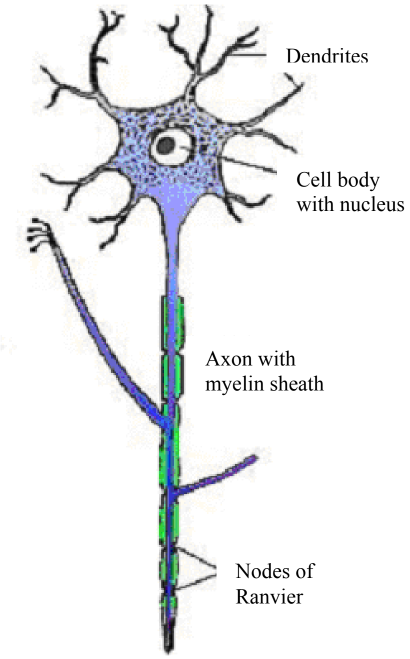


Principles of Neural Networks

A Simple Introduction

Dr. Norbert Waleschkowski
FH Darmstadt
University of Applied Sciences
Department of Computer Science

Excerpt from Lectures on
“Neural Networks and Applications”
Darmstadt, Summer Term 2000

Table of Content

Table of Figures

1 Introduction and Motivation

Have a look at figure 1. What does it show? It is just a picture consisting of black spots on a white background. After a while the shape of a dog (Dalmatian) is silhouetted against the background. There is no line or conture resp. in the picture but one sees the silhouette of the dog clearly. A spatial impression arises. Humans recognize the dog without any problem. But today‘s computers are unable to detect it. They can‘t tell a spot on the coat of the dog from a spot which is a simple distraction. Where do these impressions come from?

Figure 1.1: Dalmatian in a park landscape

The corresponding cognitive visual process represents an exceptional performance of human brains. Think of having turned the picture upside down before presentation. Probably, it would have been much more difficult, may be almost impossible for a human to recognize the dog now. Structures well known to humans are obviously a necessary assumption for the recognition process that happens to take place in the brain. Figure 2 shows another example for the phenomenon of human process of recognition.

How are such pictures represented in the brain? Which kind of information is stored there? Und how does the recognition process take place in detail? Such questions arise for other cognitive human capabilities like speech recognition mutatis mutandis.

In strong contrast to such human capabilities computer based visual recognition systems of today are absolutely unable to interpret such a picture.

Confronting capabilities of human brains with computers pinpoints that artificial systems fail where humans brains solve problems easily and without any problem and vice versa. Modern computers can solve problems easily which are extremely difficult resp. impossible for humans. Just consider their incredible computing power or their ability to master gigantic data sets and so on. One modern supercomputer has a computing power which exceeds the computing capability of the whole population of the world. Such a computer performs much more than 5 billion computing steps per second and takes it on with the population of the whole world. Obviously, biological brains on the one hand and electronic computers on the other hand seem to have virtually orthogonal capabilities.

The black and white horses on this picture of Bew Doolittle seem to mute to brindle horses in a partially snowy meadow - a mysterious capability of our brain. (Singer, Wolf: Auf dem Weg nach innen, FAZ newspaper from 27.2.1998, Nr.49, p. 41)

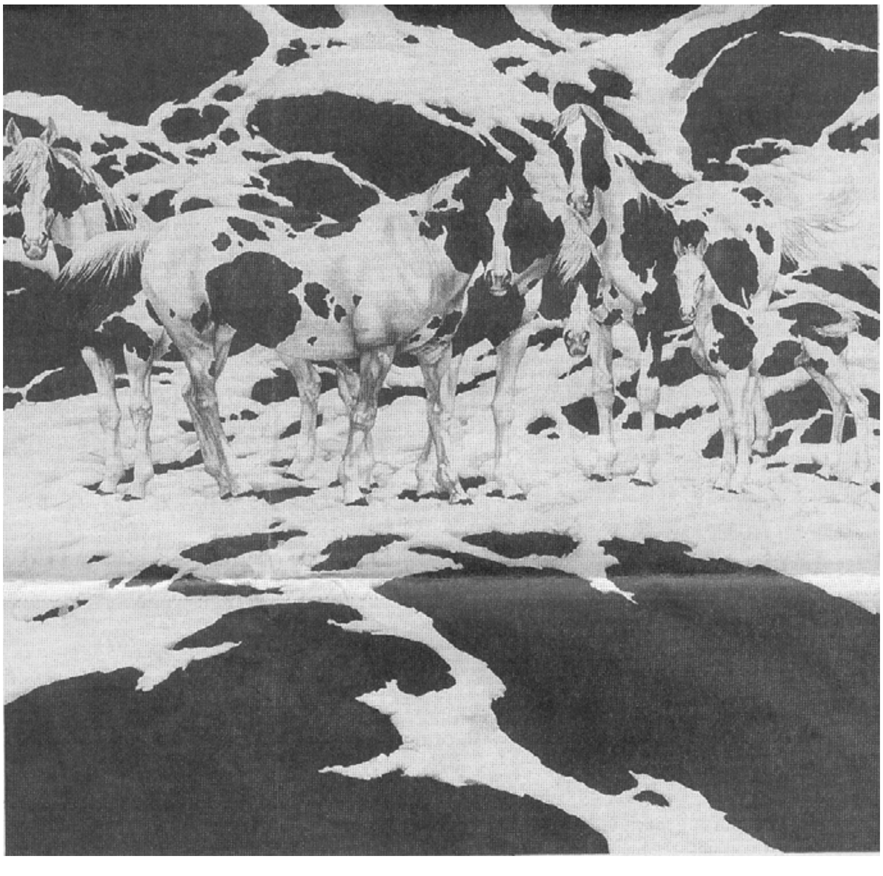


Figure 1.2: Black and white horses in a snowy landscape

2 Cognition and Representation

Cognitive processes and mental representations are an interesting field of research for philosophers, linguists, psychologists as well as for computer scientists. However, each discipline investigates the interconnection of mind and matter typically within its own scientific context. In humanities and (natural) sciences every object under research is investigated within its specific framework and its corresponding boundaries. Each discipline has its own viewpoints and perspectives. Interdisciplinary approaches try to interconnect the different specific research fields in order to achieve continuative results with regard to consciousness, cognition and mind.

Cognitive sciences essentially try to investigate cognitive processes and structures of human mind. In this context, the mind is the entirety of all (possible) states as there are thoughts, ideas, memories, desires, intentions and feelings. The concept of mental representation plays an essential role. In cognitive psychology this term refers to system internal states which map and represent, respectively, system external states, e.g. real world objects.

The concept of mind is discussed on various levels. In its core the intention of the philosophy of mind is made up of explaining the existence of mind in an entirely material physical world. It has been proven that there is a strong correlation between mental events and neural processes. Thus mental and psychological activities are closely coupled to material substrates. As a consequence, the relationship between mind and matter is not only an object of investigation within philosophy but also a centrical topic within brain science. A reductionistical approach which equates mental processes with neural ones has led to a high degree of contradiction among experts. The question arises what this correlation could tell us about the true nature of the relationship between both phenomena as well as about the nature of mind. The deeper reason for this controversial discussion – which clearly indicates the existing cleft between humanities and sciences - is based upon the different objectives of both disciplines. While philosophy tries to understand and conceive the essence of things neurobiology has the goal to identify and understand the laws and rules of relationships between observed events, e.g. the relationship between mental and neural processes. To what extent neurosciences can contribute to fundamental insights to the brain-mind problem beyond the level of confirmation of functional unity between mental and neural processes is a matter of further research. As a matter of principle the interaction of both specific expertise and discipline-related approaches seems to be essential for a better and further understanding of mind.

Also in computer science a discipline emerged in order to understand intelligent reasoning and behaviour and to simulate it by computational means and - in so doing - making it better available to scientific investigation.

This discipline is called Artificial Intelligence (AI). The goal of AI is to map cognitive processes to computers. The roots of AI go back into the late fifties of the 20th century. In 1956 Jim McCarthy created the term "Artificial Intelligence" on a conference in Dartmouth. This date is considered the birth of AI discipline. In fact it is often criticized that AI is unable in principle to contribute substantially to a better understanding of the concept of cognition because its models are strongly simplified and too far away from the biological archetype. However, the objective and vital goal of AI is to better understand the complex system of cognition.

3 Symbolic vs. Neural Representation

Today AI comes along with a bundle of different approaches, methodologies and technologies. In the early days Artificial Intelligence was considered in the narrow sense as representation and processing of symbolic information. Symbolic AI was based upon the assumption of imitating or simulating any intelligent operation only by means of symbols and symbol processing. Hence, cognition and perception could be modelled by symbolic representation. This approach was based upon two assumptions:

1st assumption of symbolic AI (Physical Symbol System Hypothesis)

The *„Physical Symbol Systems Hypothesis"* is a necessary and sufficient condition to

- explain intelligent human knowledge or action in a "representational" manner and to
- describe cognition as an architecture of syntactically connected symbols.

2nd Assumption of symbolic AI

All intelligent operations require substantial knowledge which is stored in the system and may be accessed if needed.

The semantic network shown in figure 2.1 is a simple example of symbolic knowledge representation. Concepts like apples, bananas, lemons and fruits each represented by a node (for an entity or a concept) and by edges (for a relationship resp. an association) which interconnect relevant nodes. Apples, bananas and lemons are fruits, which is represented by an "is-a"-edge. Fruits taste typically sweet. This is represented by a "taste"-edge between the nodes "fruit" and "sweet". This property of "fruit" is inherited to the nodes "apple" and "banana" along the "is-a"-edge. The exceptional fact that lemons are sour is represented by a separate "taste"-edge between "lemon" and "sour".

Thus the property “sweet” is overwritten by this specific edge expressing lemons being sour.

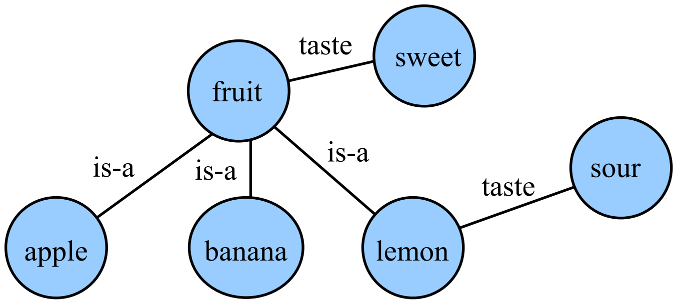


Figure 2.1: A simple semantic network

Symbol orientated knowledge processing may be executed according to the classical sequential concepts of computer architecture which is known as the so-called von-Neumann architecture. It is based on one or a few powerful processors which perform computing processes in a sequential manner. In the early days of electronic data processing such computers were also called “electronic brains”. The term “brain” should indicate some kind of analogy to biological brains. Today we know that there is almost no similarity between sequential computer models and biological brains.

Aside the main line of symbolic AI the field of connectionistic systems emerged, also called parallel distributed processing (PDP) or neurocomputing. At the beginning such systems have also been considered within the context of cybernetic systems. Cybernetics investigates the principles which apply both for technical systems as well as for living organisms. Its most important concepts are receipt, processing and transmission of information as well as certain control functions.

The connectionism is based upon the assumption that the organization and structure of the brain and its information processes is the decisive reason for its outstanding performance. The connectionism is considered to represent neurophysiological concepts based on continuous and a massive parallel system approach while symbol processing is considered to rely on psychological concepts based on a serial system approach. But it turned out that artificial “intelligent” systems failed when faced with problems that could easily be solved by humans. The crucial factor for this big gap is that typical computers systems are organized in a completely different manner from the principles of biological brains.

4 The Organization of the Brain

Because the basic structure of artificial neural networks is adopted to those of brains it is necessary to explain this basic structure of brains and some principles of its performance in some more detail.[1]

The human brain is the organ of thought, memory and emotion. It is also called the organ of mind. The brain is a complex network of nerve cells, which exchange signals with each other. The neocortex is responsible for the intelligent behaviour and occurs only in higher species. This part of the whole nervous system is phylogenically the youngest portion of the brain. A human brain can be divided into a left and a right hemisphere. The central fissure is the most important connection to enable communication between both hemispheres. Each hemisphere itself can be divided into four main lobes, which are forehead lobe (lobus frontalis), parietal lobe (lobus parietalis), temple lobe (lobus temporalis) and backhead lobe (lobus occipitalis). The brain is not a monolithic system, rather it consists of specialized areas to coordinate specific tasks as there are e.g. spatial reception, motor function, speech and language etc.. The main tasks of the brain encompass processing sensory information and storage of information as well as thinking, speaking, and learning.[2]

[1] Information concerning the organization and the structure of the brain are taken from John R. Anderson (1992) Kognitive Psychologie. Eine Einführung, pp. 35-39.

[2] Zimbardo, P. G. (1983). Psychologie, Heidelberg, p. 88.

