WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF MATHEMATICS
AND INFORMATION SCIENCE

Neural Networks

Lecture 2

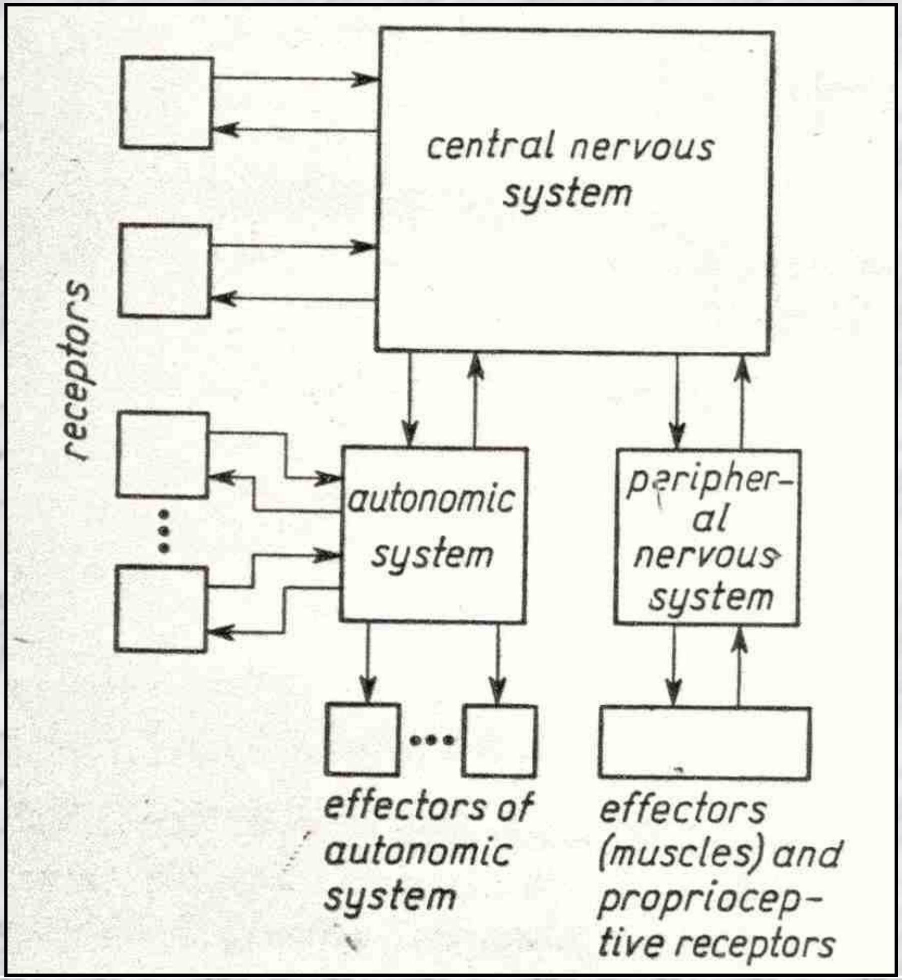
Principles to which the nervous system works

Some biology and neurophysiology

Nervous system

- **central nervous system**
- **peripheral nervous system**
- **autonomic nervous system**

Diagram of the nervous system



Some biology and neurophysiology

Central nervous system has three hierarchical levels:

- the spinal cord level,
- the lower brain level,
- the cortical level.

The **spinal cord** acts as the organ controlling the simplest reaction of the organism (spinal reflexes)

Some biology and neurophysiology

Lower region of the **brain** and regions in the **cerebellum** are coordinating the motor activities, orientation in space, general regulation of body (temperature, blood pressure etc.)

Cerebral cortex establish interrelations between lower regions and coordinating their functions. Decision are taking, information is stored in cerebral cortex,

Some biology and neurophysiology

Peripheral nervous system composed of the nerve processes running out from the brain and spinal cord.

Nerves are the connections for communication between centers and organs.

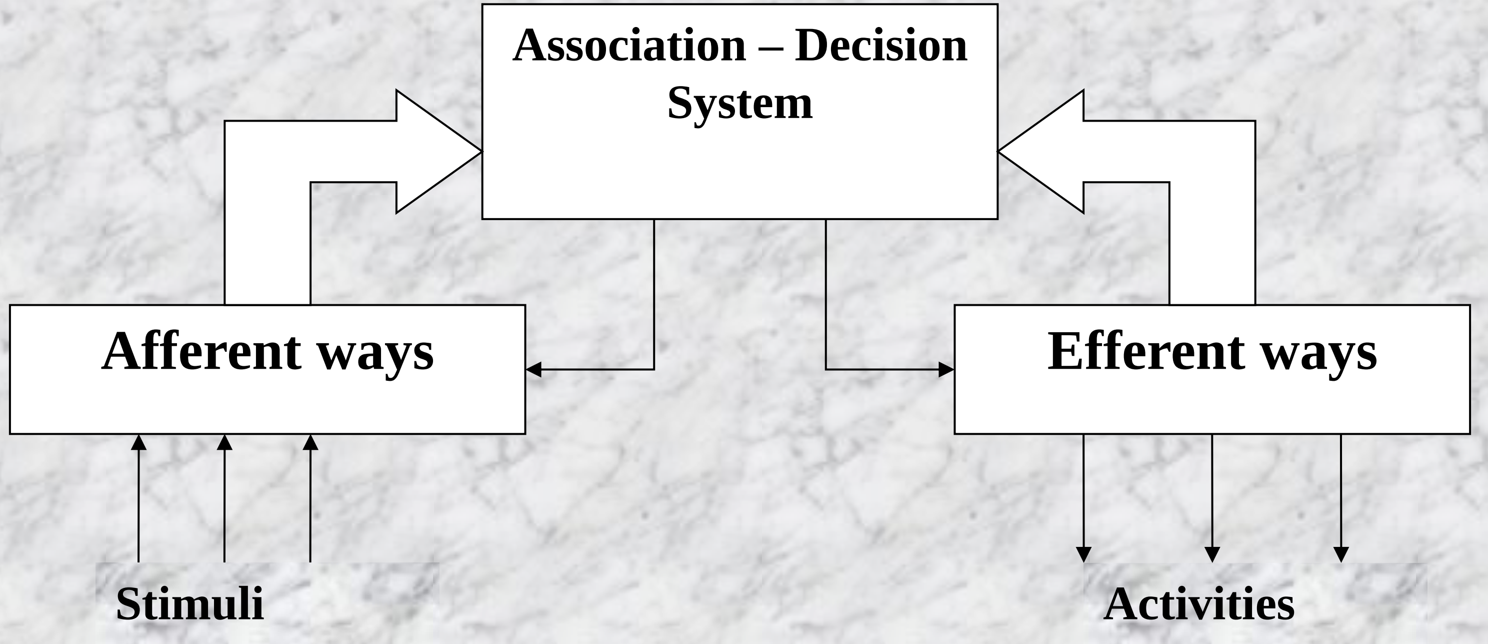
Some biology and neurophysiology

The task of the Autonomous nervous system is to control the most important vital processes such as breathing, blood circulation, concentration of chemicals in the blood etc.

Functional scheme of connections of the nervous system:

1. an afferent system
2. a central association decision making system
3. an efferent system

Some biology and neurophysiology



Some biology and neurophysiology

Afferent ways

an afferent system in which signals arriving from the environment are transmitted and analyzed, the degree and mode of analysis is controlled by superior coordinating and decision making system,

multi level and hierarchical structures supplying the brain with information about external world (environment).

Some biology and neurophysiology

The efferent system

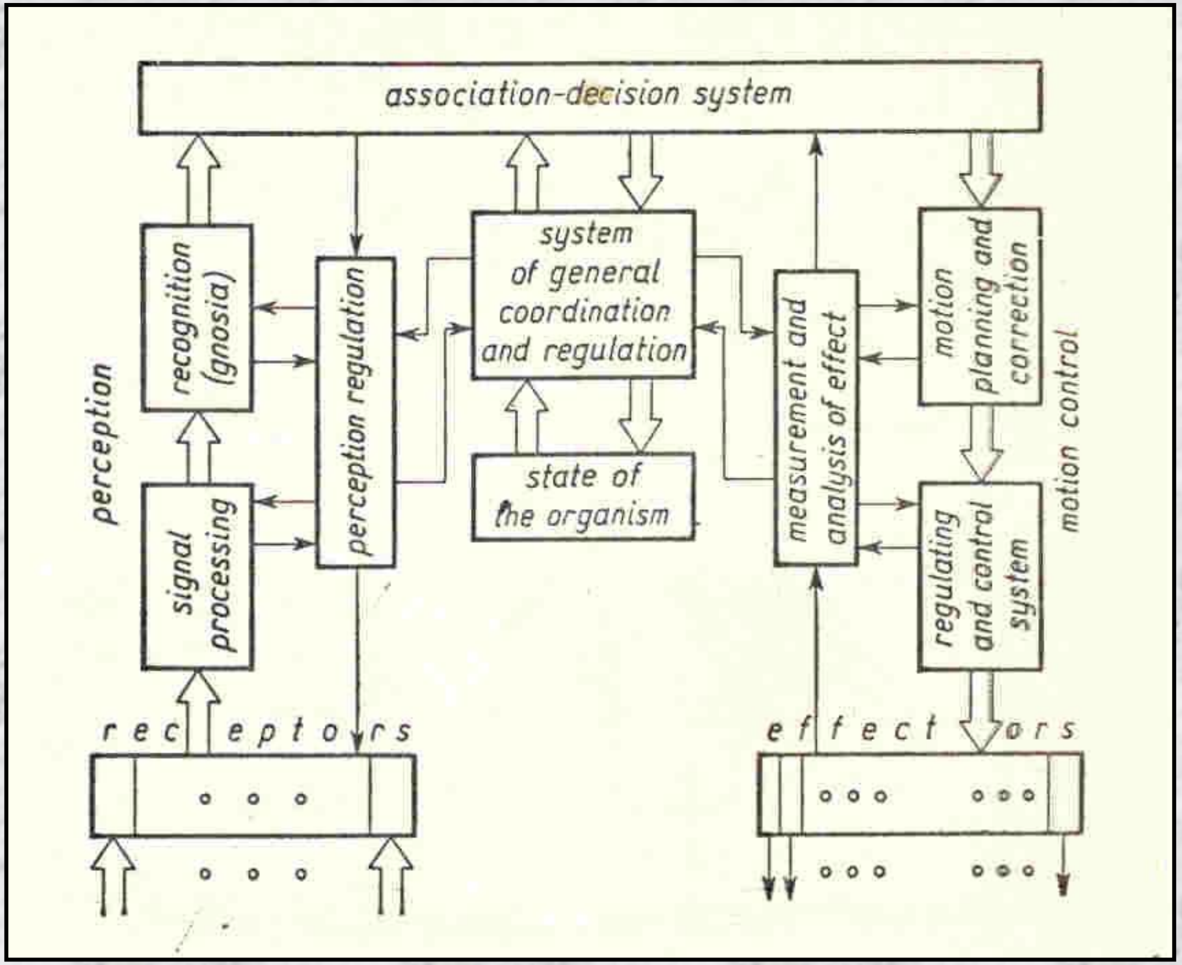
in which, on the basis of the decision taken a plan of reaction of the organism is worked out, on the base of static and dynamic situation, experience and optimization rules, output channels of a nervous system responsible for transmission and processing of signals controlling the effectors

Some biology and neurophysiology

The central association and decision making system

where a decision about the reaction of the organism is worked out on the basis of the state of the environment, the state of the organism, previous experience, and a prediction of effect

Some biology and neurophysiology



Nerve cell models

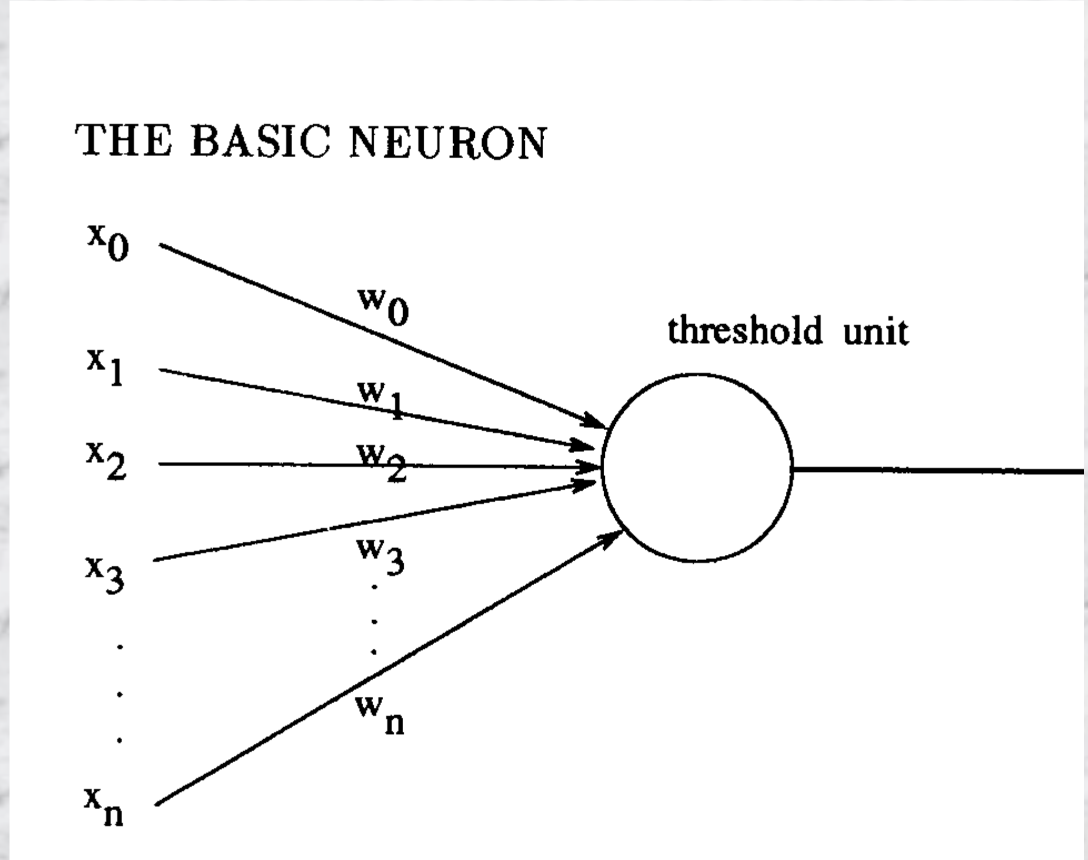
The first model of neuron was proposed in 1943 by W.S. McCulloch and W.Pitts

The model came from the research into behavior of the neurons in the brain. It was very simple unit, thresholding the weighted sum of its inputs to get an output.

It was the result of the actual state of knowledge and used the methods of mathematical and formal logic.

The element was also called *a formal neuron*.

Nerve cell models



Nerve cell models

The formal neuron was characterized by describing its state (or output).

Changing of the state from inactive (0) to active (1) was when the weighted sum of input signals was greater than the threshold; and there was no inhibitory input.

Nerve cell models

Model assumptions:

1. The element activity is based on the „all-or-none” principle.
2. The excitation (state 1) is preceded by a constant delay while accumulating the signals incoming to synapses (independent from the previous activity and localization of synapses).
3. The only neuronal delay between the input simulation and activity at the output, is the synaptic delay.

Nerve cell models

Model assumptions:

3. **Stimulation of any inhibitory input excludes a response at the output at the moment under consideration.**
4. **The net structure and neuron properties do not change with time.**

Nerve cell models

The discrete time is logical, because in the real neuron, after the action potential, the membrane is non-excitable, i.e. another impulse cannot be generated (appr. 1 ms). This interval is called the ***absolute refractory period***.

It specifies the maximum impulse repetition rate to about 1000 impulses per second.

Mathematical models of a nerve cell

The methods of selection of the properties of neural element depends not only on previous results, our level of knowledge – but mainly from the phenomena to be modeled.
Another properties will be important while modeling the steady states, another for dynamic processes or for the learning processes.

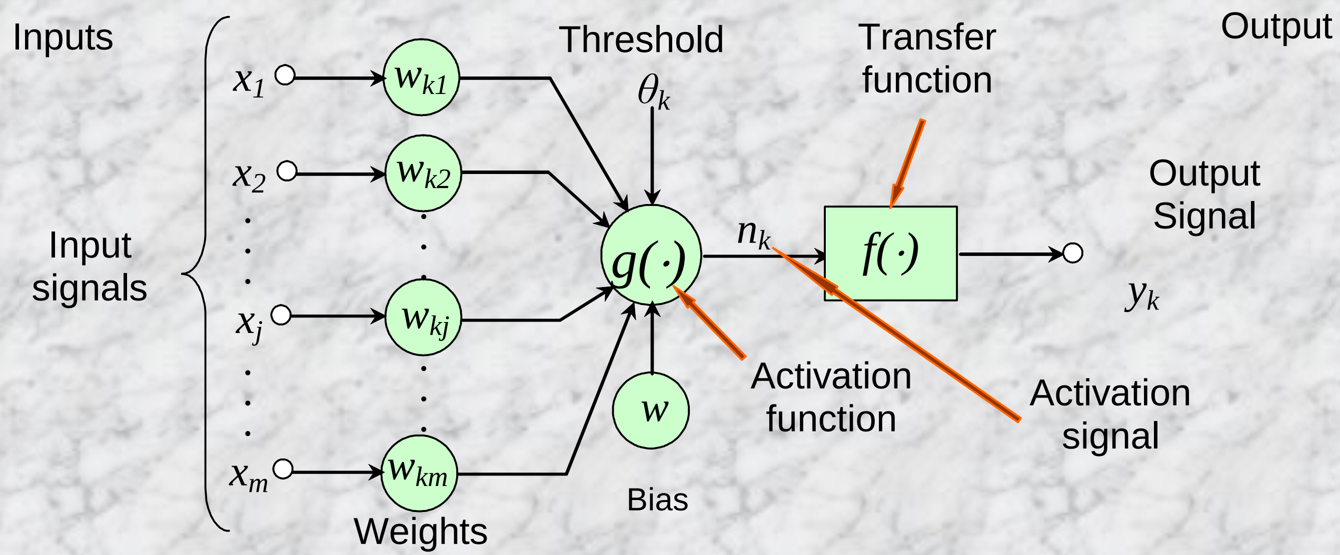
But always, the model has to be as simple as possible.

Neural cell models

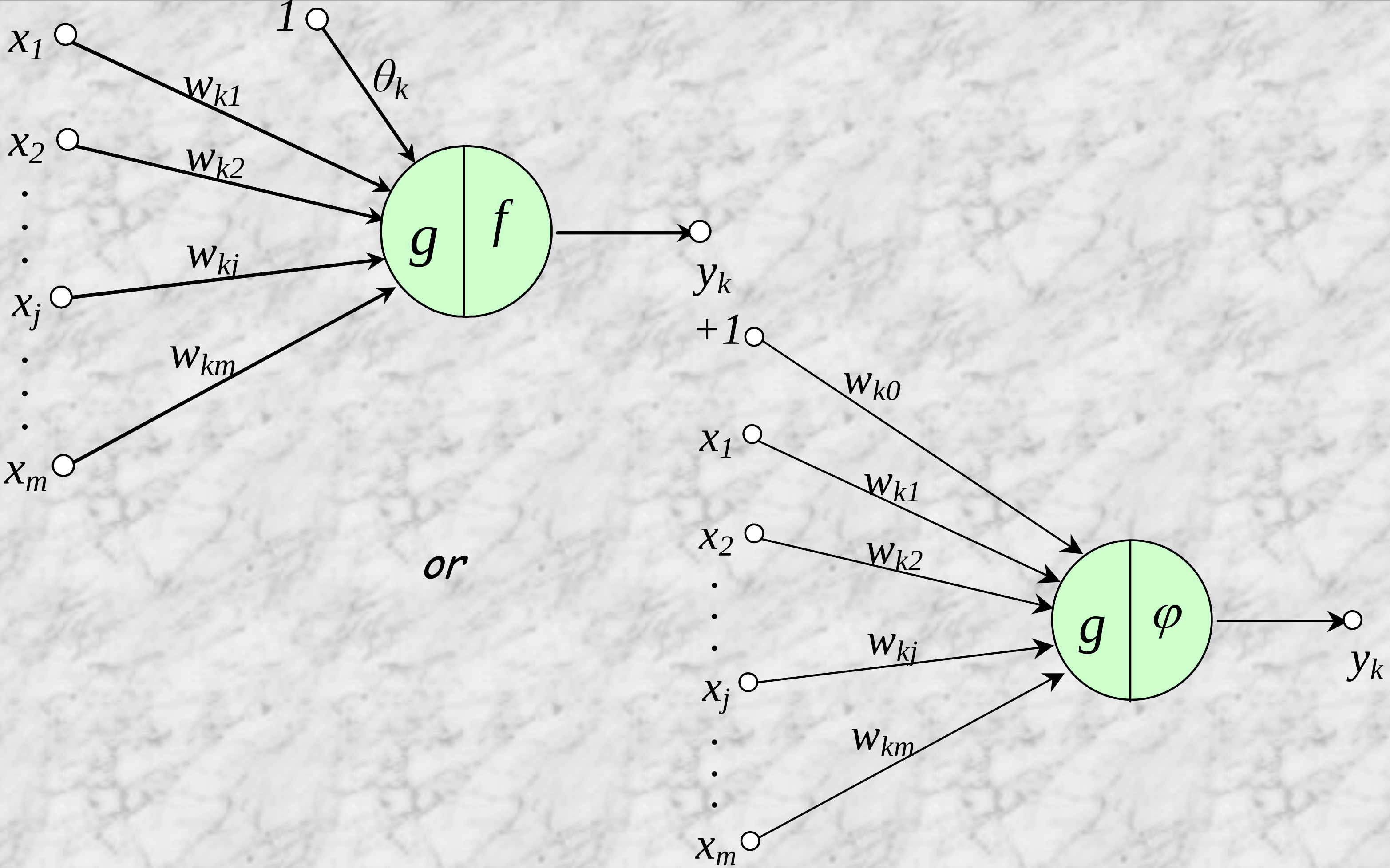
Static model of neural cell

Two kinds of neurons: static and dynamic.
Static model of neuron (used in continuous and discrete static nets and dynamic discrete sets).

