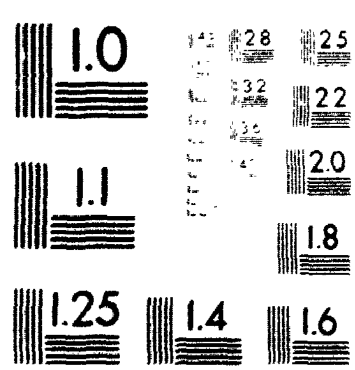
1

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT



PRECISION℠ RESOLUTION TARGETS

NEUROSOLVER: A NEURAL NETWORK BASED ON A CORTICAL COLUMN

by

ANDRZEJ BIESZCZAD

A thesis submitted to

the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Computer Science

OTTAWA-CARLETON INSTITUTE FOR COMPUTER SCIENCE

SCHOOL OF COMPUTER SCIENCE

CARLETON UNIVERSITY

Ottawa, Ontario

December 1992

ISBN 0-315-84104-4

The undersigned recommend to the Faculty of Graduate Studies and Research acceptance of the thesis

Neurosolver: A Neural Network Based on a Cortical Column

submitted by Andrzej Bieszczad, M.Sc.

in partial fulfilment of the requirements for

the degree Master of Computer Science

Thesis Supervisor

Director, School of Computer Science

Carleton University
December 14, 1992

Abstract

In this thesis, a neural network based on a biological cortical column is presented. The cortical column has been found to play the fundamental role in information processing in the cerebral cortex. The model of the column presented in this work displays similar functionality. The stress is laid on the cooperative behavior of many artificial columns interconnected in a network. The network is capable of recording trajectories of time-related events. Those recorded trajectories let the network use such time dependencies to perform breadth-first searches. The device can solve stimulus-response type problems in the given domain and because of that is called a neurosolver. The neurosolver can use a context update mechanism to perform dynamic searches. Two important features of the neurosolver, its generality and modularity, can be used to mimic hierarchical and, at the same time, parallel and distributed functionality of the cortex in an artificial environment.

Acknowledgments

I would like to express my gratitude to Prof. Wilf Lalonde for lending me Burnod's work *"An Adaptive Neural Network: The Cerebral Cortex"* that inspired me to devote my Master thesis to the neurosolver. I also wish to thank him for the patience he showed in reading the thesis and my other works and for very valuable and technically sound comments.

To my children, Maciek and Kasia, and to my wife, Anna, I would like to say that I am sorry for not always spending with them as much time as they would wish for.

Table of contents

Introduction

The human brain has always been a focal point for many thinkers who were enchanted by the mysteries of perception, memory, recollection, abstraction and reasoning. The first known reference to the brain comes from an ancient Egyptian surgeon who recorded a number of cases of head and neck injuries along with their analysis. He noted with a surprise that although the head was injured, several patients had their motor or sensory capabilities impaired. Those observations must have been not well received among the contemporaries of the surgeon, since for the next twenty centuries the cardio-centric hypothesis was preferred by philosophers. Even Socrates and later Aristotle subscribed to that theory, although Aristotle proposed many innovative ideas in his treatises on memory. Democritus, circa 400 BC, located thoughts in the brain that was using "psychic atoms" to communicate with the rest of the body. Hypocrites confirmed that thesis through thorough clinical observations. In the third century BC, in Alexandria, Erasistratus dissected thousands of bodies mostly of criminals given to him by the kings. That enterprise was quite cruel, since many criminals were alive when the experiments took place, but the encephalono-centric hypothesis was enriched enormously. Later, again the brain lost its position in the world dominated by catholic bishops. Most of the scientists and philosophers were preoccupied with the proper position of the soul in their treatises, and placing any aspect of the intellect in the brain was close to a blasphemy. It was only in the seventeenth century when the encephalono-centric theories were revived by Gassendi.

The modern history of the brain research, or neuroscience as that branch of science has been called for several years, starts with the names of researchers like Ramón a Cayal [27], Broadmann [5] and Lorente de Nó [19]. Ramón a Cayal proved the neuronal architecture of the brain. Broadman divided the brain's cerebral cortex into 52 areas based on the micro-anatomy of the cortex and assigned a function to each of them. Lorente de Nó put forward the principles of intra-cortical circuits of neurons. More recent and known names are those of Hebb [13] and McCulloch and Pitts [22] who contributed to the field of neural networks, although at the time they worked that branch of science did not exist. Over the last thirty years the knowledge and ideas of the early researchers have been enriched by scientists like von der Malsburg [20], Szentagothai, Mountcastle, Hubel [15], Wiesel [15], Kandel [16], Schwartz [16] and others and utilized by researchers like Rosenblatt [29], Widrow [36], Hoff [36], Kohonen [18], Hopfield [14], Grossberg [10], Rumelhart [21], Ballard [4] and many others who engineered devices that mimic the behavior of the brain. Their contributions are enormous. There are many excellent publications describing the past and the present of neuroscience, neural networks and cognitive science. A two volume collection of papers edited by James Anderson and entitled "Neurocomputing" [3] may be a good source of knowledge about those achievements.

In spite of that enormous and common drive to research the brain, thanks to which many discoveries have been made, many theories have been proved and many uses of the new knowledge have been found over several thousand of years, there is still a set of unknown to be explored by the current and future generations of scientists

In this thesis, the work on a special type of a neural network based on a model of a cortical column will be presented. The cortical column, the "module-concept", has been proposed as an anatomical entity by Szentagothai [31]. Later, Mountcastle [23] enriched the hypothesis by describing the functional context of the cortical column. Today, the cortical column is widely accepted as a basic functional building block of the brain's cerebral cortex. It has been observed in every part of the cortex, although some aspects of its internal architecture vary from area to area. The behavior of the column seems to be similar in every region of the cortex as is its connectivity pattern. Many neuroscientists believe that the cortical column is also a basic computational component of the brain.

The goal of this thesis is to duplicate the model of the cortical column that was proposed by Burnod [7] and to get better understanding of it through simplifications and modifications. The final model used in this work will be compared with original Burnod's model in Chapter 3 and Chapter 4.

It was an assertion for this thesis that the network in which the model of the cortical column is used should be universal and modular. The universality would make it possible (through the learning process) to customize the same network for uses in various domains, much like analogous VLSI chips can be customized for different applications. The modularity of the network would allow several small, specialized networks to be combined into a bigger, higher level network that could treat more complex problems. The network should be completely or at least regularly connected. Ultimately that architecture should be implementable in VLSI.

From the functional perspective, an attempt has been made to design a general purpose problem solver using the model of the cortical column. The network of cortical columns should be capable of computing solutions to problems in a given, well defined domain. The relationships between the objects in the domain are stored as temporal data during the learning process. The recollection mechanism produces solutions to presented problems in a form of sequences of state change patterns. The learning phase is not separated from the performance phase as is the case in almost all other neural network architectures. Instead, like in the human brain, the learning is a continuous process. The topographical relationships of the objects of the domain were used explicitly, because the problem of creating such maps has been settled by others, for example Kohonen.

To test the network, or the neurosolver as the network of cortical columns is called in this thesis, a workbench has been implemented in Smalltalk-80. The workbench includes the model of the cortical column and several variations of network architecture. The workbench includes a user interface with which it is easy to set the parameters and interact with the neurosolver, but most of all to observe its behavior. The neurosolver was also used in a simple application, a rat maze, to test some of its capabilities. The maze is also implemented in Smalltalk-80 with a proper user interface. The descriptions of both applications are parts of this work.

The author finds it valuable to present the basics of the functional anatomy of the brain, particularly the cerebral cortex, and micro-anatomy and connectivity of the cortical column before the description of the model and the architecture of the network. The description is far from being complete. For more details the user can consult one of the many good textbooks on neuroanatomy and neuroscience, for example by Schmitt [30] or Kandel and Schwartz [16].

CHAPTER 1

An insight into the human brain.

EVOLUTION OF THE BRAIN

The brain of a man, one of the most complex systems known, is the result of thousands or even millions years of evolution. Simple vertebrates had only a simple nervous tube, or spinal cord, that was sufficient for survival of the specie. That tube evolved to receive sensory information through various nerve fibers and send motor signals to contract animal's primitive muscles. It is shown in Figure 1.

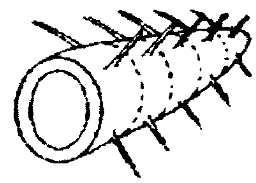


Figure 1. A primitive brain - a spinal cord.

Soon, it became apparent that it is easier to move in one direction rather than in many. It was more important now to know what was happening in the direction of the movement than in any other, so together with the evolution of the body there was also a specialization of the control system. The frontal, or anterior, with respect to the preferred direction of the movements, part of the spinal cord was receiving sensory signals that became more important than the signals from other parts of the body. That part refined its sensory capabilities and the brainstem, a precursor of the human brain, was born. The brainstem is illustrated in Figure 2 with its three parts marked.

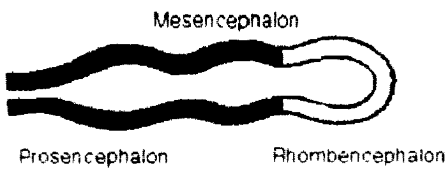


Figure 2. The brainstem - a precursor of the human brain.

New challenges from the changing environment induced further development in the motor and sensory systems. The brainstem grew, so the new functionality could be controlled in a better, more direct manner. The frontal part of the brainstem was enlarged by addition of two cerebral hemispheres. The back, or posterior, part evolved into the cerebellum. Those new additions are shown in Figure 3. The cerebral hemispheres acquired the functionality for data analysis and decision making, like movement origination. The function of the cerebellum was to coordinate the execution of the movement commands. All mammals have brains like that shown in Figure 3, although the development of certain functions, and the parts of the brain supporting those functions, varies among species.

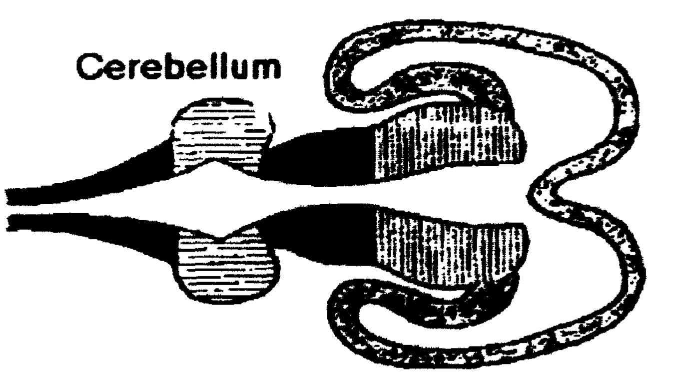


Figure 3. Mammalian brain.

Humans have the best developed brain among all mammals. The biggest change in the relative size and functionality came to the cerebral hemispheres. The cerebral cortex evolved into the control center with the most sophisticated functionality. It is no longer just a place where the facts are analyzed and actions generated in response. In the course of phylogenetical development, the relatively simple nervous tube evolved into the origin of intelligence.

ANATOMY OF THE BRAIN

Usually one thinks about the brain as being completely enclosed in the skull. In fact, however, the brain can be divided into three main parts (Figure 4), not necessarily located in the head:

- spinal cord,
- brainstem and
- forebrain.

The spinal cord is the lowest element of the brain that is also the oldest phylogenetically and still fulfills the same tasks of receiving the sensory signals from the body parts and sending the motor commands to the muscles as in primitive vertebrates. It is divided into a number of segments with each segment servicing more or less the part of the body at the same height as the segment. The sensory nerves that enter the spinal cord from the back (or top in the animals, therefore called dorsal or superior) provide sensory information. They make synapses with the neurons that will carry that information to the thalamus and further to the cortex in the forebrain (see below). The tracts carrying the motor signal connect to the muscle's neurons in the frontal part of the spinal cord (or bottom in the animals, therefore called ventral or inferior).

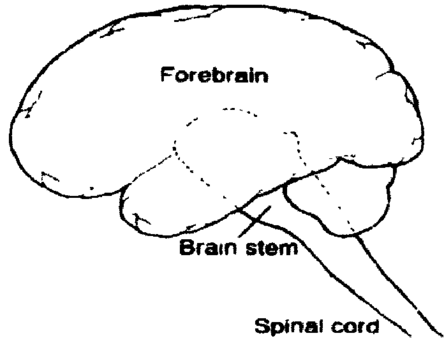


Figure 4. Three basic parts of the human brain.

The functionality of many parts of the brainstem has not been completely determined. Generally speaking, the brainstem handles the basic life supporting functions. In many animals, the brainstem provides the highest level of functionality, that in better developed species were overtaken by the forebrain, like vision and audition. There are several entities that are relatively well understood. The hypothalamus is involved in almost all aspects of behavior, like feeding, sleeping, sexual behavior, temperature regulation, emotion control and movement. The cerebellum is regarded as the center for equilibrium, postural reflexes and coordination of movements. Another interesting part of the brainstem is the reticular formation that has been assigned responsibility for consciousness, general arousal and controlling basic functions like breathing, heartbeat, body temperature, chemistry of the blood, etc.

The newest component in the evolution of the brain, the forebrain, and particularly the cerebral cortex, is the location of what we consider to constitute intelligence. The part of the forebrain and cerebral cortex will be described in the next sections in more detail, since they are more important as seen from the perspective of the work described in this thesis.

ANATOMY OF THE FOREBRAIN

The forebrain consists of the following five anatomical parts:

- neocortex or cerebral cortex,
- thalamus,
- basal ganglia and
- limbic system,
- olfactory bulbs.

Cerebral cortex

The cerebral cortex is the youngest brain structure and the relation of its size to the size of whole brain is the most visible difference between the brains of humans and animals. Because an average human is evidently more intelligent than an average animal, the neocortex has been thought to be the location of that value-added functionality. Many important experiments with animals and with humans confirmed that the cortex is the highest level center of information processing. Almost all sensory data are transmitted to the cortex where they are analyzed. The response to the stimulus that depends heavily on the past experience, memory, is calculated, and the signals initiating proper action are sent to the lower level brain structures that carry out the commands.

The nature of the information processing that goes on in the cortex is the main interest of this thesis. Therefore, the cerebral cortex will be discussed in detail in several further sections of this work

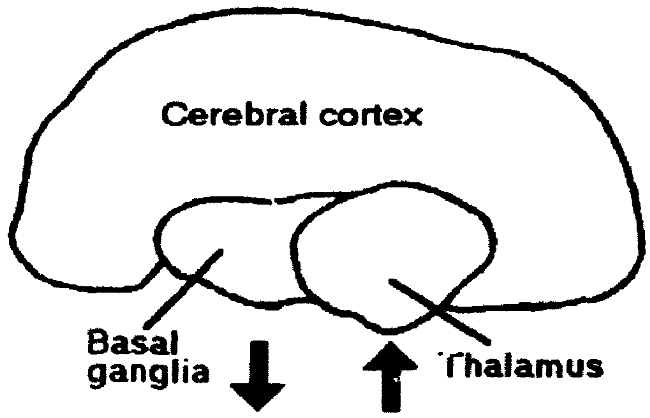


Figure 5. The thalamus and the basal ganglia: the input and the output of the cortex..

Thalamus

The sensory information, with the exception of smell data (see below), is sent to the cerebral cortex for analysis. The pathway, however, is not a direct one. The thalamus is the input switching system that passes the signals detected by senses to appropriate areas of the brain. Principally, the signals are directed to the primary somato-sensory areas, the visual areas, the auditory areas, etc. In addition, there are pathways to secondary and higher areas, but they have relatively small throughput. The primary, secondary and tertiary areas will be discussed later in this chapter.

The thalamus is not just a simple relay station. There are many reciprocal connections from the cortex, that carry signals that are used to modulate incoming input data. The data that is not important in a given context may be suppressed, and vice versa, a weak signal that the cortex indicates as essential may be amplified. The thalamus may even be instructed to expect certain input, so the proper action can be undertaken to diminish the difference between the expected and actual sensory data.

Basal ganglia

There are many basic functions of the motor system that are controlled by the spinal cord, like the spinal reflex which ensures that as one group of muscles contracts, the opposing ones relax. More complex movements are the responsibility of higher and higher levels in the motor system. The basal ganglia is a part of the motor system that is, in lower vertebrates, the highest level center for motor control. In humans and higher vertebrates, the basal ganglia is the source of multiple signals initiating and terminating partial movements. It uses inhibition and modulation to control lower level motor subsystems in the brainstem and spinal cord, ensuring that movements are smooth. It closely cooperates with the cerebellum, that is responsible for coordination of the movements.

The basal ganglia is the main relay station for nerve fibers going from the motor cortex to the subordinate subsystems.

Limbic system

Initially, the limbic lobe was thought to provide olfactory functionality, because of its connection to the olfactory bulbs. That theory has been abandoned as was the hypothesis that there is a single function that the system realizes. It is still a long way to go before all of the parts of the limbic lobe can be assigned a proper function. There is enough evidence, however, to state that the hippocampus that is a part of the limbic lobe is critical for memory storage, spatial organization, organization of movements, inhibition and learning.

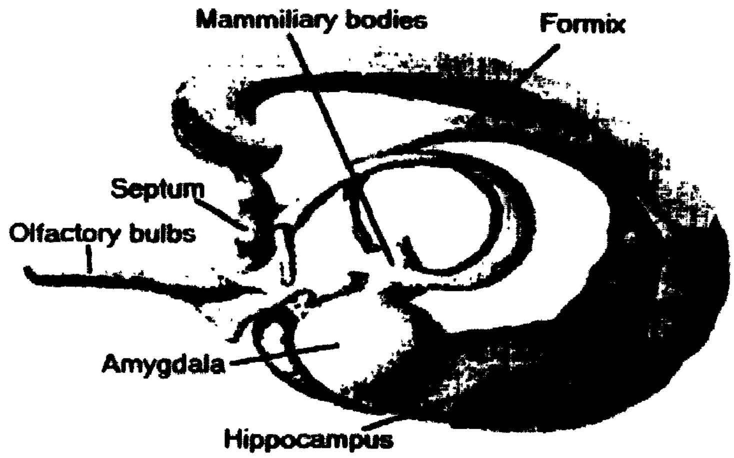


Figure 6. The limbic system and the olfactory bulbs.

The hippocampus has abundant connections with other parts of the limbic system. Many of the experiments showed that there are reward and punishment centers in the limbic lobe. Consequently, the data processed by the brain is being weighed in the hippocampus and the information considered worthy is stored in the memory. The data that is not important is discarded or stored only as long as it is needed for the current task to complete. There are no details, however, about the underlying mechanisms of that functionality.

Another part of the limbic lobe, the septum, seems to be responsible for emotions, something that usually is attributed to whole limbic system.

Olfactory bulbs

The sense of smell is considered to be the oldest sense that nas been mastered by the brain. That sense was very important in the evolution of vertebrates, since it could guide the animal to food and, as a result, increase its chances for survival. The olfactory bulbs that are part of the limbic system are responsible for the smell. It is very interesting, that the sense of smell is the only sense that is not handled by the cerebral cortex. It is probable, that the judgment centers developed in the limbic system because of the proximity to the smell center. Something that smelled good was worth remembering, while the opposite was best to forget. That might be the reason why the limbic system is so important to memory.

ANATOMY OF THE CEREBRAL CORTEX

The cortex is a thin (a few millimeters), convoluted tissue that covers all lower level brain structures located inside the skull. There is a layer of gray matter, so called because of the high concentration of grayish cell bodies, and white matter, that is located below the former, again so called because of the high concentration of white neuronal fibers connecting neurons in various parts of the brain. The space between the neurons and fibers is filled by glial cells. The function of glial cells is not definitely determined, but most of the researchers assign them some support functions, like nutrition and waste disposal. Neurons, interconnected by a network of axons and dendrites, are the fundamental information processing units.

A single neuron

There are many types of neurons in the central nervous system in general, and in the cerebral cortex in particular. The anatomy and functionality of all of them is, however, similar. One of the most important neurons in the cortex, the pyramidal cell, is shown in Figure 7. The pyramidal cell consists, as other neurons do, of the cell body or soma, an axon and dendrites. The axon and the dendrites are fibers that are attached to the soma. Usually, the axon is longer than the dendrites, because they carry signals from the neuron to other, sometimes very distant, areas of the nervous system. Axons make connections with somas of other neurons, with their dendrites and sometimes with other axons. The task of the dendrites is to capture signals from the axons in the neighborhood of the neuron and bring them to the soma. All signals are integrated in the neuron and the resulting integral determines the level of excitation or activity of the neuron. This activity is, in turn, transmitted over the axon to another set of neurons where the activity recalculation scheme is repeated.

The location where an axon make a contact with another entity is called a synapse. There are two types of synapses: symmetric, commonly viewed as inhibitory, and asymmetric, with excitatory nature. If an axon makes an inhibitory synapse, the activity carried over the axon will have a negative effect on the activity of the recipient. The excitatory synapse has a reverse effect on the receiving cell.

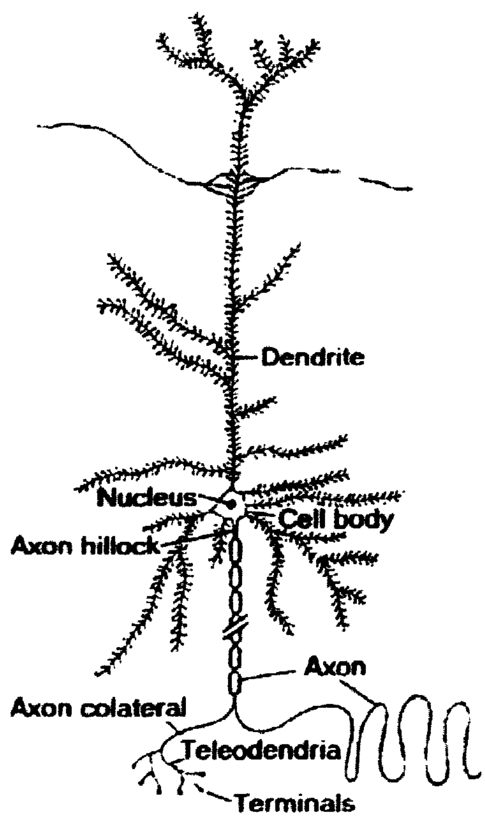


Figure 7. A neuron.

The topology of the connections that a neuron makes with other cells is determined by the genetic code. However, the synapses are highly adaptive devices, since their conductivity, i.e. the ability to transmit signals, may be modified. The extent of that modification depends on the pattern of activity in the pre-synaptic and post-synaptic entities. If there are certain regular changes in the patterns of activity of a number of neurons, and consequently, their synapses, then these changes may be encoded in the pattern of conductivity or strengths of the synapses transmitting signals between the involved cells. Later, the pattern of activity can be recalled even if only a part of the original pattern is known. The adaptivity of the synapses is the fundamental mechanism that the brain uses to store information and perform calculations. All artificial neural networks that attempt to mimic the circuitry of biological nervous systems make use of some kind of adaptive synapse. This computational paradigm is very often called connectionism.

The details of the communication between the cells, the transmission of the signals, the integration of the inputs and all related processes at the cellular and inter-cellular level are fascinating. They are however not directly related to the content of this work. One of the best publications on the subject is the book by Shepard [32].

Lateral organization

The cortical hemispheres

The cerebrum is a major mass of the brain composed of many milions of nerve fibers and covered with the cerebral cortex. The cerebrum consists of two hemispheres, as shown in Figure 8, divided by the medial longitudal fissure[1]. The hemispheres are connected by the corpus callosum, a tract of many nerve fibers that are used to exchange information between both halves of the cerebrum. The low level functionality of the hemispheres is similar, but they control opposite sides of the body: the left hemisphere controls the right

[1]Fissure - a cleft in the surface of the cortex that is deep enough to indent the brain ventricles.

side of the body, and vice versa. That arrangement is as surprising now as it was for Erasistratus. Ramón a Cayal [27] attempts to explain that phenomenon by the need for maintaining continuity of the visual image created in the brain from the signals sent by the sensors in the retinas. Coghill [8] suggest that this anatomical arrangement developed in the primitive vertebrates (he studied Amolystoma) that used a coil as their basic movement. The sensory neurons on one side of the body excite the motor neurons on the other side, so the animal can move away from any noxious stimulus. The olfactory balbs are not crossed, the only such system in the brain, because the animal wants to move in the direction of the food that it smells. There are other hypothetical explanations, but the issue is far from being setlled, though.

Figure 8. The cerebral hemispheres.

Psychologists believe that the hemispheres differ in the high level functionality that they provide. Although it may vary from one individual to another, usually the left hemisphere is a logical brain. The processing of information in the left hemisphere has a symbolic nature. Given a set of clues, the brain follows the logical paths that ultimately lead to the solution of the problem. On the contrary, the right hemisphere is an artistic brain that uses images rather than symbols. The processing of information in the right hemisphere has a global, associative nature. The sensations that are delivered to the right hemisphere are correlated, resulting in a recognition of shapes, colors, sounds, etc. The right hemisphere, however, has to contact the left side of the brain to attach proper symbolic labels to the recognized objects. The same happens in the opposite direction. If there is no obvious logical pattern to the resolution of the problem, the right hemisphere might be helpful by associating facts or sensations that are not linked by any logical reasoning.

The cortical lobes

Each of the two hemispheres of the cortex is divided into four[2] anatomical regions called lobes that are illustrated in Figure 9. Although these are anatomically defined areas, they are often used in conjunction with the functionality observed in the specific area. That is however far from being accurate. Much better reflection of the functionality of various regions of the cortex are the topographic maps described in further sections. The lobes are, however, convenient orientation terms that are used commonly by neuroscientists

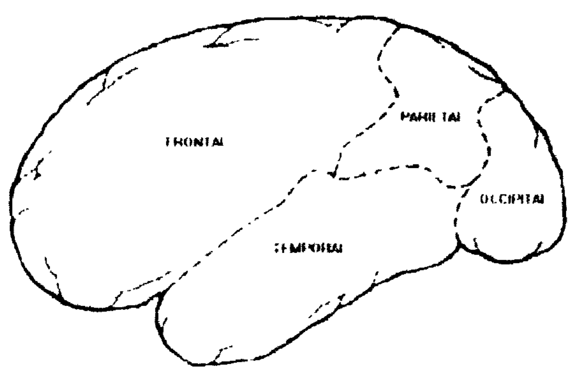


Figure 9. The cerebral lobes.

[2]Sometime anatomists count the limbic system as another lobe, bringing the number of lobes to five.

Laminar organization - The cortical layers

Neuroscientists divide the cortex into six layers, depending on how the neurons are arranged. Despite such a generalization, there are some variations in the anatomy of different parts of the cortex. In some areas of the brain certain layers might be thinner, almost non-existent, while others are thicker than average. Some researchers argue that there are more layers than six, and actually in the literature there are letters added to the layer numbers, like IVa, IIIb, etc., to indicate sub-layers.