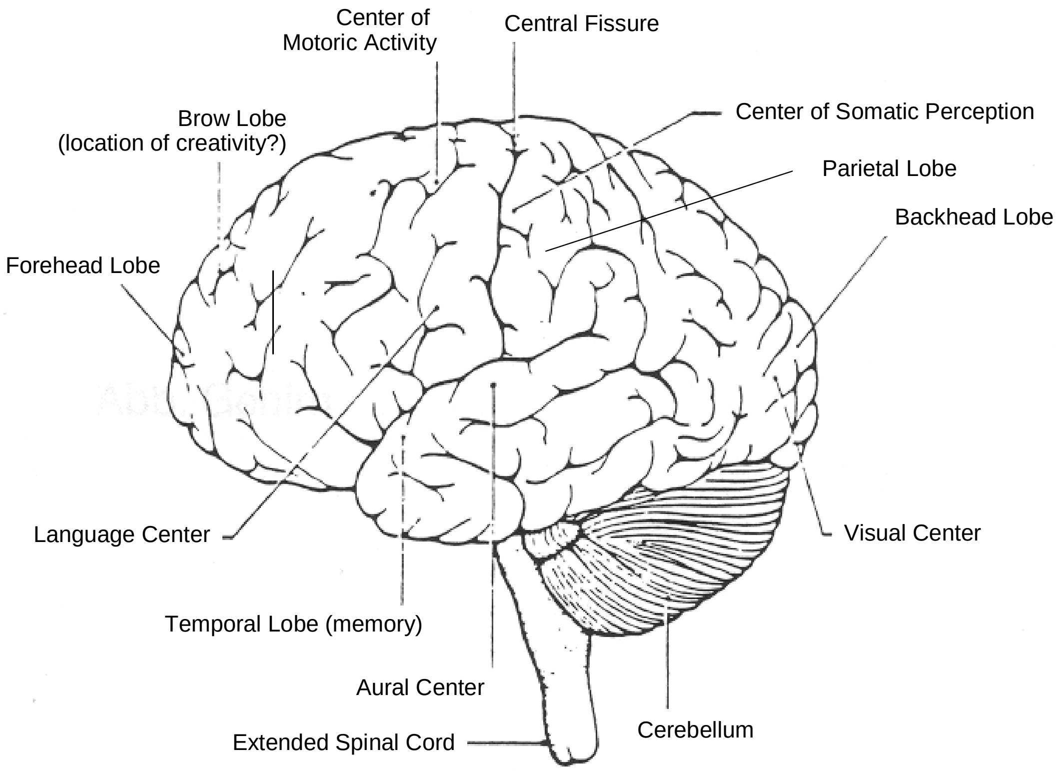


Figure 4.1: The human brain, seen from the left side

The cerebellum ("little brain") belongs phylogenically to the older portion of the brain. The extended spinal cord (medulla oblongata) controls basic activities as breathing, swallowing and beating of the heart. Basic motor functions as well as arbitrary movements are controlled by the cerebellum. The thalamus serves as an interface for motor and sensory information, which are transferred between lower brain areas and the cerebrum. The hypothalamus coordinates metabolism and body temperature as well as drives like hunger and thirst. The subcortical brain structures are closely associated to superior areas of the brain because only the proper interplay between all parts of the brain guarantees a smooth functioning of human body and mind in their entirety.

In contrast to many other mammals the human neocortex is highly developed. There are about 100 billion neurons within the neocortex. The neocortex itself looks like a soft duster that has been shoven together and has a corrugated appearance, which looks like a walnut. Like a paper-tissue it consists of several layers. It has a size of appr. 0.2 m^2 and a thickness of a few mm. Per mm^2 there are ca. 100,000 neurons, each of them are connected with up to a few 1,000 other neurons.[3] The number of connections (synapses) is about 10^{15} in total. Typically bigger portions of the neocortex are activated

[3] It is interesting aspect that the neocortex only fits into the cranium because of its many wrinkles and foldings thus reducing its volume. Regarding this, humans are different from many lower mammals.

simultaneously. Caused by this extraordinary high degree of interactive processing this type of computation is called to be massively parallel. Within a single neuron, information processing takes place by an ongoing modification of its electrical potential. The principles of neural information processing are discussed in more detail in the next chapter.

4.1 The Structure of a Neuron

In the brain there are (at least) two types of neurons, which are called nervous cells (neurons) and glia cells. The vast majority of cells are glia cells, which primarily have some kind of support or backing function. Whether glia cells contribute directly to neural processing is not completely clear up to now.[4] An undisputed fact is that neurons play a central role in cortical information processing. In fact there is a great number of neurons that differ with respect to size and form, but almost all cell types have a common basic structure. Figure 4.2 visualizes this structure of a typical neuron.

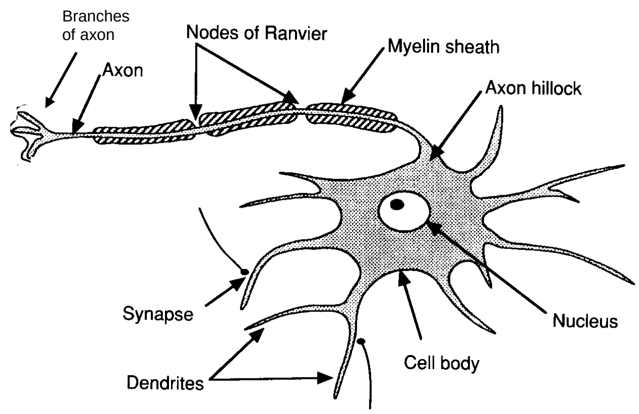


Figure 4.2: Scheme of a biological neuron

The graphic depicts the major components of a typical nerve cell. These components are dendrites, synapses, the cell body with the nucleus, and a single axon. The cell body of the neuron has a size of approximately 5 to 100 micrometers. The membrane of a neuron separates the intracellular plasma from the interstitial fluid external to the cell.

4.2 Principles of Neural Information Processing

Neurons communicate with each other by sending a signal (pulse) via an output cord which is called axon. The length of an axon varies from a few millimeters up to one

[4] Roth, Gerhard (2001) Fühlen, Denken, Handeln. Wie das Gehirn unser Verhalten steuert, Frankfurt, p. 100.

meter. There is only one axon per neuron but it may split up later into multiple cords. The axon is surrounded by the myelin sheath. So-called nodes of Ranvier interrupt the myelin sheath periodically along the length of the axon. The axon is not a very good conductor. From a certain threshold this results in an action potential spike at the first node of Ranvier following the principle “all or nothing”. As this node depolarizes, it triggers the depolarization of the next node of Ranvier and so on. The action potential will be transmitted along the complete axon. Once an action potential has passed a point, that point isn’t capable of being re-excited for about 1 msec. Thus the frequency of nerve-pulses is limited to about 1000 per second (which is very low compared to computer processors).

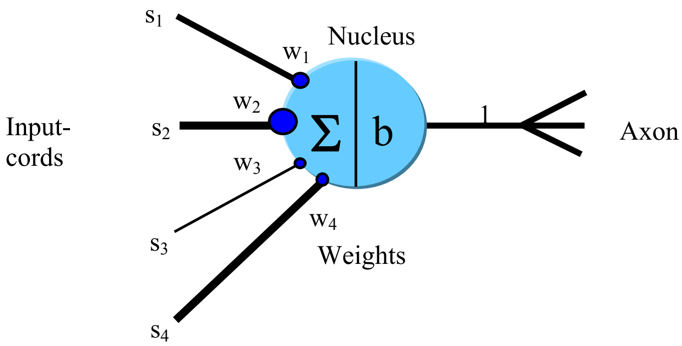
Synapses connect the axons of one neuron to various parts (dendrites) of other neurons which absorb the incoming pulses and transmit them into the cell body[5]. The activity that occurs at the connection between two neurons is called synaptic junction. Communication between neurons occurs as a result of the release of substances (neurotransmitters) by the presynaptic cell and of the subsequent absorbation of these substances by the postsynaptic cell. Those nerve pulses can result in local changes of the potential within the cell body of the receiving neuron. These potentials spread through the cell body. They can either be excitatory, i.e. the decrease the polarization of the cell, or inhibitory, i.e. they increase the polarization of the cell. The input potentials are summed up at the axon hillock. If the size of depolarisation is sufficient then an axon potential is generated running along the axon; the neuron “fires”, i.e. the decisions take place in the cell body.

All of the neural information processing described above happens by reception, processing and transmission of neuroelectrical and neurochemical currents. The ratio of voltage between the inner cell and the outer environment of the cell results in the emission of chemical substances. The excitability of a nerve cell is based on the fact that there is a potential difference of about 70mV across the cell membrane (inside negative) which is caused by an unequal concentration of sodium (Na+), potassium (K+), calcium (Ca2+), chlorine (Cl^-) ions and some other anions. Each of these ion distributions has a specific equilibrium potential. Outside the cell there are mainly sodium and chloride ions. Inside the cell there are mainly potassium ions. The potential difference then is caused by diffusion forces. Ions can diffuse across the cell membrane. The membrane is selectively more permeable to some ions rather than to others which finally causes the

[5] Following Roth (2001, p. 101) synapses may occur between axones and dendrites, axons and cell bodies, axons and axons and dendrites and dendrites.

polarization of the cell. A very important aspect of neural information processing is to integrate the incoming pulses. The integration happens due to the weight of the synapses which may have an excitatory or inhibitory effect which may change over time. Mainly it is this plasticity which constitutes the learning capability of biological brains.[6]

I want to demonstrate the principle functionality of a neuron considering as example a simple formal model of a neuron using some fictitious values incoming signals, synaptic weights and thresholds.



<u>Example:</u>

$s_1 = 1{,}0$	and	$w_1 = 1$	→ $a = 1$
$s_2 = 1{,}5$	and	$w_2 = 2$	→ $a = 3$
$s_3 = 0{,}5$	and	$w_3 = 0{,}5$	→ $a = 0{,}25$
$s_4 = 10$	and	$w_4 = 0{,}01$	→ $a = 0{,}1$

Total input = 13
Weighted Input = 4,35
Threshold = 3 → Neuron fires
Threshold = 5 → Neuron does not fire

Figure 4.3: An artificial neuron

The incoming signals s_1, s_2, s_3 and s_4 may have individual and different values. In this strongly simplified case the total input of the neuron has an amount of 13. Multiplying the incoming signals with the weights (strength of synapses) w_1, w_2, w_3, w_4 leads to a weighted total input of 4.35. In case of a threshold of 3 this would lead to the effect that the neuron will “fire”. In case of a threshold of 5 this would mean that the neuron would not fire. On the lowest level, thinking seems to be nothing more than simple addition. Based on this insight René Descartes’ famous dictum *I think therefore I am (Cogito ergo sum)* may be modified *I think therefore I sum up* [7] and be understood as a metaphor for intelligent capabilities of the brain on a neural level.

[6] Anderson, John R. (1992). Kognitive Psychologie, p. 31.

[7] Hofstadter, Douglas (1989). *Gödel, Escher, Bach. - Ein endloses geflochtenes Band*, p. 363.

4.3 Neural Assemblies

Neural assemblies resp. assemblies of neurons are responsible for the distributed representation of knowledge. The one-cell[8] doctrine of that a single neuron represents a certain term or concept had to be skipped due to a lack of evidence The principle of a grandmother cell which appears only to be activated if the grandmother appears in mind is considered a metaphor for local representation (z.B. Hofstadter 1989). In case of local representation biological brains would be extremely fault-prone. If a neuron drops out or dies off the corresponding concept would be irrevocably lost.

Numerous studies have led to the result that knowledge is not represented by single neurons but by groups of neurons which are called neuron assemblies. The activation of a single neuron does not matter, it is considered meaningful only in the context of a synchronized activation of multiple neurons. Thus a single neuron is not considered to represent a functional unit of thinking.

Groups of neurons which are often activated simultaneously are aligned to neuron collectives after all. These are called assemblies (e.g. Dorffner 1991). The formation of such assemblies may even embrace different areas of the brain. On a neural level such assemblies may be considered as the atoms of thinking. Concepts of the outer world are represented in form of such assemblies.[9] Conscious states like cognition, perception and memory are considered to be based upon the interplay of such assemblies. Following Singers oscillation theory such assemblies oscillate in-phase, i.e. they are activated simultaneously or not at all. Thus the representation of objects or concepts on a neural level seems to be an activated assembly. In a certain sense such assemblies could be considered the atoms of mind. Therefore, human thinking beyond this level does not exist.