Model of a static neuron k

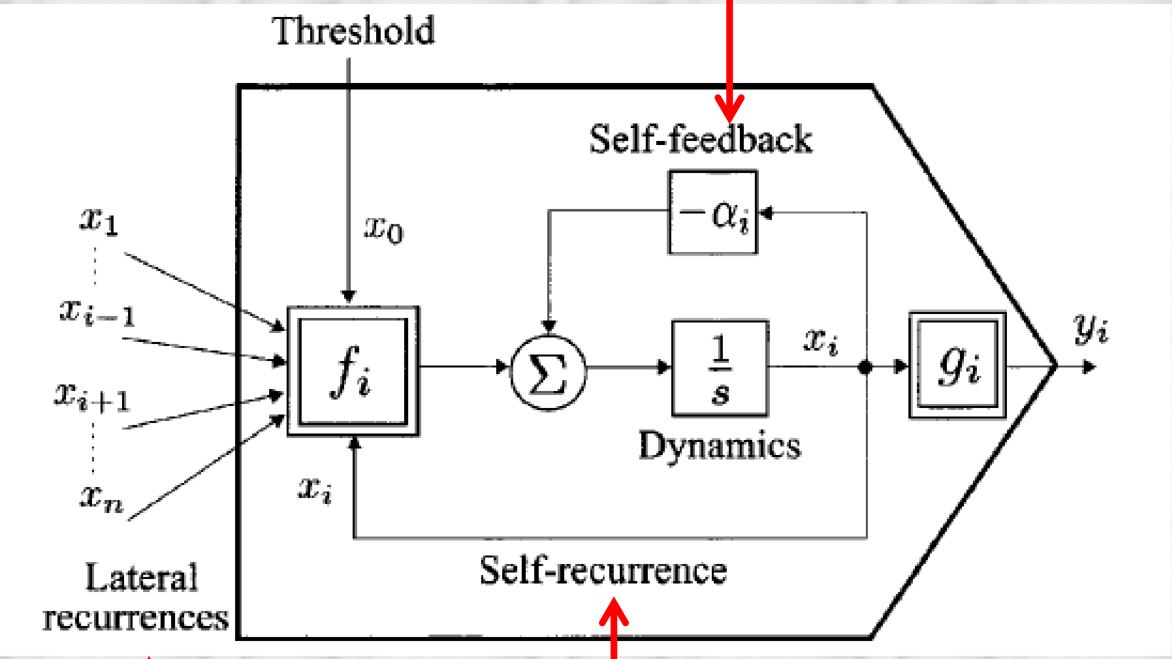


Simplified models of static neuron



Dynamic model of neuron

Internal self-feedback



Self recursion

External recursion

Activation functions

1. *Adder*
2. *Product*
3. *Maximum*
4. *Minimum*
5. *Dominant*
6. *Cumulative sum*

Activation functions

Sum function

$$n_k = \sum_{\mathrm{j}=1}^{m} \mathrm{w}_{k\,\mathrm{j}}\, x_{\mathrm{j}} + \theta_k \quad or \quad n_k = \sum_{\mathrm{j}=0}^{m} w_{kj} x_j$$

Product function

$$n_k = \prod_{j=1}^{m} w_{k\,j} x_j$$

Maximum

$$n_k = \max_{\mathrm{j}} \left\{ w_{k\,\mathrm{j}} x_j \right\}$$

(k – neuron's number, j - number of neuron's input)

Activation functions

Minimum

$$n_k = \min_{j} \{w_{kj} x_j\}$$

Dominant function

$$n_k = \sum_{j=1}^{m} \mu_j \quad \textit{where} \quad \mu_j = \begin{cases} 1 & \text{if} \quad w_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases}$$

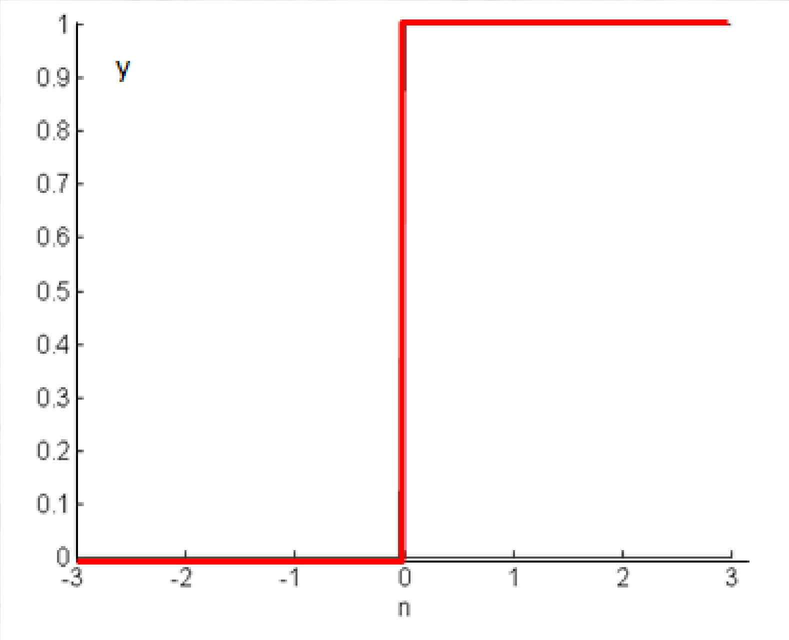
Cumulative sum function

$$n^{(i+1)} = n^{(i)} + \sum_{j=1}^{m} w_{k,j}^{(i)} x_j^{(i)} \quad \textit{where} \quad i \text{ - iteration}$$

(k – neuron's number, j - number of neuron's input)

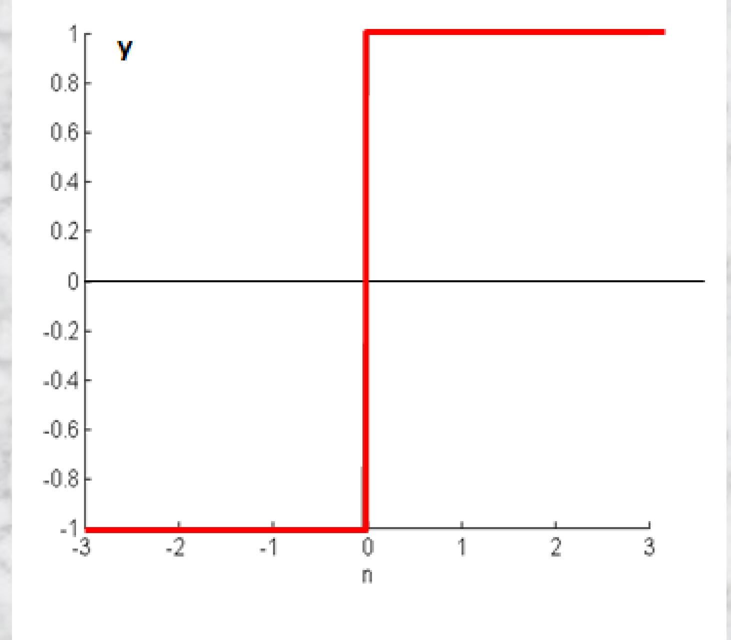
Transfer functions

Unipolar function



$$y = \begin{cases} 0 & if \quad n < 0 \\ 1 & otherwise \end{cases}$$

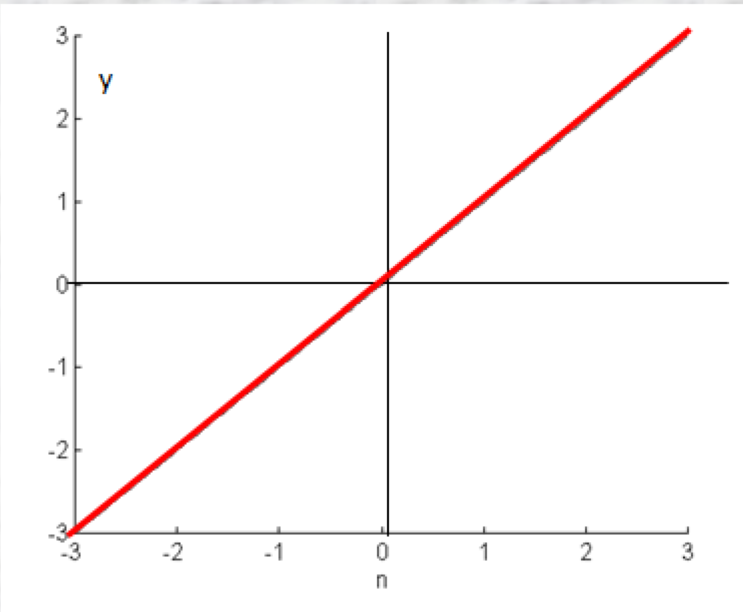
Bipolar function



$$y = \begin{cases} -1 & if \quad n < 0 \\ +1 & otherwise \end{cases}$$

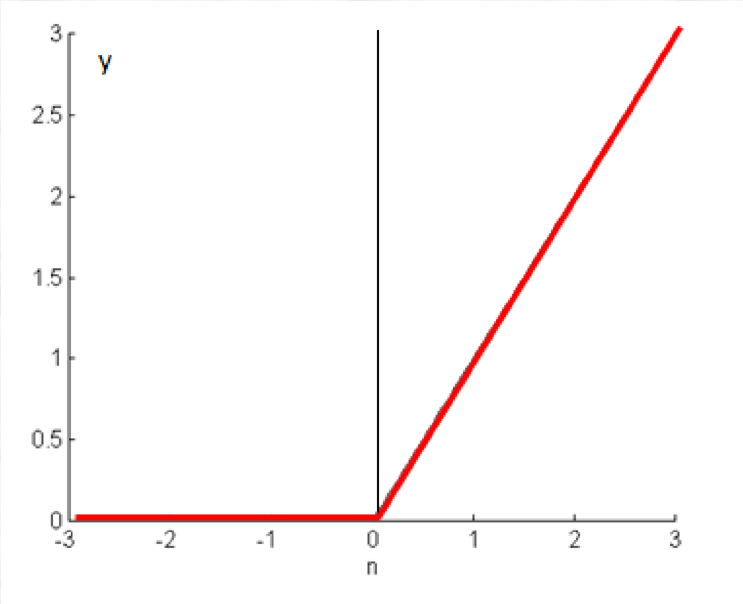
Transfer functions

Linear function



$$y = n$$

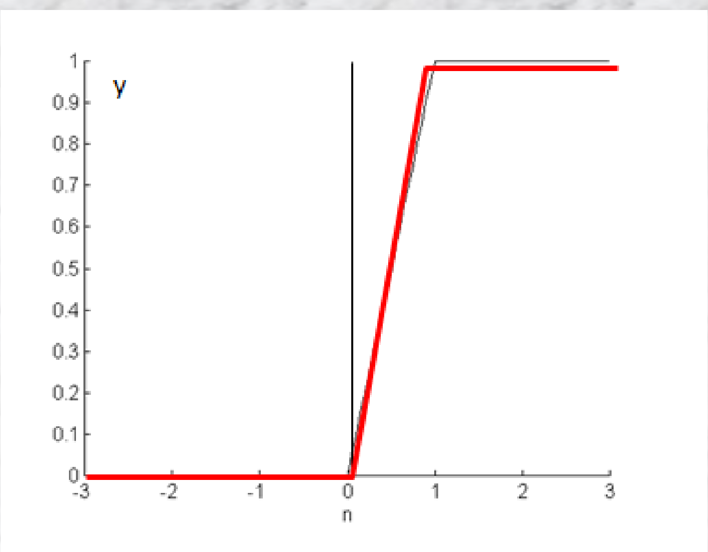
Linear positive function



$$y = \begin{cases} 0 & if \quad \mathrm{n} < 0 \\ n & otherwise \end{cases}$$

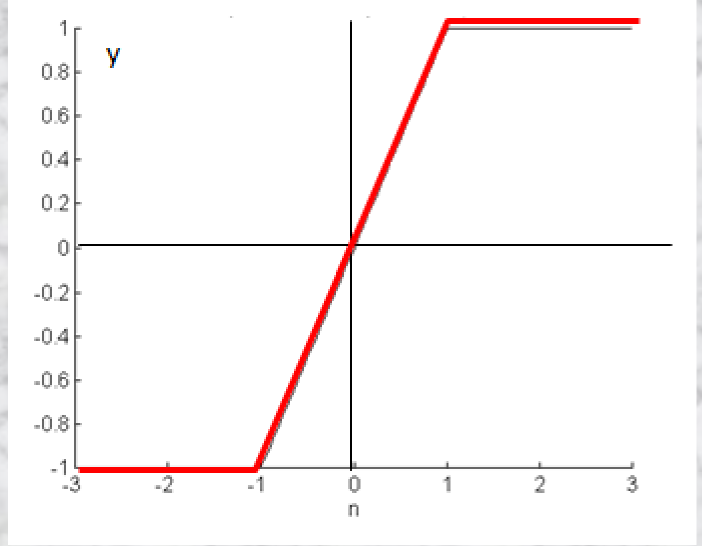
Transfer functions

Linear function with saturation (non symmetric)



$$y = \begin{cases} 0 & if \quad n < 0 \\ n & if \quad 0 \le n \le \alpha \\ 1 & if \quad n > \alpha \end{cases}$$

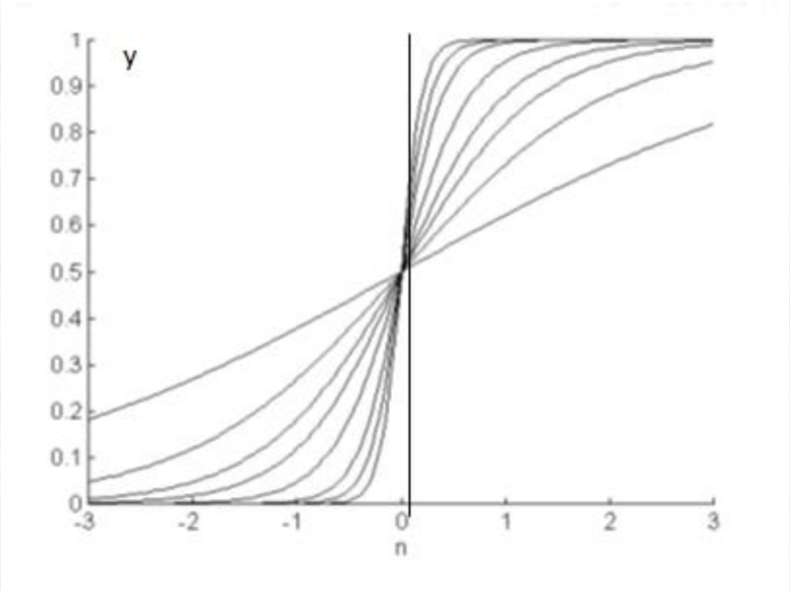
Linear function with saturation (symmetric)



$$y = \begin{cases} -1 & if \quad n < -\alpha \\ n & if \quad -\alpha \le n \le \alpha \\ +1 & if \quad n > \alpha \end{cases}$$

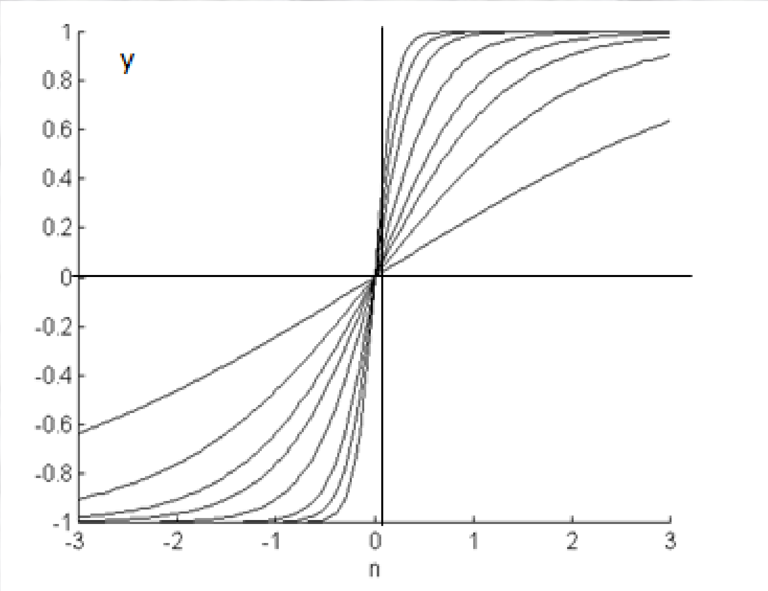
Transfer functions

Unipolar sigmoidal function (log) (non symmetric)



$$y = \frac{1}{1 + \exp(-\lambda n)}$$

Bipolar sigmoidal function (log) (symmetric)

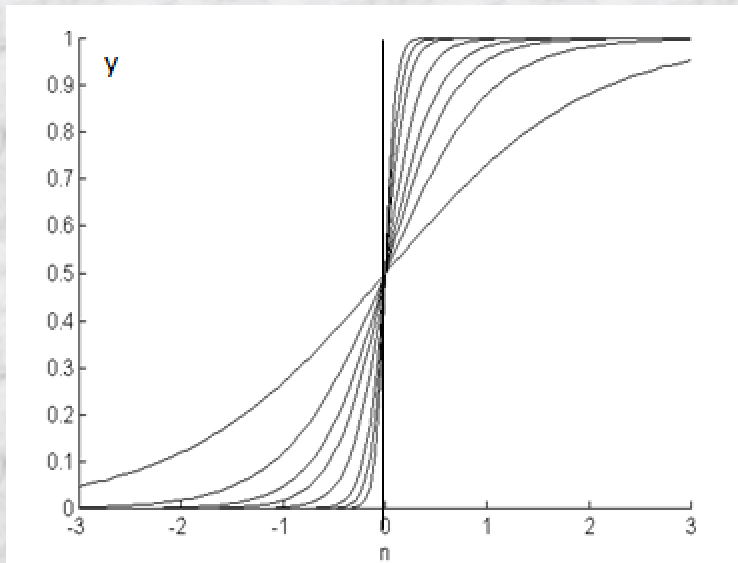


$$y = \frac{2}{1 + \exp(-\lambda n)} - 1$$

$$\lambda \in \{0.5; 1; 1.5; 2; 3; 5; 7; 10\}$$

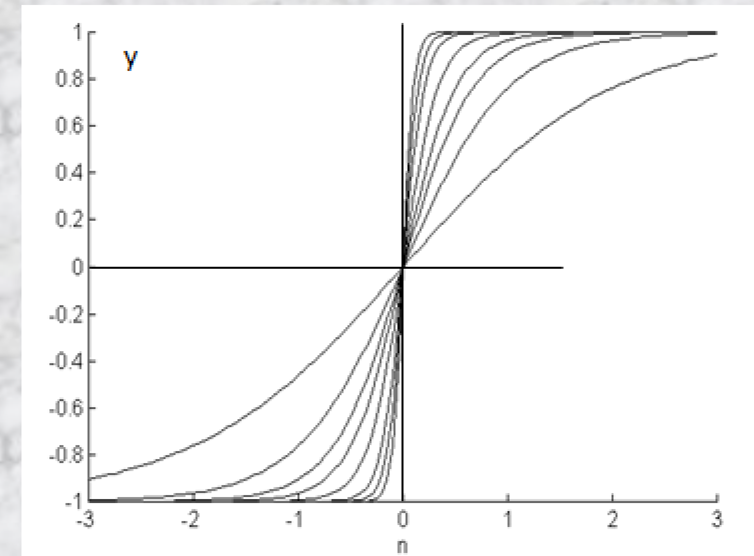
Transfer functions

Unipolar sigmoidal function (th) (non symmetric)



$$y = \frac{1}{2}\tanh(\lambda n) + \frac{1}{2} = \frac{1}{2}\,\frac{\exp(\lambda n) - \exp(\lambda n)}{\exp(\lambda n) + \exp(\lambda n)} + \frac{1}{2}$$

Bipolar sigmoidal function (th) (symmetric)

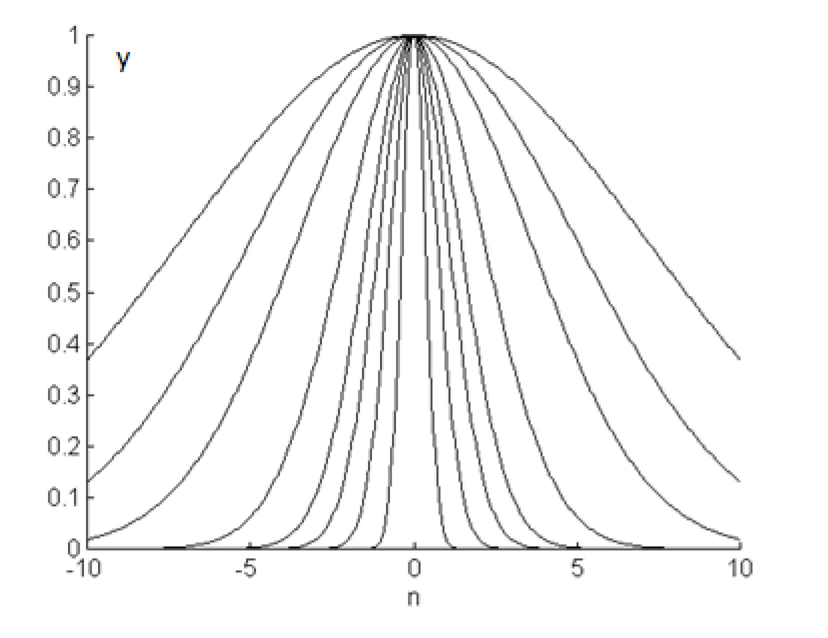


$$y = \tanh(\lambda n) = \frac{\exp(\lambda n) - \exp(\lambda n)}{\exp(\lambda n) + \exp(\lambda n)}$$

$$\lambda \in \{0.5; 1; 1.5; 2; 3; 5; 7; 10\}$$

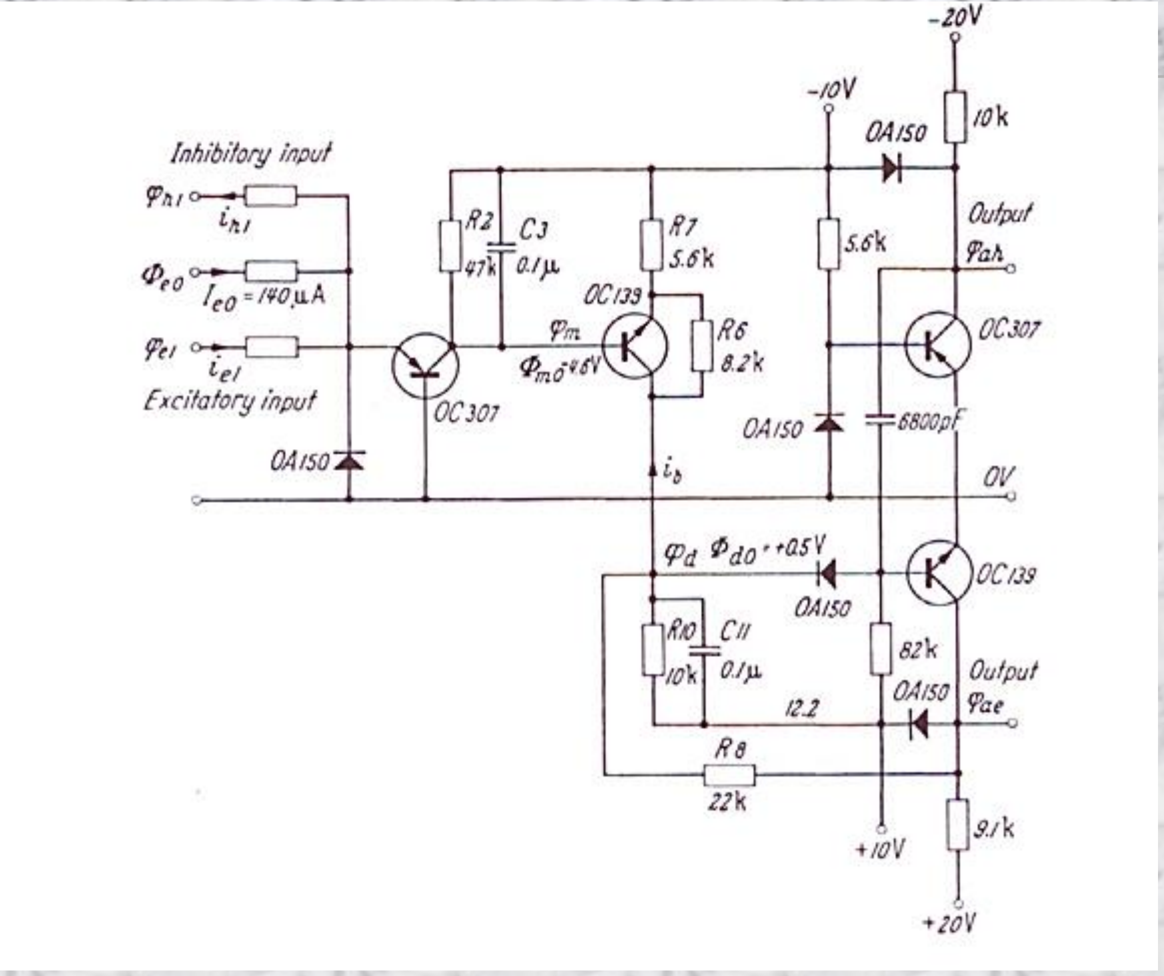
Transfer functions

Gauss function (Radial Basis)