There are certain functional attributes that can be associated with the cortical layers. For example, layer IV receives signals from afferents[3] coming mostly from the thalamus or olfactory system, and is therefore considered to be an input layer. This layer is very thick in primary sensory areas (see topographical maps below) and very thin in the motor regions of the brain. The layers below layer IV, send efferents[4] to the lower brain structures and are considered to be the output layers. That functionality explains why they are thick in the motor areas: the signals that are generated there must be carried to motor neurons that ultimately will cause contraction of muscles. The layers above layer IV make connections mostly with other cortical areas.

[3]The tracts that bring signals from somewhere else.

[4]The tracts that carry signals to other locations.

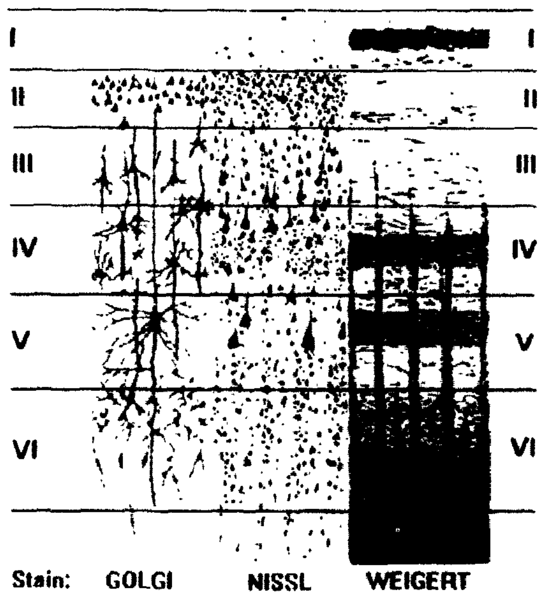


Figure 10. Cortical layers.

The detailed description of those interactions will be presented in Chapter 2.

TOPOGRAPHY OF THE CEREBRAL CORTEX

Since the times of Gall and his phrenology theory[5], many scientists performed extensive studies of various aspects of the cortex trying to divide it into more or less meaningful regions. In the course of those explorations, studies of the brain were published that described various characteristics of the cerebral cortex. There are three topographical maps of the brain that are widely used in the field today: functional, projection and cytoarchitectonic maps. Although the maps are based on different research methods used, the areas that were defined are quite similar. That seems to prove that there is a close relationship between the anatomy, connectivity and functionality of the brain areas.

---

[5]In short, phrenology was a theory that certain physical characteristics of the skull (like bumps) and indirectly the brain, can be the basis for assigning functions to the regions of the brain. Although today that theory sounds amusing, it was the first attempt to create a functional map of the brain, an endeavor that was attempted by many scientists later on.

Cytoarchitectonic maps

Cytoarchitectonic maps are constructed by examining the cells in every region of the cortex. Their type, density, arrangement, etc., is taken into account. The map that is used most often was created by Broadmann at the beginning of this century and one of its versions is shown in Figure 11. Broadmann examined various parts of the cortex without any overall plan, so the numbers that he assigned to the areas are a little chaotic. The numbers are used very widely to describe the location of a function realized by the specific part of the brain. For example, area 17 is commonly used as a reference to the primary visual cortex, while area 41 is a reference to the primary auditory cortex (see next section).

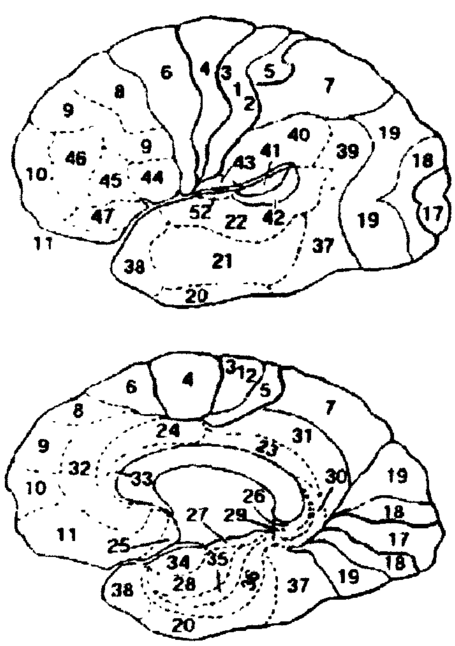


Figure 11. Cytoarchitectonic map of the brain.

Projection maps

Projections maps are created by tracing the efferents that leave the cortex and make connections with various subsystems in the lower structures of the brain and by looking for the source of the afferents carrying signals to the cortex. It has been noted that for each type of sense, there is an area in the cortex that receives most of the sensation signals carried by the axons of the neurons in the sensory system. Those areas are called primary sensory areas. The neurons from those areas send axons to secondary areas and from there to tertiary areas. That regularity led to the hypothesis that information processing in the cortex is hierarchical. Now, however, there is a clear evidence, that although there are obviously hierarchies in the processing of sensory data, there are also many additional pathways that contribute to the parallel nature of the overall functionality of the brain. Figure 12 illustrates the main projection maps and the pathways between the primary, secondary and tertiary areas. The primary areas are black and the secondary areas are gray. The arrows from or to the secondary areas indicate the tertiary regions. It is worth noting, that while primary areas are highly uniform in the character of the incoming signals, the other areas are more and more inclined to process signals with diversified origins. They are called associative areas (particularly the tertiary areas), since they correlate signals from many sources.

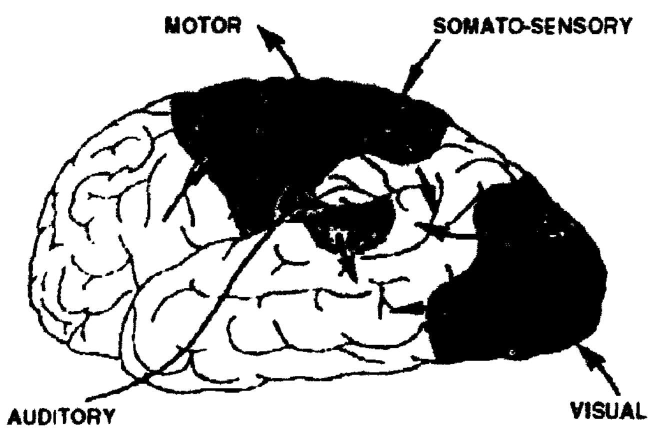


Figure 12. Projection map of the brain

Functional maps

The functional maps are drawn taking into account the data obtained by one or more of the following experimental techniques:

- stimulating cortex areas and registering what behavior is induced,
- stimulating cortex areas and registering what sensations are reported by the subjects,
- recording the activity of the cortex while performing certain behavioral tasks, or
- observing the changes in the behavior after cortical damage.

Some neuroscientists use animal lobotomy to register changes in the animal behavior. That technique can hardly be classified as scientific, since it is as cruel as the experiments of Erasistratus.

The best known functional map of the cortex is the one prepared by Penfield [24] and his coworkers at the Montreal Neurological Institute. They were created by examining cortices of patients awaiting brain surgery under local anesthetic. They noticed that each part of the body responds to a stimulus applied to a specific location of the cortex. What is more, the topology of the body is preserved in that mapping, although relative proportions of the areas corresponding to various parts of the body disagree with the differences between their actual physical sizes. For example, there is a relatively large area of the cortex devoted to hands and fingers as compared to the area controlling the trunk. That is easy to explain. There is much higher degree of precision required from the hands and fingers than from the torso, so more resources must be devoted to command hands.

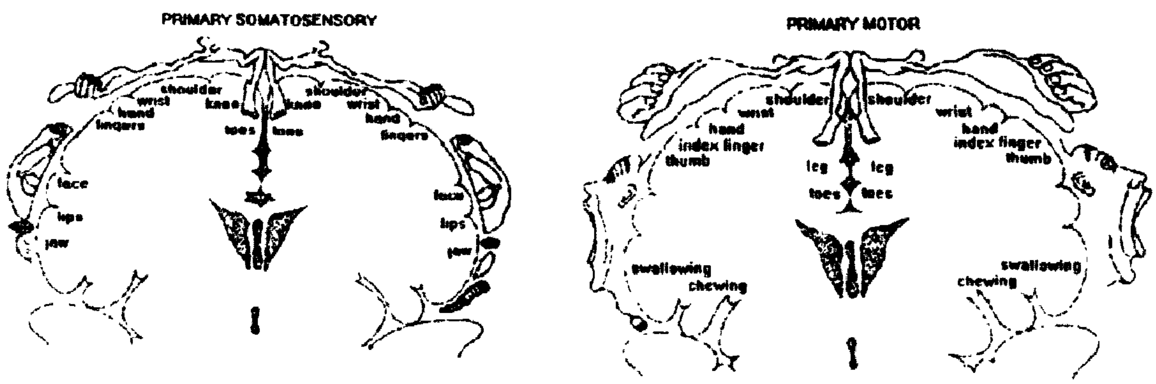


Figure 13. Homunculi - functional maps of the somatosensory and motor areas of the brain.

Similar mapping of the body parts has been discovered by recording the sensations that were reported by the subjects upon a stimulus being applied to the cortex. The areas of the first mapping (stimuli ⇒ movement) are localized in the anterior[6] part of the parietal lobe more or less along the postcentral gyrus[7]. The areas of the second mapping (stimuli ⇒ somatic sensation) are located in the posterior[8] part of the frontal lobe, along the precentral gyrus.

Figure 13 shows a popular illustration of Penfield's observations in a form of homunculi, little men, that are more or less symmetrical with respect to the central sulcus[9]. That symmetry is as important to the control of movement as the topology preserving nature of the mappings. Both areas, motor and somatosensory areas, exchange information during voluntary movements, so the part controlling the movement (motor areas) is

[6]=frontal

[7]Gyrus - a ridge on the surface of the cortex.

[8]=back

[9]Sulcus - a cleft in the surface of the cortex that is shallower than a fissure.

aware of the progress in the movement (as reported by the sensors sending signals to the somatosensory areas).

The region of the cortex responsible for speech has been discovered in the previous century by Broca [6]. That area is today called by his name. The low level speech understanding was discovered by Wernicke [34] and is as well called by the name of the scientist. Those areas are located in the medial[10] superior[11] part of the temporal lobe (Broadmann area 41, 42). The occipital lobe is the part of the brain were visual information processing takes place (Broadmann area 17, 18, 19). That function was assigned for the first time by Holmes [13] who was treating soldiers with head injuries in the WW1 and noticed that injuries to different parts of the occipital lobe cause blindness in different parts of the eye.

GENERAL FUNCTION OF THE CORTEX

The functionality of the cortex is still not well researched. However, the following three types of computation that the cortex performs, comprises most of the cortex functions:

- control of voluntary movements,
- pattern perception,
- cognitive mapping.

---

[10]=middle

[11]=top

Control of voluntary movements

The cortex receives the sensory signals from the thalamic afferents. That information is the basis for determining the responses to the observed state of the environment. Any complex action that is a part of that response is planned in the cortex. Every item is the sequence, the schedule of the response, is sent to the lower brain structures for execution. The functionality of some of those underlying structures was discussed briefly earlier in this chapter. The cycle repeats itself when the sensors notice the change caused by the response, and again report that state to the cortex. Through that feedback, the cortex ensures that not only a proper action is undertaken, but also that it is carried out accurately.

In tests on animals with the cortex removed, the ability to generate movements, as complex as walking, eating, drinking, mating, is not lost. However, they are not able to perform any of those movements as a part of an overall goal. They may carry food, but they will not hoard it. They may perform some elements of grooming, but they do not really groom.

Pattern perception

The sensory system is bombarded with a variety of bits and pieces of information about the environment. That information is correlated in the cortex. The cortex is the place where the nature of the observed changes is comprehended. Without that comprehension it would not be possible to react properly to the incoming data, since the same singular symptom can be a part of many, sometime quite different, syndromes.

Animals without the cortex recognize distinct signals like place and intensity of light, but are not able to differentiate between the patterns that use those signals.

Cognitive mapping

The question about the localization of the memory has not been answered yet. The fact is that many things can be learned without the cortex. However, the cortex provides an additional functionality. Cognitive maps that are constructed in the cortex during learning, can be used later to perform actions that were not learned.

Decorticated animals perform as well or almost as well as healthy ones on tasks like classical conditioning, approach learning, cue learning, etc. However, they are not able to learn dependencies that are distant, and not a part of the learned sequence. If a healthy rat learns to run from the south side of a maze to the north side, it will reach the north side even if positioned in the west. Clearly, it had built a map during the learning that it uses later on. That map is located in the cortex, since that ability is absent in rats without the cortex, although they can learn the path from the south side to the north side.

Cognitive maps are constructed very fast. Sometimes it takes only one or two trials. The speed suggests that built-in neural connections are utilized.

Functional organization of the cortex - Luria's model

There has been a lot of evidence collected since the nineteenth century that the posterior cortex receives signals from the sensory systems and is generally more concerned with the sensory function than the anterior cortex. It has also been proved through various experiments and observations of humans with cortex lesions, that the anterior cortex is responsible for the motor system, and much less responsible for sensory processing. The interactions with the sensory systems are rather indirect through the modulating influence that is procured by the cortex efferents connecting with the thalamus and other sub-cortical structures. A hierarchical model of the cortex function has been built that involves three types of the cortical areas:

1. Primary areas, sensory and motor.

2. Secondary areas, sensory and motor.

3. Tertiary areas that are called associative.

Many researchers proposed theories based on this basic, hierarchical model. Luria's model seems to be a good example. Although it has been proved since then that a purely hierarchical theory of the cortex function is incorrect, Luria's model is a good top level description of the processes and pathways that constitute overall cortex functionality.

In Luria's model, the cortex is divided into two basic parts with respect to the performed functionality. The posterior part, that includes the occipital, parietal and temporal lobes is the sensory unit. The anterior part of the cortex is the motor unit. It comprises only one lobe: the frontal one.

Figure 14 illustrates the sensory cortex with sub-regions representing the primary, secondary and tertiary sensory units (graded shadowing). The signals from the thalamus arrive to the primary areas (most dark) where they are analyzed with respect to their location, intensity and patterns of activation. To properly fulfill that task, the primary cortices are organized in arrays or maps representing those characteristics. The homunculus that was presented earlier is an example of such a map in the somatosensory area. The results of the primary analysis are transmitted to the secondary sensory areas (medium gray) where sensation is consolidated, but still retains its modality. The integration of various types of sensations occurs in the tertiary, associative, areas (white). Luria believed that the sensations are assigned their symbolic meanings in those areas and, consequently, is the area where the abstract thinking starts.

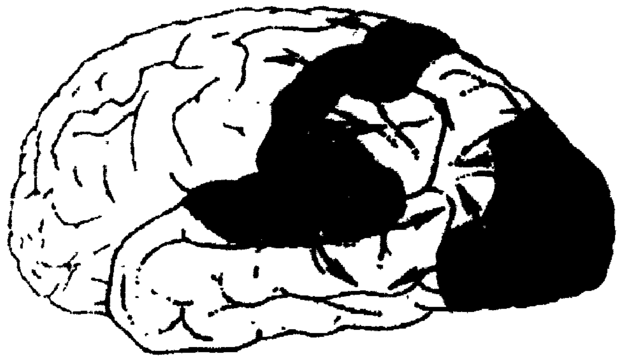


Figure 14. Luria's model - The sensory cortex.

The motor cortex is illustrated in Figure 15. The direction in which the signals are passed from one area to another are reversed here; i.e., the tertiary area sends efferents to the secondary area and the secondary area, in turn, sends the signals to the primary area. The tertiary motor area is the highest level structure in the brain. It is the area in which where intentions are created. Those intentions are translated into complex behavioral patterns that are divided into singular actions in the secondary areas (pre-motor cortex). The primary motor cortex has an organization similar to the primary sensory cortex. It similarlycontains topographical maps (like that illustrated by the motor homunculus). Those maps are used to direct the actions into the proper locations with the proper intensity.

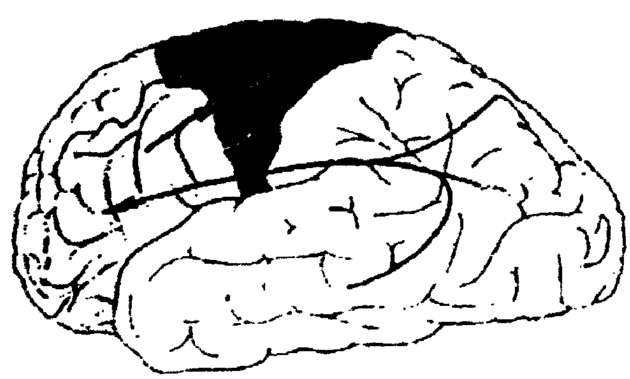


Figure 15. Luria's model - The motor cortex.

The problem with the Luria's model is not that it is wrong, but rather that it is not complete. As new research techniques and tools were designed, more and more data was discovered that contradicted the pure hierarchical model. There are many cortical connections that do not fall into any hierarchical pathways. That mixture of the traditional hierarchical view and new parallel extensions to that model underline the complexity of the function performed by the cortex.

There have been many books written about the brain that include descriptions of the cortex. Some of them are listed in the bibliography; e.g., [16], [30] and [18]. The reader seeking more details is referred to one of them.

CHAPTER 2

The cortical column

In Chapter 1, the laminar organization of the cortex was presented. This organization is based on the types of neurons that can be found in the cortex and their arrangement. The same criterion was used by Broadmann to divide the cortex into many distinct regions. In this chapter, a closer look at the cytoarchitectonic structure of the cortex is taken. The researchers discovered that although various areas of the cortex can be distinguished, there is a certain characteristic micro-arrangement of a number of neurons that is common to all of them. That arrangement is called the cortical column. In the course of many experiments, it became obvious that the cortical column constitutes a second fundamental level[12] information processing unit. The architecture, behavior and computing capabilities of the cortical column and a network of cortical columns are discussed in this chapter.

TYPES OF NEURONS IN THE CORTEX

Many types of neurons have been found in the cortex. Some of them are included in the illustration in Figure 16. The most characteristic ones are pyramidal cells, easy to

[12] A neuron is the first level organizational unit.

recognize by, and called after, the pyramidal shape of their somas. Other neurons are more difficult to classify, and they are often referred to by several common terms: stellate cells, interneurons, non-pyramidal cells. They might be bipolar, multipolar or bitufted with respect to the shape of the tree of their dendrites. According to the concentration of spines on its dendrites, an interneuron may be spiny or smooth. Usually, interneurons do not send axons outside the area in which their body is located. Many of the interneurons have been given names after the shape of their dendritic or axonal trees.

The pyramidal and non-pyramidal neurons will be described in greater detail in the following few sections.

Pyramidal neurons

Pyramidal neurons usually have their somas in layers 2 and 3 or 5 and 6. Characteristic for a pyramidal cell is not only its soma, but also its apical[15] dendrite. The apical dendrite usually crosses several cortical layers. That design gives the pyramidal neuron the capability to integrate a variety of inputs specific to different cortical layers. The cells that have their somas in layers 2 and 3 have shorter apical dendrites than the cells with their somas in layers 5 and 6. Both types of pyramidal neurons, supragranular - those with their somas above the layer 4 that is called granular - and infragranular - with somas in the layers 5 or 6 - receive only excitatory synapses. The main sources of input signals to pyramidal neurons are afferents incoming from other neural structures or other cortical areas. The signals can be transmitted directly by synapses with the afferents or axons on the somas and dendrites of the pyramidal cells or indirectly through other cortical neurons.

[15] That is oriented perpendicularly to the surface of the cortex.

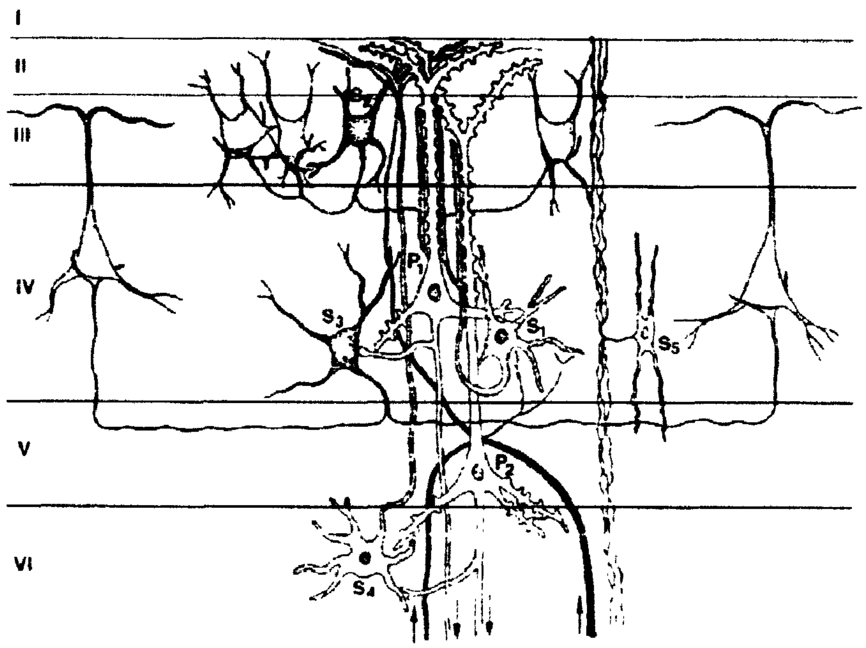


Figure 16. The most important cortical neurons.

It has been demonstrated that the axons of pyramidal neurons constitute the main output of the cortex. Using various tracing techniques many researchers found that the axons of the infragranular pyramidal cells usually project to other, lower, neural structures. They carry activation signals for actions that are carried out by the appropriate subsystem. For example, they can activate a movement by sending signals to the basal ganglia or even further to the spinal cord. The supragranular pyramidal neurons have been found to send their axons usually to proximal[14] or distal[15] pyramidal neurons in the cortex. Those axons constitute main cortical pathways that distribute activation throughout various areas of the cortex in order to integrate and correlate different aspects of the perceived

[14]=close

[15]=far

phenomenon. That columnar activity is internal to the cortex and represents abstract thinking, i.e., reasoning without the involvement of any physical resources.

The cortex grows starting with the lower layers, so it may explain why the pyramidal cells in the lower layers fulfill more basic functions providing control for the lower level subsystems. When the higher layers are added to the cortex, the extra resources can be used on more sophisticated functionality. That is why the supragranular pyramidal neurons can migrate toward other cortical regions creating in that way framework for thinking.

Spiny-stellate neurons

It has been demonstrated through tracing experiments that layer 4 of the cortex is the location where the most of the thalamic efferents terminates. Layer 4, a layer that is said to be granular because of the high density of various interneurons, is commonly considered to be the input layer of the cortex. That layer is very thick in the primary sensory areas and very thin in other regions that do not have direct input from the thalamus.

Spiny stellate cells (cell labeled S_1 in Figure 16) are found exclusively in the middle layers (mostly in layer 4) of the cortex. Many neuroscientists believe that they are, together with the pyramidal neurons, the main targets of thalamic signals. Spiny stellate cells, however, span their dendritic trees locally with respect to the laminae and the cortical area. The axons of spiny stellate neurons have similar, local, ramification, but in addition to layer 4 they may project to adjacent layers 3 or 5. Due to the property of limited localization, spiny stellate interneurons are not integrating units.

Spiny stellate cells have excitatory nature. They project to the somas or apical dendrites of pyramidal neurons and other interneurons. This organization is conducive to processing input data.

Cortico-cortical tracts may also terminate in layer 4. The term "input", therefore, includes both types of input: external - thalamic - and internal -cortical.

Chandelier neurons

A chandelier neuron, whose dense pattern of axonal ramifications resembles candles of a chandelier, is mostly found in layers 2 or 5. The cells with their somas in layer 5 send their axons up, and vice versa, the cells with their somas in the supragranular layers send their axons down.

The behavior of chandelier cells is inhibitory. They synapse with other neurons, mostly pyramidal and spiny stellate, usually in the vicinity of the initial segment of the output axon. For that reason, they are thought to be gating the output from the post-synaptic neuron.

Chandelier neurons can be found mostly in sensory areas.

Basket neurons

Basket interneurons (cell labeled S_2, S_3 in Figure 16) are so called because they have axons shaped in the form of a basket around the cell bodies and proximal dendrites belonging to pyramidal cells. Their somas are usually in layers 3 and 5. They project horizontally, but also cross several layers before they reach the destination. Basket cells receive signals from collateral axons of pyramidal cells.

Basket cells exhibit an inhibitory behavior. They may, therefore, suppress the activity in pyramidal neurons in the neighborhood while the activity in the region is high. Such behavior enhances the contrast between neighboring groups of cells.

The density of basket neurons is relatively high in the motor areas of the cortex.

Double bouquet neurons

The somas of double bouquet interneurons (cell labeled S_5 in Figure 16) are found mostly in supragranular or granular layers. Their axons form dense vertical branches directed down and up. That gave them their name - the axonal branches look like bouquets.

Double bouquet neurons are inhibitory, but because the preferred targets of their axons are other inhibitory cells, their overall role is dis-inhibitory. Their role might be a modulation of cortical activity across the layers.

Bipolar neurons

Bipolar cells are similar to double bouquet neurons, because of the dendritic and axonal trees that have as well preferential vertical orientation. They receive signals from supragranular pyramidal cells.

Bipolar neurons are excitatory. Hence, their task might be to move high activation from the supragranular layers to the infragranular ones.

Smooth stellate neurons

Smooth stellate neurons are inhibitory. Their somas have been detected in many layers of the cortex, but mainly in layer 4. These interneurons are multipolar and often are called short axon or local circuit cells, because they usually project to the same or neighboring cortical layers.

Smooth stellate neurons are considered a general processing elements, because they synapse with wide variety of other neurons. They can connect to somas, dendrites, spines and axons of pyramidal and non-pyramidal cells.

THE CORTICAL COLUMN

In the course of their experiments, neuroscientists noticed that reaction to a stimulus is highly localized in the cortex. They observed that the response to a sensory input is independent of the depth of the placement of the electrodes that detect and measure the activity of the cortex. Slight differences in the promptness of the response have been observed. When the electrode is placed in the granular layer 4, there is the shortest delay between the stimulus and the response. That is in agreement with the earlier observations that layer 4 is the main input of the cortex. Efferents connect directly to the somas and dendrites of the receiving cells. The supragranular layers get activated next. There are fewer direct connections between afferents and the cells in those layers, so the activity is carried over the interneurons. The additional synapses that are required are the reason for the delay. The infragranular layers are activated last. One reason is that they have only their apical dendrites in the proximity of the afferents, while the supragranular cells have their somas and basal dendrites closer to the terminals of the axons of the input tracts with their large surfaces more inclined to make connections. Additionally, spiny stellate cells have their axons usually projecting upward, and that favors the upper cells as the targets of the propagation of the activity. The fact that the infragranular layers are activated last in the cortex, underlines their role as the output zones. Axons of the pyramidal cells carry the integrated signals to other regions of the cortex or to other neural structures.