Figure 4.4 visualizes the principle of distributed representation. It shows a set of neurons that represent four concepts: “gorilla”, “ape”, “chimpanzee” and “like-bananas”. Each concept is represented by multiple neurons indicated by a borderline around it. Some concepts overlap others, i.e. they share the same neurons. Overlapping regions represent concepts with some kind of semantic association in between. Notice that the concepts „chimpanzee“ and „gorilla“ overlap each other. The intersection could

---

[8] Barlowe, Horace (1972). Single units and sensations: a neuron doctrine for perceptual psychology?, In *Perception* 1, pp. 371-394.

[9] Engel, Andreas K., König, Peter (1998). Das neurologische Wahrnehmungsparadigma. Eine kritische Bestandsaufnahme, S. 167.

be considered the generalized concept of “ape”. Due to this fact of sharing neurons the learning process “a gorilla likes bananas” automatically extends its scope to the concept of “chimpanzee”. Hence it may be derived ”a chimpanzee likes bananas”.

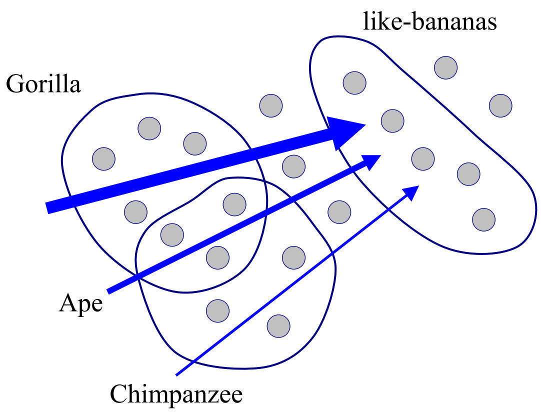


Figure 4.4: Distributed representation

The concept of distributed representation allows the intrinsic recognition of similarities. Thus the assumption for spontaneous generalization is provided. Furthermore, this type of organization leads automatically to a high degree of fault tolerance. If a single neuron fails, this does not lead to the loss of a concept. (It doesn’t make any difference, if you remove a single drop of water from a bucket full of water.)

This applies also to artificial neural networks, of course. Nevertheless, a local representation of concepts is often chosen for systems with a neural architecture due to very pragmatic reasons. This type of knowledge organization is much easier to understand and to interpret for humans than distributed and abstract patterns of activities. Despite the fact that human brains are organized in a distributed manner, it seems to be very difficult for humans to „think distributed“.

5 Artificial Neural Networks

5.1 The Early Days

5.1.1 McCulloch-Pitts-Networks

The forties of the 20th century mark the beginning of neural computing concepts. In 1943 the American physicists McCulloch and Pitts could prove that each logical function could be represented by a network built of simple binary threshold elements (McCulloch, Pitts 1943). Later on, these elements were called McCulloch-Pitts units resp. McCulloch-Pitts (McCP) neurons. They could receive input signals of “0” and “1”, weight these signals either with +1 or –1 and compare the incoming net result with a threshold. If the net result reached or exceeded a threshold the units delivered an output pulse of “1” otherwise “0”. These neurons were not adaptive; the weights had to be determined manually. Hence they were unable to learn. Even if they would have been adaptive from a technical point of view, there was no way how to perform the adaptation. At this time no learning theory was available providing an algorithm how to achieve learning. Figure 5.1 shows how to implement the logical functions AND, OR and NOT with such threshold units.

x y	x AND y
0 0	0
0 1	0
1 0	0
1 1	1

x y	x OR y
0 0	0
0 1	1
1 0	1
1 1	1

x	NOT x
0	1
1	0

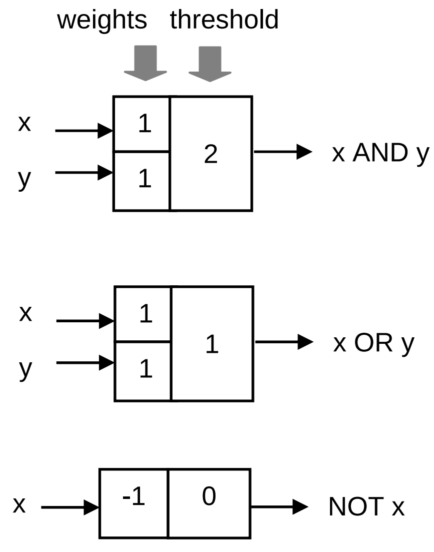


Figure 5.1 McCulloch-Pitts neurons for AND, OR and NOT.

As is generally known all other logical functions can be built by using only AND, OR and NOT. E.g., XOR can be built as follows (x XOR y) = (x OR y) AND NOT (x AND y). Hence, it was clear that even networks consisting of very simple threshold units could realize the capability of any computing device. By appropriate interconnection of McCP-neurons any computable function can be realized. In other words: For each computable problem there is a network, which solves this problem. This is a theoretical result. In practice it is unclear how to find such a network. There is no method available how to achieve a solution for a real and concrete problem. In principle "there-is" propositions of this kind are not constructive. In mathematics such propositions are called existence theorems. There is a solution but there is no way how to find it. E.g. due to the fundamental theorem of algebra an equation of n^{th} order has n solutions but for n > 4 there is no analytic algorithm available how to achieve these solutions. The same applies for McCP-networks. For the theory of cognition the proposition "For each computable problem there is a McCP-network which solves it." is of the highest relevance but is irrelevant from a practical point of view to find a solution.

The work of McCulloch and Pitts had a great and decisive impact on the theory of computer science. Among other things there have been parallels to the theory of cybernetics which was worked out by Norbert Wiener in the fifties of the past century.

5.1.2 Hebbian Learning

A further decisive step in connectionism was the first learning rule in the form of a mathematical formula. In 1949 the American psychologist Donald Hebb introduced this rule, which is known today as the Hebbian learning rule, in his famous work "The Organization of Behaviour". He came to realize that learning is basically the physiological synaptical modification of nerve cells and their connections. Thus, it turned out that the decisive assumption for learning is the plasticity of the synapses of a nerve cell.

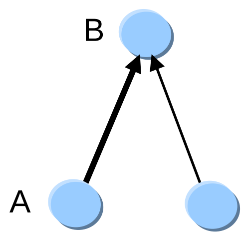


Figure 5.2 Learning is synaptic adaptation of neurons

The Hebbian learning rule may be linguistically expressed as follows: "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased." (Donald D. Hebb. The Organization of Behaviour. Wiley, New York 1949, p. 50)

Hebb expressed this fact in form of the following formula

$$w_{new} = w_{old} + c \cdot \text{Output A} \cdot \text{Output B}.$$

As a result, the connectivity between neurons A and B will be increased.

5.1.3 The Perceptron

In 1958 the American psychologist Frank Rosenblatt introduced the so-called perceptron. Rosenblatt was primarily interested in investigating the following questions

- How is information about the real world percepted at all?
- How is this information organized in the brain?
- How does this information affect the behaviour and the process of recognition?

In a later work Rosenblatt characterized a perceptron as follows (Rosenblatt 1962): “Perceptrons … are simplified networks, designed to permit the study of lawful relationships between the organization of a nerve net, the organization of its environment, and the “psychological” performance of which it is capable. Perceptrons might actually correspond to parts of more extended networks and biological systems; in this case, the results obtained will be directly applicable. More likely they represent extreme simplifications of the central nervous systems, in which some properties are exaggerated and others suppressed. In this case, successive perturbations and refinements of the system may yield a closer approximation.”

Rosenblatt investigated so-called photo-perceptrons, which receive optical stimuli via the retina.

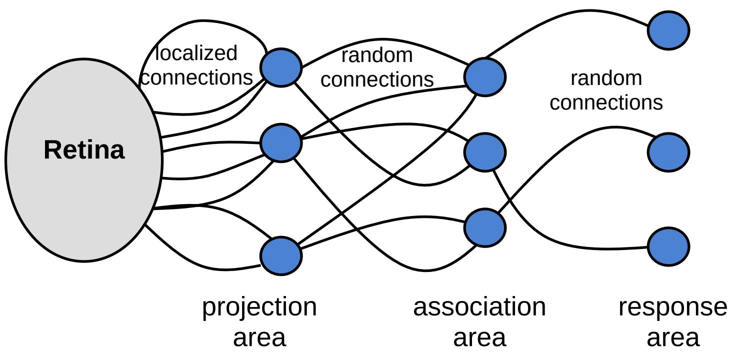


Figure 5.3 The photo-perceptron of Rosenblatt

If reducing this structure to the most important elements, which are actually responsible for the process of recognition, then the intrinsic perceptron remains as a network with the following structure. The input-layer corresponds to the so-called association area of the perceptron and the output-layer corresponds to the response area.

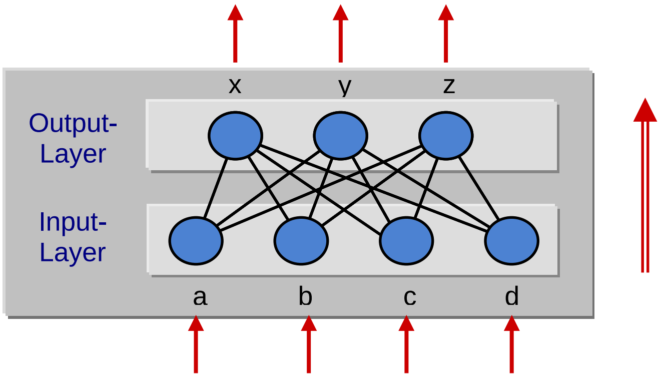


Figure 5.4: The perceptron as a two-layer network

The neurons of a perceptron are organized in two layers. Within one and the same layer the neurons are not interconnected at all, but both layers are interconnected completely, i.e. each neuron of one layer is connected with each neuron of the other layer.

The lower layer is called the input-layer. The job of the input-layer is to receive the input signals of an artificial retina and to transfer it to the upper layer, which is also called the output-layer where the actual processing takes place, i.e. here the output values of the perceptrons are computed. The computation takes place within one processing cycle. Due to the one directional processing mechanism from input-layer to output-layer this type of processing is called feed-forward propagation. What really happens is that the perceptron performs some kind of a vector transformation, i.e. the input vector to the input-layer is transformed into a new vector of the output-layer. (In figure 5.4 the input vector (a,b,c,d) is transformed into the output vector (x,y,z)).

Rosenblatt's work culminated in the proof of the so-called perceptron convergence theorem. It states that, if a problem can be learned by means of a perceptron then the perceptron will find a solution after a finite number of training steps. But this theorem does not answer the question whether a solution for a problem can be found by a perceptron at all. This is not the case.

The next figure 5.5 shows in its upper third the table of the logical functions AND, OR and XOR. (Remark: The XOR-function denotes the exclusive OR-function which means "either ... or".) In the middle third of the figure an appropriate perceptron is presented for AND and OR-functions and indicated for XOR-function. In the inner of the neuron the thresholds are placed. If a threshold has been reached or exceeded the neuron fires a signal with the value "1". The weights of the output neuron are denoted at the edges.