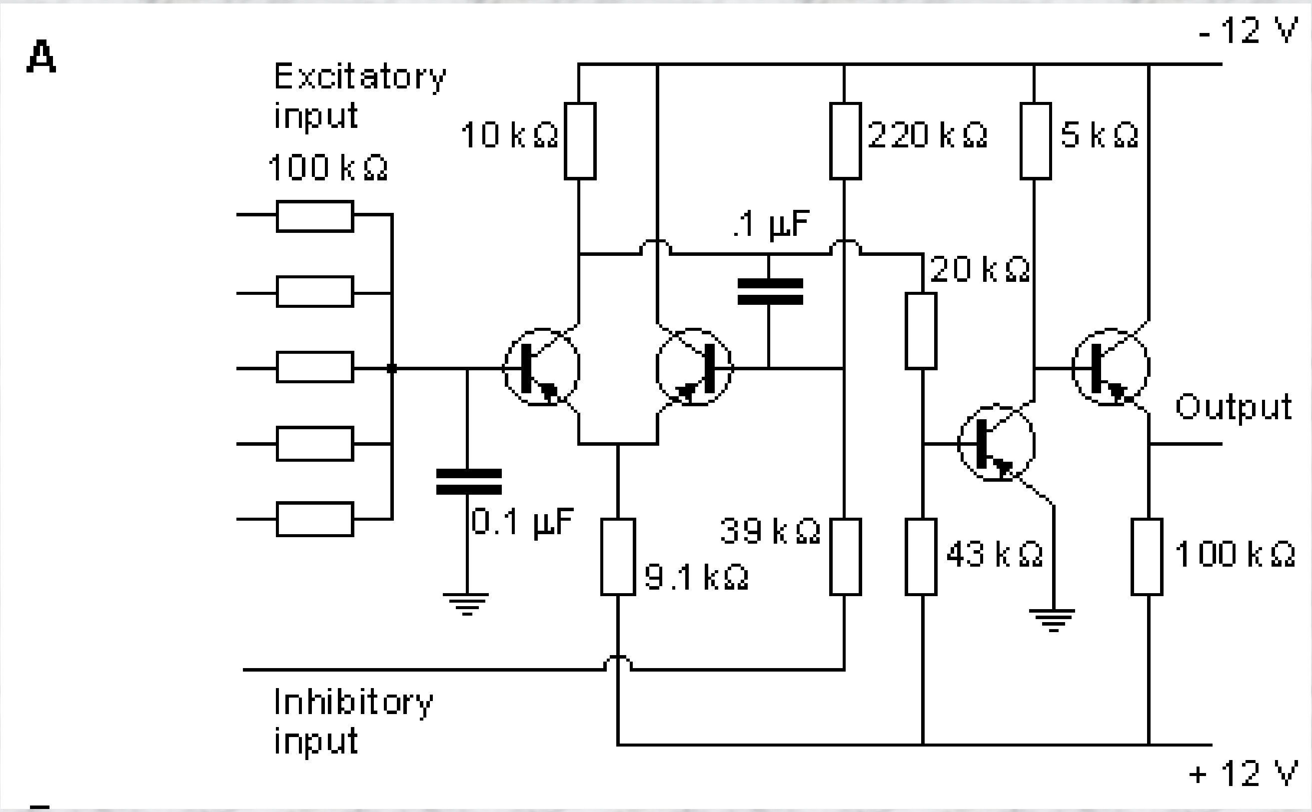
$$y = exp\left(-\frac{n^2}{\sigma^2}\right) \qquad \sigma \in \{0.5; 1; 1.5; 2; 3; 5; 7; 10\}$$

Electronic models



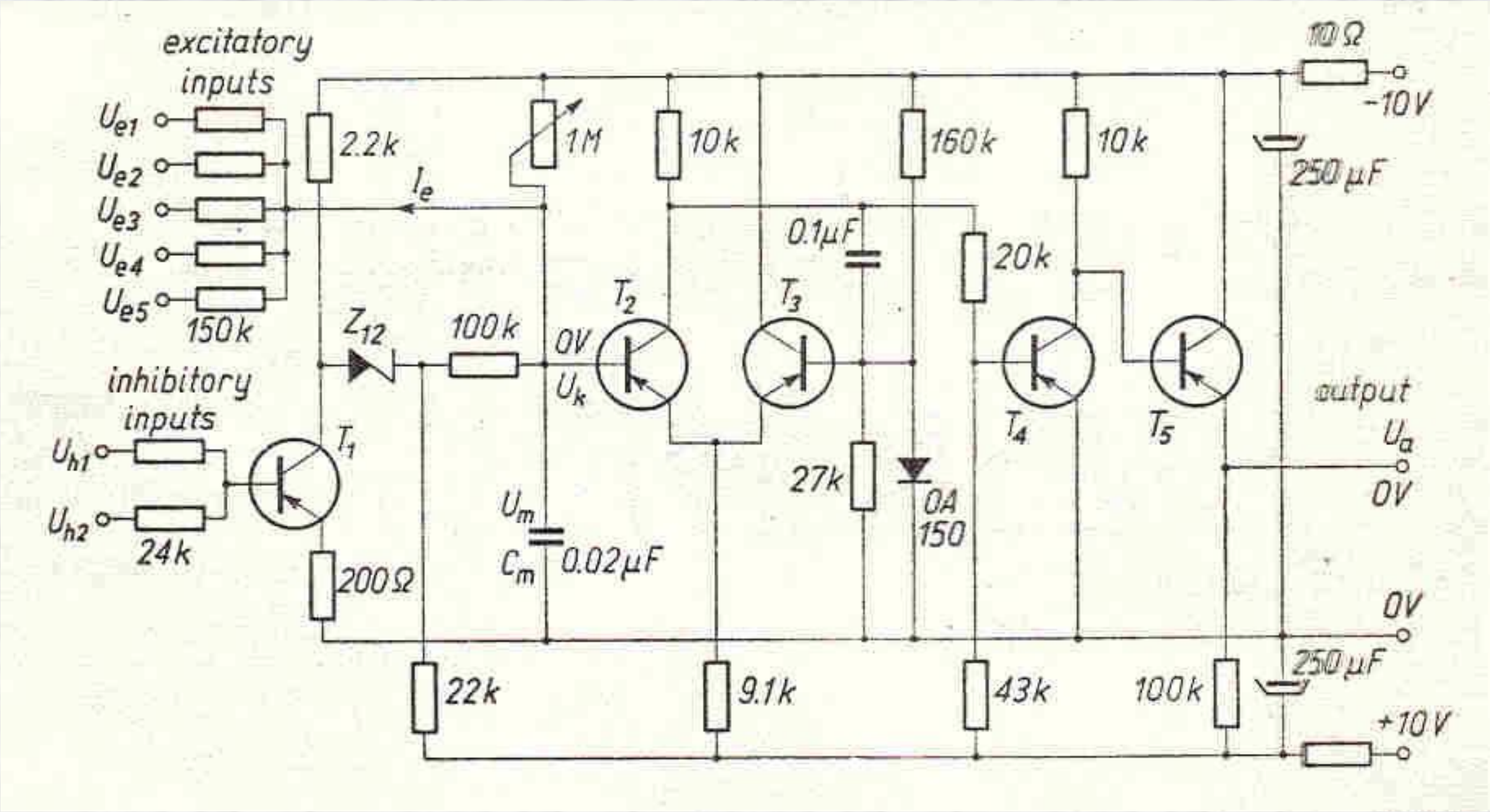
Electronic neural cell model due to McGrogan

Electronic models



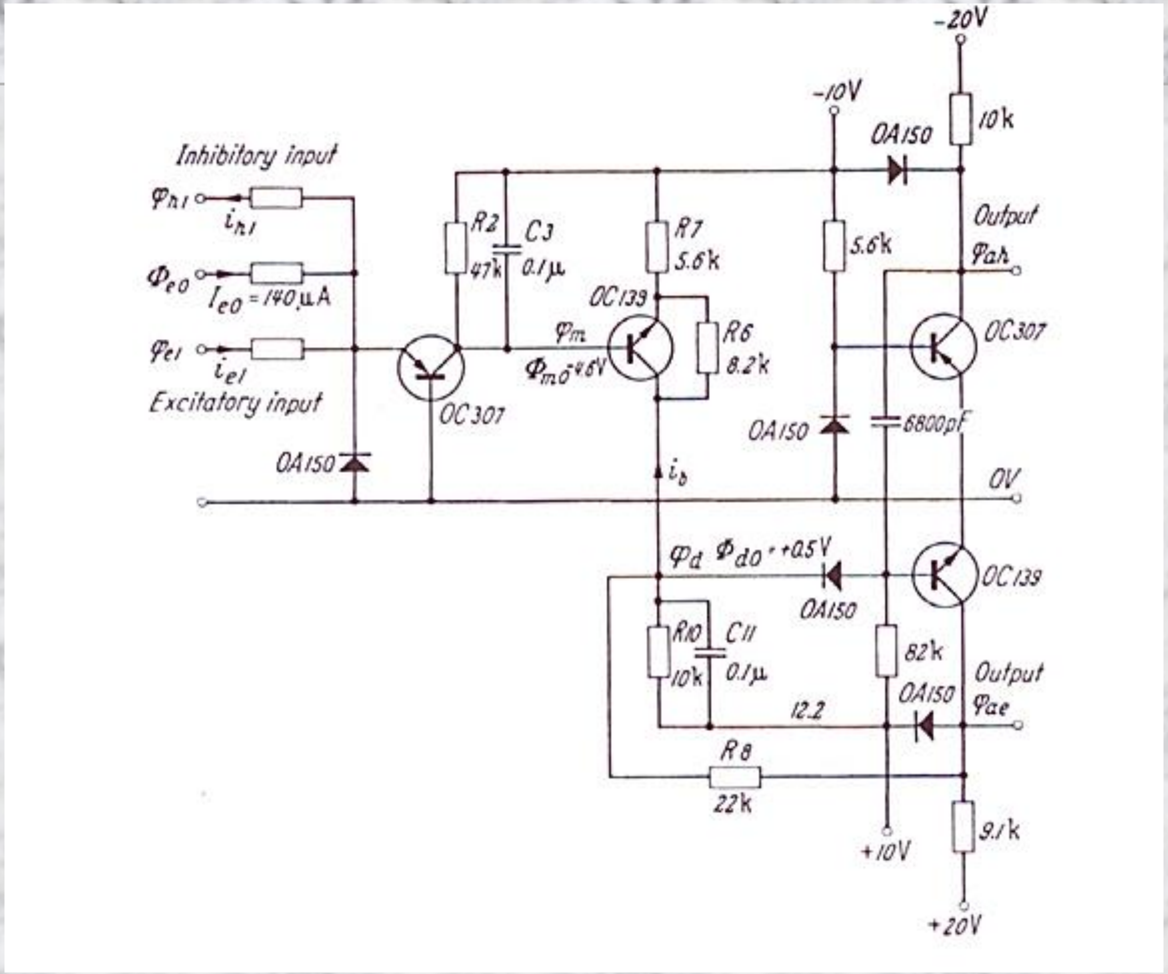
Electronic neural cell model due to Harmon.

Electronic models



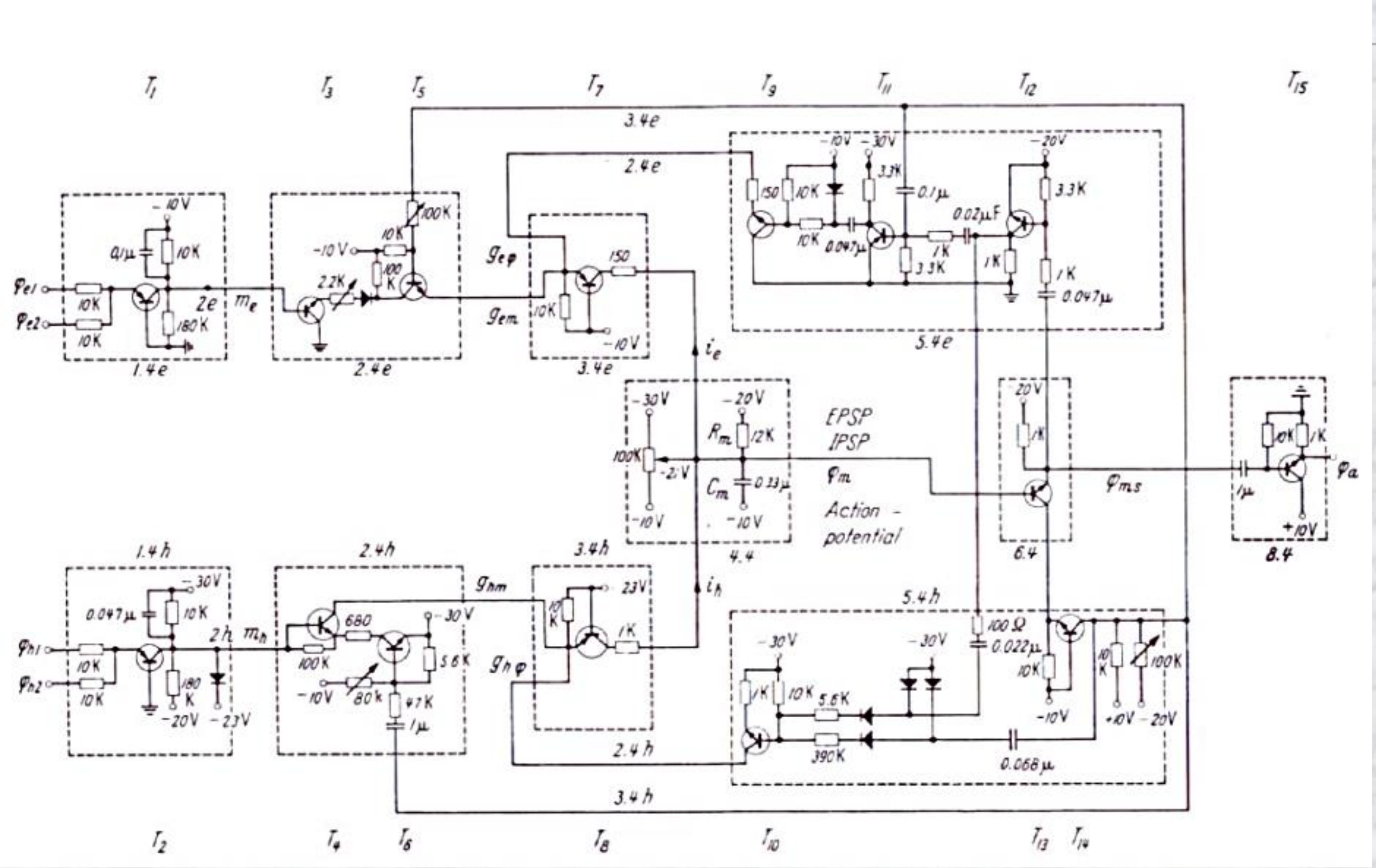
Electronic neural cell model due to Harmon.

Electronic models



Electronic neural cell model due to Taylor.

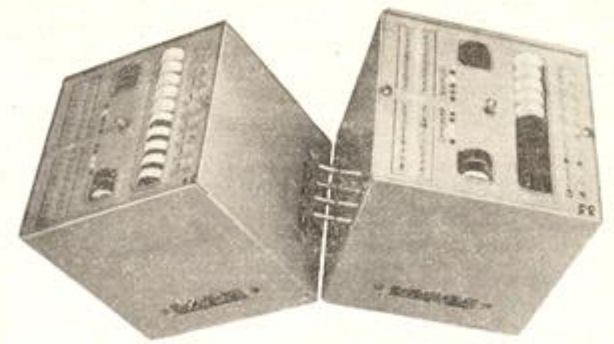
Electronic models



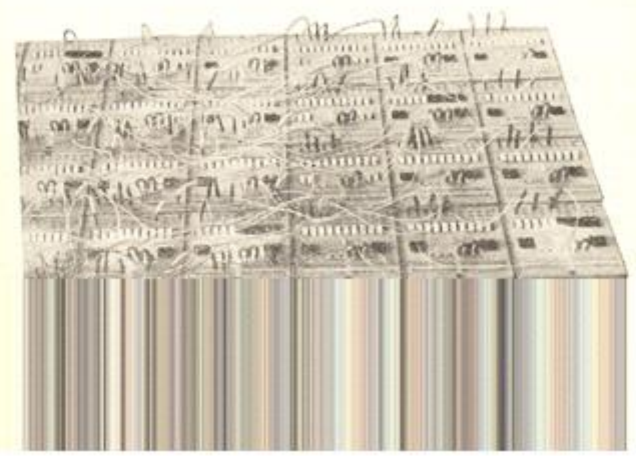
Electronic neural cell model due to Küpfmüller and Janik

Models built in the Bionics Laboratory, PAS

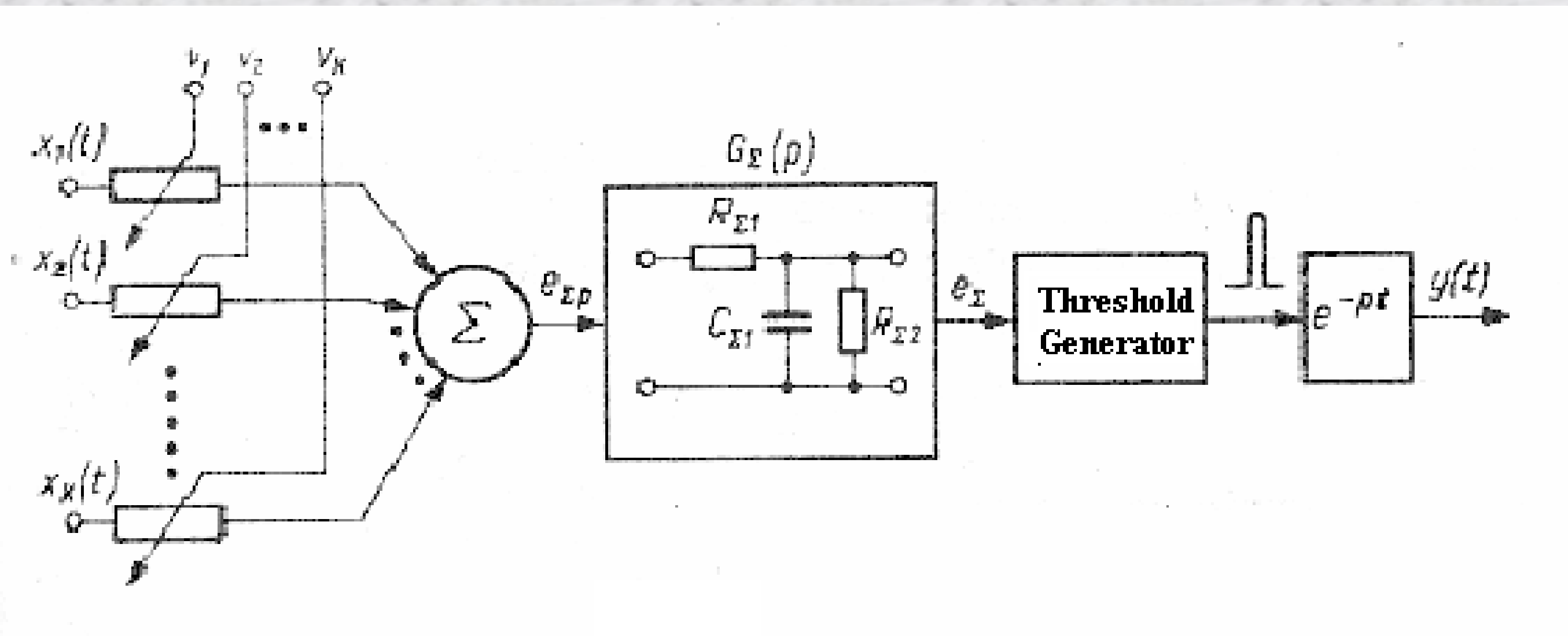
Neuron model built in the Bionics Laboratory IA PAS, in 1969



Neural network model built in the Bionics Laboratory IA PAS, in 1969



Simplified model of a neural cell



McCulloch Model

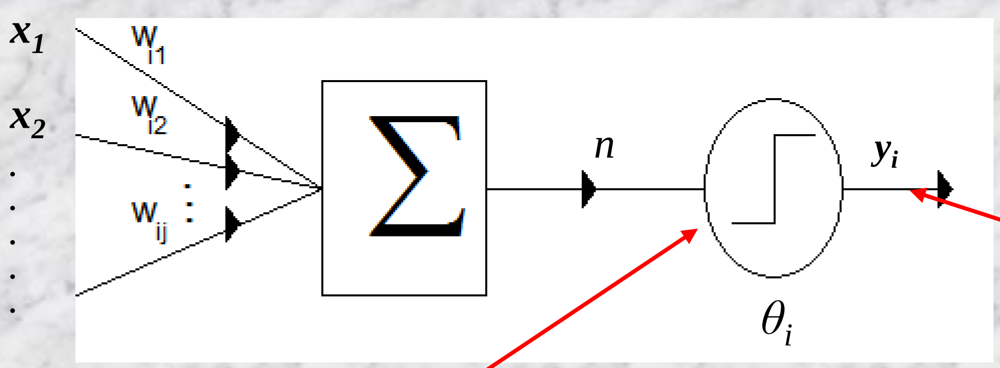
McCulloch-Pitts Model

In 1943 Warren McCulloch and Walter Pitts proposed the first simple mathematics model of a neuron as a two-values threshold element.