Another important observation is that if the electrode is moved a little bit, it detects activation that is the response to possibly completely different stimulus. It has been proved that neighboring groups of neurons that span the area of the cortex with a diameter of 0.5 to 1.0 mm respond best to different stimulus. Little activity is spread laterally, i.e., high activity in one mini-circuit does not cause a change in the activity of its neighbors.

These findings suggested that there is a functional unit that is vertically oriented crossing all cortical layers. Szentagothai was the first to put forward that idea. He called the mini-circuit of interconnected cortical neurons that respond in a uniform way to a stimulus a cortical column. Since then, there have been many followers, and currently the existence of a functional unit that retained the name given it by Szentagothai [31] is a commonly approved fact.

In Chapter 1, the topology preserving mapping that occurs in the cortex was described. The columns are an integral part of that phenomenon. The columns of a specific region responsive to a particular sensation may represent different aspects of the perception. For example, there is a region in the somatosensory cortex responsive to stimuli applied to a hand. The columns in that region are activated by the sensory information coming from sensors of various types. There are columns that respond to tactile data, temperature, position of the joints, etc. Such distant sensations do not fit very well with the isomorphic mapping scheme that represents whole body as one image in the cortex. However, recent studies show that there are many representations of the body. Some projections can be repeated many times in various regions, implying that the particular sensation space that they map is a component of many more complex perception spaces. Such a two-tier organization can be explained by the need to represent a multidimensional world on a two-dimensional plane, the cortex. There have been, unfortunately, only a few areas in which spaces of the first and second level indices are clearly defined.

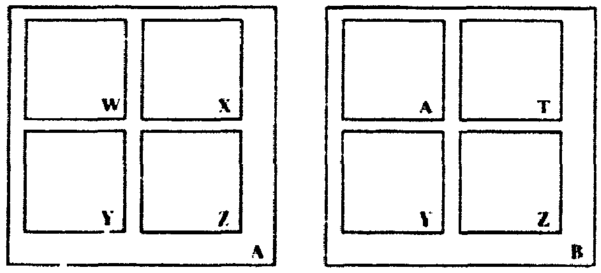


Figure 17. Two-tier organization of the topology preserving maps in the cortex.

Figure 17 illustrates the two-tier organization of the above topology preserving maps (after Ballard [4]). Two regions with the first level index spaces A and B contain several regions with the second level index spaces. Some of those sub-regions, namely Y and Z, are repeated in A and B. This implies that a sensation that can be indexed by Y is a component of two different compound perceptions. As illustrated by A in the figure, an entity that maps to a higher tier in one location may be mapped to a lower lever tier in another region. From the perspective of this thesis, there are two important things worth noting:

- there are uniform functional units -cortical columns - in the cortex and
- a functional map, in fact a network of a number of interconnected columns, may be used for various purposes.

The composition of cortical columns in various parts of the cortex, particularly between its motor and sensory parts, might differ. There are, however, enough similarities that allow the degree of generalization that has been presented in here.

CONNECTIVITY OF THE CORTICAL COLUMN

The role of the basket cells enhancing the contrast between neighboring columns through inhibitory connections with pyramidal cells belonging to other columns was described before. That aspect of the connectivity pattern of the cortical column is a vital part of the capability of the cortex to self-organize itself into feature maps.

The remaining patterns of connectivity of a column with more distal entities are summarized in Figure 18. The figure shows a cross-section of the cortex with images that are generated by three types of staining, i.e., marking the elements of the cortex so they can be photographed. There are some differences in the connectivity of the cortical columns in the motor and sensory areas that are disregarded here. The same degree of generalization is made as it was done with the cortical column itself.

The afferents arriving to a cortical column usually have their terminals in layer 4. The medial section of the cortical column is, therefore, considered to be an input interface. There are two sources of input signals incoming to the interface: cortical - from other regions of the cortex, and thalamic - from the thalamus. The cortico-cortical afferents can interconnect proximal or distal areas of the cortex. They include connections between the two hemispheres of the brain, called callosal afferents.

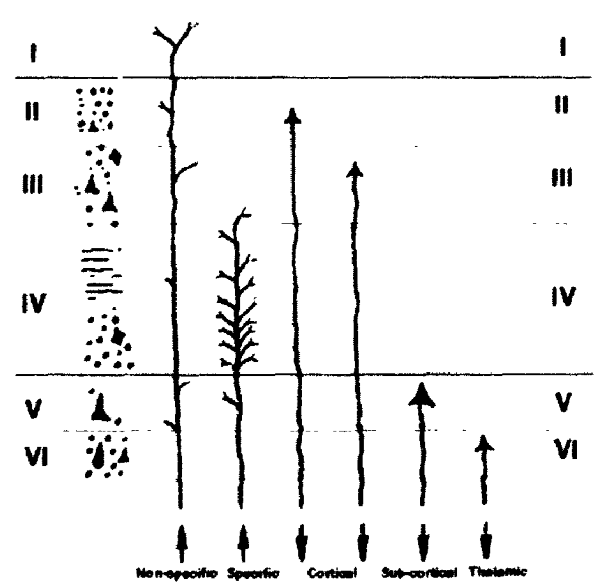


Figure 18. Types of efferents and afferents of a cortical column by layer.

The activity is distributed across the whole column through the internal interneuron connections. The activity is easily transmitted to the supragranular pyramidal neurons in the upper part of the column because of the favorable location of their somas and the dendritic trees. Higher levels of input activity are required to excite the infragranular pyramidal neurons in the lower part of the column.

It is important to indicate that the input activity of a column is a sum of the cortico-cortical and thalamo-cortical afferents. Consequently, the activity of a column depends on the sensations induced by the external stimuli as well as on the internal processes occurring in the cortex.

The pyramidal neurons were described as the output cells of the cortex. They fulfill the same role with respect to the cortical column to which they belong. Generally, the distance to which a pyramidal cell projects depends on the laminar position of its soma in the cortex. The deeper the soma is located, the further the axon transmits the neuron's activity. That arrangement can be explained by analyzing the ontogeny of the brain and the cortex. The growth of the cortex begins with the lower layers and the neurons in those layers are used to accomplish the most basic functionality. That requires long distance connections that can reach as far as the spinal cord. The level of the functionality that is still unrealized increases when the higher layers of the cortex start to develop. The more complex the functionality provided by a specific neural system is, the closer to the cortex that subsystem is located and, as a consequence, the shorter the length of the efferents. Finally, connections between cortical areas are sprouted from the higher cortical layers.

Hence, a cortical column has two output interfaces:

- cortico-cortical efferents, realized by the axons of the supragranular pyramidal neurons in the higher section of the column and

- cortico-sub-cortical efferents, realized by the axons of the infragranular pyramidal cells in the lower sections of the column.

The cortico-cortical connections are considered to be the basis for the higher level functionality that is usually associated with abstract thinking and problem solving. Spreading activity across the cortex can be considered an attempt to correlate asynchronous input data with the information stored in the long term memory. If the activity is low, it might be sufficient to activate only the supragranular pyramidal neurons of the receiving column, so the information is processed without any external output.

The cortico-sub-cortical efferents carry the signals that represents commands to be executed by the lower level neural subsystems. There are, for example, axons that terminate in the basal ganglia or the spinal cord and the signals carried over those connections trigger more or less complex motor actions. The level of excitation of the infragranular pyramidal neurons represent an overall state of the column, because of their integrating characteristic described before.

The cortico-sub-cortical efferents include also the cortico-thalamic efferents that are used to modulate the sensory data that are being transmitted via the thalamo-cortical afferents to the cortex. The cortex may control the level of the input activity by enhancing the sensitivity to the sensations that are being considered important or suppressing the signals that are regarded to be a noise.

THE CORTICAL COLUMN AS A PROCESSING ELEMENT

The cortical column is an elementary information processing unit of the cerebral cortex. It receives an input that comes to the granular layer and is distributed vertically across the column: first to the supragranular, later to the infragranular layer. The level of the input activity that is required to activate the upper pyramidal neurons is lower than the level required to activate the infragranular cells. The activity in the higher sections of the column represents a concept or feature with which the column is associated. When the pyramidal cells in layers 2 or 3 are activated, they can excite other columns, also with some well defined representation, via the cortico-cortical efferents. In that way, various features, concepts, aspects of the sensation, etc., can be correlated. If several columns projects to the same column, the integrated input activity might be sufficient to not only activate the upper pyramidal neurons, but also the lower pyramidal cells. That might trigger an external action that will, in turn, change the overall pattern of activity in the cortex through the feedback mechanism.

The following three states of a cortical column can be defined on the basis of the observations presented so far:

- no, or very low, activity, suggesting that the feature or concept associated with the column has neither been detected by the sensors nor recalled from the memory (definitive NO),
- activity only in the higher sections of the column, suggesting that the feature or concept has been recalled from memory due to some internal processes (MAYBE), and
- activity in the whole column, suggesting that the feature or concept has been confirmed as present in the current context (definitive YES).

In each of those states, the column may receive input signals from other cortical regions or the thalamus. The columnar output is completely passive in the NO state. While in the MAYBE state, the column sends signals only through its cortico-cortical efferents, because the activity is too low in the lower section of the column. The YES state results in the firing of the infragranular pyramidal neurons. In addition to the possibility of inducing an external action, high activity in the lower layers may activate inhibitory interneurons that suppress the activity in the column. If a partial activation of a column is considered as an anticipation of the presence of a feature or concept, or as a goal, the phenomenon that shuts down the activity in the column after the column is completely activated can be treated as a neural implementation of goal satisfaction.

PROPERTIES OF A NETWORK OF CORTICAL COLUMNS

The cortical columns through their efferents and afferents constitute a network of interconnected processing units. Some of the properties of the cortex were described in Chapter 1. Those properties were related to the capability of the cortex to organize into topographical maps. The maps in the sensory cortex reflect the physical world as perceived by the human sensors and communicated to the cortex via the thalamo-cortical connections. On other hand, the maps that are created in the motor cortex are organized according to the layout of the body parts that they control. The motor cortex includes the frontal areas that are considered to be the location of the cortex where the most abstract processing occurs. There are less efferents leaving the cortex from those areas – most of them terminate somewhere else in the cortex, possibly closer to the regions directly controlling the motor neurons.

The capability to generate feature maps comes from the competitive nature of lateral interconnections between columns that was described earlier. It is a very interesting aspect of the overall functionality of the cortex, that has been studied by many researchers. There have been several artificial networks implemented, for example Kohonen's network, that behave in a similar way, i.e., where local specialization develops when sample data is fed into the network. Ultimately, certain features or concepts will activate a single unit or a well defined set of units. The relationships between the images in the network as represented by the patterns of the connectivity are a reflection of the relationships in the training sample data. There is a strong evidence that the same process occurs in the cortex.

From the perspective of this work, other capabilities of a network of cortical units are more interesting. It has been suggested in the previous section, that a lightly activated (i.e., only the pyramidal neurons in the upper sections of the column are active) column can represent an anticipation of a certain feature or concept. That is equivalent to saying that the column represents a certain goal. If the anticipation is satisfied, a conclusion is that the goal has been reached. The initial anticipation probably is the result of a desire or need born somewhere else in the brain. The limbic system may translate such needs into signals that are carried over the cortex afferents to proper columns. Many columns are probably involved in such a complex task. Each of the involved columns represents a sub-goal that must be satisfied to satisfy the global goal. The activity that represents a sub-goal, that from the local perspective is just another goal, is spread across the cortex through the cortico-cortical efferents. None of the receiving columns get highly activated, unless they are also anticipating or getting external inputs from the sensors. In such a case, the anticipation of the receiving columns is fulfilled, i.e., the goal is satisfied, and the columns are shut down by the inhibitory interaction between the lower and upper layers.

Although that is only a hypothesis, there is some research data that seem to support the described behavior. It has been demonstrated, that before a hand can be moved, for example, to a certain desired position, the column in the sensory cortex that represents the hand in that position is activated slightly before the actual movement takes place. The hand is in another position at that time, so the activity cannot be caused by a sensation. It seems that the cortex is presented with the goal that is personified by the activity in the column. The activity that spreads throughout the cortex finally reaches the column that represents the current position of the hand. Because the hand is actually in that position, the sensors send signals to the column's input. The result is that column now becomes highly activated. That activity can easily be transmitted to the primary motor cortex that is more or less symmetrical, with respect to the topographical maps, to the somato-sensory cortex. That may trigger a motor action that will move the hand into another position that is closer to the goal position. The next sensory column once again gets two inputs: one from the sensors and another from the anticipating column, so the same process occurs. Finally, the hand is guided to the desired position, so the first column in the chain gets highly activated and shuts down. The goal has been achieved. The hand is in the desired position.

The process of moving a hand has been dramatically simplified here to illustrate the capability of the cortex to perform searches. The search trees in the cortex might be very complex with many branches at many levels. The basic mechanism however seems to be the same at every point.

There are certain activities that are being performed routinely, without any dose of uncertainty. That may be due to that fact, that certain cortical paths are repeated many times, so the influence of certain columns on others becomes so strong that the activity is transmitted without any loss of its intensity between the columns. Therefore, if a column get activated with a sensory input, it may activate the one that is tightly coupled even if the other cell is not is the anticipating state. If that happens in the somato-sensory cortex, the activity can be transmitted to the motor cortex and the movements can be generated as described earlier.

In the next parts of this work, the observations made in this and the preceding chapters will be the basis for the attempts to build a model of the cortical column. A network of such model processing units has characteristics like and exhibits behavior similar to biological cortical networks.

CHAPTER 3

A model of the cortical column

OVERVIEW

For years, scientists have been looking at the brain as a prototype for computing machines. In some way, the classical von Neuman architecture is the result of yet another attempt – and a very successful one! There have been many others trying to duplicate the elegance with which the nature solved the computational needs of new generations of mutants in the never ending process of evolution. A new multidisciplinary area of science has been founded that encompasses the research related to reverse engineering of the brain. It appears that it is not that easy to imitate the simplicity and, at the same time, the complexity that is so integral to the human brain. There are many fascinating publications describing the theory and applications of neural networks and the reader in encouraged to refer to such material for the details. This work is this author's endeavor to mimic Nature.

A biological cortical column that has been presented in Chapter 2 is proposed in here as a prototype for an artificial information processing element. On one hand, the column is the smallest entity in the cortex that can represent abstract data. On the other hand, it is the largest cortical object that has been relatively well researched and can be described in terms of facts and not hypotheses. In the opinion of this author, a neuron, that is usually modeled in neural networks, is too simple to be used to process higher levels of information. In turn, groups of neurons, cell assemblies, that represent abstract information in a distributed way are hard to manage. The cortical column seems to be a tempting choice for a building block for neural networks whose purpose is to process abstract data. The column is positioned in this work in the role similar to electronic logic gates that can be used to process complex information at relatively abstract levels.

This model is based on the model proposed by Burnod [7]. However, in addition to some unique features of the model of the cortical column presented in this thesis, notably the learning schema, the main emphasis is laid on the computational capabilities of a network of interconnected columns. To continue the analogy with electronic devices, the network of artificial columns could be compared to an EPROM chip that can be loaded with any specific data and used as a module in more or less complex combinations of chips. The generality, or in other words universality, and modularity of the network were the main assertions in the design of the model of the column as well as the network.

There is a more detailed comparison of the model presented in this work and Burnod's model in the last section of this chapter. The computational networks that utilize both models will be compared in the next chapter.

In the description of the biological cortical column a special emphasis was put on the features that are used in the model presented in this chapter. The details of inter-neural connectivity are dropped in favor of generalized relationships. The inhibitory interconnections within a neighborhood of a column are ignored as well. Lateral inhibition is required to build discriminatory maps. Such a process can be achieved with other networks (for example Kohonen [18]). Therefore in this work, an assumption is made that the nodes of the network do have specific meaning or features or concepts associated with them. That could be accomplished by training a feature extraction and topology preserving network and clamping the outputs of the nodes that generalize specific features onto the network of columns as illustrated in Figure 19.

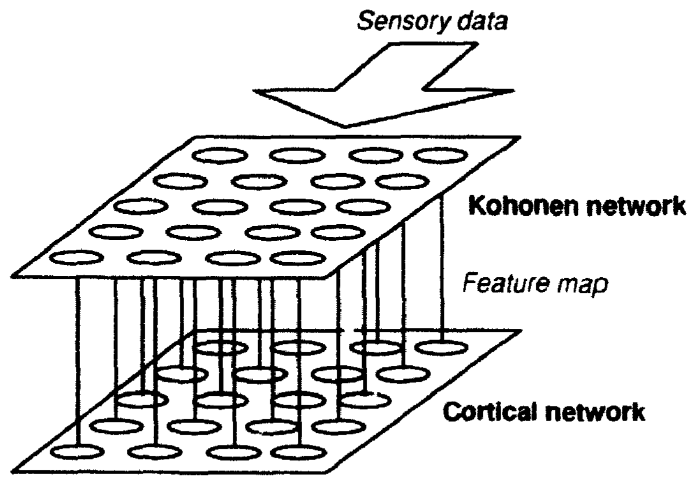


Figure 19. Preprocessing input data using a Kohonen network.

The network of artificial columns is called neurosolver, because it is capable of performing searches in the space of all possible states of the network activity. The problems are presented as points in that space in the form of an increased activity in a node or a set of nodes that represent a goal, i.e., the desired end state. The network solves the problem by subsequently firing, i.e., activating at the high level, all sets of the nodes that constitute the solution path. It is the path from the current state (a set of premises) to the goal. That capability will be discussed in detail in the next chapter.

ARCHITECTURE OF AN ARTIFICIAL COLUMN

In Chapter 2, while presenting the anatomy of the biological cortical column, two distinct parts were denoted as playing important roles in the processing capabilities of the column as well as the network of columns. The upper section of the column including the supragranular layers and encompassing the upper pyramidal cells is a prototype for the upper division of the model. The section intersecting the infragranular layers that contains the lower pyramidal neurons corresponds to the lower division of the model.

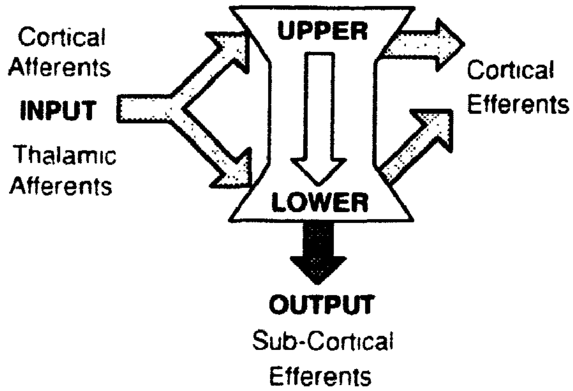


Figure 20. An artificial column.

The section of the biological cortical column that intersects the granular layer receives afferents, both cortical and thalamic, so in the model it is represented as an input. The external data are passed into the column through the thalamic input. The cortical input is used to interconnect the columns into a network. It is also used to set the activity of the column at the low level indicating the goal – problem to solve. Direct synapses and some of the interneurons connect the input layer with the upper and lower pyramidal cells. In the model, there are paths that transmit the input activity into both divisions.

There are also interneurons that connect the upper pyramidal neurons with the lower pyramidal cells. That is reflected in the model by upper-lower connection.

From now on, the term column will be used to refer to the model. If there is a danger of a confusion between the model and the biological cortical column a proper adjective will be added.

CONNECTIVITY

The artificial cortical column is a building block for constructing computing networks. The external connectivity of the column plays, therefore, a crucial role. The connectivity of the column is illustrated in Figure 21.

There are two connections that, as internal to the column, do not take part in inter-connecting columns in a network. The Upper-Lower connection transmits the activity from the upper division to the lower division. The internal Lower-Upper connection realizes the inhibition (suppressing) of the upper division by high activity in the lower division.

The Upper-Upper connection corresponds to the biological columnar efferents realized by the axons of the upper pyramidal cells carrying signals from one column to another. Those axons originate in the supragranular layers and terminate in the cortex.

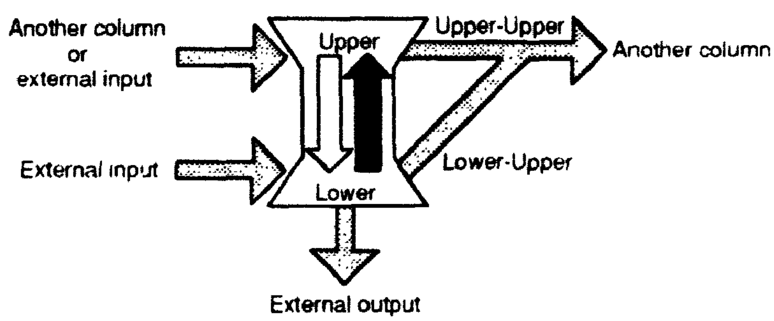


Figure 21. The connections of the artificial column.

The Lower-Upper connection, in turn, corresponds to the axons of the lower pyramidal cells. Some of those axons (mostly from the higher located - in layer 5 - lower pyramidal cells) carry signals to other cortical areas.

Most of the axons of the lower located low pyramidal cells, those in layer 6, project to the sub-cortical regions. In the model, the artificial counterparts constitute the output of the column.

In the model, some important simplifications have been made in comparison to the biological cortical column. Firstly, the input is being injected into the upper or lower division of the column. The reader may remember, that in the biological column, the granular layer is the input layer . The activity, however, is carried to the upper and lower divisions through the interneurons. Secondly, there is no difference between the location of the termination of the cortical connections from the upper division and from the lower division. They project to the same targets. The biological efferents originating in the layer 5 usually project to more distant areas than those originating in the upper layers of the cortex.

FUNCTIONALITY

The artificial column is a three-state device. If there is no input activity and no sustained internal activity from the past, the column is inactive. The interpretation of the inactive state from the information processing perspective is that the concept represented by the column is not present in the current computing context. It is a definitive NO. There is no output activity whatsoever from a column in that state.

The upper division of the column can be activated by action potentials from other columns, incoming through connections from both, upper and lower, divisions, and from the external, cortical, afferents. The input activity of the upper division is calculated according to the following formula:

$$\text{inputActivity} = \sum_{\text{activeConnections}} \text{actionPotential} * \text{connectionStrength}$$

activeConnections are those from the columns that have not been recently activated directly by this column. That rule prevents the existence of self-exciting pairs of columns. It does not, however, prevent a longer self-exciting loops.

After the input activity is integrated in the upper division, the division sends action potentials via connections to other columns. The pattern with which the activity diffuses throughout the network depends on the strengths of the connections.

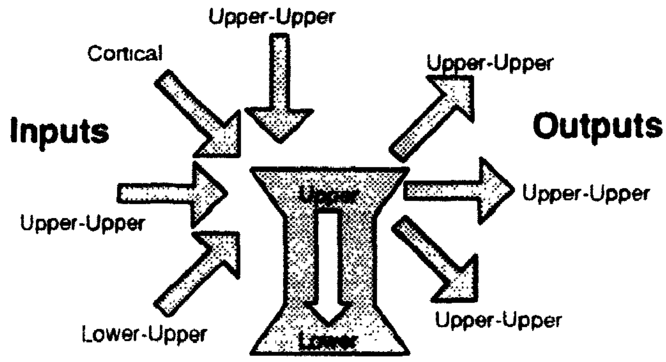


Figure 22. The integrating capabilities of the upper division.

The activity in the upper division is transmitted, as well, to the lower division. The Upper-Lower channel does not have any resistance, so the lower division gets exactly same level of activity as the upper one. Although there are connections leaving the lower division, the activity is not transmitted anywhere until the threshold level is reached. If that happens, it is said that the column fired. Then the activity is distributed through action potentials to the receiving columns. The strength of a specific connection determines how large an influence on the receiver the firing of the column will have.

Both the lower and upper divisions receive external afferents. These afferents play an important role as the inputs to the system. The cortical afferents are incoming to the upper division, and the thalamic afferents are incoming to the lower division. The cortical input can be used to express goals by activating the upper division of the column[16] that represents a specific, desirable state, feature or concept. A similar process occurs in the biological brain where the limbic system can trigger an activity of a column that represents, for example, a desire. The persistence of that activity denotes the goal to achieve - the desire to satisfy. The goal is reached when the activity is suppressed. That will happen when the column fires. The firing of the column can also trigger some external action. Figure 23 illustrates that process.

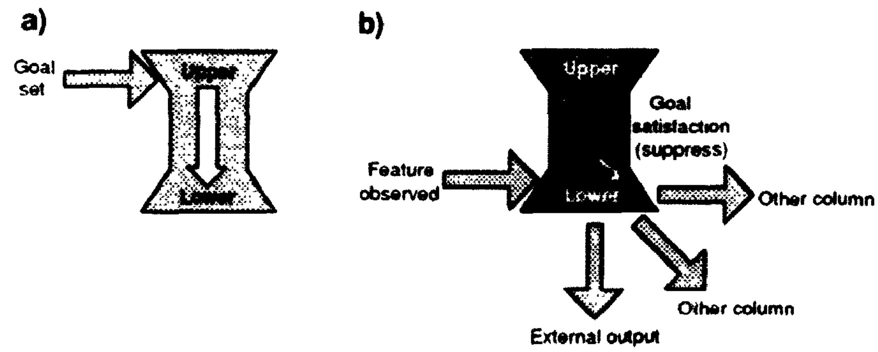


Figure 23. Setting the goal (a) and its satisfaction (b).

The upper division of the column is a vehicle for processing abstract data through the changes to the activity patterns depicting concepts or sequences of concepts. On other hand, the lower division is in touch with the physical world as represented by the activity of the sensors.

The upper division is the integrating part of the column, since it correlates signals from other columns. The lower division is the decisive component of the column. It determines whether the concept is perceived and controls the output that may, in turn, alter the environment. Ultimately, the lower division is a vehicle that drives the behavior of the column toward goal satisfaction.

[16]Or upper divisions of a set of columns if the concept has a distributed nature.

After firing, the column becomes insensitive for a while. No input activity, thalamic nor cortical, is accepted. That prevents two mutually connected columns from firing in an oscillating manner. Such behavior has been observed in biological systems as well.

ADAPTIVITY

The strength of the connections between the columns determines the nature of the changes to the patterns of network activity. That aspect of the behavior of the network defines the information processing characteristics of the network. The capability to construct an information processing device by a dynamic process is the very core of the interest of this work. Like in other types of neural networks, that capability is achieved by the adaptivity of the strengths of the inter-columnar connections.

In the model presented in this work, there are two rules of how the connections are modified depending on the patterns of changes in the activity of a column and the columns that transmit signals to and receive signals from that column. The first rule (the feedback rule) states that if a specific column fires, the strengths of all Upper-Upper connections to the columns that fired directly before are increased. That process is illustrated in Figure 24. Column B is the center of the example. Columns A and C are connected to B. At a certain point in time, let us say τ, column A fires; columns B and C do not. In the next tick, $\tau + 1$, column B fires. According to the adaptation rule, column A fired directly before column B, so the connection to the upper division of column A from the upper division of column B is strengthened. The connection to column C stays the same or is, at least relatively, weakened.

