynamics of the neural network.

5.2 Classification

We now turn to the discussion of a specific type of neural network, namely networks for classification. The word "class" is used here to denote a set of objects, patterns, exemplars, or "specimen," which share certain properties and whose belonging to the class allows certain inferences to be made. Classification is often considered a key operation in neural information processing which also underlies many other performances. For example, a frog might classify a retinal image as that of a fly appearing in a close-by position, and this classification together with prior knowledge about flies may then trigger a feeding attack. The inference takes the form of a "syllogism" of classical logic in which the classification is premise 1, in the example "the imaged object is a fly." Premise 2 would be that members of the fly class are edible and the conclusion is to eject the tongue in the corresponding direction. The decision on the class membership of the retinal image is a "predicate," that is, a logical statement that can take the values "TRUE" (1) or "FALSE" (0), but no intermediate values. A central idea of neural network theory is to treat the activation function of a neuron as a predicate of this type, where the specimen is the incoming activity pattern and the class description is given by the weight vector. This idea is known as the "perceptron," after Rosenblatt (1958).

For the time being, we assume the classes to be known. In this situation, classification is the same as object recognition in the sense that each object corresponds to a class or set of stimuli. Neural networks for classification are trained in supervised learning schemes with labeled data sets that provide a teacher signal for each input pattern. This teacher signal contains both, the knowledge about the possible class and the potential membership of the current pattern. The problem of finding relevant but hitherto unknown classes from large bodies of input patterns will be called categorization in this book and is addressed in the section on competitive learning.

5.2.1 The Perceptron

Figure 5.4 shows a simple example of a classification carried out by a neuron with just two inputs; let us say from two light sensitive receptors positioned side by side. Of course, there are not too many classes into which such two-pixel "images" can be put, but the general principle can be nicely illustrated in this case. One possible class could be that of all images getting brighter to the right, i.e., the rightward intensity gradients. How can we build a detector for this question, that is:

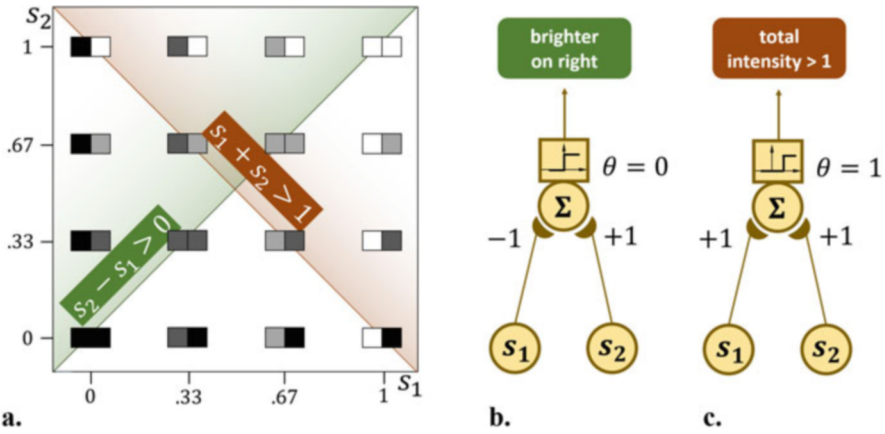


Fig. 5.4 Two simple perceptrons with only two input lines. (**a**) Feature space of two-pixel "images" (intensities s_1 and s_2), decision boundaries, and response areas. (**b**) Perceptron implementing the predicate "brightness increasing to the right." (**c**) Perceptron implementing the predicate "total intensity above 1." All patterns activating perceptron b lie in the upper left (greenish) triangle of the feature space, while all patterns activating perceptron c lie in the upper right triangle (reddish)

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. 5.4b). If we apply the step nonlinearity 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, that is, the set of all possible two-pixel images. With image intensities between 0 and 1, this amounts to the interval $[0, 1] \times [0, 1] \subset \mathbb{R}^2$ shown in Fig. 5.4a. 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, while other patterns will elicit no response. In the example of Fig. 5.4b, all light intensity pairs whose right pixel is brighter than the left one fall in the greenish triangle above and left of the diagonal $s_1 = s_2$. Figure 5.4c shows a second example of a perceptron with two inputs calculating the predicate "average brightness above 0.5." All patterns satisfying this condition are right and above the line $s_1 + s_2 = 1$ shown as a reddish triangle in Fig. 5.4a.

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 or optical nerve fibers which in humans is in the order of 10^6. A class is conceived of as a region or a subset (usually without holes) in feature space that contains all exemplars of the class and no other patterns. In the examples of Fig. 5.4, these are the triangular areas discussed above. For an

incoming pattern, the problem of classification then amounts to the decision whether the pattern falls into the class region or not.

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 $\mathbf{s}$ and with weights $w_1, ..., w_n$ forming a weight vector $\mathbf{w}$. The weighted sum of the inputs is passed through a step nonlinearity with threshold θ

$$u = \sum_{j=1}^{n} w_j s_j = (\mathbf{w} \cdot \mathbf{s}) \tag{5.10}$$

$$a = f(u) := \begin{cases} 0 \text{ if } u \leq \theta \\ 1 \text{ if } u > \theta. \end{cases} \tag{5.11}$$

The perceptrons in Fig. 5.4b, c have $\mathbf{w} = (-1, 1)^\top$ with $\theta = 0$ and $\mathbf{w} = (1, 1)^\top$ with $\theta = 1.0$, respectively. Note that the weight vector $\mathbf{w}$ has the same dimension as the stimulus vector and can therefore be interpreted as a vector in feature space. It may, however, take negative values, as is the case in the example of Fig. 5.4b.

5.2.2 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," that is, a structure in feature space which separates the stimuli that the perceptron does or does not respond to. Crossing this boundary will mean that the potential u passes the threshold; the boundary is therefore determined by setting $u = \theta$ in Eq. 5.11. In the resulting equation $(\mathbf{w} \cdot \mathbf{s}) = \theta$, the weight vector $\mathbf{w}$ is a constant and, for two-dimensional feature spaces, the set of all patterns $\mathbf{s}$ satisfying the equation is a straight line orthogonal to $\mathbf{w}$, passing the origin with distance $\theta/\|\mathbf{w}\|$ (Fig. 5.5, left part). 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 and to the summation part of the activation function (Eq. 5.10).

Another way to think of linear classification relates to the projection property of the dot product in Eq. 5.11 (cf. Box 5.1). The potential u, i.e., the projection of the input on the weight axis, will be the same for all input vectors falling on a common line orthogonal to the weight vector. The classification itself can then be carried out on the weight axis. The result is of course the same as in the intuition given before.

In perceptrons with three input lines ($n = 3$), the feature space will have three dimensions, see the right part of Fig. 5.5. The weight vector still defines a straight line in that space. At a distance $\theta/\|\mathbf{w}\|$ along that line, the potential will take the value θ. The decision boundary then becomes a plane orthogonal to the weight vector and passing through the point $\theta\mathbf{w}/\|\mathbf{w}\|^2$ on the weight vector line.

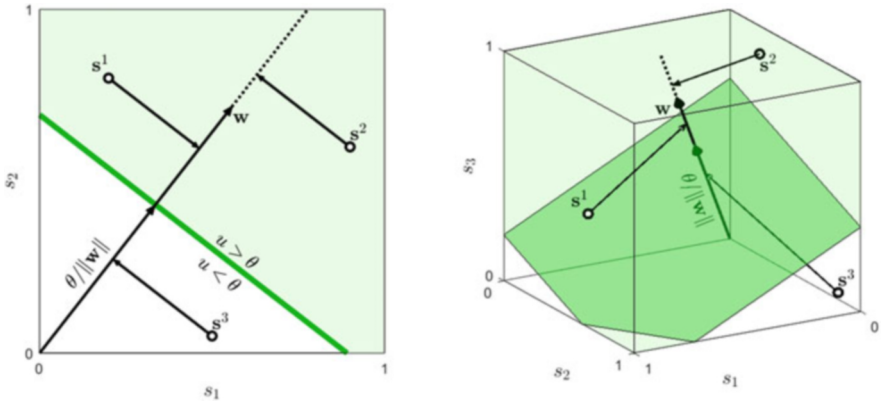


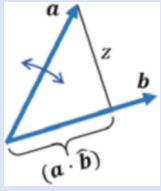
Fig. 5.5 Linear classification. **Left.** The perceptron classifies feature vectors to one side of the linear hyperplane (heavy green line) $u = (\mathbf{w} \cdot \mathbf{s}) = \theta$ in feature space. The distance at which this hyperplane passes the origin is $\theta/\|\mathbf{w}\|$. Each input vector $\mathbf{s}^t$ is projected onto the weight vector direction, yielding the potential u. The projection is orthogonal if we assume $\|\mathbf{w}\| = 1$. The class detected by this perceptron corresponds to the light green area. **Right.** Feature space $[0, 1]^3$ for a perceptron with three input lines. As before, the decision boundary is the hyperplane orthogonal to the weight vector, passing the origin at distance $\theta/\|\mathbf{w}\|$. It is shown as a green plane clipped to the unit cube. The class for this perceptron consists of all points above the decision plane shown as a greenish volume. Note that the origin of the feature space appears in the back, for better visibility. $\mathbf{w}$: weight vector and $\mathbf{s}^1, \ldots, \mathbf{s}^3$: sample input vectors

Mathematically, this plane is still defined by the equation $(\mathbf{w} \cdot \mathbf{s}) = \theta$, 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 an n-dimensional feature space will always be a subspace with dimension $n - 1$, cutting the feature space into two parts. Such subspaces are called "hyperplanes." They relate to high-dimensional "hyperspaces" in the same way as an ordinary plane relates to ordinary three-dimensional space.

Box 5.3 The Cauchy–Schwarz Inequality

Consider two vectors $\mathbf{a}$, $\mathbf{b} \neq 0$. As we saw in Box 5.1, the length of the projection of $\mathbf{a}$ on the direction defined by $\mathbf{b}$ is given by

$$\mathbf{p}_{ab} = (\mathbf{a} \cdot \hat{\mathbf{b}}) = \frac{(\mathbf{a} \cdot \mathbf{b})}{\|\mathbf{b}\|}.$$



(continued)

Box 5.3 (continued)
Let z denote the length of the perpendicular dropped from $\mathbf{a}$ on $\mathbf{b}$. From the Pythagoras theorem, we have

$$\|\mathbf{a}\|^2 - z^2 = (\mathbf{a} \cdot \hat{\mathbf{b}})^2 = \left(\frac{(\mathbf{a} \cdot \mathbf{b})}{\|\mathbf{b}\|} \right)^2 .$$

Since $z^2 \geq 0$, this implies

$$\|\mathbf{a}\|^2 \geq \frac{(\mathbf{a} \cdot \mathbf{b})^2}{\|\mathbf{b}\|^2}$$

and further

$$(\mathbf{a} \cdot \mathbf{b}) \leq \|\mathbf{a}\| \|\mathbf{b}\|.$$

Equality obtains if and only if $\mathbf{a}$ and $\mathbf{b}$ are "collinear," that is, if $\mathbf{a} = \lambda \mathbf{b}$ for some $\lambda \in \mathbb{R}$. In this case, $z = 0$.

The last formula is known as the Cauchy–Schwarz inequality. It holds not only in n-dimensional spaces for arbitrary choices of n but also in infinite-dimensional function spaces where a dot product is given by

$$(f \cdot g) = \int f(x) g(x) \, dx,$$

see also Box 2.2.

The significance of the Cauchy–Schwarz inequality for neural network theory lies in its relation to the problems of the "optimal stimulus" (i.e., the stimulus most strongly activating a neuron with a given weight vector) and the "matched filter" (i.e., the weight vector most suitable to detect a given pattern): If the norm (length) of the input vector of a perceptron is fixed, the response will be maximal, if the stimulus and weight vectors are colinear, that is, if the input vector equals the weight vector up to a constant factor. Note that for stimulus vectors representing images, the norm is just the total contrast of the image. Normalization will therefore not affect the image contents.

From the discussion so far it may appear that a perceptron will always respond to the patterns that fall beyond the decision plane where $(\mathbf{w} \cdot \mathbf{s}) > \theta$, not to those on the side of the coordinate origin where $(\mathbf{w} \cdot \mathbf{s}) < \theta$. Perceptrons for such "absence" classes can, however, be constructed by observing that the condition $(\mathbf{w} \cdot \mathbf{s}) < \theta$ is

equivalent to $(-\mathbf{w} \cdot \mathbf{s}) > -\theta$. We have seen already that the synaptic weight may take any sign; going from $\mathbf{w}$ to $-\mathbf{w}$ is therefore no problem. Negative thresholds may seem biologically implausible at first glance but may be realized by a strong additional input that leads to a firing event whenever inhibitory potentials remain low. To account for this problem (and also others), the threshold is often treated as another weight ($w_{n+1} := -\theta$) applied to a constant input $s_{n+1} \equiv 1$, such that $-\theta = w_{n+1} s_{n+1}$. The activation function of Eq. 5.4 then becomes

$$a^{t+1} = \alpha(\mathbf{s}^t) = h\left(\sum_{i=1}^{n} w_i s_i^t - \theta\right) = h\left(\sum_{i=}^{n+1} w_i s_i^t\right), \tag{5.12}$$

where h is the binary switching function from Eq. 2.27.

Optimal Stimulus

On inspection of Fig. 5.4b, it can be seen that the perceptron detecting a brightness increase to the right has a weight vector $\mathbf{w} = (-1, 1)^\top$, which shows a rightward increase of weight just as the image to be detected by the perceptron shows a rightward increase in intensity. This observation reflects a general principle, namely 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 Sect. 2.2.1 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 potential is therefore doubled by doubling input intensity, this question is only meaningful if we normalize the input pattern, i.e., if we consider only stimuli $\mathbf{s}$ satisfying $\|\mathbf{s}\| = 1$. In this case, we call

$$\mathbf{s}^* := \underset{\mathbf{s}}{\operatorname{argmax}}(\mathbf{w} \cdot \mathbf{s}) \tag{5.13}$$

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 $\mathbf{s}$ and $\mathbf{w}$ point in the same direction

$$\mathbf{s}^* = \lambda \mathbf{w} \quad \text{for some} \quad \lambda \in \mathbb{R}. \tag{5.14}$$

Formally, this result is a consequence of the Cauchy–Schwarz inequality discussed in Box 5.3. The situation is completely analogous to the discussion of optimal stimuli and matched filters in Sect. 2.2.1. In fact, if we consider continuous layers of neurons (Eq. 5.2), 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 Eq. 2.11.

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

$$\bar{s} := \frac{1}{n}\sum_{i=1}^{J} s_i \quad \text{and} \quad \bar{w} := \frac{1}{n}\sum_{i=1}^{J} w_i, \tag{5.15}$$

we obtain

$$\text{cov}(\mathbf{s}, \mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(s_j - \bar{s})(w_i - \bar{w}) = \frac{1}{n}(\mathbf{s} \cdot \mathbf{w}) - \bar{s}\bar{w}. \tag{5.16}$$

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

5.2.3 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. 5.4. 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, linear separability (e.g., the XOR-problem) and the locality of the perceptron.

Linear Separability

Linear classification requires that a hyperplane in feature space can be found such that the members and nonmembers of the class fall on different sides of this plane. This is not always possible. Suppose, for example, that we want to construct a two-pixel perceptron that responds with $a = 1$ if one and only one of its inputs is active. In logic, the related predicate is called "exclusive or" or XOR. The four possible inputs, $\mathbf{s}^i = (0, 0), (0, 1), (1, 0), (1, 1)$, form a square in the two-dimensional feature space (Fig. 5.6d). 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.

This problem can be solved by using cascades of linear classifiers or multilayer perceptrons. For the XOR-problem, we need three initial perceptrons detecting the presence of s_1 and s_2, as well as the "inclusive or" operation, responding to the presence of s_1, s_2, or both. They form the input of a fourth, higher level perceptron denoted as output neuron a in Fig. 5.6a. The initial perceptrons are called "hidden" because their activities do not explicitly show up in the classification procedure; they are neither input nor output. We denote the activity of the hidden units by the

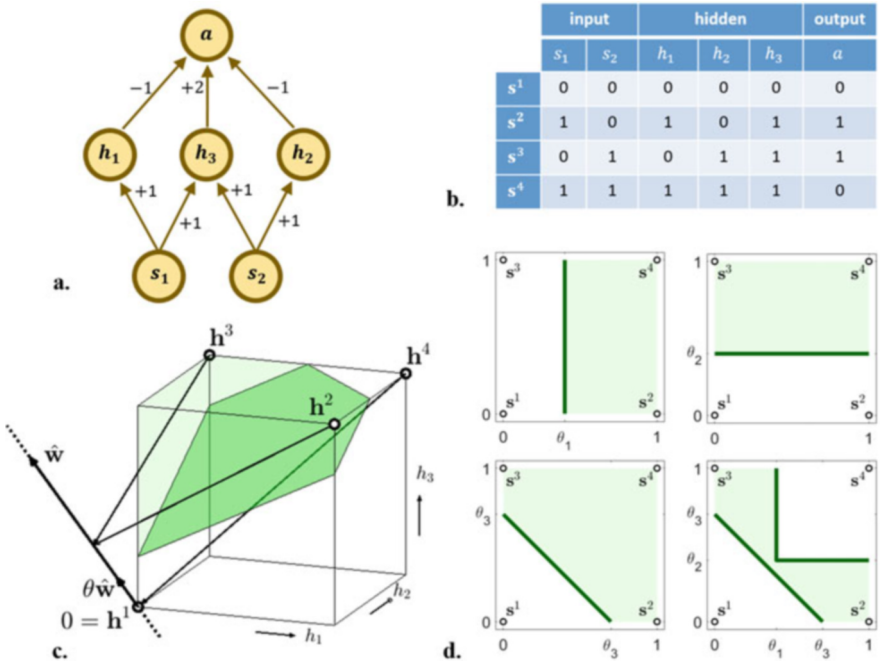


Fig. 5.6 Three-layer perceptron implementing an "exclusive or" (XOR) gate. (**a**) Network topology with two input lines and three hidden units. The numbers next to the arrows are the synaptic weights. (**b**) Truth table for the four possible binary input patterns $\mathbf{s}^1, \ldots, \mathbf{s}^4$. (**c**) Three-dimensional feature space for the final unit, taking the hidden units as input lines. The possible inputs $\mathbf{s}^1, \ldots, \mathbf{s}^4 \in \mathbb{R}^2$ are mapped to the activity patterns of the hidden layer, $\mathbf{h}^1, \ldots, \mathbf{h}^4 \in \mathbb{R}^3$, which are marked by black circles on the cube. In the three-dimensional feature space, linear separation of "true" and "false" cases with a decision plane is possible. The decision plane is indicated in green; $\mathbf{w}$ denotes the weight vector, $\mathbf{w} = (-1, -1, 2)^\top$. (**d**) Decision areas for the original s_1, s_2 feature space. Top and lower left show the feature spaces for the hidden units 1–3 with thresholds $\theta_{1,2} = 0.4$ and $\theta_3 = 0.7$. The decision region for the overall network (output neuron a) appears in the lower right. The decision boundary is no longer a straight line and allows the desired classification for the XOR-problem

letter h. Figure 5.6a shows the synaptic weights as numbers; the thresholds for all units can be set to anything between 0 and 1. With this network the XOR-problem can be solved, as is indicated in the truth table appearing in Fig. 5.6b.

This example shows that perceptrons can do more than just linear classification if we allow for multiple layers. The advanced capability is owed to the hidden units, which embed the classification problem into a high-dimensional space. Figure 5.6c shows the position of the resulting $\mathbf{h}$ vectors for the XOR-problem at four 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, namely the plane $-h_1 + 2h_{12} - h_2 = \theta$ also shown in the figure. The "true" cases (0, 1, 0) and (1, 0, 0) fall

above this plane and yield the output 1, and the "false" cases (0, 0, 0) and (1, 1, 1) lie below in the decision plane and yield the output 0.

When projected back into the s_1- and s_2-plane, the decision boundary is no longer linear (i.e., a line), but a polygon with the desired separation properties. This is shown in Fig. 5.6d. The decision boundaries of the two-layer perceptrons h_1 through h_3 are straight lines, while the overall decision boundary is only piecewise linear or polygonal. It is composed of sections of the decision boundaries of the hidden units and shows the increased flexibility gained by adding more layers. Indeed, it is possible to produce virtually any decision boundary in a three-layer perceptron with a sufficient number of hidden units (Minsky and Papert 1988; Cybenko 1989).

Locality

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 for which we have discussed the solution in the previous section. For larger numbers of input lines, solutions are also possible in three-layer perceptrons. One can show, however, that each such perceptron will need at least one hidden unit looking at all inputs simultaneously (Minsky and Papert 1988, theorem 3.1.1). Similarly, if the "connectedness" of complex geometrical shapes such as intertwined spirals is considered, one hidden unit is required receiving input from the entire input layer (Minsky and Papert 1988, theorem 0.8). This is in conflict with the idea of breaking down vision or other problems of information processing into local bits and pieces, each processed by a unit with a small receptive field. After the original publication of Minski and Papert's book in 1969, this problem 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. In addition, recent work on deep learning clearly shows the power of the perceptron approach in visual information processing and beyond (Krizhevsky et al. 2017).

► **Key Point: Linear Classification** Simple model neurons (two-layer perceptrons) fire if the weighted sum of inputs exceeds a threshold. This implements a linear classification where the decision boundary in feature space is a hyperplane orthogonal to the weight vector. Polyhedral (piecewise linear) decision surfaces can be generated in multilayer perceptrons.

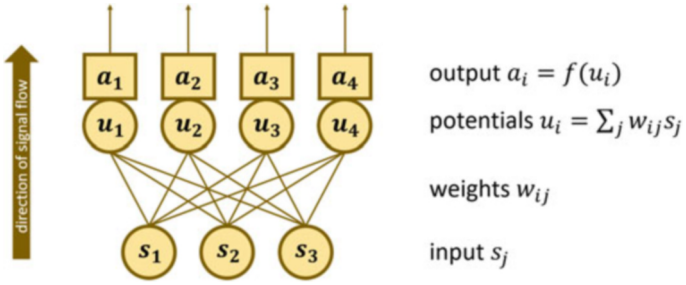


Fig. 5.7 Set of four two-layer perceptrons $a_1, \ldots, a_4$ with common input vector $s_1, \ldots s_3$. The topology is a complete feed-forward connectivity between the input and output layers. Note that numbering is within layers; for example, the weight w_{22} is between different neurons (i.e., s_2 and u_2), not recurrent from neuron 2 onto itself, as was the case in Fig. 5.2a

5.3 Supervised Learning and Error Minimization

Until now, we have assumed that the weights are given, or set by the modeler, which of course raises the question how this can be achieved. Mathematically, there exists an elegant solution which, for multilayer perceptrons, is known as backpropagation, an example of supervised learning. We start our discussion with the two-layer perceptron.

Consider a perceptron whose input layer is an array of pixels of a visual sensor such as a camera target or the retina. On this array, visual patterns are presented which are to be classified by the perceptron. 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 same retina may be used by many perceptrons each checking for their own pattern, for example, the other letters of the alphabet. The architecture is shown schematically in Fig. 5.7. 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, $T(\mathbf{s}) = 1$ if $\mathbf{s}$ is the image of the capital letter "A" and $T(\mathbf{s}) = 0$ if it is not. $T(\mathbf{s})$ is called the teacher signal, and a data set together with the teacher signal for each pattern, $\{(\mathbf{s}^t, T(\mathbf{s}^t))\}$, is said to be "labeled." The question is then how can we optimize the weights such that the perceptron produces the desired outputs T for each input presentation? Once this is achieved, the learning phase ends, and the network is run on novel stimuli.

5.3.1 Two-Layer Perceptron

The original learning rule for perceptrons was derived from a simple heuristic. Assume a unit's output is higher than the teacher signal, that is, the network produces a false alarm. In this case, the input weights that have been carrying activation should be reduced. Likewise, if the output is below the teacher signal, the pattern is missed and the weights of active input lines have to be increased.

Weights of non-active input lines are left unchanged in all cases. Formally, this rule can be written as

$$\Delta w_{ij} = w_{ij}^{t+1} - w_{ij}^{t} = \eta(T_i^t - a_i^t)s_j^{t-1}, \tag{5.17}$$

where t is again the time step. The weight change is positive (increasing) if the output was too low and negative (decreasing) if the output was too high. If the response was correct, no weight changes are applied. $\eta \in \mathbb{R}^+$ is a "learning rate" which is set to some suitable value.

Minsky and Papert (1988) present a proof for the "perceptron convergence theorem," which states that for logical neurons and $\eta = 1$, this learning rule will converge and find a correct solution as long as such a solution at all exists, that is, as long as the trained classes are linear separable. In practical applications, η is often gradually decreased with each learning step. This procedure is known as "annealing"; it will bring learning to an end, but the convergence to the correct solution cannot be guaranteed.

The perceptron learning rule given in Eq. 5.17 assumes that the weights that transmitted a signal in the last presentation can be held "responsible" for the achieved output and should therefore be adjusted if the output was wrong. However, this type of credit assignment to individual weights is often not possible in complex network topologies. In a more systematic approach, learning is therefore treated as an optimization problem. To this end, we define the performance error of the network by comparing actual and desired outputs. This is formulated in terms of an "objective 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 Eq. 5.12 and omit the time indices:

$$a_i = f\left(\sum_{j=1}^{n+1} s_j w_{ij}\right) = f((\mathbf{s} \cdot \mathbf{w})). \tag{5.18}$$

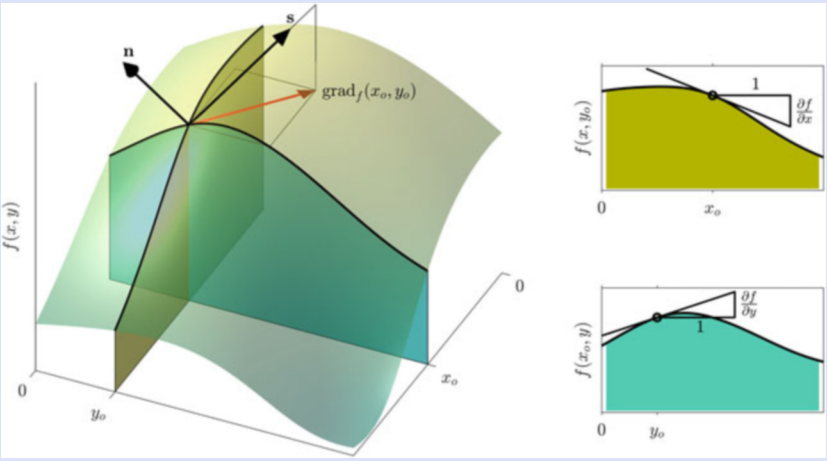
Since we know what the response of the perceptron to a stimulus $\mathbf{s}$ should be—it is defined by the teacher signal $T(\mathbf{s})$—we can now define a performance measure of the perceptron via the squared error E :

$$E(\mathbf{w}, \mathbf{s}) := [a(\mathbf{s}) - T(\mathbf{s})]^2 = \left[f\left(\sum_{j=1}^{n+1} s_j w_j\right) - T(\mathbf{s})\right]^2. \tag{5.19}$$

E is the squared error for a single input pattern $\mathbf{s}$. In "online" learning, it would be used to adjust the weights after each stimulus presentation. The perceptron would then do better for the last presented pattern, but not necessarily for the next one. Still, in the long run, online learning gives good results. It is also more efficient than "offline" learning in which the entire stimulus set is presented once and the average

error over all presentations is calculated and used for one subsequent optimization step. In any case, we treat the argument **s** of the error function as a constant and consider the effects of changes of **w** only.

Box 5.4 Partial Derivatives and the Gradient of a Multivariate Function



Consider a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, i.e., a function of n variables that evaluates to a number. For $n = 2$, we can plot this function as a surface or landscape as shown in the left part of the figure. For univariate functions (i.e., $n = 1$), the derivative at a point x_o is $f'(x_o)$ which can be thought of as the inclination of a tangent touching the plot of the function. For $n = 2$, this idea generalizes to that of a tangent plane characterized by the *normal* vector **n** also marked in the figure. Other characteristics of the tangent plane are the inclinations in the x and y directions and the slope vector **s** pointing into the steepest uphill direction.

The tangent plane is related to the n partial derivatives of f, which are defined as

$$\frac{\partial f}{dx_i} = f_{x_i} = \lim_{h \rightarrow 0} \frac{f(x_1, ..., x_i + h, ..., x_n) - f(x_1, ..., x_i, ..., x_n)}{h}$$

for each coordinate direction. They can be thought of as ordinary derivatives of sections passing through the landscape in the coordinate directions and define the inclinations in the coordinate directions as is shown in the figures to the right.

The vector of all partial derivatives is called the gradient. In differentiable functions, it is defined for all coordinate values and is therefore itself a

(continued)

Box 5.4 (continued)
function $\mathbb{R}^n \to \mathbb{R}^n$, i.e., a vector field. Setting $(x_1, ..., x_n) = \mathbf{x}$, we write

$$\text{grad}_f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f}{\partial x_n}(\mathbf{x}) \right)^\top .$$

The gradient vector at point (x_o, y_o) is shown as a red arrow in the figure. It is the direction of the steepest ascent in the landscape which equals the direction of the slope vector when projected into the coordinate plane. The length of the gradient vector corresponds to the maximal local inclination and is zero in a horizontal plane, on top of a peak, or at the ground of a trough.

As an exercise, the reader may want to show that the unit vectors $\mathbf{n}$ and $\mathbf{s}$ in the figure are given by the following expressions, where $\wedge$ denotes normalization:

$$\mathbf{n} = (-f_x, -f_y, 1)^\wedge \quad \text{and} \quad \mathbf{s} = (f_x, f_y, f_x^2 + f_y^2)^\wedge .$$

5.3.2 Gradient Descent

The task, then, is to minimize $E(\mathbf{w})$ in the components of $\mathbf{w}$. Minimization of functions of several variables such as $E(\mathbf{w})$ 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 applied:

The error function represents a "surface" over the space spanned by its variables, in our case $w_1, ..., w_n$ (see Box 5.4). Every error value $E(w_1, \ldots, w_n)$ is an "elevation" in that landscape, and the optimum (minimum) corresponds to the deepest "depression" or trough in that landscape.

Optimization is carried out as an "iterative" or stepwise process in which an initial value $\mathbf{w}^0$ for the weights is chosen (usually at random) and a direction $\Delta\mathbf{w}$ is sought such that $E(\mathbf{w}^0 + \lambda\Delta\mathbf{w}) < E(\mathbf{w}^0)$ for some λ, that is, $\Delta\mathbf{w}$ is a direction in which the error value decreases. This direction can be found by calculating the gradient $\text{grad}_E(\mathbf{w}^0)$ as explained in Box 5.4. The weights are then set to $\mathbf{w}^1 = \mathbf{w}^0 - \lambda\text{grad}_E(\mathbf{w}^0)$ which completes the first iteration step; the next step starts from $\mathbf{w}^1$, and so on.

The choice of the step length λ is crucial. It should be short enough to avoid jumps over small valleys and large enough to enable fast results. In numerical analysis, elaborate methods for step length control have been developed. In neural network modeling, λ is usually not optimized but chosen "small enough." Indeed, in online learning, where every learning step uses a different pattern and therewith reacts to a different objective function, step length control may be unnecessary.