The McCulloch-Pitts neuron calculates the weighted sum of input signals incoming from other neurons and produce at the output value 1 (on) or 0 (off) depending the sum is greater or smaller from the threshold value.

McCulloch-Pitts Model

McCulloch and Pitts model of a single neuron



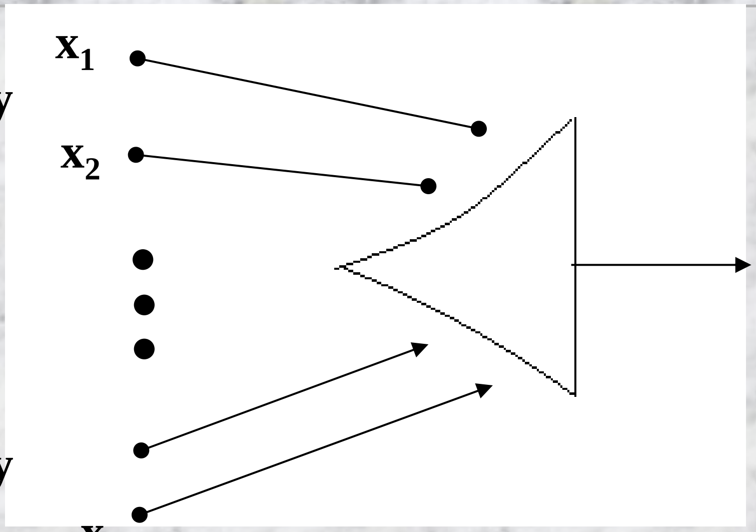
$$y_i(t+1) = f\left(\sum_j w_{ij} x_j(t) - \theta_i\right)$$

Transfer function

$$f(x) = \begin{cases} 1 & if \;\; n \geq 0 \\ 0 & otherwise \end{cases}$$

Neural cell models

McCulloch and Pitts Model



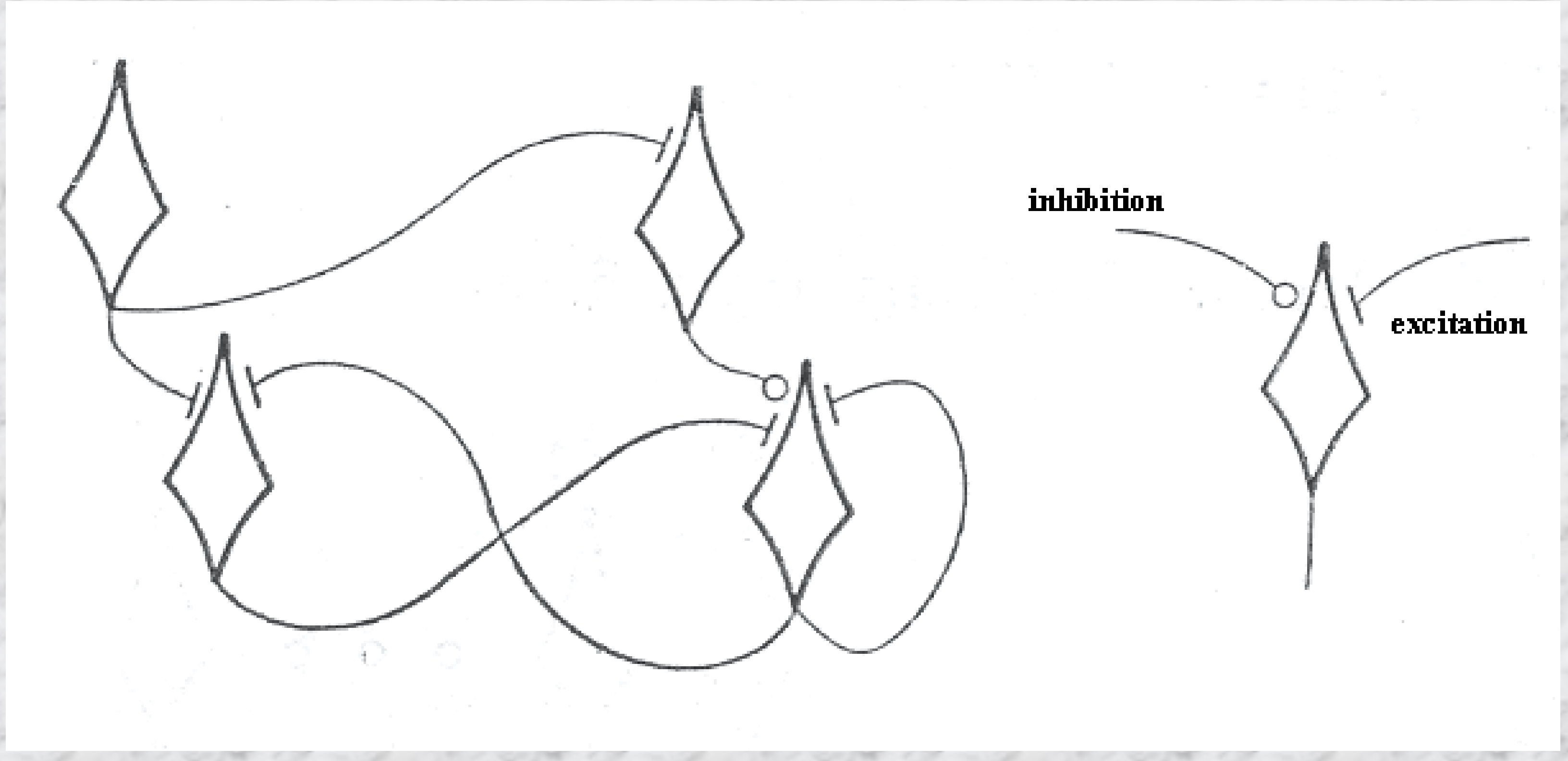
$$y(t+1) = \frac{1}{2}\operatorname{sgn}\left[\sum_{j=1}^{n} v_j x_j(t) - \Theta\right] + \frac{1}{2}$$

or

$$y(t+1) = \begin{cases} 1 \ \textbf{when} \ \sum_{j=1}^{n} v_j x_j(t) \geq \Theta \\ 0 \ \textbf{when} \ \sum_{j=1}^{n} v_j x_j(t) < \Theta \end{cases}$$

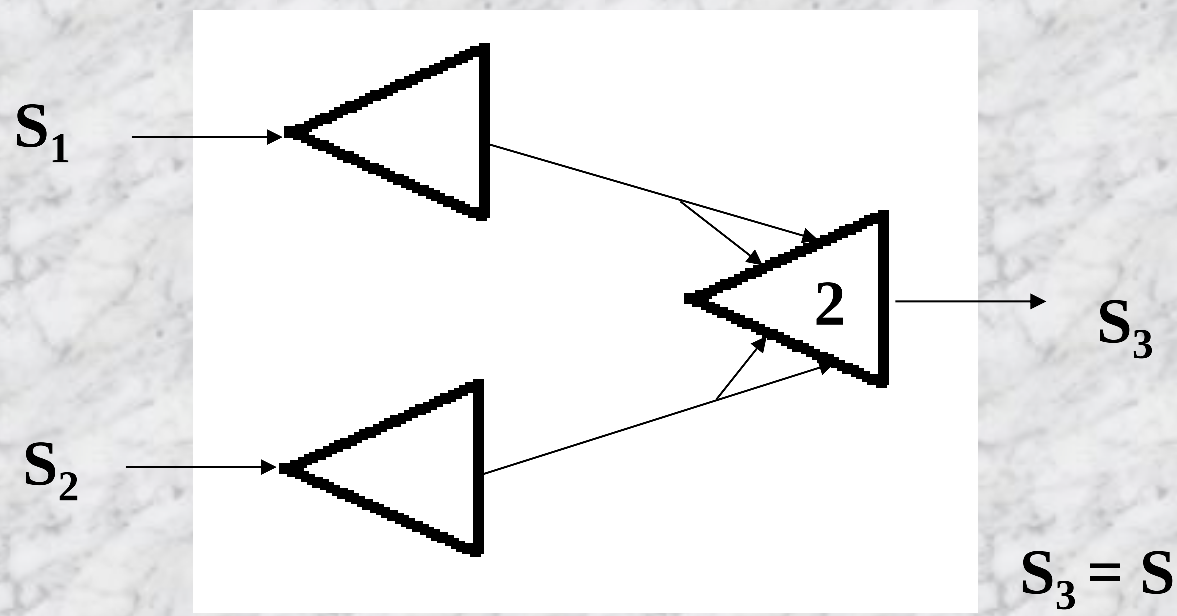
Neural cell models

McCulloch and Pitts models



Simple nets build from McCulloch &Pitts elements

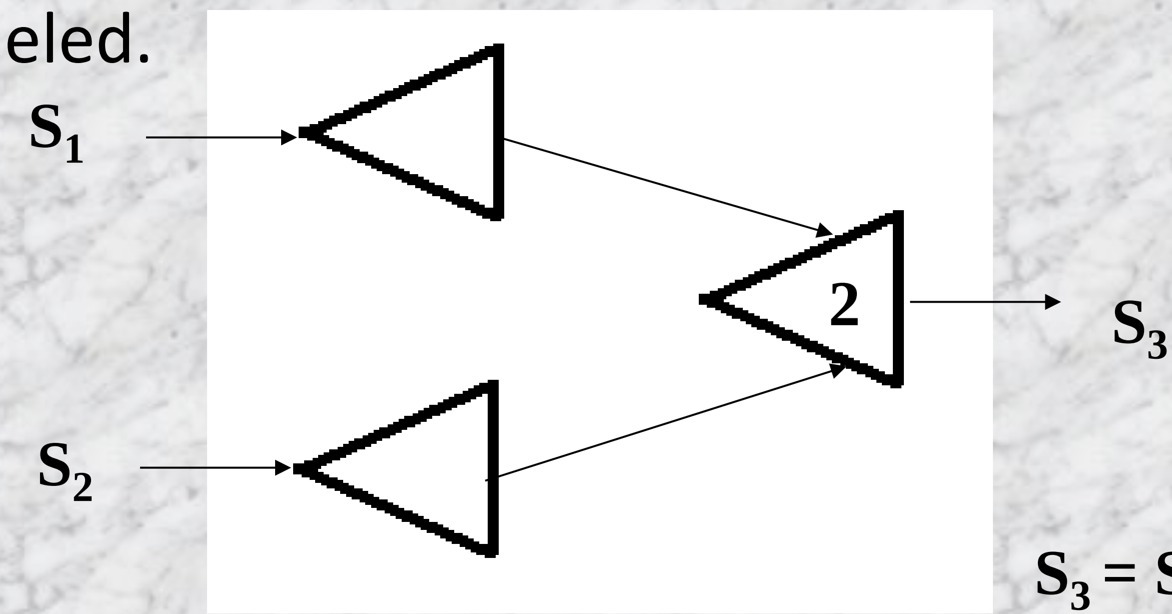
From these simple elements - formal neurons - the nets simulating complex operations or some forms of the behavior of living organisms can be modeled



$S_3 = S_1 \vee S_2$

Simple nets build from McCulloch &Pitts elements

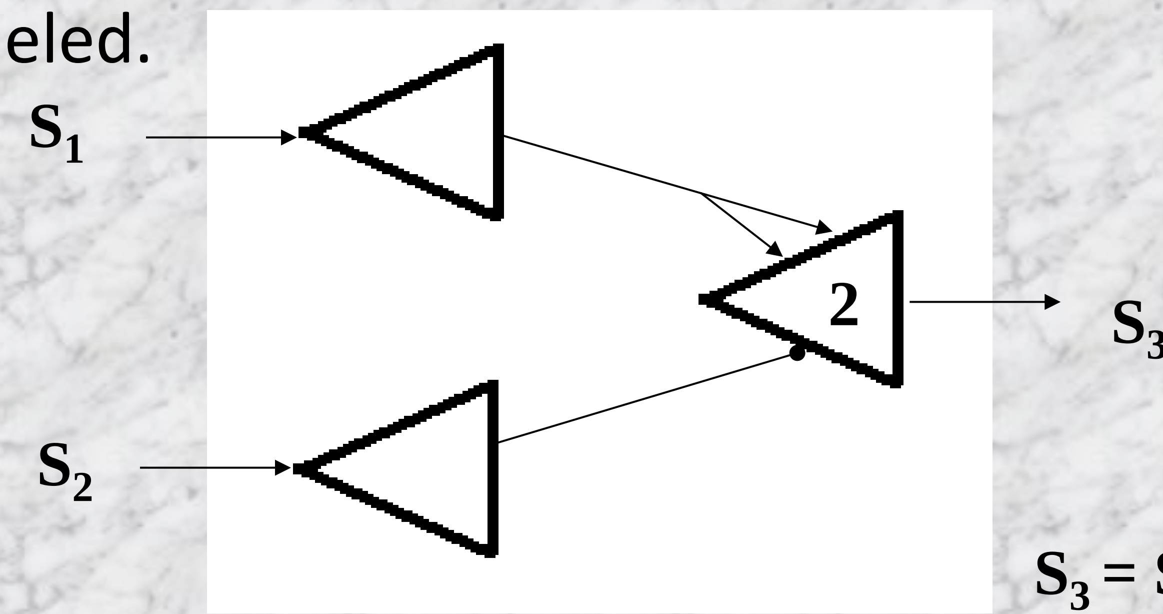
From these simple elements - formal neurons - the nets simulating complex operations or some forms of the behavior of living organisms can be modeled.



$S_3 = S_1 \wedge S_2$

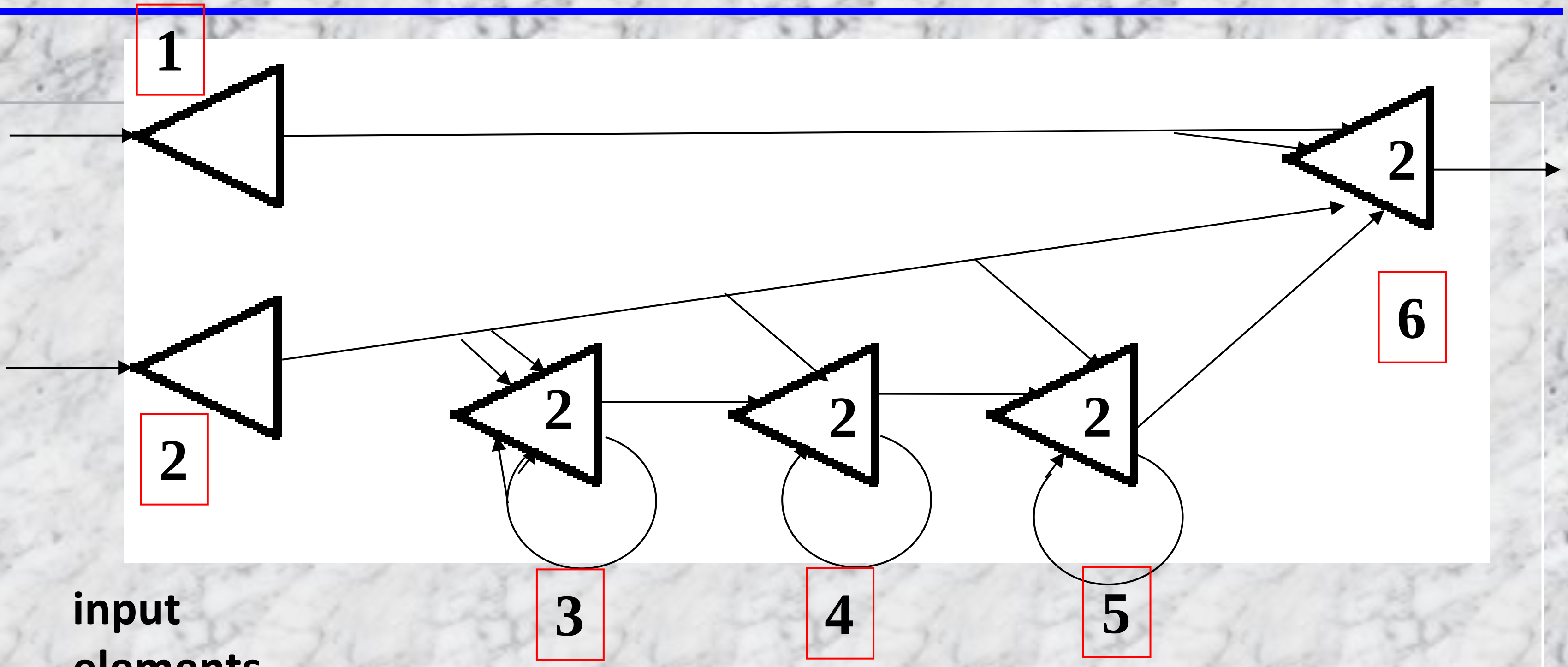
Simple nets build from McCulloch &Pitts elements

From these simple elements - formal neurons - the nets simulating complex operations or some forms of the behavior of living organisms can be modeled.



$$S_3 = S_1 \wedge \sim S_2$$

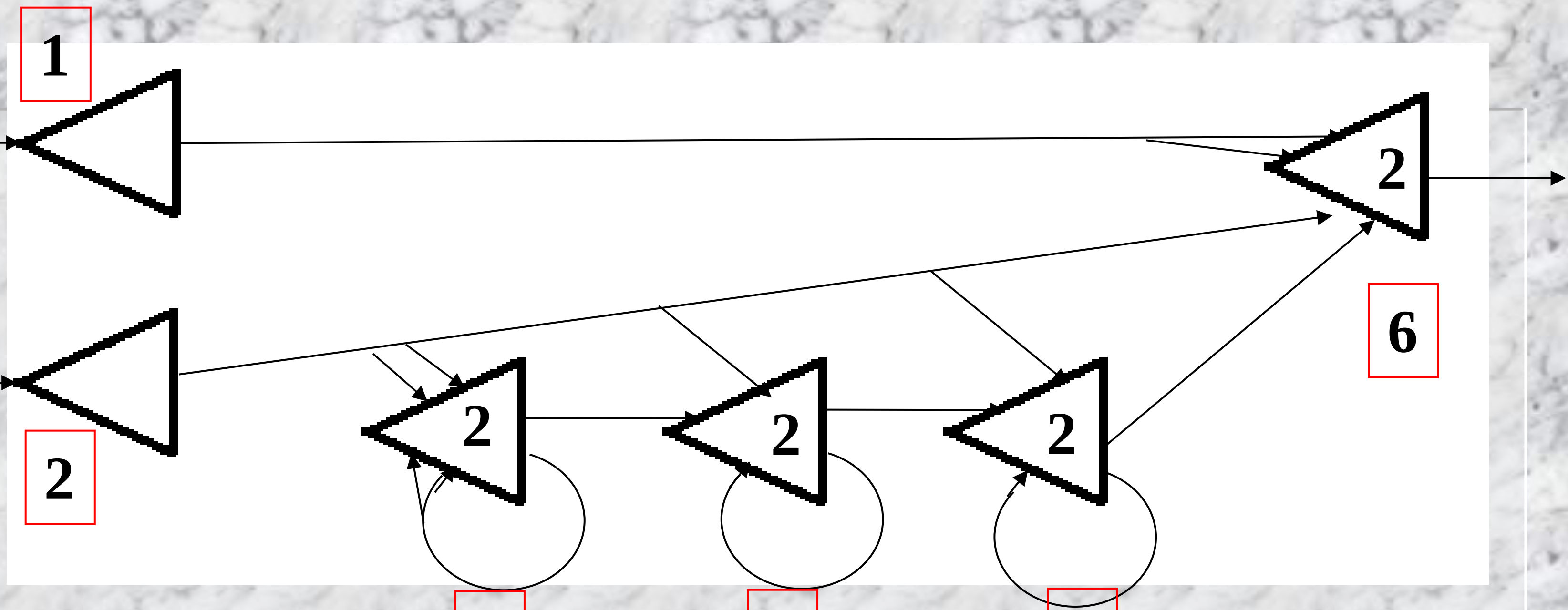
Simple nets build from McCulloch &Pitts elements, facilitation phenomenon