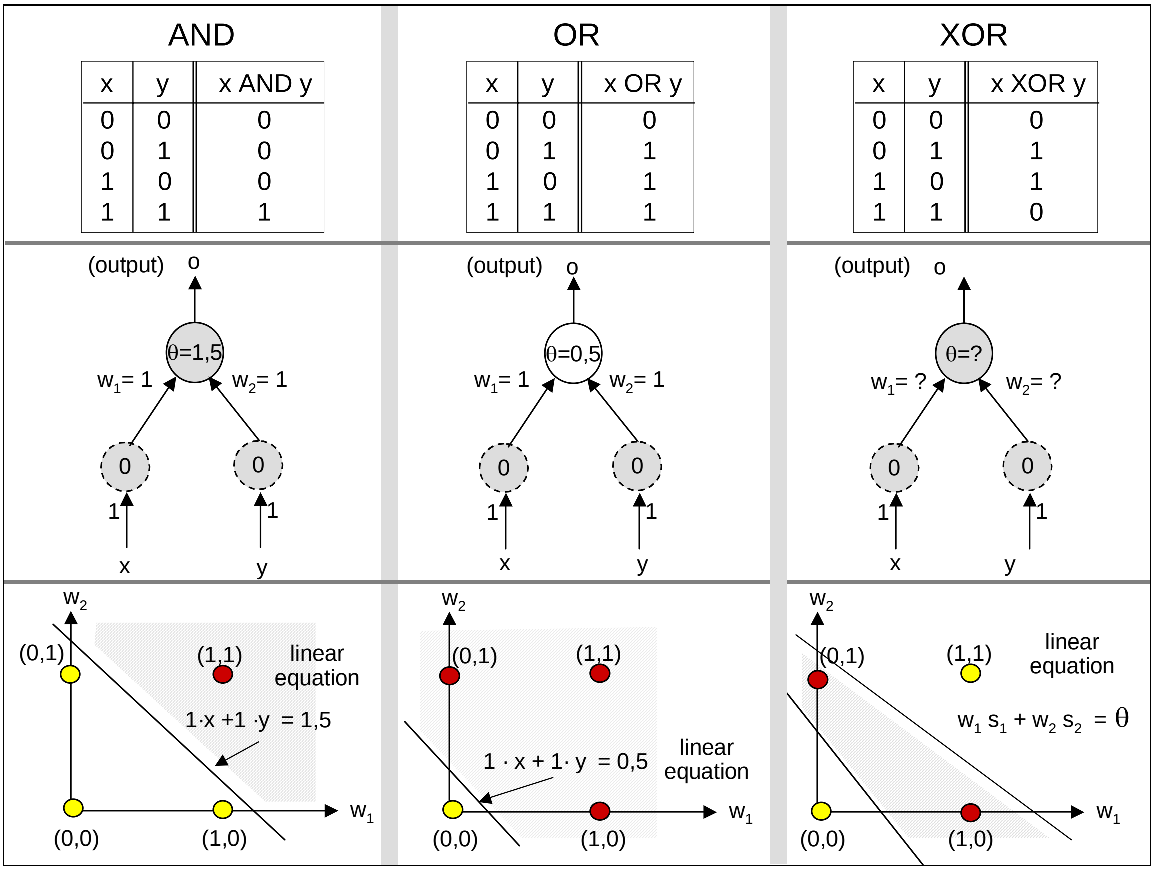


Figure 5.5: AND-, OR- and XOR-problem

Each perceptron realizes a linear inequality ($w_1 \cdot x + w_2 \cdot y \geq \Theta$). If one considers the linear equation $w_1 \cdot x + w_2 \cdot y = \Theta$, it is obvious that the perceptron makes a classification decision. The set under consideration is divided into two classes. An appropriate separation is only possible for the AND and OR-function. Here the “yellow” (light) spots are separated from the “red” (dark) spots by a straight line. (The “red” (dark) spots denote tupels of the function which should be associated with value “1”, the “yellow” (light) spots denote such tupels which are associated with value “0”.) This cannot work for the XOR-function because a straight line cannot divide the red (dark) and yellow (light) spots as required. Therefore another straight line is required as shown in the illustration. This simple example shows that a perceptron can solve only such kinds of problems which can be solved by dividing a plane into two parts by a straight line or a 3-dimensional space into two parts by a plane or a 4-dimensinal space into two parts by a 3-dimensinal space and so on. Such problems are called linearly separable problems. A space, dividing another space into two parts and with dimension reduced by 1 is called a hyperplane, i.e. in an n-dimensional space, hyperplanes are objects of n-1 dimensions. (An n-dimensional space is usually referred to as a hyperspace.) Suitable arrangements of hyperplanes can be used to partition an n-dimensional space into various distinct regions. Generally, a perceptron determines a hyperplane within the weight space which separates it (the weight space) into two parts.

The underlying reason for this inability of the perceptron to solve the XOR-problem is based on the fact that a perceptron performs a special type of transformation which is called an association. Such an operation is a kind of similarity operation. Similar inputs will lead to similar outputs, to associations. This is not the case for the XOR problem. The input vectors (0,0) and (0,1) are similar to a degree of 50%, as well as (0,0) and (1,0) but lead to different results. ((0,0) → 0, (1,0) and (0,1) → 1). On the other hand (0,1) and (1,0) lead to the same result (1), but they are different to 100%. Perceptrons can only mimic associations, they are unable to perform other kinds of operations. An instance is missing which performs some kind of encoding that enables a perceptron to perform a more complex type of operation.

The works of Rosenblatt were a milestone in the discipline of neurocomputing and marked the beginning of a stormy development phase. At the same time the discipline of symbolic Artificial Intelligence emerged. One of the leading protagonists of symbolic AI was Marvin Minsky who also concerned himself intensively with the theory of perceptrons. In 1969 Minsky published together with his colleague Papert a book entitled “Perceptrons” (Minsky, Papert 1969). With this work Minsky and Papert could prove that only a certain class of problems - so-called linearly separable problems - could be solved with perceptron-like networks. But such a simple problem as the XOR-problem could not be solved by means of a perceptron. (The XOR-problem is <u>the</u> classical problem in neurocomputing; it is included in almost every book on neural networks.)

The appearance of this book, Perceptrons, is often credited with causing the demise of neural network technology. The fact of limited capability of perceptrons curtailed most research in this field. The results of Minsky and Papert meant a serious throwback for the upcoming and strongly emerging neural network technology. Obviously only problems with limited complexity could be solved by neural networks. This lack seemed to be inherent and a matter of design and construction; it seemed to be irreparable. For more than a decade neural network technology fell into a state of an unnoticed discipline, not worth researchers spending much time and energy on deep and detailed study. But still, during this time period, some isolated islands of research continued. The so-called “return to respectability” in the early 1980’s was initiated by the revolutionary works of John Hopfield who introduced a new type of neural network strongly inspired by phenomena from solid state physics.

5.2 Network Topologies

A perceptron is a neural network with a specific structure at to how neurons and their connections are organized, i.e. it embodies a specific form of connectivity structure. The question comes up if there are other network types. A neural network structure can be defined as a collection of (artificial) neurons interconnected in the form of a directed graph. The type of connectivity is also called topology of a neural network.

There is a great variety of network topologies. The type of connectivity determines to a high degree the knowledge and behaviour of a neural network. Using a very rough classification one can distinguish the following network types:

1. layered feed-forward networks (e.g. perceptron; often mathematically motivated)
2. recurrent networks with feedback effects (often completely interconnected and physically motivated)
3. primarily biological motivated networks, e.g. Kohonen networks

There are very many different network types in detail. This fact in itself seems a likely supposition that some kind of a universal network type has not been found up till now. Each network type has its strengths and weaknesses and is able to mimic certain neural aspects. Nevertheless, there are many special solutions to treat specific problem types. It is evident that there is still a very long way to go for a sound understanding of biological brains and neural activities in their entirety.

Furthermore, it is important to emphasize that network types and learning strategies must be carefully adjusted. Learning strategies are algorithms, which must “fit” to the underlying network structure.

All of these networks are called neural networks because of their ability to simulate important neural aspects like distributed representation and parallel processing even if they are motivated from very different viewpoints, as will be shown in the following discussion, e.g. by physical phenomena. Nevertheless, the term *neural network* makes sense because predominating neural aspects are more important than different motivations.

5.2.1 Layered Networks

Let us consider layered networks at first. In layered networks the neurons are organized in form of layers. Neurons within one layer are not connected at all, but neighbouring layers are completely interconnected, i.e. each neuron of one layer is connected with each neuron of the neighbouring layer. The simplest two-layer network is the perceptron. As demonstrated with the XOR-problem perceptrons have limited problem solving capabilities. But the limitations of perceptrons can be overcome by introducing additional layers. Networks with more than two layers are called multi-layer perceptrons. Figure 5.6 shows a multi-layer perceptron with three layers. The layers between input- and output-layer are called hidden layers because they are invisible for a user.

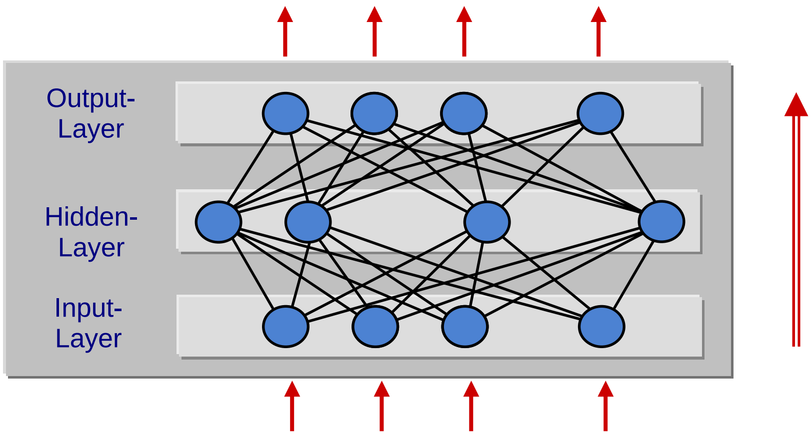


Figure 5.6 A multi-layer perceptron

The next figure illustrates how to solve the XOR-problem with a three-layer network.

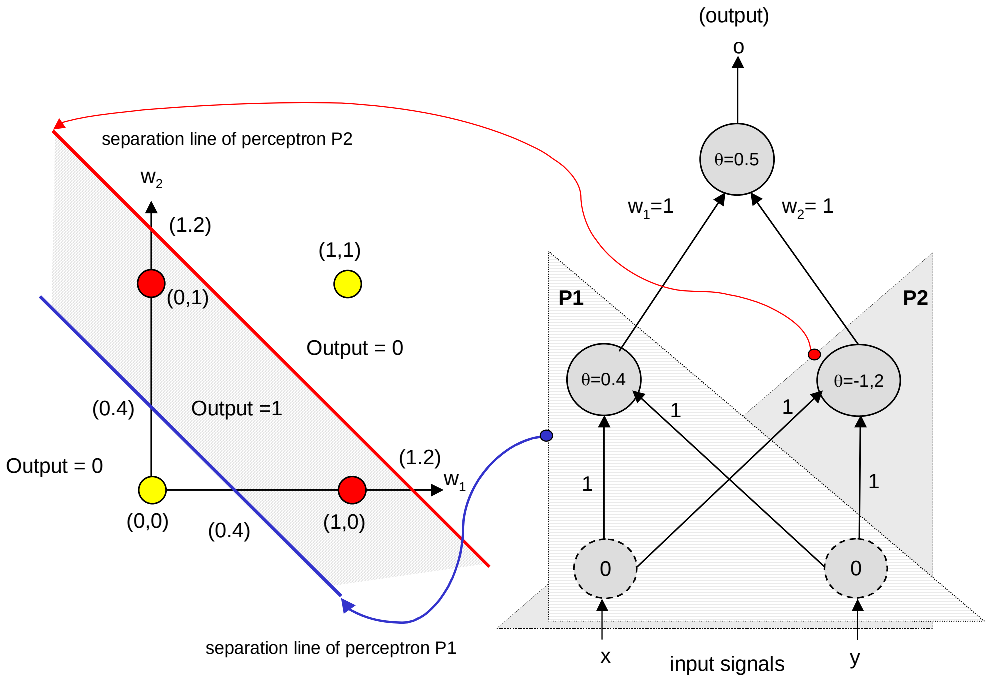


Figure 5.7 A three-layer perceptron as a solution of the XOR-problem

By adding new layers more complex regions may be may be shaped in weight space as shown in the next figure, taken from Lippmann 1987.[10]

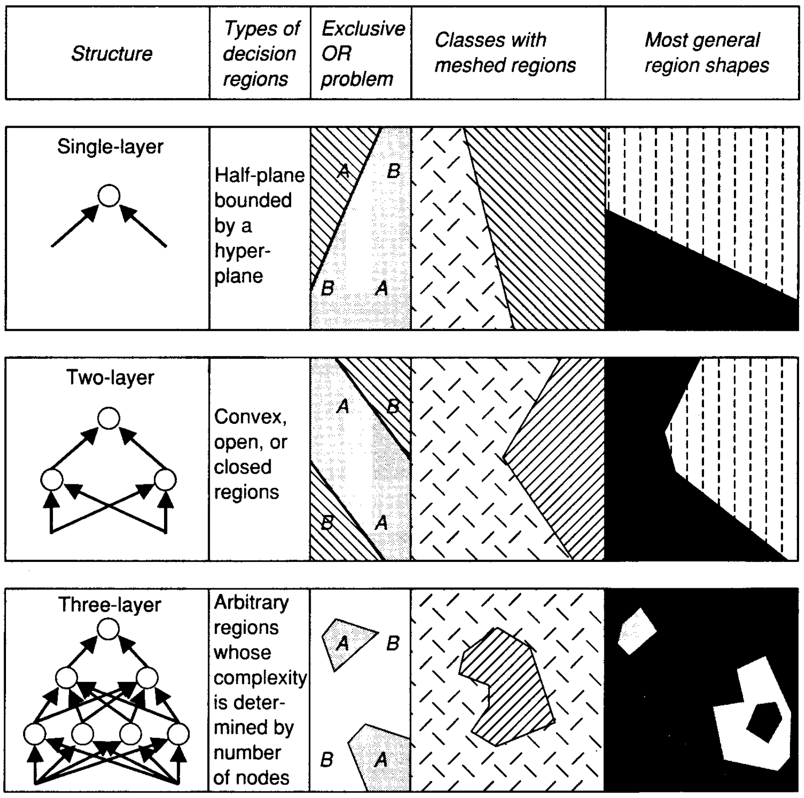


Figure 5.8 Hierarchized perceptrons

[10] Lippmann uses a slightly different terminology. He calls a n-layered Network a (n-1)-layer network, because he does not take into account the input layer due to its pure passive behaviour of information receipt and "as-is" transfer function. Hence, he calls a (two-layered) perceptron a „single layer network".