The optimization process is terminated as soon as the gradient drops below some threshold value. This is the case not only if the bottom of a trough is reached but also if the error function is locally flat. If multiple troughs or "local minima" exist, the result will depend on the initial value from which iteration started. The set of all starting points, from which a given local minimum is reached, is also called this minimum's "basin of attraction."

5.3.3 The δ-Rule

Iterative optimization procedures such as the one just described can be used as models of biological learning where each iteration corresponds to one learning step. 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 may treat $\mathbf{s}$ in Eq. 5.19 as a constant and reduce the error value E by an appropriate adjustment of $\mathbf{w}$. Following a suggestion by Widrow and Hoff (1960), we calculate the partial derivatives of $E(\mathbf{w})$ with respect to the weights, using the chain rule. For component w_k, we obtain

$$\frac{\partial}{\partial w_k} E(w_1, ..., w_n) = \frac{\partial}{\partial w_k} [a - T]^2 = 2[a - T] \frac{\partial a}{\partial w_k}. \tag{5.20}$$

Next, we observe that $a = f(u) = f(\sum_j w_j s_j)$ (Eq. 5.18) and apply the chain rule a second time:

$$\frac{\partial}{\partial w_k} E(w_1, ..., w_n) = 2[a - T] f'(u) \frac{\partial u}{\partial w_k}. \tag{5.21}$$

Keeping in mind that $u = \sum_j w_j s_j$, we find that $\partial u / \partial w_k = s_k$ since in the sum all terms with $j \neq k$ are independent of w_k and therefore yield the derivative zero. We obtain

$$\frac{\partial E}{\partial w_k}(w_1, ..., w_J) = 2\,[a - T]\;\; f'(u)\; s_k. \tag{5.22}$$

When calculated for all k, Eq. 5.22 gives the gradient direction in the error landscape E. By comparison with Eq. 5.17, 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

$$\delta := T - a \tag{5.23}$$

and call δ the correction signal. Including the term $f'(u)$ into the learning rate $\eta \in \mathbb{R}^+$, we obtain the so-called δ or Widrow–Hoff learning rule:

$$\Delta w_k = \eta\, \delta\, s_k \quad \text{for} \quad k = 1, \ldots, n + 1. \tag{5.24}$$

Here, k runs up to $n + 1$ to allow for the extra weight replacing the threshold, see Eq. 5.12.

The learning process is illustrated in Fig. 5.8, except for the fact that in online learning, each learning step responds to the performance with the last presented pattern, not to the overall performance over all training patterns. The error landscape on which minimization is performed therefore changes in every learning step. As pointed out before, this somewhat surprising approach works better than offline learning, where each weight vector would be tested with all stimuli before performing the next weight adjustment step. This is at least partially due to the problem of overlearning, that is, the learning of idiosyncrasies of the training set which may arise from statistical fluctuations. 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., for a layer of perceptrons as depicted in Fig. 5.7, we can consider each output neuron individually and obtain

$$\Delta w_{ij} := \eta \delta_i s_j = \eta (T_i - a_i) s_j . \tag{5.25}$$

▶ **Key Point: Perceptron Learning Rule** The weights of synapses active in the last decision are increased if the output was too low and decreased if the

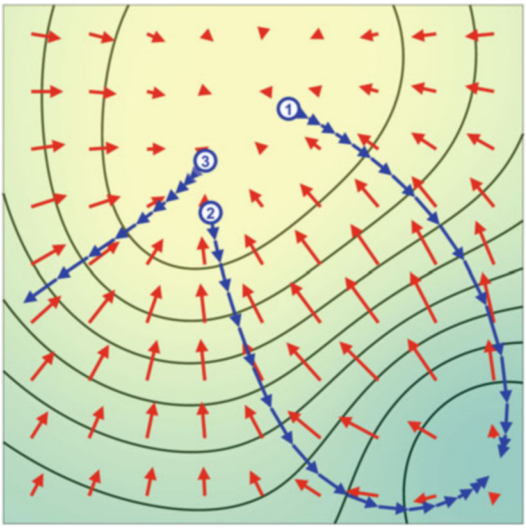


Fig. 5.8 Gradient field and gradient descent. The figure shows a two-dimensional error function $E(w_1, w_2)$ as a contour plot. Superimposed are the gradients at each position (red arrows). Note that they are orthogonal to the contour lines and longer in regions where the contour lines are dense. Following the arrows from most starting points will lead to the maximum of the function in the upper central area. Likewise, following the reverse direction, $-\text{grad}_f(x, y)$, leads to smaller functional values. The blue arrows show gradient descent trajectories from three different starting points marked 1–3. While trajectories 1 and 2 approach the minimum in the lower right, trajectory 3 misses the global minimum and leads into the plain regions to the left

output was too large. This rule can be justified by a least square optimization approach.

5.3.4 Multilayer Perceptrons: Backpropagation

In multilayer perceptrons the response of an output neuron is determined not only by its own weights but also by the weights 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 then perform a gradient descent in the resulting error landscape. With some calculation, it is straightforward to derive the error minimization algorithm known as backpropagation (see Fig. 5.9).[12] 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. 5.25, and the according weight update $\Delta w_{ij} = \eta \delta_i h_j$ is applied. In addition, hidden layer corrections ε_j can be obtained from the output layer corrections. With the notations shown in Fig. 5.9, we have:

$$\varepsilon_j = \sum_i \delta_i w_{ij}, \tag{5.26}$$

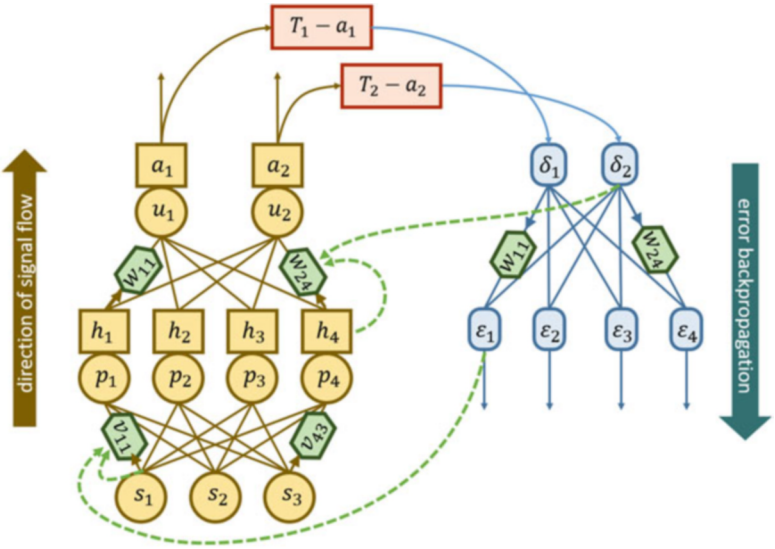
from which the hidden layer weights are updated as $\Delta v_{jk} = \eta \varepsilon_j s_k$. Equation 5.26 is called "backpropagation" because the summation is over the first index of the weights, as if the correction signals would run backward from the output units to the hidden units and were summed there according to the hidden-to-output weights (see blue arrows in Fig. 5.9). This process can be extended to networks with more than one hidden layer, by simply applying Eq. 5.26 with the convention that if δ denotes the correction signals in layer n, and ε denotes the correction signals in layer $n-1$. As the number of layers increases ("deep neural networks"), convergence of the minimization procedure becomes numerically unstable and needs to be supported by additional mechanisms.

Backpropagation is thus an elegant mathematical algorithm for the gradient calculation in the multilayer perceptrons, not an actual neurobiological process.

5.3.5 Deep Neural Networks

Deep neural networks are modified multilayer perceptrons with large numbers of layers. The modifications are motivated by neurobiological results to various extents. We briefly mention two ideas relevant for neuroscience, convolutional networks, and relational nonlinearities. For an overview of deep learning research in general, see LeCun et al. (2015); Schmidhuber (2015), and Goodfellow et al. (2016).

[12] An early reference to backpropagation is Rumelhart et al. (1986); for a historical discussion of backpropagation and deep learning, see Schmidhuber (2015).



		Activation dynamics (forward propagation)		Error backpropagation and weight dynamics	
Layer	3	activities (output)	$a_i = f(u_i)$	error	$\delta_i = (T_i - a_i)$
		potentials	$u_i = \Sigma_j w_{ij} h_j$	weight adjustment	$\Delta w_{ij} = \lambda\, \delta_i h_j$
	2	activities (hidden units)	$h_j = f(p_j)$	error	$\varepsilon_j = \Sigma_i w_{ij} \delta_i$
		potentials	$p_j = \Sigma_k v_{jk} s_k$	weight adjustment	$\Delta v_{jk} = \lambda\, \varepsilon_j s_k$
	1	activities (input)	s_k	–	

Fig. 5.9 Backpropagation. Brown and yellow colors: activation dynamics (signal flow). Red: supervision (comparison of output and teacher signal). Blue: backpropagation of error signal. Green: weights and weight dynamics. An input $(s_1, s_2, s_3)^\top$ is passed through a three-layer perceptron with the standard activation dynamics. p, h: potential and activity of hidden units and u, a: potential and activity of output units. The output $(a_i, a_2)^\top$ is compared to a teacher signal $(T_1, T_2)^\top$. The resulting correction signal $(\delta_1, \delta_2)^\top$ is then used to adjust the third layer weights according to the δ-rule. An example is given for weight w_{24} which would be adjusted by $\Delta w_{24} = \eta \delta_2 h_4$; $\eta \in \mathbb{R}^+$ is the learning rate. The backpropagation step is the calculation of the second layer correction signals ε_j by a weighted sum of the δ_i. These can be used to adjust the second layer weights, again by the δ-rule. An example is given for v_{11}. For further explanation see text

One modification is the use of "convolutional" layers in which a subsequent layer looks onto a previous one by shift-invariant receptive fields, much like in the lateral inhibition network introduced in Chap. 2. Within such layers, the neurons are organized in a grid with spatial coordinates. Let a_{ij} and b_{kl} denote two neurons in subsequent layers at grid locations (i, j) and (k, l), respectively. We assume that both layers use the same grid dimensions; they would then be called convolutional if the weight between neurons a and b does not depend on all four indices i, j, k, l, but

only on the differences $i-k$ and $j-l$. This is also called "weight-sharing," because the synaptic weights at the same relative position within each neuron's receptive field will be the same. The backpropagation algorithm can be modified to enforce the equality of all weights with the same differences between their indices (Waibel et al. 1989; LeCun et al. 1998). Convolutional layers are usually used at the front end of the neural network. In vision, they develop receptive fields similar to the Gabor functions discussed in Chap. 3 (see, e.g., Krizhevsky et al. 2017).

The second modification is the use of relational nonlinearities (see Eq. 2.40) particularly in combination with downsampling between layers with different numbers of neurons. For example, if the grid size is reduced from one layer to the next by a factor of 2 in each direction, the most active neuron will be determined in each 2×2 subarray of the preceding layer, and only the activity of this neuron is passed on to the subsequent layer. If the preceding layer is a convolutional array of edge detectors for a given orientation, say, the subsequent layer will not only keep the specificity for edge orientation but will also show some positional invariance. This mechanism is reminiscent of the relation of simple and complex cells in the visual cortex and was used to model pattern recognition in the visual pathway; see, for example, Fukushima (1980) and Riesenhuber and Poggio (1999).

Deep neural networks are now a standard tool for artificial intelligence and machine learning. In computational neuroscience, they are mostly used for the analysis of large bodies of experimental data; their merits in the modeling of actual brain processes are discussed in the next section.

► **Key Point: Multilayer Perceptrons** A learning rule for multilayer perceptrons based on least square optimization of the weights is called backpropagation. In networks with large numbers of layers (deep neural networks), the number of adjustable synapses is reduced by weight-sharing in convolutional layers.

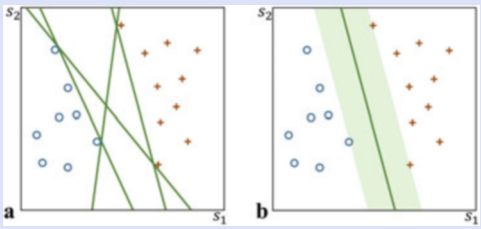
Box 5.5 Other Types of Classifiers

In the pattern recognition literature, many other approaches to the problem of pattern recognition have been developed, most of them based on the notion of the feature space. We briefly discuss support vector machines and Bayesian classification.

Support Vector Machines address a problem arising in the generalization from fixed training sets to novel data. Figure **a.** shows a data set with two classes (red plus signs and blue circles) together with various possible decision boundaries.

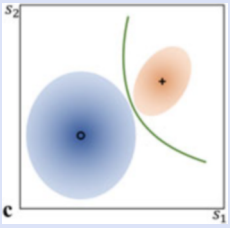
(continued)

Box 5.5 (continued)



The perceptron may find any one of these. When applied to a novel data set, the extremal lines shown here might be suboptimal because novel pattern may scatter and fall on the wrong side of the line. A systematic approach shown in figure **b.** is to search for a decision boundary separating the classes with the widest possible margin, such that generalization errors are less likely. It is determined by 3 (in general $n + 1$) so-called support vectors, i.e., data points touching the green margin, hence the name "support vector machine" (SVM). The SVM is a linear classifier; for linearly non-separable problems the original data are embedded in a higher dimensional space, similar to the approach realized by the hidden units of a perceptron. For the XOR-problem (Fig. 5.6), data vectors (s_1, s_2) would be replaced by $\mathbf{h} = (s_1, s_2, s_1 s_2)^\top$, where the function $f(s_1, s_2) = s_1 s_2$ is called a "kernel." The decision boundary in $\mathbb{R}^3$ is then given by the hyperplane $h_1 + h_2 - 2h_3 = -1$ which corresponds to the hyperbolic decision boundary $s_2 = (1 + s_1)/(2s_1 - 1)$ in the original feature space.

Bayes Classifiers are built on the idea that incoming patterns are noisy versions of one of the two standards or prototypes $\mathbf{p}_1$ and $\mathbf{p}_2$, marked by plus and circle in figure **c.** If the noise distribution is known, we can calculate the conditional probabilities of seeing a certain exemplar, given it is a version of one prototype or the other. The prototype under which the observed data is more likely is called the maximum-likelihood estimate of the specimen's class.



The probability distributions $P((s_1, s_2)|\mathbf{p}_1)$ and $P((s_1, s_2)|\mathbf{p}_2)$ are shown as ellipses in the figure. They are often assumed to be Gaussians with mean

(continued)

Box 5.5 (continued)
$\mathbf{p}_i$, whose covariance matrices have to be estimated in a learning phase. If the distributions of both prototypes are circular with equal variance (radius), the decision boundary is a straight line or hyperplane, namely the perpendicular bisector of the prototype loci. In general the decision boundaries are quadrics, in $\mathbb{R}^2$ parabolas, hyperbolas, or ellipses.

5.4 The Perceptron and the Brain

Neurons involved in the recognition of objects and complex visual patterns have been described in the so-called ventral stream of the visual pathway leading from the areas of the occipital lobe (areas V1, V2, and V3) to area V4 and into the temporal lobe, most notably the inferior temporal (IT) sulcus (Mishkin et al. 1983; Kravitz et al. 2011). Following the ideas of Hubel and Wiesel discussed in Chap. 3, the intricate specificities of IT neurons are thought to result from a series of processing steps starting with edge detection by the simple cells in primary visual cortex. The next steps would be the generation of some positional invariance by the V1 complex cells and the "hypercomplex" cells found in areas V2 and V3 (Hubel and Wiesel 1965). These latter cells are also called "end-stopped," because they respond best to oriented bars that end inside their receptive field or continue in a different direction (corners or T-junctions). Further steps are thought to be the analysis of motion, optic flow, and binocular disparity; for review, see Orban (2008) and Rolls (2023). With the discovery of more and more visual areas (Felleman and Van Essen 1991), this leads to the idea that the areas of the visual pathway together with their anatomical lamination constitute a multilayer perceptron or deep learning network as described in the previous section; see, for example, Fukushima (1980) and Serre et al. (2007). In this section, we will discuss three neurobiological questions related to this idea.

5.4.1 Feedback and Feedforward

Modeling the visual pathways as a feed-forward multilayer perceptron is a serious simplification of the actual neuroanatomy. Cortical areas are composed of layers or "laminae" of neurons, but these are extended volumes each containing in itself complex circuits or networks (see, e.g., Douglas and Martin 2004; Shepherd and Grillner 2017). Unlike the layers of a perceptron, within which the neurons are assumed to be unconnected, cortical laminae thus have an internal connectivity which includes rich feedback. Indeed, feedback in cortical processing occurs on all levels, within each layer, between the layers of an area, and between cortical areas (Felleman and Van Essen 1991).

Still, the multilayer perceptron might be a valuable model of visual information processing under time limits. Reaction times in visual object recognition tasks are

often below or in the order of a second. In one experiment, Thorpe et al. (1996) had subjects look at a large number of natural images of landscapes and outdoor scenes and asked them to hit a button whenever they spotted an animal. The images were shown for just 20 milliseconds at a high pace and subjects reached 94% correct judgments with an average reaction time of 445 ms. Even more surprisingly, event-related potentials recorded from the scalp revealed a difference between pictures with and without animals already after 150 ms. If we assume that the signal takes about 30 ms to pass the retina and 10–15 ms for the passage of each further synapse, this implies that in rapid classification tasks, there is simply not enough time for feedback mechanisms to operate. Rapid classification would thus rely on the feed-forward connectivity only, for which case the multilayer perceptron is an appropriate model.

Based on this argument, Serre et al. (2007) used a multilayer perceptron as a model of rapid object recognition in humans and report good agreement for classification performance and its variation over various image types. However, even with the substantial progress recently made in deep neural networks, important differences between the performances of human observers and artificial neural networks remain (Wichmann and Geirhos 2023). One problem seems to be the recognition of objects from unusual viewpoints, which is a problem for human observers, but much more so for the neural networks tested in the study. Another puzzling problem is the occurrence of so-called adversarial patterns that are grossly misclassified by the neural network although they differ from correctly classified images only by minimal noise-like variations invisible to human observers (Szegedy et al. 2014). This means that the decision boundary runs between the correctly classified image and its adversary and may indeed from narrow "pockets" which have no effects in the training set but lead to errors if a test pattern falls inside. The problem might be related to the lack of top-down information in the feed-forward network, but at the time of this writing, this idea is speculative.

In any case, multilayer perceptrons and deep neural networks do provide promising models of human pattern recognition, mostly because there are no alternatives that perform nearly as good. Still the abundance of feedback coupling in the visual cortex indicates that the mechanisms are not completely understood.

5.4.2 Hierarchy and Processing Steps

Is there further evidence for step-by-step processing beyond the initial steps (simple, complex, and hypercomplex) named by Hubel and Wiesel and, if so, what exactly are these steps? The answer to this question is harder than one might think, because pinning down the detailed specificities of a given neuron is not always easy. Indeed, specificities are usually partial and overlapping such that the stimulus properties represented in a cortical area become visible only when looking at larger populations or ensembles of neurons, a problem that we will return to in Sect. 7.2. Here we will discuss some instructive examples.

Cortical area V4 is placed intermediate between the occipital areas V1, V2, and V3 and the IT cortex. Pasupathy and Connor (2002) analyzed V4 neurons with curved contours and irregular three-dimensional shapes presented at various positions and orientations. By systematically varying the shapes and orientations, they mapped the neurons' responses and found localized tuning profiles on the resulting shape-map. Neurons also show some degree of positional invariance. This behavior has been reproduced in a feed-forward neural network model by Cadieu et al. (2007). Shape recognition may thus be considered a further step in the processing hierarchy for object recognition.

In a mature vision system, i.e., after a sufficient amount of learning, the initial processing stages can be expected to be fixed and optimized for the average requirements of the daily visual tasks. In newborn kittens, however, it has been shown that even the early stages of visual processing must be learned. Kittens are born with their eyes closed and open their lids only after a few days. It is therefore easy to control visual experience from birth. Blakemore and Cooper (1970) raised kittens in visually deprived environments showing either vertically or horizontally oriented contrast edges but no other orientations. When recording from visual cortex, they found that the animals raised with vertical stripes only showed mostly neurons tuned to vertical orientations and no neurons tuned to horizontal ones. Vice versa, animals raised in horizontally striped surround developed neurons for this orientation, but not for vertical ones. Learning of oriented edge detectors is also found in deep learning where the initial convolutional layers develop Gabor-like receptive fields (Krizhevsky et al. 2017). It is, however, generally modeled with unsupervised learning schemes, in particular competitive learning. We will come back to this in the next chapter.

The formation of detectors for task-relevant features can also occur during training for a novel task. Sigala and Logothetis (2002) trained monkeys with line drawings of faces varying in four possible dimensions, eye height, eye separation, nose length, and mouth height. The monkeys received a reward if they correctly classified a novel face in one of the two categories and learned to do this task with high precision. In the experiments, only two of the feature dimensions were necessary to solve the task, while the others contained no information. The authors show that after training, a population of neurons in the IT cortex represents these relevant feature dimensions but not the irrelevant ones. This is exactly what one would expect from a perceptron trained for the same task. Of course, other classification schemes would generally make the same prediction.

As a final example for the steps in the processing hierarchy, we consider the problem of view invariance in object recognition. Three-dimensional objects and faces may look very different when looked at from different sides, and both humans and monkeys are able to recognize these views as the same, at least if the difference is not too large. Logothetis et al. (1994) investigated view invariance with unfamiliar three-dimensional objects in which long-term memory of object shapes will play only a minor role. They suggest a feed-forward architecture for view invariance in which an earlier layer recognizes specific object views, while view invariance is achieved by combining units tuned to different views of the same object into a

common output neuron. In the IT cortex of monkeys trained for view-independent recognition tasks, view-dependent units have indeed been found (Logothetis and Sheinberg 1996; Tanaka 1996); they may represent one of the later steps in the object recognition hierarchy.

5.4.3 The Role of Single Neurons

The processing stream of a feed-forward network ends in one unit or in a number of unconnected units, each of which represents a certain category or class. This leads to the question of how specific the classes should be. For example, in face recognition, is it enough to assume units selective for the face of a particular person? Or do we need separate classes for the same face seen from different sides, the face with different facial expression, the person wearing sunglasses or a hat, and so on? If we want to describe every possible version of a given face by its own class, the number of classes will get very large. The idea that neurons represent classes of higher and higher specificity is generally discussed as the "grandmother cell" approach (see Gross 2002). In its extreme version, it leads to a "combinatorial explosion" of neurons needed to represent huge numbers of classes and their subdivisions and is therefore unlikely or even logically impossible. Just how far the specialization of classes may go, however, is a matter of research.

Neurons selective for face pictures showing particular persons have been reported from monkeys (Perrett et al. 1987) and humans (Quiroga et al. 2005). This is remarkable and shows that the grandmother cell approach and with it feed-forward processing may go a long way in visual object recognition. At the same time, it should be noted that the recorded neurons also show a certain degree of invariance. For example, in the Quiroga et al. (2005) study, one neuron responded to all seven pictures showing the actress Jennifer Aniston (but no further people) with varying dresses and poses, while responses to 80 other pictures were weak or absent. In subsequent studies, neurons were also found which responded to pictures of specific persons and to their names, be they presented in writing (visual stimulus) or as spoken text (auditory stimulus), see Quiroga et al. (2009).

One problem of the grandmother cell approach is that if neurons for very special classes exist, they would be activated only in very rare cases. The brain would then need to maintain huge numbers of neurons which are almost never used. A systematic approach to the problem of the specificity of neural coding has been suggested by Barlow (1972). In his "neuron doctrine for perceptual psychology," Barlow states that the *"sensory system is organized to achieve as complete a representation of the sensory stimulus as possible with the minimum number of active neurons."* This implies that each neuron should be specialized to a portion of the perceptual space of about equal relevance whose presence is signaled by the neuronal firing. It is an economic principle that guarantees an equal share of processing load over all neurons and adapts the coarseness of representation to the information processing tasks. In artificial neural networks, it has been picked up by the theory of sparse coding (Olshausen and Field 1996) which will be discussed in Sect. 6.4.

- **Feed-Forward Models of Visual Cortex** Multilayer perceptrons are used as models of object and face recognition in the visual and inferior temporal cortex. The "grandmother cell" debate addresses the level of specialization reached by feed-forward processing.

5.5 Summary and Further Reading

1. In artificial neural networks, neurons are modeled as threshold gates in which a weighted sum of inputs (the "potential") must exceed a threshold in order to generate an output. This is formalized by the activation function.
2. The state variables of the system are the activities of the neurons (activity vector) and the matrix of synaptic weights. They are updated at synchronous time steps.
3. Learning is modeled as weight adjustment as specified by a learning rule. Learning rules depend on the pre- and postsynaptic activities (Hebbian learning) plus a variety of additional parameters such as a teacher signal (supervised learning), an overall payoff signal (reinforcement learning), or the activity of other neurons (competitive learning).
4. A basic scheme for classification or pattern recognition is the perceptron. The two-layer perceptron projects each input vector on its weight vector (dot product) and fires if the projection length exceeds the threshold. It thus divides the feature space along a linear decision boundary into two class regions.
5. The perceptron learning rule implements iterative minimization of classification error. It is a supervised learning scheme since the error is determined using a teacher signal. The learning rule can be derived from the minimization of classification error by gradient descent.
6. In multilayer perceptrons, the iterative minimization approach leads to a learning scheme known as backpropagation.
7. Deep neural networks are multilayer perceptrons with convolutional layers and relational nonlinearities. They are the basis of modern artificial intelligence.
8. In the inferior temporal sulcus and other parts of the temporal lobe, neurons specifically responding to pictures of particular persons or objects are found. Some of their properties can be modeled by feed-forward multilayered networks.

Texts

Goodfellow et al. (2016): *Comprehensive text on deep learning and its foundations in linear algebra.*

Minsky and Papert (1988): *Expanded edition of a classical text on perceptrons presenting rigorous proofs for many theorems in the style of discrete mathematics.*

Rolls (2023): *Overview of the various computations performed by the different parts of the brain. The chapter on the ventral visual system gives extensive neurobiological background to the topics discussed here.*

Shepherd and Grillner (2017): *Edited collection of papers describing the connectivity and neurophysiology of a wide variety of biological neural networks, including also visual cortex.*

Suggested Original Papers for Classroom Seminars

Haynes and Rees (2006): *Review of "mind reading" studies, i.e., the prediction of visual stimuli presented to a subject from brain scanning data obtained during stimulus presentation. The problem is treated as one of pattern recognition and can be used to illustrate the feature space approach.*

Serre et al. (2007): *Classifier model based on the multilayer perceptron. The layers are interpreted as areas of the visual and temporal cortices. Pattern classification abilities are tested with real images.*

Sigala and Logothetis (2002): *Experimental study showing that neurons in monkey IT cortex learn relevant features for a novel classification task.*

References

Adrian, E.D., and A. Forbes. 1922. The all-or-nothing response of sensory nerve fibers. *Journal of Physiology* 56: 301–330.

Barlow, H.B. 1972. Single units and sensation: A neuron doctrine for perceptual psychology? *Perception* 1: 371–394.

Blakemore, C., and G.F. Cooper. 1970. Development of the brain depends on the visual environment. *Nature* 228: 477–478.

Cadieu, C., M. Kouh, A. Pasupathy, C.E. Connor, M. Riesenhuber, and T. Poggio. 2007. A model of V4 shape selectivity and invariance. *Journal of Neurophysiology* 98: 1733–1750.

Cybenko, G. 1989. Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals, and Systems* 2: 303–314.

Douglas, R.J., and K.A.C. Martin. 2004. Neuronal circuits of the neocortex. *Annual Review of Neuroscience* 27: 419–451.

Eccles, J.C. 1964. *The Physiology of Synapses*. Berlin: Springer.

Felleman, D.J., and D.C. Van Essen. 1991. Distributed hierarchical processing in the primate visual cortex. *Cerebral Cortex* 1: 1–47.

Fukushima, K. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* 36: 193–202.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep Learning*. Cambridge: The MIT Press.

Gross, C.G. 2002. Genealogy of the "grandmother cell". *The Neuroscientist* 8: 512–518.

Haynes, J.-D., and G. Rees. 2006. Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience* 7: 523–534.

Hebb, D.O. 1949. *The Organization of Behaviour*. New York: Wiley.

Helmstaedter, M., K.L. Briggman, S.C. Turaga, V. Jain, H.S. Seung, and W. Denk. 2013. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature* 500: 168–174.

Hubel, D.H., and T.N. Wiesel. 1965. Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology* 28: 229–289.
Kandel, E.R., Y. Dudai, and M.R. Mayford. 2014. The molecular and systems biology of memory. *Cell* 157: 163–186.
Kempermann, G. 2019. Environmental enrichment, new neurons, and the neurobiology of individuality. *Nature Reviews Neuroscience* 20: 236–245.
Kravitz, D.J., K.S. Saleem, C. Baker, and M. Mishkin. 2011. A new neural framework for visuospatial processing. *Nature Reviews Neuroscience* 12: 217–230.
Krizhevsky, A., I. Sutskever, and G.E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* 60: 84–90.
LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86: 2278–2324.
LeCun, Y., Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature* 521: 436–444.
Logothetis, N.K., and D.L. Sheinberg. 1996. Visual object recognition. *Annual Review of Neuroscience* 19: 577–621.
Logothetis, N.K., J. Pauls, H.H. Bülthoff, and T. Poggio. 1994. View-dependent object recognition by monkeys. *Current Biology* 4: 401–414.
McCulloch, W.S., and W. Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5: 115–133.
Minatohara, K., M. Akiyoshi, and H. Okuno. 2016. Role of immediate-early genes in synaptic plasticity and neuronal ensembles underlying the memory trace. *Frontiers in Molecular Neuroscience* 8: 78.
Minsky, M.L., and S.A. Papert. 1988. *Perceptrons, Expanded Edition*. Cambridge: The MIT Press.
Mishkin, M., L.G. Ungerleider, and K.A. Macko. 1983. Object vision and spatial vision: Two cortical pathways. *Trends in Neurosciences* 6: 414–417.
Olshausen, B., and D. Field. 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381: 607–609.
Orban, G.A. 2008. Higher order visual processing in macaque extrastriate cortex. *Physiological Reviews* 88: 59–89.
Pasupathy, A., and C.E. Connor. 2002. Population coding of shape in area V4. *Nature Neuroscience* 5: 1332–1338.
Perrett, D.I., A.I. Mistlin, and A.J. Chitty. 1987. Visual neurones responsive to faces. *Trends in Neurosciences* 10: 358–364.
Quiroga, R.Q., L. Reddy, G. Kreiman, C. Koch, and I. Fried. 2005. Invariant visual representation by single neurons in the human brain. *Nature* 435: 1102–1107.
Quiroga, R.Q., A. Kraskov, C. Koch, and I. Fried. 2009. Explicit encoding of multimodal percepts by single neurons in the human brain. *Current Biology* 19: 1308–1313.
Ramón y Cajal, S. (1899–1904). *Textura del sistema nervioso del hombre y de los vertebrados*. Madrid: Nicolás Moya.
Riesenhuber, M., and T. Poggio. 1999. Hierarchical models of object recognition in cortex. *Nature Neuroscience* 2: 1019–1025.
Rolls, E.T. 2023. *Brain Computations and Connectivity*. Oxford: Oxford University Press.
Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65: 386–408.
Rumelhart, D.E., G.E. Hinton, and R.J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323: 533–536.
Schmidhuber, J. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61: 85–117.
Schultz, W., P. Dayan, and R.R. Montague. 1997. A neural substrate of prediction and reward. *Science* 275: 1593–1599.
Serre, T., A. Oliva, and T. Poggio. 2007. A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences* 104: 6424–6429.

