Neuroengineering (I)
1. Artificial Neural Networks

- **Scuola di Ingegneria Industriale e dell'Informazione** – Politecnico di Milano
- Prof. Pietro Cerveri

Neuroscience

How to represent brain abilities: memory, learning, reasoning, computing, ……

Biological basis: neuron, neuron dynamics, population dynamics, co-activation, ...

Dynamics with (sensorial) or without input (recall, dreams)

Biophysical/chemical modeling

when you interconnect units with a known behaviour you can achieve many different behaviour.

PARADIGMA:
EMERGIES PROPERTY

un comportamento che emerge differente da quello che avresti singolarmente al neurone

Engineering

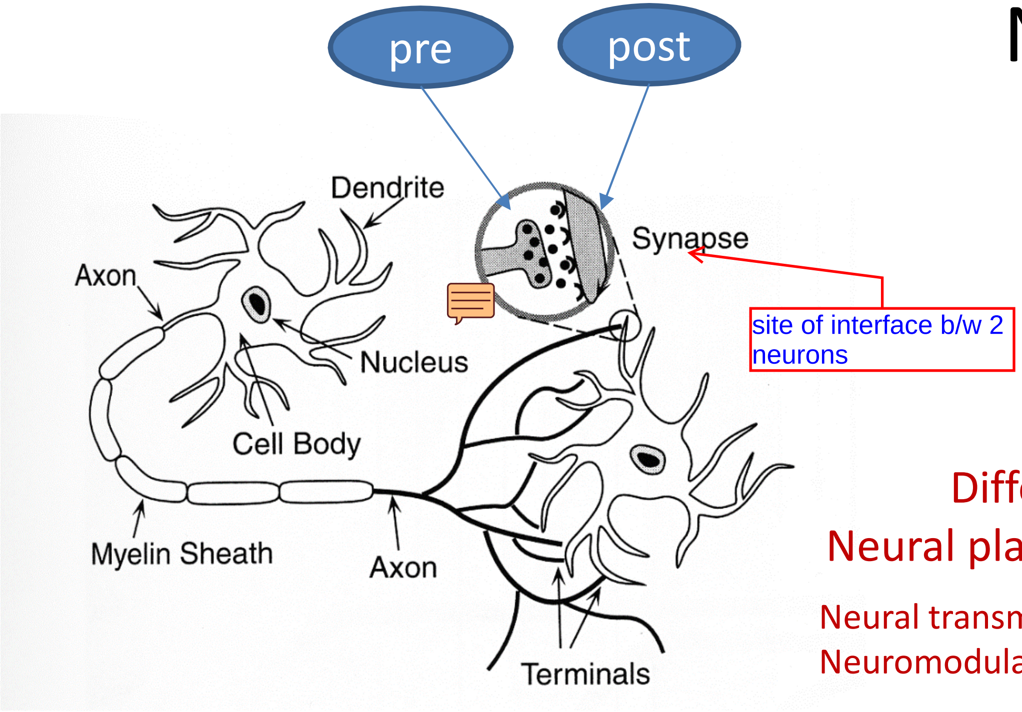
Neural inspired computational models:

- Classification functions
- Input/output mapping
- Function learning
- Control models
- Artificial memories
- Pattern synthesis
- Data dimensional reduction

Artificial Neural networks (ANN)

Sub-Symbolic computation

Neuron biological basis



Functional structure

Dendrite: post-synaptic signal transfer
Cell body (soma): input integration and output organization
Axon: pre-synaptic transfer

Different type of neurons (~10^{11} neurons)
Neural plasticity and learning (~10^{14} interconnections)

Neural transmission: single synaptic transmission
Neuromodulation: multiple synaptic transmission into a brain area

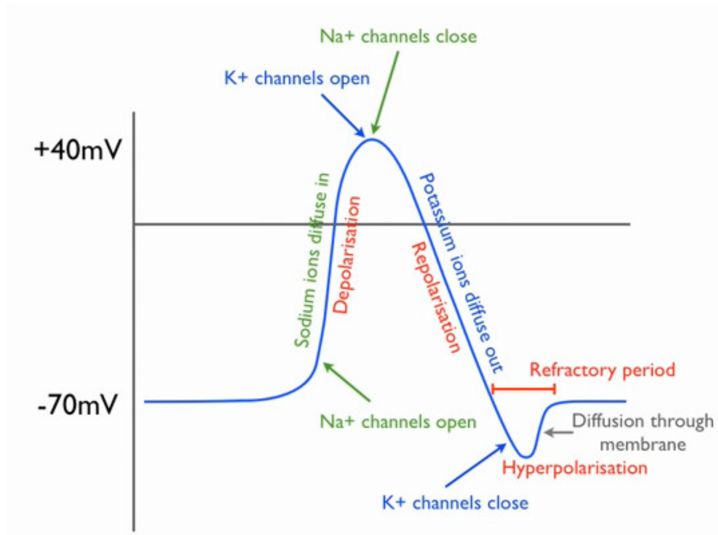
Synapses, the structures at the interface between two neurons, are constantly modified by neuronal interaction

Biochemical signals: neurotransmitters (~10^2) in the pre-synaptic neuron (amino acids, monoamines, peptides, …)

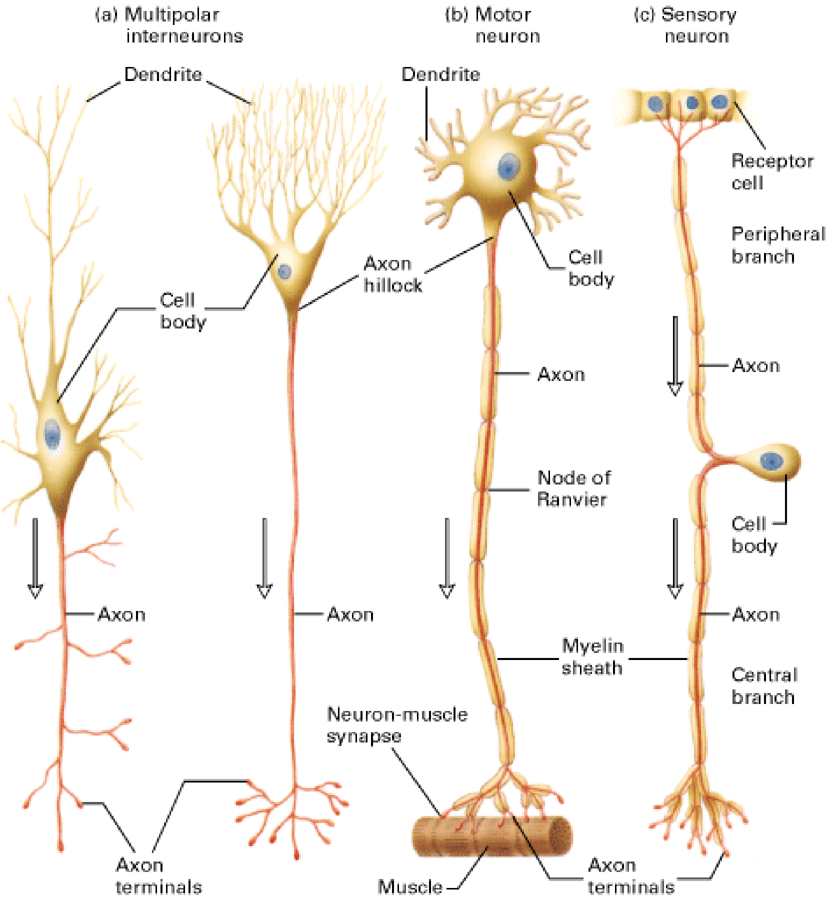
Receptors in the post-synaptic neuron

Excitatory or inhibitory influence

Multiple connection: neuron is firing (electric signal) when the total of excitatory influences are greater than those of inhibitory influences



Neuron diversity



These different characteristics can be implemented into artificial models of the neuron.

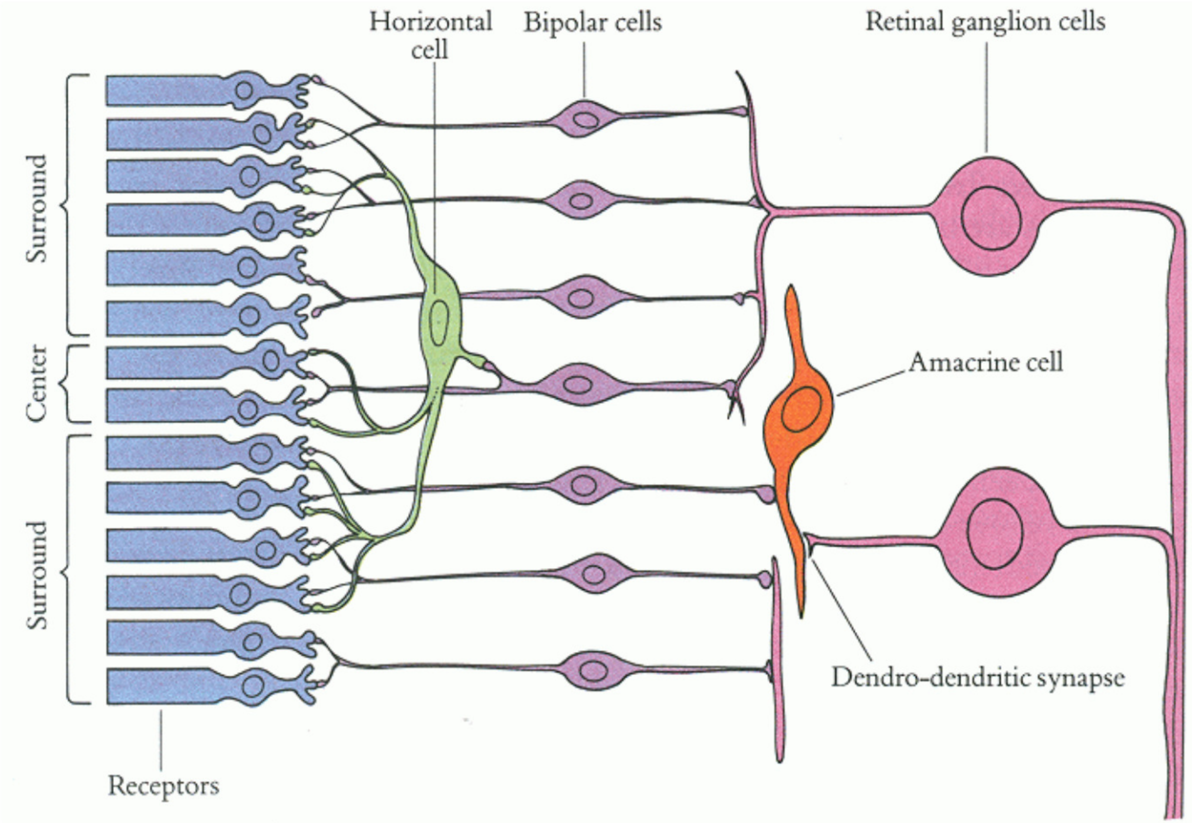
Sensory neural cell (bipolar)
Motor neural cell (unipolar)
Interneuron (multipolar, unipolar)