Signals incoming to input: 1 directly excite the element 6

Signal incoming to input: 2 excites 6 after 3 times repetition

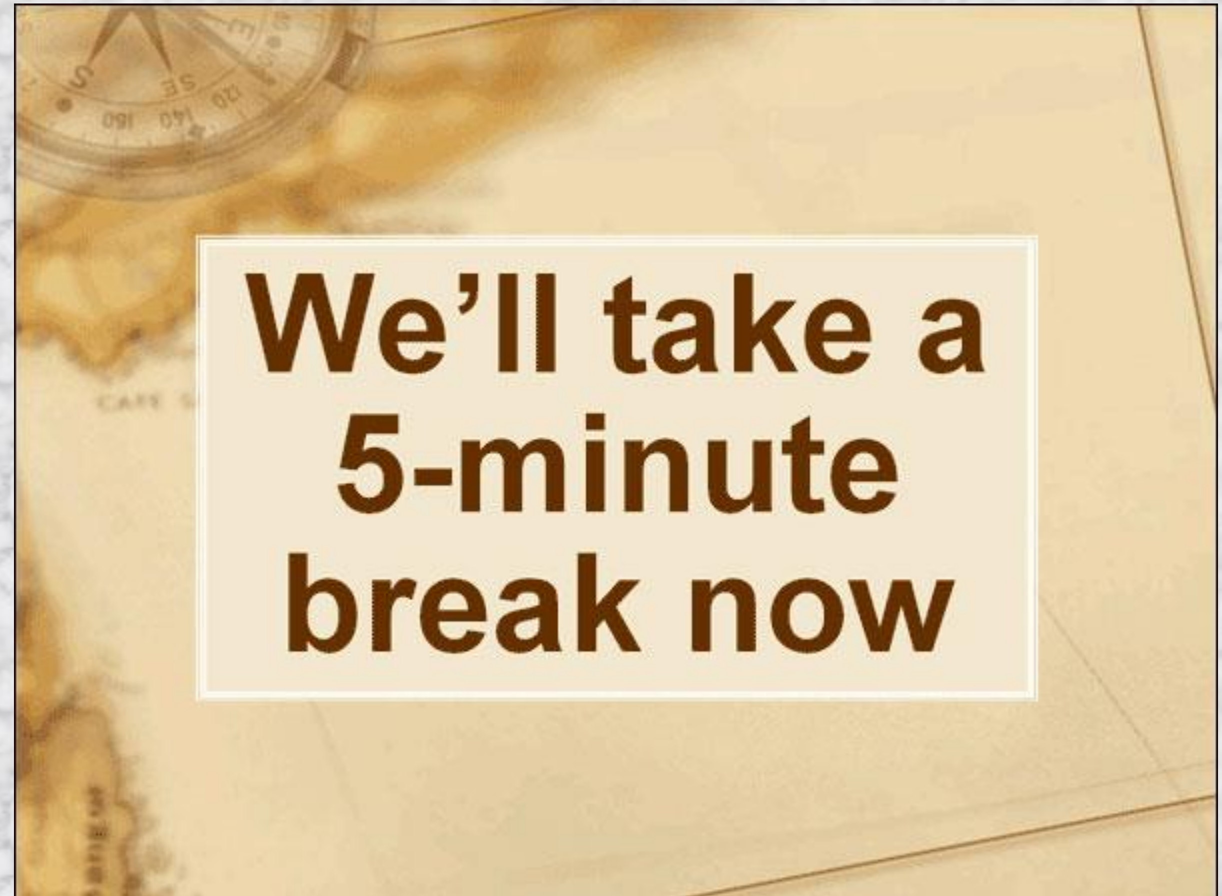
Simple nets build from McCulloch &Pitts elements, facilitation phenomenon



First excitation of 2 excite 3 but is not enough to excite the others

This excitation yields to self excitation (positive feedback) of the 3

Output from 3 approaches 4 (but below threshold). Totally 2 and 3 excite 4


We’ll take a
5-minute
break now

McCulloch symbolism

McCulloch symbolism

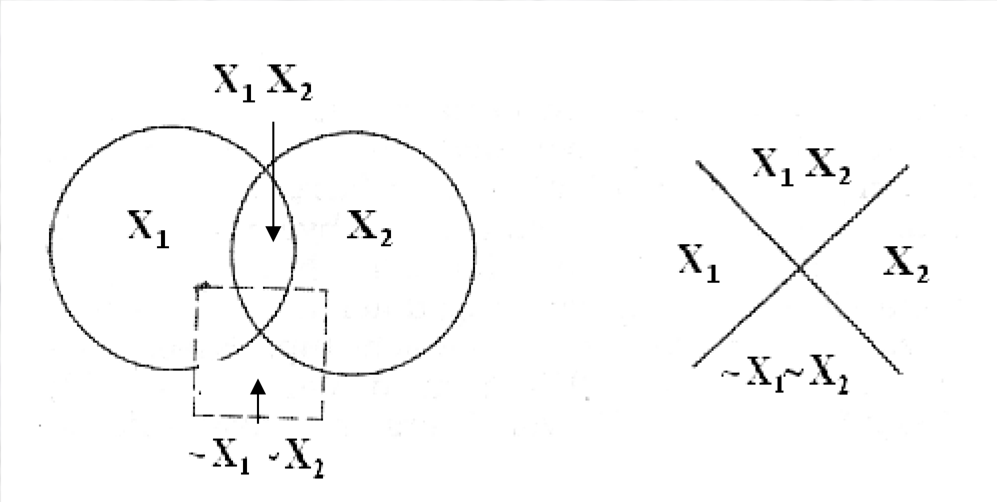
The symbolism introduced by McCulloch at the basis of simplified Venn diagrams is very useful in the analysis of logical networks

Two areas X_1 i X_2 correspond to two argument logic function. Symbol X_1 means the input signal $x_1 = 1$, its complement – signal $x_1 = 0$. The same for X_2.

McCulloch symbolism

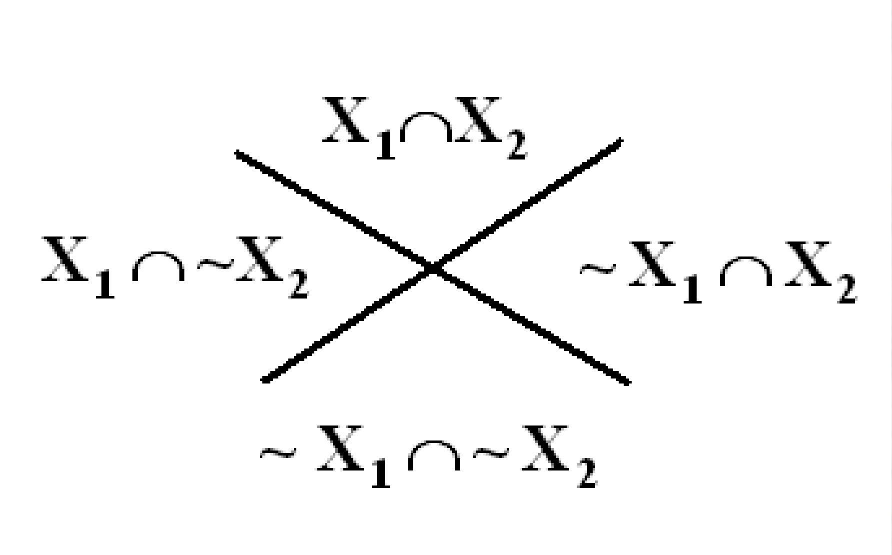
We have four fragments denoted:

$X_1 \cap X_2$, $X_1 \cap \sim X_2$, $\sim X_1 \cap X_2$ and $\sim X_1 \cap \sim X_2$



McCulloch symbolism

Instead of circles we can used crosses

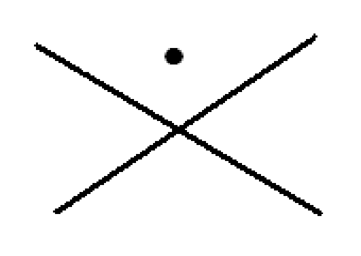


McCulloch symbolism

Symbolic notation – cross with dots

conjunction –
AND operation

$$X_1 \cap X_2 \Rightarrow$$

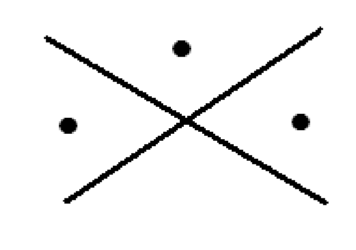


$$X_1 \cup X_2 = (X_1 \cap \sim X_2) \cup (\sim X_1 \cap X_2) \cup (X_1 \cap X_2)$$

disjunction -

OR operation

$$\Rightarrow$$



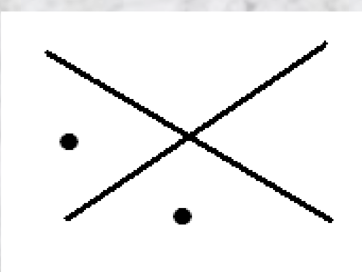
McCulloch symbolism

Symbolic notation – cross with dots

negation –

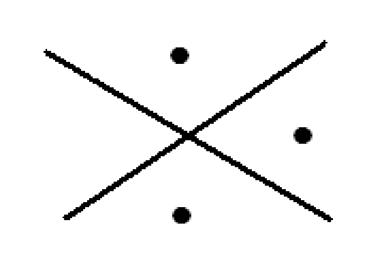
NOT operation

$\sim X_2 \quad \Rightarrow$



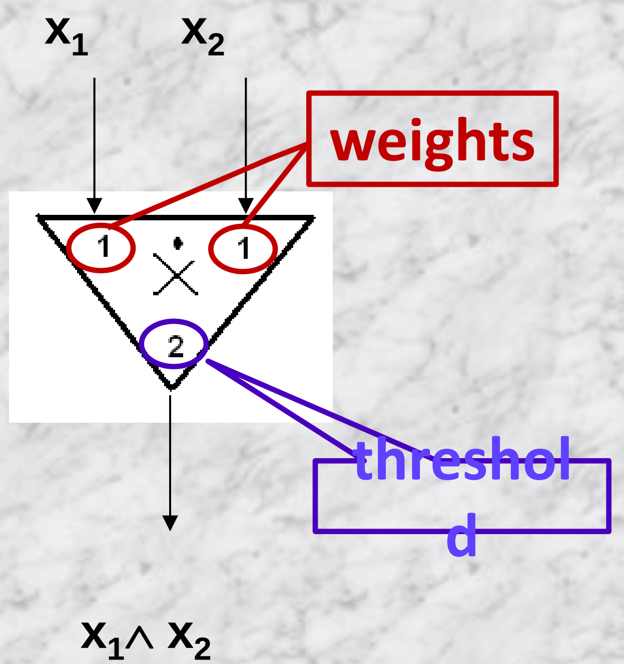
implication

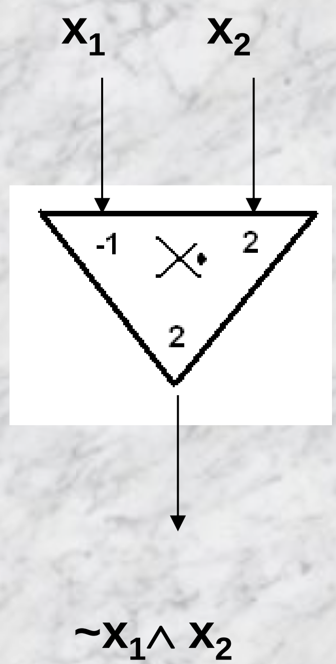
$X_1 \Rightarrow X_2 \quad \Rightarrow$



McCulloch symbolism

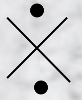
The function depending of parameters

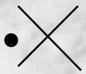
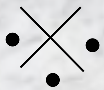




McCulloch symbolism

Operations performed

x_1  $x_2 \equiv (x_1 \wedge x_2) \vee (\sim x_1 \wedge \sim x_2)$

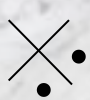
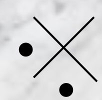
x_1  $($ x_2  $x_3)$ $\equiv x_1 \wedge \sim [(x_2 \wedge \sim x_3) \vee (x_3 \wedge \sim x_2) \vee (\sim x_2 \wedge \sim x_3)]$

$$\equiv x_1 \wedge \sim [\sim (x_2 \wedge x_3)]$$

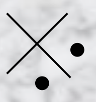
$$\equiv x_1 \wedge \sim (\sim x_2 \vee \sim x_3)$$

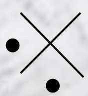
McCulloch symbolism

Direct operations on symbols

   $\equiv$ 

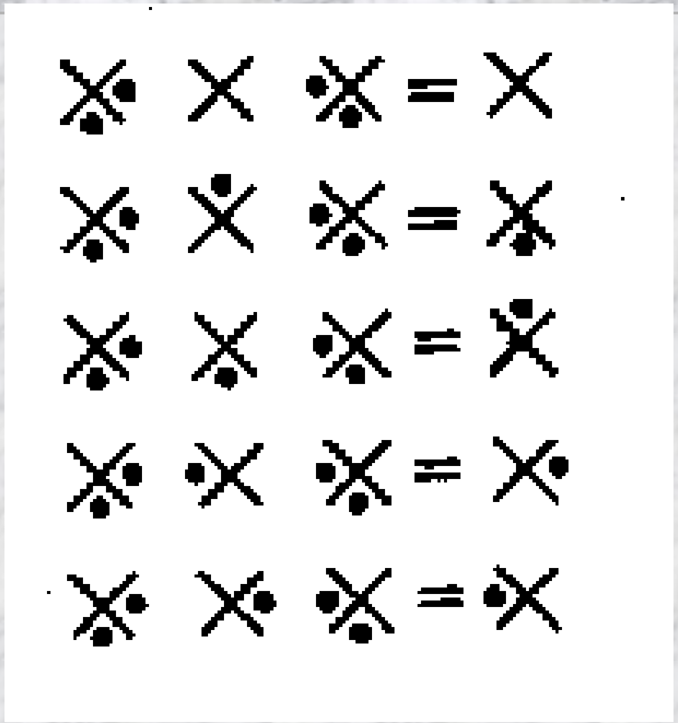
Proof:

x_1  $x_2 \equiv (\sim x_1 \wedge \sim x_2) \vee (\sim x_1 \wedge x_2) \equiv \sim x_1$

x_1  $x_2 \equiv (x_1 \wedge \sim x_2) \vee (\sim x_1 \wedge \sim x_2) \equiv \sim x_2$

$\sim x_1$  $\sim x_2 \equiv \sim x_1 \wedge \sim x_2 \equiv x_1$  x_2

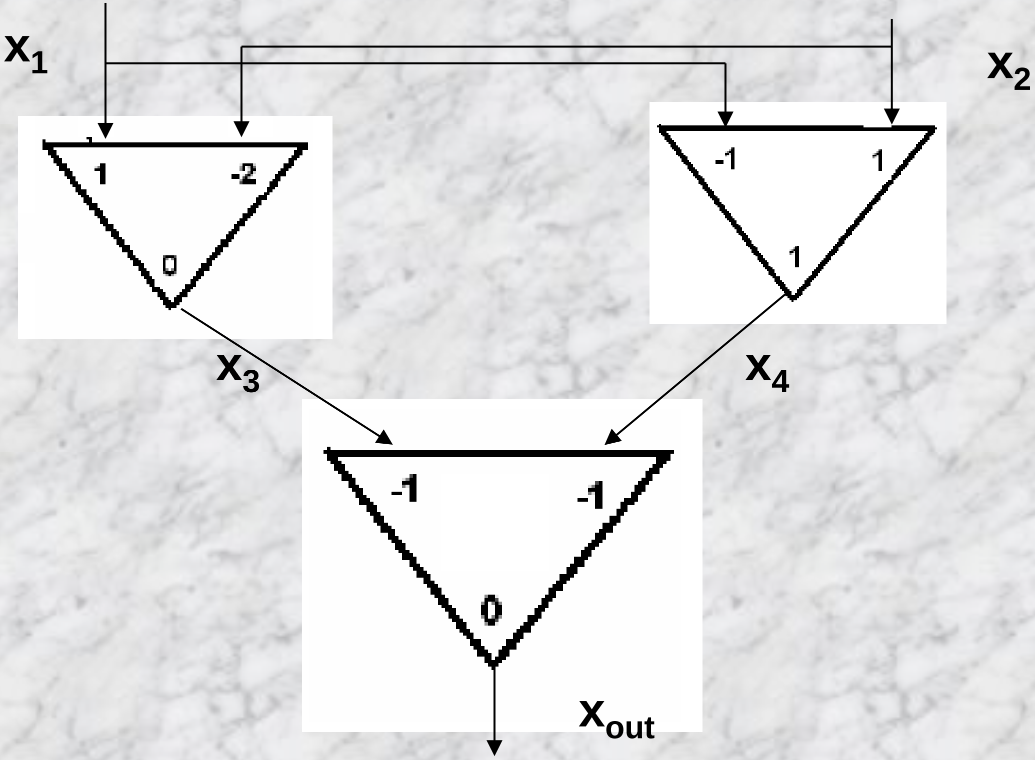
McCulloch symbolism



Applications of McCulloch symbols, operation on the symbols

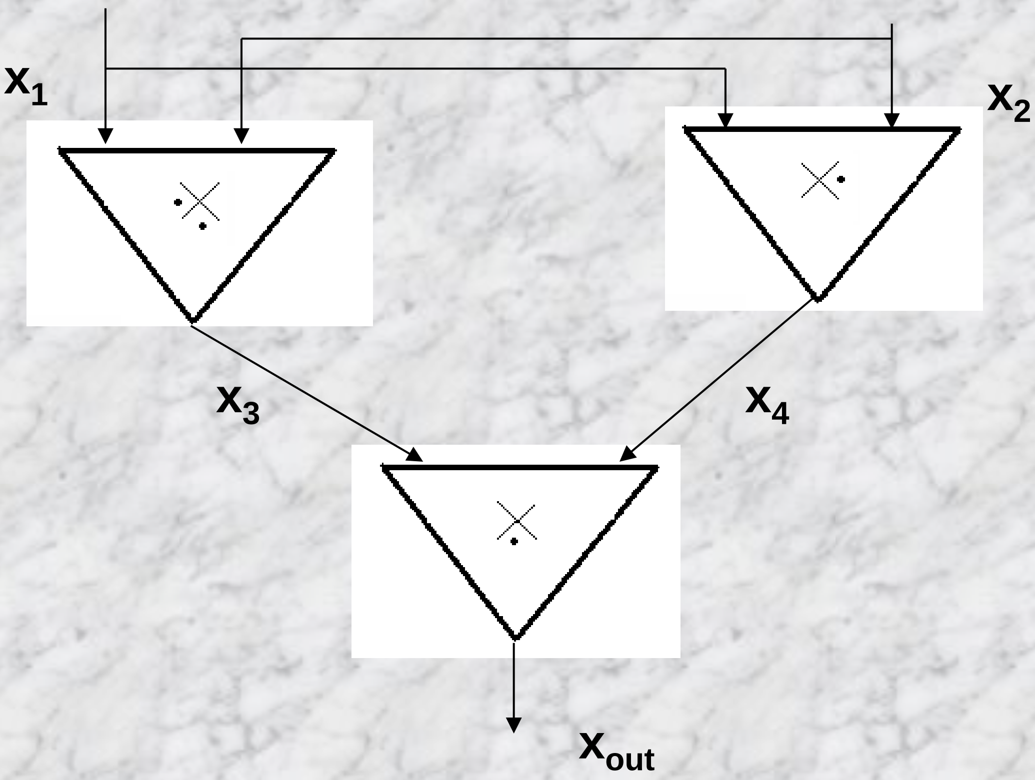
McCulloch symbolism

Analysis of the simple nets composed from logical neurons



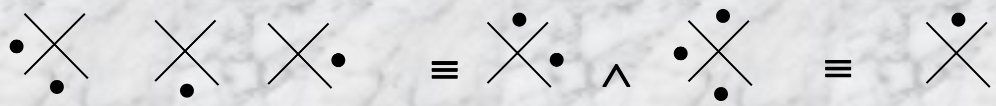
McCulloch symbolism

Simplified notation

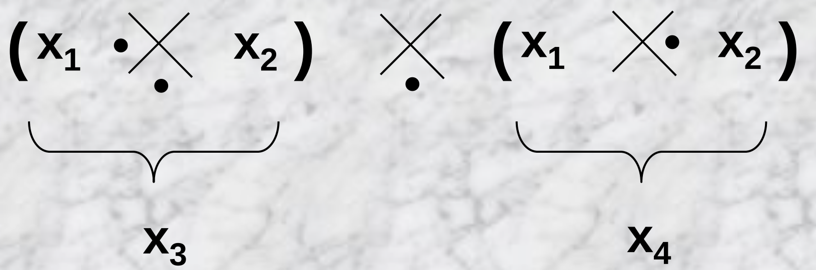


McCulloch symbolism

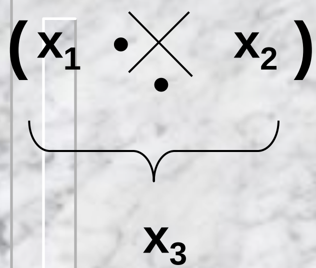
The middle cross denotes an operation performed on the two symbols on either side. For example, the operation below means the operation which is not entered in either symbol on the left or symbol on the right should be written down as the result.