Serre, T., L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. 2007. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29: 411–426.

Shatz, C. 1992. The developing brain. *Scientific American* 267: 61–67.

Shepherd, G.M., and S. Grillner. 2017. *Handbook of Brain Microcircuits*. 2nd ed. Oxford: Oxford University Press.

Sigala, N., and N.K. Logothetis. 2002. Visual categorization shapes feature selectivity in the primate temporal cortex. *Nature* 415: 318–320.

Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. 2014. Intriguing properties of neural networks. arXiv:1312.6199.

Tanaka, K. 1996. Inferotemporal cortex and object vision. *Annual Review of Neuroscience* 19: 109–139.

Thorpe, S., D. Fize, and C. Marlot. 1996. Speed of processing in the human visual system. *Nature* 381: 520–522.

Waibel, A., T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang. 1989. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37: 328–339.

Wichmann, F.A., and R. Geirhos. 2023. Are deep neural networks adequate behavioral models of human visual perception? *Annual Review of Vision Science* 9: 501–524.

Widrow, B., and M.E. Hoff. 1960. Adaptive switching circuits. In *IRE WESCON Convention Record*, 96–104.

Artificial Neural Networks with Interacting Output Units

6

Abstract

The simple perceptron has just one output unit whose response is considered the truth value of a binary predicate. Even if many output units act in parallel on a common body of input and hidden layers, they are distinguished only by their specific teacher signals but are otherwise independent of each other. In this chapter, we turn to models in which the activity and learning of one output neuron directly affects the activity and learning of all other outputs. The simplest case is the matrix model of associative learning which stores input–output associations in a distributed or "holographic" way; it is based on covariance matrices and the outer product learning rule. A biologically more realistic form of learning is competitive learning in which the eigenvectors of the covariance matrix of the input data set play an important role. It allows the self-organization of efficient data representations including decorrelation and the formation of continuous feature maps. Sparse coding is a related scheme that optimizes representations not by redundancy reduction but by the identification of meaningful sub-patterns. Finally, we discuss the formation of continuous field attractors or activity "bumps" moving on continuous feature maps. Such attractors are used as models of working memory.

Learning Objectives

- Types of computation carried out by neural networks
- Concepts from multivariate statistics relating to neural networks: covariance matrix, regression, principal components, and independent component analysis

(continued)

H. A. Mallot, *Computational Neuroscience*,
https://doi.org/10.1007/978-3-031-75705-1_6

- Covariance matrices and the formation of associative memories by the outer product learning rule
- Competitive learning, the eigenvectors of the input covariance matrix, and the self-organization of neural representations
- Sparse coding and the identification of meaningful features or components in data sets
- Continuous-field attractors and reverberating activity in working memory models

6.1 Tasks of Neural Information Processing

Neural information processing involves a number of partially overlapping tasks and mechanisms, for which models have been developed. Before we move on to the discussion of further network types, we therefore give an overview of some of the computations carried out by the brain and their relations to the various network architectures.

Sensory filters extract relevant features from the sensory input. They are generally described by receptive field functions and tuning curves (Chaps. 2, 3, and 7). Examples from the visual system include edge detection, visual motion, stereo disparity, etc. In the auditory system, neurons have been described which are tuned to complex spatiotemporal sound events such as phonemes in human speech or interaural time differences used in directional hearing. Sensory filters may also integrate information from multiple sensory modalities such as the visual and auditory appearance of an object, for example, the look and barking of a dog. Neurons responding only weakly to the individual modalities but strongly to combined stimuli have been described in the superior colliculus (Stein and Stanford 2008). This behavior is called "superadditivity" and may be realized as coincidence detection on input from multiple modality-specific sensory filters.

Classification as described in Chap. 5 can be considered a special and elaborated case of sensory filtering. As in the example of the bug detector in frogs, completed classification will act as a trigger for behavior or further processing steps. Classification goes beyond filtering in that it puts a specimen into a known class of other specimen which can be dealt with according to common rules or procedures. Occasionally, such classes are also called categories, but this must not be confused with the term "categorization" as defined below.

Categorization is the discovery of meaningful classes in large bodies of data. For example, by looking at writings in an unknown language, one might be able to identify letters or words as repeating patterns whose variation is noise. This is possible even without a teacher signal and can be modeled in neural network theory as competitive learning or sparse coding. When applied to sensorimotor

or other spatiotemporal events, categorization relates to the discovery of rules which might then be used to predict the consequences of actions.

Association is the transformation of an input vector to an output vector. For example, when reaching for a visual target, the sensory input has to be transformed into a vector of muscle commands or joint angles. This can also be treated as a separate classification task performed by each motor neuron, but in this case the proper coordination of the individual muscles or motor neurons needs to be prescribed by a teacher signal. In contrast, looking at motor coordination as a vector transformation allows for more efficient theories. In memory, association allows for content addressability and the distribution of memory traces over the entire network. The theory of associative memory has been developed in neural network theory and is now widely applied in applications such as search engines in the Internet or in brain–machine interfaces.

Control links sensory input to some motor output suitable to maintain a desired steady state. This may involve simple input–output associations but also goal-dependent top-down processes. For example, in coordinated locomotion, central pattern generators in the spinal cord provide spatiotemporal pattern for muscle activation leading to smooth gaits and gait transitions. Learning and adjustment of control loops may be achieved as reinforcement learning.

Representation of task-relevant information in working memory is needed for the supervision and execution of longer-lasting plans and maneuvers. Working-memory contents are often modeled as stable patterns of activity ("attractors") maintained by reverberation in a net with strong feedback connectivity. An example is the representation of an animal's heading direction with respect to some world-centered reference frame in the hippocampal and thalamic head-direction systems.

In this chapter, we will discuss a number of different and largely independent network approaches developed to deal with some of the above problems.

6.2 Associative Memory

The processing and representation of information in the nervous system are "distributed" processes in the sense that many neurons participate and take an overall equal share of processing load. Contents or pieces of information can therefore not be pinned down to individual neurons as their exclusive carriers. This is maybe best realized in associative networks (Hebb,[1] 1949) whose mathematical treatment as "matrix memories" was pioneered by Steinbuch (1961), Willshaw et al. (1969), and Kohonen[2] (1972).

[1] See Footnote 8 in Chap. 5.

[2] Teuvo Kohonen (1934–2021), Finnish computer scientist.

Associative networks compute mappings between input and output vectors. This is a common task in the brain, which takes sensory inputs and produces motor outputs in the action–perception loop. A simple example from sensori-motor control is the vestibulo-ocular reflex, in which input data obtained from the semicircular canals are associated with (i.e., transformed into) motor commands for the neck and eye muscles stabilizing gaze. In another example involving also memory, an input vector coding a retinal image 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 the presented image. Biologically plausible models of these performances include elaborate circuitry and many intermediate processing steps; for the vestibulo-ocular reflex, see, for example, Ito (2006). We do not discuss these models in detail but focus on a general principle: that is, the associative properties of simple two-layer networks, which go a long way to solve association problems, at least if the layers are large. The described mechanisms are also used in medical engineering and brain–machine interfaces. For example, the mapping of multielectrode recordings from the motor cortex to the "intended" angles of the arm joints have been learned in associative networks and used to control the movement of arm prostheses with neural commands. One important question for neural network theory is to find mechanisms by which such association rules can be learned.

Mappings between input and output vectors are also calculated by banks of perceptrons with common input as shown in Figs. 5.7 and 5.9. In this section, however, we focus on the role of correlation and covariance matrices in association. Nonlinearities are generally assumed weak and may be entirely neglected.

6.2.1 The Feedforward Associator

Consider a network composed of two subsets of units (layers), called input and output layers (Fig. 6.1). The activity vectors are denoted by $\mathbf{s} = (s_1, \ldots, s_J)^\top$ for the input layer and $\mathbf{a} = (a_1, \ldots, a_I)^\top$ 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 this convention differs from our previous definition (Eq. 5.7): c_{ii} is not the coupling of a unit with itself, but rather the coupling of two separate units—one in the input layer and one in the output layer—which happen to have the same index number. For a linear activation transfer function (i.e., without nonlinearity), we have:

$$a_i = \sum_{j=1}^{J} c_{ij} s_j \quad \text{or} \quad \mathbf{a} = \mathbf{C}\mathbf{s}. \tag{6.1}$$

If (in agreement with Eq. 5.7) we denote by $\mathbf{W}$ the $(I + J) \times (I + J)$ weight matrix of the entire set of neurons with activity vector $(s_1, s_2, \ldots, s_J; a_1, a_2, \ldots a_I)^\top$, the relation between the weight matrix of the combined set and the $I \times J$ connectivity

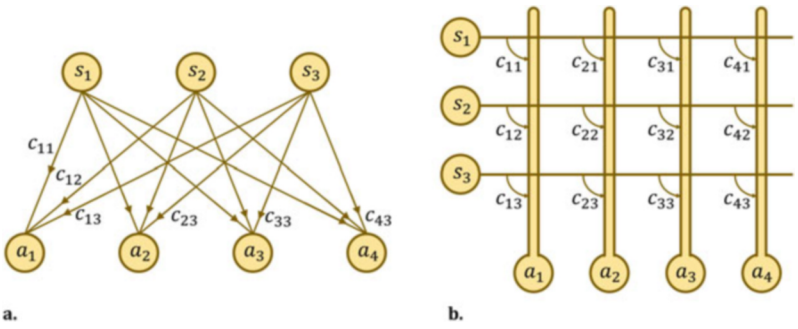


Fig. 6.1 Two-layered network with 3 input units ("sensors" $s_1, \ldots, s_3$) and 4 output units $a_1, \ldots, a_4$. The topology is complete feedforward, that is: each input unit connects to all output units. Weights from s_j to a_i are labeled c_{ij}. (**a**) Layered plot as used also for multilayer perceptrons in Chap. 5. (**b**) Matrix plot with "axons" emanating from the s_j units, "dendrites" leading to the a_i units, and connections c_{ij} at each crossing point. Note that the two representations are completely equivalent

matrix of the feedforward connections, $\mathbf{C}$, is expressed as:

$$\mathbf{W} = \left(\begin{array}{c|c} \mathbf{0}_{JJ} & \mathbf{0}_{JI} \\ \hline \mathbf{C} & \mathbf{0}_{II} \end{array}\right), \tag{6.2}$$

where $\mathbf{0}_{KL}$ is a $K \times L$-matrix with all zero coefficients. If all c_{ij} are different from zero, $\mathbf{W}$ describes a complete feedforward connectivity. Note also that this topology is identical to the two-layer perceptron with multiple outputs (Figs. 5.7, 6.1a) although the graphical representation shown in Fig. 6.1b is more commonly used in the context of association.

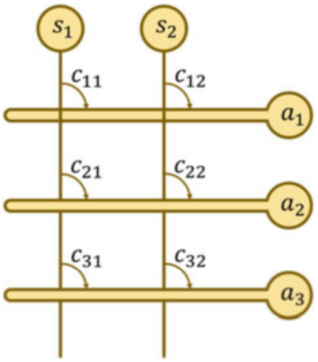
Example: A 2 × 3 Associator

Consider the associator shown in Fig. 6.2 and assume that we want to implement in this associator the input–output pair $(\mathbf{s}^1, \mathbf{a}^1)$ with $\mathbf{s}^1 = (1, 0)^\top$ and $\mathbf{a}^1 = (1, 0, 1)^\top$. As before, the upper index marks a presentation made at a certain time step or simply a presentation number. It is easy to see that this association is implemented by the weight matrix

$$\mathbf{C}^1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}; \tag{6.3}$$

we can demonstrate this by simply calculating the output vector as

$$\mathbf{a}^1 = \mathbf{C}^1\mathbf{s}^1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}. \tag{6.4}$$



$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{pmatrix}$$

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$$

Fig. 6.2 A 2×3 feedforward associator oriented to show the structure of the connectivity matrix C

The weight matrix $\mathbf{C}^1$ was found by setting to the value of 1 all weights for which both the presynaptic signal s_j and the desired postysnaptic signal a_i are present. In our example, where all activities are either 0 or 1, this is equivalent to setting

$$c_{ij} = a_i \, s_j . \tag{6.5}$$

In matrix notation, we obtain

$$\mathbf{C} := \begin{pmatrix} c_{11}, c_{12} \\ c_{21}, c_{22} \\ c_{31}, c_{22} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} (s_1, s_2) = \mathbf{a}\, \mathbf{s}^\top . \tag{6.6}$$

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*[3] of the two vectors. Equation 6.5 and its matrix form 6.6 are therefore called the outer product rule. It states that associations can be stored by calculating the outer product of output and input vectors and using it as a connectivity matrix. We will give a formally correct account of this idea below.

The learning rule 6.5 looks like a version of the Hebb rule in which the pre- and postsynaptic activities with respect to the connection c_{ij} are combined. It is, however, not incremental, but describes a kind of "one-shot" learning where the weight is set and fixed after just one presentation of stimulus and desired output.

[3] The outer product of two vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ is defined as $\mathbf{x}\mathbf{y}^\top$. It exists for vectors of arbitrary dimensions n, m and results in a $n \times m$ matrix with coefficients $x_i y_j$. Recall for comparison that the dot product discussed in Sect. 5.1.3, $\mathbf{x}^\top \mathbf{y} = (\mathbf{x} \cdot \mathbf{y})$, is defined only for vectors of equal dimension and evaluates to a number. In contrast to the outer product, the dot product is also called the *inner product*.

Also, a_i is not the actual activity of the network but the *desired* response, which has to be presented to the network much like a teacher signal in supervised learning.

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., ($\mathbf{s}^2 = (0, 1)^\top$, $\mathbf{a}^2 = (0, 1, 1)^\top$). From the outer product rule, we obtain

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \mathbf{C}^2 = \mathbf{a}^2 \mathbf{s}^{2\top} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} (0, 1) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}. \tag{6.7}$$

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

$$\mathbf{C} = \mathbf{C}^1 + \mathbf{C}^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}. \tag{6.8}$$

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

6.2.2 The Outer Product Rule

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

$$\mathbf{a} = \mathbf{C}\mathbf{s}. \tag{6.9}$$

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 vectors:

$$\mathbf{C} = \mathbf{a}\mathbf{s}^\top = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_I \end{pmatrix} \cdot (s_1, s_2, \ldots, s_J) = \begin{pmatrix} a_1 s_1 & a_1 s_2 & \ldots & a_1 s_J \\ a_2 s_1 & a_2 s_2 & \ldots & a_2 s_J \\ \vdots & \vdots & & \vdots \\ a_I s_1 & a_I s_2 & \ldots & a_I s_J \end{pmatrix}. \tag{6.10}$$

As the response of our network, we obtain

$$\mathbf{C}\mathbf{s} = \left(\mathbf{a}\mathbf{s}^\top\right)\mathbf{s} = \mathbf{a}\left(\mathbf{s}^\top\mathbf{s}\right) = \mathbf{a}\,\|\mathbf{s}\|^2, \tag{6.11}$$

by virtue of the associative law of matrix multiplication. This is the desired output, up to a factor $\|\mathbf{s}\|^2$. We can incorporate this factor in the weight matrix by choosing the coefficients $c_{ij} = a_i s_j \|\mathbf{s}\|^{-2}$. Alternatively, it can be dealt with by requiring $\|\mathbf{a}\| = 1$ and adding a normalizing nonlinearity at the output of the associator.

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, such that $\mathbf{s}^p$, $p = 1, \ldots, P$ denotes the p-th input vector; the letter p is mnemonic for presentation. For each presentation (or association pair), we calculate the connectivity matrix by the outer product rule. As in the example, we now simply add up these outer products and obtain

$$\mathbf{C} = \sum_{q=1}^{P} \mathbf{a}^q \mathbf{s}^{q\top} = \begin{pmatrix} \sum_q a_1^q s_1^q, & \sum_q a_1^q s_2^q, & \ldots, & \sum_q a_1^q s_J^q \\ \sum_q a_2^q s_1^q, & \sum_q a_2^q s_2^q, & \ldots, & \sum_q a_2^q s_J^q \\ \vdots & \vdots & & \vdots \\ \sum_q a_I^q s_1^q, & \sum_q a_I^q s_2^q, & \ldots, & \sum_q a_I^q s_J^q \end{pmatrix}. \tag{6.12}$$

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 and the normalization with the sample size (cf. Eq. 5.16). Note that in Eq. 5.16 the covariance was calculated over the components of a stimulus vector, whereas here, the sum runs over the presentations. Mathematically, both definitions are possible, but the interpretation is not the same.

$\mathbf{C}$ in Eq. 6.12 is a "mixed" covariance matrix of $\mathbf{a}$ and $\mathbf{s}$. Later, we will also encounter the (standard) covariance matrix of the input set, $(1/P) \sum_P \mathbf{s}^p \mathbf{s}^{p\top}$.

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

$$\mathbf{C}\mathbf{s}^p = \left(\sum_q \mathbf{a}^q \mathbf{s}^{q\top} \right) \mathbf{s}^p = \sum_q \mathbf{a}^q \left(\mathbf{s}^{q\top} \mathbf{s}^p \right). \tag{6.13}$$

The term $\mathbf{s}^{q\top} \mathbf{s}^p$ 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. If we have more association pairs than input lines ($P > J$), the input patterns are no longer orthogonal and mixed outputs will appear.

The mathematical issue at this point is the linear independence of the input vectors. A set of k vectors $\mathbf{x}^k \neq 0$ is said to be linear independent if neither of them can be expressed as a linear combination of the others: that is, if $\sum_k a_k \mathbf{x}^k = 0$,

it implies $a_k = 0$ for all k. The number of linear independent vectors in a set is limited by the dimension of the containing vector space. In three-dimensional space, three linear independent vectors span a parallelepiped with nonzero volume while linear dependent vectors would be coplanar (contained in a common plane) or even colinear. Pairwise orthogonal vectors are also linear independent.

6.2.3 General Least Square Solution

If the number of association pairs exceeds the number of input lines, or if the input vectors are not linear independent, 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

$$\mathbf{S} := [\mathbf{s}^1; \mathbf{s}^2; \ldots; \mathbf{s}^P] = \begin{pmatrix} s_1^1, & s_1^2, & \ldots, & s_1^P \\ s_2^1, & s_2^2, & \ldots, & s_2^P \\ \vdots & \vdots & & \vdots \\ s_J^1, & s_J^2, & \ldots, & s_J^P \end{pmatrix} \tag{6.14}$$

and

$$\mathbf{A} := [\mathbf{a}^1; \mathbf{a}^2; \ldots; \mathbf{a}^P] = \begin{pmatrix} a_1^1, & a_1^2, & \ldots, & a_1^P \\ a_2^1, & a_2^2, & \ldots, & a_2^P \\ \vdots & \vdots & & \vdots \\ a_I^1, & a_I^2, & \ldots, & a_I^P \end{pmatrix}. \tag{6.15}$$

These are simply obtained by writing the column vectors $\mathbf{s}^p$ and $\mathbf{a}^p$ one after the other. We may now write Eq. 6.9 in matrix form, expressing the desired associations for all p jointly in the expression

$$\mathbf{A} = \mathbf{CS}, \tag{6.16}$$

where $\mathbf{A}$ is a $I \times P$ matrix, $\mathbf{C}$ is $I \times J$, and $\mathbf{S}$ is $J \times P$.

If there are only J association pairs and input vectors, and if these vectors are pairwise orthogonal and of unit length, the matrix $\mathbf{S}$ will be square and orthonormal: that is, we have $\mathbf{S}^{-1} = \mathbf{S}^\top$ (see Box 5.2). We therefore obtain $\mathbf{C}$ simply by multiplying Eq. 6.16 with $\mathbf{S}^\top$ from the right, yielding $\mathbf{C} = \mathbf{AS}^\top$ as in the example of Sect. 6.2.1. If $\mathbf{S}$ is not orthonormal but still invertable, which is the case if the $\mathbf{s}^p$ are linearly independent, we may write $\mathbf{C} = \mathbf{AS}^{-1}$. In the general case, however, if $\mathbf{S}$ is not invertable, which is always the case if $P \neq J$, we can find the least square

approximation by multiplying Eq. 6.16 with $\mathbf{S}^\top$ from the right. The $J \times J$ matrix $\mathbf{S}\mathbf{S}^\top$ will be invertable as long as the vectors $\mathbf{s}^1, \ldots, \mathbf{s}^P$ span the full J-dimensional input space. Since this will usually be the case, especially if $P \gg J$, we obtain

$$\mathbf{C}_{\text{opt}} = \mathbf{A}\mathbf{S}^\top(\mathbf{S}\mathbf{S}^\top)^{-1} = \mathbf{A}\mathbf{S}^+. \tag{6.17}$$

This matrix minimizes the squared error between the desired outputs $\mathbf{A}$ and the actual outputs $\mathbf{C}_{\text{opt}}\mathbf{S}$. The matrix $\mathbf{S}^+ = \mathbf{S}^\top(\mathbf{S}\mathbf{S}^\top)^{-1}$ is called the *Moore–Penrose pseudoinverse* of $\mathbf{S}$. Pseudoinverses also occur in regression analysis, where they are used for the solution of general linear problems; see also Box 6.1.

Box 6.1 Associative Memory and Regression

In regression analysis, a *dependent variable* (regressand) y is modeled as a function of an *independent variable* (regressor) $\mathbf{x}$ and a set of *coefficients* $\boldsymbol{\beta}$, to which a noise term ϵ is added:

$$y = f(\mathbf{x}; \boldsymbol{\beta}) + \boldsymbol{\epsilon}$$

We first assume that y is a scalar, while $\mathbf{x}$ and $\boldsymbol{\beta}$ are vectors with dimension n. If f is linear, it can be written as the dot product of $\mathbf{x}$ and $\boldsymbol{\beta}$:

$$y = \mathbf{x}^\top\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

For P measurements with independent variables $\mathbf{x}^p$ and outcomes y^p for $p = 1, \ldots, P$, the resulting system of equations can be expressed in matrix notation as

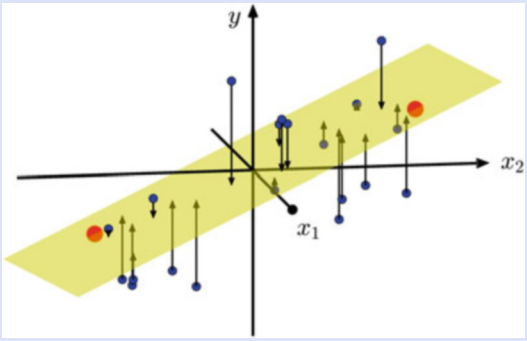
$$\begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^p \end{pmatrix} = \begin{pmatrix} x_1^1, & x_2^1, & \ldots, & x_n^1 \\ x_1^2, & x_2^2, & \ldots, & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^P, & x_2^P, & \ldots, & x_n^P \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix} + \boldsymbol{\epsilon} \quad \text{or} \quad \mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

In the multivariate case, we will have multiple dependent variables y_j, $j = 1, \ldots, m$, each with their own set of coefficients β_{ij}. The model equation then becomes

$$\begin{pmatrix} y_1^1, & y_2^1, & \ldots, & y_m^1 \\ y_1^2, & y_2^2, & \ldots, & y_m^2 \\ \vdots & \vdots & & \vdots \\ y_1^P, & y_2^P, & \ldots, & y_m^P \end{pmatrix} = \begin{pmatrix} x_1^1, & x_2^1, & \ldots, & x_n^1 \\ x_1^2, & x_2^2, & \ldots, & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^P, & x_2^P, & \ldots, & x_n^P \end{pmatrix} \begin{pmatrix} \beta_{11}, & \beta_{12}, & \ldots, & \beta_{1m} \\ \beta_{21}, & \beta_{22}, & \ldots, & \beta_{2m} \\ \vdots & \vdots & & \vdots \\ \beta_{n1}, & \beta_{n2}, & \ldots, & \beta_{nm} \end{pmatrix} + \boldsymbol{\epsilon}$$

(continued)

Box 6.1 (continued)
We write this equation as $\mathbf{Y} = \mathbf{XB} + \boldsymbol{\epsilon}$. Except for the explicit mentioning of the error term $\boldsymbol{\epsilon}$, this equation is identical to Eq. 6.16 for an $n \times m$ associator with $\mathbf{A} = \mathbf{Y}^\top$, $\mathbf{S} = \mathbf{X}^\top$, and $\mathbf{B} = \mathbf{C}^\top$. To see this, remember the relation $(\mathbf{CS})^\top = \mathbf{S}^\top \mathbf{C}^\top$. In regression analysis, $\mathbf{X}$ would be called the design matrix and the least square solution for $\mathbf{B}$ would be calculated with the pseudoinverse as in Eq. 6.17: $\mathbf{B}_{\text{opt}} = (\mathbf{X}^+)^\top \mathbf{Y}$.



The figure shows an example with $n = 2$ and $m = 1$. The graph of the linear function f is a plane passing through the origin (shown in yellow); it is completely determined by $n = 2$ input–output values shown as large red dots. These could be the association pairs stored in an associative memory.
In regression, many input–output pairs are considered (small blue dots). The yellow plane then becomes the least square estimator of the functional relationship between $\mathbf{x}$ and y. The length of the arrows projecting each data point to the plane is $\boldsymbol{\epsilon}_p$.

6.2.4 Applications

The basic concept of associative networks is closely related to linear regression; see Box 6.1. It is simple but powerful, in particular if large numbers of neurons and high dimensional feature spaces are considered. We briefly discuss three applications.

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 by a reference number or address, but by some sort of meaning. For example, one could code written names (letter strings) of n characters into n groups of 26 input neurons (one for each letter of the alphabet, $J = 26n$) and code $k \times l$-pixel images into the activities of $I = kl$ output neurons. In a training session,

each image would be associated with a written name and the network learns the covariances between the activities of the input and output neurons. If a string is presented at the input, the associated image will thus appear at the output layer.

Associative memory is also *distributed* in the sense that deleting a number of internal connections will not delete specific memory items, since the output depends on all stored covariances. Therefore, the result of partial deletion of the connectivity matrix $\mathbf{C}$ will be an unspecific deterioration or blurring of the output image, not the loss of individual images or memory items. By the same token, misspellings of the input string will not lead to erroneous outputs as long as the input strings are sufficiently different. Memories with these properties are also called "holographic."

Both properties, content addressability and distributedness, are of course highly plausible for models of biological memories. They are also realized in search engines and Internet browsers which also make extensive use of covariance matrices.

The capacity of associative memories is limited by the constraint that the input vectors must be linearly independent. Therefore, in a network with J input lines, just J association pairs can be stored without loss. This limitation does not necessarily devalue associative memory. Especially if the number of input lines J is large, as is clearly the case in the brain, capacity limitations may be outweighed by the obvious advantages of associative memory: content addressability and distributedness.

Autoassociation and Attractor Neural Networks

The simple associator of Fig. 6.2 can be turned into a network with complete between-layer connectivity by feeding the output back to the "dendrites" in the upper part of the figure (recurrent network). 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 "autoassociators" have been used to improve degraded patterns or to reconstruct images from fragments (Kohonen et al. 1977). In the nonlinear 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 numbers of neurons in the motor and parietal cortices of a monkey (see, for example, Wessberg et al. 2000). Together with these recordings, movements of the monkey's arm are also recorded and encoded in terms of the angles and angular velocities occurring at the shoulder and elbow joints or the hand trajectory. The "association" between the neural activities during some time window and the pose of the arm can be learned along the lines discussed above; indeed, this amounts to a linear regression analysis of the arm movement with neural activity as a regressor.

Once the transformation matrix is known, new neural activities can be used to predict or "decode" 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). Advanced versions of this approach have enabled quadriplegic patients to control the movements of a robotic arm by their cognitive efforts, for example, for bringing a drinking bottle to their mouth; for an overview, see Lebedev and Nicolelis (2017). Brain–machine interfaces make use of all available methods of data analysis, including artificial neural networks but also statistics, control theory, and others. They provide valuable and powerful models of the operations performed by real neural networks in the brain, which have to solve the same problems.

▶ **Key Point: Associative Memory** Pairs of input and output vectors (association pairs) can be stored in matrix memories using the outer product learning rule. The mathematics is related to regression analysis and the matrix pseudoinverse.

Box 6.2 Covariance Matrices

Consider a multivariate random vector $\mathbf{x}$ of length n for which a sample of P examplars $\mathbf{x}^1, \ldots, \mathbf{x}^P$ is available. We can calculate the sample means and variances of each component as well as the sample covariances of each pair of components as

$$\bar{x}_i = \frac{1}{P}\sum_{p=1}^{P} x_i^p, \qquad \mathrm{var}(x_i) = \tfrac{1}{P-1}\sum_{p=1}^{P}(x_i^p - \bar{x}_i)^2, \quad \text{and}$$

$$\mathrm{cov}(x_i, x_j) = \frac{1}{P-1}\sum_{p=1}^{P}(x_i^p - \bar{x}_i)(x_j^p - \bar{x}_j).$$

The division by $P-1$ rather than by P makes these quantities unbiased estimators of the true (co)variances of the statistical population. Clearly, we have $\mathrm{var}(x_i) = \mathrm{cov}(x_i, x_i)$. The covariance matrix collects the covariances of all pairs of the vector components, with the variances on the diagonal. As in Eqs. 6.14 and 6.15, we write the sample of column vectors as a data matrix

$$\mathbf{X} = [\mathbf{x}^1, \ldots, \mathbf{x}^P].$$

With $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_n)^\top$, the covariance matrix of the data set $\mathbf{X}$ is then given by

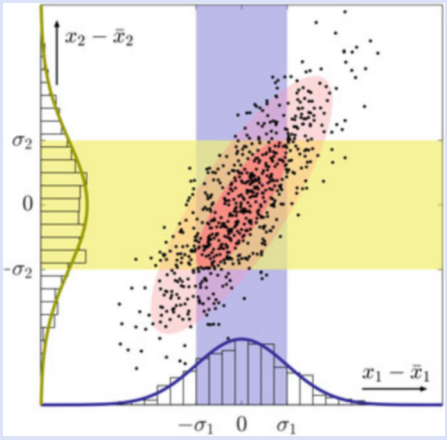
$$\boldsymbol{\Sigma} = \begin{pmatrix} \mathrm{var}(x_1), & \mathrm{cov}(x_1, x_2), & \ldots, & \mathrm{cov}(x_1, x_n) \\ \mathrm{cov}(x_2, x_1), & \mathrm{var}(x_2), & \ldots, & \mathrm{cov}(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}(x_n, x_1), & \mathrm{cov}(x_n, x_2), & \ldots, & \mathrm{var}(x_n) \end{pmatrix} = \frac{1}{P-1}\mathbf{X}\mathbf{X}^\top - \frac{P}{P-1}\bar{\mathbf{x}}\bar{\mathbf{x}}^\top.$$

(continued)

Box 6.2 (continued)
The covariance matrix is square ($n \times n$) and symmetric. It is an important element of the probability density function of the n-dimensional normal distribution

$$p(\mathbf{x}) = \frac{\exp\left\{-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right\}}{\sqrt{(2\pi)^n \det\boldsymbol{\Sigma}}}$$

where $\det\boldsymbol{\Sigma}$ denotes the determinant of $\boldsymbol{\Sigma}$. The figure shows the case $n = 2$. Each dot is a vector $\mathbf{x}^p$. The "marginal" distributions for x_1 and x_2 are shown as histograms and density curves, respectively, with $\sigma_i = \sqrt{\mathrm{var}(x_i)}$. The blue and yellow



bands mark the standard deviations of the marginal distributions. The inner red ellipse (error ellipse) contains all points with $(\mathbf{x}-\bar{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) \leq 1$, on average 39% of the sample. Its area is $\pi \det\boldsymbol{\Sigma}$. The outer red ellipse has doubled semiaxes and contains 86% of the data points.