Difference in the neurotransmitters

Way of connections between cells: signals connection, like, linearly propagating, but we can have also "orizontaly" propagation, like it can be exchanged in the neuron of the same layer and then passed on. At the beginning we have many neurons, and as we move to the end we can see easily that the number is decreasing. So we can probably say that the info is like compressed, or anyway processed. Like I'll have the same patter that can be described by less signals. So, part of the processing is compressing and extracting the main component (like PCA)

Neural connections

RETINA

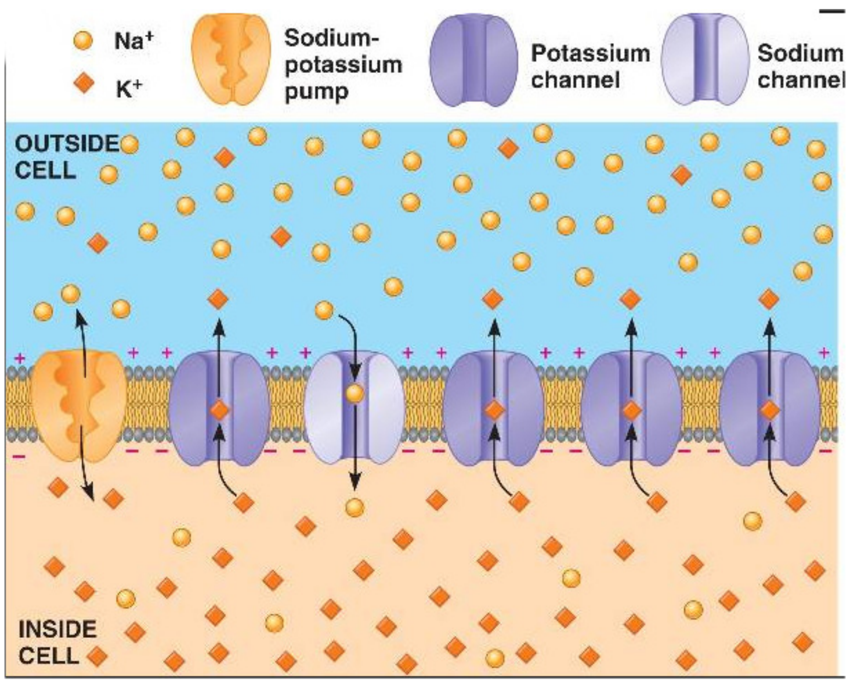


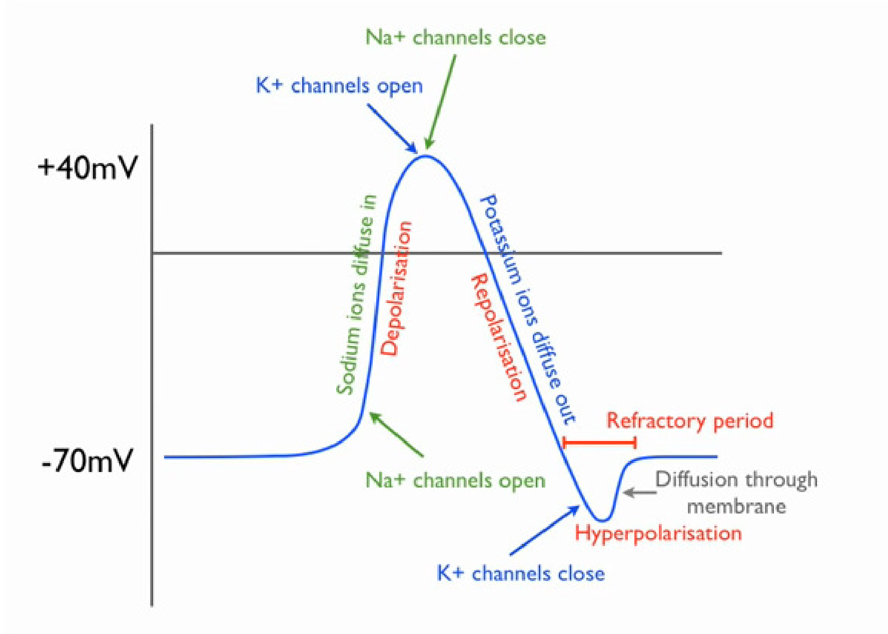
Signal processing
Signal integration
Signal compression

Optical nerve

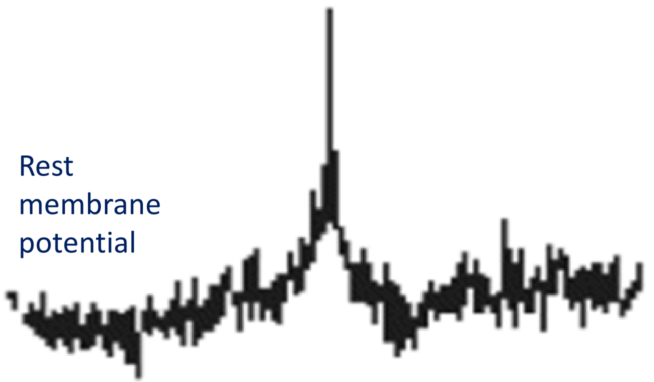
no

Biochemical dynamics of the neural cell





$$E = \frac{RT}{F}\ln\left\{\frac{P_k[K]_o + P_{Na}[Na]_o + P_{Cl}[Cl]_i}{P_k[K]_i + P_{Na}[Na]_i + P_{Cl}[Cl]_o}\right\}$$



Goldman, Hodgkin, Katz equation

P_i is the permeability of the cellular membrane to the i-th specie

At rest, K ions rule the voltage

no

Biophysical/chemical modeling approach

– **Hodgkin–Huxley model (single neuron)** Conductance-based model

- **Nobel price Medicine 1963**

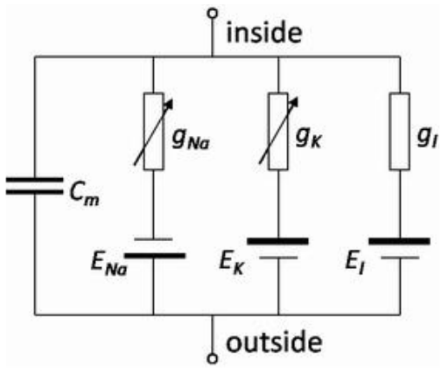
$$I = C_m \frac{\mathrm{d}V_m}{\mathrm{d}t} + \bar{g}_K n^4 (V_m - V_K) + \bar{g}_{Na} m^3 h (V_m - V_{Na}) + \bar{g}_l (V_m - V_l),$$

$$\frac{dn}{dt} = \alpha_n (1 - n) - \beta_n n$$ Potassium channel activation

$$\frac{dm}{dt} = \alpha_m (1 - m) - \beta_m m$$ Sodium channel activation

$$\frac{dh}{dt} = \alpha_h (1 - h) - \beta_h h$$ Sodium channel inactivation

α_i and β_i are rate constants for the *i*-th ion channel, which are voltage dependent only
l: leak current that consists mainly of Cl^- ions (voltage independent channel)

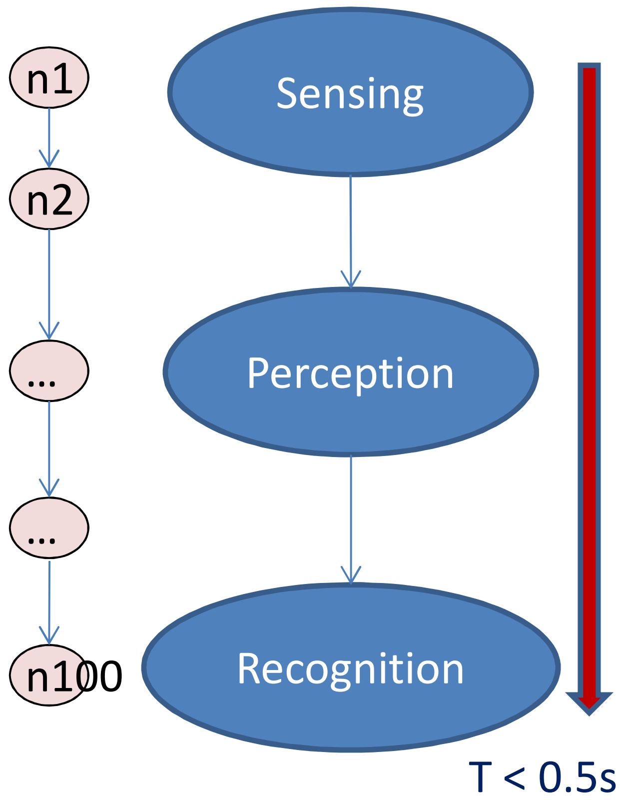


Equivalent circuitry for neuronal membrane

Time requested to process the neural signal: it can be easily demostrated that in recognition task (we see for example a face) it's very fast. This very short time requested implies two explanation:
1) there's only one line of processing. But it should be very fast, and it's not compatible with the time recorded. We should also take in consideration the max freq of the neuron, (e.g. 100hz for some neurons), this means there can't be a single line, there must be many lines that are grouping the info towards the high level center of processing. And indeed it's how it works: the brain is a parallel computed. This characteristic is mapped on the network in the pc.