Minsky and Papert have surely been aware of the fact how to solve the XOR-problem by introducing new layers. Limited capability of perceptrons has been a key issue of their book but Minsky and Papert raised other subjects of concern as e.g. scaling, too. However, why the harsh criticism of neural network technology then, if the problem can be easily solved.

Learning Strategies of the Perceptron

The key for understanding lies in the learning strategies. A three-layer perceptron as presented in figure 5.7 solves the XOR-problem. But here the weights have been determined manually. This may be possible for such small problems like XOR but it is impossible for bigger problems. E.g. for image recognition problems it is absolutely not applicable; such problems are too complex for manual weight adaptation.

Also, biological neural systems are not born pre-programmed with all the knowledge and abilities that they eventually have. A learning process that takes place over a period of time somehow modifies the network to incorporate new information.

But how does this learning process work? How can we make an artificial neural network learn? In other words: How do we program a neural network? The complete knowledge of a neural network is contained in its topology, or to be more precise in its weights. The complete weight set determines the complete behaviour of a neural network. Weights can be set manually or automatically. Setting weights manually is a very difficult and time-consuming task. In case of distributed representation of knowledge it is almost impossible. Setting weights automatically seems to be much more appropriate. Hence, it is important to specify learning strategies to enable the network to learn resp. to be trained automatically. If the network has been trained sufficiently, normally by using a representative training set, training can be terminated and the adapted network may go to work afterwards. Hence, one can distinguish two operating phases of an artificial neural network

1. the training resp. learning phase
2. the operational resp. recall phase

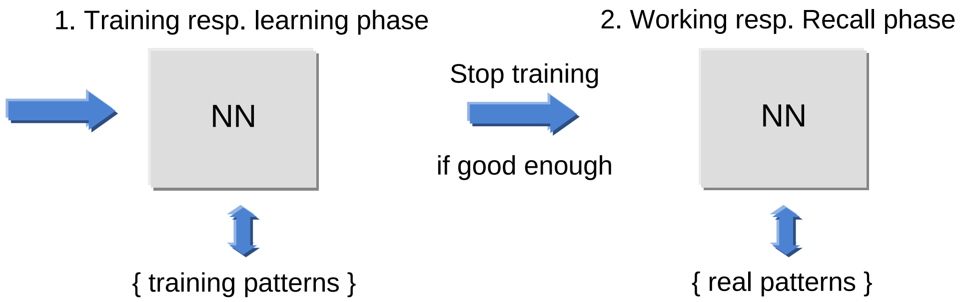


Figure 5.9: Operating phases of a neural network

A criterion for a sufficiently successful training could be to classify the training set or a significant subset of it properly. As long as the training results are not satisfying training has to be continued. As discussed in the Hebbian learning rule, training means the adaptation of weights. The learning rule determines exactly how to adapt the set of weights of the neurons.

It quickly turned out that the Hebbian learning rule is rather rough; the correction of the weights always takes places in equal steps despite whether the error of the neuron is big or small. To refine the adaptation of the weights the Hebbian rule was modified in such a way that the adaptation of weights is a function of the error made. The smaller the error the smaller the correction of the weights. Due to its behavior this rule was called the delta rule where delta means the size of the error. The delta rule modifies the weights as follows

$$w_{new} = w_{old} + c \cdot \text{Output A} \cdot [\text{Error of B}]$$

The only difference to the Hebbian rule is to replace the output of neuron B by the error of neuron B, where the error (delta) is defined as the difference of the expected value (or target value) and the real value (or actual value), i.e. *error := target value – actual value*. (Hence, the delta rule is only a variation of Hebb's rule.) The delta rule leads to a convergence of weights.

The discussion of these learning rules shows that the error is detected at the neurons of the output-layer of a neural network. A two-layer network may be trained using one of these rules. But what happens with multi-layer networks? How can an error be associated to a neuron of a hidden-layer? Both Hebb and the delta rule will fail in such cases. How does one measure the error of a hidden neuron if there is not even a definition for such an error? There was no idea how to define the concept of an error for hidden neurons at all. As a consequence, if only the output-layer of a multi-layer neural network may be trained all other layers remain as they are. Virtually one can combine them to one single layer - like a black box - that behaves like the (n-1) layers. As a result, again we get a perceptron with the same well-known set of limitations. Hence, we remain on the complexity level of a perceptron. That is the core of the problem identified by Minsky and Papert.

5.1 The Multi-Layer Perceptron and the Backpropagation Learning Rule

As shown in figure 5.8, any kind of region may be shaped in the weight space by adding new layers. As a result one gets networks with many layers. The core problem was the training of such networks; an appropriate learning algorithm was missing. To solve this problem many different approaches had been proposed, typically strongly depending on the type of problem.

The decisive breakthrough happened in 1986 by the introduction of the so-called backpropagation learning strategy by Rumelhart, Hinton and Williams (1986). One step was to replace the hard-limiter function by a continuously differentiable function. A threshold decision corresponds mathematically with a hard-limiter function. Once the threshold has been reached the neuron fires. A necessary assumption for the introduction of a learning rule, which also includes hidden layers, was to replace the hard-limiter by a continuously differentiable function that behaves basically as a hard limiter to make sure that the essential behavior of a neuron still stays the same. Sigmoid (S-shaped) functions represent such a class of functions with an asymptotic behaviour where 1 is the upper asymptote and 0 is the lower asymptote (s. figure 5.9). Thereby the abrupt jump of the hard-limiter is replaced by a quick and steep but continuous rise. By variation of the functions' gain parameter the curve may be shaped as indicated in the right hand side of figure 5.10. It seems that sigmoid functions are able to mimic the behaviour of biological neurons appropriately and without difficulty.

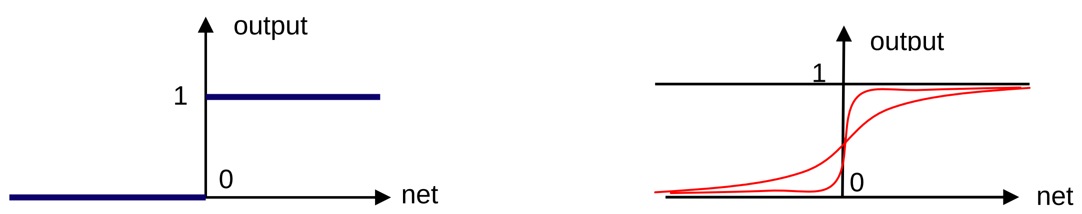


Figure 5.10 Hard limiter and sigmoid functions

The backpropagation learning strategy was able to associate an error also to hidden-layer neurons. This error took into account the errors of all neurons in the upper layer. If we only consider the output-layer neurons then the backpropagation rule looks rather similar to the delta rule. The mathematical proof of this rule is essentially based on the chain rule of differential calculus for functions with multiple variables, therefrom the necessity of continuously differentiable activity functions. Therewith a general learning rule for multi-layer networks was available.

The backpropagation algorithm works as follows.

Step 1: A training vector (resp. pattern) is presented to the input-layer of a multi-layer perceptron. The information is propagated layer per layer through the network (feed-forward propagation). As a result, we get an output vector (resp. pattern) at the output-layer.

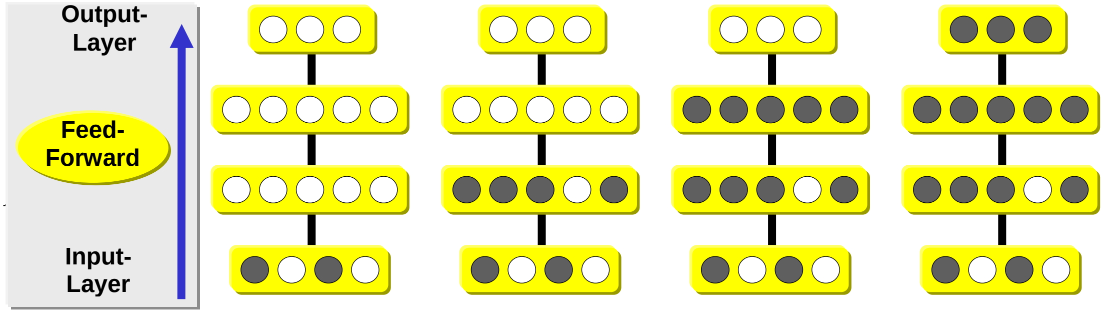


Figure 5.11 Forward propagation of input information for MLP

Step 2: At first the error at the output-layer is determined by calculating the difference between expected value and actual value. Then this error and some other information is used to adapt the neurons of the output-layer. Afterwards error information is backpropagated into the layer beneath. Therefore the error of this layers' neurons is derived based on the errors of the output-layer followed by an adaptation of weights using this artificial error. This procedure takes place layer per layer until the input-layer has been reached. Then the adaptation of weights is terminated.

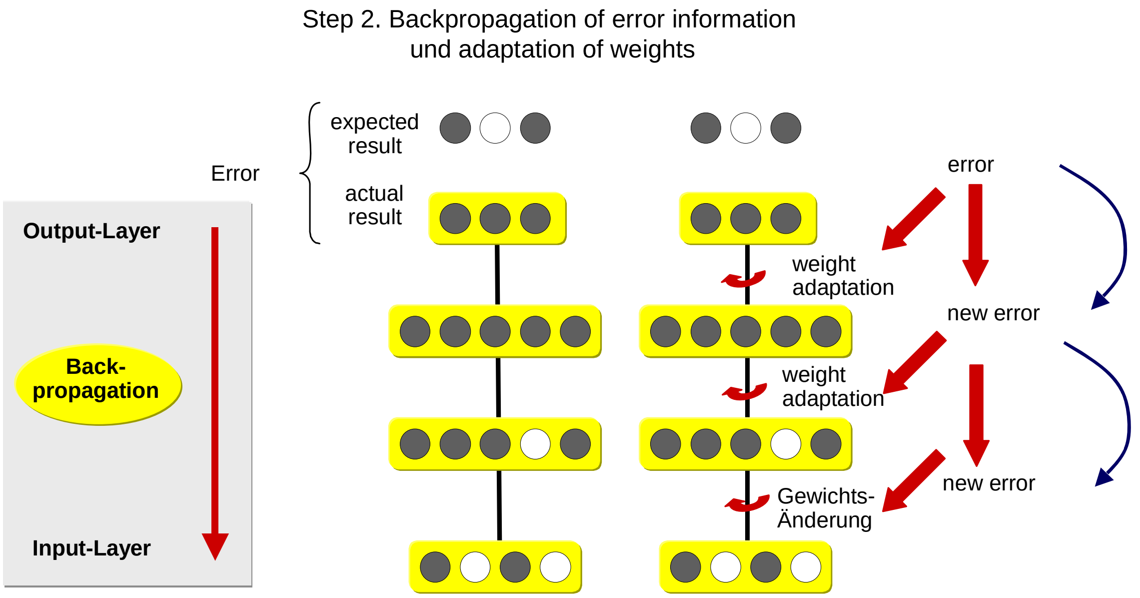
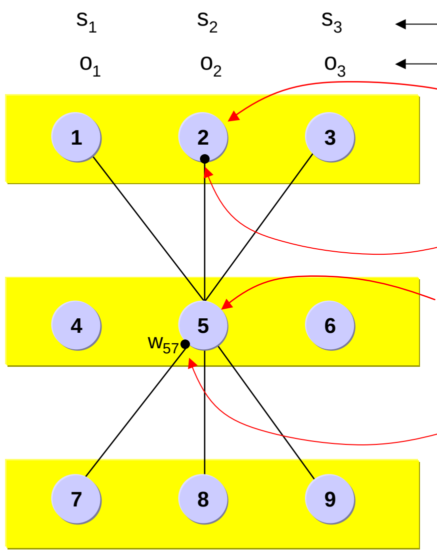


Figure 5.12 Backpropagation of error information and weight adaption for MLP

Let's discuss the backpropagation algorithm in some more detail using a three-layer perceptron. Figure 5.13 depicts such a network. For instance at neuron 2 of the output-layer the output value o_2 occurs whereas s_2 is the expected value.



$$\delta_2 = (s_2 - o_2)a'(net_2)$$

$$\Delta w_{25} = c \cdot \delta_2 \cdot o_5$$

$$\delta_5 = a'(net_5)(\delta_1 w_{15} + \delta_2 w_{25} + \delta_3 w_{35})$$

$$\Delta w_{57} = c \cdot \delta_5 \cdot o_7$$

$$w_{new} = w_{old} + \Delta w$$

Figure 5.13 Error and weight adaptation of backpropagation algorithm

Using the delta rule the error was defined by the difference (s_2 - o_2). In case of using the backproagation algorithm the term (s_2 - o_2) · a'(net_2) is defined as error of neuron 2. Here a'(net_2) denotes the derivative of activity function a(.) at net_2.