6.3 Self-Organization and Competitive Learning

6.3.1 Exponential Weight Growth in Simple Hebbian Learning

Hebbian learning as introduced in Fig. 5.3a involves a product of the pre- and postsynaptic activities s_j and a_i. A simple formulation is

$$w_{ij}^{t+1} = w_{ij}^t + \eta\, a_i^{t+1} s_j^t \quad \text{or} \quad \mathbf{W}^{t+1} = \mathbf{W}^t + \eta \mathbf{a}^{t+1}\, \mathbf{s}^{t\top} \tag{6.18}$$

where $\eta \geq 0$ is 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.

If we present the same stimulus repeatedly, the Hebb rule will lead to an ever stronger response since the synapses triggered by the stimulus will grow with every stimulus presentation. In order to avoid this exponential weight growth, it is necessary to foresee some type of normalization, or competition, as will be discussed below. Before we turn to normalizing Hebb rules, however, we take a closer look at the exponential growth immanent in Hebbian learning, which reveals the role of the eigenvectors of the input covariance matrix. For this, we also omit the static nonlinearity; the activation function then reduces to

$$\mathbf{a}^{(t+1)} = \mathbf{W}^{(t)}\mathbf{s}^{(t)}. \tag{6.19}$$

Here we have written upper indices with brackets, to distinguish them from powers which will also become relevant in this section; see Footnote 11 in Chap. 5. Combining this with Eq. 6.18, we obtain

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \eta\mathbf{W}^{(t)}\mathbf{s}^{(t)}\mathbf{s}^{(t)\top}. \tag{6.20}$$

On average, the presentation of individual stimuli $\mathbf{s}^{(t)}$ can be replaced by the application of the uncentered covariance matrix

$$\mathbf{\Sigma} = \frac{1}{t_{\max}}\sum_{t=1}^{t_{\max}}\mathbf{s}^{(t)}\mathbf{s}^{(t)\top} \tag{6.21}$$

(see Box 6.2).[4] We then obtain

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)}(\mathbf{I}+\eta\mathbf{\Sigma}) = \mathbf{W}^{(1)}(\mathbf{I}+\eta\mathbf{\Sigma})^t \tag{6.22}$$

where $\mathbf{I}$ is the identity matrix (see Box 5.2) and the unbracketed t on the right side is an exponent, not a time index.

Over time, the weight matrix $\mathbf{W}^{(t)}$ in Eq. 6.22 will usually diverge since the eigenvectors of the covariance matrix $\mathbf{\Sigma}$ are nonnegative.[5] It is interesting, however, to analyze the path along which divergence proceeds. To this end, we consider an eigenvector $\mathbf{v}_i$ of $\mathbf{\Sigma}$ with eigenvalue λ_i. Since every vector is eigenvector of the identity matrix with eigenvalue 1, it is easy to see that $\mathbf{v}_i$ will also be an eigenvector

[4] Covariance matrices are often denoted by the capital Greek letter sigma ($\mathbf{\Sigma}$) to stress the analogy with the variance (σ^2) of one-dimensional variables. $\mathbf{\Sigma}$ must not be confused with the sum sign $\sum$.

[5] Covariance matrices like $\mathbf{\Sigma}$ are symmetric and "positive semi-definite," meaning that $\mathbf{x}^\top\mathbf{\Sigma}\mathbf{x} \geq 0$ for all vectors $\mathbf{x} \in \mathbb{R}^n$. This implies that all eigenvectors are real and satisfy $\lambda_i \geq 0$.

of $\mathbf{I} + \eta\mathbf{\Sigma}$, now with the eigenvalue $1 + \eta\lambda_i$. The powers $(\mathbf{I} + \eta\mathbf{\Sigma})^t$ have the same eigenvectors $\mathbf{v}_i$ with eigenvalues $(1 + \eta\lambda_i)^t$.

Let i^* denote the index of the largest eigenvalue[6] of $\mathbf{\Sigma}$; since $\eta \geq 0$, $1 + \eta\lambda_{i^*}$ will also be the largest eigenvalue of $\mathbf{I} + \eta\mathbf{\Sigma}$. In repeated multiplication with $\mathbf{I} + \eta\mathbf{\Sigma}$, this eigenvalue will therefore grow faster than all other ones which can be neglected in the long run. If we renormalize the weight matrix after each learning round by dividing it through $(1 + \eta\lambda_{i^*})$, we may write

$$\lim_{t\to\infty} \left(\frac{\mathbf{I} + \eta\mathbf{\Sigma}}{1 + \eta\lambda_{i^*}} \right)^t = \mathbf{v}_{i^*}\mathbf{v}_{i^*}^\top. \tag{6.23}$$

Equation 6.23 implies that an input vector $\mathbf{s}$ presented to the system will always be projected on $\mathbf{v}_{i^*}$, the eigenvector of $\mathbf{\Sigma}$ with the largest eigenvalue. In multivariate statistics, this vector is also known as the first principal component of the data set $\mathbf{S}$, while the projection is called the score (see Box 6.3). Equation 6.23 thus establishes a relation between Hebbian learning (with some normalization) and principal component analysis: Competitive learning enables neural networks to discover statistically efficient data representation by self-organization. This will be made more explicit in the next section.

The effect sketched in this section is also used in an algorithm for the computation of the eigenvectors of large matrices, known as power iteration. Let $\mathbf{M}$ be a diagonalizable matrix, $\mathbf{v}_{i^*}$ its eigenvector with largest eigenvalue, and $\mathbf{b}_0$ any vector with $(\mathbf{b}_0 \cdot \mathbf{v}_{i^*}) \neq 0$. Then the iteration $\mathbf{b}_{k+1} = \mathbf{M}\mathbf{b}_k / \|\mathbf{M}\mathbf{b}_k\|$ will converge to $\pm\mathbf{v}_{i^*}$.

6.3.2 The Oja Learning Rule

The simple Hebb learning rule as formulated in Eq. 6.18 will lead to unlimited weight growth. In Eq. 6.23, we have considered a normalization scheme based on the eigenvalues of the input covariance matrix to demonstrate a relation between the input covariance matrix and the evolving weights. Now, we will discuss a more realistic scenario in which the eigenvalues of the covariance matrix need not be known beforehand. Rather, the weight vector of each neuron, $\mathbf{w}_i = (w_{i1}, \ldots, w_{iJ})^\top$, is normalized to unit length after each learning step. The result is mathematically more clear-cut: The weight vector itself will approach the first principal component of the data set. The scheme is called competitive learning since the growth of one synapse leads to an unspecific weight decay or "forgetting" in all

[6] For this, we need to assume that no other eigenvalue is equally large, which will usually be the case if the data set is large enough. Mathematically, the requirement is that the multiplicity of λ_{i^*} is 1. If two eigenvalues are equal, $\lambda_i = \lambda_j$, every linear combination of the corresponding eigenvectors, $a\mathbf{v}_i + b\mathbf{v}_j$ will also be an eigenvector with the same eigenvalue, yielding a two-dimensional "eigen-space."

other synapses terminating on the same neuron: that is, synapses "compete" for growth; see also Fig. 5.3b.

We consider a version of this idea that was suggested by Oja[7] (1982) for the input weights of just one neuron without nonlinearity. The "network" topology is that of a two-layer perceptron, and the weight matrix reduces to the weight vector of the sole output neuron. The situation is modeled by the activation function

$$a^t := \sum_{j=1}^{J} w_j s_j^t = \mathbf{w}^\top \mathbf{s}^t \tag{6.24}$$

and a set of input vectors presented as a temporal sequence $\mathbf{s}^t := (s_1^t, \ldots, s_J^t)^\top$. We introduce the normalizing Hebbian rule (also known as Oja's learning rule) as

$$w_j^{t+1} = \frac{w_j^t + \eta a^t s_j^t}{\sqrt{\sum_{j=1}^{J} \left(w_j^t + \eta a^t s_j^t\right)^2}}; \quad \mathbf{w}^{t+1} = \frac{\mathbf{w}^t + \eta a^t \mathbf{s}^t}{\|\mathbf{w}^t + \eta a^t \mathbf{s}^t\|}. \tag{6.25}$$

The denominator guarantees that $\|\mathbf{w}^t\| = 1$ at all times. The learning rate $\eta > 0$ is not constant but will be gradually reduced to zero ("annealed"), so that convergence is enforced.

The resulting development of the weight vector is illustrated in Fig. 6.3 for a neuron with just three input lines. Due to the normalization, the total length of the weight vector is constant; that is to say, all the weight vector can do during learning is moving with its tip on the surface of a sphere, or hypersphere, if more input lines are considered. The changes of the weight vector, $\Delta\mathbf{w} = \mathbf{w}^{t+1} - \mathbf{w}^t$, will therefore always be orthogonal to $\mathbf{w}^t$, i.e., tangent to the sphere in which $\mathbf{w}^t$ is a radius.[8]

We now use this property to reformulate the learning rule from Eq. 6.25, thereby dropping the time argument. Without the normalization, $\Delta\mathbf{w}$ consists only of the Hebbian term $a\mathbf{s}$, multiplied with the learning rate η. The normalization can be approximated by subtracting from $\eta a\mathbf{s}$ a multiple of $\mathbf{w}$ such that the total weight change will be orthogonal to $\mathbf{w}$. This condition is satisfied by the simpler learning rule

$$\Delta\mathbf{w} = \eta a[\mathbf{s} - a\mathbf{w}] = \eta[a\mathbf{s} - a^2\mathbf{w}] \tag{6.26}$$

as can be seen from the vanishing dot product of $\Delta\mathbf{w}$ and $\mathbf{w}$. Indeed, we have

$$\begin{aligned}(\Delta\mathbf{w} \cdot \mathbf{w}) &= \eta a[(\mathbf{s} \cdot \mathbf{w}) - a(\mathbf{w} \cdot \mathbf{w})] \\ &= \eta a[a - a] = 0.\end{aligned}$$

[7] Erkki Oja (born 1948), Finnish computer scientist.

[8] Strictly speaking, this is true only in an infinitesimal analysis with continuous time, where $\Delta\mathbf{w}$ is replaced by the derivative $d\mathbf{w}/dt$. For small learning rates η, this is no problem.

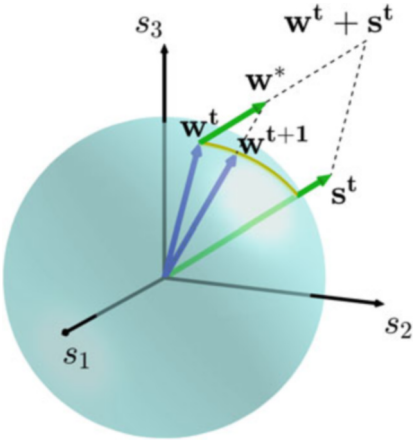


Fig. 6.3 Geometric interpretation of Oja's learning rule, Eq. 6.25. In a three-dimensional feature space, the blue sphere marks the possible loci of the weight vector, which will always have length 1. Let $\mathbf{w}^t$ denote the weight vector at time t. Assume now that a stimulus $\mathbf{s}^t$ is delivered. The new weight vector $\mathbf{w}^{t+1}$ is obtained by adding a certain multiple of the stimulus vector to the old weight vector (here $\mathbf{w}^* = \mathbf{w}^t + \eta\mathbf{s}^t$) and normalizing the sum. The tip of the weight vector will move on the surface of the sphere, along the yellow line, toward the stimulus vector. 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

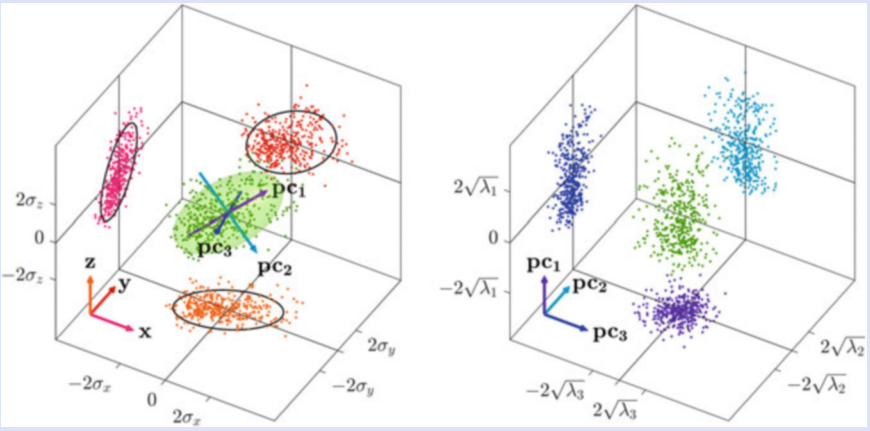
Equation 6.26 is an approximation of Eq. 6.25 that can be formally derived by a Taylor expansion of $\Delta\mathbf{w}$ about $\eta = 0$. Alternatively, $\Delta\mathbf{w}$ in Eq. 6.26 can be considered the result of a Gram–Schmidt orthogonalization of $\mathbf{w}$ and $\mathbf{s}$. For our discussion, it suffices to take it as an approximation of Eq. 6.25.

Box 6.3 Principal Component Analysis, PCA

A sample of data vectors $\mathbf{x}^p$, $p = 1, \ldots, P$ 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. This axis is called the first principal component (pc). The data set is then collapsed in this direction and the next pc is calculated as the direction of largest variation in the collapsed, $n - 1$-dimensional space, and so on.

(continued)

Box 6.3 (continued)



The left figure shows a three-dimensional cloud of data points (green dots) together with the two-dimensional scatter plots for the projections on the (x, y), (x, z), and (y, z) planes (reddish colors); the distribution is centered, $\bar{\mathbf{x}} = 0$. The green ellipsoid is defined by the points satisfying $\mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x} = 2$ where $\boldsymbol{\Sigma} = (\sum_p \mathbf{x}^p \mathbf{x}^{p\top})/P$ is the covariance matrix of the data set. The bluish vectors mark the axes of the ellipsoid. They are the eigenvectors of $\boldsymbol{\Sigma}$ and are called the principal components (pc) of the data set. The lengths of the half axes of the standard error-ellipsoid ($\mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x} = 1$) equal the square roots of the eigenvalues associated with each pc, $\sqrt{\lambda_i}$.

PCA creates a new coordinate system as shown on the right side of the figure. It is calculated via an eigenvalue decomposition of the covariance matrix $\boldsymbol{\Sigma}$,

$$\boldsymbol{\Sigma} = \mathbf{V}^\top \mathbf{D} \mathbf{V}$$

where $\mathbf{V} = (\mathrm{pc}_1, \mathrm{pc}_2, \mathrm{pc}_3)$ is the orthonormal matrix of eigenvectors of $\boldsymbol{\Sigma}$, and $\mathbf{D}$ is the diagonal matrix of the eigenvalues $\lambda_1, \ldots, \lambda_n$. For covariance matrices, it is guaranteed that the decomposition exists ("the matrix is diagonalizable") and that the λ_i are real and nonnegative.

The green data points are the same in both plots; on the right side, they have been rotated by the matrix $\mathbf{V}$ such that $\mathrm{pc}_1, \ldots, \mathrm{pc}_3$ are the new coordinate axes. The projections ("score-score-plots") show that all correlations have been removed, since the clouds align with the new coordinate axes. The projections of the data on pc1 (vertical axis in the right figure) has the largest variance of all possible projections. It therefore contains the largest possible amount of information in just one number. The principal components are

(continued)

Box 6.3 (continued)
vectors in feature space and can therefore themselves be considered patterns, weight vectors, or receptive fields.

Next, we substitute from Eq. 6.24 into Eq. 6.26, keeping in mind that $a = \mathbf{w}^\top \mathbf{s} = \mathbf{s}^\top \mathbf{w}$ due to the commutativity of the dot product. This yields

$$\Delta \mathbf{w} = \eta[\mathbf{s}\overbrace{\mathbf{s}^\top \mathbf{w}}^{a} - \overbrace{\mathbf{w}^\top \mathbf{s}}^{a}\overbrace{\mathbf{s}^\top \mathbf{w}}^{a}\mathbf{w}]. \tag{6.27}$$

Due to the associativity of matrix multiplication, we may bracket the outer products $\mathbf{s}\mathbf{s}^\top$. In the temporal average, i.e., for many stimulus presentations, these outer products become the covariance matrix of the training set, $\mathbf{\Sigma}$ (see Eq. 6.21). We obtain

$$\Delta \mathbf{w} = \eta(\mathbf{\Sigma}\mathbf{w} - (\mathbf{w}^\top \mathbf{\Sigma}\mathbf{w})\mathbf{w}). \tag{6.28}$$

If $\mathbf{w}$ equals a normalized eigenvector of $\mathbf{\Sigma}$, say $\mathbf{v}_i$, we have $\Delta \mathbf{w} = \eta(\lambda_i \mathbf{v}_i - \lambda_i \|\mathbf{v}_i\|^2 \mathbf{v}_i) = 0$. The normalized eigenvectors of $\mathbf{\Sigma}$ are therefore fixed points of weight evolution in the normalizing learning scheme. The convergence of the system is harder to show but basically follows the argument given in Sect. 6.3.1 for linear systems. As in Eq. 6.23 we denote by $\mathbf{v}_{i*}$ the eigenvector of $\mathbf{\Sigma}$ with the largest eigenvalue. Then, if the initial weight vector $\mathbf{w}^{(1)}$ contains a nonzero component of $\mathbf{v}_{i*}$, i.e., if $\mathbf{v}_{i*}^\top \mathbf{w}^{(1)} \neq 0$, this component will outgrow the components of all other eigenvectors because of its larger eigenvalue. The weight vector $\mathbf{w}^{(t)}$ will therefore converge to $\mathbf{v}_{i*}$ or $-\mathbf{v}_{i*}$ with the sign equaling the sign of $\mathbf{v}_{i*}^\top \mathbf{w}^{(1)}$.[9] For a detailed proof of the convergence, see Oja (1982).

This result means that by the normalizing Hebb rule, a perceptron will automatically adjust its weight vector to the eigenvector of the data covariance matrix $\mathbf{\Sigma}$ with the largest eigenvalue. If the data set has zero mean, this equals the first principal component of the data set, i.e., the longest axis of the cloud of data points, see Box 6.3. This result was to be expected from our graphical consideration in Fig. 6.3 since the weight vector is attracted by the projections of all stimulus vectors on the unit sphere. Since the activation function of the perceptron performs a projection on the weight vector (Sect. 5.2.2), the Oja rule automatically finds the axis representing the largest possible variability of the data set in just one number.

If we add a threshold nonlinearity to the activation function (Eq. 6.24), learning only occurs if the dot product $\mathbf{w}^\top \mathbf{s}$ is positive. This means that the perceptron needs

[9] Note that the sign of an eigenvector can be chosen arbitrarily, since with $\mathbf{v}_i$ all vectors $a\mathbf{v}_i$ with $a \in \mathbb{R}\backslash\{0\}$ will be eigenvectors with the same eigenvalue. We normally require $\|\mathbf{v}_i\| = 1$, but the sign remains arbitrary.

to start with a weight vector not too distant from the stimulus distribution. 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.

Oja (1989) presented an extension of this approach to associative networks with multiple output neurons. In this case, all output units see the same input and application of the learning rule of Eq. 6.25 would therefore lead to identical weight vectors for all output neurons. If, however, a suitable element of inhibition between the output neurons is added, the weight vectors of k output neurons will approach the first k principal components of the data set. For image data, these are the standard image features such as oriented edges or blobs. We will not follow the suggestion by Oja (1989) but discuss an alternative way of keeping multiple weight vectors apart from each other: the Kohonen self-organizing feature map.

▶ **Key Point: Normalizing Hebb Rule** In Hebbian learning with weight vector normalization, the weight vector converges to an optimum. Neural activity, i.e., the projection of a data vector on the weight vector, will then keep as much information about the data as possible. If the data set is centered, the optimum is reached at the first principal component of the data set.

6.3.3 Self-Organizing Feature Map (Kohonen[10] Map)

Principal components do not resemble actual input patterns but rather their differences or the dimensions along which various inputs can be distinguished. If we think of neurons as detectors for meaningful patterns or events, principal components are useful to reduce the number of feature dimensions that we need to consider, but eventually, we would be interested in more "holistic" representations of the relevant pattern contained in the input set. One approach to this end is the self-organizing feature map (SOM) introduced by Kohonen (1982).

The basic network topology of the SOM is a feedforward associator as shown in Figs. 6.1 and 6.4a. Since all neurons of the output layer see the same inputs, some mechanism has to be added to prevent them from learning the same weight vectors. Kohonen (1982) introduces a second type of competition, this time not between the input weights of a given output neuron (Fig. 5.3b) but between the output activities (Fig. 5.3c). The idea is to use lateral inhibition (see Fig. 2.3) in the output layer to generate a "winner-take-all" dynamics, in which the neuron with the strongest activation dominates all other neurons and eventually remains the only active neuron in the output layer (Malsburg 1973; Kohonen 1982). Since weight change depends on the correlation of input and output activity, this "winner" neuron would then be the only one who's weights would be adjusted in a given time step. In the next time step, another neuron might become the winner and will then learn different weights. By varying the width of excitatory and inhibitory coupling in lateral inhibition, the

[10] See Footnote 2.

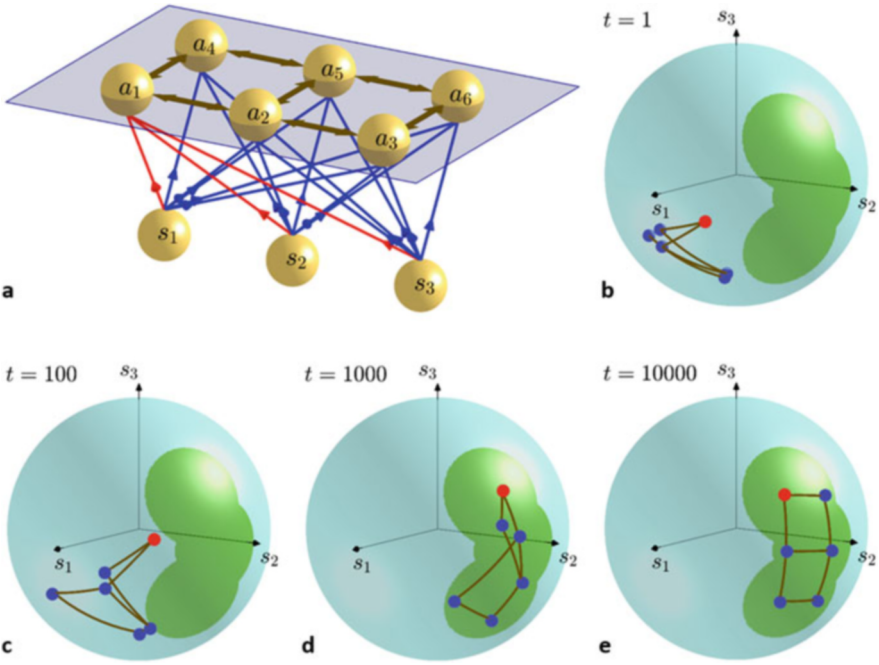


Fig. 6.4 Self-organizing feature map. (**a**) Network topology with input layer (s_j), input weights (blue and red arrows), map layer (a_i) and map layer adjacencies (dark brown arrows), in this example with the topology of a 3 × 2 grid. (**b**)–(**e**) Unit-hypersphere in the feature space. For each map unit a_i, its input weight vector $\mathbf{c}_i = (c_{i1}, c_{i2}, c_{i3})^\top$ is shown as a blue or red dot. The dark brown lines show the map layer adjacencies. The green area marks the region in which input vectors $\mathbf{s}^t$ occur with high probability. (**b**) The weights are initialized at random, leading to an unordered arrangement of weight vectors. (**c**)–(**e**) During learning, the weight vector grid "crawls" into the interesting area of feature space. It also "unfolds" such that adjacent units now have similar weight vectors (i.e., receptive fields). Weight vector $\mathbf{c}_1$ is shown in red so that its learning trajectory can be tracked

network might also generate local clusters of active neurons which would then tend to learn similar patterns.

Let us consider once again a linear hetero-associator (Fig. 6.4a), i.e., a feed-forward projection of an input layer with sensory units $\mathbf{s} = (s_1, s_2, \ldots, s_J)^\top$ to an output layer with activities $\mathbf{a} = (a_1, a_2, \ldots, a_I)^\top$. As suggested by Kohonen (1982), we do not model lateral inhibition and the winner-take-all dynamics explicitly but directly incorporate its most important features into the model: adjacency and eligibility for learning.

Adjacency is modeled by arranging the neurons in a grid as shown by the dark flat arrows in Fig. 6.4a. For each unit k, we can thus define a neighborhood $\mathcal{N}_k$ as the set of units whose distance to unit k is less than a certain threshold. In the example of Fig. 6.4a with distance threshold 2, we have $\mathcal{N}_1 = \{1, 2, 4\}$, $\mathcal{N}_2 = \{1, 2, 5, 6\}$, etc.

Eligibility for learning means that weight changes will be made only for those neurons which remain active after the lateral inhibition step. As in Sect. 6.2.1, the activation dynamics is given by an input–output connectivity matrix $\mathbf{C}$:

$$a_i = \sum_{j=1}^{J} c_{ij} s_j := \mathbf{c}_i^\top \mathbf{s}. \tag{6.29}$$

Here, $\mathbf{c}_i = (c_{i1}, c_{i2}, \ldots, c_{iJ})^\top$ is again the vector of all input weights of unit i, i.e., the receptive field of the unit. The Kohonen map then uses the following competitive weight dynamics: Let

$$k := \underset{1 \le i \le I}{\operatorname{argmax}} \{a_i\} \tag{6.30}$$

be the index number of the output unit generating the strongest activation after a given stimulus presentation. This unit will be the one whose weight vector most closely resembles the input vector in the sense of the dot product; it is usually called the "winner" unit. The winner neuron and the members of its neighborhood will then undergo a learning step according to the rule:

$$\text{for all } i \in \mathcal{N}_k: \quad \mathbf{c}_i^{t+1} := \frac{\mathbf{c}_i^t + \eta \mathbf{s}^t}{\|\mathbf{c}_i^t + \eta \mathbf{s}^t\|}. \tag{6.31}$$

This learning rule is identical to the Oja rule (Eq. 6.25) except for the factor a_i which is missing above. That is to say, a_i is assumed to take the value 1 for the winner neuron and its neighborhood while it is zero for the other output units. As in other cases discussed previously, the learning rate η may be made time dependent (decreasing to zero), to enforce convergence (annealing). In this scheme, competition occurs at two points: first, neurons compete to become the winner neuron, whose neighborhood will be eligible for learning, and second, synapses of the learning neurons compete for synaptic growth or reduction.

Figure 6.4b–e illustrates the learning process in a self-organizing feature map. The blue sphere represents the unit-hypersphere on which the weight vectors $\mathbf{c}_i$ move. The green area shows a region in feature space where stimuli are likely to occur (or rather the projection of this region on the unit sphere). The network starts with a random initialization (Fig.6.4b) where the weight vectors of each output unit are independent of the unit's respective position in the map; the map therefore looks scrambled. During learning, the weight vectors move into the green area because at each learning step, the winner and its neighbors are attracted toward the current stimulus vector which occurs somewhere within that range. 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 generated by the shared learning in the neighborhoods. As a result, map units will become "experts" for a certain stimulus type or feature. At the same time, the map

as a whole will reflect similarities between the features represented by each neuron. Input pattern with high frequency of occurrence will be represented by more units (larger grid regions) than rare patterns.

► **Key Point: Self-Organizing Feature Maps** Feature maps are layers of neurons in which the weight vector of each neuron (its receptive field) represents an important feature of the data set while neighboring neurons tend to respond to similar features. Such maps are abundant in the brain and constitute one of the major principles of neural computation. Kohonen maps are a model for feature-map formation.

6.3.4 Applications

Decorrelation

In statistics and technical data analysis, principal components are mostly used for redundancy reduction: The projections of the data vectors on the principal components (the "scores") are uncorrelated such that the scores of component i do not repeat information which is also present in the scores of the other components. In neural networks, where principal components are realized as weight vectors, the scores correspond to the resultant activity in the output neurons. If these activities are uncorrelated, redundancy is minimal and each spike conveys a maximal amount of information.

One interesting example is color vision, in particular the encoding of the three cone signals of the outer retina (S, M, and L for short, middle, and long wavelengths) into color opponent channels realized by the retinal ganglion cells. Since the absorption spectra of the three cone mechanisms are widely overlapping, their responses will be correlated, even if the spectra of the incoming lights would be not. Buchsbaum and Gottschalk (1983) used the known cone absorption spectra (or rather the Vos–Walraven primaries as their psychophysical counterparts) to calculate the covariance matrix of the cone activities for illumination with random spectra. From this, optimal decorrelated channels can be predicted by principal component analysis. It turns out that an achromatic channel (brightness) with positive contributions from all three cone mechanisms has the largest eigenvalue and therefore carries most of the information. This is probably why black-and-white photographs are often perceived to approximate their colored counterparts satisfactorily. The other two principal components are color opponent: that is, they combine cone signals with different signs. The stronger of these is $L - M$ or $M - L$ (red-green opponent), while the weaker is $S - M + L$, with a strong imbalance between the three contributions. The actual channels realized by retinal ganglion cells are achromatic, $\pm(L - M)$, and $S - (M + L)$, which is very close to the model prediction.

It might be argued that the wiring of the retinal cone mechanisms to the ganglion cells need not be learned but could also be genetically fixed. Even if this was true, the optimality of color coding demonstrated by the decorrelation approach

would be interesting. Adaptive wiring mechanism are, however, likely to play a role given the irregular and highly subject-dependent pattern of the retinal cone mosaic. Adaptive wiring of irregular cone mosaics has been modeled as the decorrelation of the activities of small retinal patches by Wachtler et al. (2007).

Feature Maps in the Brain

Processes of self-organization and competitive learning seem to underlie the formation and maintenance of map-like representations in the brain. The best studied case is probably the map of orientation selectivity in primary visual cortex as described by Hubel and Wiesel (1963) and Blasdel and Salama (1986). Within a patch of about 1mm^2 of visual cortex, neurons tuned to all edge orientations (see Sect. 3.2) are found in an orderly map-like arrangement such that nearby neurons tend to respond to equal or similar orientations. Along the depth axis (from the pia to the white matter), orientation tuning is preserved, such that neurons form iso-orientation columns. A patch containing neurons of all orientations and "ocularities" (sensitivity to stimuli delivered through the left or right eye) is also called a hypercolumn. Hypercolumns are local maps of orientation and ocularity nested or "intercalated" into the overall retinotopic map of the visual field, which, in humans, extends over some 15cm^2 of the occipital lobe. In the original model by Hubel and Wiesel (1977), sometimes called the "ice cube model," orientation columns are actually slices, or "slabs," stacked linearly one behind the other in a direction of the cortical surface orthogonal to the boundaries of ocular dominance. As an alternative, radial arrangements in "pinwheels" have been suggested by Braitenberg and Braitenberg (1979) and were later confirmed by the imaging studies of Blasdel and Salama (1986).