Neural transmission and computation speed (PARALLELISM)

- Duration of the spike: few milliseconds (~5ms)
- Spike speed about 25 m/s (but it depends on axon diameter and myelinated fibers - typically tenfold)
- Frequency: from few Hz up to some hundred Hz
- Face recognition is usually performed in few hundred of milliseconds
- <u>Hundred-step rule</u>: the fastest computation is less than half a second so that the neural processing cannot take more than about 100 serial steps
- High parallel signal transmission
    - Neural transmission is about few bits
    - Information is actually distributed throughout connections



no

	Von Neumann computer	Biological neural system
processor	Complex High speed One or a few (now many actually)	Simple Low speed A large number
memory	Separated from processor Localized Noncontent addressable	Integrated into processor Distributed Content Addressable
Computing	Centralized Sequential Stored programs	Distributed Parallel Self-Learning
Reliability	Very vulnerable	Robust
Expertise	Numerical and symbolic manipulation	Perceptual processing
Operating environment	Well-defined, well constrained	Poorly defined, unconstrained

Anoter feature of the brain is the ability to learn: it's more effective if the changes are more "stable", stability of learning. We have sort of ability of being "ready to learn" true even more for kids. It's related to what is called plasticity of the human brain. You can think of a network with connections not specifically define, at the beginning all the neurons are connected with all the other neurons, sort of fully connections, and learning basically means to prove (remove?) some connection and reinforce some other connections. The ability to learn is related to repetition of observations, it's a way to reinforce. The same input is given (feeded) to the network more than just one time.This way storing lasts a lot of time. Important: the learning is not changing the topology of the network! The number of neurons, the interconnection, are not changed during the process: what is changing is e.g. the weights associated with each entering signals to the network, or for example the threshold of the neuron. We are not "cutting" the axons! What your brain is doing is just acting at a biochemical level on the connections of the axon, but also on the soma: it could be "more effective in getting the signal".

Learning concept

"Learning is a process by which an activity/ability is either originated or modified by reacting to a situation, unless the characteristics of the activity change cannot be explained on the basis of innate response tendencies, current organism states."

Learning is not pre-programmed: external stimuli are mandatory

At neural level: learning means increasing of the efficiency of the neural signal transmission (synapsis)

At population level: dynamic coherence (principle of synchronism) between neural signals strength the interconnection among neurons

Reinforcement learning: repeated stimuli facilitate learning (experience/practice improve learning)

Learning does not change the topology of the neural structure
Learning modifies the neural connections

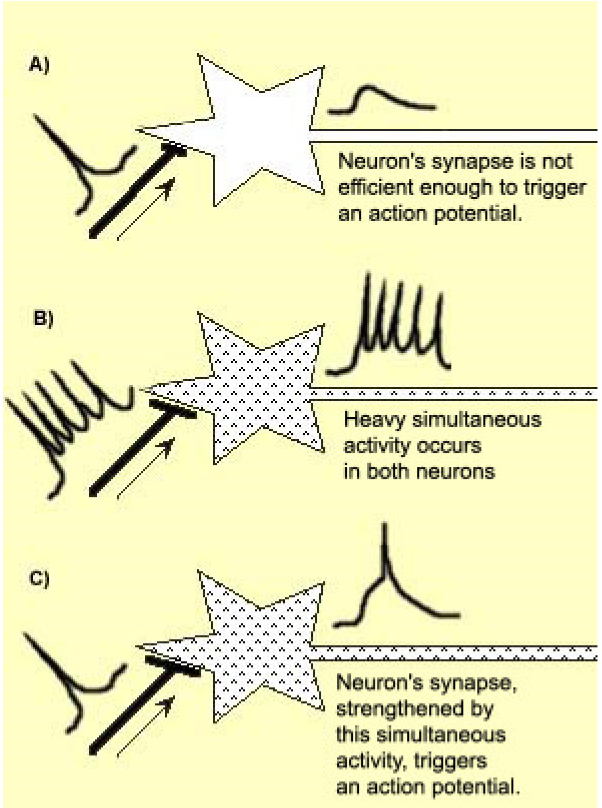
We can say that every sinapse has an efficiency of transducing the stimulus to the PSP: From the propagation of AP, release of neural transmitter, moving into the cell, generating a depolarization. This exchange of energy has a sort of "gain". I start with a signal with a power of 0.1, with an output of 1,it means that i treat the same input with a signal of x10. This means that the learning has increased the efficiency. The interface between the axon and the dendrite is not "fixed" that interface can for example incraese the distance (helped by special proteins that can cementing in case of learning). Cementing synaptic interface is like building a preferential pathway. In biology it's called point of memory. That transmission is now blocked, if I'm enable to reinforce the learning, the cementification becomes more effective.

Neural plasticity and learning

Learning depends on the plasticity of the circuits in the brain - the ability of the neurons to make changes in the efficiency of their synaptic transmission

The brain can thus be said to store information in networks of modified synapses (the arrangement of which constitutes the information) and to retrieve this information by activating these networks

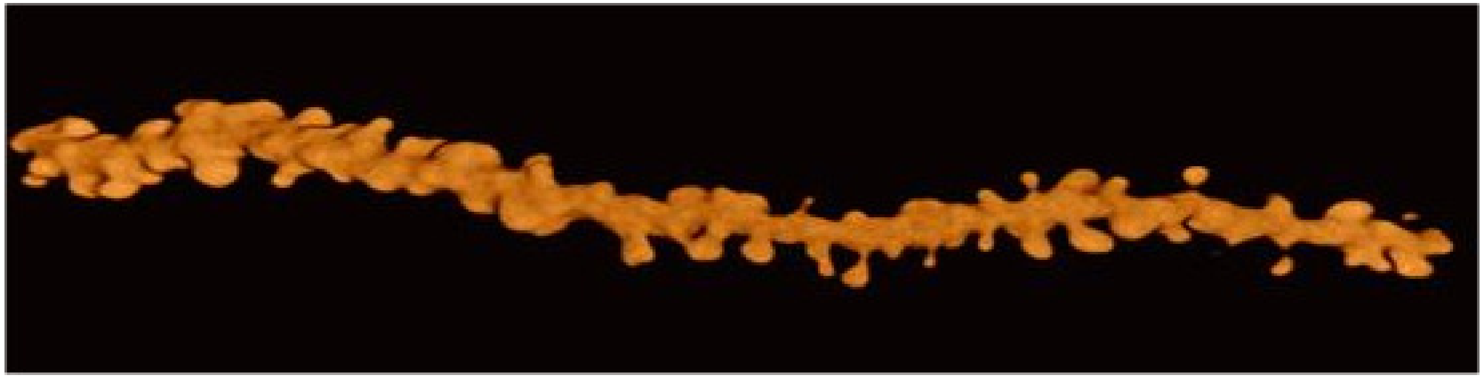
Multiple inputs
Synchronous inputs
Short-term plasticity
Long-term plasticity
Excitation vs Inhibition effects



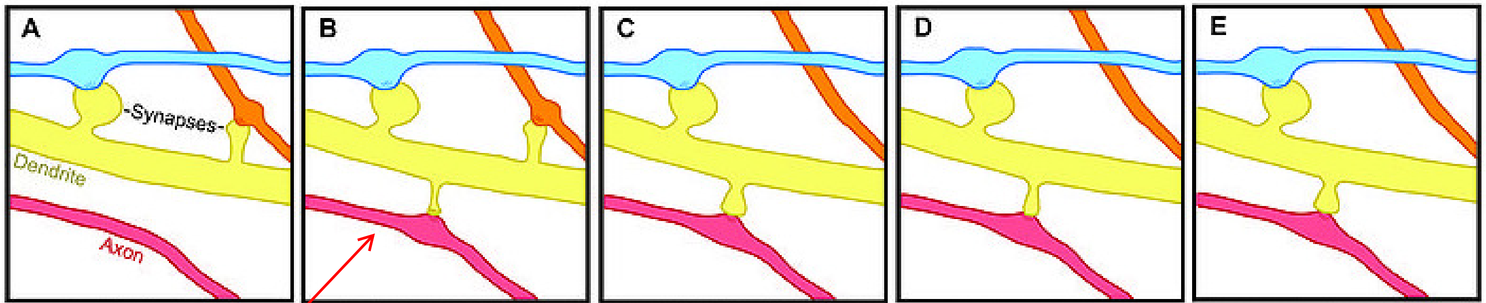
"cells that fire together wire together”

We know that during the youth we have a lot of interconnections that are not specialized, and during learning some connections are reinforced and others are pruned (removed) there's a sort of TURNOVER of the synaptics interfaces.

Dendritic spines



Cells sending electrical impulses encourage receiver dendrites to grow, over time, towards the stimulus and eventually create new pathways (emerging dendritic spines) for improving the efficiency of the message transmission (for the future)



Original situation | New experience/learning | Prolonged experience/training | Experience removed | Repeated experience/relearning

agreement wiith anoter neuron, the axon has grown a synapse to the third neuron

When we were born we a huge number of interfaces and during youth this number increased. So learning, in contrast of what we could expect is a DEPLATION of connections! Lots of connections are removed, the network are specialized. I'm learning for example motor control, a sort of program. When you learn there's no need to consciously thinkly on how to act with your muscles. This is away to have a memory and access to it without request. This introduces DIFFERENT TYPES OF MEMORY:
1) conscoius (what did I study yesterday?) requires a conscious action
2) unconscious (like walking)
These have the same neural base, a network that at the beginning is very shallow, and learning means to specialize, and cutting connections (pruning your tree).