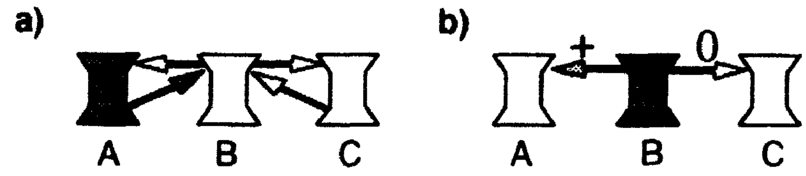


Figure 24. The adapting process of an Upper-Upper connection.

It is important to note that after the connection strengths are modified, it will be more likely that activity in the upper division of column B will cause changes in the activity of column A rather than in column C.

The second adaptation rule (the feed forward rule) says that the strengths of all Lower-Upper connections are increased between the columns that fire now and those that fired just before. Figure 25 illustrates the process of modifying the Lower-Upper connection. In the figure, there are again three columns: A, B and C. Column B fires at the time τ and sends the action potentials to columns A and C. At the time $\tau + 1$, column C fires. The adaptation process increases the strength of the connection between the lower division of column B and the upper division of column C.

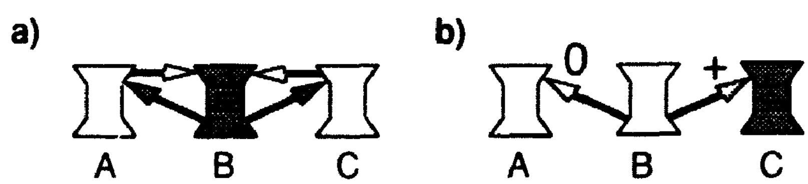


Figure 25. The adapting process of an Lower-Upper connection.26

The implications of the feed forward rule of adaptation are similar to those stated for the feedback rule: after the modification, firing[17] of column B will have bigger impact on the activity in column C than in column A.

---

[17]A column fires when the activity level of the lower division is higher than the output threshold, so the column transmits activity from the lower division only if firing.

The reader should note that both adaptation rules work at the same time. For the clarity of the description they were separated in the examples.

The ability of a column to adapt the strengths of the connections with other modules depending on the patterns of network activity will be analyzed from the network's perspective in the next chapter.

CONNECTION STRENGTH

There are currently many formulas for the strengths, or weights, of the connections between the nodes of neural networks in use or trial. Most of the learning schemes, the algorithms ruling the adaptivity, like that described in the previous section, are slow. They require many iterations of presenting samples before the networks can learn; i.e., recognize classes of input patterns, extract features, suppress input noise, etc. One of the premises of this work was to design a formula for the strength of a connection that would allow an input pattern to be recorded fast.

The model was implemented using two different approaches to the computation of the strength of a connection:

- hebbian and
- probabilistic.

The probabilistic approach was proposed by Burnod [7, 2]. The learning rules are, however, different in this work.

Hebbian style

The hebbian approach employs the modification rule that was proposed by Hebb [13]: if two cells tend to be activated together then the strength of the connection between them is increased; if the opposite holds, then the connection strength is weakened. That rule has been modified for the purpose of this work due to the sequential nature of the computations of a network of cortical columns. The analysis of the coactivation aspects of cortical columns is not attempted in this work, though that might be part of the future experiments.

In the terms of the hebbian rules, the modification scheme described in the previous section can be re-worded. If the value of a connection strength is designated ρ, then $\rho \in [0,1] \subset \Re$. The strength of the connection that links the upper division of the column that has just fired with the upper division of the column that fired directly before gains a constant or variable value $\varepsilon \in [0,1] \subset \Re$. The variable ε can be a function $\mathbf{f}$ of the previous strength ρ_{τ} or the activities, $\alpha_{pre\ synaptic}$ and $\alpha_{post\ synaptic}$, of the columns that are connected by the link, or more complex combination of all of those. Generally, using a variable ε is advantageous over using a simple constant. The adaptation rule for the Lower-Upper connection is similar:

$$\rho_{t+1} = \rho_t + \varepsilon \quad \text{where} \quad \varepsilon = \mathbf{f}(\rho_t, \alpha_{pre\ synaptic}, \alpha_{post\ synaptic}) \in [0,1] \subset \Re$$

Probabilistic style

In the probabilistic approach, certain statistical parameters are recorded for every division of each column, let us say C_i and every connection, C_iC_j, and later used to calculate the strength of the connection. When a division activity increases from low level to high level of activity, then the counter C_{up} is increased. Another counter, C_{in}, is increased for each column that sent action potential to the inputs of the column in question, so it counts the global influences on the column. The third statistical parameter, C_{out}, counts all receivers that fired as a consequence of receiving action potentials after this column fired. A counter of influences, C_iC_{jcons}, is maintained for each connection. It is increased each time the connection carries an action potential from the pre-synaptic column, the transmitter, that fired to the post-synaptic column, the receiver, that fires as a consequence. These statistical data are used to calculate the strength of both, the incoming and outgoing, connections.

There are several probabilities calculated for the purpose of calculating the strength of a given connection. In the following formulas, A and B are pre-synaptic and post-synaptic, with respect to the connection in question, cortical divisions respectively.

- The probability indicating how likely the change of the activity of the pre-synaptic division from low to high is to generate a successful action potential (i.e., such that it will take part in influencing the post-synaptic division):

$$P_{cons} = \frac{AB_{cons}}{A_{up}}.$$

- The probability of how inclined the action potential carried through the connection is to increase the activity of the post-synaptic division:

$$P_{in} = \frac{AB_{cons}}{B_{in}}.$$

- The probability of how prone the post-synaptic division is to change its activity from low to high upon reception of any action potential from any input:

$$P_{up} = \frac{B_{up}}{B_{in}}.$$

- The probability describing the likeliness of an action potential being carried over the connection after the pre-synaptic division's activity moves from low to high, changing the state of the post-synaptic division:

$$P_{out} = \frac{AB_{cons}}{B_{out}}.$$

- The probability of how prone the post-synaptic division is to influence other columns after changing its state from low to high:

$$P_{up} = \frac{B_{up}}{B_{out}}$$

The strength of a Lower-Upper connection is calculated using the first three coefficients:

$$\sigma_{AB} = P_{cons} * P_{in} * P_{up}$$

The strength of the Upper-Upper connection may be computed using the same formula but with the first and two last coefficients instead:

$$\rho_{AB} = P_{cons} * P_{out} * P_{up2}$$

There is a number of alternate ways to combine P_{cons}, P_{in}, P_{up}, P_{out} and P_{up2} in formulas for the strength of the connection. For example, for an Upper-Upper connection it might be advantageous, for the reasons that will be explained in Chapter 4, to just use $\rho_{AB} = P_{cons}$.

The strength of the connection between the upper and lower division is fixed and its value is 1. Consequently, any activity in the upper division is transmitted to the lower division. The strength of the connection in the opposite direction is fixed to -1. If the activity in the lower division exceeds the threshold, the activity in the upper division is suppressed.

Although the first impression might be that the hebbian and the probabilistic approaches are different, a little closer look allows us to restate the latter in the same manner as the former:

$$\rho_{\tau+1} = \rho_\tau + \varepsilon \quad \text{where} \quad \varepsilon = \mathbf{f}(\rho_\tau, \alpha_{\text{pre-synaptic}}, \alpha_{\text{post-synaptic}}) \in [0,1] \subset \Re$$

The difference is that the function **f** is now given in statistical rather than analytical terms. The reader may note that the formula using the defined probabilities may include the decay and inhibition components. The decay is implied by the use of the statistical counters C_{up}, C_{in}, C_{out} and C_iC_{jcons}. If one of the inputs does not contribute to the activity of the column, then, in consequence, the strength of the connection from the corresponding division is decreased, because B_{up} and B_{in} are increased, and AB_{cons} stays the same. If the action potential carried by one of the outputs does not contribute to a high activity of the receiver, so AB_{cons} does not change again, then the strength of the connection is decreased, because A_{up} and A_{out} are increased.

Another counter, C_{down}, that stores the number of the cases when the activity of the column went down, is included in the model. C_{down} could be used in the definitions of analogous probabilities that would provide an inhibitory factor in the formula. The inhibition is included in the considerations of future enhancements to the model.

THRESHOLDS

Each division of a column has a number of activity thresholds:

- a low activity threshold defines the inactive state,

- a high activity threshold defines the active state and
- an output threshold indicates the minimum activity that can be transmitted in the form of an action potential to the receivers through the output connections.

All activity thresholds in the model have been fixed; i.e., they are not adaptive. The output threshold of the upper division is 0. The output threshold of the lower division is the same as the high activity threshold. The latter determines when the column fires, because the lower division is the output component of the column.

COMPARISON WITH BURNOD'S MODEL

In his work *"An Adaptive Neural Network: The Cerebral Cortex"* ([7]), Burnod proposed a columnar automaton, i.e., a model of a cortical column, in an attempt to explain the functionality of the cortex from the lowest to the highest functional level. Burnod's model is based on the concept that was first proposed by Szentagothai in his work *"The Module Concept in Cerebral Cortex Architecture"* ([31]). The concept was considerably refined, particularly the aspect of its connectivity and general functionality, by Mountcastle in *"An Organization Principle for Cerebral Function: the Unit Module and the Distributed System"* ([23]) and Zeki and Shipp in *"The Functional Logic of Cortical Connections"* ([37]). Burnod followed the naming convention of his predecessors and called the automaton module, rather than column. A functional module is defined as a set of columns that have a homogeneous activity. Although the existence of the upper and lower pyramidal columns and different connectivity patterns of the upper and lower sections of the column was known before, Ballard in *"Cortical Connections and Parallel Processing: Structure and Function"* ([4]) was the first to propose the model that clearly identifies three distinct parts: the upper, the intermediate and the lower divisions.

Burnod's original contribution is the concept of a call and action tree and the use of the trees to explain the functionality of the cortex.

In his original work, Burnod used the divisions in the descriptions of the processes occurring in the cortex. The automaton, however, is viewed as one entity that can have high or low activity or no activity at all. In his studies, Burnod uses upper-upper connections and upper-lower connections. The upper-upper connections used in this thesis have a similar to Burnod's application; i.e., to spread columnar activity in a call tree. The upper-lower connections are not used in our model. In Burnod's model they represent the probability that cortical inputs alone will induce the high activity in the receiving module.

Unfortunately, Burnod's original work and later publications (e.g., [2]) do not provide full details about the model. The activation rules for a module use the states of the internal and external inputs and the previous state of the module. First, the global external input is calculated. An a priori specified mask defines which individual external inputs will contribute to the global input of the whole unit. Next, each of the individual internal inputs is used to calculate corresponding local internal and external outputs. The calculation is based on the truth table that uses the global external input as a modulation factor for the inputs. Burnod uses E0, E1 and E2 to denote, respectively, none, low and high activity. Each of the local outputs may be in one of those three states. The state that is predominant on all of the internal outputs is assumed to be the state of the global internal output of the module. Similarly, the state predominant on the local external outputs becomes the state of the global external output. It is not clear which of these outputs is to be used in the next step of the column automaton.

The calculations of the local outputs use the strengths of the connections between the modules expressed by two probabilities. Those probabilities employ the statistical data collected during the learning. The type of statistics and the modification rules differ considerably from those presented in this thesis. The rules for calculating the states specified by the truth table are not uniform: i.e., they differ depending on the position of the module in the network. Burnod does not state it explicitly, but each module must be a priori assigned one of the several rules specified in the table.

The global outputs can be expressed by the following formulas:

$$OI_i(\tau) = \mathbf{f}(II_i^j(\tau), IE_i^k(\tau), OE_i(\tau - 1), P0_i^j, P2_i^j)$$

$$OE_i(\tau) = \mathbf{g}(II_i^j(\tau), IE_i^k(\tau), OE_i(\tau - 1), P0_i^j, P2_i^j)$$

where:

OI and OE stand for internal and external outputs respectively,

i and j are the indices of the receiving and transmitting modules respectively, and k indexes the external receptive field,

II denotes the internal input,

IE stands for the external input;

P0 and P2 are defined below.

In this thesis, more classical forms of the activity and output functions are used in the model of the cortical column. The activation rules, therefore, differ considerably from those used by Burnod and his coworkers.

The learning rules of Burnod's model are of a probabilistic nature. There are three types of counters for each local input used to express the probabilities. The first counter, C., counts the cases when the activity at the E2 level on the local input was accompanied by the activity at the E0 level on the global output. The second counter, C_+, counts the cases when E2 on the local input is accompanied by E2 on the global output. The third counter, C_{E2}, is increased each time there is a high activity, i.e., E2, on the local input. The probabilities P0 and P2 are calculated as follows:

$$P0_i^j = \frac{C_-^j}{C_{E2}^i}$$

$$P2_i^j = \frac{C_+^j}{C_{E2}^i}$$

P0 represents the probability that the learning module is inactive before a strong input from the transmitting module. Conversely, P2 represents the probability that the learning module is highly activated before a strong input from the transmitting module. Those definitions are in contrast to the modification rules used in this thesis.

In addition to the differences stated so far, there are several aspects unique to the model presented in this thesis. Firstly, we explicitly use the two divisions as two separate entities. The interactions between the lower and upper divisions are also specific to this work. A new type of a connection has been introduced; i.e. the lower-upper connection. Generally, our model is less probabilistic in nature; i.e., the probabilities are merely used to express the strengths of the connections. The implication is that it is possible to use an alternate, hebbian, style of learning. That would be difficult in Burnod's model.

For more details, the reader is referred to Burnod's original work or to Alexandre et al., [2]. In the latter, some of Burnod's coworkers explain the essence of Burnod's model in much simpler terms than the original and analyze some of its applications.

In Chapter 4, a comparison between the networks of columns and modules will be presented.

CHAPTER 4

The neurosolver - a network of cortical columns

NEUROSOLVER - AN INTRODUCTION

Every biological cortical column, as well as the artificial model introduced in the previous chapter, represents a certain amount of information. The location of the column in the network is determined by the place that the piece of information that the column represents, be it a feature, concept, idea, wish, etc., occupies in the corresponding physical domain that has been mapped onto the network. Therefore, any pattern of columnar activity in the network may be treated as a point in the space of all possible states of the observed environment. The connections between the nodes, as defined in the model presented in this work, describe temporal relationships between individual bits of information represented by the nodes. If so, then groups of connections represent temporal relationships between two adjacent states (adjacent with respect to time) in the domain. Hence, anything that happens in that domain has a corresponding sequence of patterns of network activity.

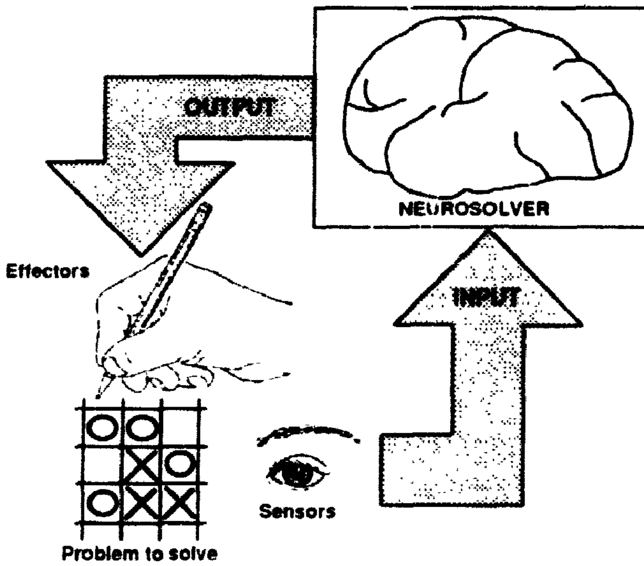


Figure 27. Applying a neurosolver.

The work described in this thesis is just the first step in the direction to achieve the ultimate goal of this author, i.e., to build a device that would mimic the functionality of the human cortex. Such a device we will call a neurosolver. The network of artificial columns as described in this thesis is the first incarnation of the neurosolver. The neurosolver is a device that is capable of recording the behavior of any physical system or object. The object can be observed by the system of sensors that detect its state. The states and, more importantly, the patterns of their changes are input to the neurosolver. The neurosolver modifies its inter-columnar connections according to the adaptation rules described earlier in this work. On the other end, each column has a determined meaning and may output signals that afflict the manipulators ready to alter some aspect or aspects of the observed object.

The recorded information may be used to activate required actions of the manipulators by presenting a goal. It is a certain state of the object, a point in the space of all possible states. The neurosolver is capable of activating the path in that space that leads from the current state to the goal state through a number of intermediate states. In the course of that activation, some of the columns involved fire and control the manipulators in the same way as it was observed and recorded in the past. Through the sequence of the manipulations, the required state of the object is achieved. The goal has been satisfied, the problem - solved: that is the origin of the name of the device: neurosolver.

The neurosolver starts to interact with the subject system as a *tabula rasa*. It gains all its experience, and problem solving capabilities, through the interchange of the sensory and manipulation control data with the system through the inputs and outputs. There is no separate learning cycle - the neurosolver learns while servicing the system, though at the beginning there is not much it can do. In the neurosolver described in this work, it is possible, in addition to the mixed mode, to run separately in the learning or performing mode.

In the further parts of this chapter of the thesis, the mechanics of the problem solving capabilities of the neurosolver are explained. To better visualize the behavior of the network, all considerations involve a simplification that each goal is initially given as an activation of a single column. Usually, any complex problem requires a distributed representation. That will be visible when the sub-goals are analyzed.

ARCHITECTURE

A neurosolver is a network of interconnected artificial columns. The connectivity follows the rules described in Chapter 3. Figure 28 illustrates the connectivity for a number of columns that are shown in a cross-section of the neurosolver. A column receives signals from the sensors through a thalamic input that is a part of the external input to the neurosolver. The cortical input is another external input to the neurosolver, and to each column, but it does not come from the sensors. It is used to present the goals, or in other words tasks, for the neurosolver. It is also used to provide additional clues that may contribute to the resolution of the problem. Each column contributes to the overall output of the neurosolver, although only certain signals may be in fact used by the manipulators. The cortical input signals activate the upper division of the column and are propagated to upper divisions of other columns through the upper-upper links. When a column fires, i.e., its lower division gets highly activated, then the lower-upper links carry the signals to the upper division of other columns as well.

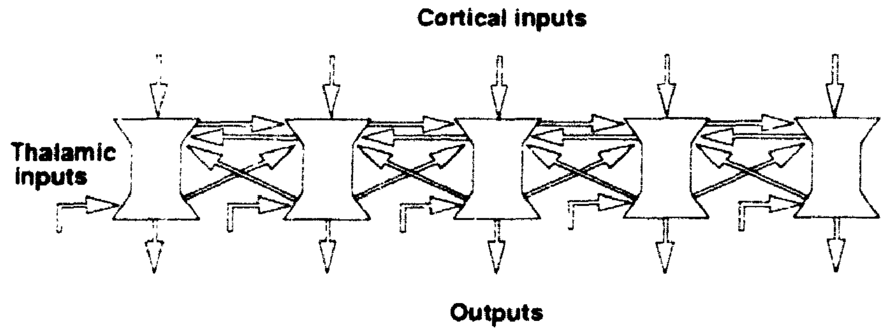


Figure 28. A cross-section of a neurosolver.

The neurosolver that is analyzed in here has a matrix architecture, as illustrated in Figure 29. The number of nodes is the same for each row and column of the matrix, although they could have different sizes without impacting the behavior of the network. Each node is connected, through lower-upper and upper-upper connections, to its neighbors in eight directions on the plane: vertical, horizontal and diagonal.

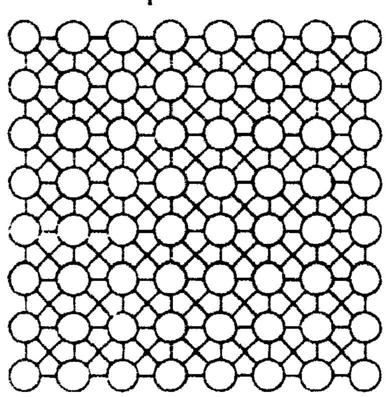


Figure 29. A top view of a neurosolver

That architecture is more suitable to describe the behavior of the neurosolver than a completely inter-connected network. The model has both modes implemented, but the completely inter-connected architecture is difficult to simulate due to a large number of connections that must be tested for growth in each cycle. The planar architecture be much simpler to build in hardware as will be described in the last chapter.

LEARNING

The initial strength of the connections in the neurosolver is zero, that is no activity can be propagated from one column to others. When the sensors start to communicate the sequences of events occurring in the observed system, the connections between firing columns are adjusted according to the rules described in Chapter 3. That process is illustrated in Figure 30.

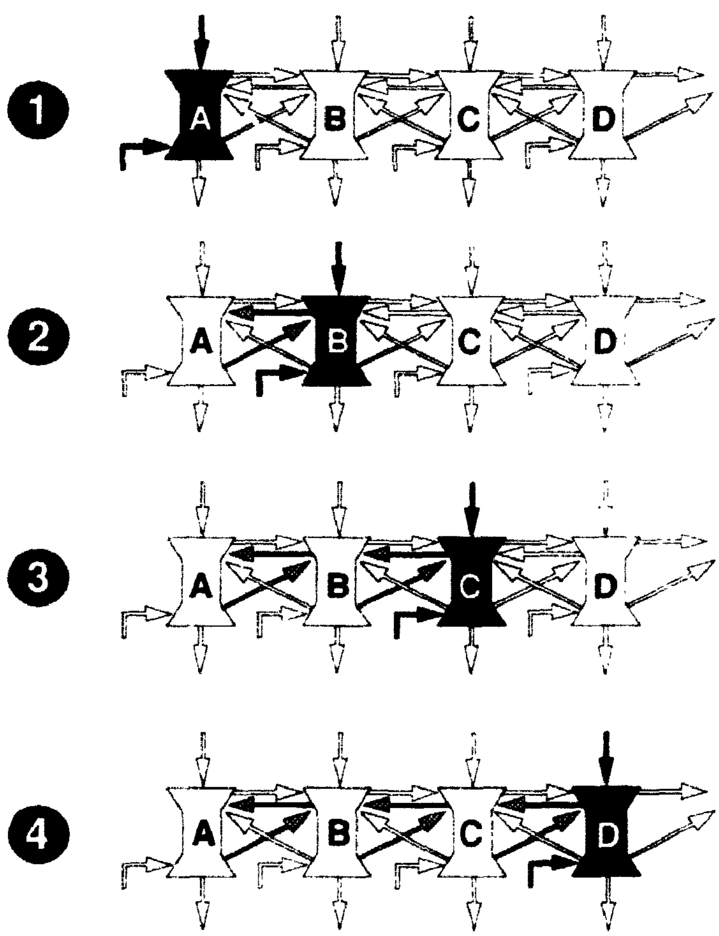


Figure 30. Learning a sequence.

In the cross-section of a part of a neurosolver there are four columns: A, B, C and D. The initial strengths of the connections between all the columns are zero. However, when, for example, column A fires and that is followed by the firing of column B, then the upper-upper connection between B and A is strengthened using the feedback rule. Additionally, the value of the strength of the lower-upper connection between A and B is increased as well using the feed forward rule. If column C fires next in the sequence, than the connections between B and C are modified in the same way as between A and B. Column D is the next to fire in the illustration. Again, the connections between C and D are modified appropriately. After some time depending on the learning schema used, two chains are recorded as shown in Figure 31. When using the probabilistic learning rules, the chains are formed just after one presentation of the sequence. The hebbian schema usually require more time to form the associations.

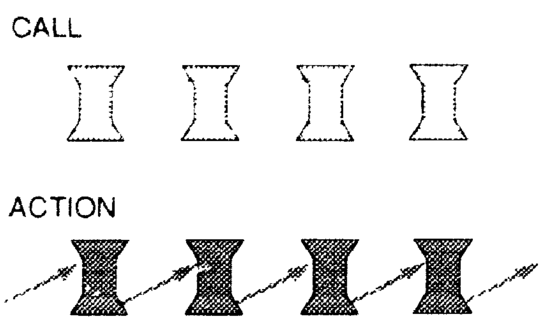


Figure 31. Two learned chains.

The first chain links the upper divisions of the columns through the upper-upper connections. It is created in the reverse direction to that of the firing columns. If the firing of column B is treated as a consequence of the firing of column A, than spreading a low level activity from B to A may be understood as a call, or a search, for the reason of B firing. The same reasoning applies to all columns in the chain. Therefore, if D is activated at a low level and that activity is propagated to C to B to A, than that is a search for the reasons of D firing. If D represents a desired state of the observed system, than A-B-C-D is one of the possible paths to satisfy that goal. The chain that is generated between the upper divisions is, therefore, called a call chain.

The increasing strength in the lower-upper connections between A, B, C and D, constitutes another chain. That is called an action chain, because if any of the column of the chain gets highly activated, it will cause the next column in the chain to fire as well. That type of chain requires many more repetitive presentations of the input sequence than the call chain.

CALL TREES - A BREADTH-FIRST SEARCH

A call tree

There might be many causal sequences leading to a particular state of the system. There might be, therefore, many chains leading to the same node of the neurosolver that has been interacting with that system. When a goal node is activated, like node G in Figure 32, its activity spreads along all chains that were recorded. It is not a single call chain created anymore; it is a call tree. The activity will spread in steps into all directions that may be the solution to the problem. One of such steps is illustrated in Figure 32.

Figure 32 includes the nodes (shadowed) that are already in the call tree. The arrows that are outlined indicate the recorded direction of the call – the connections that were strengthened in the past due to the columns firing in the opposite direction. One of the chains, or paths, of the tree has been labeled. Column G is the goal, that is the root of the call tree. Each of the subsequently activated columns, A, B and C, becomes a sub-goal to achieve the main goal G. In Figure 32, column D is activated as the next sub-goal and added, in that way, to the chain G-A-B-C-D. In the same step, many other new leaves are added in the same way to the tree.