Malsburg (1973) modeled the self-organization of maps of orientation columns in a feedforward model as shown in Fig. 6.4a. In the map layer, a winner-take-all dynamic ensures that only one neuron is learning at every stimulus presentation. The input consists of images of oriented edges realized as black-and-white pixels. The output units develop preferences for particular edge orientations and adjacent units tend to have similar preferences. The model is also able to explain the deficits of cortical orientation tuning found in visually deprived kittens raised in environments with only vertical stripe patterns (Blakemore and Cooper 1970). For a review of models of the self-organization of feature maps in the brain, see Bednar (2012).

Adult Plasticity

Cortical representations change with use. In the tactile modality, the body surface is represented in the somatosensory cortex as an orderly "somatotopic" map with clear regions devoted to the limbs, individual fingers, the face, etc. This representation is sometimes referred to as the "somatosensory homunculus." After repeated stimulation of a skin patch, the representation of this patch will increase while the adjacent representations move somewhat away and shrink (Buonomano and Merzenich 1998). In the long run, the allocation of cortical resources is thus optimized to the behavioral needs. A particularly clear example of this reorganization is the increase in the size of the finger representation of the left hand in musicians

playing string instruments (Elbert et al. 1995), a process that continues over many years of training. In the opposite case, if a body part receives less stimulation, its representation will shrink. This is most obvious in the amputation of a limb in which case the representation of the amputated limb remains and may lead to so-called phantom perceptions (Ramachandran and Hirstein 1998). Since this representation is deprived of sensory input altogether, it will undergo a process of cortical reorganization in which the phantom representation shrinks and may eventually vanish while the neurons formerly dealing with the somatosensory input from the now amputated limb will develop new specificities. Such processes of cortical reorganization would also be expected from self-organizing maps, although the details of the underlying mechanisms may be more complex (see, for example, Turrigiano and Nelson 2004).

6.4 Sparse Coding

The winner-take-all dynamics of the Kohonen map ensures that in each stimulus presentation, only one neuron will be activated. After learning, the receptive field of each neuron therefore consists of the stimuli falling in a certain neighborhood of the weight vector which is limited by the neighborhoods of the adjacent weight vectors. The feature space (or rather the unit hypersphere $\|\mathbf{s}\| = 1$) is thus tesselated into Voronoi[11]-like cells, each representing the receptive field of one unit. Kohonen mapping is therefore also considered a technique for "vector quantization."

If the map contains many neurons, the resulting representation will be sparse, i.e., only one map neuron will be active at any one time. In addition, the receptive fields of the neurons will be small, meaning that each neuron will be highly specialized. This leads back to the problem of the grandmother cell discussed in Sect. 5.4. In the context of competitive learning, this problem is addressed by explicitly modeling the "sparsity" of a network: that is, the minimal number of active neurons needed to represent a stimulus. Sparse coding is thus an attempt to implement Barlow's (1972) "neuron doctrine for perceptual psychology."

Coding at an intermediate level of sparsity is useful if the set of input stimuli comprises a number of distinguishable pattern each of which occurs repeatedly with some variation, independent of the other patterns. The input signals will then be mixtures of these patterns, and an ideal representation should be able to undo the mixing and represent each mixture by its original components. This problem is also known as blind source separation; a common example would be to distinguish different simultaneous speakers in a room. A cue that can be used is that within each pattern presentation, the features received from the same source will be statistically dependent, while the features from different sources appear

[11] For a set of points $(\mathbf{p}_i)_{i=1,\ldots m} \in \mathbb{R}^n$, the Voronoi tesselation is given by the "cells" $R_i = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{p}_i - \mathbf{x}\| \leq \|\mathbf{p}_j - \mathbf{x}\| \text{ for all } j \neq i\}$. This definition is easily generalized to metric spaces such as the hypersphere.

with statistical independence (see also Box 6.4). Finding the source patterns is therefore also known as independent component analysis (ICA), see Hyvärinen and Oja (2000).

Olshausen and Field (1996) present an algorithm for generating representations that encode each stimulus with a minimal number of active neurons, each of which would specialize to one of the source patterns. The algorithm is formulated as an optimization problem with an objective function reflecting two requirements, the representation of the stimuli with high fidelity, and a small number of neurons active in each presentation. First, the error in the stimulus representation is formalized as

$$e_r = -\|\mathbf{s} - \mathbf{C}^\top \mathbf{a}\|^2 \tag{6.32}$$

That is to say: if the input weights of each unit are multiplied with the unit's activity and summed together, they should reproduce the input vector. This requirement is motivated by computational considerations, not by known biological mechanisms. Since synapses cannot be used backward, the term $\mathbf{C}^\top \mathbf{a}$ would imply that reciprocal weights $\mathbf{C}^\top$ from the map layer to the input layer exist and that the comparison of input and reconstruction is made in the input layer.[12]

Second, the sparsity requirement is formulated as a cost equaling the total activity in the map, for example:

$$e_s = -\sum_i |a_i| \tag{6.33}$$

The joint error term with scaling factor $\lambda \in \mathbb{R}^+$ is $e = e_r + \lambda e_s$; it is minimized in two steps. In the first optimization, the activity values for a given presentation p, $\mathbf{a}^p$, are optimized subject to the current connectivity matrix, which is assumed fixed at this stage. A single optimization step reads

$$\Delta \mathbf{a} = \mathbf{C}(\mathbf{s} - \mathbf{C}^\top \mathbf{a}) - \lambda \,\mathrm{sgn}(\mathbf{a}) \tag{6.34}$$

where the sign-function sgn comes in as the derivative of the absolute value function from Eq. 6.33; it is applied pointwise to each vector component. Activities are thus not calculated feedforward as in Eq. 5.4, but involve competition to achieve sparsity. In a neural network, this might be realized by a global inhibition in the output layer. The biological meaning of the term $\mathbf{C}\mathbf{C}^\top \mathbf{a}$ is less obvious; its implementation would require additional neural layers.

[12] If the error is reduced to zero, Eq. 6.32 can be rewritten as $\mathbf{s} = \mathbf{C}^\top \mathbf{a}$ or $\mathbf{a} = (\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}\mathbf{s}$. The matrix $(\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}$ is therefore the connectivity matrix actually learned by the network; it equals the transpose of the pseudoinverse of the matrix $\mathbf{C}$ from Eq. 6.32, $(\mathbf{C}^+)^\top$.

Once the optimal activities for the current connectivity matrix are found, the learning starts by modifying the weights such that the representation itself will be optimized:

$$\Delta \mathbf{C} = \frac{\eta}{P} \sum_{p=1}^{P} \mathbf{a}^p (\mathbf{s}^p - \mathbf{C}^\top \mathbf{a}^p)^\top \tag{6.35}$$

where η is a learning rate and $p = 1, \ldots, P$ numbers the stimulus presentations. The rule is reminiscent of the outer product rule $\mathbf{C} = \sum_p \mathbf{a}^p \mathbf{s}^{p\top}$ (Eq. 6.12) if we replace the stimulus vector by the difference of the stimulus and its reconstruction, $\mathbf{s} - \mathbf{C}^\top \mathbf{a}$.

Simulations by Olshausen and Field (1996) show that the algorithm is able to find the "sources" or independent components in data sets providing mixtures of these components. In the calculation, the components are the rows of the optimized connectivity matrix $\mathbf{C}$, i.e., the receptive fields associated with each output unit a_i. When trained with natural images, the receptive fields are oriented Gabor patches, whereas training with letter images and superpositions thereof would recover the original letters.

Figure 6.5 shows an example based on the MNIST data set of handwritten characters published by LeCun et al. (1998). Figure 6.5a shows an original image for each of the ten numerals from 0 to 9. A principal component analysis of a

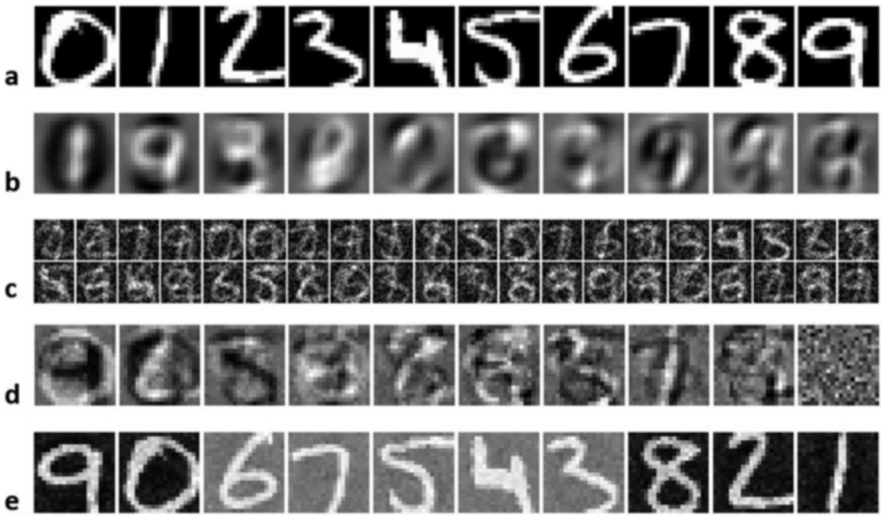


Fig. 6.5 (**a**) Sample images from the MNIST data set (LeCun et al. 1998). (**b**) First ten principal components computed from 1000 images from the data set. (**c**) Examples for mixtures generated by superimposing two of the sample images shown in a, plus noise. (**d**) First ten principal components of a set of 200 mixtures as shown in (**c**). (**e**) Receptive fields of units in a sparse coding network with ten output units, trained with the same 200 mixture images used also in (**d**)

large set of different instances of these numerals (Fig. 6.5b) shows no systematic resemblance with individual number images. This is not surprising since the principal components are orthogonal, and each higher component shows only the differences between the data set and the variability captured already by the previous components. Reconstructions will always require combinations of several of the principal components and are therefore not sparse.

In Fig. 6.5c, we consider mixtures generated by combining two of the ten original images appearing in part a of the figure together with a small amount of noise. Also in this case, the PCA does not recover the original images (Fig. 6.5d). The receptive fields of the sparse coding network shown in part e, however, closely resemble the source patterns. Sparsity is obtained since every mixture can be represented by strong activity in two of these units plus some diffuse activity representing noise. Unlike principal components, the receptive fields (weight vectors) are not orthogonal and do not have a well-defined ordering. Also, it is important to start the network with the appropriate number of units (one for each source), since otherwise sparsity cannot be achieved.

Box 6.4 Correlation and Independence

Two random variables x_1 and x_2 with probability densities $p(x_1)$ and $p(x_2)$ are statistically independent if the joint density equals the product of the two marginal densities: $p(x_1, x_2) = p(x_1)p(x_2)$.

Covariance is defined as $\mathrm{cov}(x_1, x_2) = E(x_1 - Ex_1)(x_2 - Ex_2)$ where E denotes the expected value. This definition is in agreement with Box 6.2 where we defined the sample covariance as an estimator of the true covariance in the statistical population.

With the definition of the expected value of a continuous variable, $Ex = \int xp(x)\,dx$, it is easy to see that independence implies uncorrelatedness:

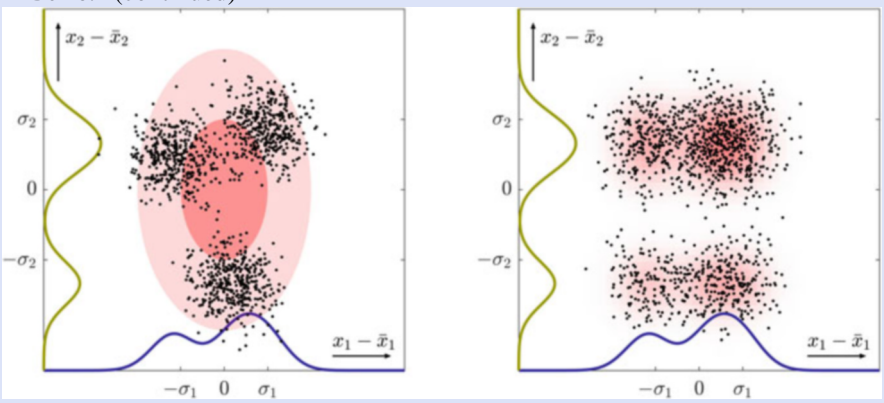
$$p(x_1, x_2) = p(x_1)p(x_2) \;\Rightarrow\; \mathrm{cov}(x_1, x_2) = 0.$$

The reverse is true only for variables having a normal distribution:

$$\left.\begin{array}{c} p(x_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\{-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\} \\ \mathrm{cov}(x_1, x_2) = 0 \end{array}\right\} \;\Rightarrow\; p(x_1, x_2) = p(x_1)p(x_2).$$

(continued)

Box 6.4 (continued)



The figure shows two distributions of two-dimensional data vectors (x_1, x_2). In both cases, x_1 and x_2 are uncorrelated and have the same marginal distributions $p(x_1) = \int p(x_1, x_2)\, dx_2$ (blue) and $p(x_2) = \int p(x_1, x_2)\, dx_1$ (green). The covariance matrices (see Box 6.2) of both distributions are identical; they are diagonal with entries σ_1^2, σ_2^2. The covariance matrix is shown as an error ellipse in the left part of the figure. In the distribution shown in the right figure, but not in the left one, the variables x_1 and x_2 are independent. It is generated by multiplication of the marginal distributions, $p(x_1, x_2) = p(x_1)p(x_2)$ such that the condition for independence is satisfied.

In computational neuroscience, sparse coding has been employed to tackle the question of how natural data sets (mostly images) can best be represented by receptive fields and how these optimal representations relate to the experimentally identified specificities of sensory neurons. Note that for this question, the biological realism of the learning procedure is not crucial, because optimality is interesting in itself, irrespective of how it may have been achieved. In any case, sparsity of neural firing is indeed an element of neural information processing; see, for example, Froudarakis et al. (2014). Models of neural specificity based on sparse coding have been suggested, for example, for edge orientation in simple and complex cells (Olshausen and Field 1996; Hyvärinen and Hoyer 2001), or in the processing of binocular dispariy (Ecke et al. 2021).

► **Key Point: Sparse Coding** Sparse networks represent each stimulus with a low number of neurons active at any one time. For this, the receptive fields of the neurons need to represent meaningful sub-pattern of the input set. Sensible sub-pattern can be found by the learning procedure described in this section or more generally by independent component analysis (ICA).

6.5 Continuous-Field Attractor

So far, we have considered neural information processing either as filtering: that is, the projection of input vectors on existing weight vectors or receptive fields, or as weight dynamics optimizing the weight vectors in one way or another. While the filtering approach does not involve memory at all, weight dynamics is the standard model of learning and longterm memory. Besides these two approaches, a third type of information processing is activation dynamics in which activity reverberates in the network and thereby forms a type of working memory. This type of processing is indeed very common in the brain, given the fact that the internal connectivity vastly outnumbers the input and output connections through the afferent and efferent nerves.

In computational neuroscience, activation dynamics is studied in a number of contexts, including, for example, the generation of spatiotemporal activity patterns for motor control ("central pattern generators," Grillner 2003; Ijspeert 2008), the dynamics of inhibition in cortical circuits and its role in the formation of spatiotemporal receptive fields (e.g., Krone et al. 1986; Isaacson and Scanziani 2011), cortical oscillations (Buzsáki and Vöröslakos 2023), etc. Here, we focus on one important example of information processing with activation dynamics, namely the continuous-field or Amari[13] attractor (Amari 1977), that has recently gained substantial interest in modeling the hippocampal system of place and grid cells (Samsonovich and McNaughton 1997; Baumann and Mallot 2023).

The continuous field approach models neural layers as two-dimensional sheets with activity distribution $a(x, y, t)$ in the same way as we did in Chaps. 2 and 3. Connectivity within the layer is modeled by convolution with a kernel $w(x, y)$ such that the total input from the net acting on a neuron at position x, y would be given by

$$n(x, y, t) = \iint w(x - x', y - y')\, a(x', y', t)\, dx' dy', \tag{6.36}$$

where the integral is taken over the entire layer.

Without input, the potential of each cell is assumed to decay to zero with time constant τ. This can be modeled by a relaxation equation of the type discussed already in Sect. 1.2.2

$$\tau \frac{\partial u}{\partial t} = -u. \tag{6.37}$$

[13] Shun'ichi Amari (born 1936). Japanese engineer and neuroscientist.

To this equation, we add the inputs from the net, $n(x, y, t)$, and possible external stimuli $s(x, y, t)$ as driving forces and obtain (Wilson and Cowan 1973):

$$\tau \frac{\partial u(x, y, t)}{\partial t} = -u(x, y, t) + n(x, y, t) + s(x, y, t) \tag{6.38}$$

The activity $a(x, y, t)$ depends on the intracellular potential $u(x, y, t)$, from which it is derived by a point nonlinearity f:

$$a(x, y, t) = f(u(x, y, t)). \tag{6.39}$$

The kernel $w(x, y)$ is assumed to be independent of time, which is of course a simplification for the case that transmission times are low. It may take negative values to model inhibition.

Figure 6.6a illustrates a one-dimensional version of Eq. 6.38: that is, a section through a layer of neurons with lateral inhibition. For the one-dimensional kernel, we have $w(x) > 0$ for $x < x_o$ and $w(x) < 0$ for $x > x_o$. In the figure, we chose $x_0 \approx 1.5$, such that each neuron receives activity from itself and its immediate neighbors, while it is inhibited by more distant neurons. Inhibition decays toward zero for $r > 3$. Note that the type of lateral inhibition discussed here is a feedback connectivity, whereas lateral inhibition discussed in Sect. 2.2.2 is entirely feedforward.

Amari (1977) considers the dynamics of Eq. 6.38 with a step nonlinearity (i.e. $a(x, t) \in \{0, 1\}$) and $s(x, t) = 0$ and proves the existence of stable regions of activity in the network. The argument is simple: Consider an activity "bump" of width b, i.e., an activity pattern with $a(x) = 1$ in the interval $(0, b)$ and $a(x) = 0$ everywhere else. The kernel w is a difference of Gaussians or a similar function taking positive values for $|x| < x_o$ and negative values fur $|x| > x_o$ (see Fig. 6.7). We denote by n_+ and n_- the network influences on a neuron located at the right margin of the bump, $x = b$:

$$n_+ = \int_{\max(0, b-x_o)}^{b} w(x - b)dx \tag{6.40}$$

$$n_- = -\int_{0}^{\max(0, b-x_o)} w(x - b)dx; \tag{6.41}$$

they are marked as green and red areas in Fig. 6.7.

The neuron at the margin but just outside the activated region, $x = b$, will receive the input $n_+ - n_-$. If $0 < b < x_0$, there will be only excitatory inputs ($n_- = 0$) and the potential at the margin will grow. The marginal neuron will therefore be included in the active region, meaning that the width b will increase. As soon as the region width exceeds the width of positive coupling, $b > x_o$, the marginal neuron at b will receive some inhibition from neurons more than x_o away. Region width may

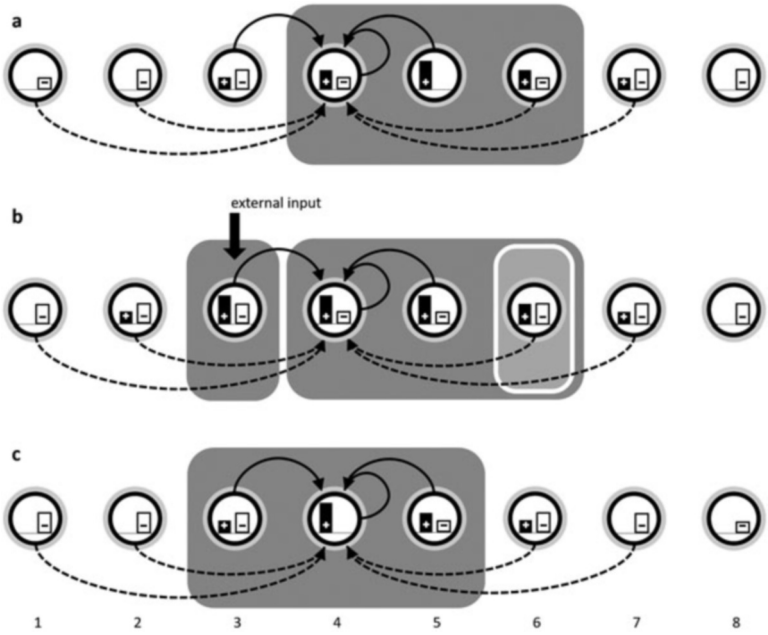


Fig. 6.6 Continuous-field attractor. (**a**) Eight neurons of a lateral inhibition network. The inputs of neuron 4 are shown as solid lines and arrows for excitation and dashed lines for inhibition; for the other neurons, the same pattern is assumed. The histograms inside each unit indicate total received excitation and inhibition, n_+ and n_-. Assume that neurons 4, 5, and 6 are currently active and that a neuron will fire in the next time step if excitation exceeds inhibition. In this case, the three neurons will form a stable activity packet or "bump," marked by the gray oval. In (**b**), neuron 3 receives an additional external input that makes it fire. Neuron 6 will therefore receive additional inhibition (from neuron 3) but no additional activation; it will therefore stop firing. The external input left of the packet will therefore cause a shift to the left. (**c**) Eventually, the activity bump will stabilize at its new position even after the external input stops (Reprinted from Mallot 2023)

still grow but as it does, inhibition n_- will get stronger. A stable state is reached as soon as excitation and inhibition in net input cancel:

$$n = n_+ - n_- = \int_0^b w(x)\,dx = 0, \tag{6.42}$$

see Fig. 6.7. Stable regions of activity therefore exist for lengths b satisfying Eq. 6.42. They may occur everywhere due to the shift invariance of the system.

Activity "bumps" can exist only with the width determined by Eq. 6.42; in Fig. 6.6, they include three neurons. In the kernel, inhibition must exceed excitation ($\int_{-\infty}^{\infty} w(x)\,dx < 0$), because otherwise the active region will grow without limit. Bumps can be initiated by external stimuli with other extends, but will grow or shrink to their stable size as soon as the external stimulus ceases. Figure 6.6b, c shows another important property of the attractor: If a stimulus occurs close to an

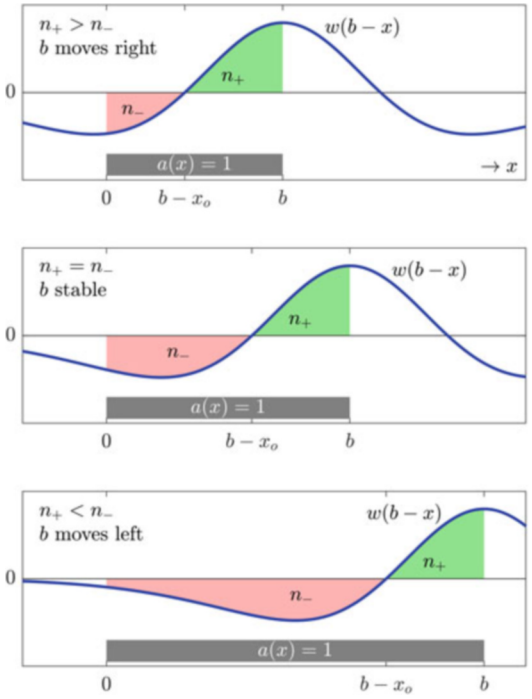


Fig. 6.7 Stability of the continuous-field attractor. The gray bar shows the extent of the active region or "bump," $a(x) = 1$ for $0 < x < b$. The blue curve is the kernel centered at the right margin of the active region, $w(x - b)$. The green and red areas marked n_+ and n_- indicate the amount of excitation and inhibition delivered by the active neurons to the neuron just outside and right of the active region ($x = b$); see Eqs. 6.40 and 6.41. **Top**: For small regions, most of the active neurons deliver excitation to the marginal neuron; the region will grow. **Middle**: The active region stays stable if $n_+ = n_-$. **Bottom**: Large regions are unstable since $n_+ < n_-$. Note that the same argument can be made also for the left margin of the active area, $x = 0$

existing bump, the neuron on the far side of the bump (away from the stimulus) will receive additional inhibition. As a marginal neuron, it formerly received just enough excitation to cancel the inhibition from within the bump, but with the additional inhibition from the stimulus, it will stop firing. As an effect, the bump will move toward the location where the stimulus was delivered and stays there even after the stimulus is switched off.

Moving bumps of reverberating activity are used in various models of working memory (Zylberberg and Strowbridge 2017), but most extensively in models of the hippocampal system for space (Samsonovich and McNaughton 1997; Baumann and Mallot 2023). In the fruit fly *Drosophila*, bumps of moving activity have been directly observed in the so-called protocerebral bridge (Kim et al. 2017). This structure has the topology of a ring on which the bump moves left and right driven

by sensory input from body turns. The current position of the bump represents the forward direction of the animal in a geocentered reference system.

► **Key Point: Continuous-Field Attractor** Neural layers with lateral feedback inhibition support stable bumps of activity that will move in response to external stimuli. This is considered a mechanism of working memory.

6.6 Summary and Further Reading

1. Neural information processing involves a variety of distinguishable tasks for which different network models have been developed. Besides sensory filtering and classification, these include the representation and storage of information, planning and working memory, and motor control.
2. Matrix memories model the storage of associations in a distributed and content addressable way.
3. Competitive learning generates optimized data formats akin to techniques known from signal processing theory. These include the decorrelation of redundant input data and the formation of orderly feature maps.
4. Representations with intermediate levels of abstraction can be constructed by sparse coding.
5. Bumps of neural activity exist as stable attractors on continuous neural fields with lateral inhibition. Moving attractors on neural feature maps are used to model processes of working memory and spatial imagery.

Texts

Arbib (2002): *Rich source of theoretical approaches to neuroscience, covering also many models for specific tasks.*

Haykin (2008): *Established textbook with a focus on mathematics and engineering.*

Trappenberg (2023): *Latest edition of an established textbook covering a wide range of network topologies with a serious interest in neuroscience.*

Suggested Original Papers for Classroom Seminars

Malsburg (1973): *This paper, discussed also in Sect. 6.3.4, first demonstrated how competitive learning can lead to the formation of cortical orientation columns.*

Olshausen and Field (1996): *This paper shows how receptive field organization in the visual cortex can be derived from the principle of sparse coding.*

Samsonovich and McNaughton (1997): *This paper pioneered the use of the continuous field attractor in models of the hippocampal network for space. It was written before the discovery of grid cells by Hafting et al. (2005) but*

covers the main features of the current model. For a tutorial introduction to the computational neuroscience of path integration, see also Chap. 5 of Mallot (2023).

Valentin et al. (1997): *Instructive application of principal component analysis and associative memory to the analysis of face images. This paper will help understand what principal components are and how they can be used in information processing in general.*

Wessberg et al. (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. See also Sect. 6.2.4.*

References

Amari, S.-i. 1977. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics* 27: 77–87.

Arbib, M.A. 2002. *The Handbook of Brain Theory and Neural Networks*. 2nd ed. Cambridge: The MIT Press.

Barlow, H.B. 1972. Single units and sensation: A neuron doctrine for perceptual psychology? *Perception* 1: 371–394.

Baumann, T., and H.A. Mallot. 2023. Gateway identity and spatial remapping in a combined grid and place cell attractor. *Neural Networks* 157: 226–339.

Bednar, J.A. 2012. Building a mechanistic model of the development and function of the primary visual cortex. *Journal of Physiology – Paris* 106: 194–211.

Blakemore, C., and G.F. Cooper. 1970. Development of the brain depends on the visual environment. *Nature* 228: 477–478.

Blasdel, G.G., and G. Salama. 1986. Voltage-sensitive dyes reveal a modular organization in monkey striate cortex. *Nature* 321: 579–585.

Braitenberg, V., and C. Braitenberg. 1979. Geometry of orientation columns in visual cortex. *Biological Cybernetics* 33: 179–186.

Buchsbaum, G., and A. Gottschalk. 1983. Trichomacy, opponent colours coding and optimum colour information transmission in the retina. *Proceedings of the Royal Society (London) B* 220: 89–113.

Buonomano, D.V., and M.M. Merzenich. 1998. Cortical plasticity: From synapses to maps. *Annual Review of Neuroscience* 21: 149–186.

Buzsáki, G., and M. Vöröslakos. 2023. Brain rhythms have come of age. *Neuron* 111: 922–926.

Ecke, G.A., H.M. Papp, and H.A. Mallot. 2021. Exploitation of image statistics with sparse coding in the case of stereo vision. *Neural Networks* 135: 158–176.

Elbert, T., C. Pantev, C. Wienbruch, B. Rockstroh, and E. Taub. 1995. Increased cortical representation of the fingers of the left hand in string players. *Science* 270: 305–307.

Froudarakis, E., P. Berens, A.S. Ecker, R.J. Cotton, F.H. Sinz, D. Yatsenko, P. Saggau, M. Bethge, A.S. Tolias. 2014. Population code in mouse V1 facilitates readout of natural scenes through increased sparseness. *Nature Neuroscience* 17: 851–857.

Grillner, S. 2003. The motor infrastructure: From ion channels to neuronal networks. *Nature Reviews Neuroscience* 4: 573–586.

Hafting, T., M. Fyhn, S. Molden, M.-B. Moser, and E.I. Moser. 2005. Microstructure of a spatial map in the entorhinal cortex. *Nature* 436: 801–806.

Haykin, S. 2008. *Neural Networks and Learning Machines*. 3rd ed. Upper Saddle River: Pearson Prentice Hall.

Hebb, D.O. 1949. *The Organization of Behaviour*. New York: Wiley.

Hopfield, J.J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* 79: 2554–2558.

Hubel, D.H., and T.N. Wiesel. 1963. Shape and arrangement of columns in cat's striate cortex. *Journal of Physiology* 165: 559–568.

Hubel, D.H., and T.N. Wiesel. 1977. Ferrier Lecture: Functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society (London) B* 198: 1–59.

Hyvärinen, A., and P.O. Hoyer. 2001. A two-layer sparse coding model learns simple and complex cell receptive field and topography from natural images. *Vision Research* 41: 2413–2423.

Hyvärinen, A., and E. Oja. 2000. Independent component analysis: algorithms and applications. *Neural Networks* 13: 411–430.

Ijspeert, A.J. 2008. Central pattern generators for locomotion in animals and robots: A review. *Neural Networks* 21: 642–653.

Isaacson, J.S., and M. Scanziani. 2011. How inhibition shapes cortical activity. *Neuron* 72: 231–243.

Ito, M. 2006. Cerebellar circuitry as a neuronal machine. *Progress in Neurobiology* 78: 272–303.

Kim, S.S., H. Rounault, S. Druckmann, and V. Jayaraman. 2017. Ring attractor dynamics in the *Drosophila* central brain. *Science* 356: 849–853.

Kohonen, T. 1972. Correlation matrix memories. *IEEE Transactions on Computers* c-21: 353–359.