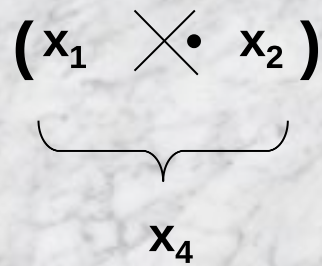
Proof



McCulloch symbolism



$$x_3 \equiv \sim x_2$$



$$x_4 \equiv \sim x_1 \wedge x_2$$

x_3  x_4 $\equiv \sim x_3 \wedge \sim x_4 \equiv$

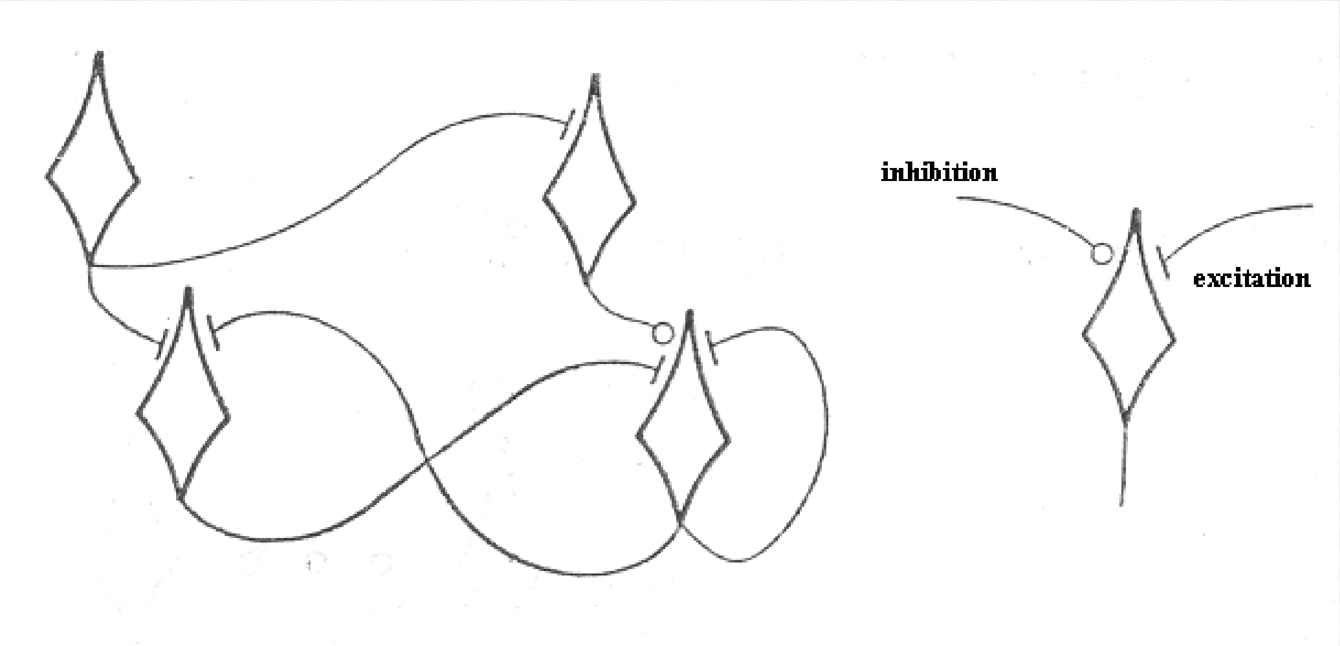
$$\equiv \sim(\sim x_2) \wedge \sim(\sim x_1 \wedge x_2) \equiv$$

$$\equiv x_2 \wedge (x_1 \vee \sim x_2) \equiv$$

$$\equiv (x_2 \wedge x_1) \vee \underbrace{(x_2 \wedge \sim x_2)}_{\varnothing} \equiv$$

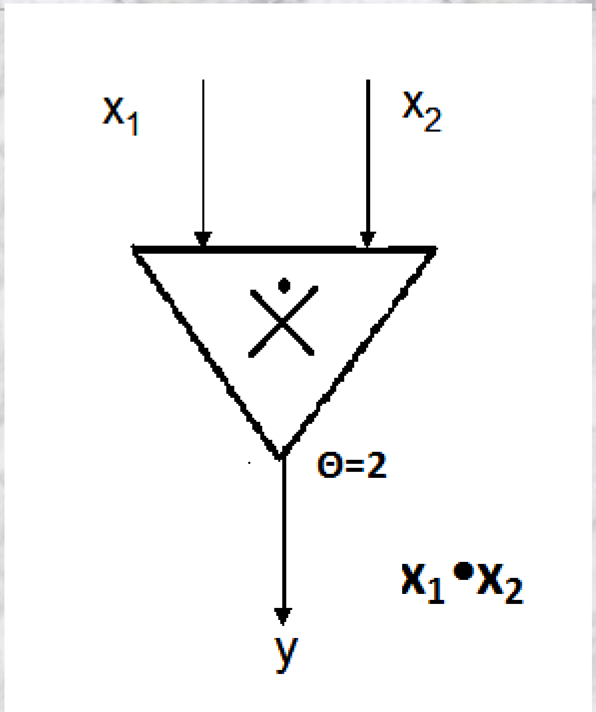
$$\equiv x_1 \wedge x_2$$

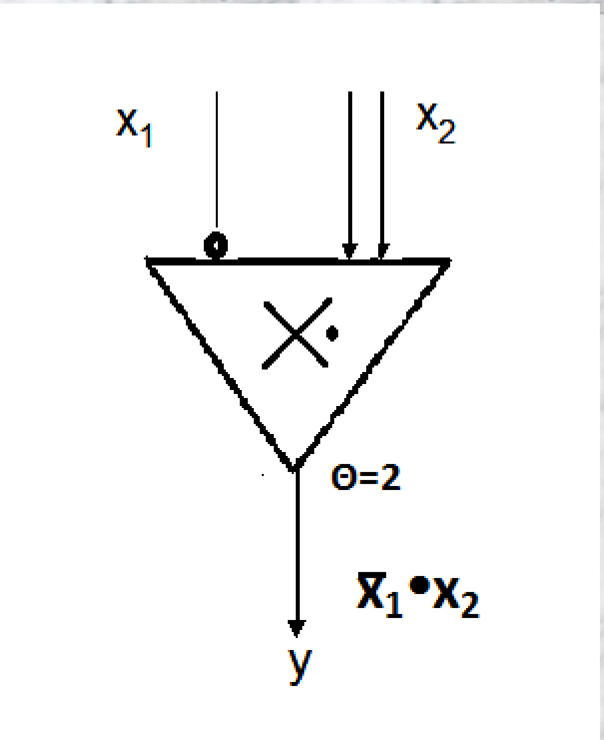
McCulloch symbolism



The net with the loops

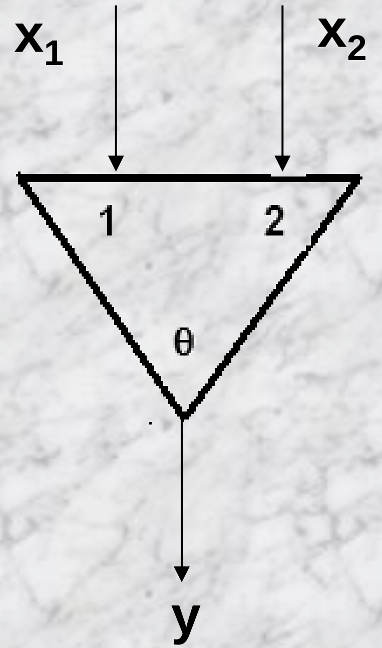
McCulloch symbolism





Use of McCulloch symbols to denote the function of a neuron

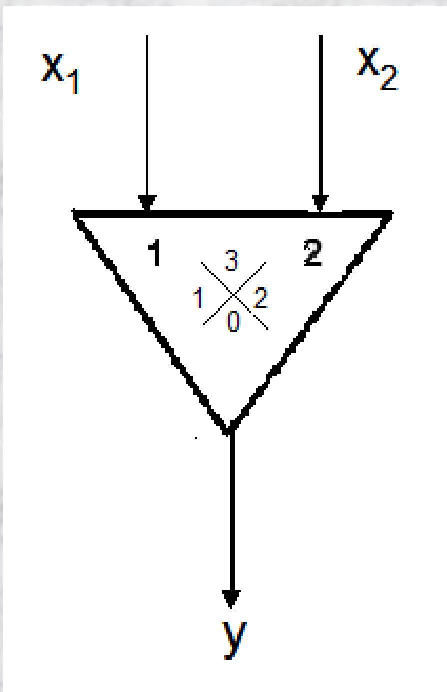
McCulloch symbolism



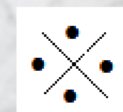
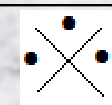
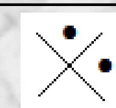
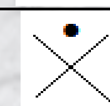
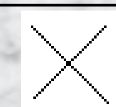
Different function realization depending of the threshold value

$\Theta = 0$	$y = 1$
$\Theta = 1$	$y = x_1 \vee x_2$
$\Theta = 2$	$y = x_2$
$\Theta = 3$	$y = x_1 \wedge x_2$
$\Theta = 4$	$y = 0$

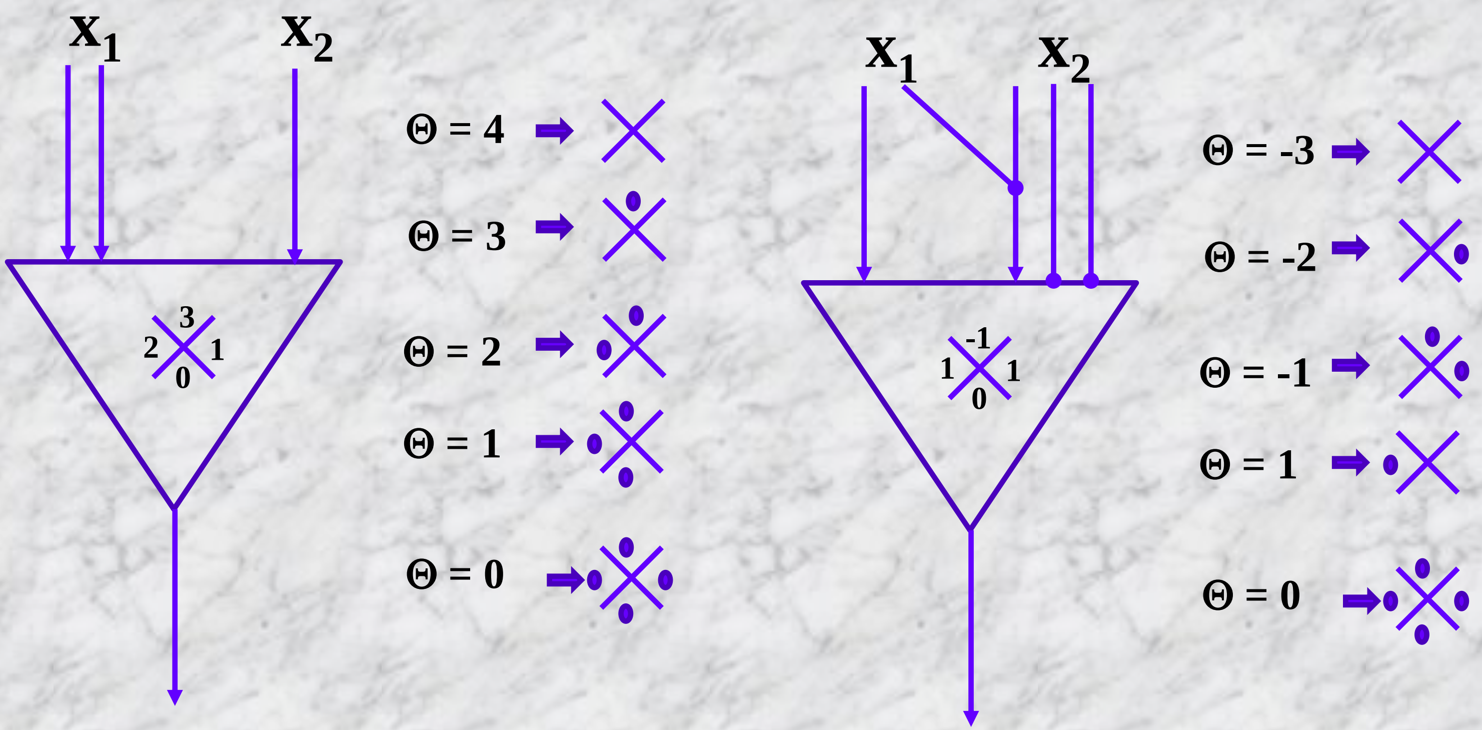
McCulloch symbolism



Projection of the function variability realized by the neuron according to parameters value

$\Theta = 0$	
$\Theta = 1$	
$\Theta = 2$	
$\Theta = 3$	
$\Theta = 4$	

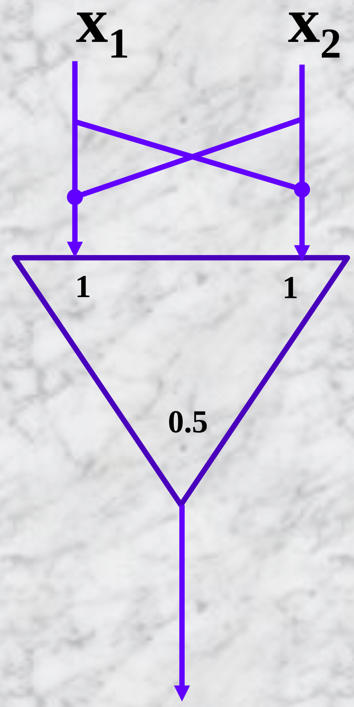
McCulloch symbolism



Threshold influence for the neuron reaction

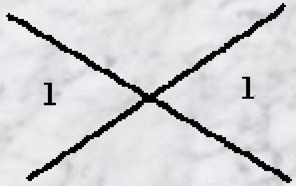
McCulloch symbolism

Presynaptic inhibition

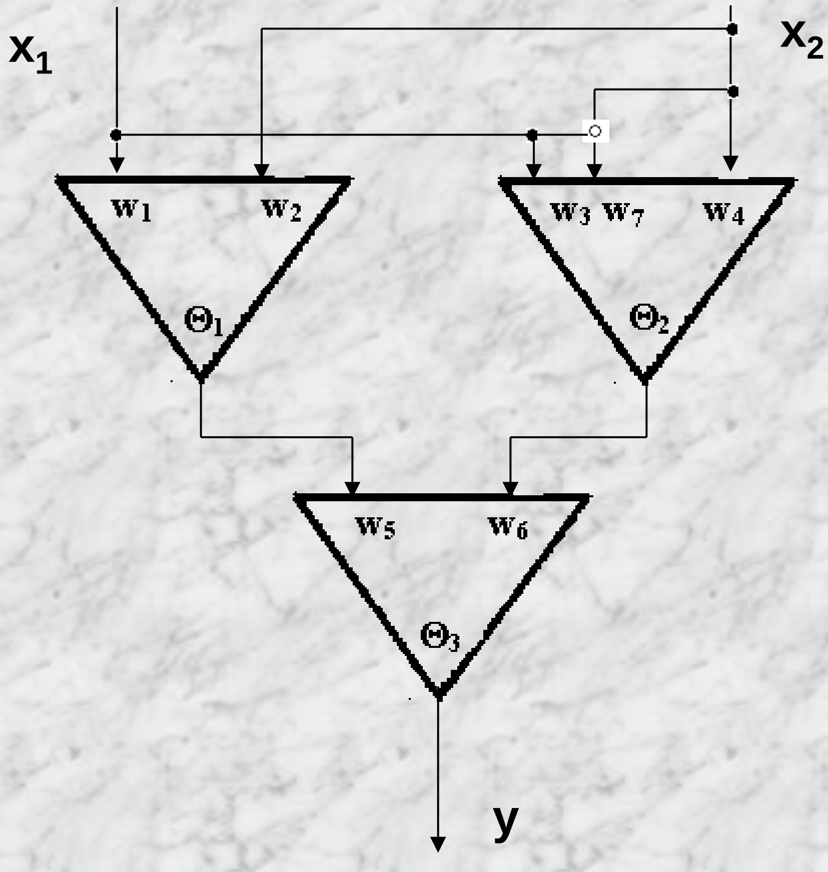


$(x_1 \wedge \sim x_2) \vee (\sim x_1 \wedge x_2)$

Threshold influence



McCulloch symbolism

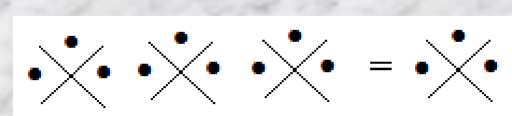
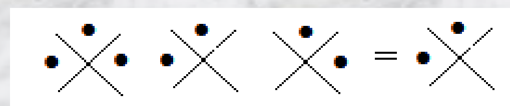
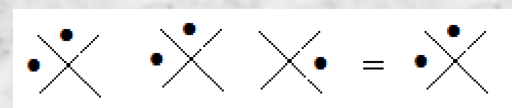
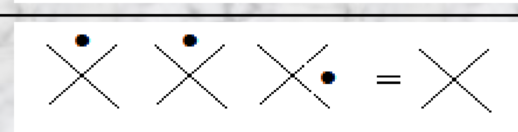


Threshold influence

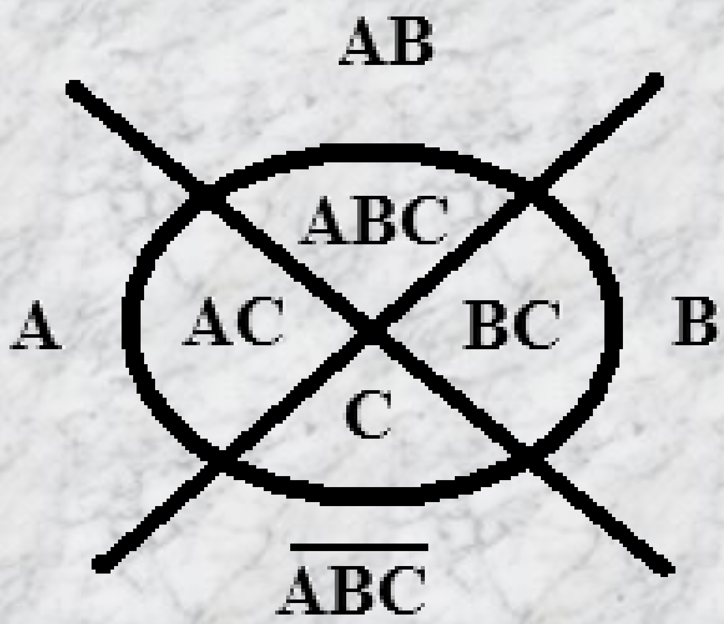
w_1	2
w_2	1
w_3	1
w_4	1
w_5	2
w_6	1
w_7	4

ver	Θ_1	Θ_2	Θ_3
1	-0.5	0.5	-0.5
2	0.5	1.5	0.5
3	1.5	2.5	1.5
4	2.5	3.5	2.5

McCulloch symbolism

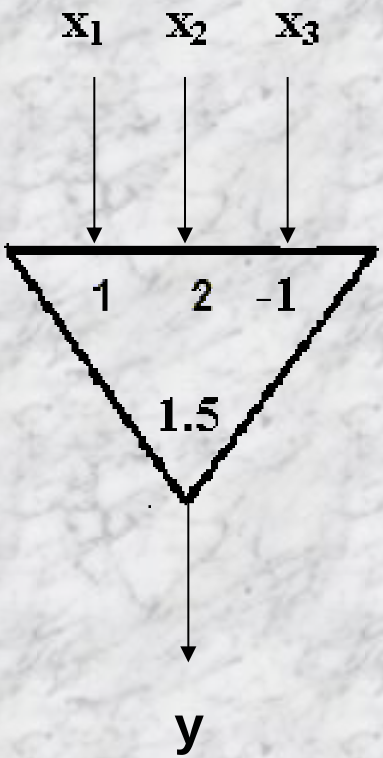
ver	McCulloch notation
1	
2	
3	
4	

McCulloch symbolism for three inputs

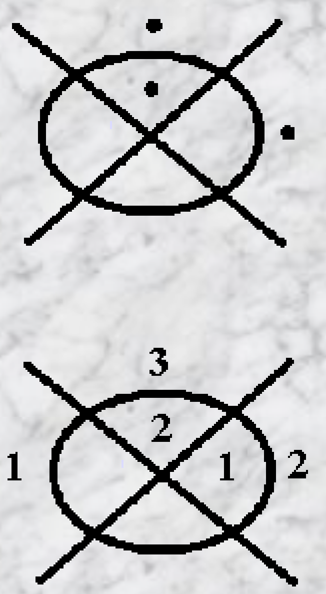


Venn diagram and McCulloch symbols for three inputs. Unknown are marked by A, B and C.

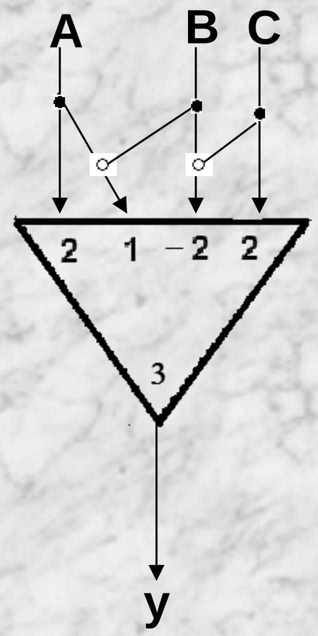
McCulloch symbolism for three inputs



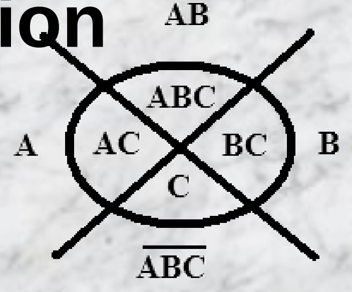
The element with three inputs

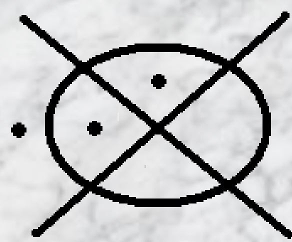


McCulloch symbolism for three intputs

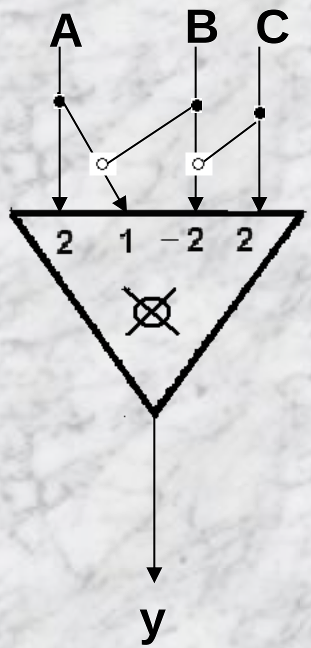


Example of the realization of the function of 3 variables by the neuron with presynaptic inhibition



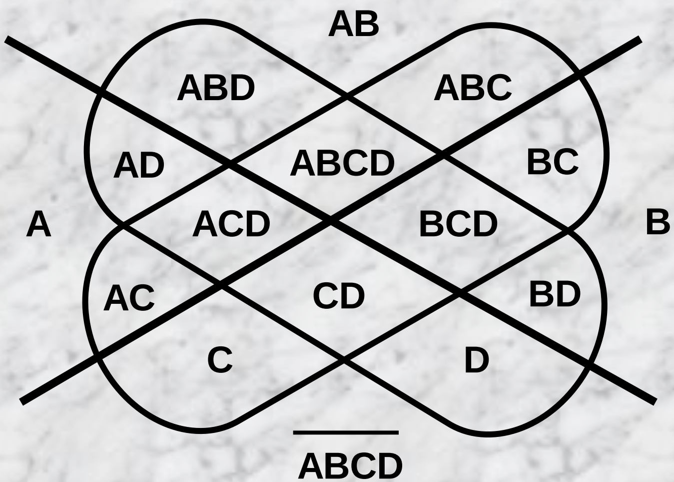


McCulloch symbolism for three intputs



Example of the realization of the function of 3 variables by the neuron with presynaptic inhibition, threshold influence

McCulloch symbolism four outputs

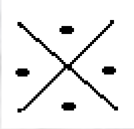
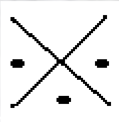
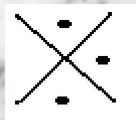
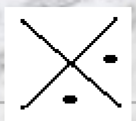


Att: $AB = AB\bar{C}\bar{D}$

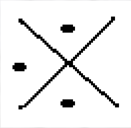
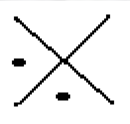
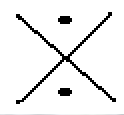
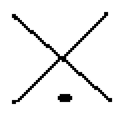
Venn diagram and McCulloch symbols for four inputs. Unknown are marked by A, B, C and D.