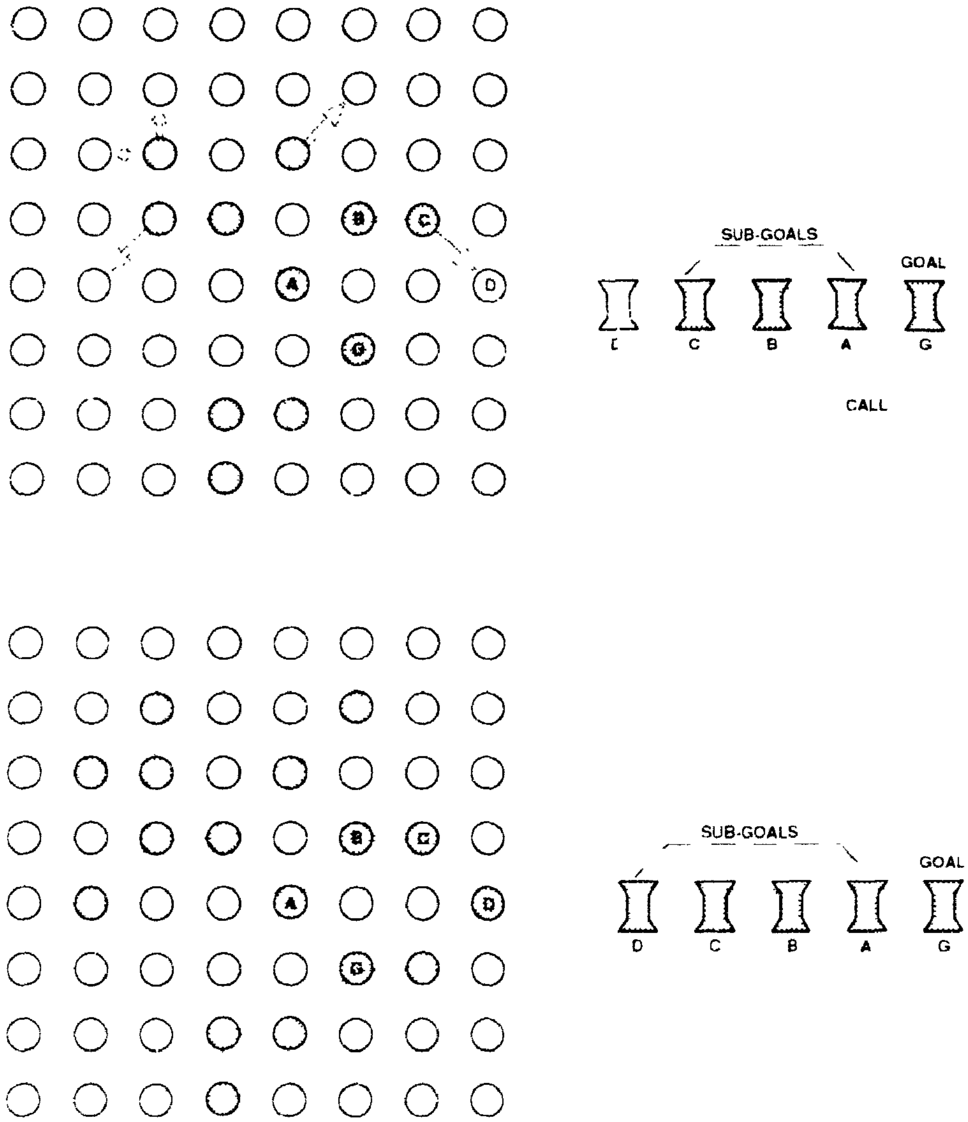


Figure 32. The construction of a call tree.

Triggering the resolution

The activity initiated by the goal column spreads throughout the network until one of the columns in the tree fires. That may happen due to the sensory input, as illustrated in Figure 33, or through the accumulation of sufficiently high activity in the upper division caused by action potentials arriving through several cortical inputs. Although the activity may continue to spread out along other branches of the tree, the propagation in this particular chain is terminated. The external input to the firing column means that there is an observation made now or a clue, or an axiom, set a priori, that indicates that this path may lead to achieving the goal. The firing column, column D in Figure 33, is called the trigger of the solution to the posed problem.

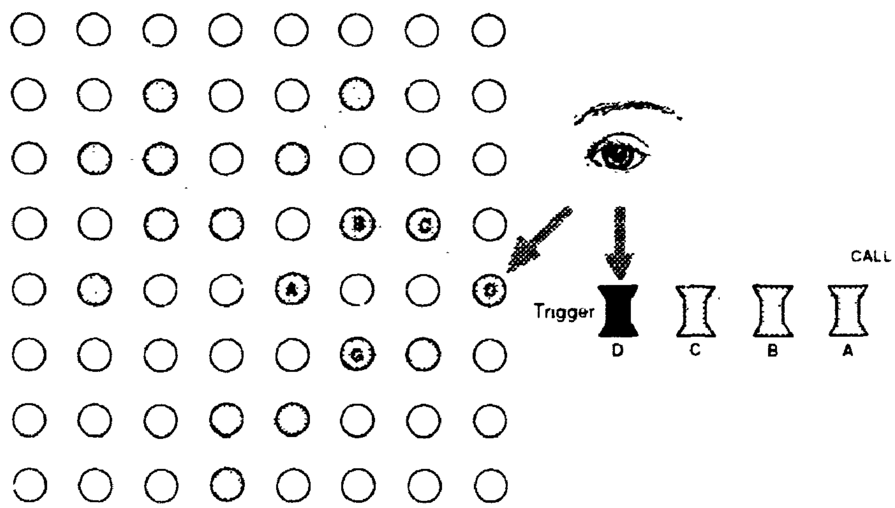


Figure 33. The resolution trigger.

Note that the use of an eye and a hand in the figures in this chapter is only a means to better visualize the processes occurring in the neurosolver. The reader should treat those symbols as representations of any type of sensations and manipulations. For example, the trigger can be just a state of mind that is sufficient to trigger an action along the path toward the solution of the problem. The effector that is represented by the hand can range from a robot arm to a voice synthesizer explaining the solution to the user using pre-built rules associated with each node.

The resolution path

The trigger of the problem resolution, the column that fires first in the search path, is, subsequently, shut down. The firing and shutting down of column D means that the sub-goal represented by that column has been satisfied. If the lower-upper connection to the next node in the tree in the direction toward the root, from D to C in Figure 34, is strong enough, the action potential that it carries to the recipient may cause the next column to fire. That is shown in the figure: column C fires. Similar processes occur now in column C. There might be a connection from the firing column to the effectors, so in addition to perhaps triggering the firing of the next column in the search tree, the firing of that column might have some impact on the observed system because of the changes to the activity patterns of the output of the neurosolver that impact the effectors. In that way, a part of the overall task has been carried out.

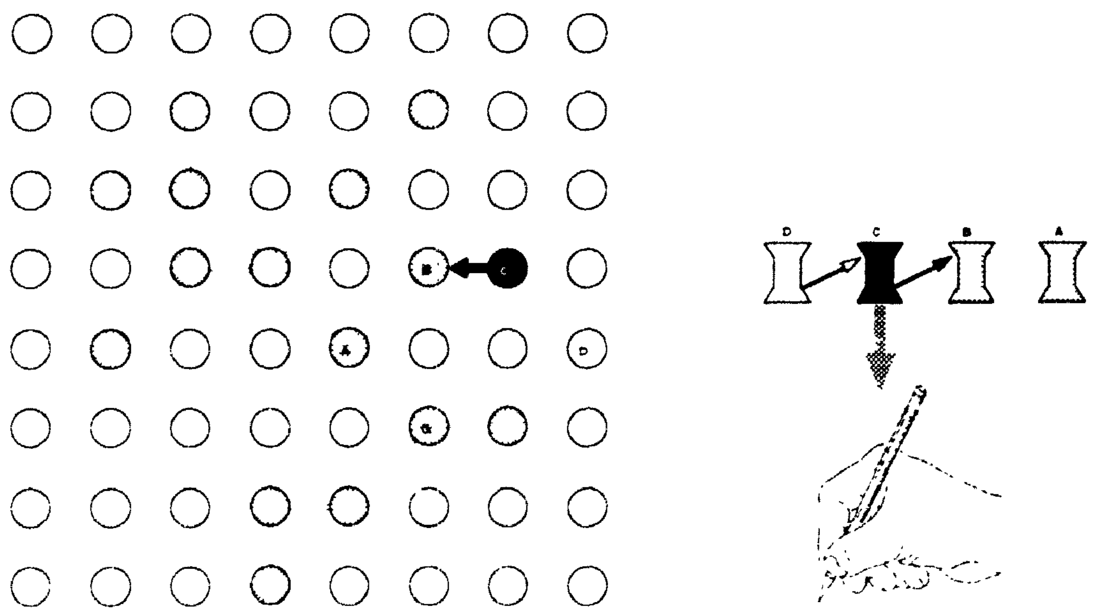
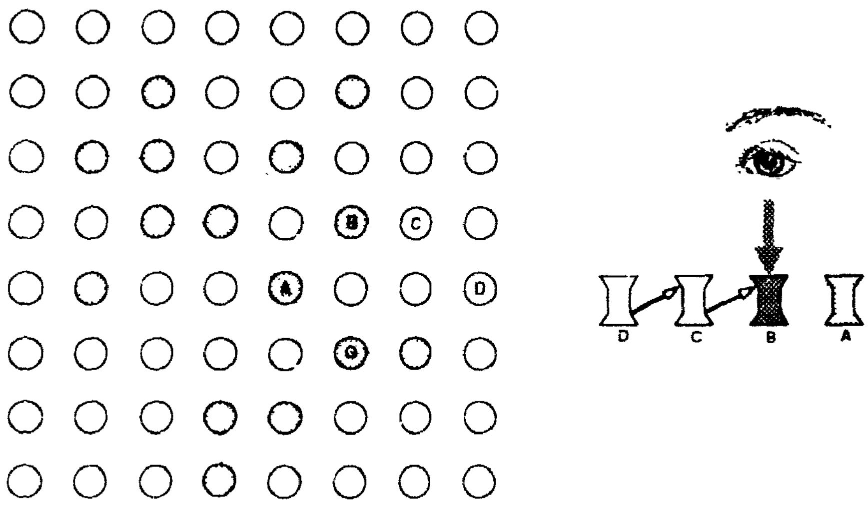


Figure 34. The resolution.

Sensory-guided resolution

It may happen, that the lower-upper connection between the firing column and the target columns in the call tree are weak. None of them can fire. In that case, the system cannot decide what the next step in the resolution of the problem should be. The resolution is suspended waiting for further clues. The activity can again be spread throughout the network in an attempt to search for the clues. The explorations may lead to partial resolutions being found in other parts of the call tree. That may affect the system. For example, one of the columns in the original path that was not getting sufficiently high signals might suddenly receive a thalamic input and finally fire. The implication is that the clue the neurosolver has been waiting for has been found. The resolution may proceed further in this branch.

In Figure 35, there is a simple example of a sensory-guided resolution. After firing columns D and C, the strength of the connection between C and B is too low to activate B by the action potential alone. However, when column C fired, an action potential had been sent to the effectors that altered the system. The change to the system has been noticed by the sensors and column B is notified about that by receiving a signal through its thalamic input. That, together with the activity caused by the input from column C, is sufficient to fire column B. The firing of column B may, in turn, cause another change to the system, and column A may, in turn, have increased activity on its thalamic input.



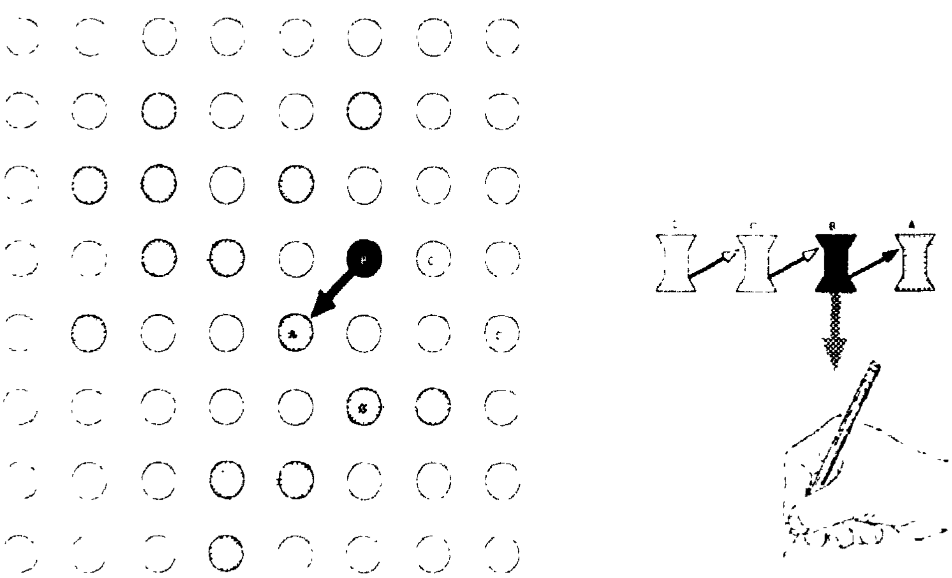


Figure 35. A sensory-guided resolution.

Although the firing is not sufficient to activate a complete resolution path, that can be achieved through the interaction between the internal processes occurring inside the neurosolver and external processes being observed by the sensors and controlled, at least partially, by the neurosolver through the effectors.

If the neurosolver is used to solve a mental task, the input may be controlled by the operator who presents the system with the known facts. In a situation where the neurosolver has some doubts about the next step in the solution, it may generate questions by including all options that are associated with the columns that could fire next. That may be the facts that became known in the course of the problem resolution or were known a priori, but the operator did not include them in the initial set of the input data, for example if they did not seem to be related or be of critical importance to the solution.

The goal satisfaction

When a column fires, and is shut down by the inhibition action described before, the activity in its upper division, of course, disappears. Subsequently, all columns that have been activated by the firing column in the call tree will also shut down as the result, unless they receive input signals from other sources. Actually, the whole sub-tree of the call tree is shut down. In Figure 36, such a case is exemplified by firing column A. A is another column fired in the chain D-C-B-A-G. When it fires, the activity in the sub-tree for which A is the root vanishes.

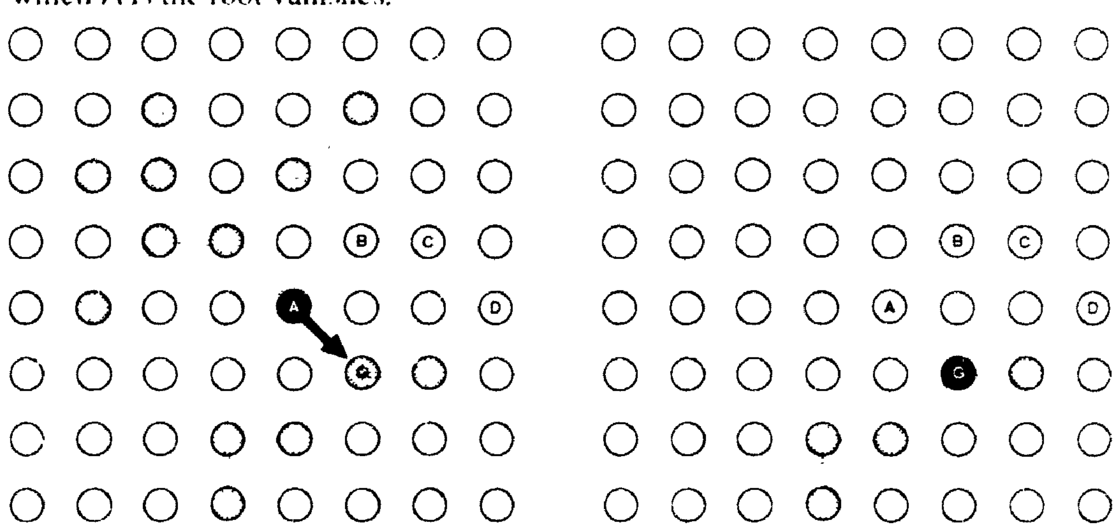


Figure 36. The goal satisfaction.

The next column to fire, in the example, is column G. That is the initial goal with which the neurosolver was presented. When G fires, the goal has been satisfied. The reason for the existence of the call tree disappears, so ultimately, the activity in all branches of the call tree ceases.

Through the mechanism of a call tree, a breadth-first search has been performed. The resolution path, D-C-B-A-G in the example, generated a number of output signals that constitute the solution to the posed problem.

ACTION TREES

It may happen that certain sequences of events occurring in the system and corresponding changes to the patterns of columnar activity happen very often. As a consequence, the strength of the connections from lower to upper divisions of the columns involved grow considerably in the direction of the firing sequence. The connections become so strong that they are able to induce a high level of activity in the recipients in the chain. Each of the columns involved realizes a part of a certain action by carrying action potentials to the effectors. Hence the chain generated in that way is called an action tree.

There is an action chain illustrated in Figure 37. The sequence D-C-B-A was observed in the past so many times, that it is sufficient to fire column D to trigger the action, i.e., the lower-upper connections from D to C, from C to B and from B to A are strong enough to carry the action potential on their own. The firing of each of the columns, D, C, B and A, causes some change to the system. All those changes constitute the action that can be associated with the action chain in this example, D-C-B-A. In more general case, not only a chain but an action tree can be created in a similar way.

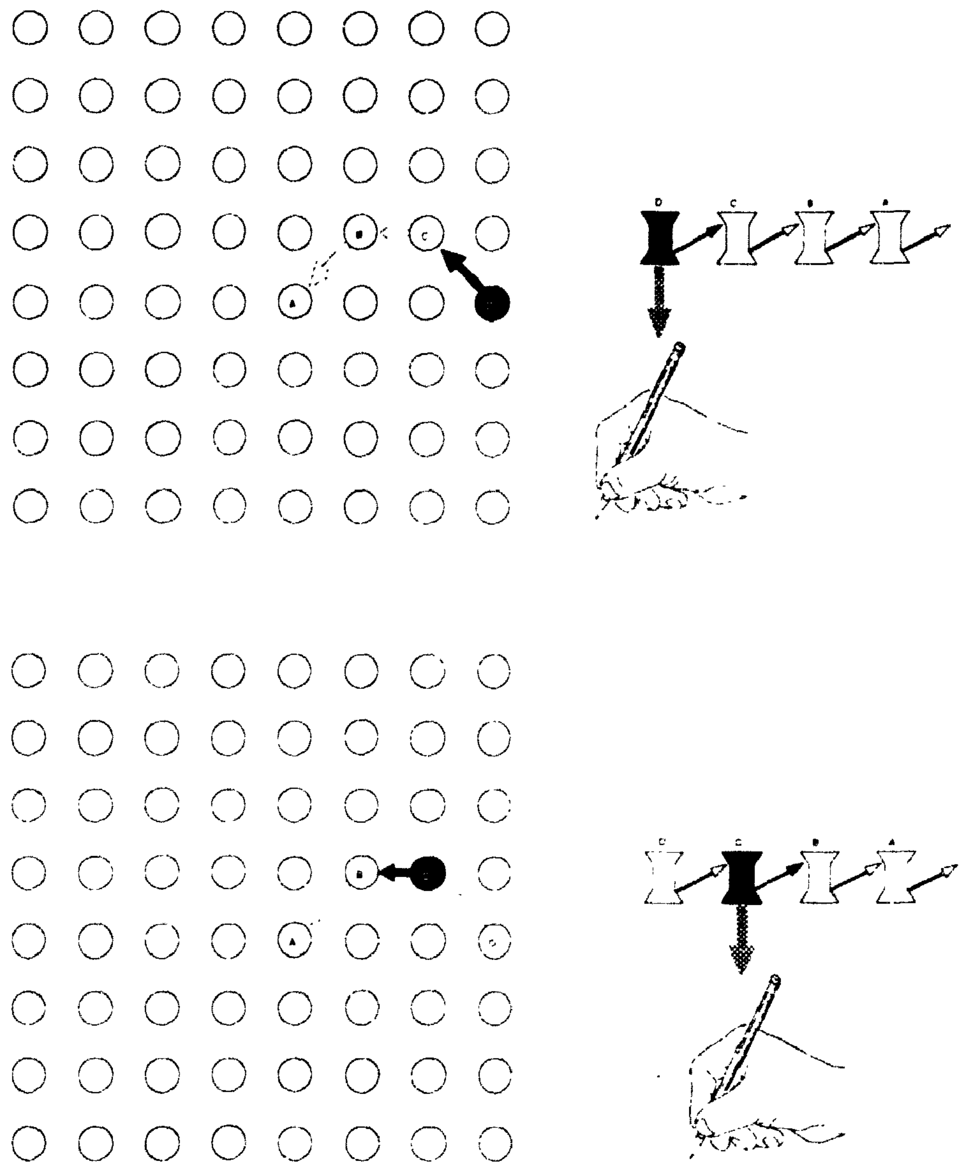


Figure 37. The firing of an action tree.

HIERARCHY AND PARALLELISM - USING A NEUROSOLVER MODULES.

Hierarchy of neurosolvers

When examining the brain, it has been indicated that some cortical areas are interconnected in hierarchies. That is the case with, for example, primary, secondary and tertiary visual areas. The topology mapping in each of those areas is different and encodes increasing levels of abstraction. Several neurosolvers can be used in a similar way. Each of the neurosolvers may be trained with data at different levels of abstraction, so different feature maps are generated for each of the neurosolvers. The concepts or features are also associated between the levels. There are three such neurosolvers in Figure 38. At the highest level of abstraction, there is a call tree in Neurosolver 1. At some point, column A is activated. That column has been associated with several other columns in the lower level of the hierarchy - with those that represent related concepts or constitute the higher level concept represented by A. The activity spreads not only in Neurosolver 1, but also in Neurosolver 2. The same may happen between Neurosolver 2 and Neurosolver 3 when the activity reaches column B in Neurosolver 2. The search at the lower level of hierarchy may be critical to the solution of the problem, if, for example, there is not enough evidence at the higher level of the hierarchy to undertake any action, i.e., to fire any column in the call tree at that level.

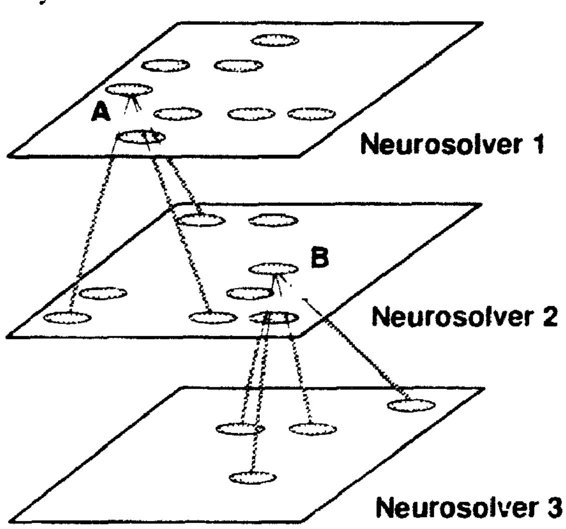


Figure 38. A hierarchy of neurosolvers.

Parallel processing

A call tree realizes a breadth-first search and, at least in the envisioned hardware implementation, ensures parallel processing of the problem. That is only one aspect of the parallel nature of computing in the neurosolver. Another type is to spread the activity into different neurosolvers that have been associated. In Figure 39, the call tree active in Neurosolver 1 has two branches: G-A-B and G-M-N. The last columns in both branches have association with columns in other neurosolvers, therefore the activity is transmitted and new call trees are generated in those neurosolvers. The search is now performed in parallel in three different neurosolvers. The neurosolvers may be organized into a hierarchy or may map just various aspects of the subject system. Finding sufficient clues in any of the maps, may contribute to the solution to the overall problem.

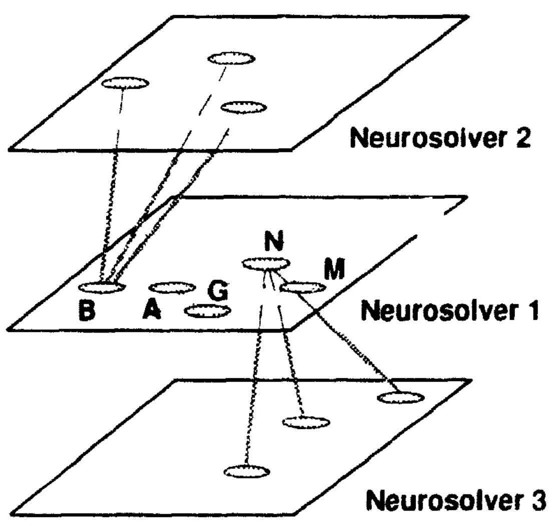


Figure 39. Parallel processing.

DISTRIBUTED REPRESENTATION AND THE NEUROSOLVER

Not every problem can be divided in such a way that nice hierarchies may be used in the search for the solution. Usually, some non-linearity is involved, so the use of the neurosolver as described in the previous section is not possible or difficult. Actually, the same has been observed in the human brain. Luria's models of information processing are very illustrative, but they do not constitute a complete description of the processes occurring in the cortex. It has been found that in addition to the pathways that accommodate the hierarchical view of the processing, there are other connections that indicate a distributed nature of the processing.

The neurosolver could be used to process information that has a distributed form, like in the example in Figure 40. Each state in the overall domain Ψ, a point $\mathbf{X} \in \Psi$, is actually a vector of three states - each from a different sub-domain:

$$\mathbf{X} = (X_1, X_2, X_3), \qquad \text{where } X_i \in \Psi_i$$

There is a neurosolver for each of the sub-domains Ψ_i. An abstract **X** is activated when X_1, X_2 and X_3 are activated at the same time. Each of the components may be a part of two different vectors, points in the main domain. Such a representation is more compact that one involving only one neurosolver for the domain Ψ with a column for each state **X**. An abstract call tree in the domain Ψ involves, in fact, three call trees in each of the sub-domains. An example of such a tree is given in Figure 40. There are three points involved **A**, **B** and **C**, that are in fact three vectors: (A_1, A_2, A_3), (B_1, B_2, B_3) and (C_1, C_2, C_3) of points in the sub-domains - columns in each of the three neurosolvers. The call tree **A-B-C**, translates to three call trees, one in each of the neurosolvers: A_1-B_1-C_1, A_2-B_2-C_2 and A_3-B_3-C_3.

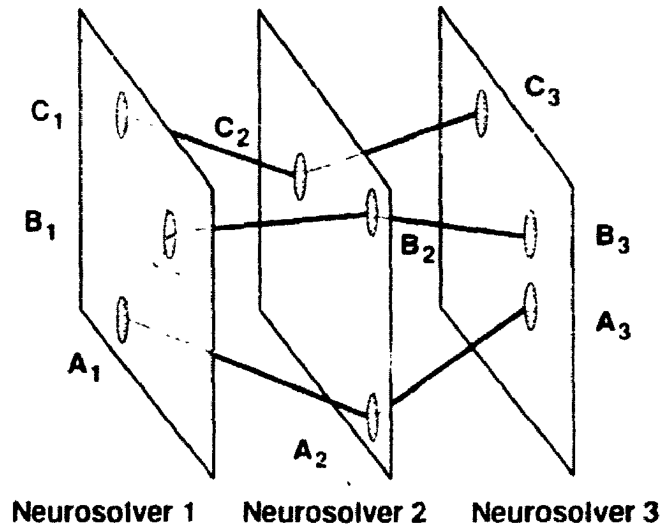


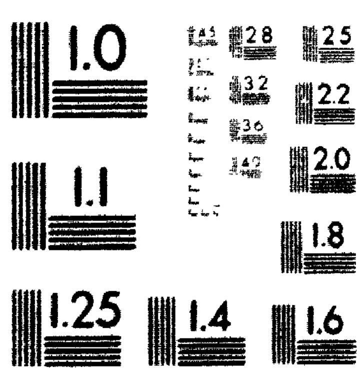
Figure 40. A neurosolver in a distributed representation environment.