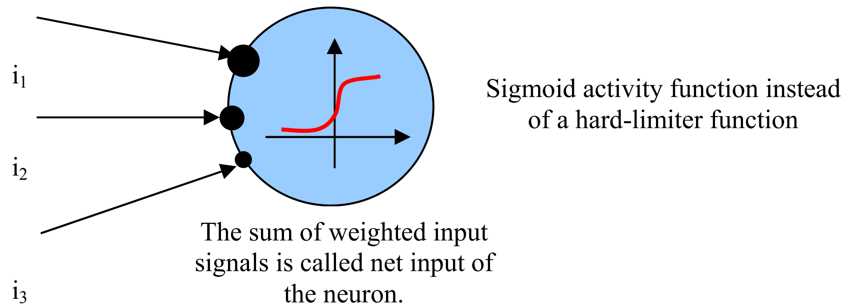


Figure 5.14 Neuron with a sigmoid activity function

The activity function can have values between 0 and 1. Thus the error (s_2 - o_2) will be decreased by multiplying it with a'(net_2). As a consequence, the adaptation of the weights using the backpropagation algorithm is somewhat slower than using the delta rule. This new "artificial" error is now used to modify the weights of neuron 2. E.g. the weight w_{25} between neuron 2 and neuron 5 is modified by the term $c \cdot \delta_2 \cdot o_5$. c is a learning factor, usually a number between 0,1 and 0,3, and o_5 is the actual output value of neuron 5. In an analogue manner, the weight w_{14} between neuron 1 and neuron 4 is modified by $c \cdot \delta_1 \cdot o_4$. For comparing these terms with the delta rule there is a high degree of similarity. However, Hebbian learning seems to be the mother of all learning strategies.

The adaptation of the weights of the hidden-layer neurons is a little bit more difficult. Let us consider neuron 5. This neuron is associated with the artificial error δ_5= a'(net_5) · [$\delta_1 \cdot w_{15} + \delta_2 \cdot w_{25} + \delta_3 \cdot w_{35}$]. One sees immediately that the definition of this error is based on the errors δ_1, δ_2, δ_3 of the neurons of the upper (output) layer. This is a so-called recursive definition. If there would exist more hidden layers, the errors of their neurons would be based on the errors of the upper layer respectively and so on.

The question arises as to how many layers are needed to solve a problem. Do we need more layers for difficult and complex problems than for simple problems? As a matter of fact, each network with more than three layers can be reduced to a network consisting of no more than three layers. Generally speaking, each (calculable) nonlinear vector transformation may be performed by a three-layer network.[11] This fact does not answer the question whether more than three layers may be useful from a practical point of view. The less layers are available the more carefully one has to use the resources of the network. Using sparsely or minimally designed networks even very slight weight modifications may easily disturb the highly sensitive network. It may happen easily that an "unlearning" effect may occur, i.e. learned patterns may get lost.

Finally, it stands now to reason that multi-layer networks have been classified as mathematically motivated. First of all, it does not appear biologically motivated that neurons within one and the same layer are not interconnected. Secondly, a learning strategy, which is based on differential calculus, appears to be fairly artificial. Apparently, the availability of appropriate mathematical algorithms has been a crucial factor to establish such networks. Although an airplane does not fly like a bird, the flying of birds has inspired researchers to understand flying and to make it possible. Essential parts of the original have been copied, but other significant portions have been solved using technology.

The introduction of the multi-layer perceptron combined with the backpropagation algorithm contributed significantly to the renaissance of neurocomputing in the 1980's. If one considers operative real-world and field-tested systems in neural architecture the majority (appr. 80%) of all such systems are multi-layer perceptrons. Thus, it is only just and equitable to denote multi-layer perceptrons in combination with the backpropagation algorithm as the workhorse of neurocomputing.

Finally, it should be mentioned that in 1988 Minsky and Papert published a revised and expanded edition of their book *Perceptrons* (Minsky, Papert 1988), due to the new insights. Nevertheless, many concerns still raised by Minsky and Papert should not dismissed lightly, e.g. scaling.

5.3 Hopfield-Networks

In the following section, a new network type will be discussed considering a simple image recognition problem as an example. The American physicist John Hopfield could prove in 1982 that a certain class of back coupled networks has a strong analogy to so-called spinglasses. He introduced a network type, which is closely adopted to physical phenomena of such spinglasses (Hopfield 1982). This work is considered to mark the beginning of the return to respectability of neurocomputing. Later on, this network type was called Hopfield network. In Hopfield networks all neurons are completely interconnected, i.e. each neuron is connected with each other neuron very similar to

[11] Human brains are said to have six layers, but here neurons are strongly interconnected as well within a layer as embracing multiple layers, too.

physical systems. Hopfield's motivation for designing such networks were phenomena in solid-state physics resp. in statistical mechanics, which occur in so-called spinglasses. In solid-state physics solid bodies with an erratic arrangement of its atoms are called spinglasses.

There are solid bodies with a specific magnetic behaviour. The overall magnetic behaviour of such a solid body could be described as a consequence of their interaction of atoms and molecules. E.g. the so-called ferromagnetism – a crystal property – is based on such interactions. The most salient feature of such ferromagnetic material is that even a weak outer magnetic field, exceeding a certain threshold, causes a very strong magnetization of the entire spinglass. This effect is based upon the existence of magnetic and non-magnetic particles. Each particle has a so-called Ising-spin, which describes the magnetic orientation of that particle. There are two types of magnetic orientation which are denoted by "+1" and "-1". The particles have a mutual impact on each other in such a manner that in certain solid bodies stable regions resp. domains emerge in which all particles are aligned in a parallel or anti-parallel manner as depicted in figure 5.15.

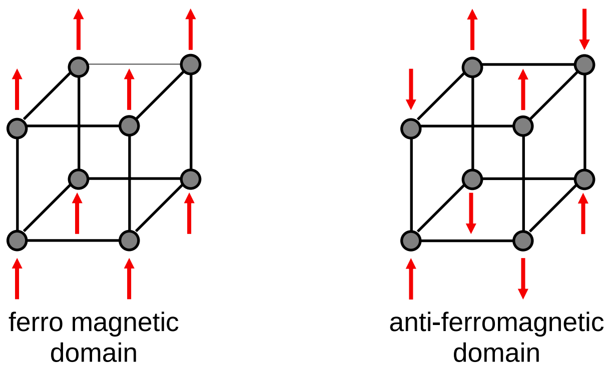


Figure 5.15 Magnetic domains in a spinglass

An individual particle is – so to speak – aligned due to the resulting net impact of its environment. If perturbations occur single spins will be adjusted to its correct orientation due to the forces of its neighboring atoms. The entire alignment process can mathematically be described per particle. Within the spinglass an energetic process takes places. Typically physical systems may be better described by a global quantity, which is the energy, instead of describing the behaviour of a great many of particles. Physical systems tend inherently towards a stable energy state, which represents a local energy minimum. If a pen held in one's hand is released it falls down to the floor thus reaching a state of minimal potential energy. A pot of hot water cools down until it has reached the temperature of its environment, and a ball rolls down a hill towards the bottom, always moving in the direction of the steepest downward slope. In all cases an automatic physical process of energy minimization takes place leading to a stable state which represents a local minimum. A local minimum is also called attractor state. If one imagines the process of energetic state changes as a movement in the energy landscape of the system, then this movement is always directed downwards. This means that the energetic state of a physical system either decreases or stays constant until a stable energy state has been reached. At the beginning the system resides in an attractor valley and is attracted downwards into the bottom of the valley. (This applies under the assumption that no external energy is fed, of course.)

Hopfield took advantage of this fact when designing his new network in close analogy to a spinglass. A Hopfield network consists of n binary neurons, which are completely interconnected. The neurons may have only the values 0 and 1 as valid output values, according to the Ising-spin "-1" and "+1", as well as a threshold according to the outer magnetic field. The weights between two neurons correspond closely to the forces between two particles. Due to the physical law "action = reactio" the weights between two neurons are chosen symmetrically. Also the dynamic alignment function of a particle was transferred to the neurons adequately. As a consequence, the resulting network should behave like a spinglass. According to this, a Hopfield network should have a corresponding quantity for the energy of a spinglass and a global energy function. Also for Hopfield networks this quantity is also called "energy". Hence, Hopfield networks will also tend to achieve automatically a stable state of local minimal energy in the energy landscape.

Spinglass	Hopfield-network
spin	activity
(symmetric) coupling forces	(symmetric) weights
dynamic behaviour of a particle	dynamic behaviour of neuron
global energy function	global „Energy“ function

The question comes up what is the input and what is the output of a Hopfield network because there is no input- and no output-layer? The input is a set of initial values of 0 and 1 for all neurons of the network. Then the network is left to itself. Due to its inherent dynamics it will automatically tend to reach a state of minimal energy. If one imagines the neurons as being bulbs (0 = "off, 1 = "on") then the process of finding a stable energy state could be visualized as a flickering of all bulbs. Normally, the flickering will be strong at the beginning of the process and will become lower over time until the flickering stops and the bulbs have reached a final “off” or “on”-state. Then the ordered set of final activity values of the neuron represent the output of the network. Apparently, Hopfield-networks are well suited to solve optimisation problems due to their construction in close analogy to a physical system.

Because each neuron is coupled with all others it may well happen that feedback effects occur because a neuron may have an effect on its own via loops. That’s why such network types are also called recurrent networks.

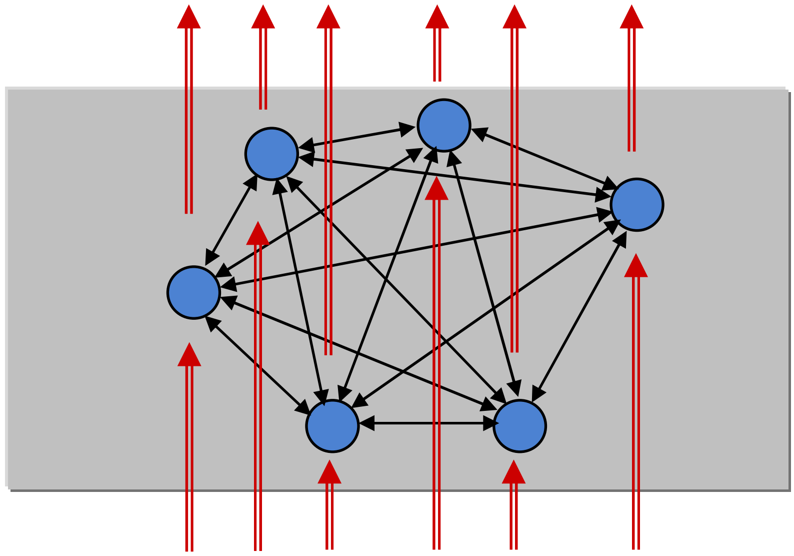


Figure 5.16 Structure of a Hopfield-network

To solve a problem with the means of a Hopfield-network it is necessary to correlate the target function of the problem with the energy function of the network, i.e. the target function has to be “disguised” in such a manner that it looks like an energy function to the network to be optimized. The question arises how to represent the problem appropriately. We will discuss the question considering a simple image recognition problem as an example. We will try to recognize and to reconstruct noisy and incomplete images with means of an appropriate Hopfield-network. (One may think of a real world problem behind it like recognition and identification of people based on stored pictures e.g. for an application type like fighting against crime.) Given twenty pictures consisting of 20x20 pixels, each of them with 200 white and 200 black pixels. The first picture shows a “little man”, the second picture one of 19 other random patterns[12].

[12] This example is taken from Ritter, Martinetz, Schulten 1990.

Figure 5.17 Pixel pictures: “Little man” and a random pattern

It is easy for humans to recognize the „little man“ due to his symbolic and iconic appearance. This is not the case for all of the other 19 pictures due to their random nature. Now a Hopfield-network should be enabled to recognize noisy patterns and to assign it to the most similar of these 20 patterns.

The first questions come up. What should an appropriate network look like? How to represent a picture to present it to the network? How can we achieve the goal that the recognition process can be performed by the network?

At first, a picture has to be represented as a vector because networks perform nothing more than a vector transformation. There are some options for a representation resp. a coding. One solution is to represent a white pixel by “0” and a black one by “1”. The picture is now read line-wise (resp. column-wise) and transformed into a vector of length 400. The “little man” is then transformed to the vector (0,0,0,0, …,1,1,1,1).

Now we build a network consisting of 400 completely interconnected neurons. These 400 neurons may be virtually positioned in a plane so that they correspond to a 20x20 picture in order to “lay” a picture upon the network in a one-to-one manner that pixel 1 corresponds to neuron 1, pixel 2 to neuron 2 and so on. By this means each component of the input-vector is exactly associated to a certain neuron and vice versa. Now we are ready to present the input vectors to the network. The next figure illustrates the principle. A noisy picture of our “little man” is now presented to the network. The “little man” is not only recognized but also completely reconstructed. If a stable state of the network has been reached, the reconstructed picture may be accessed from the network by reading the ordered activity values of the neurons of the network. Then the output-vector is decoded and transformed into a picture.