Logical functions of two unknown

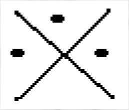
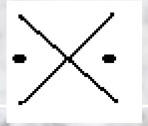
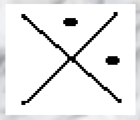
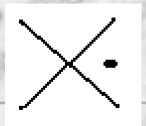
Logical functions of two unknown

Function	Formula	Description	Diagram	00	01	10	11
Const 1	1	$(A \wedge B) \vee (A \wedge \sim B) \vee (\sim A \wedge B) \vee (\sim A \wedge \sim B)$		1	1	1	1
NAND	$\sim(A \wedge B)$	$(A \wedge \sim B) \vee (\sim A \wedge B) \vee (\sim A \wedge \sim B)$		1	1	1	0
Implication	$A \Rightarrow B$	$(A \wedge B) \vee (\sim A \wedge B) \vee (\sim A \wedge \sim B)$		1	1	0	1
Negation A	$\sim A$	$(\sim A \wedge B) \vee (\sim A \wedge \sim B)$		1	1	0	0

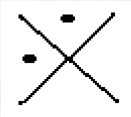
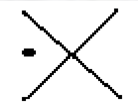
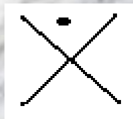
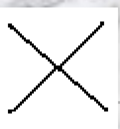
Logical functions of two unknown

Function	Formula	Description	Diagram	00	01	10	11
Implication	B⇒A	(A∧B)∨(A∧~B)∨ (~A∧~B)		1	0	1	1
Negation B	~B	(A∧~B)∨(~A∧~B)		1	0	1	0
equivalence	A≡B	(A∧B) ∨ (~A∧~B)		1	0	0	1
NOR	~(A∨B)	(~A∧~B)		1	0	0	0

Logical functions of two unknown

Function	Formula	Description	Diagram	00	01	10	11
disjunction	A∨B	(A∧B)∨(A∧~B)∨ (~A∧B)		0	1	1	1
non-equivalence	~(A≡B)	(A∧~B)∨(~A∧B)		0	1	1	0
B	B	(A∧B)∨(~A∧B)		0	1	0	1
negation of implication	~A∧B	(~A∧B)		0	1	0	0

Logical functions of two unknown

Function	Formula	Description	Diagram	00	01	10	11
A	A	(A∧B)∨(A∧~B)		0	0	1	1
negation of implication	A∧ ~ B	(A∧~B)		0	0	1	0
conjunction	A ∧ B	(A∧B)		0	0	0	1
constant 0	0			0	0	0	0


We'll take a 5-minute break now

Neural Networks

What is a Neural Network ?

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

Chris Stergiou

Neural Networks-definition

[Zurada, J.M., *Introduction to Artificial Neural Systems,* 1992]

Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store, and utilize experiential knowledge

[Cichocki A. & Umbehauen R. *Neural Networks for Optimization and Signal Processing,* 1994]

(...) an artificial neural network is an information or signal processing system composed of a large number of simple processing elements, called artificial neurons, or simply nodes, which are interconnected by direct links called connections and which cooperate to perform parallel distributed processing in order to solve a desired computational task

Neural Networks-definition

Haykin, S., *Neural Networks:A Comprehensive Foundation,* 1994

A neural network is a massively parallel distributed processor made up of simple processing units (known as neurons), which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the human brain in two respects:

- Knowledge is acquired by the network from its environment through a learning process.
- Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge

DARPA Neural Network Study (1988, AFCEA International Press, p. 60):

... a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.

Neural Networks

At the beginning was the idea that it is enough to build the net of many randomly connected elements to get the model of the brain operation.

Question: **how many element is necessary for the process of self organization ??**

Neural Networks

Research of McCulloch, Lettvin, Maturana, Hartlin and Ratliff.

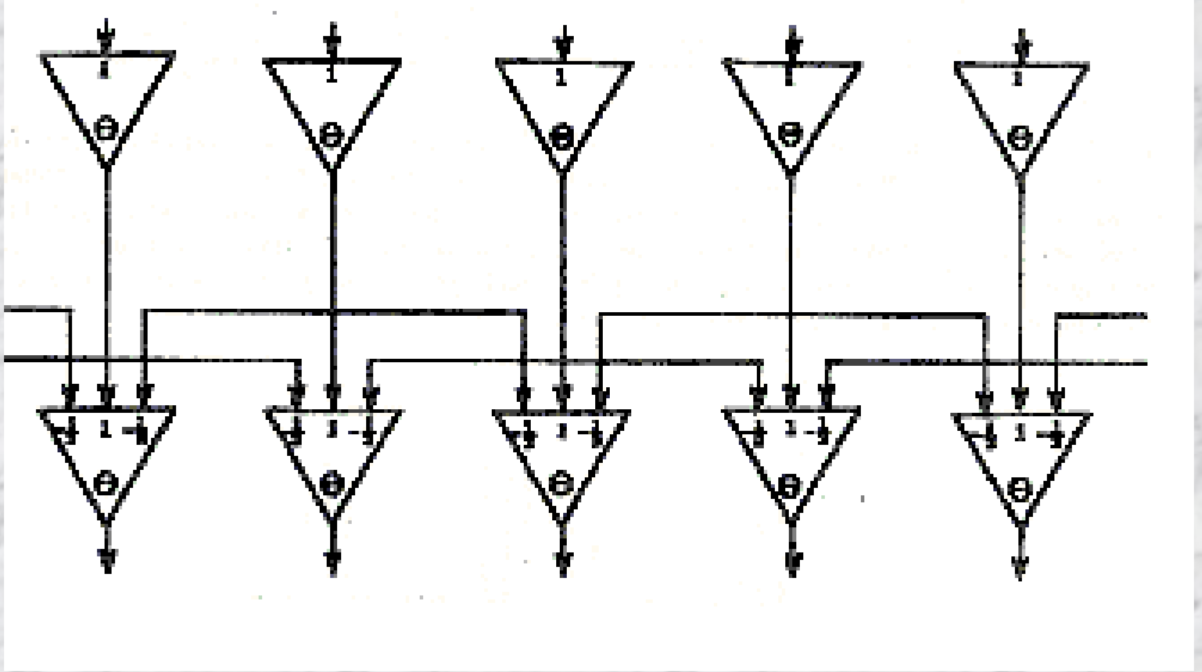
Research on the frog's eye and specially on the compound eye of the horseshoe cram - *Limulus*.

Hubel and Wiesel research on the mammals visual system.

Some parts are constructed in the very special, regular way.

Neural Networks

Two – layers chain structure



Neural Networks

The input layer of photoreceptors and the layer of processing elements which will locate the possible changes in the excitation distribution.

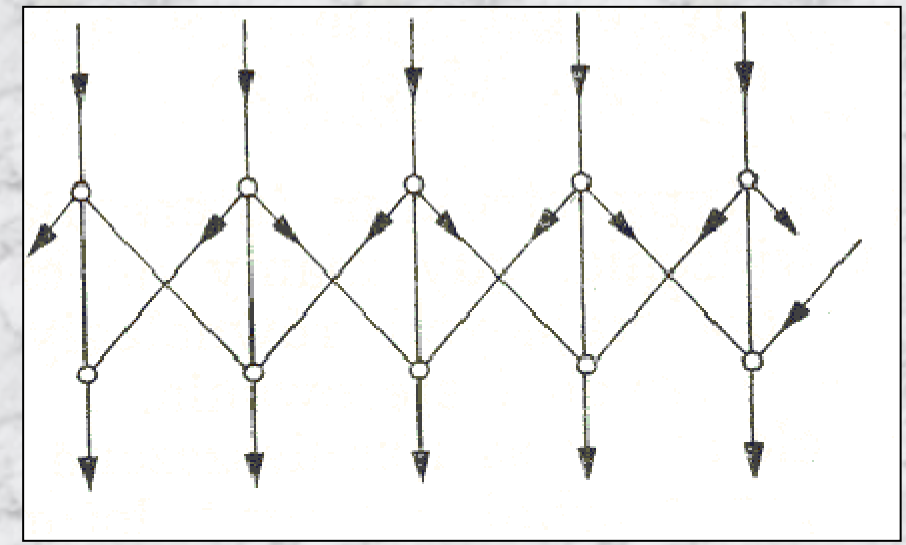
Connection rule:

Each receptor cell is to excite one element (exactly below). In addition to the excitatory connections there are also inhibitory connections (for the simplicity - to the adjacent cells only) which reduce the signal to the neighbors.

Neural Networks

The inhibition range can differs.

This is known as the of *lateral inhibition*



Neural Networks

As can be easily seen the uniform excitation of the first layer will not excite the second layer. The excitatory and inhibitory signals will be balanced.

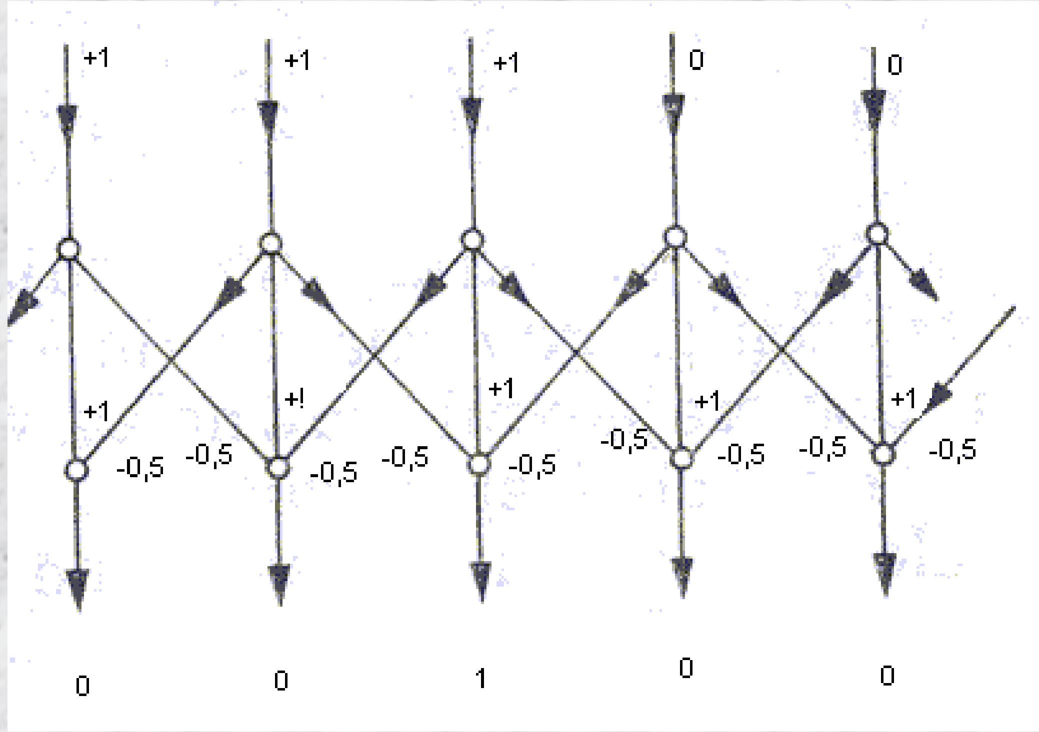
A step signal is a step change in the spatial distribution. The distribution of output signal is not a copy of the input signal distribution but is the convolution of the input signal and the weighting function.

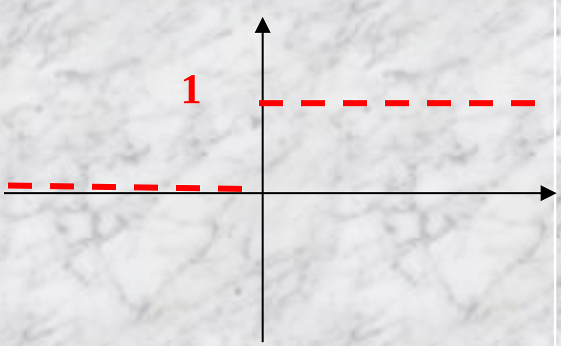
Neural Networks

The point in which the step change occurs is exaggerated at each side by increasing and decreasing the signal resulting in the single signal at the point of the this step.

Neural Networks

Input Signals





Output Signals

Elements' transfer function

Neural Networks

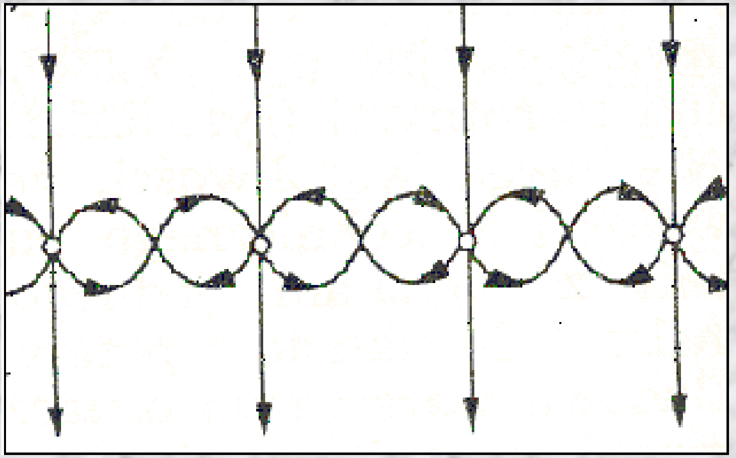
As you can see such a network gives the possibility to locate the point where the changes in the excitation were enough high (terminations, inflections, bends etc.).

From the neurophysiology we know on the existence of the opposite operation ***lateral excitation.***

These nets allows to detect the points of crossing or branchings etc.

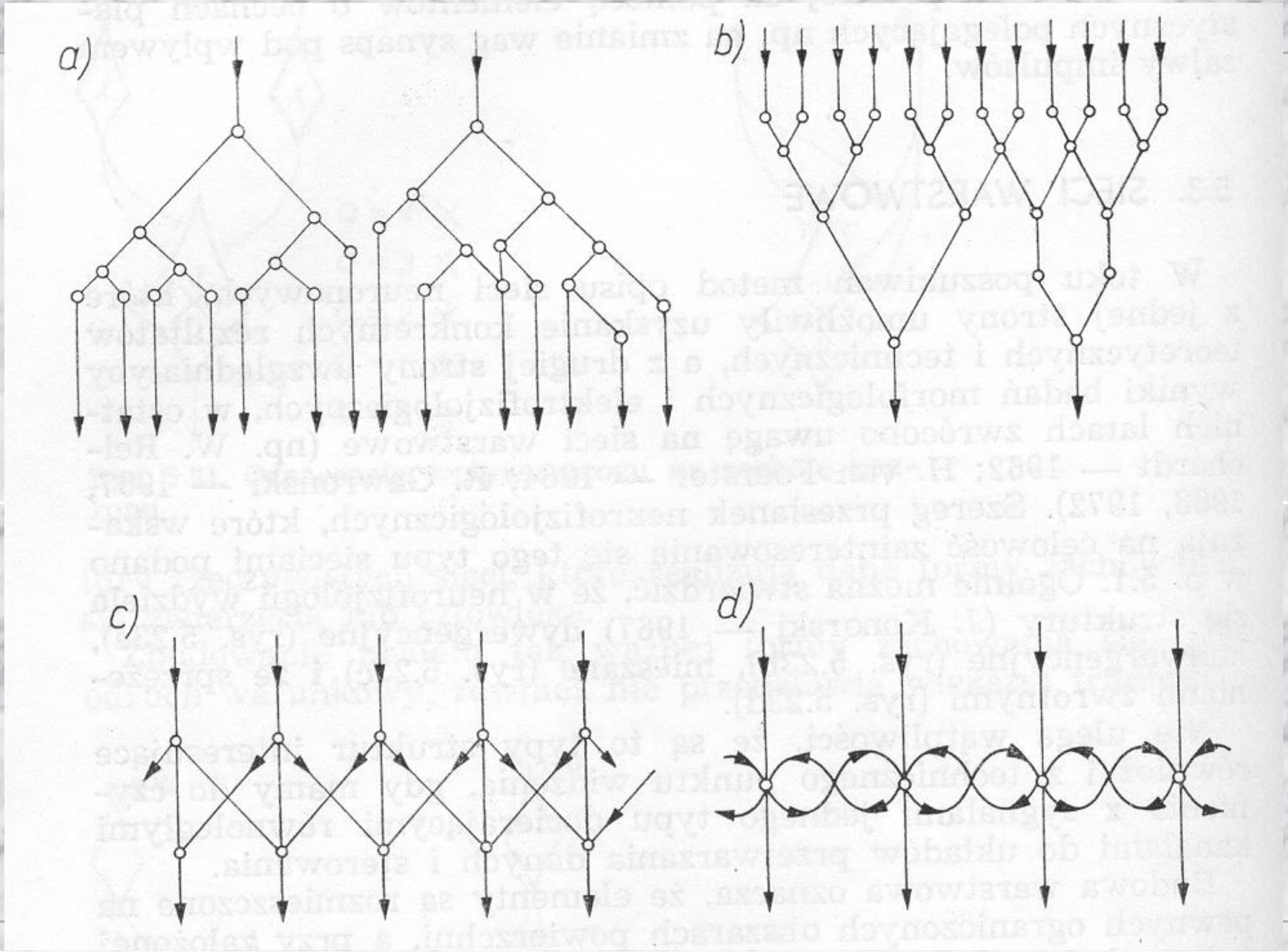
Neural Networks

The lateral inhibition rule can be realized be the one dimensional net with negative feedback



<u>Attention:</u> elements are nonlinear and the feedback loops make analysis difficult; such the networks can be non stable and the distribution of the input signals does not depends univocally from the input signals.

Another simple neural nets



The Perceptron

The Perceptron

In 1962 Frank Rosenblatt introduced the new idea of the perceptron.

General idea: a neuron learns on its mistakes!!

If the element output signal is wrong – the changes are to minimize the possibilities that such the mistake will be repeated.

If the element output signal is correct

– there are no changes.

The Perceptron

The one layer perceptron is based on the McCulloch & Pitts threshold element. The simplest perceptron - Mark 1 – is composed from four types of elements:

- layer of input elements, (square grid of 400 receptors), *elements type S* receiving stimuli from the environment and transforming those stimuli into electrical signals
- associative elements, *elements type A,* threshold adding elements with excitatory and inhibitory inputs

The Perceptron

- output layer – *elements type R,* the reacting elements, randomly connected with the *A* elements,
  set of *A* elements correspond with to each element,
  R passes to state 1 when its total input signal is greater than zero
- control units

Phase 1 - learning. At the beginning, e.g presentation of the representatives of the first class.
Phase 2 – verification of the learning results
Learning of the second class etc..

The Perceptron

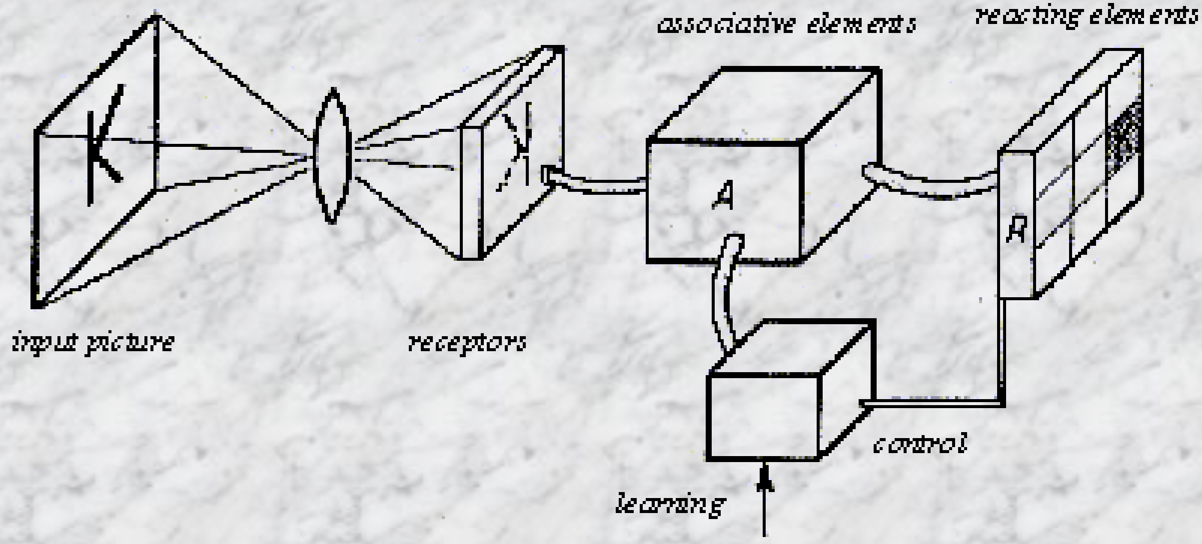
Mark 1:

400 threshold elements of the type *S*; if they are enough excited – they produce at the output one the signal +1 and at the output two the signal -1.

The associative element *A*, has 20 inputs, randomly (or not) connected with the *S* elements outputs (excitatory or inhibitory). In Mark 1 was 512 elements of type *A*.

The *A* elements are randomly connected with the elements type *R*. In Mark 1 was 8 elements of type *R*.

The Perceptron



A block diagram of a perceptron. On the receptive layer the picture of the letter K is projected. As the result, in the reacting layer, the region corresponding to letter K (in black) is activated.

The Perceptron

Each element *A* obtain „weighted sum" of an input signal.
When the number of excitatory signals > than the number of inhibitory signals – at the output the +1 signal is generated.
When < there is no signal generation.
Elements *R* are reacting on the added input from the elements *A*. When the input is > than the threshold – The +1 signal is generated, otherwise – signal 0.

Learning means changes in weights of active elements *A*.

The Perceptron

Simplified version:

Two layers – input and output. Active is only the layer two. Input signals are equal 0 or +1. Such the structure is called one layer perceptron.

Elements (possibly only one) of the output layer obtain at their input the weighted signal from the input layer. If this signal is greater than the defined threshold value – the signal +1 is generated, otherwise the signal 0.

The learning method is based on the correction of weights connecting the input layer with the elements of the output layer. Only the active elements of the input layer are the subject of correction.