Dendritic spines: learning

Dendritic spines are motile and dynamic structures that undergo a constant turnover, even after birth

Long-term memory is mediated in part by the growth of new dendritic spines (or the enlargement of pre-existing spines) to reinforce a particular neural pathway

In youth, dendritic spine turnover is relatively high and produces a net loss of spines, with the rate of the elimination surpassing the rate of the formation

but… evidence for loss of dendritic spines as a consequence of aging

Numerous brain disorders are associated with abnormal dendritic spines

Cocaine and amphetamine use have been linked to increases in dendritic branching and spine density in the nucleus accumbens (pleasure, fear, aggression, impulsivity)

Refreshing memory is a way to maintain memory and dont forget. With aging this ability is progressivly lost, this means that indeed very old people are able to recall very old memories, but they are not able to easily recall stuff happened some days ago. This means that the memories that have been acquired in the youth are more stable than the one acquired when old. This is the core of a lot of pedagogist approaches to learning to young people.

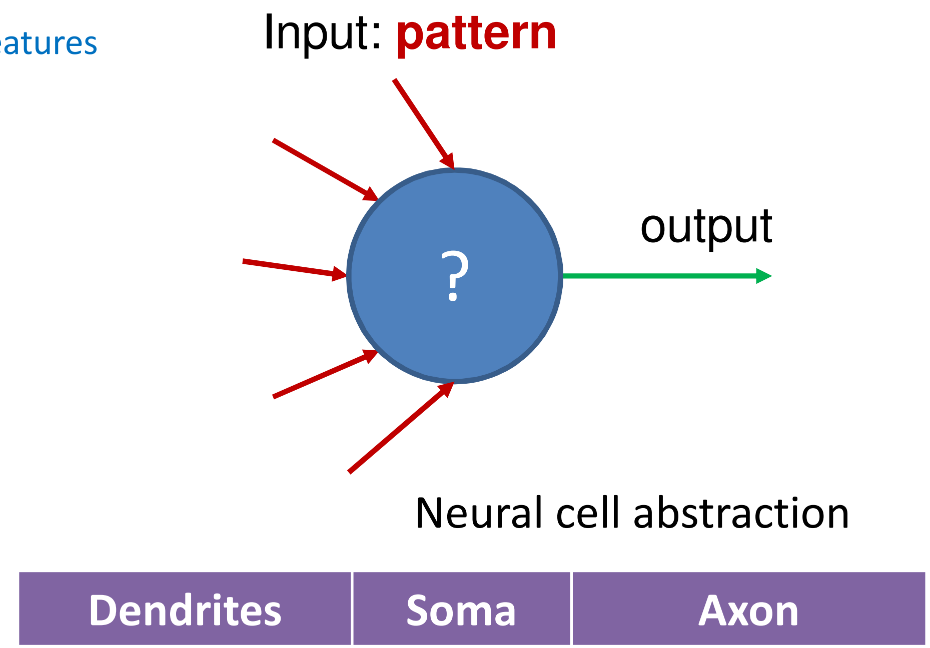
Rappresentation of a neuron: circle that get inputs and produces an output. Signal that are monodimensional: a number. The grouping of the input is called pattern. The pattern is the mathematical vector rappresentation of the input. PATTERN -> VECTOR OF NUMBER. So, our mathematical numbers undergo an integration, the simplest is a summation, to be more precise we can assume that each entering line is weighted by a factor (called WEIGHT). So it's performing an integration by way of sommation. This is the equivalent to the action potential. Which has to be compared to a threshold of the neuron. Matematically means meaking a difference. This way the weights and threshold are sort of "structural parameter" parameters of the neuron. If I change these I can change the behaviour of the neuron. This is the first way of changing the behaviour. Then I'm taking the result and it's mapped by what is called activation function. It's a way of processing the action potential. It's a simple function. But from a biological point of view, not all the function are related to the ability to simulate the saturation of the amplitude of the spike, the time step between two spikes. What I see is not a different maxima of the output, but a distance between two spikes, which has a maximum. *The fractory period*

Modeling approach of a neuron (STRUCTURE)

Information processing point of view

Input/output mapping

- Input signals
  - ❑ PATTERN: multi-dimensional vector of features
- Input signal processing
  - ❑ INTEGRATION
- Action potential
  - ❑ WEIGHTED SUM
- Activation threshold
  - ❑ COMPARISON
- Activation function
  - ❑ MAPPING TO OUTPUT



The output is not free to achieve any value that it wants, if biologically we have a saturation of frequency, here we have a saturation of the output value. The math neuron is not behaviour like a biological neuron! If I'd like to maintain a sort of corrispondace b/w the two I must have an output with a saturated output.

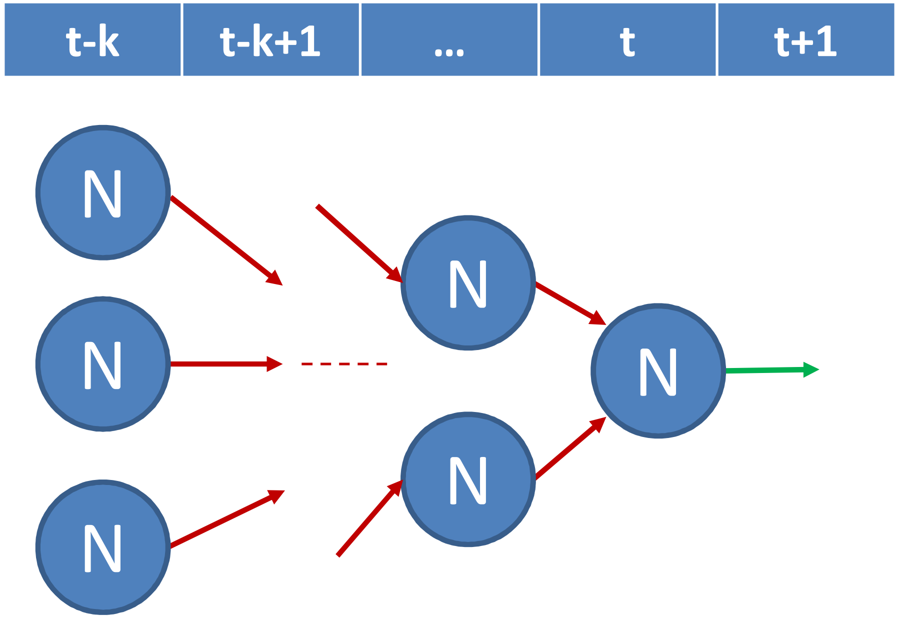
I must also consider that a biological neuron changes the output in time! Time by time, step by step it's changing the output. We can assume that a neuron, or set of, has a sort of dynamic. If I define a temporal direction (past towards the future), and I consider a discrete time (because it's easier to implement). I can assume the input is entering at time t, and the output is produced at time t+1 (so it's integrating at time t). When we consider the properties of an overall network I can consider a set of neurons that are producing the output at a certain time tx. It's not easy to understand the output of a complex network, we must feed it and see the output. relationship of EMERGENT PROPERTIES: a priory is hard to know the behaviour of the network, I have to see it working.

Modeling approach of a neuron (DYNAMICS)

Information processing point of view

Input/output mapping

- Temporal dynamics
  - Discrete time
  - Synchronism/Asynchronism
  - Input at time t, Output at time t+1
- Neuron interconnection
  - NEURAL NETWORKS



Multiple interconnected neurons reframe a network featuring properties that cannot extrapolated from the single neuron function

EMERGENT PROPERTIES

Learn:
1) tune the parameter
2) compute the weights and threshold
do not touch the topology

If I define specific connections the learning is not allowed to change or shut them. The lerning is achieved from observations. I must feed the network and look what happens.

Implement learning:
1) supervised learning
2) unsupervised

Artificial Neural Networks

2-component system

1. Set of primary units (neurons, nods, units) each one endowed with input and output lines. Output is triggered by an activation function
2. Set of interconnection lines (channels) among such units: each line is characterized by a number (weight or connection coefficient or synaptic effectiveness) which accounts for the relevance of the signals

Dynamics: discrete time scale

- It allows observing the state of the system only at time t, t+1, t+2,..
- Given the potential at time t, a specific rule provides the potential at time t+1

Learning the network parameters

- From observations (multiple training data)
- With or without supervision (reference output for training input)

Classification: associate a pattern to a class. It can be binary or multiple classification. Function approximation means that I'm entering with a domain of my variable, and I'm outputting the codomain of the variables. I don't have the analitycal function, lke y=x2+3. I'm indeed searching y=f(X) I don't know the real formulation though! We have just a sample of this function in some points. So, domain is the input, codomain the output.
Data synthesis: I'm entering with a dimension of the pattern, and I'm outputting a pattern with a lower dimension. I can also implement a network for forecasting. The input would be the past history of the value, and the output is the future. Here we don't have a math formulation but a network of neurons.