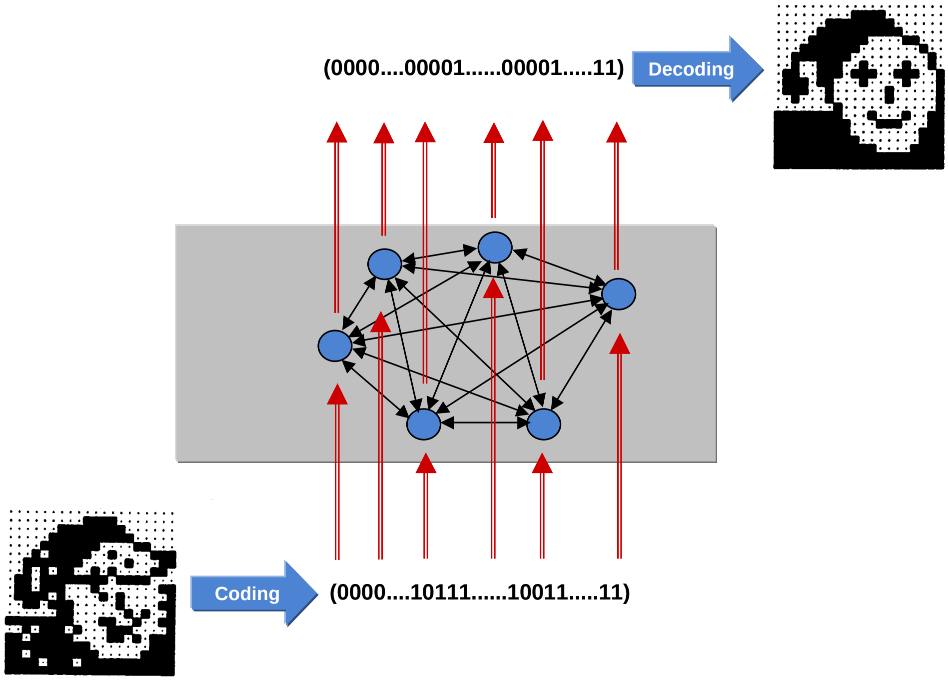


Figure 5.18 Recognition process of a slightly noised „little man“

The recognition process should be described in some more detail. By finding the appropriate weights the energy landscape of the corresponding energy function is shaped in such a manner that its state in a local minimum corresponds exactly to a pattern to be learned. E.g. the energy landscape of the network must have a local minimum exactly there where the activity pattern of the network represents a local minimum which corresponds exactly to the activity pattern of the "little man". The same applies for all other of the remaining 19 pictures. If a noisy picture of our "little man" is presented to the network, this noisy picture corresponds to a network activity, which is located at the hill of the "little man"-attractor valley in the energy landscape of the network. The network may be compared with a ball, which rolls downwards into the bottom of the attractor valley. The ball cannot roll down into another attractor valley because in that case the network has to climb up the hill of the attractor valley of the "little man" which would mean that external energy is fed to the system. This is impossible for a Hopfield-network because it behaves like a physical system. The network cannot perform such an activity, otherwises it would behave like a *perpeetum mobile* which is impossible from a physical viewpoint.

Figure 5.19 illustrates this situation. The „little man“ corresponds to the local minimum *M-LM* of the energy landscape. The noise in the picture correlates with the distance to the bottom of the attractor valley. The bigger the noise the bigger the distance from the bottom. As long as the noise does not remove the network out of the attractor valley it will slip again into the bottom of this valley, if left to its own dynamics. If the noise is too big, then the network slips into another attractor valley of another picture. In figure 5.19, it slips into the attractor region of minimum M-2 which represents picture 2.

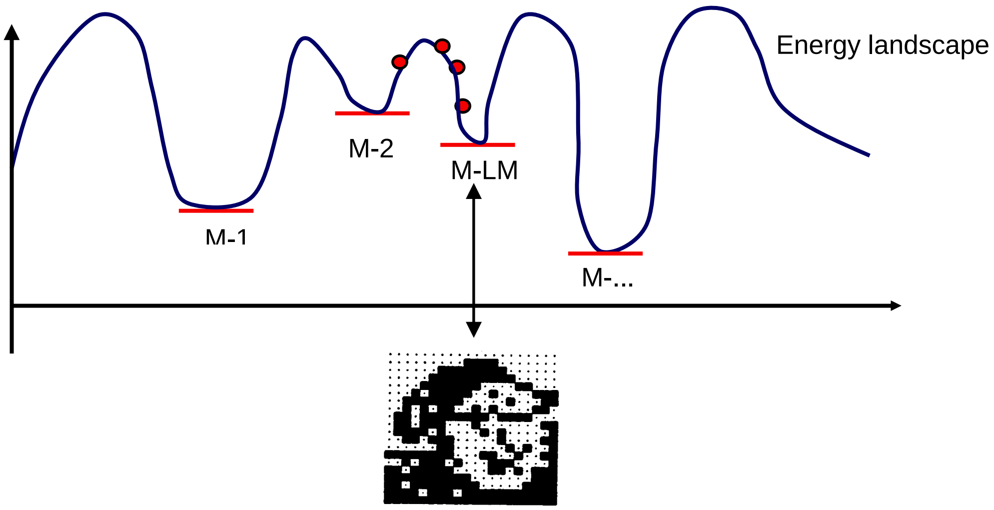


Figure 5.19 Energy function of a Hopfield-network

For the matter of illustration let us consider some further examples.

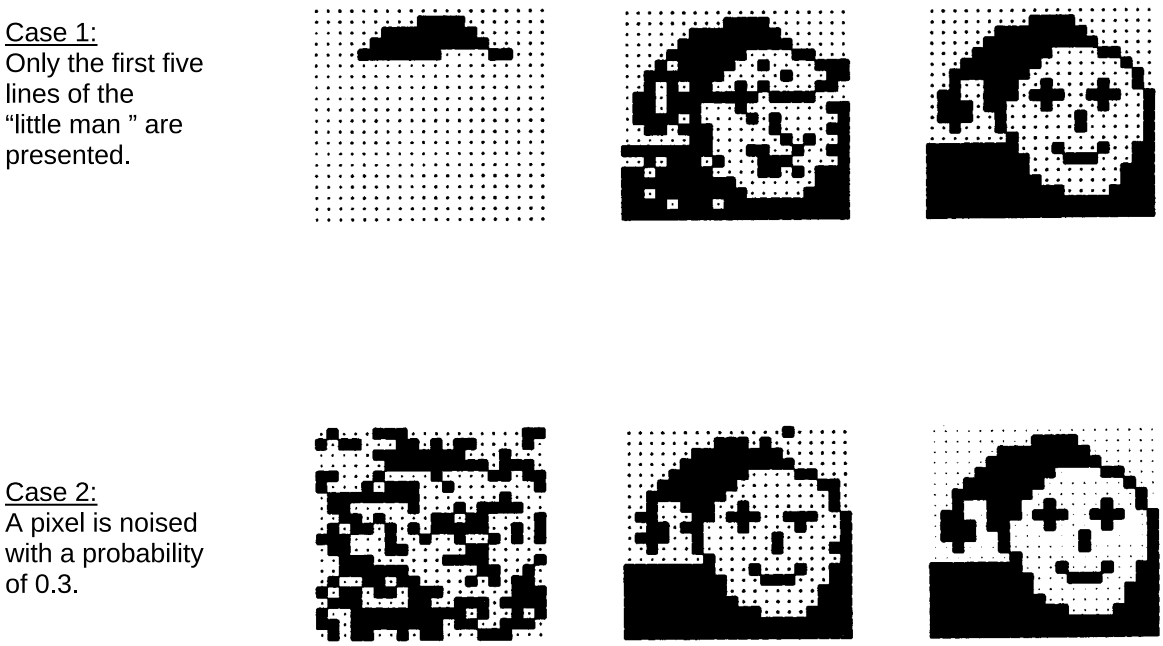


Figure 5.20 Noisy „little men“

In case 1 only the first 5 lines are presented to the network. The figure shows that the picture is fully recognized and reconstructed after a few steps. The network has not only 25% but appr. 62.5% of information, due to the fact that the missing 75% are correct to appr. 50% because the picture has 200 white and 200 black pixels. Therefore, 50% of the missing part of the picture is correct information.

In case 2 the "little man" is noised with a probability of 0.3. For a human the "little man" is no longer recognizable. But figure 5.20 shows that the network has corrected the majority of failures rather quickly. Soon afterwards the picture has been completed correctly. Obviously the "noise" was not strong enough to pull the network out of the attractor valley of the "little man"

In case 3 (see figure 5.21) the "little man" is noised with a probability of 0.4. Now the network is no longer able to reconstruct the "little man". It converges against one of the other 19 random pictures. Due to the heavy noise the picture is now more similar to another picture. The network has left the attractor valley of our "little man" and has moved into an attractor region of another picture slipping into its bottom.

Case 3:
Each pixel is noised with a probability of 0.4

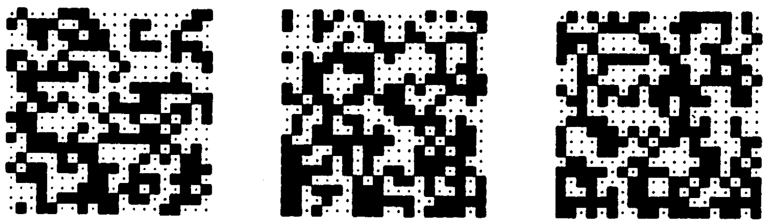


Figure 5.21 A „little man" noised by 40%

6 Lerning Strategies

Up to now, we have discussed the Hebbian, the delta, and the backpropagation learning rules. Taking a look to the formulas, one can see a high degree of similarity. Basically, they share the same structure. The main difference is based upon different definitions of an error the network performs. In principle, all rules are variations of Hebbian learning. This shows that Hebbian learning is still the mother of many learning strategies.

Learning strategies may by classified for different criteria. There is

- supervised learning,
- reinforcement learning, and
- unsupervised learning (self organization).

Hebbian-, delta- and backpropagation-learning obviously belong to the category of supervised learning which means that there is a teaching instance that controls and supervises the learning process.

An example for unsupervised, self-organized learning are so-called Kohonen-networks (Kohonen 1988), which show a much stronger relationship to biological brains than perceptrons and Hopfield-networks, but could not be discussed in the framework of this paper.

7 Application Classes

In the following we will list some typical application classes, which could successfully be treated by means of neural networks. These are applications like

- Iris recognition (bio-metrical method for identification of persons, appr. 8-times more precise than a finger print)
- Balance sheet analysis (typical patterns of financial ratios)
- Acoustic quality control (acoustic control of porcelain and tiles)
- Data compression (for images and sounds)
- Process control
- Pattern recognition (see our example of image recognition)
- Vision and scene analysis
- Speech recognition
- Silicon Retina (electronic eye)
- Silicon Ear (electronic ear)
- Optimization
- Vision based inspection
- Autonomous robots

and many others.

7.1 NetTALK – A Neural System to Generate Speech

In the following section, I discuss an early neural application generated in the field of computer linguistics which had been presented in 1986 by Sejnowski and Rosenberg shortly after the development of the backpropagation algorithm. This system is able to read and pronounce written English text with the reading ability comparable to an eight year old pupil.

As is well known, the English spoken language differs to a large extent from the English written language. For native speakers, this deviation does not cause any difficulties. Being confronted with this task with regard to an implementation in a computer the transition from written language to spoken language turns out to be a problem.

The number of phonemes varies from language to language. German, for instance, comprises approximately 40 phonemes. The language which can be produced by means of the smallest number of phonemes of all existing languages is the Papua language which contains only 11 phonemes, six consonants and five vowels. At the other end of the spectrum the South African Khosian language !Xu contains a number of 141 phonemes.[13]

As opposed to English, there are far less phonological rules for German. Thereby, it is easier to pronounce an unknown word correctly.

The following example shows a context dependent deviation of a German phoneme.

Example: Milch (milk) phonetic transcription: /milç/ and
Buch (book), phonetic transcription: /bu:x/

The phonemes [ç] and [x] are in complementary distribution to each other, i.e. the respective phonemes only occur in particular phonological contexts. Thus, they are not to be considered different phonemes but allophones of the same underlying phonemes.