Kohonen, T. 1982. Self-organized formation of topological correct feature maps. *Biological Cybernetics* 43: 59–69.

Kohonen, T., P. Lehtiö, J. Rovamo, J. Hyvärinen, K. Bry, and L. Vainio. 1977. A principle of neural associative memory. *Neuroscience* 2: 1065–1076.

Krone, G., H.A. Mallot, G. Palm, and A. Schüz. 1986. The spatio-temporal receptive field: A dynamical model derived from cortical architectonics. *Proceedings of the Royal Society (London) B* 226: 421–444.

Lebedev, M.A., and M.A.L. Nicolelis. 2017. Brain-machine interfaces: From basic science to neuroprostheses and neurorehabilitation. *Physiological Reviews* 97: 767–837.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86: 2278–2324.

Mallot, H.A. 2023. *From Geometry to Behavior. An Introduction to Spatial Cognition*. Cambridge: The MIT Press.

Malsburg, C. von der. 1973. Self–organization of orientation sensitive cells in the striate cortex. *Kybernetik* 14: 85–100.

Oja, E. 1982. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology* 15: 267–273.

Oja, E. 1989. Neural networks, principal components, and subspaces. *International Journal of Neural Systems* 1: 61–68.

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

Ramachandran, V., and W. Hirstein. 1998. The perception of phantom limbs. The D. O. Hebb lecture. *Brain* 121: 1603–1630.

Samsonovich, A., and B.L. McNaughton. 1997. Path integration and cognitive mapping in a continuous attractor neural network model. *Journal of Neuroscience* 17: 5900–5920.

Stein, B.E., and T.R. Stanford. 2008. Multisensory integration: current issues from the perspective of the single neuron. *Nature Reviews Neuroscience* 8: 255–266.

Steinbuch, K. 1961. Die Lernmatrix. *Kybernetik* 1: 36–45.

Trappenberg, T.P. 2023. *Fundamentals of Computational Neuroscience*. 3rd ed. Oxford: Oxford University Press.

Turrigiano, G.G., and S.N. Nelson. 2004. Homeostatic plasticity in the developing nervous system. *Nature Reviews Neuroscience* 5: 97–107.

Valentin, D., H. Abdi, B. Edelman, and A.J. O'Toole. 1997. Principal component and neural network analyses of face images: What can be generalized in gender classification? *Journal of Mathematical Psychology* 41: 398–413.

Wachtler, T., E. Doi, T.-W. Lee, and T.J. Sejnowski. 2007. Cone selectivity derived from the responses of the retinal cone mosaic to natural scenes. *Journal of Vision* 7(8): 6:1–14.

Wessberg, J., 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. 2000. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature* 408: 361–365.

Willshaw, D.J., O.P. Buneman, and M.C. Longuet-Higgins. 1969. Non-holographic associative memory. *Nature* 222: 960–962.

Wilson, H.R., and J.D. Cowan. 1973. A mathematical theory of functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* 13: 55–80.

Zylberberg, J., and B.W. Strowbridge. 2017. Mechanisms of persistent activity in cortical circuits: Possible neural substrates for working memory. *Annual Review of Neuroscience* 40: 603–627.

Coding and Representation

7

Abstract

Neural activity does not in itself contain much information about the stimulus. Spikes elicited by different stimuli look quite the same, but they do occur in different cells. The information is thus contained in the specificity of the cell generating the spike, together with the spiking activity itself. This may be considered a modern account of the notion of the "specific sense energies" formulated by Müller (1837): Stimulation of the eye is perceived as light, color, or motion because it activates visual pathways; similarly, stimulation of the ear is perceived as noise, a pitched tone, or a human utterance, because it activates auditory parts of the brain. The detailed information encoded by each neuron is described by its tuning curve which summarizes the neuron's response to possible stimuli. 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. This chapter explores the consequences of population coding for neural information processing. It also discusses localization: that is, the fact that neighboring neurons in the cortex tend to have overlapping receptive fields and tuning curves. Cortical areas therefore often form "maps" of represented parameters such as visual field position, acoustic pitch, or the surface of the body.

Learning Objectives

- Interplay of spike rate, spike timing, and neural specificity in the coding of information by single neurons and neural populations
- Basic concepts from information theory and their application to neuronal firing
- Population coding as the main coding scheme in the brain

(continued)

H. A. Mallot, *Computational Neuroscience*,
https://doi.org/10.1007/978-3-031-75705-1_7

- Perceptual phenomena related to population coding: interpolation, hyperacuity, contrast enhancement, and aftereffects
- Combination of population coding and local connectivity in topological maps
- The log-polar function as a model of retinoptic mapping in the visual cortex

7.1 Specificity Revisited

In Sect. 2.1, we introduced the specificity of a neuron as the subset of stimuli to which the neuron is responding and formalized it in the notion of the receptive field. We now generalize this idea by considering *representational* specificity of neurons which do not necessarily belong to the sensory system. In the motor cortex, this may be the specificity for a certain volume in grasp space to which the hand will move after the neuron's firing; in the head-direction cells of the thalamus, firing indicates the direction of current bodily movement in a geocentric reference system; in the inferior temporal cortex, face neurons will fire for the recognition of familiar people irrespective of the sensory modality (look, voice, and written name) that they are recognized from. In these examples, specificity in not just about the "reception" of stimuli, but applies to highly processed perceptions, intentions, and memories alike. As pointed out by Barlow (1972) and discussed in Sect. 5.4.3, the firing of a neuron indicates some meaningful event or piece of information, which we might call its "representational field." Like receptive fields, representational fields can be described as profiles or functions; the domain of these functions, however, is not just a sensory surface such as the skin or the retina, but a possibly high-dimensional space of experimental parameters and conditions, and, indeed, anything that may be represented in the brain. In the sensory system, the profiles of representational fields are known as tuning curves.[1]

Tuning Curves

Consider an experiment in which a parameter x of the stimulus or task is continuously varied. A neuron is said to be tuned to this parameter, if it responds strongly in conditions where the parameter takes a specific value, but not in others. The responded value is called the preferred parameter value $\hat{x}$ (read "x hat").

Figure 7.1 shows a number of examples for tuning curves from the visual and acoustic systems (a–c) as well as firing fields of hippocampal place cells (d) as an example of a "representational" field. In a, b, and d, the dependent variable is activity

[1] The term has been coined in analogy to the "tuning" of a radio receiver to the carrier frequency of a transmission. In its present meaning, it seems to have first been used in the context of hearing, that is for the curves shown in Fig. 7.1c.

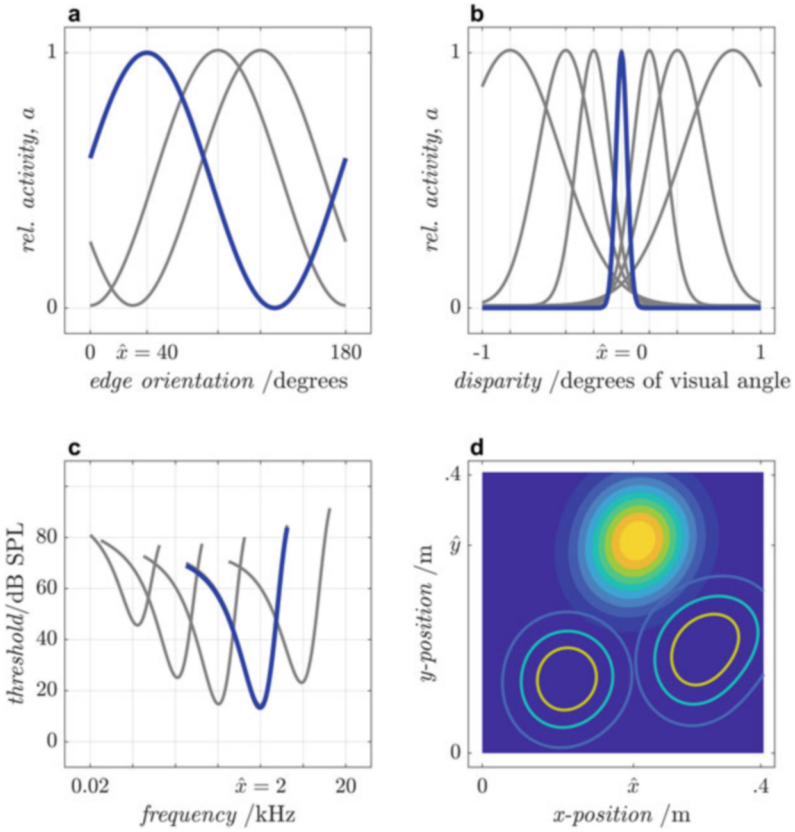


Fig. 7.1 Tuning curve examples from different neural systems. All curves are schematic. $\hat{x}$, $\hat{y}$: preferred parameter values. (**a**) Simple cells in primary visual cortex are tuned to the orientation of contrast edges, measured as an angle from the visual horizontal (see Sect. 3.2). The figure shows curves for three cells; the one shown in blue has preferred orientation $\hat{x} = 40°$. Tuning is wide with large overlaps between curves and roughly follows a cosinusoidal function (Hubel and Wiesel 1959). (**b**) Tuning to stereoscopic disparity, or the horizontal offset of image position in the left and right eye. Tuning is sharp for small disparities and gets wider for positive and negative disparities, i.e., for objects appearing further out or closer than the fixation point (Poggio et al. 1988). (**c**) In hearing, tuning curves for sound frequency are usually measured as threshold curves and therefore appear inverted. The minimum of each curve marks the point of highest sensitivity of a neuron from the cochlear nerve where the threshold is lowest. The curves look similar for single-cell recordings (Ohlemiller and Echteler 1990) or when measured with the psychophysical masking paradigm (Moore 2013, Fig. 3.2). Thresholds of sound pressure level (SPL) are measured in decibel (dB). (**d**) Place cells in the rat hippocampus fire when a freely running rat passes a certain position in the maze, in this case in a 40 by 40 cm square arena. This is an example of a two-dimensional tuning curve (or representational field) with the spatial x- and y-coordinates as parameter vector (O'Keefe et al. 1998). The heat map shows the activity of one cell centered at $\hat{\mathbf{x}} = (\hat{x}, \hat{y})$, while the activities of two other cells are indicated as contour lines

or spike rate, averaged over some experimental epoch. In Fig. 7.1c, the y-axis shows the stimulus strength at threshold, which is a monotonic function of the inverse of activity. Activity itself would thus be a maximum curve peaking at the preferred parameter value $\hat{x}$ in all cases.[2] The parameter value where maximum response is obtained is called the preferred parameter and denoted as $\hat{x}$ or $\hat{\mathbf{x}}$ for scalar and vectorial parameters, respectively.

In analogy to Eq. 2.1, we formalize the situation as

$$f(x) = \mathrm{P}(e = 1 \mid x) \tag{7.1}$$

where f denotes the tuning curve, e is the binary firing variable from Eq. 2.1, and x is the encoded parameter. The probability P would usually be taken over one experimental epoch such as a stimulus presentation or a grasping trial. As a consequence, f is an average spike rate, just as the variable a used before. The parameter value

$$\hat{x} = \operatorname{argmax} f(x) \tag{7.2}$$

is called the preferred parameter of the cell.

The receptive field profile as defined in Eq. 2.1 is a special case of the above definition. Receptive field profiles can be considered tuning curves for stimulus position.

Coding Schemes

Neurons are said to be tuned to a given parameter, if the corresponding tuning curve has a sharp, single peak. We distinguish tuning curves with and without overlap (Fig. 7.2b, c) and contrast them with monotonic response curves as shown in Fig. 7.2a.

1. Intensity code (Fig. 7.2a): The activity of the neuron is proportional to (or a monotonic function of) the encoded parameter. This coding scheme is rare in the brain at least for parameters other than stimulus intensity itself. One example is the contrast response function in primary visual cortex where increased contrasts result in higher activity (Albrecht et al. 2002). The resolution of intensity coding is limited by the dynamic range of the neuron (0 to <500 spikes per second). The absolute firing rate of a neuron also provides a measure of confidence for the presence of the neuron's preferred stimulus, an interpretation which is also compatible with the neurophysiological result.

[2] An interesting exception from the single-peak shape is found in the entorhinal grid cells which have multi-peaked and indeed periodic firing fields in space (Hafting et al. 2005). In this case, the parameter they are tuned to is not the rat's position in space itself, but the offset of this position from the nodes of a fixed triangular lattice. All possible offsets fill a rhomboid as the primitive cell of the triangular lattice, and the tuning curve over this rhomboid has a single peak. See also chapter 5 of Mallot (2023).

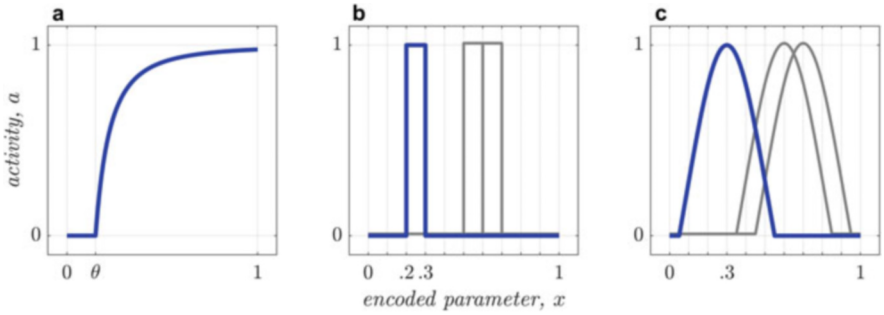


Fig. 7.2 Three schemes for coding a stimulus parameter into neural activity (rate coding). (**a**) Intensity coding: above a threshold θ, activity is a monotonic function of the encoded parameter. (**b**) Labeled line coding with nonoverlapping channels. The blue curve shows the tuning curve for one neuron; two others are shown as gray lines. Parameter values falling into the same channel cannot be distinguished. (**c**) Population coding or labeled line coding with overlapping channels. Each parameter value is represented by the activities of many neurons from which it can be reconstructed with high resolution

2. Channel coding or labeled line coding without overlap (Fig. 7.2b): The set of all possible parameter values—for example, the interval (0, 1)—is partitioned into n parts, or bins. For each bin i, a detector neuron exists firing with some fixed probability p whenever the parameter falls within the respective bin. The tuning curve of each detector neuron is p inside the bin and 0 elsewhere. Technically, this coding scheme is known from analog-to-digital converters (ADC). In neural systems, it is probably not used to encode continuous variables but may play a role in winner-take-all networks and decision-making processes.
3. Population code or labeled line code with overlap (Fig. 7.2c): Each stimulus is coded by a set or population[3] of neurons each of which is tuned to a section of the coded stimulus. The tuning curves overlap. This is the standard way of neural coding as shown in the examples given in Fig. 7.1. Neurons may also be tuned to multiple parameters, such as the simultaneous tuning to orientation, spatial frequency, and motion direction in simple cells of the primary visual cortex. In this case, the parameter x becomes a vectorial quantity $\mathbf{x}$, as is also the case in the tuning to place shown in Fig. 7.1d.

Population codes need more than one neuron to encode a parameter value, whereas in simple channel coding, one neuron would suffice. Despite this apparent disadvantage, it can be shown that population codes are superior to nonoverlapping channels in many respects. This will be discussed in Sect. 7.2.

[3] The terms "ensemble" and "ensemble coding" are used alternatively to "population" and "population coding."

Rate Coding vs. Spike Time Coding

In Eqs. 2.1 and 7.1, we have assumed that information is encoded in average spike rates and that the detailed timing of spikes is irrelevant. In the behaving animal, this may not be appropriate since behavioral responses will often be required already after one or a few spikes arriving at the sensory systems. In some cases, this problem may be overcome by averaging over ensembles of neurons rather than over time epochs, but this requires that many neurons be employed with the same task. It has therefore been suggested that besides the average firing rate, exact spike timing may also carry important information.

One case in point is the encoding of sound frequency in the spiral ganglion and cochlear nerve. In addition to the tuning behavior shown in Fig. 7.1c, a second coding scheme is used, which is known as volley coding (see, for example, Heil and Peterson 2015, and the textbooks of physiology). Spikes are phase-locked to the sound wave since they tend to occur at a certain bending stage of the outer hair cell stereocilia. At low frequencies, the spike train of a single fiber will therefore be in fixed phase with the sound wave. At high frequencies, however, the fibers cannot produce a spike at every cycle of the sound wave. Since the phase locking remains, the periodicity of the sound wave is still maintained in the "volleys" of spikes generated by a number of fibers. Volley coding is observed up to a sound frequency of about 3 to 4 kHz and seems to play a role in the perception of pitch of complex sounds, in particular for missing fundamental stimuli. For higher tones, the perception of musical intervals rapidly declines, which is probably the reason why the highest tones used in instrumental music are at about 4 kHz (highest key on the piano keyboard: C8 = 4.186 kHz); see Moore (2013), Chapter 6.7.

Another shortcoming of the rate-coding approach lies in the fact that neurons are extremely good in detecting the temporal coincidence of spiking events which would not be reflected in spike rates. In directional hearing, for example, humans can distinguish the direction of the source of a pure tone of 900 Hz at 1° offset left or right from the midline, at least if the stimulus is lasting for some time (see Moore 2013, chapter 7.2). This corresponds to a time delay between the two ears (interaural time delay, ITD) of just 10μs: that is, one hundredth of the ordinary duration of a spike. This performance is thought to depend on binaural neurons as have been described in the auditory systems of birds and mammals (Grothe et al. 2010). In our context, it is interesting to note that the ITD specificity of these neurons is again expressed as a tuning curve: that is, once the binaural level is reached, rate coding again is obtained.

Exact spike timing relative to ongoing brain rhythms may also carry information. One such rhythm found in electroencephalography (EEG) is the theta-oscillation of some 6–10 Hz originating from the hippocampus. The detailed timing of place cell activity (overall spike rate appearing in Fig. 7.1d) shows a characteristic pattern of phase relation to the theta cycle, known as phase precession: The phase of a particular neuron's firing indicates whether the animal is entering or leaving this cell's firing field. In the visual cortex, phase relations between spikes and the surrounding local field potential have been shown to play a role in the encoding

of movie stimuli (Montemurro et al. 2008). For an overview of phase coding, see Wang (2010).

Temporal patterns in spiking activity have also been linked to the idea of "cell assemblies" as a major element of neural information processing. The term goes back to Donald Hebb[4] (1949) and implies that information is not primarily conveyed by the activity of single neurons, but in spatiotemporal activity patterns over highly connected sub-nets of the brain. The activation of an assembly would represent complex objects or events, integrating the specificities of the members of the assembly (see Palm et al. 2014, for review). Activation would spread and reverberate within the assembly and may "ignite" other assemblies to form associations. In single-cell recordings, assemblies would show in temporal regularities of spike trains or in correlations and synchronizations between the spike trains of different members of the same assembly. Indeed, it has been shown that the frequency of synchronous spiking events of a pair of neurons may be partially independent of the current spike rates of these neurons and that synchronous events encode additional information (Riehle et al. 1997). However, the identification of assemblies and meaningful temporal spiking patterns in neurophysiological data remains difficult. We will not pursue this idea further in this introductory text.

► **Key Point: Neural Coding** The brain codes information mostly in the relative activity (spike rate) of groups of neurons with partially overlapping specificities (population coding). Absolute spike rates reflect stimulus intensity or confidence. Temporal patterns of activity play a role in acoustical signals, coincidence detection, phase coding, etc.

7.2 Population Code

Tuning curves of individual neurons are generally wide and show substantial overlap with the tuning curves of other neurons (see Fig. 7.1). The firing of a neuron therefore will provide only rough information about the parameter value. In order to obtain a more precise estimate, it is necessary to look at the pattern of activities over larger groups or "populations" of neurons. Decoding the information from the population activity, if at all required in the brain, involves an extra computation step. At first glance, population coding may look like a compromise that the nervous system is forced to make since nonoverlapping tuning curves as shown in Fig. 7.2b are hard to realize in biological systems. It has, however, a number of advantages concerning both the information capacity and information processing. In this section, we will discuss these advantages as well as simple decoding schemes.

[4] See Footnote 8 in Chap. 5.

7.2.1 Information Content of Population Codes

Shannon[5] Entropy

In the theory of information, pioneered by Shannon and Weaver (1949), the information or "entropy" of a message is defined as the minimal number of binary questions needed on average to reconstruct that message (Cover and Thomas 2006; Stone 2015).

This definition needs a few explanations. First, by *binary* questions, we mean questions that can be answered "yes" or "no." If the message consists of numbers, we could ask: "is it 4?"; but also: "is it smaller than 5?" In both cases, the alternative would be the logical complement: that is, "it is different from 4!" or "it is larger than or equal to 5!" Second, the term *"minimal"* refers to the questioning scheme applied. For example, if the message is a letter, we could ask for each letter of the alphabet one by one, which is less efficient than using narrowing questions of the type: "is it in the first half of the alphabet (i.e., letters A–M)?" Third, the same optimal questioning scheme is used for all messages arriving on a given channel and *averaging* is over this set of messages.

Another way to think about this definition is to take the answers to the binary questions corresponding to each message, using "1" for "yes" and "0" for "no." The questioning scheme then transforms each message into a string of zeros and ones and the information contained in the message is the length of its corresponding string. The questioning scheme is then called the "code."

In order to formalize these ideas, we consider a set of symbols $x_1, \ldots, x_n$ from which a message is drawn. This set is called the "alphabet" and may consist of the 26 letters of the English alphabet, the four base pairs of a DNA string, or the 2^k possible patterns of (0, 1)-activity that a population of k neurons can produce in each time step. The alphabet contains the possible outcomes of a discrete random variable X with values in $\{x_i, i = 1, \ldots, n\}$; its distribution is given by the probabilities

$$p_i = \mathrm{P}(X = x_i), \quad i = 1, \ldots, n \tag{7.3}$$

with $\sum_{i=1}^{n} p_i = 1$. In the simple case illustrated in Fig. 7.3a, n is a power of 2, $n = 2^3 = 8$ such that a regular tree with three levels is formed. Let us assume that all probabilities p_i are equal, which implies

$$p_i = \frac{1}{8} = 2^{-3} \quad \text{for all } i. \tag{7.4}$$

In the questioning scheme of Fig. 7.3a, each question cuts the uncertainty about the outcome into half, such that three questions suffice to distinguish between the eight alternatives. Rewriting Eq. 7.4 as

[5] Claude E. Shannon (1916–2001). US mathematician and engineer.

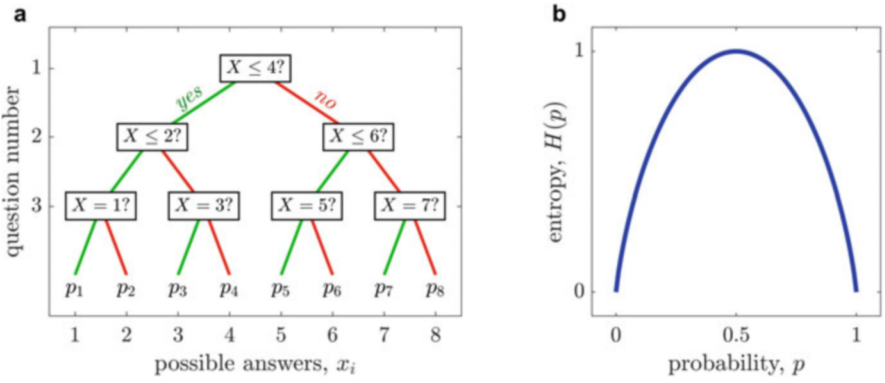


Fig. 7.3 (**a**) Questioning tree for the recovery of a message x that may take eight different values. With three binary questions, $2^3 = 8$ alternatives can be distinguished. $p_1, \ldots, p_8$ are the probabilities that x takes the respective value: $p_i = \mathrm{P}(x = i)$. (**b**) Binary entropy function. The information obtained from a binary question is largest if the alternatives are equally likely ($p = 0.5$)

$$3 = -\log_2 p_i\,,$$

we notice that the minimal number of questions required to recover the value x_i of X equals the negative of the dual logarithm of the probability p_i of this value occurring. The average number of required questions is therefore the expected value of $-\log p_i$,

$$H(X) := \mathrm{E}(-\log_2 p_i) = -\sum_{i=1}^{n} p_i \log p_i\,. \tag{7.5}$$

H is called the entropy[6] of the random variable X or its distribution $(p_1, \ldots, p_n)$; it is measured in "bits" or "bits per symbol."

If $n = 2$, we may write $p_1 = p$ and $p_2 = 1 - p$ and obtain the binary entropy function

$$H(p) = -p \log p - (1 - p) \log(1 - p); \tag{7.6}$$

it is illustrated in Fig. 7.3b. Easy calculation yields $H(0.5) = 1$, which means that one question is needed to recover a binary message whose alternatives are equally likely. The message thus contains 1 bit of information, and this also marks the maximum that can be recovered by one question.

[6] This name was chosen to emphasize the analogy to Boltzmann's formula for thermodynamical entropy, $S = k \ln W$. The letter H in meant to be an upper case Greek Eta (for entropy), not the eighth letter of the Latin alphabet.

This is quite intuitive, and it shows that good questioning schemes should use questions that split the remaining alternatives into equally probable halves. What is less intuitive, however, is that unequal alternatives contain smaller, and indeed non-integer amounts of information. For example, for $(p, 1-p) = (0.75, 0.25)$, we have $H(0.75) = H(0.25) = 0.811$. We might therefore wonder what exactly is meant by asking 0.811 questions. The answer lies in the averaging process mentioned in the definition of entropy: in the case of unequal alternatives, it is possible to device questioning schemes that recover more than one symbol with just one question. To see this, let us ask for two symbols at a time, i.e., for the outcome of the vectorial random variable (X^{t-1}, X^t), where t indexes time. With the alphabet $\{0, 1\}$ and the distribution $(0.75, 0.25)$, we might start by asking "are the next two digits 00?" The probability that this is true and that we are done with just one question is $q_1 = 0.75^2 = 0.5625$. If the answer is "no," the next question could be "are the next two digits 01?". The probability that this is true is $q_2 = 0.25 \times 0.75 = 0.1875$. If the answer is "no," the third question would be "are the next two digits 10?" Since any answer to the third question will make the result clear, the probability of being done after exactly three questions is $q_3 = 0.25 \times 0.75 + 0.25 \times 0.25 = 0.25$. The average number of questions to be asked to recover two digits is therefore

$$H(X^{t-1}, X^t) = q_1 + 2q_2 + 3q_3 = \frac{9}{16} + 2\frac{3}{16} + 3\frac{4}{16} = \frac{27}{16} = 1.6875. \tag{7.7}$$

On average, we will thus recover the message with just $1.688/2 = 0.844$ questions per symbol. This is still slightly more than the 0.811 bits calculated above, due to the fact that the answers to the questions are still not equally likely. The so-called source coding theorem of information guarantees, however, that it is possible to device a questioning scheme which gets arbitrarily close to the optimum, for example, by asking for three or more symbols at a time (Cover and Thomas 2006; Stone 2015). The optimal questioning scheme, or code, transforms the message into a string of zeros and ones whose length is the number of bits in the message.

The Entropy of Overlapping Channels

We may now apply these ideas to different neural coding schemes. We will assume that the stimulus parameter x is uniformly distributed over the interval $(0, 1)$. Consider first the labeled line code without overlap and n equally spaced channels (Fig. 7.4a). Each channel has the characteristic or tuning curve

$$f_i(x) = \begin{cases} 1 & \text{if } \frac{i-1}{n} < x \leq \frac{i}{n} \\ 0 & \text{otherwise} \end{cases} . \tag{7.8}$$

By coding the signal in this scheme, it is digitized to steps of $1/n$, each represented by one of the activity patterns $(1, 0, .., 0)$, $(0, 1, \ldots, 0)$, $\ldots$, $(0, 0, \ldots, 1)$. Different parameter values falling into the same interval cannot be distinguished.

The probability of each of these n activity patterns to occur is $1/n$, since we assumed that the parameter x is equally distributed in the interval $(0, 1)$. With n

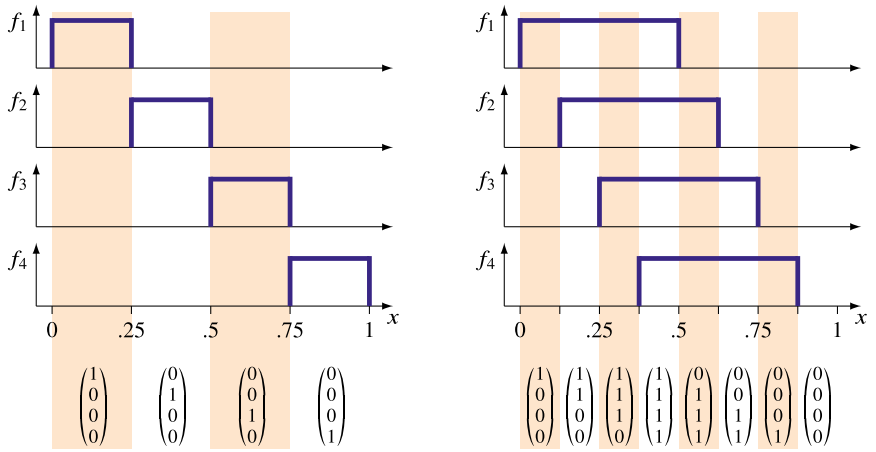


Fig. 7.4 Information transmission in channel-coded systems without overlap (**a**) and with overlap (**b**) With overlap, more information can be transmitted. x is the encoded parameter, and $f_1, \ldots, f_4$ are the tuning curves of four neurons (channels). The vertically printed vectors are the channel activities $(f_1(x), f_2(x), f_3(x), f_4(x))$ for each x-interval. For further explanation, see text

equally likely activity patterns, we may calculate the entropy

$$H = -\sum_{i=1}^{n} \frac{1}{n} \log_2 \frac{1}{n} = \log_2 n. \tag{7.9}$$

With $n = 4$ as in the example of Fig. 7.4a, we can thus transmit $\log_2(4) = 2$ bits per time step. The information transmitted by each single neuron is $H(1/n)$ where H is the binary entropy function from Eq. 7.6. It approaches 0 for large values of n.

Consider now a coding scheme with overlapping channels as shown in Fig. 7.4b. The width of each tuning curve is $1/2$, and the tuning curves of the n channels are shifted by $1/(2n)$:

$$f_i(x) = \begin{cases} 1 & \text{if } \frac{i-1}{2n} < x < \frac{i-1}{2n} + \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}. \tag{7.10}$$

As can be seen from Fig. 7.4b, there are $2n$ different activity distributions in this case. Therefore, the entropy is

$$H = -\sum_{i=1}^{2n} \frac{1}{2n} \log_2 \frac{1}{2n} = 1 + \log_2 n. \tag{7.11}$$

As compared to the case without overlap, information content is increased by one bit, in the example, 3 instead of 2 bits. Put differently, carrying the same information in the interval code would require twice as many neurons. The information conveyed by each single neuron is $H(0.5) = 1$ for all n. This means that in addition to carrying more information, the overlapping code also has more redundancy. We will see in Sect. 7.2.3 how this redundancy generates additional benefits of population coding.

▶ **Key Point: Shannon Entropy** The number of bits of a message is the number of binary questions needed on average to recover the message with an optimal questioning scheme, or "code." When encoding a continuous parameter, a set of wide channels with overlap is more efficient than the same number of narrow, non-overlaping channels.