Typical applications of ANN

- *Pattern* classification and recognition

$l = f(\boldsymbol{x}) \qquad \boldsymbol{x} \in X \subset R^h \quad l \in C \subset N$

- Function approximation

vector $\boldsymbol{y} = f(\boldsymbol{x}) \qquad \boldsymbol{x} \in X \subset R^h \quad \boldsymbol{y} \in Y \subset R^k$

- *Data synthesis and compression*

vector $\boldsymbol{h} = f(\boldsymbol{x}) \qquad \boldsymbol{x} \in X \subset R^h \quad \boldsymbol{h} \in H \subset R^k$ *k<<h*

- Time-series forecasting

$\boldsymbol{x}(t) = f(\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \boldsymbol{x}_{t-3}, \dots)$

f *is unknown*

Neural network contributes

1943 – McCulloch and W. Pitts
- Formal neuron

1949 – Hebb
- Plasticity of the connections
- Synaptic strength and reinforcement

1958 – F. Rosenblatt
- Supervised learning
- Perceptron

1960 – B. Widrow and M. Hoff
- Layered networks

1969 – M. Minsky and S. Papert
- Limits of the perceptron as linear classifier

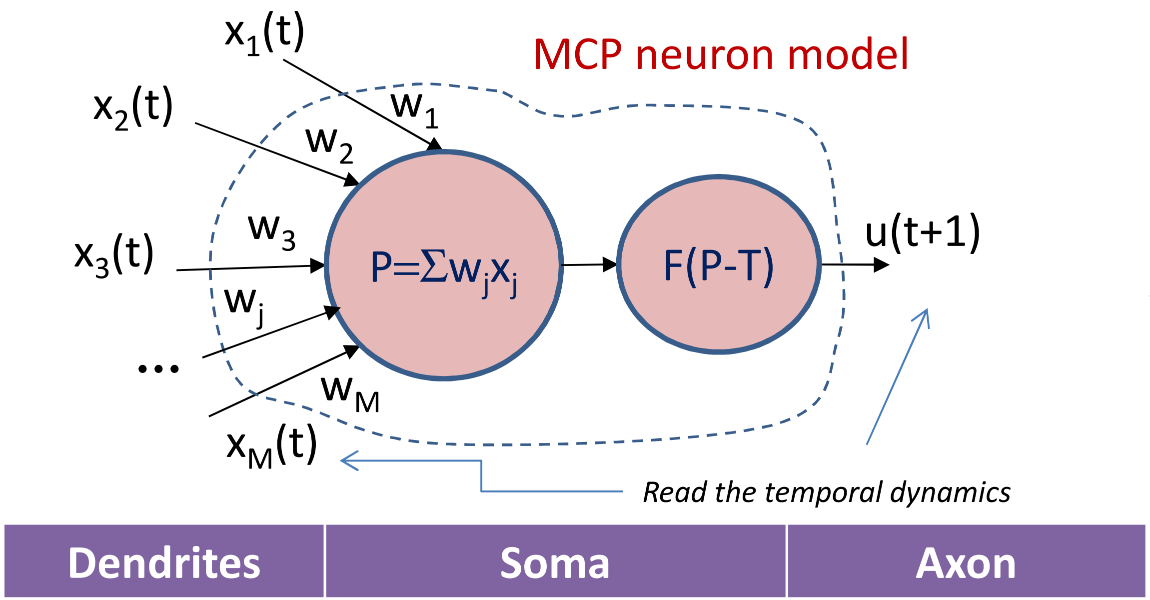
1982 – J. Hopfield
- Auto-associative memory
- Connectionism: isomorphism between any structured knowledge representation and an opportune neural network

1982 – D.E Rumelhart
- Back-propagation principle overcomes perceptron limits
- General learning ability

First model. We have the integration, the AP is the sommation of the input, to each line we have weights. Positive weight by convention are excitatory contributions, negative the opposite (inibitory contributions). Then the AP is compared with the threshold, the result is applied to the activation function. Simplest way to use this method: process binary signals, producing binary outputs. Simplest function to have a 2 state output: heavy side function. Also called step function (gradino).This is how a neuron can behave like a logic port.

Neural network genesis

- 1943 - McCulloch and Pitts (MCP)
  - Mathematical model of the cerebral cortex based on binary threshold units (formal neurons)
  - A network of formal neurons can realize any Boolean function
  - The network is equivalent to an automata (finite state machine)



w_j : Weight

>0 Excitation

<0 Inhibition

x_j : state of the input neurons

u : state of the output neuron

F: activation function

T: activation threshold

P activation potential P=f(wj,xj)

Threshold binary unit (MCP)

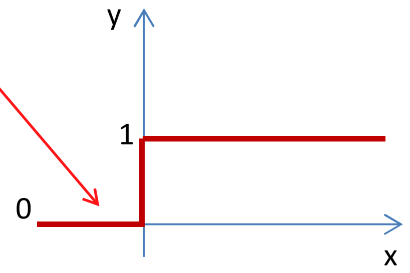
- *T* is real number named threshold of the unit
- P(t) action potential at time t

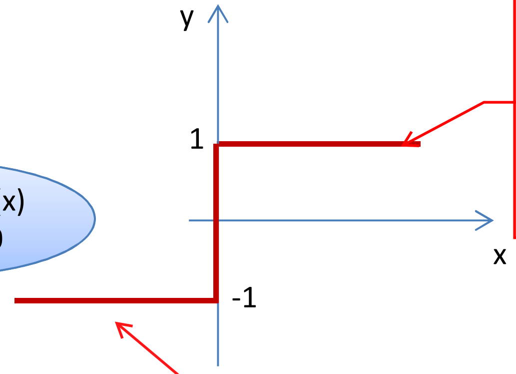
$$P(t)=\sum_{j=1}^{M} w_j x_j(t) = w_1x_1(t)+w_2x_2(t)+w_3x_3(t)+..+w_Mx_M(t)$$

- u(t+1) is the output with
  - u(t+1) = F(P(t)-T)
    - F is the Heaviside (step) function
      - F(x) = 1 if x>0 ,
      - F(x) = 0 if x<=0
  - u(t+1) =sgn(P(t)-T)
    - sgn(x) is the signum function
      - sgn(x) = 1 if x>0
      - sgn(x) = 0 if x=0
      - sgn(x) = -1 if x<0

Actually sgn(x) = -1 if **x<=0**

input neglected





I can use the sign function to have -1 and +1. It's just a matter of convenience and convention.What can change the behaviour is choosing one or the other if I'm considering a network! OFC if the input is 0 then it's neglected no matter what the Wj, while if the input is -1 it's considered!

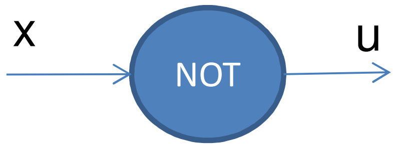
input considered

Boolean units

- Boolean networks
  - Signals are 1 (TRUE) or 0 (FALSE)
  - OR, AND, NOT units
  - The weight setting discriminate the behavior of the unit

NOT

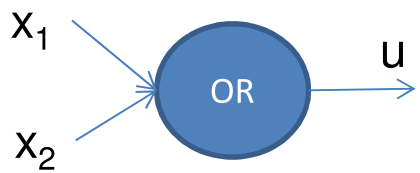
x	u
0	1
1	0



$u(t+1) = F(w*x(t)-T)$

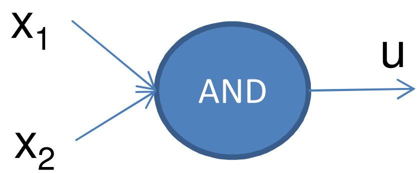
OR

x_1	x_2	u
0	0	0
0	1	1
1	0	1
1	1	1



AND

x_1	x_2	u
0	0	0
0	1	0
1	0	0
1	1	1



$u(t+1) = F(w_1 * x_1(t) + w_2 * x_2(t) - T)$

These can be tailored just changing the weights and threshold.

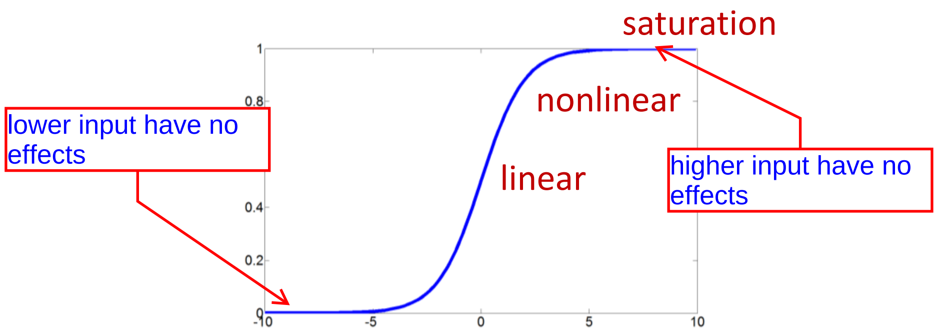
I can easily extend the span of neuron by changing the output function, the activation function. I can modify and use some boundary functions, like a logistic one. In that case the output of the neuron is continous.

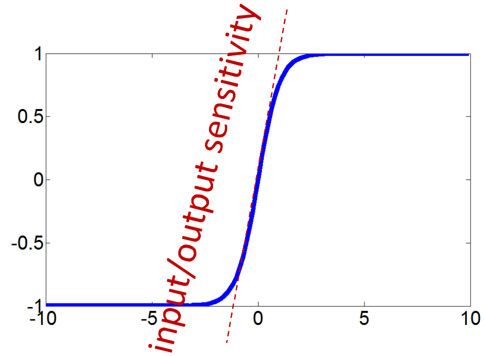
Sigmoidal output

The output can assume any real value between 0 and 1 or -1 and 1

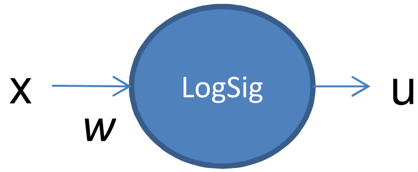
$$u(t+1)=\frac{1}{1+e^{-(P(t)-T)}}$$

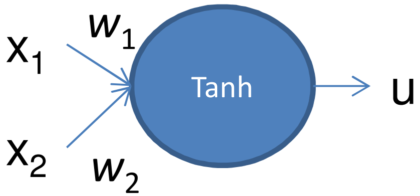
$$u(t+1)=\tanh\left(P(t)-T\right)=\frac{e^{(P(t)-T)}-e^{-(P(t)-T)}}{e^{(P(t)-T)}+e^{-(P(t)-T)}}$$