The reader should note, that to use the neurosolver in such a configuration, it is necessary to have a completely interconnected version of the model. This is needed because the associations between the points in different neurosolvers will usually be in conflict with the topology mapping. It may not, therefore, be possible to create cell trees that connect each column only to its neighbors. Some patterns may require a distal column to be the next node in the tree. Additionally, each layer of a multilayer neurosolver must be completely connected with a neighboring layer. In Figure 40 each column in layer 1 is connected to every column in layer 2. Similar connections are in place between layer 2 and layer 3. The connections between columns in layer 1 and 3 are not needed in this case.

The use of hierarchies and distributed representation has not been actually tried in the simulator. In the author's opinion, however, those concepts are very important and seem to be natural for more complex applications of the neurosolver.

1

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT



PRECISION[SM] RESOLUTION TARGETS

2 OF/DE 2

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

NEUROSOLVERS AND ARTIFICIAL INTELLIGENCE

The neurosolver is a learning machine that learns from examples. The examples in this case are temporal patterns that are supplied by the sensors or by the operator. The learning uses a self-organization algorithm. There is no teacher, so the patterns are recorded and generalized without any heuristics to check their correctness. The neurosolver will learn a bad behavior in the same way as a good one.

The examples can be treated as cases that are being stored as references for further computations. In that respect, the computing performed by the neurosolver resembles the case-based reasoning paradigm. The searches correspond to the retrieval of the related cases that are used to infer the right response to the posed problem. The search has a breadth-first nature and is performed in a distributed and hierarchical manner. Figure 41 illustrates an example of case-based reasoning exhibited by the neurosolver.

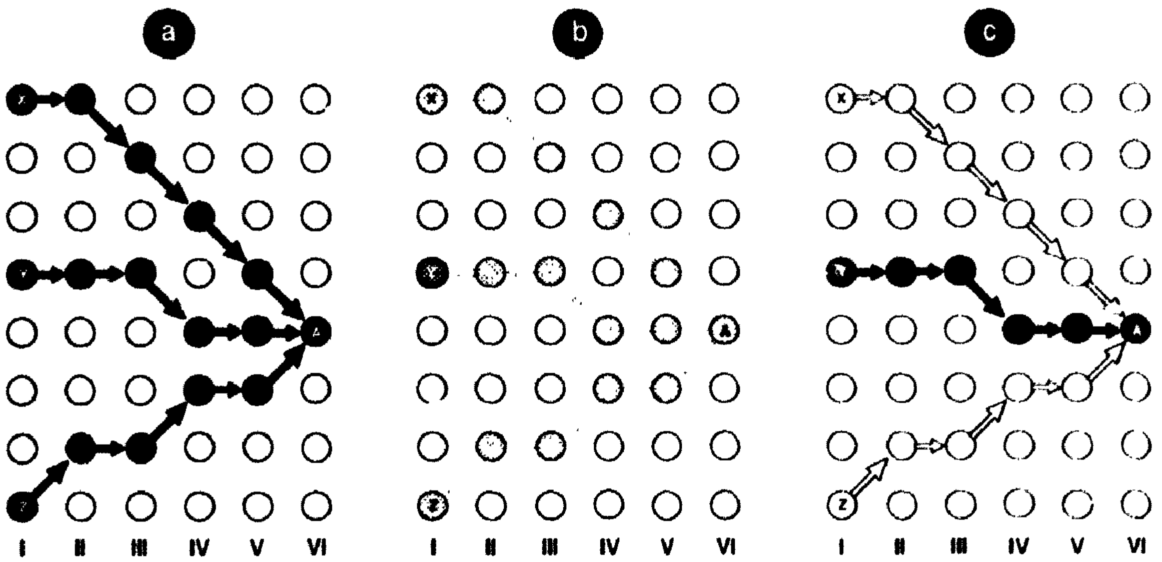


Figure 41. Case-based reasoning in the neurosolver.

The context of the example might be a diagnostic system with several levels of tests (I-VI). In part (a) of the figure, several cases are stored that lead to node A. A might be a fault in the context. The paths from X, Y or Z to A might represent various repair services depending on certain condition at the first and, perhaps, at higher diagnostic levels. If the fault A occurs, like in the part (b) of the figure, the call tree is generated with the leaves being the nodes at level 1. If one of the leaves is activated, e.g., Y in the example, then the chain leading to that node fires as illustrated in the part (c) of the figure. That might generate a repair advice, including the requests for additional diagnostic tests if at some level the activity is too low to fire any column. All activated columns that might fire are included in such request. After the test is done, one of the involved nodes fires and the proper path is continued. It would be advantageous to use a neurosolver with completely inter-connected nodes between two neighboring levels. A simulated version could be optimized by providing of an algorithm that removes the connections that never are utilized.

The collection of call and action trees can also be treated as a production system. If each temporal pattern is viewed as a series of condition-consequence pairs, then what the neurosolver encodes are rules. Each call tree corresponds to backward-chaining. Firing of a column implies that one of the condition has been satisfied; firing of a branch represents a match. A mixture of action and call trees may constitute forward chaining. An example of forward chaining with the use of call and action trees is shown in Figure 42.

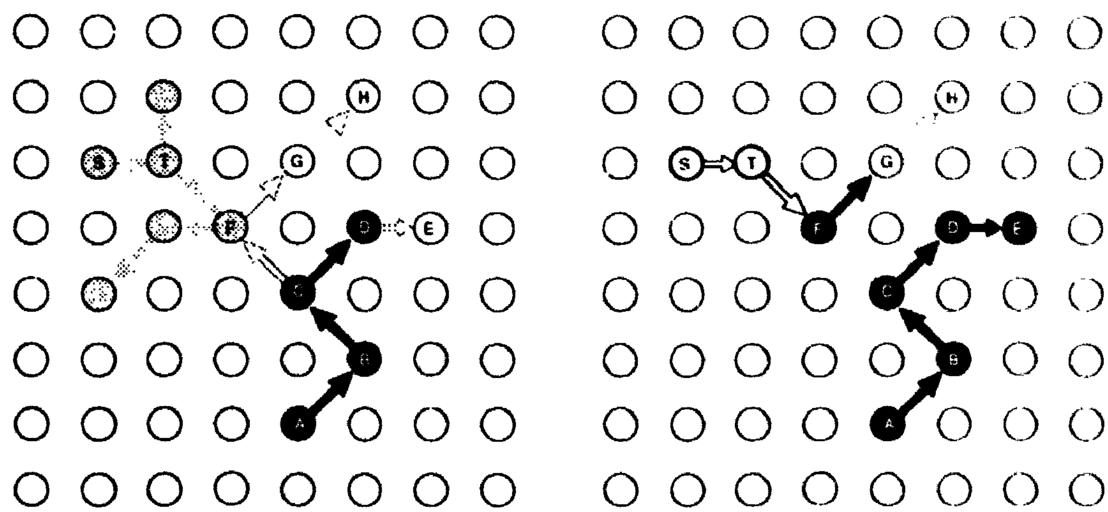


Figure 42. Forward chaining in the neurosolver.

The rules corresponding to the A-B-C action chain fire, because the columns A, B and C fired. The strength of the connection between C and D is sufficiently strong to fire D as well. The connection from C to F is too weak (F does not belong to the action tree) to cause the firing of column F, but it is enough to excite its upper division. A call tree spreads from there that may be triggered by an activity is some leaf, S in the example, causing the corresponding sub-tree (S-T-F) to fire. That ultimately leads to firing of F, so any action tree that starts from that node, F-G-H in the example, will be activated. That might be treated as a continuation of the action started at A and the call tree that was rooted in F represents matching.

Comparison with Burnod's network

The last section of Chapter 3 is devoted to the comparison between the model of the cortical column presented in this thesis and the model proposed by Burnod in [7] and [2]. In this section, the respective networks are also compared.

From a practical point of view, the network as described in Burnod's original work is of no use, because its design is driven by the biological data and the need to explain certain capabilities of the cortex. Although the descriptions of how to create call and action trees are presented, there is no explanation of what type of network and connectivity to choose initially. Burnod describes how the cortex works, so all pathways are taken for granted, as indicated by the research data. There are attempts in the first two parts of research [7] to explain how the brain grows, but at this moment such descriptions are of no practical use.

Burnod's coworkers present some practical applications of his model in [2]. The network that they use is hard-wired. The arrangement of the modules is based on the connections between the cortical columns and areas in the parts of the cortex that process visual information. There are several sub-networks that correspond to the hierarchies in the visual cortex. Each module has:

- a number of internal input-output connections with the neighboring units in the same area,
- a number of internal input-output connections with other areas,
- an external output which is either a feedback or a response outside the network and
- the external input that is either a stimulus or a feed forward input from other maps.

The external connections are organized in continuous overlapping receptive fields.

That network was used for pattern recognition. In the course of learning, a pattern is presented on the input (the retina) and the classification area has all modules inhibited but one, i.e., the module that represents the classification of the pattern. The activity is allowed to spread in both direction, i.e., from the retina to the primary area, to the secondary area, and on to the tertiary area and from the module representing the classification of the pattern in the reverse direction. After the connections are modified according to the rules explained in Chapter 3, the next pattern is presented. During the recognition process, an unknown pattern is presented on the retina and all modules in the classification area are set to E1 (the states were defined in Chapter 3). The call trees are generated and one of them is triggered. As a consequence, one of the classification modules gets activated at the E2 level and suppresses the remaining modules in that area by inhibitory interactions. The pattern has been recognized.

The approach to network design taken in this thesis is diametrically different. The connections are arranged in regular patterns, so modularity and generality can be achieved. It would be impossible to use the network presented in [2] in other than pattern recognition applications. The neurosolver aims at being a general problem solver.

The learning of the neurosolver is also in deep contrast with the learning that has been described above. This incarnation of Burnod's network is a pattern associator, while the neurosolver learns the behavior of the system by observation.

In summary, Burnod's original work contains many ideas that were implemented in this thesis. The subsequent attempts by Burnod and his colleagues to materialize the ideas are not equally impressive as his former attempt. The work on this thesis started before their results were published. Hence, there are many differences not only in the details of the implementations, but also in the general frameworks.

CHAPTER 5

A neurosolver workbench

Overview

It would be very difficult to discover the behavior of the model presented in the previous chapters without a proper user interface. It would be particularly difficult to observe the changes in the state of the nodes of the neurosolver. There are many connections, so every change to the level of activity of any column results in large propagation trees. It is even more difficult to trace when a number of nodes is changing in parallel. To ease the task of setting and modifying the neurosolver's parameters and observing the changes to the activity propagation schemes, a testing/modeling workbench has been implemented .

Smalltalk-80 was chosen as an implementation platform for several reasons. Smalltalk-80 is an object-oriented language, so it is convenient as a modeling tool. The polymorphism provides a mechanism for working with multiple model alternatives, so testing various models of an object is easy. The same may be said about inheritance - subclasses of the class of the model object can be used to test behavior nuances with the same common core behavior. Last but not least, the user interface is relatively easy to build after the learning curve has been overcome and thereafter easy to modify and maintain. Also, an important factor was portability. Smalltalk-80 has been ported to many platforms and ensures that the image can be interpreted by the virtual machine on any of the supported platforms. An IBM PC clone was used for development, while the School of Computer Science of Carleton University uses mostly Apple's MACs in the graduate labs.

Figure 43 shows the main display of the neurosolver test workbench. The left side of the window displays the matrix of the neurosolver's columns. The number of columns that are displayed depends on the size of the neurosolver. Each column is represented by two circles that correspond, in turn, to two divisions of the column: lower and upper. The workbench permits the pattern of activity of the neurosolver to be set and observed. The activity of each division is expressed by a shade of gray. Seven levels of activity have been chosen, because Smalltalk provides that many built-in bitmap patterns. There are, therefore, seven thresholds of activity when the color of the division changes. In addition, each column, as well as a division, may be examined and/or altered separately through the use of the Smalltalk's inspector.

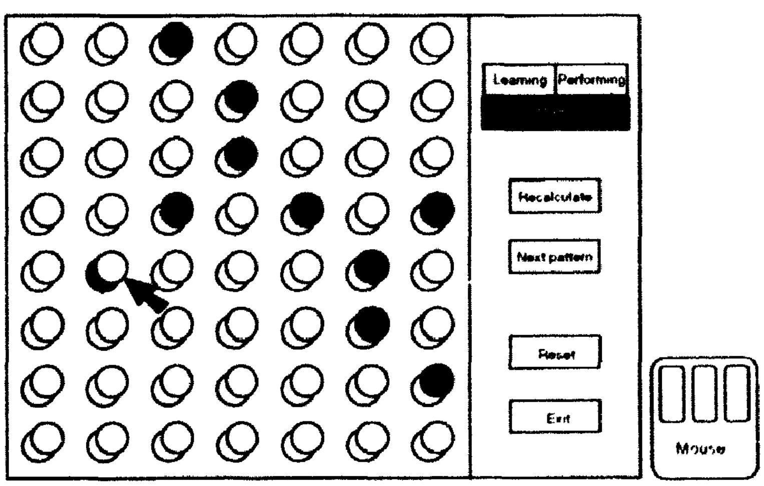


Figure 43. The neurosolver testing workbench.

The right side of the window contains several control buttons. There are buttons provided to change the mode of the neurosolver operation from learning to performing to a mix of both at the same time. Those buttons are exclusive switches, i.e., only one can be selected at a time. The remaining switches are action buttons that control the process of computing.

STARTING THE WORKBENCH

The workbench is started by sending createAs:size:learningMode: to the class Neurosolver. The first parameter indicates one of the three connection schema that can be chosen when instantiating the neurosolver. The initialization method accepts the following architecture requests:

- #simpleMatrix, only the neighbors along the North-South and East-West lines are connected,
- #matrix, all neighbors are connected,
- #completelyConnected, all columns are inter-connected.

The second parameter in the method indicates the size of the matrix expressed in the number of columns per each side. The third parameter indicates the desired learning mode (described further in this chapter). The selected size and architecture cannot be changed after the workbench has been instantiated. The learning mode can be altered.

For example,

```
Neurosolver createAs: #completelyConnected size: 7 learningMode: #probabilistic
```

will instatiate a neurosolver consisting of 49 completely inter-connected columns and open a window with the workbench for that instance of neurosolver. The probabilistic learning schema will be active initially.

SIMULATING PARALLEL PROCESSING

Ideally the neurosolver should be a multiprocessor VLSI device, with each processor modeling a cortical column. Parallelism would be a natural mode of operation, assuming that the input and output are folded into the same paradigm (i.e. all simultaneous sensations are transmitted at the same time). In the software implementation that parallel character of processing must be simulated.

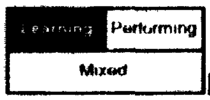
The operation of the workbench is divided into several phases. Before any processing can take place, the user has to decide about the mode of operation and type of the learning. That is described in the following sections. After that has been settled, the processing loop is entered. With the help of mouse buttons, the thalamic and cortical inputs of any column can be changed. Many columns might be selected depending on the needs. Next, Recalculate is selected to trigger the recalculation of activities of all columns. New activity levels are posted to connections as action potentials, so they can be used in the next step to propagate the activity throughout the network. At this point, the user can enter another activity pattern of the sequence being presented. If the sequence has been completed, the user can select Next pattern to inform the neurosolver about the new pattern. Pressing that button does not clear the sequences that have been already learned (i.e. the weight parameters of the connections are not reset).

To completely reset the neurosolver, so a new set of sequences can be presented, the user can use Reset. That will reset not only the activities, action potentials and thalamic inputs of all columns, but it will also reset all parameters that are used to calculate the strengths of the connections.

Exit should be used to quit the simulation session. There is a need to do some garbage collection after each session, so it is not possible to close the window using the title bar menu's Close that is the standard way of closing windows in Smalltalk.

MODES OF OPERATION

The neurosolver can be run in one of the three modes of operation: learning, performing and a mix of both. Each of those modes can be set by pressing a button with a respective description.



In the learning mode, the neurosolver adjusts the strengths of its connections in the direction (i.e. plus or minus) and by the amount depending on the changes in the activity pattern, as described in the previous chapters. That mode is used when learning phase must be separated from the performance phase.

There are two learning schema that can be used by the neurosolver: probabilistic and hebbian, as indicated in the description of the model. The schema can be activated by selecting an entry in the workbench menu that displays the schema that is not currently used. If that schema is selected it becomes the active one and the label in the menu is changed to another alternative as illustrated in Figure 44.

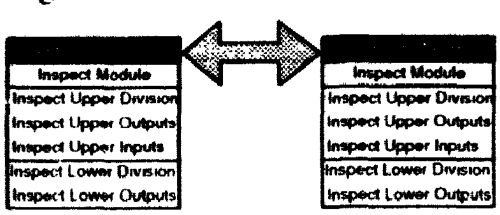


Figure 44. The workbench popup menu.

In the learning mode, the neurosolver learns the sequences of the firing columns. In a biological equivalent of the neurosolver, such signals would be coming from the environment via sensors and thalamus (only the olfactory system is an exception to this rule). There are various thalamo-cortical tracts and all of them end up somewhere in the cortex with connections to individual neurons of a column or, usually, many columns. That is simulated in the workbench by pointing with the pointing device to the specific column and using the left mouse button (the RedButton in Smalltalk-80 jargon) to increase the thalamic input to a maximum, as illustrated in Figure 45. That will cause the column to fire when the recalculation is requested.

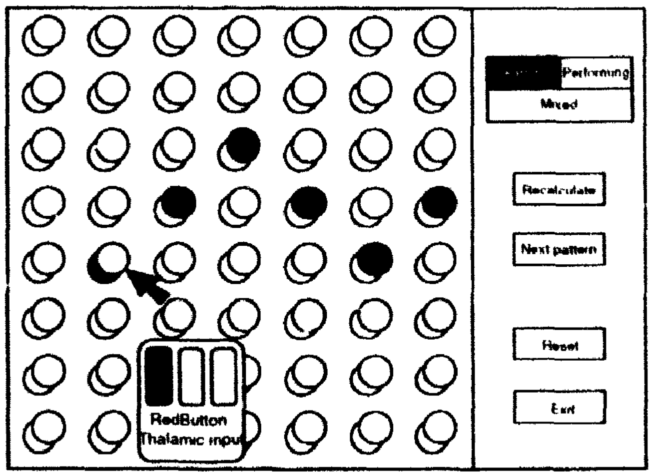
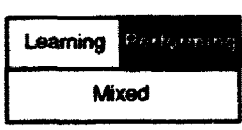


Figure 45. Using mouse to simulate thalamic inputs during learning.

To recover from mistakes, the right mouse button (the YellowButton) can be used to reset the thalamic input of the column.

After pressing  that instructs the neurosolver to recalculate and propagate the patterns of activity, the active learning schema is applied to modify inter-modular connections as described in Chapter 4.



The performing mode can be used to test the behavior of the neurosolver after learning, or in other words to use the neurosolver to compute solutions to problems in a given domain. It is useful for the case when further adaptation (learning) is not required or desired. In this mode, it is possible to change the thalamic inputs of the columns, reflecting the current state of the environment (perception), and cortical inputs, representing goals to satisfy. The mouse buttons can be used as illustrated in Figure 46. Pressing a mouse button alone increases the value of the respective input; using the mouse button with left Shift button decreases the value of the input.

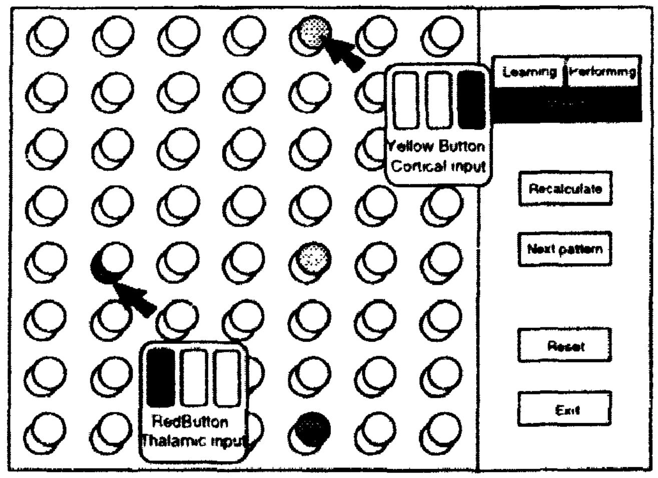
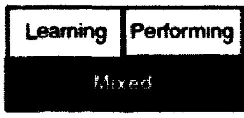


Figure 46. Using the workbench for computations.

To solve a problem, it is required to activate a goal. As explained in Chapter 4, a goal is presented by activating the upper divisions of the columns representing the desired state. That is achieved by increasing the cortical inputs that may represent, for example, wishes coming from the limbic system. The current state is set by activating the thalamic inputs of all involved columns. The color of the column representing the goal changes depending on the activity applied.

Depressing Recalculate causes recalculation of the activity and propagation throughout the network. A call tree, as described in Chapter 4, is constructed. The call tree can be observed by changing colors of the columns involved. If a column fires, its color changes to black. It may trigger other firings, so the solution path to the presented problem is marked by columns changing the color to black.

Note, that the firing columns represent the output of the neurosolver, that could be used to modify the environment. Any change to the environment would be, in turn, feed back to the neurosolver via the sensors. The person interacting with the workbench simulates that behavior. If certain columns are assigned the labels representing the concepts or features, then a shade of gray appearing at any of those columns may be a question for a clue. The labels of the columns that went black, constitute the solution of the problem



The mixed mode, in which the neurosolver solves the presented problems according to the knowledge that it learned in the past, but still adapts to the changes in the environment. The behavior that the neurosolver exhibits in this mode is very close to the beh vior of biological systems. However, continuous adaptation is not always needed or even desired in engineering applications.

The use of the workbench in the mixed mode does not differ from the performance mode. The only difference is that after recalculating the activities of the columns, the neurosolver adapts the weights of all connections to reflect the last observed pattern.

INSPECTING THE NEUROSOLVER

The workbench provides the means not only to observe the changes in the activities of the columns via the graphical user interface, but also to inspect internal states of the integral parts of the neurosolver. The menu that can be invoked with the middle mouse button (the BlueButton) is shown in Figure 47. It contains several entries with which many aspects of the neurosolver, an individual column, division or connection can be examined. After selecting any of the menu entries, the cursor changes to a crosshair and the systems waits for the user to point to the desired column. After that has been done, a Smalltalk inspector window is opened for the chosen item. The inspector can be used to display the current values of the item, its attributes or related objects as well as to change any of those values.

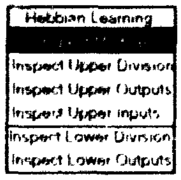


Figure 47. The inspector menu.

Selecting Inspect Module opens an instance of the inspector for a column object. With Inspect Upper Division and Inspect LowerDivision the state of the upper or lower division can be viewed respectively. Inspect Upper Outputs, Inspect Upper Inputs and Inspect Lower Outputs open inspector windows for the collections of outputs and inputs of the upper and lower divisions respectively.

THE NEUROSOLVER PARAMETERS

There are a number of system parameters that can be set before running the neurosolver workbench. The following parameters can be modified during initialization of the Division class:

- LowActivityThreshold, used in learning to determine whether the column is inhibited,
- HighActivityThreshold, used in learning to determine whether the column fired,
- LowerUpperThreshold, when exceeded the lower division inhibits the upper division and the whole column,
- LowerThreshold, when exceeded, the activity of the lower division is sent as an action potential on all of its outputs.

LearningRate can be modified in the initialization of the Connection class to set the desired rate of the learning in the hebbian mode.

Other constants used throughout the system can also be modified, but usually that would require several other related modifications.

CHAPTER 6

The rat, the maze and the neurosolver

OVERVIEW

Rats running in mazes are commonly used in research labs to test various aspects of intelligent behavior. In this work, a simulated rat maze has been built to try the neurosolver as a simple brain of an artificial rat running in the maze. The maze is shown in Figure 48.

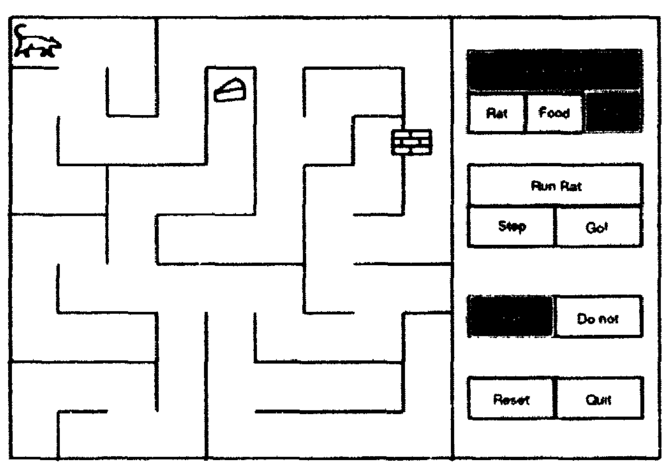


Figure 48. Rat maze with rat controlled by a neurosolver.

The maze windows consists of two parts: the maze itself and the area with the control buttons. The maze is a two dimensional matrix of a customizable number of elements, squares, with some sides of the squares, maze's walls, erased. When the walls are missing, the neighboring squares become the successive steps of the same path; i.e., the rat can move from one square to another. There are two objects that can be positioned somewhere in the maze: the rat and the food, a piece of cheese. The food is stationary, but the rat can move from one square to another – if there is no wall between the squares. The rat's movements are controlled by a neurosolver. Each square of the maze is assigned a column of the neurosolver. It is assumed that the rat can perceive the walls, therefore it will choose only valid movements. The neurosolver obtains the goal signal to get the food as a cortical input ("hunger" + position of the food) and detects the current position of the rat (thalamic inputs). The rat runs, when instructed, until the food is found.

The maze can be constructed in many ways using the maze construction buttons. The rat can be controlled by the second set of the buttons. When the rat is running, it may learn (Learn selected) or may have the learning capability disabled (Do not selected). Learn and Do not are exclusive switches (Smalltalk's OneOnSwitch).

BUILDING THE MAZE

To set the maze to the construction mode, it is necessary to press Construct Maze. This mode is exclusive with the running mode; i.e., only one of Construct Maze and Run Rat can be set (OneOnSwitchused again). One of the construction buttons, Wall, Rat or Food must be selected as well. By default, Wall is selected initially. In the wall construction mode, the cursor changes to a little picture of a part of a brick wall. The cursor can be used to erase or create walls of the maze squares. Initially, all walls are present. An existing wall can be erased by clicking the right mouse button (the YellowButton) on the wall. A wall can be posted by the same action applied to the left mouse button (the RedButton). Distances from the cursor position to all possible wall locations are calculated and the minimal distance indicates which square is affected.

After Rat is selected, the cursor changes to a small picture of a rat indicating the rat placement mode. The position of the rat can be changed by clicking with the left mouse button (the RedButton) inside one of the maze's squares.