Box 7.1 Information and Energy

The transmission and processing of information, be it in the computer or the brain, inevitably requires the consumption of energy. In humans, the brain consumes about 15–20% of the basal metabolic rate which amounts to some 150 g of glucose per day. The energetic efficiency of neural computation is therefore an important issue in brain evolution.

The energy consumption of neural computation also underlies functional brain scanning methods based on regional cerebral blood flow and oxygen supply. For example, the blood oxygen level dependent (BOLD) effect in functional magnetic resonance imaging (fMRI) is closely related to the energy consumption caused by the average spike rate in small volumes of the brain (Logothetis et al. 2001).

The basic relation of information and energy consumption is often illustrated by James Clerk Maxwell's thought experiment of a demon operating a gate between two volumes filled with gas (see chapter 8 of Stone 2015). Initially, the system is assumed to be in its thermodynamic equilibrium. The demon will then open the gate whenever a fast molecule is about to pass from volume 1 to volume 2, say, while the gate stays closed for fast molecules moving in the opposite direction. Similarly, slow molecules are allowed to pass from volume 2 to volume 1, but not in the reverse direction. After a while, fast molecules will accumulate in volume 2 which will therefore heat up while volume 1 is cooling down. The heat difference might then be used to drive a power plant.

Maxwell's demon seems to violate the second law of thermodynamics, at least if we assume that the energy needed to control and operate the gate can be neglected. However, even if opening or closing the gate would not require any energy at all, the thermodynamical entropy of the system will grow due to the control part of the process: The speed of the next approaching particle has to be measured and the door command (open or close) has to be

(continued)

Box 7.1 (continued)
set in the gate operating mechanism. For the next particle, this information has to be erased and updated and every time this happens, entropy will grow. This entropy increase suffices to save the second law; it is known as the Landauer limit and amounts to the dissipation of about 3×10^{-21} J/bit at room temperature (exactly kT J/bit where k is the Boltzmann constant and T is absolute temperature).

One might think that the Landauer limit defines the minimal energy consumed by a computer or a brain. However, both systems do much more than simply overwriting old bits by new ones: they generate, transmit, and process physical signals such as spikes or logical level pulses at clock rate. Laughlin (2011) estimates the energy consumption of a synapse to be in the order of 10^{-14} J/bit or 10^7 molecules of ATP per bit of transmitted information, while a 2009 computer needed about 5×10^{-8} J/bit. Energy consumption increases if processing is speeded up or is carried out at higher signal-to-noise ratios. Both figures are way above the Landauer limit which therefore does not seem to be the limiting factor of processing capacity.

Mutual Information and the Case of Graded Tuning Curves

We have seen above that overlapping tuning curves allow more efficient coding than nonoverlapping ones. We now turn to the question of the optimal shape of tuning curves: should they be box-shaped, taking only values zero and one, or graded with smooth peaks and shallow flanks? Graded tuning curves as shown in Figs. 7.1 and 7.2a,c are of course common in the nervous system, but it is not obvious whether this is a bug or a feature. More specifically, we could ask if information can be usefully encoded in graded differences of neural activity or if this information will be rendered useless by noise.

In order to deal with noise in the transmission and representation of information, we now treat the represented parameter x and the neural activity a as random variables, X and A, say. They form a communication channel in the sense of information theory where X is the information source and A is the receiver (see also Borst and Theunissen 1999). As a starting point, we use Eq. 7.1 with two modifications. First, we consider activity rates (spikes per time window) rather than single spikes, and second, we look at joint probability distributions rather than at conditional ones. The channel (X, A) is then described by the joint probability of observing an activity level a together with a parameter value x:

$$p(x, a) \;=\; \mathrm{P}(X{=}x, A{=}a). \tag{7.12}$$

For the sake of simplicity, we will assume that A and X are discrete, but multilevel distributions, and write $p_{ij} = p(x_i, a_j)$. The marginal distributions of the individual variables X and A are given as

$$p_i = p(x_i) = \sum_j p_{ij} \quad \text{and} \quad p_j = p(a_j) = \sum_i p_{ij}. \tag{7.13}$$

From these, we obtain the conditional distributions

$$p(x_i|a_j) = \frac{p(x_i, a_j)}{p(a_j)} = \frac{p_{ij}}{p_j} \quad \text{and} \quad p(a_j|x_i) = \frac{p(x_i, a_j)}{p(x_i)} = \frac{p_{ij}}{p_i}. \tag{7.14}$$

The idea of the information channel is to ask the following question: What, if anything, do we learn about the input X if we know the outcome of A? Clearly, if X and A are independent, that is, if $p_{ij} = p_i p_j$, the answer would be "nothing." Even if there is a dependence of X and A, we cannot expect that all information in A will tell us something about the input X. Indeed, if X is known, A may still vary as described by the conditional distribution $p(A|X)$. We therefore define the "mutual information" I of X and A as the meaningful part of the output entropy: that is, the entropy of the marginal distribution $p(A)$, reduced by the conditional entropy of $p(A|X)$:

$$\begin{aligned} I(X, A) &:= H(A) - H(A|X) \qquad (7.15)\\ &= -\sum_j p_j \log_2 p_j + \sum_i \sum_j p_{ij} \log_2 \frac{p_{ij}}{p_i} \\ &= \sum_i \sum_j p_{ij} \log_2 \frac{p_{ij}}{p_i p_j}. \qquad (7.16) \end{aligned}$$

$I(X, A)$ is symmetric, $I(X, A) = I(A, X)$, and may therefore also be defined as $H(X) - H(X|A)$; the relations are summarized in Table 7.1. In the deterministic

Table 7.1 Channel entropies. The width of the boxes reflects the quantitative relations

$H(X, A) = -\sum_{ij} p_{ij} \log_2 p_{ij}$		
$H(X) = -\sum_i p_i \log_2 p_i$		$H(A\|X) = -\sum_{ij} p_{ij} \log_2 \frac{p_{ij}}{p_i}$
$H(X\|A) = -\sum_{ij} p_{ij} \log_2 \frac{p_{ij}}{p_j}$	$H(A) = -\sum_j p_j \log_2 p_j$	
	$I(X, A) = \sum_{ij} p_{ij} \log_2 \frac{p_{ij}}{p_i p_j}$	

case shown in Fig. 7.4, $H(A|X) = 0$ and the output entropy equals the mutual information. If X and A are independent, the fraction in Eq. 7.16 takes the value 1 and $I(X, A) = 0$. Mutual information is therefore also a graded measure of statistical dependence, which is used in independent component analysis (ICA); see Sect. 6.4.

Mutual information can be used to determine the information encoded in the firing of neurons with continuous tuning curves. To see this, we need to remember that the value $f(x)$ of a tuning curve was defined as the firing probability of the neuron, given the parameter value x. In order to estimate the distribution of A, we assume that the measurement is repeated n times; the probability that the neuron will fire in j out of n cases is given by the binomial distribution

$$\frac{p_{ij}}{p_j} = P\left(A = \frac{j}{n} \,\middle|\, x_i\right) = \binom{n}{j} f(x_i)^j (1 - f(x_i))^{n-j}. \tag{7.17}$$

With this assumption, the mutual information of the channel $(X; A)$ can be calculated according the Eq. 7.16.

Figure 7.5 shows the situation for tuning curves of different width and slope at half-height. The tuning curves are shown in white; they are calculated from an ad-hoc equation allowing the independent variation of the curve's width and maximal slopes. The equation is given in the caption; it has no further significance.

The heat map is the joint distribution p_{ij} for X and A as defined in Eq. 7.17. Mutual information is largest for width $w = 0.5$ and intermediate slope: that is for a tuning curve with a roughly cosinusoidal shape. If X has uniform distribution, the neuron would be firing for about 50% of the presentations, i.e., at the maximum of the binary entropy function, Fig. 7.3b. The slopes allow to convey information about graded stimulus values in the flanks of the tuning curve. The lower row of Fig. 7.5 shows box-like tuning curves which convey less information because the flanks of the curve are not used (see also Seung and Sompolinsky 1993).

We note in passing that increasing the sampling rate in Fig. 7.5 does not affect this result. Indeed, mutual information can be defined for continuous variables simply by replacing the sums by integrals. This is not the case for entropy H which grows to infinity if the widths of the parameter bins go to zero. In the continuous case, it has to be replaced by "differential entropy" which differs from standard entropy by a constant term depending on sampling frequency only. This term occurs for joint and conditional entropy alike and therefore cancels out in the calculation of mutual entropy. For details, see Cover and Thomas (2006).

To sum up, we have shown that the information conveyed by the activity of tuned neurons can be increased by overlapping tuning curves with shallow flanks. Further increase can be obtained by combining both principles; for a full theoretical account, see Shao et al. (2023). An extreme example for the superior information capacity of graded and overlapping channels is color vision: With just three channels, i.e., the short, middle, and long wavelength cone mechanisms, more than a million colors can be distinguished. In contrast, graded activities would not add any information in the nonoverlapping case.

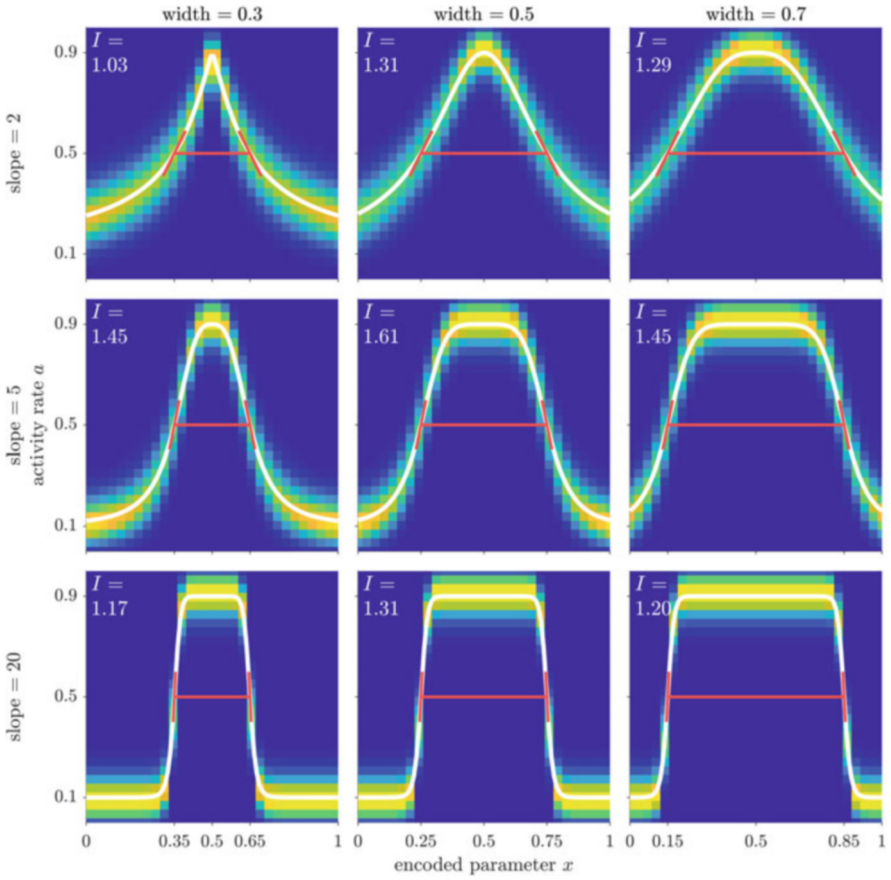


Fig. 7.5 Tuning curves as information channel. The white curve in each plot is the tuning curve $f(x) = 0.1 + 0.8/(1 + (2|x - \hat{x}|/w)^{2sw})$ where $\hat{x} = 0.5$ is the preferred stimulus, w is the width at half-height, and s the slope at half-height. The red lines show s and w. For each value of x_i, the values of a are drawn from a Bernoulli distribution with probability $f(x_i)$. This is repeated for 30 time steps, resulting in a binomial distribution $B(30, f(x_i))$. The heat maps show the joint distributions $p_{ij} = \mathrm{P}(X = x_i, A = j/30)$. Mutual information is largest for the tuning curve in the center. It has a roughly cosinusoidal shape

► **Key Point: Coding by Continuous Tuning Curves** Continuous channels convey more information than discrete channels with box-shaped tuning curves. This is described by mutual information, i.e., the information obtained about the input when the output is known.

7.2.2 Reading a Population Code

For the nervous system itself, 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. The same is true of the medical engineer who wants to build brain–machine interfaces for neuroprostheses. The underlying problem is known as the "decoding" of neural activity patterns; it was already briefly touched in the context of regression; see Sect. 6.2.4. Here we focus on approaches where the specificity of a neuron is explicitly estimated as an intermediate step.

The Center-of-Gravity Estimator

Assume we were encoding a parameter x by the activity of a population of n neurons, each with a preferred parameter value $\hat{x}_i$. As before, the preferred parameter value is the one for which the neuron's tuning curve $f_i(x)$ takes its maximum. The preferred stimulus values must be known to the experimenter in advance, for example, from an independent measurement of the tuning curves. We now present a particular stimulus value, say x_o, which will elicit strong responses in neurons whose preferred value is close to x_o and weaker responses in others. The resulting activities of each neuron, $f_i(x_o)$, can be considered a confidence measure for the presence of the neuron's preferred stimulus, or, indeed, the weight of a "vote" cast by each neuron for its preferred stimulus. The weighted votes are summed over all neurons in the population. Mathematically, we can describe the procedure by the equation

$$x^*_{\mathrm{CoG}} = \frac{\sum_i \hat{x}_i a_i(x_o)}{\sum_i a_i(x_o)} \tag{7.18}$$

where x^*_{CoG} is called the "center-of-gravity" estimator for the represented value x_o. If only two neurons are considered with $\hat{x}_{1/2} = \pm 1$, Eq. 7.18 describes a balance with two weights corresponding to the activities of the neurons and x^*_{CoG} marking the point along the beam of the balance where it needs to be supported in order not to tilt to either side (see Fig. 7.9). The center-of-gravity estimator was introduced by Georgopoulos et al. (1982) who called it the "population vector" because it was applied to a vectorial parameter, namely the target position of a hand movement in three-dimensional space.

Of course, it would be nice to prove in Eq. 7.18 that the estimate x^*_{CoG} is indeed equal to the original parameter value x_o. This, however, is not generally the case and depends on the detailed shape of the tuning curves $f_i(x)$; 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

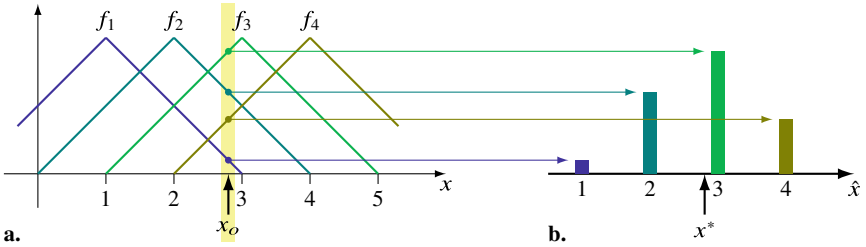


Fig. 7.6 (**a**) An array of equidistant triangular tuning curves (Eq. 7.19). Also shown is a stimulus value x_o which excites all four channels according to their tuning curves. (**b**) The activities generated in these four channels. Their center of gravity equals the original stimulus

"basis functions"; see Poggio and Girosi (1990). We will not study this problem in detail but restrict our discussion to the simple example illustrated in Fig. 7.6.

Consider a population of neurons with triangular tuning curves $f_i(x)$ centered at the preferred stimulus $\hat{x}_i$:

$$f_i(x) = \begin{cases} 1 - \frac{1}{2}|x - \hat{x}_i| & \text{for } |x - \hat{x}_i| < 1 \\ 0 & \text{otherwise} \end{cases} . \tag{7.19}$$

Figure 7.6a shows a set of four such tuning curves all of which yield non-zero output for encoded parameter values in the interval $x_o \in [2, 3]$. The preferred stimuli of each neuron have been identified with the neuron's number, i.e., $\hat{x}_i = i$.

Now, consider a stimulus with parameter value $x_o \in [2, 3]$ (Fig. 7.6a). From Eq. 7.19, it is easy to calculate the activities in the four channels. They are shown in Fig. 7.6b as colored columns. As explained before, the center-of-gravity estimator treats these columns as loads placed along 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. Easy calculation shows that for triangular tuning curves, the center-of-gravity estimator actually recovers the encoded parameter value, $x_o = x^*_{\mathrm{CoG}}$ in Eq. 7.18. For other types of tuning curves or unequal spacing of their centers, this relation will only be an approximation.

Least Squares, Maximum Likelihood, and Bayesian Estimators

An alternative approach to the decoding of population activities is based on the comparison of the current activity pattern with the patterns previously recorded for each value of the parameter. This approach does not make use of the preferred stimulus, which is itself a derived quantity, but is based on the entire shape of the tuning curves. Additionally, the statistical distribution of firing rate and parameter can be taken into account.

In one example, Wilson and McNaughton (1993) recorded from 80 place cells in the rat hippocampus and calculated tuning curves ("firing fields") $f_i(\mathbf{x})$ by averaging

the spike rate of each neuron over all time intervals that the rat spent at location $\mathbf{x} = (x, y)$; see also Fig. 7.1d. In the test phase of the experiment, the activities of the same cells, $a_i(t)$, were monitored and the trajectory of the rat was estimated by minimizing the squared difference between recorded and observed activities:

$$\mathbf{x}^*_{\mathrm{LSq}}(t) = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_i \left(f_i(\mathbf{x}) - a_i(t)\right)^2 . \tag{7.20}$$

The results show good agreement between the true and the estimated trajectories.

The least square estimator of Eq. 7.20 is optimal if the statistical distribution of firing rate is Gaussian with equal variance for all neurons. If the distributions are known explicitly, one may also ask for the parameter value for which the current measurement is most likely to occur. This idea is realized by the maximum-likelihood estimator (MLE) defined as

$$x^*_{\mathrm{ML}} = \underset{x}{\operatorname{argmax}}\, \mathrm{P}(a_1, \ldots, a_n \mid x). \tag{7.21}$$

(Seung and Sompolinsky 1993; Pouget et al. 2000). If the activities a_i are accumulated over some time window, they are often assumed to follow a Poisson distribution with mean $f_i(x)$. With the binary activity variable e used in Eq. 7.1, the conditional distribution of neuronal firing, given the parameter value, consists of just two numbers:

$$\mathrm{P}(e_i = 0 \mid x) = 1 - f_i(x) \tag{7.22}$$

$$\mathrm{P}(e_i = 1 \mid x) = f_i(x)); \tag{7.23}$$

the maximum likelihood estimator is then simply obtained as

$$x^*_{\mathrm{ML}} = \underset{x}{\operatorname{argmax}} \prod_{\{i|e_i=1\}} f_i(x) \prod_{\{i|e_i=0\}} (1 - f_i(x)). \tag{7.24}$$

When treated as a function of x, the conditional probability $\mathrm{P}(a_1, \ldots, a_n|x)$ is called a likelihood function. It often takes very small values and is therefore usually replaced by its logarithm and then called the log likelihood function. Still, its maximum is obtained by the parameter value x best supported by the measurement.

If the probability distribution of the encoded parameter x is also known, it can be used as a "prior[7]" in a Bayesian estimation scheme. The "maximum a posteriori estimator (MAP)" is given by

$$x^*_{\mathrm{MAP}} = \underset{x}{\operatorname{argmax}}\, \mathrm{P}(x \mid a_1, \ldots, a_n) = \underset{x}{\operatorname{argmax}} \frac{\mathrm{P}(a_1, \ldots, a_n \mid x)\, \mathrm{P}(x)}{\sum_x \mathrm{P}(a_1, \ldots, a_n \mid x)}. \tag{7.25}$$

[7] The Latin terms "a priori" and "a posteriori" mean before and after (the measurement), respectively: that is, with and without knowledge of the outcome of the experiment.

The term $\sum_x \mathrm{P}(a_1, \ldots, a_n \mid x)$ in the denominator does not depend on x and can therefore be neglected for the maximum search. If $\mathrm{P}(x)$ is unknown, it is usually assumed uniform and called a "flat" prior. In this case, the MAP estimator reduces to the maximum likelihood (ML) estimator.

► **Key Point: Population Decoding** Estimators for population decoding include center-of-gravity, least squares, maximum likelihood, and maximum a posteriori (or Bayes). They require increasing amounts of information about the channel specificities: preferred stimuli (CoG), full tuning curves (LSq), conditional probability of activity, given encoded parameter (ML), and additionally the probability distribution of the encoded parameter (MAP).

7.2.3 Examples and Further Properties

The significance of population coding in neural information processing is not just its superior efficiency for encoding but stems from a number of additional properties which support computation. These are related to the fact that population codes are "analogic" in the sense that the activity vector $(a_1(x), \ldots, a_n(x))$ of a population is a continuous function of the encoded parameter x as long as the tuning curves themselves are continuous. Small variations of the represented parameter therefore correspond to small variations of the representing activity pattern. We discuss a number of examples.

Hyperacuity (Sub-pixel Resolution)

If visual position is considered as a stimulus parameter, the receptive field functions studied in Chaps. 2 and 3 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 fields include the position in question.

Figure 7.7 shows population coding of visual position already on the level of retinal 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 (in the fovea 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 cast onto the fundus of the eye by a light beam arriving at the cornea. For ideal imaging conditions, i.e., with the iris light adapted and the lens exactly accommodated, this is again in the order of 30 seconds of arc.

Figure 7.7a 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 distribution of activity levels over the receptors differs between the two stimuli. The population activity—and with it the estimators discussed in the previous section—maintains the information about the separation between the two stimuli (cf. Poggio

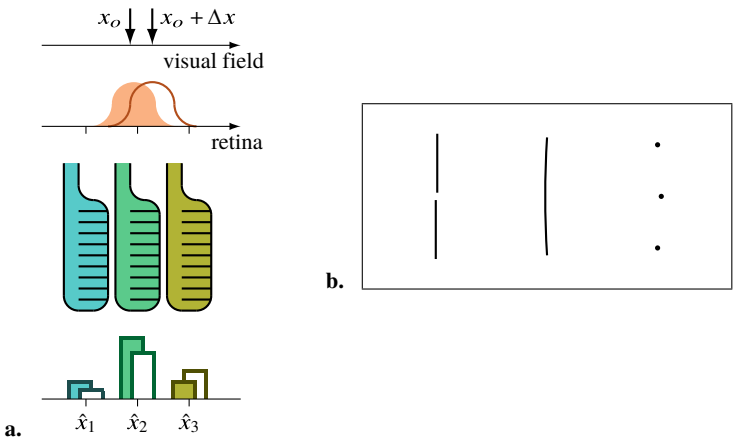


Fig. 7.7 Hyperacuity (or sub-pixel resolution) in a population code for visual position. (**a**) A stimulus appearing at position x_o in the visual field will be washed out into a blurring disk on the retina (reddish bell-shaped curve). The width of blurring is about 0.5 minutes of arc which is roughly equal to the photoreceptor spacing in the fovea. The blurring disc will activate a small population of adjacent photoreceptors with preferred stimulus positions $\hat{x}_1, \ldots, \hat{x}_3$ shown in green–blue colors. The activity pattern over this group is shown by the filled bars in the lower part of the figure. If a stimulus is presented at position $x_o + \Delta x$, a different distribution arises, even if Δx is less than the photoreceptor spacing (open bars). The difference between these two distributions encodes positional offset with sub-pixel resolution. (**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

et al. 1992). In a labeled line code without overlap, stimulus locations can be told apart only if the separation exceeds two cone widths.[8]

Figure 7.7b shows some patterns used to examine visual resolution. These patterns 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 one of three dots is displaced to the left or right of the line connecting the two outer ones (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.

It is interesting to note that the receptive field overlap of adjacent photoreceptors is a consequence of the optical properties of the dioptric apparatus of the eye, not of neural connectivity. Blurring is often considered an unwanted, if unavoidable shortcoming but it is turned into an advantage by allowing for spatial population codes already on the receptor level.

[8] Separation by one pixel is not enough, since one receptor with lower activation is needed in between two active ones. This is formalized in the sampling theorem stating that the highest spatial frequency that can be correctly sampled at n equidistant points is $n/2$. This frequency is known as the Nyquist limit.

Aftereffects and Working Range Adjustment

Sampling a parameter space with overlapping tuning curves allows to adjust the sensitivity of parameter contrasts to the range and variability of this parameter in the input signal. On a relatively short time scale (minutes), this is evident from so-called aftereffects which are an ubiquitous phenomenon in perception. Figure 7.8 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 of 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. For a review of visual aftereffects, see Thompson and Burr (2009).

Aftereffects are strong evidence for population coding in overlapping channels. In each channel, adaptation is modeled as an overall decrease of sensitivity caused by strong and sustained activity in the adaptation phase. This decrease is more pronounced in 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. 7.9). 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.

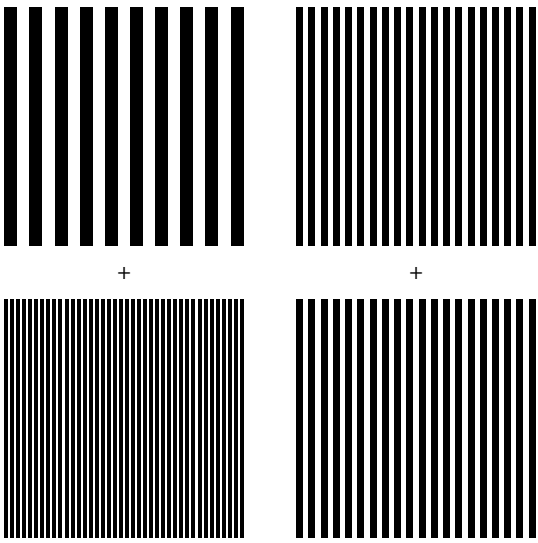


Fig. 7.8 Demonstration of the shift in perceived spatial frequency after adaptation. After closing one eye and looking at the small cross at the left for at least one minute, then looking at the cross on the right, the two striped patterns on the right appear to have different spatial frequencies. Perception emphasizes the contrast between the current pattern and the one the subject was adapted to (adapted from Blakemore and Sutton 1969)

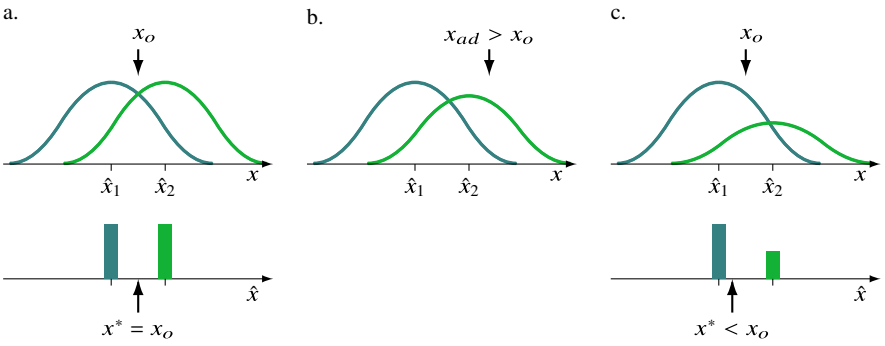


Fig. 7.9 Aftereffects are a common phenomenon in the perception of motion, color, texture granularity, etc. Braddick et al. (1978) suggest that aftereffects are based on the population activity of channels with overlapping tuning curves. (**a**) A stimulus x_o is presented intermediate between the preferred stimuli $\hat{x}_1$ and $\hat{x}_2$ of two channels. The resulting population activity shows two equal excitations of the two channels and the center-of-gravity estimator x^* reproduces the original stimulus x_o. (**b**) The system now adapts to a new stimulus, x_{ad}. Adaption is modeled as a reduction of the sensitivity of active channels, in this case channel 2. (**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

Aftereffects also occur on higher levels of perception. For example, Leopold et al. (2001) created a continuous space of faces by interpolation between a large set of face images. In the experiment, faces on a straight line in face space, passing through the average face, were used. When adapted to a face on one side of this line, the perceived difference of the average face and faces on the other side of the line was increased. This is to be expected if faces are encoded by channels with overlapping multidimensional tuning curves in face space.

Vector-Valued Parameters and Interpolation

The theory of population coding is not restricted to stimuli varying in just one parameter, see, for example, Fig. 7.1d. As a further example, we consider the encoding of pointing movements of the arm, which can be described in a two-dimensional space of pointing targets, parameterized as the 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 described as their "motor fields." The motor fields are "representational" tuning curves in the sense of our discussion, establishing a population code for pointing targets in three-dimensional space. 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 the preferred motion vector is identified for each cell. Next, a pattern of motor cell activity is recorded, from which the azimuth and elevation of the intended arm movement can be calculated by means of the center-of-gravity estimator (Eq. 7.18). By this procedure, the actual arm movement is nicely predicted. If the population vector is monitored over time while

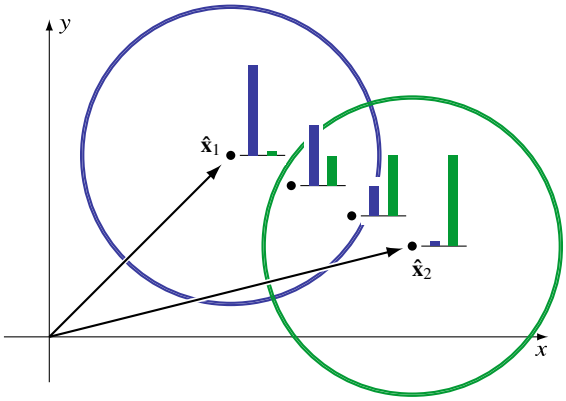


Fig. 7.10 Population coding for multidimensional parameters. In a two-dimensional parameter space, two channels are marked by their preferred stimuli ($\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$) and a circle indicating the tuning range. Four stimuli are marked by black dots in parameter space, together with the two channels' activity shown as insets. As the stimulus changes from $\hat{\mathbf{x}}_1$ to $\hat{\mathbf{x}}_1$, the center-of-gravity estimator will follow continuously

the monkey is planning a movement, it can be seen anticipating this movement: the estimator moves from the current hand position to the target prior to the actual motion execution. This process is also described as "mental rotation."

Figure 7.10 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 $\hat{\mathbf{x}}_i \in \mathbb{R}^2$; in the figure, they are represented by circles. 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 parameters are different pointing directions, the described shift of activity from one channel to another amounts of a rotation of the population vector.

Mental rotation was first described in visual perception as a mechanism of rotational invariance in object recognition (Shepard and Metzler 1971). In the population account of mental rotation sketched out above, invariance is achieved by storing multiple views of each object: that is, by a population of neurons each tuned to the object's view from a given direction. Mental "rotation" between two viewing directions then happens as a decrease of activity in the neuron coding the start view and a simultaneous increase of activity in the neuron representing the goal view (Fig. 7.10). Evidence for neurons in infero-temporal cortex tuned to view directions has been presented by Logothetis et al. (1994).

► **Key Point: Population Coding and Computation** Population coding increases coding efficiency and supports important elements of neural computation. These include sub-pixel resolution, working range adjustment (aftereffects), interpolation, and mental rotation.

7.3 Topological Maps

7.3.1 Locality and Ordered Maps

The neurons constituting a population code may occur anywhere in the brain or a cortical area as long as the neurons of the subsequent processing stage receive the input they need. For example, for the hippocampal place cell system, no obvious ordering scheme seems to exist. Rather, neighboring place cells may have far distant and nonoverlapping place fields, while close-by places may be represented by widely separated place cells.