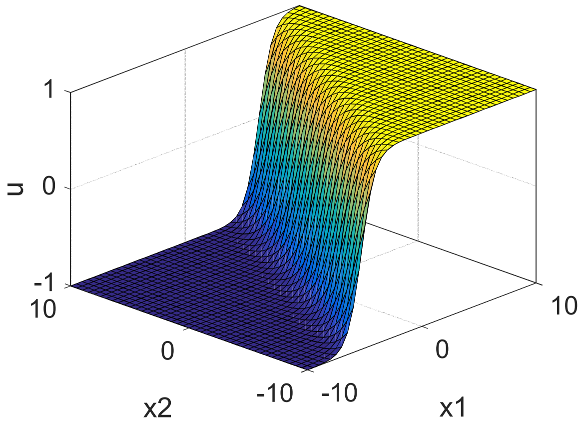




The derivative is constant! this is important

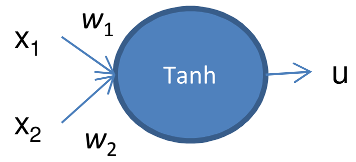




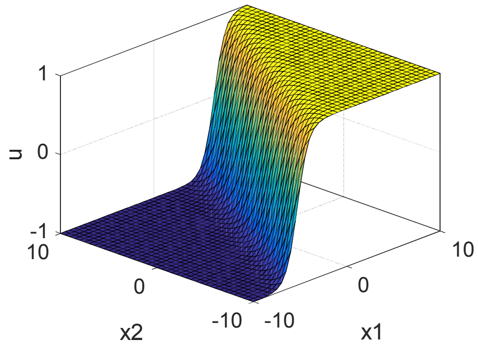


Changing the shape of the activity, and changing the weights and threshold. Every configuration correspond to a different behaviour of the neuron.

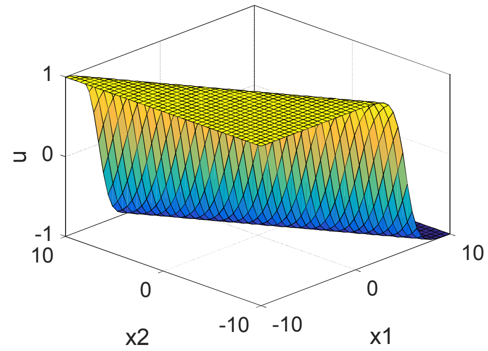
Example with TANH



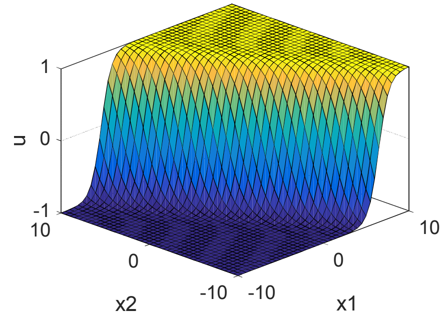
w1 = 0.85; w2 = -0.5; T = 0.5;



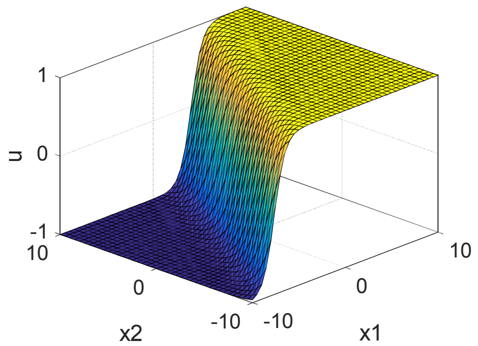
w1 = -0.85; w2 = -0.5; T = 0.5;



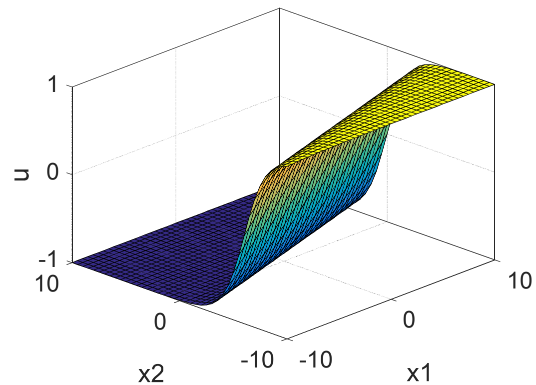
w1 = 0.85; w2 = 0.5; T = 0.5;



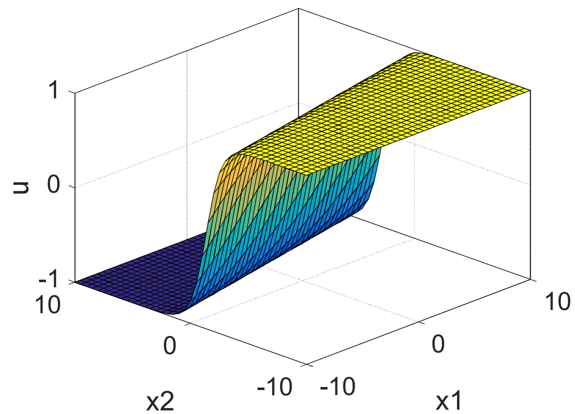
w1 = 0.85; w2 = -0.5; T = -1.5;



w1 = 0.15; w2 = -0.5; T = 1.5;



w1 = 0.15; w2 = -0.75; T = -0.5;

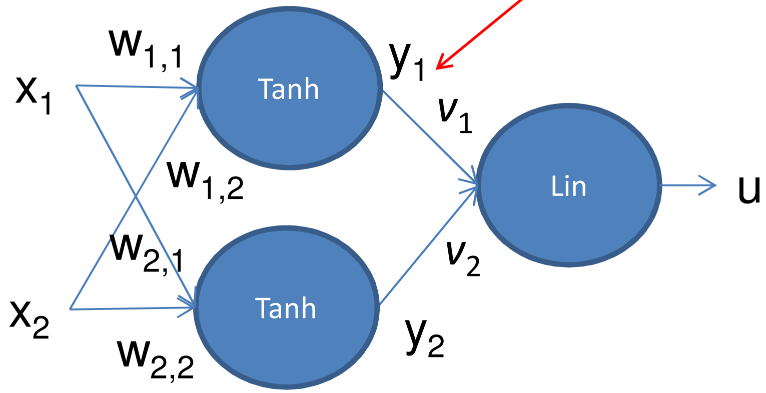


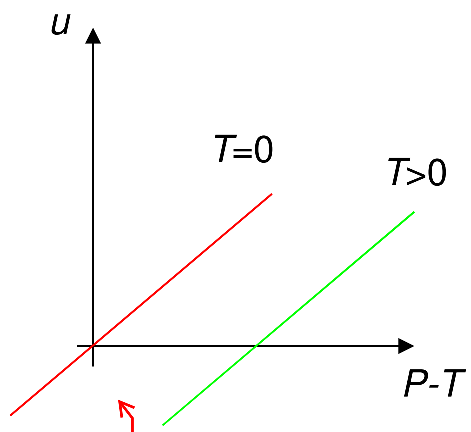
Linear units

NL modifications

- The output can assume any real value (unbounded)
  - $u(t+1)=P(t)-T$
  - With or without threshold

The two NL modifications are then linear combined and modified





I can consider also a linear activation function. Just a linear mapping of input and output.You can just tune the position of the linear function to respect the threshold. These kind of units have no biological correspondance. There's no saturation. It's still interest because we can build a funcational construction.

$$u = v_1 * y_1 + v_2 * y_2 - T_3$$

$$y_1 = \tanh(w_{1,1} * x_1 + w_{1,2} * x_2 - T_1)$$

$$y_2 = \tanh(w_{2,1} * x_1 + w_{2,2} * x_2 - T_2)$$

If I use infinite functions I can see that I have a sommation of ininite same functions but with different coefficients... LIKE FOURIER WITH SINS

Neural connections in ANN

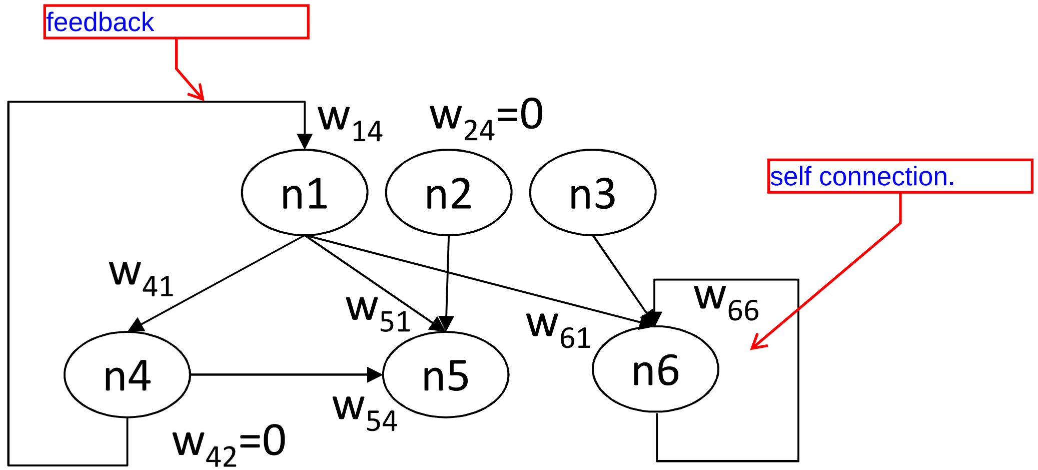
- Networks without spatial localization
- Networks with spatial localization

No spatial localization: the position of the neuron has no effect. Instead with spatial localization the physical distance of two neurons affects the efficiency of the connection.