The position of the food can be changed by exactly same action with Food selected. The cursor has a shape of a piece of cheese in the food placement mode.

Wall, Rat and Food are as well exclusive switches (OneOnSwitch).

CONTROLLING THE RAT

After constructing the maze and positioning the rat and the food, the rat is ready to begin its search for the food. To turn the running mode on, Run Rat should be selected. The two buttons that are attached to Run Rat can be used to control the rat. Pressing Step causes the rat to move one position. After making the move, the rat stops waiting for the next Step signal. If Go is selected, the rat starts to move and stops only after the food is found. Control-C can be used to break the movement, if the maze has been constructed in such a way that it is not possible for the rat to find the food.

The rat can run by itself depending on its expertise or can be guided by the pointing device. In the first case, several strategies have been tried. They can be chosen by simple modifications to the Smalltalk code. The simplest strategy, but probably the closest to the natural behavior, is to use random direction of the movement with the only restriction that the rat cannot move backwards to the position from which it moved before. That randomness is applied only if there is no memory of the past experience. If that is the case, the rat moves to the square with the strongest activity in the neurosolver's columns associated with the neighboring squares. That strategy has been chosen as a default.

Another strategy is an algorithm that guides the rat to systematically visit all allowed squares; i.e., all paths are taken until the food is found. That applies only in case when there is no activity in the columns associated with the neighboring squares. As before, in such a case, the rat moves toward the highest activity. This strategy does not possess the same biological appeal as the former.

The rat can also be guided by the pointing device by the operator. When Run Rat is selected, the cursor changes to a picture of a hand. Clicking the left mouse button (the RedButton) with the cursor positioned in any square that is a neighbor of the square in which the rat is currently located, instructs the rat to move to that location – assuming there is no wall in the way. The activity of the columns associated with the neighboring squares is not taken into account in this case.

The last control scheme is analogous to the learning techniques with a teacher. In the former schema, the learning is self-exploratory. The learning with a knowledgeable teacher is faster, because many paths that are irrelevant need not to be visited. Both algorithms for automatic control guide the rat into all possible paths, therefore the learning takes much longer.

CONCLUSIONS FROM THE RAT EXPERIMENTS

In the course of the experiments, it has been proven that the neurosolver can control the rat in the maze, assuming that the mapping is in place. The mechanism of a call tree is used to spread the activity from the position of the food (the goal) into every direction that has sufficiently strong inter-columnar upper-upper connection. The purpose of the call tree is to search for the current position of the rat. A time-out is used to move the rat in cases when no path exists yet between the current position and the goal. When there is such a path, the rat moves toward the food until it is reached and, of course, devoured. Depending on the distance from the food, the learning can be faster or slower.

After learning one path, the rat was moved into another part of the maze. Another path has been learned in the same way as the first one. The same process was repeated for many initial positions of the rat and fixed position of the food. After that, the rat was able to determine the proper path much faster, even if started from a place that had not been tried before.

When the position of the food was changed in the next series of experiments, sometimes it was easier for the rat to find the food, because parts of the paths that had been learned before could be used. Usually, however, a new learning session has to take place before the rat is efficient again.

The conclusion from the experiments with the rat and the maze is that the neursolver used to control the rat learned the orientation in the maze. In that respect, the behavior of the artificial rat is similar to the capabilities of healthy live rats running in mazes. Experiments in which healthy rats are competing against rats that were decorticated have been designed to show that such capability is provided by the cortex.

The rat maze is a very simple application of no practical use. The capability to create topology mappings is at least as important for a rat to run in a maze as the capability to record trajectories. In the next chapter, the capability to create topology mappings will be considered as a possible enhancement to the model.

During the experiments with a running rat, several improvements to the model became evident. The most important were:

- the ability to recognize paths without an end, so the rat does not follow them,

- collapsing the learning and performance phases, so the model is closer to its biological counterpart, and
- preventing self-exciting pairs of columns.

CHAPTER 7

Conclusions and directions for further work

An artificial column presented in this work exhibits very interesting behavior when applied en masse in a proper way. In the opinion of the author, it is worth to pursue the research. There are many shortcomings that must be dealt with before the neurosolver can be used in any practical application. Much more research will be needed before the neurosolver reaches its ultimate capabilities as envisioned in Chapter 4. In this section, the deficiencies are described and future plans for overcoming some of the problems and, generally, improving the neurosolver are indicated.

INADEQUACIES OF THE MODEL

Architecture

Burnod's upper-upper connection proved to serve well as the basic element of the model that provides the capability to create a call tree. However, the ability to create action trees required to modify Burnod's model. The lower-upper connection that has been added to the model serves that purpose. The interactions between the upper and lower divisions had also be implemented to provide a firing mechanism.

A completely interconnected network is difficult to simulate in software because of the large number of computations that are required. The model that is connected along the vertical, horizontal and diagonal lines requires that the columns of any pattern must be neighbors in the neurosolver. Otherwise, there are no connections, so the pattern cannot be learned.

Learning rules

The learning rules that use simple probability appeared to be too weak to develop more complex behavior in a regularly interconnected network. Their complexity was increased to include more statistical factors and additional probabilities were defined. A formula for the strength of a connection that uses a combination of the probabilities proved to be better to record temporal relationships. The modified rules proved to be suitable for the basic features of the neurosolver, i.e. creation of call and action trees. They were relatively easy to implement, because the neurosolver is simulated in software.

The hebbian learning tried in the model proved to be much weaker than the probabilistic learning. The hebbian learning uses simple rules that modify the strength of a connection by an amount that is a function of the previous strength and a constant. If the constant is small, the learning is slow. If the constant is larger then there are many action trees created quickly. From the computational point of view action trees are less desired than call trees.

Oscillation

A mechanism that prevents two columns being self-excitatory had to be built. Without such a mechanism, if one of two columns that are neighbors in two different call trees that spread in opposite directions sends an action potential to another, then the latter responds with the same signal. That happens because the activity is prone to spread in the opposite direction along another call tree. Although initially the signals are lower than the firing threshold, they are amplified in an oscillating loop and finally both columns fire.

Cycles

Although the mechanism that prevents a self-excitatory pairs of columns was implemented, it is still possible to have sequences that generate larger cycles. If one of the the columns gets activated, the activity spreads to all columns in the cycle. The columns in the cycle will be sending stronger and stronger action potentials and, finally, some of them may fire.

System stability

The neurosolver is not guaranteed to stabilize after some activity is applied. Usually, the system stabilized if small number of sequences were stored. However, if there are many overlapping trees, than the equilibrium might not be achieved at all. The neurosolver is not formalized enough to attempt to find the rules that govern its stability.

By applying the anti-oscillation mechanism, improving the learning formulas and adding the shutting down mechanism after a column fires the stability was improved.

Limited storage capacity

The behavior of the neurosolver is much closer to the ideal if only a small number of patterns are recorded. When that number grows and, additionally, there are many overlapping trajectories registered, than the system computing capabilities deteriorate.

Fixed thresholds

All thresholds are fixed in the model. They were chosen through experiments. It is still not certain what activity levels should be used, for example, in a call tree or in the trigger. Some call trees require higher activity in the root than others. That might be consistent with biology, but the use of activation levels in the neurosolver requires further investigation.

I/O system

Any column in the network contributes to the I/O system of the neurosolver. Usually, that is neither desired nor required. However, there is no mechanism in the model that would provide an adaptable I/O system.

Software implementation

A software implementation is a convenient way to simulate the behavior of the neurosolver. However, it is very inefficient due to the large number of connections to manage. Another deficiency of the simulator is that it is implemented on a sequential computer, so the parallelism is artificial.

DIRECTIONS FOR FURTHER WORK

Learning schema

The hebbian modification rules used in this work were very simple. To be a viable alternative to the probabilistic rules, they would have to be extended to include more important parameters that do influence the associations between columns, like the decay and inhibition factors. It was suggested earlier that it would be possible to implement a hebbian equivalent to the probabilistic rules that were used. That work needs further exploration. There are many formulas used in the theory of classical neural networks that should be tried be tried.

In the model presented in this thesis, the action potential is active only for one step, a tick, of the simulator. It would be beneficial to make the action potential last longer, so more distant elements of learned sequences can contribute to the activity of a column. The problem with using only first order action potentials is illustrated in Figure 49.

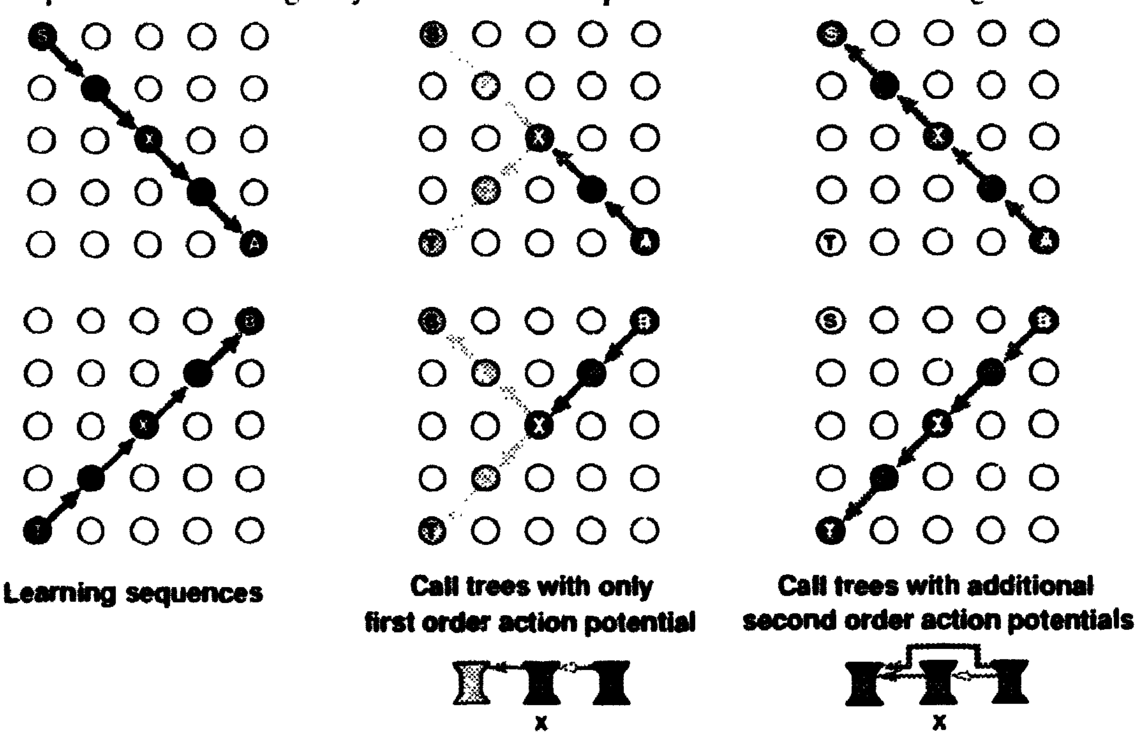


Figure 49. The problem with crossing paths and its solution.

Two trajectories are learned: from column S to column A and from column T to column B. The patterns cross at the column X. There is neither path from column S to column B nor from column T to column A. However, when a call tree is activated from column A, one of the branches leads to column T. Similarly, if column B is the goal, the search leads to column T and to column S. It happens because neither of the successors of the column X in any of the call trees is influenced by the nodes that precede the crossing point. The activity is lower in the branches to the left of column X, because the strengths of the connections are based on probabilities that use statistics collected during the learning.

If a second order action potential was used, i.e., such that takes two simulating steps, ticks, to get to the receiver, than the problems would be overcome. The node that precedes the crossing point at column X, would contribute only to the successor of column X that belongs to the same learned trajectory. The formula for calculating the activity of a column would have to reflect that influence. Each connection would have not one but two strengths: for the first and second order action potentials.

The idea of a second order action potential might be expanded to a general case of n-order action potentials. An n-order action potential accompanied by proper formulas for the strength of the connections might be helpful in overcoming the problem with the capacity of the neurosolver to store various patterns that is poor in the current implementation.

Use of the temporal inhibition

The modification rules for the connections were stated only for excitatory cases. Similar rules can be defined for the case of one column inhibiting another. The algorithms for that have been implemented in the model, but were not tested. A considerable amount of time will likely have to be spent refining and expanding those rules. The schema for learning inhibitory relationships would be analogous to those used for the excitatory connections.

The interaction between the excitatory and inhibitory rules is another area that should be thoroughly investigated.

Adaptable columnar parameters

The internal parameters of each column were fixed in the model. Various values were tested and those that satisfied the requirements best were put in place. Of course, this is very subjective. It would be much better if all parameters were adaptable in a way similar to the modification rules for the connections. In that way, each column could have developed a behavior that best suits the representation that the column stands for. In particular, the firing threshold might be tuned to the overall level of activity in which the column is involved .

To achieve adaptivity of the columnar parameters, better understanding of the processes that occur inside the biological cortical column will be required.

Adaptable intra-columnar connections

The connections between the lower and upper division of the column were fixed in the model. That is not what happens in the cerebral cortex. The adaptivity of those internal connections might be important, though without further investigation it is hard to say what is exactly the impact of those connections on the behavior of the network.

Topology mapping using lateral inhibition

In this work, the topological mapping has been taken for granted. Another technique, for example the Kohonen algorithm, was suggested for the mapping. In Chapter 2, however, it has been indicated that there are local inhibitory interconnections between the neighboring columns. That inter-connectivity resembles the connectivity between nodes in the Kohonen architecture (in fact any on-center-off-surround architecture). Those inhibitory connections might be used to generate topological maps. That process could precede the learning of temporal sequences that was the subject of this thesis. More interesting, however, would be to let the process continue, allowing in that way re-mapping of the topological relationships if the environment changes. The neurosolver could, hence, evolve with the system that it controls.

Associating inputs with sensors and outputs with effectors

In the model, each column may receive external input and generate external output. The input signals are incoming from the sensory system. The output signals are transmitted to the effectors that may manipulate the subject system. The signals are transmitted without any loss in their values. That is different from the equivalent biological systems. The afferents and efferents of each column could also be trained, so they do not have to be assigned a priori and may change in time in an attempt to adapt to arising novelties. For example, if the effectors were severed or altered, the neurosolver might have tried to utilize the resources in the best possible way.

VLSI

The neurosolver from its conception was thought to be a universal computing device. In that expect, it is very similar to electronic devices. Any chip can be used in many applications without re-engineering it. What is required to change the functionality performed by the chip is just an alternate connectivity with other devices.

The ultimate neurosolver should also be implemented in hardware. Depending on the task and the size of the neurosolver chip, only a single module, a chip, or many modules would be used. If the problems that were indicated earlier in this chapter were solved, then the neurosolver would be completely self-programming, i.e. the interfaces to other devices as well as inter-connectivity between different neurosolvers would be generated automatically. It would be possible to have a library of pre-programmed chips that would be used in additional learning sessions during which they would accommodate each other and adapt to the system within which they would work.

The software simulator has three types of inter-columnar connectivity schema possible, but the completely interconnected model is hard to use because of the large number of connections that are required to test and, perhaps, modify in each cycle. That might be a lesser problem in a hardware implementation. An example of possible hardware architecture is illustrated in Figure 50. Each node, in that example, sends vertical and horizontal double links that are connected with all other links. The connections are adaptable and assigned in the way suggested in the figure. The rows correspond to the columns that precede another in a sequence. For example, the connections α and β are modified for the sequence B → D. α corresponds to the lower-upper connection between B and D, and β - to the upper-upper connection between D and B. The case for D → B is shown as well.

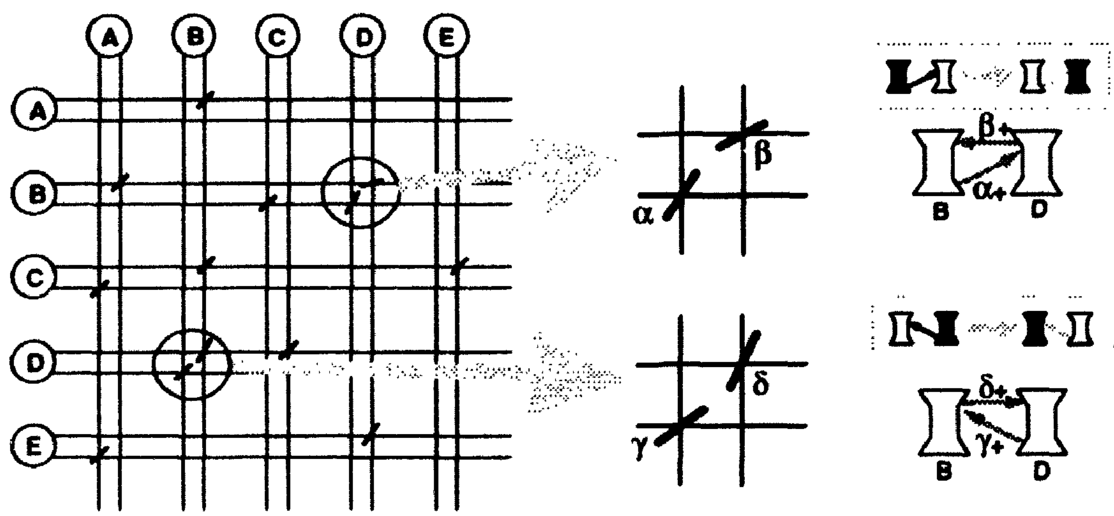


Figure 50. A possible hardware architecture for a neurosolver.

The architecture proposed in Figure 50 implements a completely inter-connected neurosolver. There are no simulation cycles, so every column is updated at the same time. It was a challenge to built such parallelism into the software simulator. It will be even more challenging to provide an adaptation mechanism that could be used in the implementation of similar functionality in hardware.

It must be noted that computing paradigm that uses connection strengths based on probabilities might be difficult or impossible to repeat in hardware.

Applications of the neurosolver

To employ its full potential, the neurosolver needs to be applied in more realistic and useful applications than the simple rat application described in this thesis. In that application, the feature most used was the ability to perform a breadth-first search. The sensor triggering capability was not used explicitly. Another important feature, i.e., the partial parallel goal specification was not needed at all in the rat application.

Some of the applications that would exercise all aspects of the computational paradigm of the neurosolver were suggested in Chapter 4. One of them is a robot controller that would provide the robot with the ability to visually guide its movements. Another application is a diagnostic system that uses the learned trajectories relating to past system deficiencies and recovery routines to diagnose complex failures by dividing the problem and resolving each sub-problem in parallel. In a more sophisticated incarnation of the latter, the neurosolver would act as a controller that can learn the behavior of a system, so any abnormality can be detected and a correction procedure performed or suggested to an operator.

The neurosolver will reach its ultimate form only through enhancements that are the consequence of attempts to use it successfully in more and more challenging applications.

BIBLIOGRAPHY

[1] Abeles, M., *"Corticonics"*, Cambridge University Press, Cambridge, 1991.

[2] Alexandre, F., Guyot, F., Haton, J. P. and Burnod, Y., *"The cortical Column: A New Processing Unit for Multilayered Networks"*, Neural Networks, Vol. 4, pp. 15-26, Pergamon Press, New York, 1991.

[3] Anderson, J. et al, eds., *"Neurocomputing"*, Vol. 1, 2, MIT Press, Cambridge, Vol. 1, 1988, Vol. 2., 1990.

[4] Ballard, D. H., *"Cortical Connections and Parallel Processing: Structure and Function"*, Behavioural and Brain Sciences, 1986.

[5] Broadmann, K., *"Vergleichende Lokalisatiolehre der Grosshirnrinde in Ihren Prinzipien Dargestellt auf Grund des Zellenbaues"*, J. A. Barth, Leipzig, 1909.

[6] Broca, P, *"Perte de la Parole"*, Bulletin de la Société d'Anthropologie, Vol. 2, pp. 235-237, 1861.

[7] Burnod, Y., *"An Adaptive Neural Network: The Cerebral Cortex"*, Masson, Paris, 1988.

[8] Coghill, G. E., *"Anatomy and the Problem of Behavior"*, Cambridge University Press, Cambridge, 1929.

[9] Dingledine, R., ed., *"Brain Slices"*, Plenum Press NY, 1984.

[10] Grossberg, S., *"The Adaptive Brain"*, Elsevier/North-Holland, Amsterdam, 1987.

[11] Hampson, S. E., *"Connectionistic Problem Solving: Computaional Aspects of Biological Learning"*, Birkhaüser, Boston, 1990.

[12] Hebb, D.O., *"The Organization of Behaviour: a Neuropsychological Theory"*, Wiley, New York, 1949.

[13] Holmes, G. M., *"Selected Papers"*, Walshe, F. M. R. Ed., 1956.

[14] Hopfield, J. J., *"Neural Networks and Physical Systems with Emergent Collective Computational Abilities"*, Proceedings of the National Academy of Sciences, Washington, 1982.

[15] Hubel, D. H. and Wiesel, T. N., *"Laminr and Columnar Distributionof Geniculo-Cortical Fibers in the Macaque Monkey"*, Journal of Computational Neurology, 1972.

[16] Kandel, E. and Schwartz, J., *"Principles of Neural Science"*, Elsevier/North-Holland, Amsterdam, 2nd edition, 1985.

[17] Kohonen, T., *"Self-Organization and Associative Memory"*, Sprinder-Verlag, Berlin, 2nd edition, 1988.

[18] Kolb, B. and Whishaw, I. Q., *"Fundamentals of Human Neuropsychology"*, W. H. Freeman and Company, New York, 2nd edition, 1984.

[19] Lorente de Nó, R., *"Cerebral Cortex: Architecture, Intracortical Connections and Motor Projections"*. In "Physiology of the Nervous System" Fulton, J., ed., Oxfrod University Press, London, 1943.

[20] von der Malsburg, C., *"Self-Organization of Orientation Sensitive Cells in the Striate Cortex"*, Kybernetik, 1973.

[21] McClelland, J. L. and Rumelhart, D. E., *"Parallel Distributed Processing"*, Vol. 1, 2, MIT Press, Cambridge, 1986.

[22] McCulloch, W. S. and Pitts, W., *"A Logical Calculus of the Ideas Immanent in Nervous Activity"*, Bulletin of Mathematical Biophysics, 1943.

[23] Mountcastle, V. B., *"An Organization Principle for Cerebral Function: the Unit Module and the Distributed System"*. In "The Mindful Brain", Schmitt, F. O., MIT Press, Cambridge, 1978.

[24] Penfield, W. and Jasper, H. H., *"Epilepsy and the Functional Anatomy of the Human Brain"*, Little, Brown, Boston 1954.

[25] Peters, A. and Jones, E.G., eds., *"Cerebral Cortex"*, Plenum Press, New York, 1984.

[26] Ramón a Cayal, S., *"Estudio Sobra la Cortezza Cerebral Humana"*, Rev. Trim. Microscopia, 4, 1899.

[27] Ramón a Cayal, S., *"Recollections of My Life"*, Memoirs of the American Philosophical Society, Vol. 8, 1937.

[28] Rayner, C., ed., *"The Atlas of the Body and Mind"*, Crescent Books, New York, 1990.

[29] Rosenblatt, F., *"The Perceptron: a Probabilistic Model for Information Storage and Organization of the Brain"*, Psychological Review, 1958.

[30] Schmitt, F. et al eds., *"The Organization of the Cerebral Cortex"*, MIT Press, Cambridge, 2nd edition, 1985.

[31] Szentagothai, J., *"The Module Concept in Cerebral Cortex Architecture"*, Brain Research, 1979.

[32] Shepard, G. M., *"Synaptic organization of the brain"*, Oxford University Press, New York, 3rd edition, 1990.

[33] Shepard, G. M., *"Neurobiology"*, Oxford University Press, New York, 2nd edition, 1988.

[34] Wernicke, C., *"Der Aphasische Symptomencomplex"*, M Cohn et Weigert, Breslau, 1874.

[35] White, E. L., *"Cortical Circuits: Synaptic Organization of the Cerebral Cortex. Structure, Function and Theory"*, Birkhaüser, Boston, 1989.

[36] Widrow, G. and Hoff, M. E., *"Adaptive Switching Circuits"*, Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, Part 4, New York, 1960.

[37] Zeki, S. and Shipp, S., *"The Functional Logic of Cortical Connections"*, Nature, Vol. 335, pp. 311-317.

APPENDIX A

Formal neurosolver

In this appendix, an attempt is made to formalize the neurosovler as a state machine.

First, we define a **Neurosolver universe** to be a system:

$$\mathcal{U} = < \mathcal{C}, \mathcal{E}, \mathcal{R}, \mathcal{M} >$$

where $\mathcal{C}$ is the neurosolver $\mathcal{E}$ is an **environment,** the system that is being watched after, $\mathcal{R}$ is a system of **sensors** or **receptors** and $\mathcal{M}$ is a system of **effectors** or **manipulators**.

We define the **Neurosolver** $\mathcal{C}$ as:

$$\mathcal{C} = < \mathcal{N}, \mathcal{S}, \mathcal{B} >$$

where $\mathcal{N}$ is a set of **columns (nodes)**, that, in fact, are pairs of **upper** and **lower divisions**:

$$\mathcal{N} = \{ N_1, N_2, \ldots, N_Z \}, \qquad N_\zeta = (N_{U\zeta}, N_{L\zeta});$$

$\mathcal{S}$ is the **domain** of the Neurosolver states:

$$\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \ldots)$$

and each state $\mathcal{S}_\lambda$ consists of the states of all columns:

$\mathcal{S}_\lambda = (S_{N_1}, S_{N_2},\dots,S_{N_Z})$, $\lambda = 1, 2, 3,\dots$

The state of column N_K can be represented as a pair of an **upper division's state** and a **lower division's state**:

$S_{N_K} = (S_{U_{N_K}}, S_{L_{N_K}})$.

$\mathcal{B}$ is the Neurosolver's **behavior**, and it consists of two functions: **state transformant** $\mathcal{T}$ and **adaptive function** $\mathcal{F}$:

$\mathcal{B} = (\mathcal{T}, \mathcal{F})$

The state transformant defines the Neurosolver's behavior as a state machine, while the adaptive function determines its adaptive capabilities.

$\mathcal{T}: (\mathcal{S}, \mathcal{I}) \rightarrow (\mathcal{S}, \mathcal{I}_C)$

where $\mathcal{S}$ is the state domain, and $\mathcal{I}$ is the domain of the Neurosolver's **inputs** consisting of two components: $\mathcal{I}_T$, **thalamic** (external) inputs, and $\mathcal{I}_C$, **cortical** (internal) inputs:

$\mathcal{I} = (\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3,\dots)$

$\mathcal{I}_\lambda = (\mathcal{I}_{\lambda T}, \mathcal{I}_{\lambda C})$, $\lambda = 1, 2, 3,\dots$

$\mathcal{I}_{\lambda T} = (I_{TN_1}, I_{TN_2},\dots, I_{TN_\Omega})$

$\mathcal{I}_{\lambda C} = (I_{CN_1}, I_{CN_2},\dots, I_{CN_\Omega})$

Each I_{TN_ω} represents the **thalamic input**, and I_{CN_ω} represents the **cortical input** to a column.