This is, however, a rather unusual situation: About 50% of the cortical surface is organized in the form of "topological" maps in which neighboring neurons have similar specificities (Sereno et al. 2022). The best-known examples are the body maps in somatosensory and motor cortex (also known as "homunculi"), the mapping of the retina to the visual cortex (retinotopy), and the mapping of the basilar membrane of the inner ear to the auditory cortex (tonotopy). In general, a topological map is characterized by three properties:

First, the neurons in a map establish a population code for the mapped parameter in which neighboring neurons have similar specificities and overlapping tuning curves. The mapped parameter may be the receptive field position in a sensory surface such as the retina, skin, or basilar membrane, but may also represent more derived features. For example, edge orientation is mapped in the hypercolumns of the primary visual cortex; views of an object from different directions are mapped in the inferotemporal cortex; and the position of a sound source as inferred from time delays between the two ears (interaural time delay) is mapped in the nucleus laminaris in birds. In motor maps, nearness in the map may reflect muscle position in the body but also the position of motor targets as in the map of saccade targets in the superior colliculus.

Second, connectivity within a map is mostly local, as in the lateral inhibition network discussed in Sect. 2.2.2; in the neocortex, it extends to about one millimeter. In a first approximation, connectivity may be considered shift invariant,, but it also includes so-called patchy connections, for example, between neurons encoding the same edge orientation in adjacent hypercolumns of the visual cortex. The connectivity range of 1 mm corresponds to the diameter of the dendritic tree of a pyramidal cell and is also the range of adult map reorganization as discussed in Sect. 6.3.4.

Third, connections between mapped areas tend to be "neighborhood-preserving," that is, axons originating from neighboring neurons in one area tend to connect to neighboring neurons in the goal area. These connections may go over long distances and are generally myelinated.

Neural maps are two dimensional, as are the cortical structures in which they are usually found. This does not mean, however, that the mapped parameter vector has to be two-dimensional as well. In the case of tonotopic maps, for example, the main mapped parameter is pitch: that is, a one-dimensional quantity corresponding also to the nearness to the oval window in the basilar membrane. On the other hand, the map of the primary visual cortex (area V1) encodes up to seven independent parameters, including visual field azimuth and elevation but also ocularity, edge orientation, motion direction, spatial frequency, and color; the full tuning curve of each neuron therefore is a function of seven variables. The mapping can thus be thought of as a map from $\mathbb{R}^7$ into $\mathbb{R}^2$. Clearly, such mappings cannot be globally continuous but require that local maps of some parameters (e.g., orientation) be "intercalated" into the overall map of visual field position, as is the case in the columnar systems for orientation and ocularity in the visual cortex. One ordering principle for intercalated or "dimension-reduction" mappings is to minimize the total cable length needed to connect neurons that encode parameters which are similar in one of the seven dimensions. Let us denote by $\mathbf{x}_i \in \mathbb{R}^7$ the vector of preferred stimuli of neuron i, by $N_i = \{j \,|\, \|\mathbf{x}_i - \mathbf{x}_j\| \leq 1\}$ its neighborhood in parameter space, and by $\mathbf{y}_i \in \mathbb{R}^2$ the neuron's position in the map. The optimal mapping would then minimize the total cable length given as

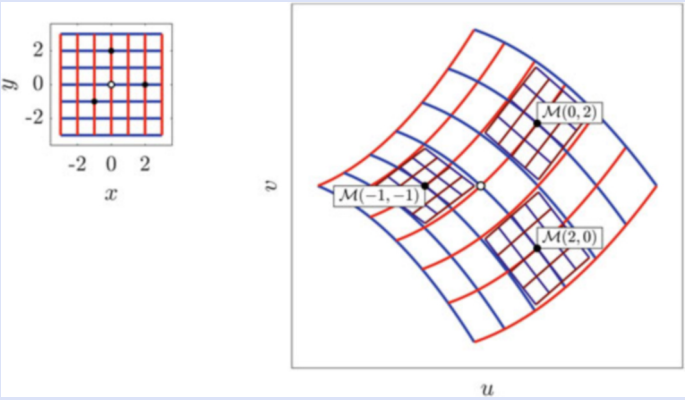
$$\sum_i \sum_{j \in N_i} \|\mathbf{y}_i - \mathbf{y}_j\|. \tag{7.26}$$

Durbin and Mitchison (1990) show that this criterion is indeed preserved in the dimension-reduction map for orientation and visual field position in V1.

The mechanisms underlying formation of local and intercalated maps on the one hand and long-distance neighborhood-preserving maps on the other may be different. Local maps can emerge from various forms of competitive learning as has been discussed in Sect. 6.3.3. Long-distance maps are formed in early ontogeny based on developmental mechanisms such as axon guidance by molecular markers (see, for example, Huberman et al. 2008). We now turn to some properties of the large-scale area-to-area maps realized by myelinated inter-area wiring.

▶ **Key Point: Neural Maps** In neural maps, neighboring neurons tend to have overlapping tuning curves and interact via local connectivity. Long-range connectivity between mapped areas is often neighborhood preserving, i.e., the projections from nearby neurons target nearby neurons in the goal area.

Box 7.2 Vector-Valued Functions and Areal Magnification



A point-to-point mapping is a function

$$\mathcal{M} : \mathbb{R}^2 \to \mathbb{R}^2 \ , \quad (u, v) = \mathcal{M}(x, y) = (\mathcal{M}_1(x, y), \mathcal{M}_2(x, y))$$

that can be visualized as a coordinate grid in the domain of $\mathcal{M}$ (left part of figure) and its distorted image in the range of $\mathcal{M}$ (right part). It may therefore be called a "rubber sheet transformation," in which straight lines are transformed into curves. Just as one-dimensional functions are locally approximated by their tangents, mappings can be locally approximated by linear functions, given by the Jacobian

$$\mathbf{J}_{\mathcal{M}}(x, y) = \begin{pmatrix} \frac{\partial}{\partial x}\mathcal{M}_1(x, y), & \frac{\partial}{\partial y}\mathcal{M}_1(x, y) \\ \frac{\partial}{\partial x}\mathcal{M}_2(x, y), & \frac{\partial}{\partial y}\mathcal{M}_2(x, y) \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial x}, & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x}, & \frac{\partial v}{\partial y} \end{pmatrix}.$$

For each point $\mathbf{x} = (x, y)$, $\mathbf{J}_{\mathcal{M}}$ is a square matrix that can be used to approximate $\mathcal{M}$ by the first-order Taylor expansion $\mathcal{M}(x + \Delta x, y + \Delta y)) \approx \mathcal{M}(x, y) + \mathbf{J}_{\mathcal{M}}(x, y) \cdot (\Delta x, \Delta y)^\top$. The figure shows these approximations for three points. The columns of $\mathbf{J}_{\mathcal{M}}$ are the tangent vectors to the "iso-x" and "iso-y" curves shown in red and blue; the rows are the gradients of the component functions $\mathcal{M}_1$ and $\mathcal{M}_2$, see Box 5.4. The area of the curvilinear squares in the (u, v)-plane changes with their position. It can be calculated by integrating the absolute value of the determinant, $|\det \mathbf{J}_{\mathcal{M}}(x, y)|$, over the corresponding squares in the (x, y)-plane. $M_a(x, y) := |\det \mathbf{J}_{\mathcal{M}}(x, y)|$ is known as the areal magnification factor.

(continued)

Box 7.2 (continued)
Angles between straight lines in (x, y)-space are preserved in the image if and only if the so-called Cauchy–Riemann equations hold:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad \text{and} \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}.$$

The Jacobian then takes the form $\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$: that is, a combined rotation and scaling. Mappings satisfying this condition are called "conformal"; the example shown in the figure does not satisfy the conformality condition.

7.3.2 Retinotopic Maps in the Visual Cortex

In the visual cortex, information from different parts of the retina is systematically ordered, resulting in a set of retinotopic maps or retinotopic representations. The cortex of primates has a large number (>20) of areas each one of which contains a more or less complete retinotopic image of the retina (Felleman and Van Essen 1991; Sereno et al. 2022). Note that with the incomplete chiasm of the mammals, only one half of the visual field is represented in each hemisphere. More specifically, the maps of each hemisphere show the contralateral half of the visual field and end at the vertical midline or vertical meridian.

The geometry of retinotopic maps was first studied in humans by the correlation of lesions of the visual cortex and the associated visual field deficits. Horton and Hoyt (1991) summarize this work in a map of the visual areas V1, V2, and V3. It shows a marked decrease of cortical magnification with visual field eccentricity and a compression of the polar angle at larger eccentricities such that the images of the upper and lower part of the vertical meridian become roughly parallel. It also shows the "mirror-belt" structure of the V2 map which follows the V1-border along the vertical meridian. When moving across this border, the direction of visual field progression reverts. The representation of the peripheral part of the horizontal meridian in area V2 is split into an upper and a lower branch and forms the better part of the outer border of the "belt" (for a schema, see Fig. 7.11c). Area V3 forms a second belt along the representation of the split horizontal meridian in V2. This overall structure is in agreement with single-cell recordings from primates (Daniel and Whitteridge 1961) and cats (Tusa et al. 1979). In fact, the mirror-belt organization of the V1-V2 complex is present in all mammals and seems to be a homologous trait inherited from their last common ancestor (Rosa and Krubitzer 1999). This may mean that it evolved together with the retinal projection to the thalamus and neocortex, while further visual areas seem to be later additions. The functional relevance of the mirror-belt organization, however, remains largely unknown.

7.3.3 Mathematical Descriptions of Retinotopic Maps

A mapping between two neural sheets with coordinates (x, y) and (u, v) can be treated as a vector-valued function

$$\mathcal{R} : \mathbb{R}^2 \to \mathbb{R}^2, \quad (u, v) := \mathcal{R}(x, y), \tag{7.27}$$

see Box 7.2. For this introduction, we assume that the mapping is "conformal": that is, that it preserves local angles and maps small circles to circles, not to ellipses. In this case, it is convenient to think of the mapping as a complex function with variable $z = x + iy$ and value $w = u + iv$, where i is the imaginary unit and $z, w \in \mathbb{C}$ (see Sect. 4.2.3). When treated as complex functions, the set of conformal mappings equals the set of complex differentiable, or "analytic," functions.

Areal Magnification

An important issue in retinotopic mapping is areal magnification, i.e., the representational area that is devoted in the cortex to each patch of visual field or retina. If mappings are modeled as vector-valued function $\mathbb{R}^2 \to \mathbb{R}^2$, areal magnification equals the absolute value of the determinant of the Jacobian of this function (see Box 7.2). In the conformal case, i.e., when the map is considered a complex differentiable functions $\mathcal{R} : \mathbb{C} \to \mathbb{C}$, the areal magnification is simply

$$M_a(z) = |\mathcal{R}'_{\mathbb{C}}(z)|^2 \tag{7.28}$$

where $\mathcal{R}'_{\mathbb{C}}$ is the (complex) derivative of $\mathcal{R}_{\mathbb{C}}$. Both definitions are equivalent by virtue of the Cauchy–Riemann equations (Box 7.2).

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 across the retina but decreases markedly from the center to the periphery. 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; the areal magnification would then be proportional to the ganglion cell density. In the primate retina, the distribution of ganglion cells has a peak around the fovea and declines toward the periphery in a more or less isotropic 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.

Let d_g denote the local density of retinal ganglion cells; in primates, it decreases approximately with the squared distance from the fovea, $d_g(z) = c_1|z|^{-2}$. With the assumption of equal representational area, $d_g(z) = c_2 M_a(z)$, we obtain the

differential equation[9]

$$|\mathcal{R}'_{\mathbb{C}}(z)|^2 = \frac{c_1}{c_2}\,\frac{1}{|z|^2}. \tag{7.29}$$

Along the positive x-axis (horizontal meridian of the retina), we have $y = 0$ and $z = x \in \mathbb{R}$. Equation 7.29 then reduces to $\mathcal{R}'(x) = c/x$ with $c = c_1/c_2$, which has the solution $\mathcal{R}(x) = c \log x$. Here, log denotes the natural logarithm. A full map satisfying Eq. 7.29 can be obtained from the complex logarithm: that is, by simply writing $\mathcal{R}_{\mathbb{C}}(x+iy) = c \log(x+iy)$ also in the complex plane (Fischer 1973; Schwartz 1977). As will be shown in the next paragraph, the complex logarithm, also known as the log-polar mapping, captures many features of the primate V1 map.

Log-Polar Mapping

As a consequence of the polar version of Euler's formula (Eq. 4.20), the complex logarithm can be described as a transformation to polar coordinates combined with a logarithmic compression of the radius. With the notation $x + iy = r \exp\{i\phi\}$, we have:

$$\begin{aligned} u &= \Re(\log(x+iy)) = \tfrac{1}{2}\log(x^2+y^2) = \log r \\ v &= \Im(\log(x+iy)) = \tan^{-1}(y/x) + n\pi = \phi, \end{aligned} \tag{7.30}$$

where $n \in \{-1, 0, 1\}$ is chosen as in the definition of the arg-function, Eq. 4.16.

The log function approaches $-\infty$ for $r = 0$ which corresponds to our choice of the density function of the retinal ganglion cells, which approaches $+\infty$ at $r = 0$. This would mean that the fovea representation requires an infinite area. The problem can be mended by adding a small real constant in the argument of the log function, leading to a map of the form $\mathcal{R}_{\mathbb{C}}(z) = \log(1+z)$; see Schwartz (1977).

Figure 7.11 illustrates the resulting maps of the primate visual cortex. Figure 7.11a shows the right part of the visual field as seen from the right eye. (The left eye's sight to the right is somewhat limited, leaving a "monocular crescent" in the right periphery which is not marked in the figure.) This field is mapped to the left visual cortex, area V1, in a way that can be fitted by the $\log(1+z)$-function discussed above (Fig. 7.11b). It is a reasonable approximation of the maps published by Horton and Hoyt (1991) and Daniel and Whitteridge (1961). Figure 7.11c shows the modified map $\log(1+\sqrt{z+x_b})$, which also produces a reasonable fit of the V1 map. In addition, it allows to model the basic structure of the V2 map (green area) if $z = x + iy$ is replaced by $z' = -x + iy$. The resulting "negative branch" of the

[9] If we write $\mathcal{R}_{\mathbb{C}} = u + iv$ and observe the Cauchy–Riemann equations $u_x = v_y$ and $u_y = -v_x$, Eq. 7.29 takes the form $u_x^2 + u_y^2 = d_g$ which is known as the "eiconal equation" in mathematical physics.

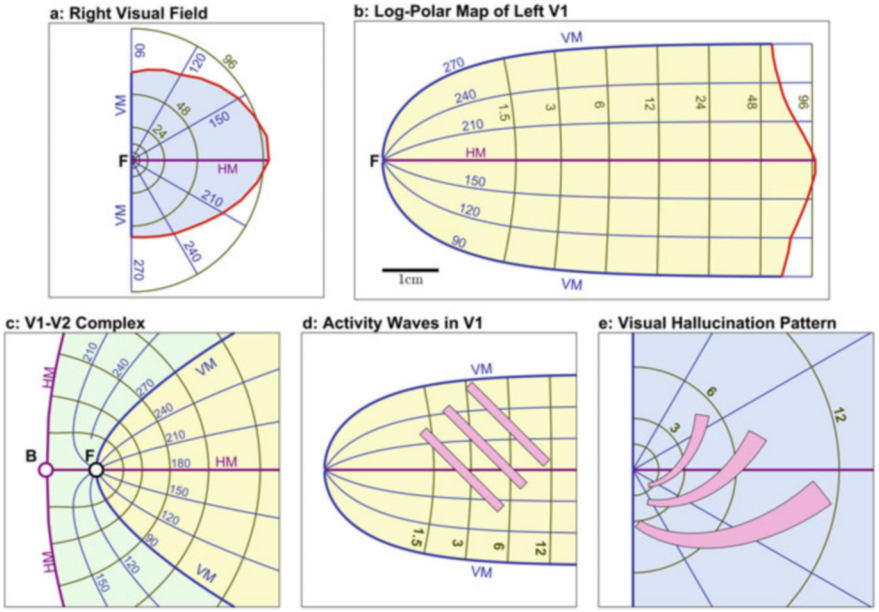


Fig. 7.11 Log-polar mapping of the visual hemifield. (**a**) Right (contralateral) visual hemifield; $F = (0, 0)$ marks the center of the fovea. The red line marks the perimeter or visual field margin. Radii and iso-eccentricity circles are labeled in degrees. HM and VM denote the horizontal and vertical meridian, respectively. (**b**) Image of the polar grid under the (suitably scaled) log-polar map. Scalebar is 1cm of cortical distance. (**c**) Model of the V1-V2 complex using a modified mapping function (see text). Area V2 shown in green. Note the bifurcation of the HM representation at point B and the reversal of the visual field map at the V1-V2 border which represents the vertical meridian. (**d**) Schematic plot of straight activity waves on V1. (**e**) When interpreted as stimuli in the visual field, the activity waves are perceived as spiral light patterns

map shows the bifurcation of the representation of the horizontal meridian at point B (representing the retinal point $(x_b, 0)$) and the mirror-belt organization of the V1-V2 complex. For better fits and applications to other species, see Schwartz (1980), Mallot (1985), and Polimeni et al. (2006).

The log-polar map as shown in Fig. 7.11b has also been linked to perceptual phenomena such as the migraine aura or other types of visual hallucinations (Bressloff et al. 2001). Such perceptions are thought to arise from activity patterns in the visual cortex which are not generated by sensory input but by internal processes of the brain. Assume that a straight linear wavefront of activity would be moving across area V1 (pink stripes in Fig. 7.11d). Following Helmholtz' rule that the object we perceive in the presence of a particular neural "impression" is that which would generate the same neural impression, were it actually present in the visual field (Helmholtz 1867, p. 428), we can predict the hallucinations resulting from straight activity waves. The result is a logarithmic spiral: that is, the image of the stripes under the inverse of the log-polar map (Fig.7.11e). In the migraine aura, spiral

phosphenes are indeed perceived, usually drifting from the center of sight to the periphery and thereby increasing in size. The spiral frontline is seen as a zigzagging, scintillating, and high-contrast "fortification pattern" which, according to Dahlem et al. (2000), may result from the activation of nearby neurons tuned to different edge orientations. It would thus reflect the "intercalated" structure of the V1 map with orientation hypercolumns nested into the overall map of visual position.

► **Key Point: Log-Polar Map** The macroscopic maps of the V1-V2 complex can be described by variants of the complex logarithm, or log-polar mapping. The variation of areal magnification across the visual field reflects the density of retinal ganglion cells. Log-polar mapping explains the spiral shape of a class of visual hallucinations.

7.3.4 Functional Relevance

The combination of population coding, neural mapping, and dynamic "bumps" of neural activity (continuous-field attractors, see Sect. 6.5) forming on these maps defines an approach to neural computation which is quite different from the ideas of classical neural network theory (e.g., the perceptron) or indeed of information processing by logical neurons (see Box 2.6). It can be characterized as a signal-flow approach to information processing which sees the brain not so much as a classification device but as part of a complex control system coordinating sensory input and motor output: that is, as the organizer of the action–perception loop.

While perceptron-like theories emphasize synaptic weight change and stepwise learning in discrete time, signal-flow in neural maps focuses also on spatiotemporally continuous activation dynamics. The best-studied system operating along these principles is probably the hippocampal system for space (McNaughton et al. 2006) but applications to working memory in general are becoming standard (Zylberberg and Strowbridge 2017). While physiological theories of neural computation already make extensive use of signal-flow models, their application in artificial neural networks is still in its beginnings.

7.4 Summary and Further Reading

1. Information is coded in the brain by the specificity of active neurons. Specificity is described by tuning curves. In a space of represented parameters, the tuning curve is the conditional probability of neuronal firing given the parameter value.
2. Populations of neurons with overlapping tuning curves constitute a population code. The overlap allows for a more efficient encoding of information, sub-pixel resolution, working range adjustment, interpolation, and mental rotation.

3. Information theory provides a tool for judging the efficiency of neural codes. It shows that overlapping and graded tuning curves are superior to nonoverlapping box-shaped tuning curves, i.e., to simple discretization of the parameter space.
4. The specificity of each neuron and the information represented in a population code can be decoded by various procedures, including center-of-gravity, maximum likelihood, and Bayesian estimators.
5. Neurons with similar specificities (overlapping tuning curves) tend to be located close to each other, thereby forming orderly maps of the encoded parameters.
6. The largest neural map is the retinotopic map of visual area V1. Its geometry follows the complex logarithmic function. Mathematically related descriptions can be given for the adjacent area V2.
7. The dynamics of localized activity "bumps" moving on neural maps constitutes a powerful principle of neural computation.

Texts

Cover and Thomas (2006): *Comprehensive text on information theory.*
Dayan and Abbott (2001): *Chapters on information theory and decoding.*
Kriegeskorte and Kreiman (2012): *Edited collection covering the principles and many examples of population coding in the visual system.*
Rieke et al. (1996): *Rigorous treatment of neural coding based on single spikes and spike-trains.*

Suggested Original Papers for Classroom Seminars

Bressloff et al. (2001): *Hallucination pattern are explained as simple waves of activity on the cortex, interpreted by the observer by "backward projection" with the retinotopic mapping function, see Fig. 7.11d, e.*
Georgopoulos et al. (1993): *Demonstration of population coding in the motor cortex. This chapter popularized the ideas of population coding and the center-of-gravity estimator, for which the authors introduced the term "population vector".*
Poggio et al. (1992): *This chapter 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.*
Wilson and McNaughton (1993): *Classical paper on population coding in hippocampal place cells. After recording the positional tuning curves of a set of place cells, ongoing activity is used to predict the current position of a rat in its maze.*

References

Albrecht, D.G., W.S. Geisler, R.A. Frazor, and A. M. Crane. 2002. Visual cortex neurons of monkeys and cats: Temporal dynamics of the contrast response function. *Journal of Neurophysiology* 88: 888–913.

Barlow, H.B. 1972. Single units and sensation: A neuron doctrine for perceptual psychology? *Perception* 1: 371–394.

Blakemore, C., and P. Sutton. 1969. Size adaptation: A new aftereffect. *Science* 166: 245–247.

Borst, A., and F.E. Theunissen. 1999. Information theory and neural coding. *Nature Neuroscience* 2: 947–957.

Braddick, O., F.W. Campbell, and J. Atkinson. 1978. Channels in vision: Basic aspects. In *Perception. Handbook of Sensory Physiology VIII*, ed. R. Held, H.W. Leibowitz, and H.-L. Teuber. Berlin: Springer Verlag.

Bressloff, P.C., J.D. Cowan, M. Golubitsky, P.J. Thomas, and M.C. Wiener. 2001. Geometric visual hallucinations, Euclidean symmetry and the functional architecture of striate cortex. *Philosophical Transactions of the Royal Society B* 356: 299–330.

Cover, T.M., and J.A. Thomas. 2006. *Elements of Information Theory*. 2nd ed. New York: Wiley.

Dahlem, M.A., R. Engelmann, S. Löwel, and S.C. Müller. 2000. Does the migraine aura reflect cortical organization? *European Journal of Neuroscience* 12: 767–770.

Daniel, P.M., and D. Whitteridge. 1961. The representation of the visual field on the cerebral cortex in monkeys. *Journal of Physiology* 159: 203–221.

Dayan, P., and L.F. Abbott. 2001. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge: The MIT Press.

Durbin, R., and G. Mitchison. 1990. A dimension reduction framework for understanding cortical maps. *Nature* 343: 644–647.

Felleman, D.J., and D.C. Van Essen. 1991. Distributed hierarchical processing in the primate visual cortex. *Cerebral Cortex* 1: 1–47.

Fischer, B. 1973. Overlap of receptive field centers and the representation of the visual field in the optic tract. *Vision Research* 13: 2113–2120.

Georgopoulos, A.P., J.F. Kalasak, R. Caminiti, and J.T. Massey. 1982. On the relation of the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience* 2: 1527–1537.

Georgopoulos, A.P., M. Taira, and A. Lukashin. 1993. Cognitive neurophysiology of the motor cortex. *Science* 260: 47–52.

Grothe, B., M. Pecka, and D. McAlpine. 2010. Mechanisms of sound localization in mammals. *Physiological Reviews* 90: 983–1012.

Hafting, T., M. Fyhn, S. Molden, M.-B. Moser, and E.I. Moser. 2005. Microstructure of a spatial map in the entorhinal cortex. *Nature* 436: 801–806.

Hebb, D.O. 1949. *The Organization of Behaviour*. New York: Wiley.

Heil, P., and A.J. Peterson. 2015. Basic response properties of auditory nerve fibers: A review. *Cell and Tissue Research* 361: 129–158.

Helmholtz, H. von. 1867. Von den Wahrnehmungen im Allgemeinen. In *Handbuch der physiologischen Optik*, Chapter 26. Leipzig: Voss.

Horton, J.C., and W.F. Hoyt. 1991. The representation of the visual-field in the human striate cortex – A revision of the classic Holmes map. *Archives of Ophthalmology* 109: 816–824.

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

Huberman, A.D., M.B. Feller, and B. Chapman. 2008. Mechanisms underlying development of visual maps and receptive fields. *Annual Review of Neuroscience* 31: 479–509.

Kriegeskorte, N., and G. Kreiman. 2012. *Visual Population Codes. Towards a Common Multivariate Framework for Cell Recording and Functional Imaging*. Cambridge: The MIT Press.

Laughlin, S.B. 2011. Energy, information, and the work of the brain. In *Work meets life: Exploring the integrative study of work in living systems*, ed. R. Levin, S. Laughlin, C. De La Roche, and A.F. Blackwell. Chapter 2, pp. 39–67. Cambridge: The MIT Press.

Leopold, D.A., A.J. O'Toole, T. Vetter, and V. Blanz. 2001. Prototype-referenced shape encoding revealed by high-level aftereffects. *Nature Neuroscience* 4: 89–94.

Logothetis, N.K., J. Pauls, H.H. Bülthoff, and T. Poggio. 1994. View-dependent object recognition by monkeys. *Current Biology* 4: 401–414.

Logothetis, N.K., J. Pauls, M. Augath, T. Trinath, and A. Oeltermann. 2001. Neurophyiological investigation of the basis of the fMRI signal. *Nature* 412: 150–157.

Mallot, H.A. 1985. An overall description of retinotopic mapping in the cat's visual cortex areas 17, 18, and 19. *Biological Cybernetics* 52: 45–51.

Mallot, H.A. 2023. *From Geometry to Behavior. An Introduction to Spatial Cognition*. Cambridge: The MIT Press.

McNaughton, B.L., F.P. Battaglia, O. Jensen, E.I. Moser, and M.-B. Moser. 2006. Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience* 7: 663–678.

Montemurro, M.A., M.J. Rasch, Y. Murayama, N.K. Logothetis, and S. Panzeri. 2008. Phase-of-firing coding of natural visual stimuli in primary visual cortex. *Current Biology* 18: 375–380.

Moore, B.C.J. 2013. *An Introduction to the Psychology of Hearing*. 6th ed. Leiden: Brill.

Müller, J.P. 1837. *Handbuch der Physiologie des Menschen für Vorlesungen*. Vol. II. Coblenz: J. Hölscher.

Ohlemiller, K.K., and S.M. Echteler. 1990. Functional correlates of characteristic frequency in single cochelar nerve fibers of the Mongolian gerbil. *Journal of Comparative Physiology A* 167: 329–338.

O'Keefe, J., N. Burgess, J.G. Donnett, K.J. Jeffery, and E.A. Maguire. 1998. Place cells, navigational accuracy, and the human hippocampus. *Philosophical Transactions of the Royal Society B* 353: 1333–1349.

Palm, G., A. Knoblauch, F. Hauser, and A. Schüz. 2014. Cell assemblies in the cerebral cortex. *Biological Cybernetics* 108: 559–572.

Poggio, T., and F. Girosi. 1990. Networks for approximation and learning. *Proceedings of the IEEE* 78: 1481–1497.

Poggio, G.F., F. Gonzalez, and F. Krause. 1988. Stereoscopic mechanisms in monkey visual cortex: Binocular correlation and disparity selectivity. *The Journal of Neurocience* 8: 4531–4550.

Poggio, T., M. Fahle, and S. Edelman. 1992. Fast perceptural learning in visual hyperacuity. *Science* 256: 1018–1021.

Polimeni, J.R., M. Balasubramanian, and E.L. Schwartz. 2006. Multi-area visuotopic map complexes in macaque striate and extra-striate cortex. *Vision Research* 45: 3336–3359.

Pouget, A., P. Dayan, and R. Zemel. 2000. Information processing with population codes. *Nature Reviews Neuroscience* 1: 125–132.

Riehle, A., S. Grün, M. Diesmann, and A. Aertsen. 1997. Spike synchronization and rate modulation differentially involved in motor cortical function. *Science* 278: 1950–1953.

Rieke, F., D. Warland, R. de Ruyter van Steveninck, and W. Bialek. 1996. *Spikes. Exploring the Neural Code*. Cambridge: The MIT Press.

Rosa, M.G.P., and L.A. Krubitzer. 1999. The evolution of visual cortex: Where is V2? *Trends in Neurosciences* 22: 242–248.

Schwartz, E.L. 1977. Spatial mapping in primate sensory projection: Analytic structure and relevance to perception. *Biological Cybernetics* 25: 181–194.

Schwartz, E.L. 1980. Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding. *Vision Research* 20: 645–669.

Sereno, M.I., M.R. Sood, and R.-S. Huang. 2022. Topological maps and brain computations from low to high. *Frontiers in Systems Neuroscience* 16: 787737.

Seung, H.S., and H. Sompolinsky. 1993. Simple models for reading neuronal population codes. *Proceedings of the National Academy of Sciences* 90: 10749–10753.

Shannon, C.E., and W. Weaver. 1949. *The Mathematical Theory of Communication*. Urbana: The University of Illinois Press.

Shao, S., M. Meister, and J. Gjorgjieva. 2023. Efficient population coding of sensory stimuli. *Physical Review Research* 5: 043205, 1–19.

Shepard, R.N., and J. Metzler. 1971. Mental rotation of three-dimensional objects. *Science* 171: 701–703.

Stone, J.V. 2015. *Information Theory: A Tutorial Introduction*. Sebtel Press.

Thompson, P., and D. Burr. 2009. Visual aftereffects. *Current Biology* 19: R11–R14.

Tusa, R.J., A.C. Rosenquist, and L.A. Palmer. 1979. Retinotopic organization of areas 18 and 19 in the cat. *Journal of Comparative Neurology* 185: 657–678.

Wang, X.-J. 2010. Neurphysiological and computational principles of cortical rhythms in cognition. *Physiological Reviews* 90: 1195–1268.

Wilson, M.A., and B.L. McNaughton. 1993. Dynamics of the hippocampal ensemble code for space. *Science* 261: 1055–1058.

Zylberberg, J., and B.W. Strowbridge. 2017. Mechanisms of persistent activity in cortical circuits: Possible neural substrates for working memory. *Annual Review of Neuroscience* 40: 603–627.

Index

H. A. Mallot, *Computational Neuroscience*,
https://doi.org/10.1007/978-3-031-75705-1