Weights modification rule

$$w_{iA}\ (new) = w_{iA}\ (old) - input_i$$

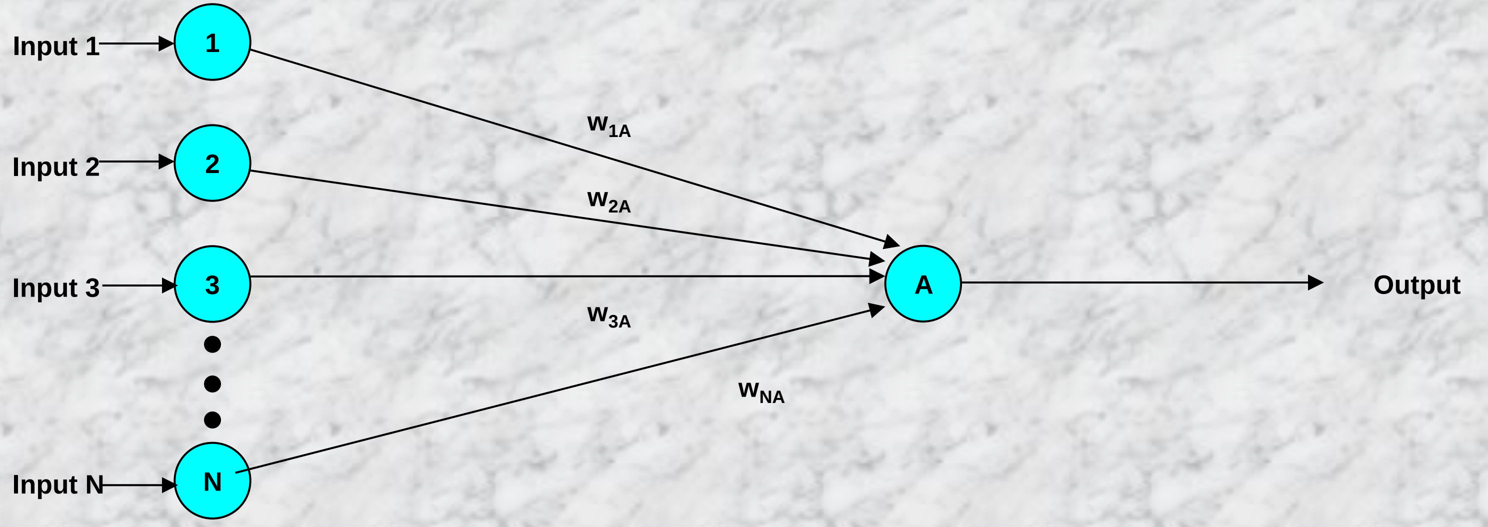
$$w_{iB}\ (new) = w_{iB}\ (old) + input_i$$

$$input_i = \pm 1$$

The Perceptron

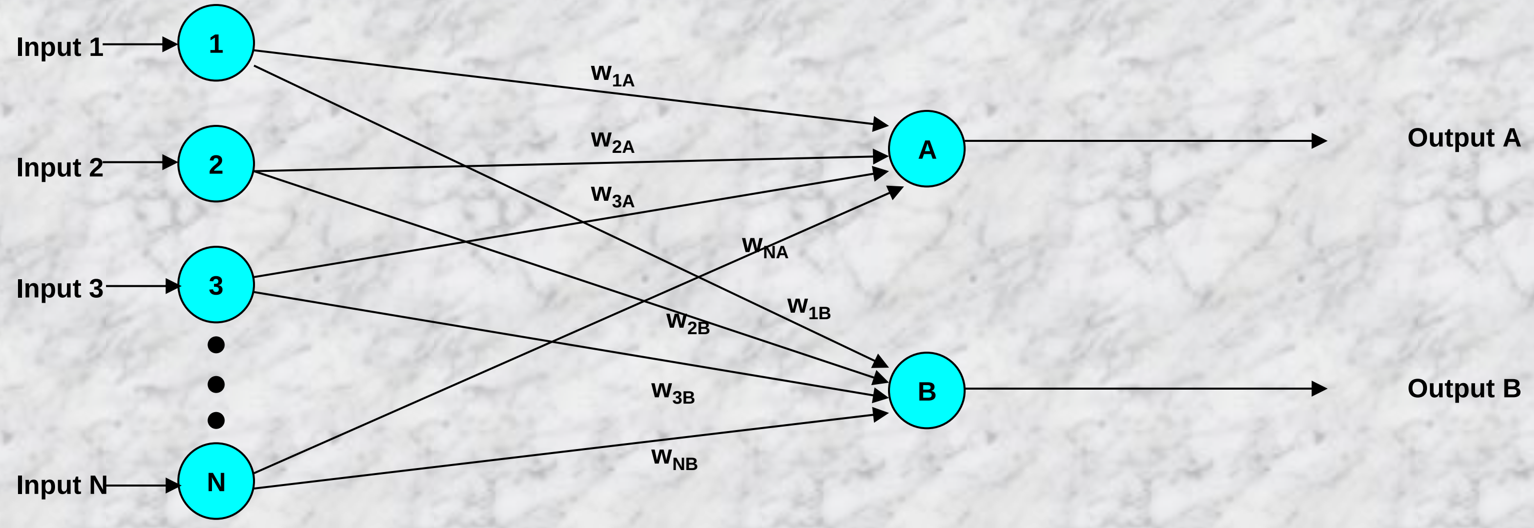
The example

The one-layer and two-elements Perceptron



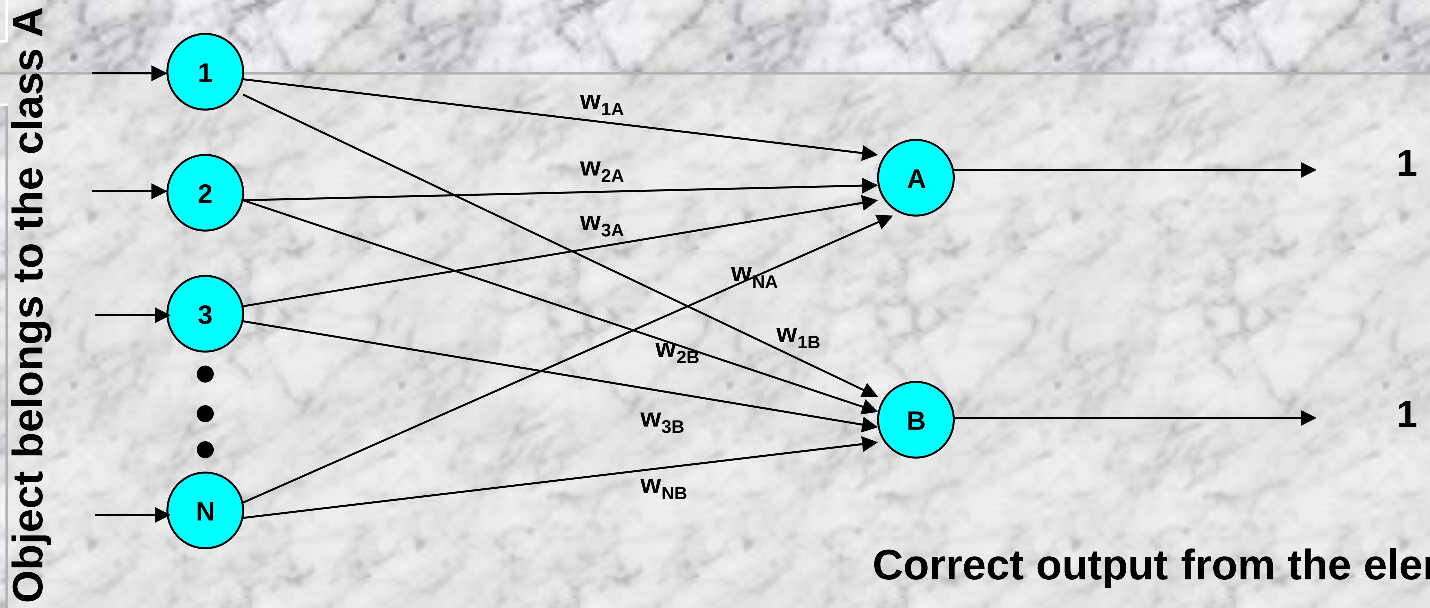
Input / Object	Output
Class A	1
Class B	0

The one-layer and two-elements Perceptron



Output / Object	Output A	Output B
Class A	1	0
Class B	0	1

Perceptron's learning



Object \ Output	Output A	Output B
Class A	1	0
Class B	0	1

Correct output from the element A

We do not change the weights incoming to the element A, w_{iA}

Incorrect output from the element B (1 instead of 0)

Input signal to B ≥ threshold value

It is necessary to decrease the weights incoming to the element B w_{iB}

Weights modification rule

Assuming

- **ζ = output (library) – output (real)**

than

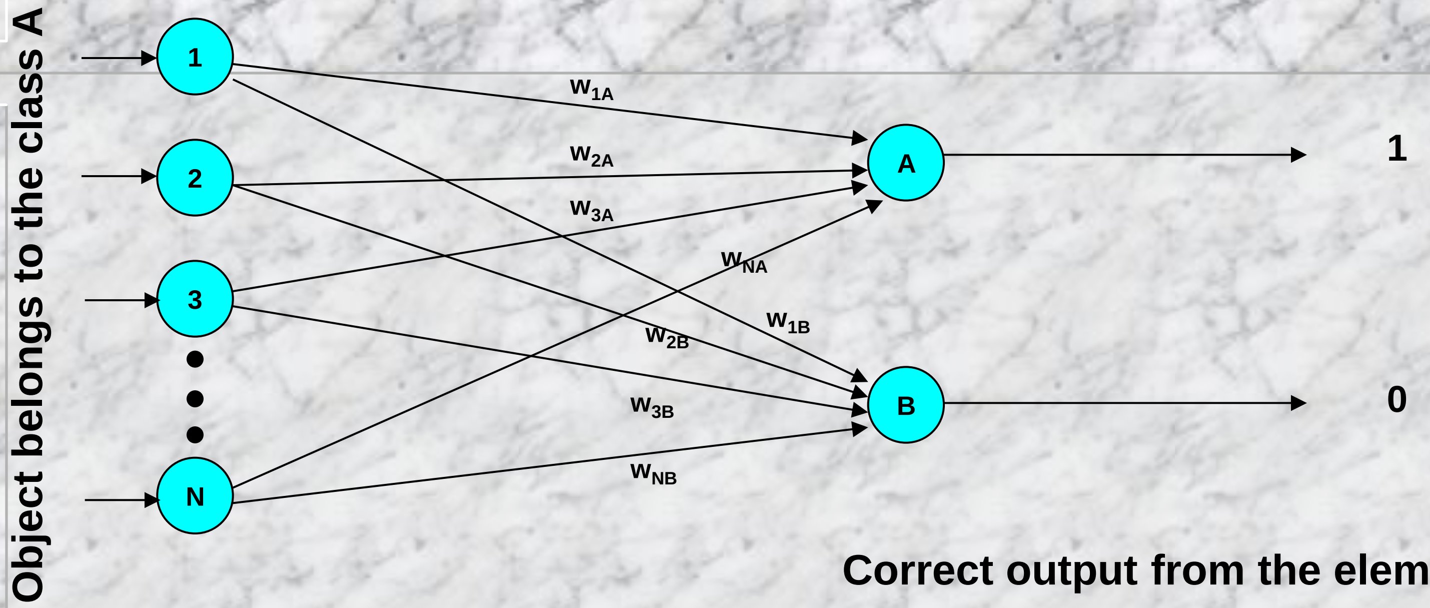
- **w_{iB} (new) = w_{iB} (old) + $\Delta\, w_{iB}$**

For example:

$$w_{iB}\,(new) = w_{iB}\,(old) + \zeta\, Input_i$$

$$Input_i = \begin{cases} 0 \\ +1 \end{cases}$$

Perceptron's learning



Output / Object	Output A	Output B
Class A	1	0
Class B	0	1

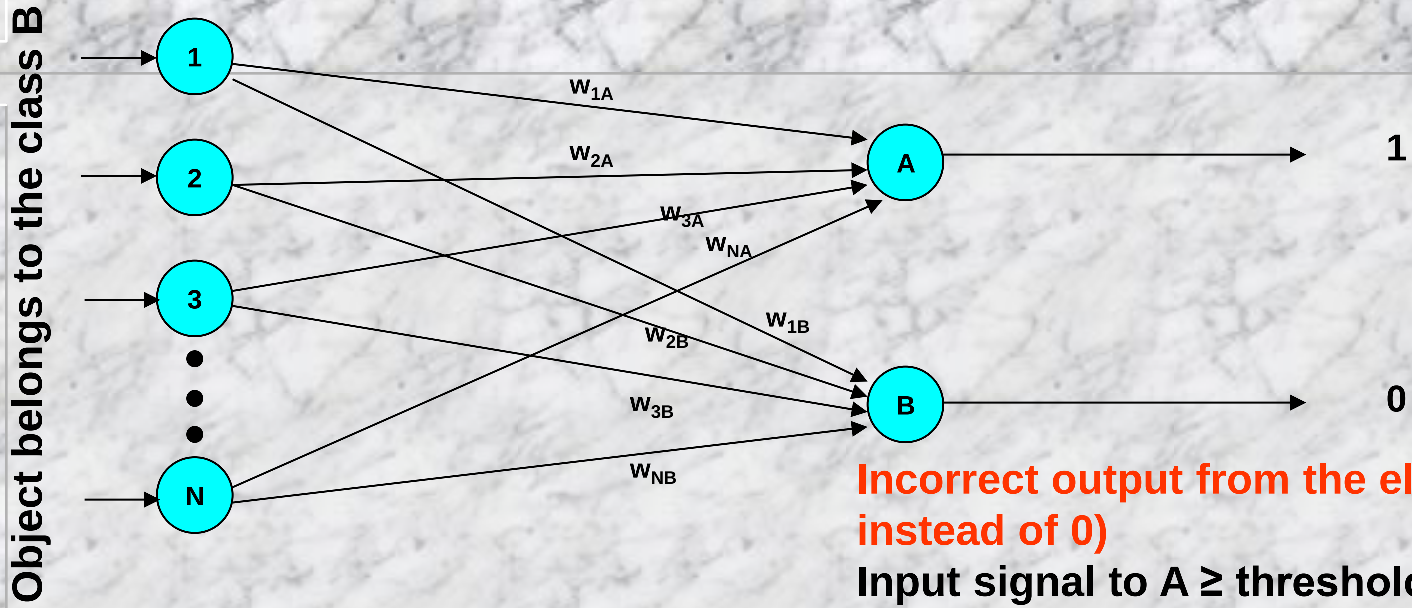
Correct output from the element A

We do not change the weights incoming to the element A, w_{iA}

Correct output from the element B

We do not change the weights incoming to the element B, w_{iB}

Perceptron's learning



Incorrect output from the element A (1 instead of 0)

Input signal to A ≥ threshold value

It is necessary to decrease the weights incoming to the element A, w_{iA}

Incorrect output from the element B (0 instead of 1)

Input signal to B < threshold value

It is necessary to increase the weights incoming to the element B, w_{iB}

Output / Object	Output A	Output B
Class A	1	0
Class B	0	1

The Perceptron learning algorithm

The perceptron learning algorithm

It can be proved that:

„ ... given it is possible to classify a series of inputs, ... then a perceptron network will find this classification”.

another words

„a perceptron will learn the solution, if there is a solution to be found”

Unfortunately, such the solution not always exists !!!

The perceptron learning algorithm

It is important to distinguish between the representation and learning.

- **Representation** refers to the ability of a perceptron (or any other network) to simulate a specified function.
- **Learning** requires the existence of a systematic procedure for adjusting the network weights to produce that function.

The perceptron learning algorithm

This problem was used to illustrate the weakness of the perceptron by Minsky and Papert in 1969:

They showed that some perceptrons were impractical or inadequate to solve many problems and stated there was no underlying mathematical theory to perceptrons.

The perceptron learning algorithm

Bernard Widrow recalls: *„..my impression was that Minsky and Papert defined the perceptron narrowly enough that it couldn't do anything interesting. You can easily design something to overcome many of the things that they proved' couldn't be done. It looked like an attempt to show that the perceptron was no good. It wasn't fair."*

XOR Problem

One of Minsky's and Papert more discouraging results shows that a single-layer perceptron cannot simulate a simple but very important function

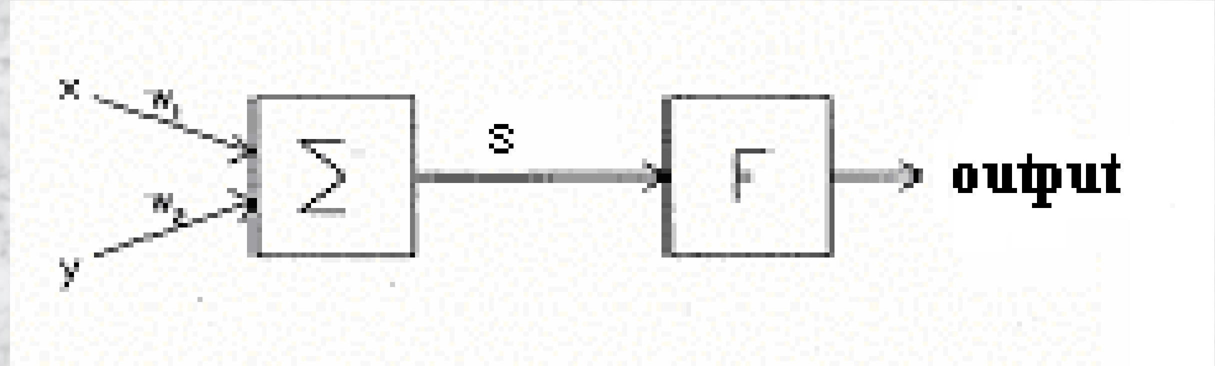
the exclusive-or (XOR)

XOR Problem

XOR truth table

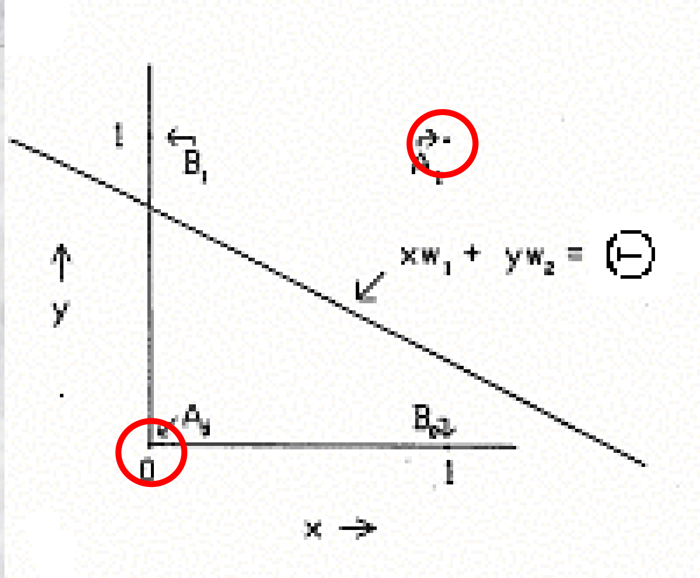
x	y	output	point
0	0	0	A_0
1	0	1	B_0
0	1	1	B_1
1	1	0	A_1

XOR Problem



Function F is the simple threshold function producing at the output signal 0 (zero) when signal s jest below Θ , and signal 1 (one) when signal s is greater (or equal) Θ .

XOR Problem



$$xw_1 + yw_2 = \Theta$$

Does not exist the system of values of w_1 i w_2, that points A_0 i A_1 will be located on one side, and B_0 i B_1 on the other side of this straight line.

Finally, what the perceptron really is ??

- Question, is it possible to realize every logical function by means of a single neuronal element with properly selected parameters??
- Is it possible to built every digital system by means of the neuronal elements??
- Unfortunately, there exist functions where it is necessary to use two or more elements.
- It is easy to demonstrate, that it is impossible to realize any function of N variables by means of single neuronal element.

Finally, what the perceptron really is??

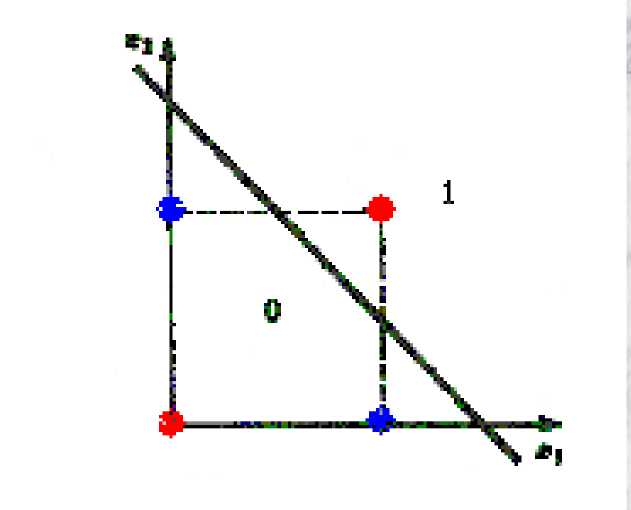
Geometrical interpretation of the equation

$$\sum_i w_i(t)x_i(t) = \Theta$$

is a plane (surface), which orientation depends from the weights..
The plane should be orientated in such the way all vertices, where output = 1 where located on the same side, i.e. the inequality will be fulfilled

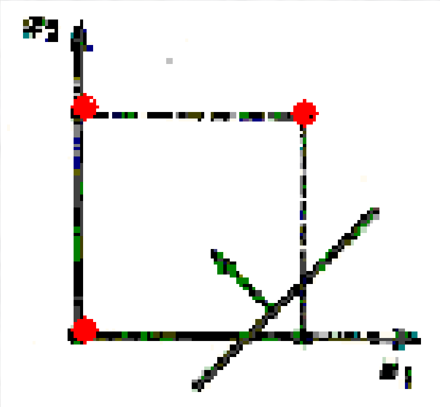
$$\sum_i w_i(t)x_i(t) \geq \Theta$$

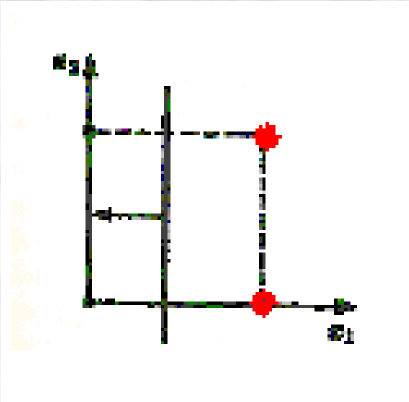
Finally, what the perceptron really is??

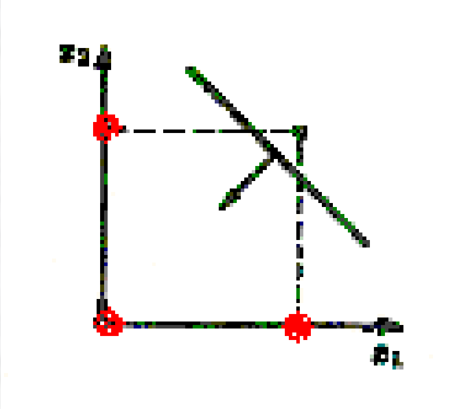


From the figure above it is easy to understand why realization of the XOR is impossible.
Does not exist the single plane (for N=2 – straight line) separating points of different color.

Finally, what the perceptron really is??







On these figures is the difficulties with the realization demanding the negative threshold values (n.b. the biological interpretation is sometime doubtful).

Linear separability

The problem of linear separability impose limitations fro the use of one layer neural nets. So, it is very important to knowledge about this property.

The problem of linear separability can be solved by the increase of the number of the network layers.