Networks without spatial localization

- w_{ij} connection weight between the j^{th} (input) and i^{th} (output) neuron
- Given that $w_{ij}=0$, it does not exist a connection between units j and i
- $w_{ij}>0$ implies that unit j excites unit i
- $w_{ij}<0$ implies that unit j inhibits unit i
- w_{ii}: self-connection

- Interconnection matrix $\{w_{ij}\}$ for a network
- $\{w_{ij}\}= \{w_{ji}\}$: symmetric network
- $\{w_{ij}\}= \{-w_{ji}\}$: anti-symmetric network



It's not fully connected network (all the neuron must be targeting all the other neurons. Here like n5 is alone.

Networks with spatial localization

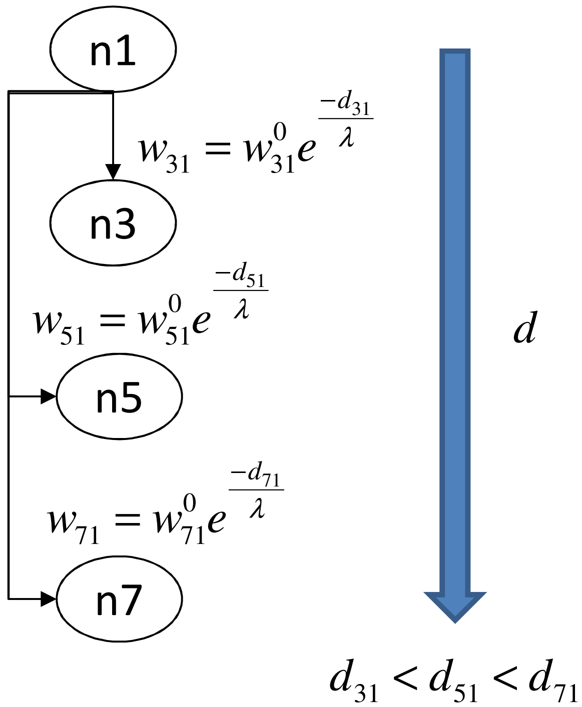
- w_{ij}^{0} does not depend on distance

$$w_{ij} = w_{ij}^{0} f(d_{ij})$$

$$f(d_{ij}) = e^{\frac{-d_{ij}}{\lambda}}$$

$$\lambda > 0$$

Neurons are allocated spatially into a metric domain

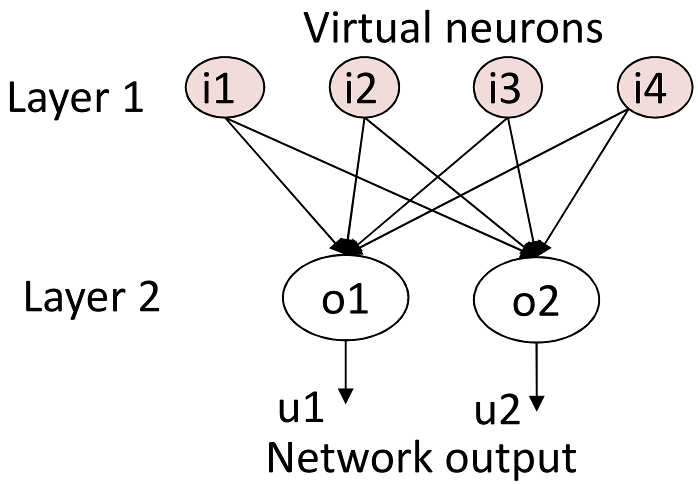


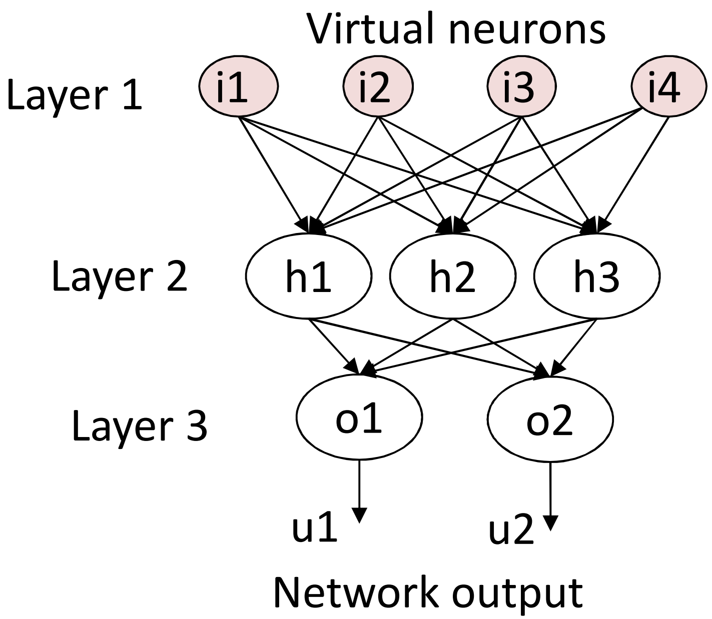
Considering the effect of distance: I can allow to have an effect on the weights. We have for example a metric of the network, a sort of coordinate system. I can act on the connection by increasing or decreasing the weight. In the course we will be dealing with the first type (no spatial loc.)

Conventional way to number the networks. The inputs are themself neurons, because the input can be assumed as the output of another layer somewhere. They are just "virtual neurons" the weights are not really relevant to the network.

ANN architectures

- Network layers



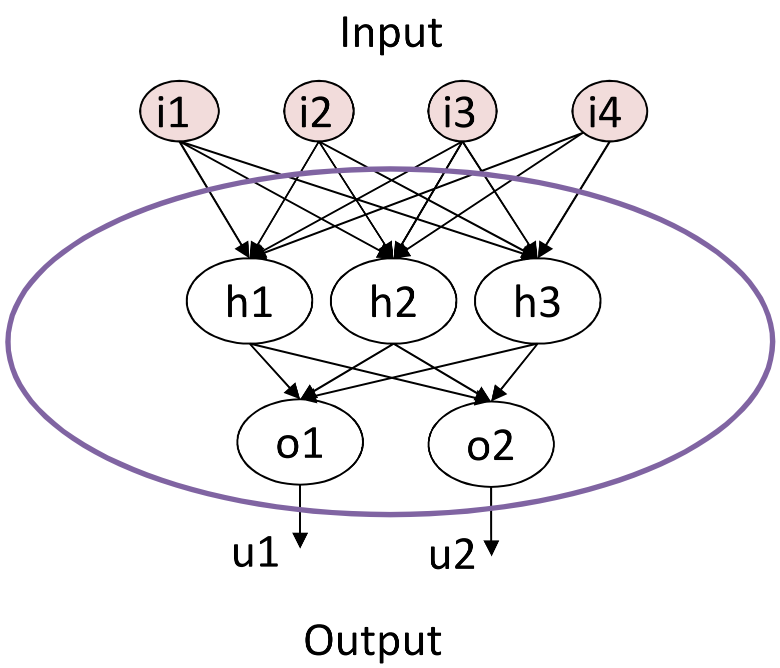


I can consider the network as a black box, I see the input i1-i4 and the output u1-u2. I dont know what is happening inside the network, I'm just seeing the output.

ANN architectures

- Network layers
- Input layer: neurons in this layer get signals from outside of the network
- Output layer: neurons in this layer produce signals towards outside of the network

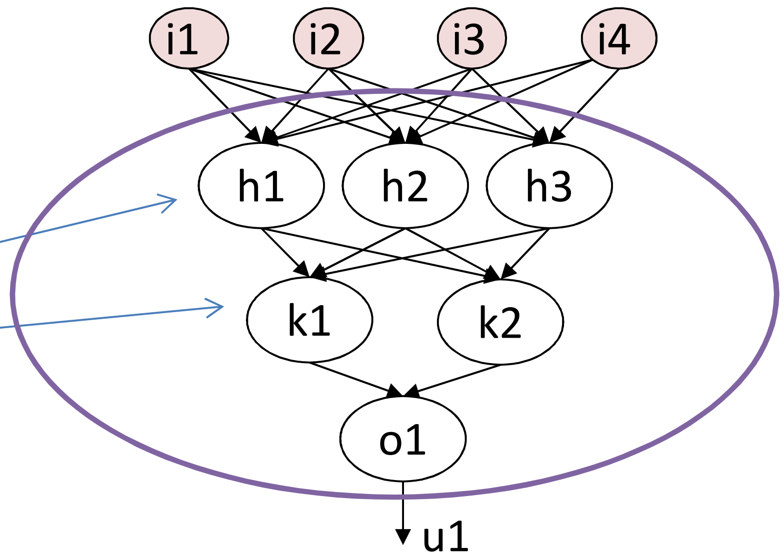
  - ❑ Input neurons are significant only for their output signals which are delivered possibly to the network neurons



Usually when we look at this kind of network we talk about hidden layers (we have two internal layer in the network in e.g.) I can measure u1 and have a refence of u1, but there's no way of having a reference for the signal before (after k1).