The transformant $\mathcal{T}$ can also be expressed in the following, distributed way:

$$\mathcal{J} = \bigcup \{ T_{N_K} : \kappa = 1, 2, \ldots, K,\ T_{N_K} : (\mathcal{S}_{N_K}, \mathcal{I}_{TN_K}, \mathcal{I}_{CN_K}) \rightarrow (\mathcal{S}_{N_K}, \mathcal{I}_{C\mathcal{N}\vartheta}) \}$$

$$\mathcal{I}_{C\mathcal{N}\vartheta} = (\mathcal{I}_{1C\mathcal{N}\vartheta}, \mathcal{I}_{2C\mathcal{N}\vartheta}, \mathcal{I}_{3C\mathcal{N}\vartheta}, \ldots)$$

$$\mathcal{I}_{\lambda C\mathcal{N}\vartheta} = \{ I_{CN\varphi} : N_\varphi \in \mathcal{N}_\varphi, \qquad \mathcal{N} \supset \mathcal{N}_\varphi \}, \qquad \lambda = 1, 2, 3, \ldots$$

If $\vartheta = \Omega$ (i.e. the column outputs to all others in the Neurosolver), then:

$$N_\varphi = N,$$

and:

$$\mathcal{I}_{C\mathcal{N}\vartheta} = \mathcal{I}_C.$$

There are two divisions of a column N_κ, and each of them has its own state: S_{UN_K} and S_{LN_K}. Each partial transformant T_{N_K} consists of two components upper:

$$T_{UN_K} : (\mathcal{S}_{LN_K}, \mathcal{S}_{UN_K}, \mathcal{I}_{TN_K}, \mathcal{I}_{CN_K}) \rightarrow (\mathcal{S}_{UN_K}, \mathcal{S}_{LN_K}, \mathcal{I}_C)$$

and lower:

$$T_{LN_K} : (\mathcal{S}_{LN_K}, \mathcal{S}_{UN_K}, \mathcal{I}_{TN_K}, \mathcal{I}_{CN_K}) \rightarrow (\mathcal{S}_{UN_K}, \mathcal{S}_{LN_K}, \mathcal{I}_C).$$

Finally, for each column, we can define the following transformants:

- **upper division state transformant**:

$$S_U = T_{SU} (S_L, S_U, I_T, I_C)$$

- **lower division state transformant**:

$$S_L = T_{SL} (S_L, S_U, I_T, I_C)$$

- **internal feedback function**:

$$I_{\lambda C} = T_{oU}(S_U, I_T, I_C)$$

Calculations of T_{sU}, T_{sL} and T_{oU} usually involve a system of inter-nodal and inter-divisional connections with appropriate strengths.

The second component of the Neurosolver's behavior, the adaptive function $\mathcal{F}$, is a novel entity for a state machine, since it allows the state tranformant $\mathcal{T}$ to be altered, so the machine adapts to changes in the environment. If we designate $\mathcal{Q}$ as the domain of the state transformant $\mathcal{T}$, we have the following:

$$\mathcal{F}: \mathcal{Q} \rightarrow \mathcal{Q}.$$

The function is realized by modifications to the strengths of the inter-columnar connections. Similar considerations for the state transformant lead us to the following partial adaptive functions for each column:

$$F_{sU}: \mathcal{Q}_{sU} \rightarrow \mathcal{Q}_{sU}$$

$$F_{sL}: \mathcal{Q}_{sL} \rightarrow \mathcal{Q}_{sL}$$

$$F_{oU}: \mathcal{Q}_{oU} \rightarrow \mathcal{Q}_{oU}$$

Environment $\mathcal{E}$ interacts with the Neurosolver through receptors $\mathcal{R}$ and manipulators $\mathcal{M}$, creating in that way an external feedback loop.

Receptors supply the Neurosolver with the image of the environment.

$$\mathcal{R} = \{ \mathcal{R}_\alpha: \alpha = 1, 2, \ldots, A, \mathcal{R}_\alpha = \bigcup R_{\alpha\beta}, \beta = 1..B, R_{\alpha\beta}: \mathcal{E} \rightarrow \mathcal{I}_{TN\beta}, N_\beta \in \mathcal{N}_{\alpha} \subset \mathcal{N} \}$$

If B = Z, ie. every column receives an input, then:

$$\mathcal{R} = \{ R_\alpha : \alpha = 1, 2, \ldots, A, R_\alpha = \bigcup R_{AN}, \forall N \in \mathcal{N}, R_{AN} : \mathcal{E} \rightarrow \mathcal{S}_{TN} \}$$

Manipulators modify the environment depending on the state of the Neurosolver:

$$\mathcal{M} = \{ \mathcal{M}_\gamma : \gamma = 1, 2, \ldots, \Gamma, \mathcal{M}_\gamma = \bigcup M_{\gamma\varepsilon}, \varepsilon = 1..E, M_{\gamma\varepsilon} : \mathcal{S}_{LN\varepsilon} \rightarrow \mathcal{E}, N_\varepsilon \in \mathcal{N}_\gamma, \mathcal{N} \supset \mathcal{N}_\gamma \}$$

If E = Z, ie, each column contributes to the output of the neurosolver, then:

$$\mathcal{M} = \{ \mathcal{M}_\gamma : \gamma = 1, 2, \ldots, \Gamma, \mathcal{M}_\gamma = \bigcup \mathcal{M}_{\Gamma N}, \forall N \in N, \mathcal{M}_{\Gamma N} : \mathcal{S}_{\Lambda N} \rightarrow E \}$$

Both receptor and manipulator functions are also realized by a system of appropriate connection strengths between columns and receptors or manipulators. They can be adaptive in a manner similar to the state transformant, but we have chosen to fix interactions with the environment.

APPENDIX B

Smalltalk-80 code for the model of a cortical column

The code that is presented in this chapter is not a complete implementation of the neurosolver. Only the most important parts of the model are included. Less important details of the model and the user interface are omitted for clarity.

The code does not include the Rat Maze application.

```
OneOnSwitch variableSubclass: #Neurosolver
    instanceVariableNames: 'columns mode learningMode '
    classVariableNames: ''
    poolDictionaries: ''
    category: 'Neurosolvers'
```

Neurosolver comment:

Neurosolver is a network for problem solving. It consists of interconnected columns with high degree of organization. The organization may be achieved by a self-organizing algorithm, or may be specified explicitly. Some of the columns may have additional external inputs (thalamic) and may output their signals outside the network (manipulators). Those additional interfaces are the basis for the external feedback. Internal feedback is realized by high inter-connectivity between the columns. Neurosolver may be a flat architecture, but its behavior is more interesting when some hierarchies are involved. Any hierarchy may be included in a flat model, but the architecture is easier to deal with if there are many interconnected cortical layers representing levels of abstraction. Such maps may be self-organized separately and later brought together through associative algorithms. The third step would be learning to solve problems.

NEUROSOLVER METHODSFOR: 'control'

learningStep

```
    "organizes the activity calculations and connection adjustments during the
learning"
    Cursor wait showWhile: [
        self calculateActivityPattern.
        self actualizeActivity.
        self calculateModificationFactors.
        self adjustConnections.
        (learningMode = #hebbian) ifTrue: [self resetInfluenceFactors].
        self resetAfterFiring.
        self setNextTick.
```

```
        ]
```

performanceStep

"organizes the activity calculation scheme during the performance"

"for every column recalculate activities, determine outputs and show all firing columns"

```
Cursor wait showWhile: [
        self inhibitAfterFiring.
        self resetCurrentActivity.
        self calculateActivityPattern.
        self commitActivity.
        self propagateActivity.
        self resetCorticalInputs.
        self setNextTick.
]
```

```
Object subclass: #Column
	instanceVariableNames: 'upperDivision lowerDivision upperLowerConnection lowerUpperConnection columnID neurosolver positionPoint '
	classVariableNames: ''
	poolDictionaries: ''
	category: 'Neurosolvers'
```

A column is a unit loosely based on a cortical column. It consists of two divisions, lower and upper, and many connections to other columns or other parts of the system (receptors, manipulators).

A column represents a concept or a part of a concept in the organized network. It connects to other columns which are somehow related to it. That inter-connectivity is achieved by self-organization or built-in by the network architect.

The basis for the active problem solving is an ability to generate sequences, since the activity representing a goal must be spread through the network to search for the solutions. In this model, the activity of the upper division represents a sub-goal in the search tree. That activity is prolonged (i.e.. the sub-goal is stacked): it ceases only if the columns representing its parent's goals are deactivated as well, or if the column fired (high activity in the lower division occurs). In the first case, most of inputting columns stops sending their signals (they could be themselves in the same situation as the column in question). In the second case, the lower division inhibits the activity in the upper division. There might be an action associated with the firing of the lower division, for example a sound or movement generation, which makes the cortex problem capabilities actually useful.

If the same set of child columns is the cause for firing the parent column, that column can is equivalent to a constant. if there are many such sets, the column exemplifies a variable (it is a place holder).

COLUMN METHODSFOR: 'initialization'

initialize

```
	""initialize a column"

	"create upper division"
```

```
"The lower division transmits its activity all the time (i.e. the threshold is always
0). Spreading of the column's activity represents the search for the solution to the
problem expressed by that column."

        upperDivision := (Division new)
                            initializeFor: self
                            withActivity: 0
                            threshold: (Division upperThreshold).

        "create lower division"
        "The activity of the lower division expresses the degree of the satisfaction of the
constraint (goal) represented by this column. If that degree is high enough, the sub-
goal can be assumed as achieved and the reason for the activity in the upper division"

        lowerDivision := (Division new)
                            initializeFor: self
                            withActivity: 0
                            threshold: (Division lowerThreshold).

        "intra-column upper-lower and lower-upper connections are realized taken into
account while computing the activity of the column"

        ^self

connectTo: conn
        | newConnection |
"connect the column's divisions with the divisions of the column passed as the
parameter"

 "There are two types of inter-conlumnar connections: upper-upper and lower upper. "

"upper-upper - feedback direction"
        newConnection := UpperUpper new.
        newConnection initializeWith: 0 from: self upperDivision to: conn upperDivision
        self upperDivision outputs add: newConnection.
        conn upperDivision inputs add: newConnection.
```

```
"lower-upper - feed forward direction"
    newConnection := LowerUpper new.
    newConnection initializeWith: 0 from: self lowerDivision to: conn upperDivision .
    self lowerDivision outputs add: newConnection.
    conn upperDivision inputs add: newConnection.
```

COLUMN METHODSFOR: 'control'.

calculateModificationFactors

```
    "count exciting or inhibiting columns for both divisions"

    upperDivision calculateModificationFactors.

    lowerDivision calculateModificationFactors.
```

commitActivity

```
    "commit the calculated activities. It has been postponed until now, because the
action potentials might have been set incorrectly, before we know whether the columns
fires inhibiting, in such a case, the upper division"

    upperDivision commitActivity.
    lowerDivision commitActivity.

    "The lower division inhibits the upper division. High activity in the lower division
means that the subgoal represented by that division is satisfied. That happens if the
activity of the upper division and the input from the outside (external feedback loop)
sum up to a value higher than the threshold."

    self highActivated ifTrue: [upperDivision realActivity: 0.0].
```

inhibitAfterFiring

```
    "The column is not responsive for a while after firing"

    (((upperDivision wasPrevHighActivated)
```

```
        or: [upperDivision wasHighActivated])
        or: [upperDivision highActivated])
    ifTrue: [upperDivision resetInputPotentials].
```

modifyConnections

"if the column fires (lower division's activations goes sufficiently up) we adjust connections from the upper division to all columns which were highly activated before. This strenghtens the preferred paths in the search tree."

```
    self fired ifTrue:
        [upperDivision outputs do:
            [ :connection |
                connection receiver column firedBefore
                    ifTrue: [connection increasePositiveStrength]
            ]
        ].

    "perform the adjustments for the inhibitory connections"
    self wasInhibited ifTrue:
            [ upperDivision outputs do: [ :connection |
                connection receiver column firedBefore
                    ifTrue: [connection increaseNegativeStrength]
            ]
        ].
```

"all connections from the lower division to the upper divisions of columns changing their states are strengthen under the condition that this column was one of the reasons for that change, i.e. the lower division activity was previously high."

```
    self firedBefore ifTrue:
        [lowerDivision outputs do:
            [ :connection |
    "for excitatory connections"
                connection receiver column fired ifTrue:
                    [connection increasePositiveStrength].
    "inhibitory connections"
                connection receiver column wasInhibited ifTrue:
```

```
                    [connection increaseNegativeStrength]
            ]
        ].
```

COLUMN METHODSFOR: 'state calculations'.

calculateActivity

"calculate the activities of both divisions of the column"

"Take into account previous activity of the upper division. The upper division has integrating capabilities"

| lowerActivity |

upperDivision calculateActivity: 0.0.

"The upper division excites the lower division. Low activity in the upper division means that the subgoal represented by the column (or part of the subgoal, since a goal may have a multicolumn representation) is being searched. The activity is transmitted to the lower division and , if the lower column is not excited enough to fire, also to *other columns. Other columns may also send their signals to this column, exciting it even more (internal feedback loop)."*

"It is assumed here that the upper division's activity is completely transmitted to the lower division (the current activity is stored in tempActivity before it is committed at the end of this method)"

lowerDivision calculateActivity: (upperDivision tempActivity).

"the column fires - both divisions are high"

```
((lowerActivity := lowerDivision tempActivity) > Division lowerUpperThreshold)
            ifTrue: [upperDivision calculateActivity: lowerActivity].
```

"display the state of the column"

neurosolver changed: #displayColumnState: with: self.

Object subclass: #Division

instanceVariableNames: 'column inputs outputs realActivity curActivity prevActivity prevPrevActivity tempActivity threshold activityUpCount activityDownCount numberOfPositiveInfluences numberOfNegativeInfluences numberOfPositivelyInfluenced numberOfNegativelyInfluenced thalamicInput corticalInput '

classVariableNames: 'HighActivityThreshold LowActivityThreshold LowerThreshold LowerUpperThreshold UpperThreshold '

poolDictionaries: ''

category: 'Neurosolvers'

Division comment:

This is a class representing a cortical column's division. There are two such division within a column: upper and lower. They are physically the same, but their behavior in the

network is quite different.

The upper division is active if a concept represented by the column is searched for. If the

cortex activates that division it can be read as: "is the concept A present under the current

conditions?". If there are many other columns transmitting to this one, it may happen that the goal represented by the question is achieved. It could read as: "from the set of active concepts (columns) is it justifiable to assume A?". All those transmitting columns represent the sub-goals required to achieve the goal exemplified by this column. Something else may occur: the receptors send excitatory signals to the column which would be read as: "from what is known about the state of the world, the concept A may be assumed". The state of the external world can be understood as the current set of the facts known to the system. Forcing the parent goal (or concept) to wait for all its sub-goals to be satisfied is similar to pushing on a stack. After the goal is achieved (i.e. the conditions for that are satisfied, which better expresses what actually is going on), the reason for the activity of the division is gone. In our architecture it is accomplished by the inclusion of another division, lower, into a column.

The lower division of a column takes all activity from the upper division, and additionally from the outside. If the activity is high enough, the lower division send an inhibitory signal *to the upper division. That signal forces the activity of the upper division to diminish (the goal has been achieved) which is equivalent to popping from a stack. The activity of the lower division disappears as well, since it came from the upper one, and most of associated external signals incoming to the division (from the receptors - thalamic system) are short, so the whole column is deactivated. The lower division send its signals to the upper divisions of related columns (next in a sequence) increasing their activity. That connection represents the direction from the sub-goal to its parent goal. If the signal is strong enough, the parent column fires as well.*

Input signals from the thalamic system are able to activate the upper division to such extent

that it causes the high level of activity in the lower division as well. The column fires meaning that the goal has been achieved. Input signals are not persistent, so the activity of the column declines.

```
DIVISION METHODSFOR: 'state calculations'

calculateActivity: value
	"calculate new activity of the node with a startup value"

	| inputActivity |

	inputActivity := value + (self getInputActivity).

	"include thalamic and cortical inputs. The thalamic input comes from the
sensors, through thalamus to the lower division. The cortical input comes from other
areas of the cortex, for example from the limbic system - wishes, to the upper division"
	"Note: if learning, there are only high, firing thalamic input (learning) signals"
	inputActivity := inputActivity + thalamicInput + corticalInput.

	"assume that 1 is maximum"
	(inputActivity > 1.0) ifTrue: [inputActivity := 1.0].

	"set the activity level to the calculated value.
```

NOTE: The activity is stored in a temporary value until it is committed after determinig whether the column fires or not. If the column fires, the activity of the upper division is set to zero, so no action potential is sent onto the outputs."

```
	self tempActivity: inputActivity.
	self realActivity: inputActivity.
```

DIVISION METHODSFOR: 'control'.

getInputActivity

"scan inputs for action potentials and sum them up. Discard signals from the divisions which were influenced by this division"

```
	| inputActivity |

	inputActivity := 0.

	self inputs do: [:connection |
		(connection transmitter wasInfluencedBy: (self column columnID))
		ifFalse: [
			inputActivity := inputActivity
				+ (connection strength * connection actionPotential).

			connection prevActionPotential: connection actionPotential.
		]
	].

	^inputActivity
```

calculateModificationFactors

"this method calculates a fan-in and fan-out of activity from the perspective of a single division"

```
	| tempFanIn |
	tempFanIn := 0.
```

```
(self activityUp)
ifTrue: [
        activityUpCount := activityUpCount +1.
        numberOfPositiveInfluences := numberOfPositiveInfluences
                                        + (tempFanIn := (self fanIn)).
].

(self activityDown)
ifTrue: [
        activityDownCount := activityDownCount + 1.
        numberOfNegativeInfluences := numberOfNegativeInfluences
                                                + tempFanIn.
].

self setFanOut.
```

fanIn

```
"determine how many inputs influenced behaviour of this division"

| tempFanIn |

tempFanIn := 0.
inputs do: [:connection |
        (connection transmitter column firedBefore) ifTrue: [
                tempFanIn := tempFanIn + 1
        ]
].
```

"there are no cortical inputs to a lower division, so we deal with an upper division, and it has two inputs from every connecting column: from the lower division and from the upper division, so if the column fired we counted twice"

```
^(tempFanIn / 2).
```

setFanOut

```
| positiveFanOut negativeFanOut |

positiveFanOut := 0.
negativeFanOut := 0.
(self firedBefore) ifTrue: [
        outputs do: [:connection |
                (connection receiver column fired)
                ifTrue: [positiveFanOut := positiveFanOut + 1].

                (connection receiver column wasInhibited)
                ifTrue: [negativeFanOut := negativeFanOut + 1].
        ].
        numberOfPositivelyInfluenced := numberOfPositivelyInfluenced
                                                        + positiveFanOut.
        numberOfNegativelyInfluenced := numberOfNegativelyInfluenced
                                                        + negativeFanOut.
]..
```

Object subclass: **#Connection**
instanceVariableNames: 'prevActionPotential actionPotential nextActionPotential transmitter receiver numberOfInfluences weight '
classVariableNames: 'LearningRate '
poolDictionaries: ''
category: 'Neurosolvers'.

Connection comment:
Connection is a class representing intra- and inter-conlumnar connections as well as connections from and to other parts of the system (thamic and manipulation systems). The connection strength represents the probability that the receiver will fire if the transmitter fire.

CONNECTION METHODSFOR: 'access'

strength

```
"return the strength of the connection"

(self transmitter column neurosolver learningMode = #hebbian)
    ifTrue: [^(self strengthHebb)]
    ifFalse: [^(self strengthProbabilistic)].
```

strengthHebb

```
"return the strength of the connection"

^weight.
```

CONNECTION METHODSFOR: 'modification'.

decreaseStrength

```
(self transmitter column neurosolver learningMode == #hebbian)
    ifTrue: [self decreaseStrengthHebb]
    ifFalse: [self decreaseStrengthProbabilistic].
```

decreaseStrengthHebb

"decrease the strength of the connection regardless whether it is positive or negative"

```
weight < 0 ifTrue:
        [weight := weight + 0.1].
weight > 0 ifTrue:
        [weight := weight - 0.1].
```

decreaseStrengthProbabilistic

"decrease the strength of the connection regardless whether it is positive or negative"

```
numberOfInfluences < 0 ifTrue:
        [numberOfInfluences := numberOfInfluences + 1].
numberOfInfluences > 0 ifTrue:
        [numberOfInfluences := numberOfInfluences - 1].
```

increaseNegativeStrength

```
(self transmitter column neurosolver learningMode == #hebbian)
        ifTrue: [self increaseNegativeStrengthHebb]
        ifFalse: [self increaseNegativeStrengthProbabilistic].
```

increaseNegativeStrengthHebb

"increase the inhibitory strength of the connection"

```
weight := weight - (0.1 / (self negativeInfluenceFactor)).
```

increaseNegativeStrengthProbabilistic

"increase the inhibitory strength of the connection"

```
numberOfInfluences := numberOfInfluences - 1.
```

increasePositiveStrength

```
(self transmitter column neurosolver learningMode == #hebbian)
        ifTrue: [self increasePositiveStrengthHebb]
        ifFalse: [self increasePositiveStrengthProbabilistic].
```

increasePositiveStrengthHebb

"increase the excitatory strength of the connection"

```
weight := weight + (((1 - weight) * LearningRate) / (self positiveInfluenceFactor)).
```

increasePositiveStrengthProbabilistic

"increase the excitatory strength of the connection"

```
numberOfInfluences := numberOfInfluences + 1.
```

```
Connection subclass: #LowerUpper
    instanceVariableNames: ''
    classVariableNames: ''
    poolDictionaries: ''
    category: 'Neurosolvers'.


UPPERUPPER METHODSFOR: 'hebbian influence'.

negativeInfluenceFactor

    | factor |

    (factor := receiver numberOfNegativelyInfluenced) > 0 ifTrue: [^factor] ifFalse:
[^1].

positiveInfluenceFactor

    | factor |

    (factor := receiver numberOfPositivelyInfluenced) > 0 ifTrue: [^factor] ifFalse:
[^1].

UPPERUPPER METHODSFOR: 'access'

strengthCausal
    "return the strength of the connection"

    "excitatory connection"
    (numberOfInfluences > 0) ifTrue: [
        transmitter activityUpCount = 0
            ifTrue: [^0.0]
            ifFalse: [^1.0].
    ].

    "inhibitory connection"
```

```
	(numberOfInfluences < 0) ifTrue: [
		transmitter activityDownCount = 0
			ifTrue: [^0.0]
			ifFalse: [^-1.0].
	].

	^0.0.
```

strengthProbabilistic

```
	"return the strength of the connection"

	^self strengthCausal.

	"can also be: ^self strengthProbabilisticSimple"

	"can also be: ^strength probabilisticComplex"
```

strengthProbabilisticComplex

```
"return the strength of the connection"
"NOTE thatalways:
	 numberOfInfluences <= transmitter activityUpCount
	and numberOfInfluences <= receiver numberOfPositivelyInfluenced"

	| ups downs probability1 probability2 influenced |

	"no influence"
	(numberOfInfluences = 0)
	ifTrue: [^0.0]
	ifFalse: [
		(numberOfInfluences > 0)
		ifTrue: [
	"excitatory connection"
			probability1 := (ups := transmitter activityUpCount) = 0
				ifTrue: [0.0]
				ifFalse: [(numberOfInfluences / ups)].
			probability2 := (influenced :=
```

```
                        receiver numberOfPositivelyInfluenced) = 0
                    ifTrue: [0.0]
                    ifFalse: [(numberOfInfluences / influenced)
                                * (receiver activityUpCount / influenced)].
        ]
        ifFalse: [
"inhibitory connection"
                probability1 := (downs := transmitter activityDownCount) = 0
                    ifTrue: [0.0]
                    ifFalse: [(numberOfInfluences / downs)].
                probability2 := (influenced :=
                        receiver numberOfPositivelyInfluenced) = 0
                    ifTrue: [0.0]
                    ifFalse: [(numberOfInfluences / influenced)
                                * (receiver activityUpCount / influenced)].
        ].

        ^(probability1 * probability2) negated
    ].
```

strengthProbabilisticSimple

```
    "return the strength of the connection"

    | ups downs |

    "no influence"
    (numberOfInfluences = 0)
    ifTrue: [^0.0].

    "excitatory connection"
    (numberOfInfluences > 0) ifTrue: [
                (ups := transmitter activityUpCount) = 0
                    ifTrue: [^0.0]
                    ifFalse: [^(numberOfInfluences / ups)].
    ].
```

```
"inhibitory connection"
(numberOfInfluences < 0) ifTrue: [
        (downs := transmitter activityDownCount) = 0
                ifTrue: [^0.0]
                ifFalse: [^(numberOfInfluences / downs) negated].
].
```

```
Connection subclass: #LowerUpper
	instanceVariableNames: ''
	classVariableNames: ''
	poolDictionaries: ''
	category: 'Neurosolvers'.


LowerUpper methodsFor: 'hebbian influence'

negativeInfluenceFactor

	| factor |

	(factor := receiver numberOfNegativeInfluences) > 0 ifTrue: [^factor] ifFalse:
[^1].

positiveInfluenceFactor

	| factor |

	(factor := receiver numberOfPositiveInfluences) > 0 ifTrue: [^factor] ifFalse: [^1].

LowerUpper methodsFor: 'access'

strengthProbabilistic
"return the strength of the connection"
"NOTE thatalways:
	numberOfInfluences <= transmitter activityUpCount
	and numberOfInfluences <= receiver numberOfPositiveInfluences"

	| ups downs probability1 probability2 influences |

	"no influence"
	(numberOfInfluences = 0)
	ifTrue: [^0.0]
```

```
"excitatory connection"
ifFalse: [
        (numberOfInfluences > 0)
        ifTrue: [
                probability1 := (ups := transmitter activityUpCount) = 0
                                                ifTrue: [0.0]
                                                ifFalse: [numberOfInfluences / ups].

                probability2 := (influences :=
                                        receiver numberOfPositiveInfluences) = 0
                                                ifTrue: [0.0]
                                                ifFalse: [(numberOfInfluences
                / influences) * (receiver activityUpCount / influences)].
        ]
"inhibitory connection"
        ifFalse: [
                probability1 := (downs := transmitter activityDownCount) = 0
                                                ifTrue: [0.0]
                                                ifFalse: [numberOfInfluences
                                                                / downs].

                probability2 := (influences :=
                                        receiver numberOfNegativeInfluences) = 0
                                                ifTrue: [0.0]
                                                ifFalse: [(numberOfInfluences
                        / influences) * (receiver activityUpCount / influences)].
        ].

        ^(probability1 * probability2)
].
```

END

17-12-93

FIN