Languages do not only differ with respect to their number of phonemes but also with regard to their phonological rules. The phonological rules of English, for example, are more complex than those of German. The complexity of articulation for reading the English language can be easily demonstrated on an artificial neural level. NetTALK which was designed to simulate the reading process of English written text will be presented in the following. NetTALK is based on a multi-layer perceptron consisting of three layers with approximately 300 neurons in total. The network was trained by means of the backpropagation algorithm using a representative training set. NetTALK functions as follows. A mask is shifted over a written English text and it selects seven characters in a sequence including empty spaces and punctuation marks such as periods and commas. During the reading process, the mask is shifted over the text character by character focusing on the central grapheme (i.e. the forth grapheme) at a time. Exceptions apply for the beginning and the end of the text only because there is no preceding and no succeeding character, respectively. The strategy is based upon the consideration of the phonological context of the respective phoneme being under focus. The phoneme sequence has to be matched with the learnt phoneme patterns in order to infer to the pronunciation. The task is to deduce the phoneme which fits best to the reading situation in which conflicts may occur if there are several competing phoneme candidates.

On the basis of the described mechanisms the underlying structure of NetTALK can be derived nearly automatically. The input layer consists of 7 groups of neurons which contain 29 neurons in each case. Each group represents one character of the 7-digit

[13] Hall, T. Alan. (2000). Phonologie – Eine Einführung, New York, de Gruyter, p. 80.

string being currently under focus. The 29 neurons of each group of neurons represent all possible graphemes such as letters and punctuation marks.

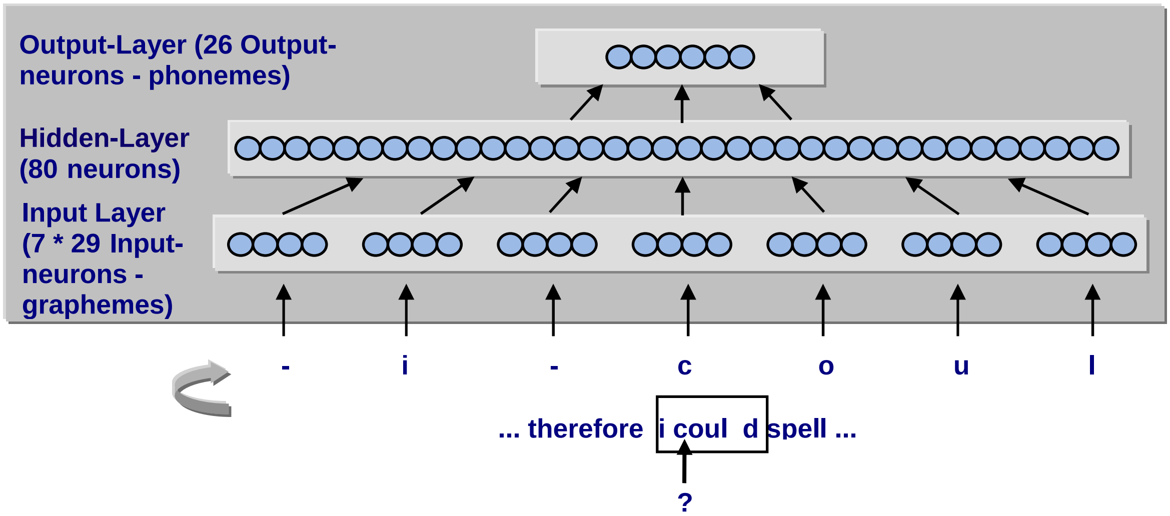


Figure 7.1 Network structure of NetTALK

In the next step, the phoneme which fits best to the central grapheme has to be determined on the basis of the phonological context. Concerning this, the output layer consists of 26 neurons; every single neuron corresponds to one phoneme. Think of a neuron as a bulb with a shunt (adaptive resistor). Each bulb may have states as “off” and “on” and all intermediate states. The bulb with the highest illuminance (intensity of light) is the phoneme candidate with the strongest evidence.

As mentioned before, the representative training set is an extremely important aspect. I want to illustrate this aspect by means of the example of the pseudo word “ghoti”. Intuitively, an English native speaker would pronounce this word as /go:ti. Another possibility, however, is to pronounce this grapheme sequence as /fiʃ/. Why that? Having trained NetTALK by means of a non-representative, one-dimensional training set such as {tough, enough, rough, women, nation, station, application, ...}, the pronunciation would result in /fiʃ/ due to

tough ↔ „f“

women ↔ „i“

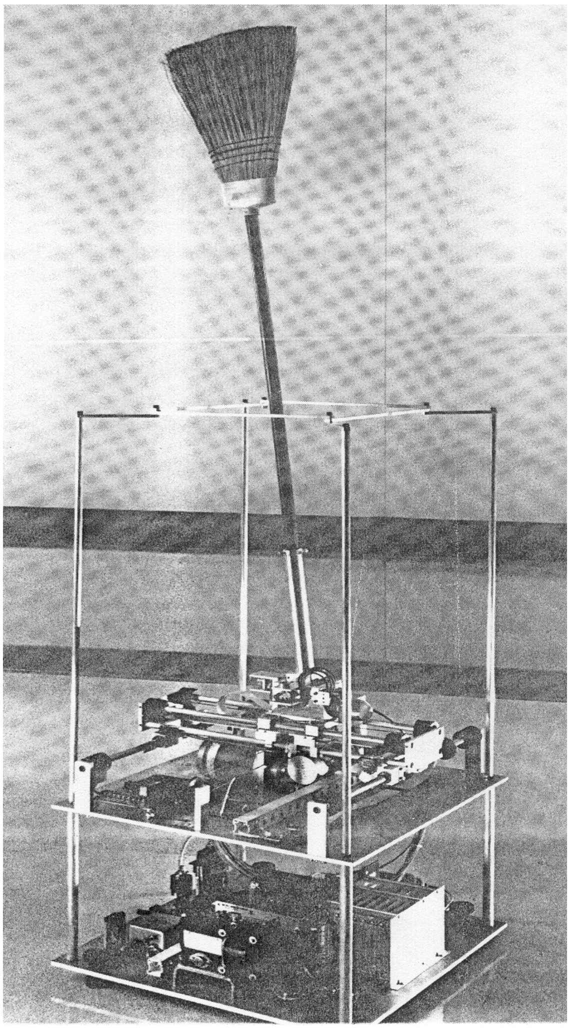
nation ↔ „ʃ“

This example illustrates sufficiently the outstanding importance of a representative training set. Some projects for the development of systems in neural architecture failed because the great importance of a well-balanced and representative training set was not taken into account sufficiently or simply because representative data were not available.

So it is unclear if there is any truth in the tenacious rumour that a neural network developed for the US Army and designed for the recognition of weapon systems on satellite images has learned to distinguish good weather from bad weather instead of recognizing weapons. Supposedly all satellite images taken for training purposes have been produced in good weather conditions almost without exception. Thereby it could easily have happened that unnoticed correlations have been learned. One never may be absolutely sure what a neural network has learned really.

7.2 A Neural Broom-Balancer

Finally, I would like to present an interesting application in order show that artificial neural networks can be used for solving typical everyday problems.



The Figure on the left hand side shows a so-called a "broom balancer" which should not be missing in any modern household. This machine is able to balance a broom even if it is inclined up to angle of inclination of 12°. This machine works properly if even a small pendulum is fixed on top of it.

From a formal point of view, such coupled pendulums are regarded as so-called dynamical systems which require a lot of calculus to be treated mathematically. For the neural broom balancer, however, the balance control does not seem to cause any difficulties.

Figure 7.2 A neural broom-balancer
This machine was exhibited over a longer period of time in the entrance hall of the Hecht-Nielsen Company, San Diego.

8 Summary and Conclusions

Distributed knowledge and massive parallel processing are essential to the underlying principles of biological brains. Though not even the function of a single neuron can be completely described and the processing of biological brains is largely unexplored, relative simple artificial neural models can be constructed realizing a distributed knowledge distribution and parallel information processing. Compared to their biological models current artificial neural network are still trivial. In spite of the substantial simplifications of the networks they are already able to yield impressive results. The great variety of artificial neural networks and numerous approaches show that neurocomputing, or connectionism, is still in its infancy. For some scientific disciplines, e.g. linguistics, connectionist models are a very controversially discussed topic and sometimes also totally rejected.

The still existing shortcomings and simplifications of current systems in neural architecture should not take away from the fact that the current neural network technology has considerably advanced the understanding of neural activity.

The principle of distributed representation approaches the notion of the mental representation on the neural level to a larger extent than any other known paradigm could do. The possibility of explicit modeling and simulation of neural aspects can contribute to the understanding of cognitive processes. However, AI-research is still far from being able to simulate the human brain. At this point in time in the research field, it is impossible to simulate biological brains in artificial systems due to the extraordinary high degree of complexity of the neural activities. As a consequence, the question arises which kind of objectives AI aims at. The motivation of neurocomputing is based on the goal of obtaining deeper insights into the structure and the organization of cognitive processes in order to be able to realize intelligent systems. The concept of the brain serves as the driving force causing the ambition of understanding just as the flight of the birds inspired human beings to construct airplanes.

On an underlying level, the internal representation of the external world is based on neural activations. As a consequence, the question comes up in which way neural activation patterns, a material substrate, is able to generate mental states of all kinds. The transformation from the neurophysiological level to the cognitive level, however, is still almost completely unexplained.

9 Literature

Anderson, John R. (1989). Kognitive Psychologie. Eine Einführung, Heidelberg.

Dorffner, Georg (1991). Konnektionismus. Von neuronalen Netzwerken zu einer „natürlichen“ KI, Stuttgart.

Engel, Andreas, König, Peter P. (1998). Das neurobiologische Wahrnehmungsparadigma. Eine kritische Bestandsaufnahme, In P. Gold & A. K. Engel (Ed.), Der Mensch in der Perspektive der Kognitionswissenschaften.

Hall, T. Alan (2000). Phonologie. Eine Einführung, New York: de Gruyter,

Hebb, Donald O. (1949). The Organization of Behaviour; Wiley, New York.

Hofstadter, Douglas R.(1989). Gödel, Escher, Bach - Ein endloses geflochtenes Band, Stuttgart.

Hopfield, John J.(1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities; Proc. Natl. Acad. Sci. USA, pp. 2554 – 2558.

Kohonen, Teuvo (1988). Self-Organization and Associative Memory; Springer-Verlag; 2nd Ed.; Berlin.

Lippmann, Richard P. (1987) An Introduction to Computing with Neural Nets. IEEE ASSP Magazine, pp. 4 – 22, April 1987.

McCulloch, Warren S.; Pitts, Walter. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics (5), pp. 115 – 133.

Minsky, Marvin; Papert, Seymour (1969). Perceptrons – An Introduction to Computational Geometry. MIT Press, Cambridge (Mass.), 1969

Minsky, Marvin; Papert, Seymour: Perceptrons: Expanded Edition, MIT Press, Cambridge (Mass.), 1988

Newell, A., Simon, H. A.(1976) Computer Science as Empirical Inquiry: Symbols and Search, Communications of the ACM, 19 (3).

Ritter, H.; Martinetz, T; Schulten, K. (1990). Neuronale Netze - Eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke; Addison-Wesley; Bonn.

Rojas, Raúl (2000). Theorie der neuronalen Netze. Berlin Heidelberg New York, Springer-Verlag.

Rosenblatt, Frank. (1958) The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psychological Review (65), pp. 386 – 408.

Rosenblatt, Frank (1962). Principles of Neurodynamics.

Roth, G. (2001). Fühlen, Denken, Handeln. Wie das Gehirn unser Verhalten steuert. Frankfurt a. M..

Rumelhart, David; Mc Clelland, James(1986). Parallel Distributed Processing Vol 1; MIT Press; Cambridge (Mass.).

Rumelhart, David; Mc Clelland, James(1986). Parallel Distributed Processing Vol 2; MIT Press; Cambridge (Mass.).

Schade, Ulrich (1992). Konnektionismus. Zur Modellierung der Sprachproduktion, Opladen.

Sejnowski, T.J.; Rosenberg, C.R. (1986). NETtalk: A Parallel Network that Learns to Read Aloud; Electrical Engineering and Computer Sciences Technical Report JHU/EECS-86/01; The John Hopkins University.

Singer, Wolf: Auf dem Weg nach innen, Beitrag in der FAZ-Beilage „Wissenschaft“ vom 27.2.1998, Nr.49, S. 41

Wiseman, S., Neisser, U.(1974). Perceptual Organization as a Determinant of Visual Recognition Memory. American Journal of Psychology, 87, S. 675-681.

Ziemke, A., Cardoso de Oliveira, S. (1996). Neuronale Repräsentation. Zum repräsentationistischen Forschungsprogramm in der Kognitionsforschung. In: A. Ziemke & O. Breidbach (Ed.) Repräsentationismus – was sonst? Braunschweig, S.1-34.

Zimbardo, Philip G. (1983). Psychologie, Heidelberg.