ANN architectures

- Network layers
- Input layer: neurons in this layer get signals from outside of the network
- Output layer: neurons in this layer produce signals towards outside of the network
- Internal (hidden) layers

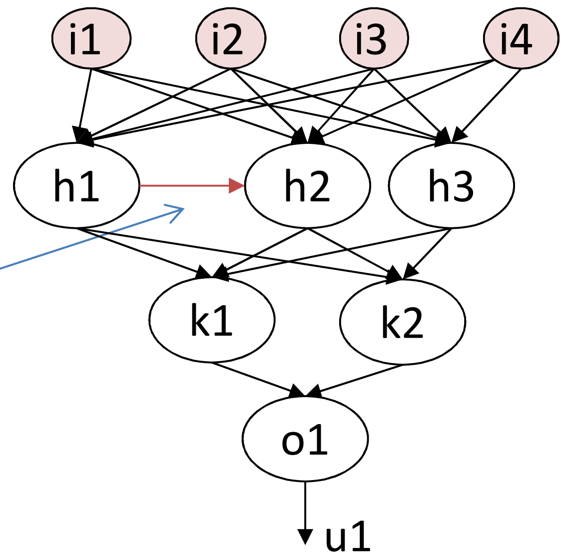


- ❑ We do not have any corresponding measure of the output of such neurons
- ❑ We can only get information (measure) about network output (u1)
- ❑ Hidden neurons – further processing (hopefully non-linear)

I can also have different kind of connections, intra layer we have a connection, h1 to h2

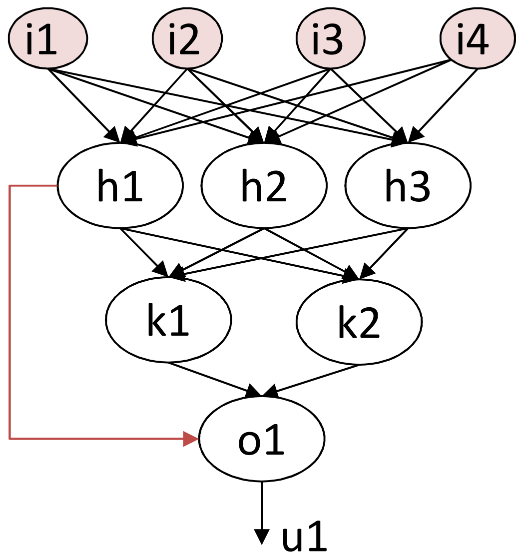
ANN architectures

- Network layers
- Input layer: neurons in this layer get signals from outside of the network
- Output layer: neurons in this layer produce signals towards outside of the network
- Internal (hidden) layers
- Connections:
  - Intra-layer



ANN architectures

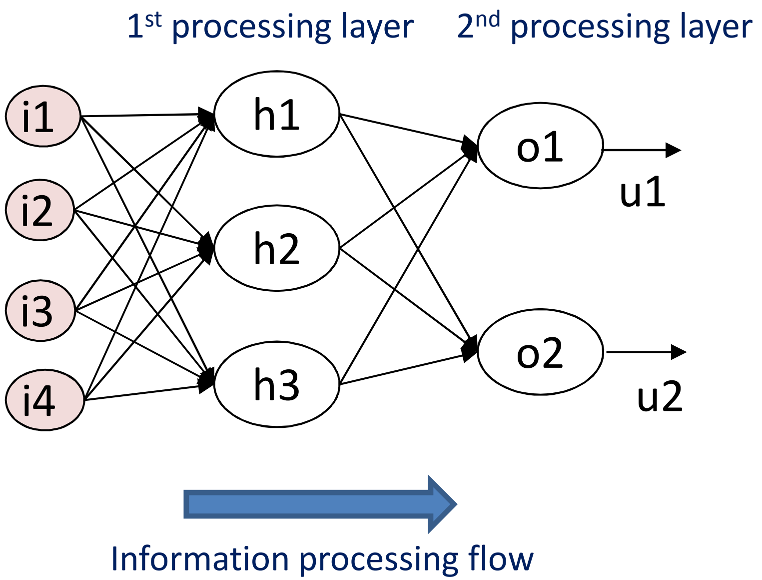
- Network layers
- Input layer: neurons in this layer get signals from outside of the network
- Output layer: neurons in this layer produce signals towards outside of the network
- Internal (hidden) layers
- Connections:
  - Intra-layer
  - Inter-layer



In case I have no one of the two cases seen I can talk about feedforward networks.

ANN architectures

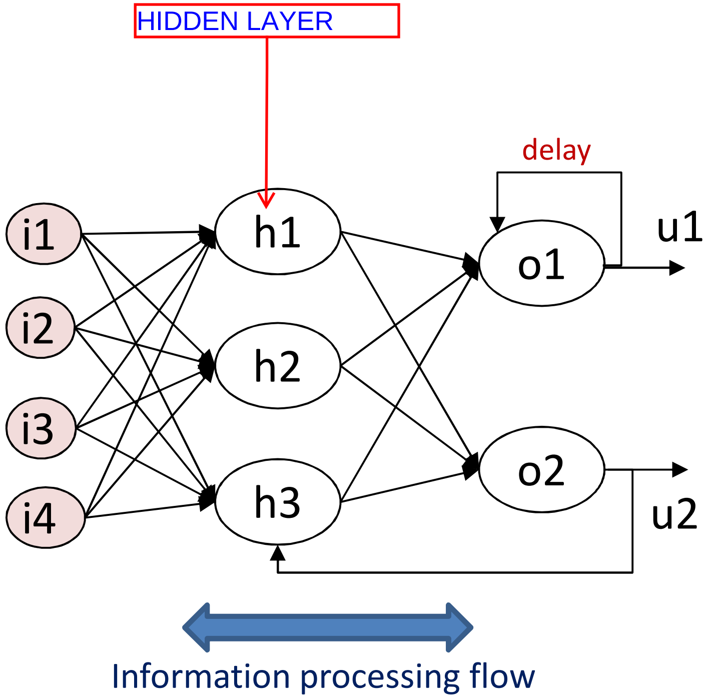
- Network layers
- Input layer: neurons in this layer get signals from outside of the network
- Output layer: neurons in this layer produce signals towards outside of the network
- Internal (hidden) layers
- Connections:
  - Intra-layer
  - Inter-layer
  - Feedforward
    - No intra- and inter-layer connections
    - No self-connections
    - No feedback connections



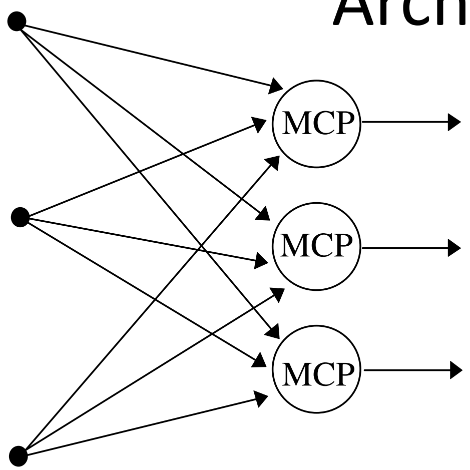
Feedback network, aka recurrent network. I have feedback, for example to an intermediate step or even to the beginning

ANN architectures

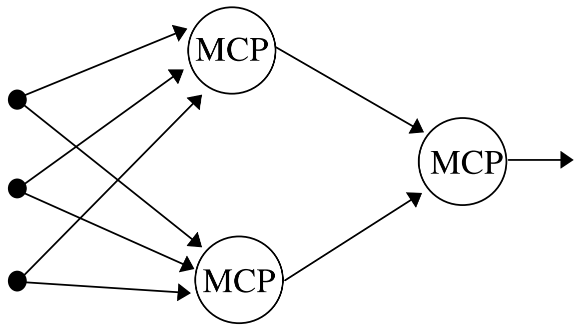
- Network layers
- Input layer: neurons in this layer get signals from outside of the network
- Output layer: neurons in this layer produce signals towards outside of the network
- Internal (hidden) layers
- Connections:
  - Intra-layer
  - Inter-layer
  - Feedforward – feedback (recurrent)
    - Need a time scale to delay the feedback signal



Architecture synthesis

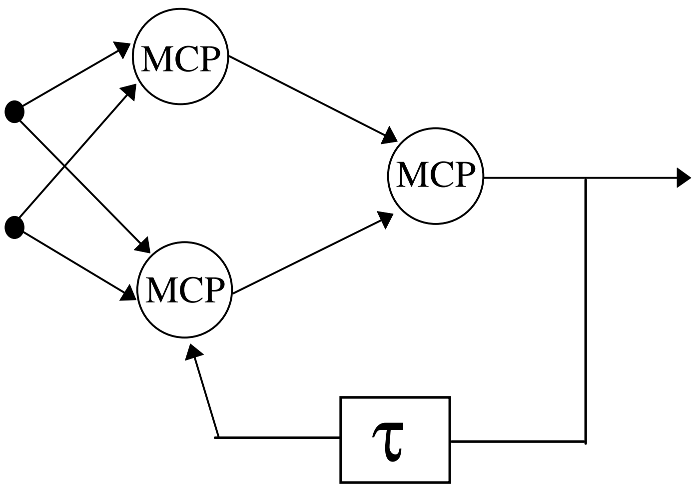


(a)

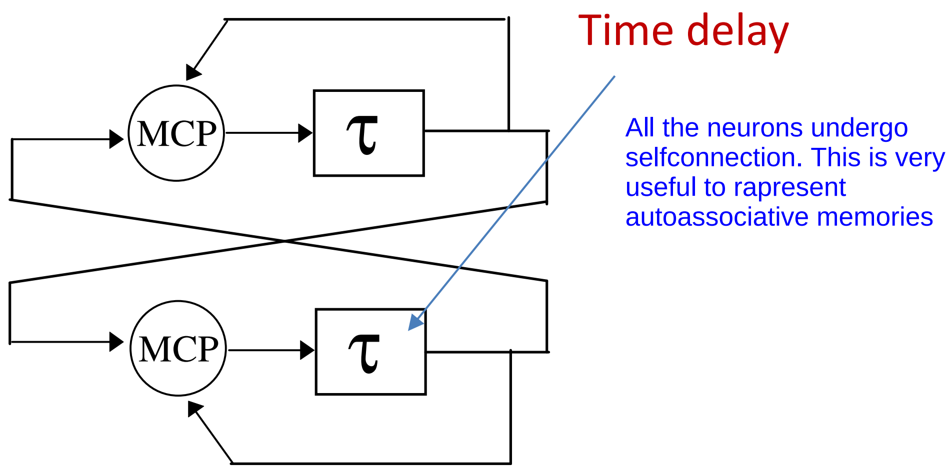


(b)

Perceptron **Multi-layer perceptron**



Recurrent



Fully-connected recurrent
Auto-associative network

Features to explore next

- Feed-forward multi-layer perceptron
- Networks that classify
- Networks that reconstruct functions
- Supervised learning
  - Stimulus-response with reinforcement
  - Learning rules and supervision criterion
  - Perceptron learning rule
  - Delta rule