ARTIFICIAL NEURAL NETWORKS MODULE 2

Dr. R.B.Ghongade,
VIIT, Pune-411048

Topics covered

- Introduction
- Neural networks (Usefulness & Capabilities)
- Nervous System
- Structure of a Nerve Cell
- Electrical Model of a Neuron
- Artificial Neuron
- Activation Functions
- Mc Culloch-Pitts Neuron (AND-NOT & XOR implementation)
- ANN Topologies
- ANN Architectures
- Learning Paradigms
- Revision of Some Basic Maths

Introduction

- BRAIN COMPUTATION
  - A highly complex non-linear and parallel computer with structural constituents called "NEURONS"
  - The **human brain** contains about 10^{11}nerve cells, or neurons
  - On an average, each neuron is connected to other neurons through approximately 10^4 synapses

	processing elements	element size	energy use	processing speed	style of computation	fault tolerant	learns	intelligent, conscious
	10^{14} synapses	10^{-6} m	30 W	100 Hz	parallel, distributed	yes	yes	usually
	10^8 transistors	10^{-6} m	30 W (CPU)	10^9 Hz	serial, centralized	no	a little	not (yet)

How is it then that the brain outperforms a digital computer?

Neural networks (Usefulness & Capabilities)

a) <u>Exploits Non-linearity</u>

- A system is linear if its output can be expressed as a linear combination of its inputs

  $y = a_1.x_1 + a_2.x_2 + \ldots + a_n.x_n$
- A system is non-linear if its outputs are expressed in not only the linear terms but also the higher order terms of its inputs

  $y = a_1.x_1 + a_2.{x_1}^2 + \ldots$
- Most of the real-world problems are non-linear and hence we need non-linear units to solve them, neurons are non-linear
- Since the brain has a large interconnection of non-linear neurons, non-linearity is distributed throughout

b) Input-Output Mapping

- By modifying free parameters we may be able to map input to desired output
- This process is called as learning with the help of a teacher
- Two types of learning
  - With the help of a teacher
  - Without the help of a teacher (also called auto-associative learning)
- We specify that for a given input what should be the output or desired response
- It is possible that we may get a different output than the desired one, in this case we accordingly modify the set of free parameters so as to get the output most closest to the desired output
- This may not be achieved immediately at first but the difference between the output and the desired value is reduced and subsequently with more iterations the outputs will match the desired response, this is the process of learning
- A teacher is required to adjust the free parameters
- Thus neural networks are different from conventional systems because they involve learning
- Ex. Learning of a child

c) Adaptability

- Neural networks can adapt the free parameters to the changes in the surrounding environment
- Humans adapt to their surroundings from time to time and thus cope up with the world!

d) Evidential Response

- Humans respond with confidence level
- Ex. “I think I am going to pass the exam”, denotes that the individual is confident up to a good degree
- Thus there is a decision with a measure of confidence

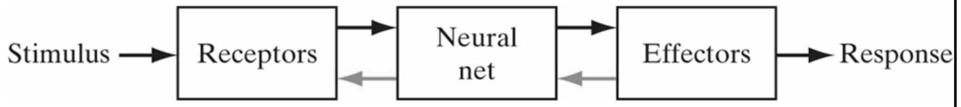
e) <u>Fault Tolerance</u>
- Even if a single connection is malfunctioning , the nervous system functions, there is no catastrophic failure
- At the maximum there is graceful degradation
- It depends on how severe the fault is, if the fault is with too many units, the degree of degradation is large and vice-versa
- It is possible to incorporate this fault tolerant mechanism in artificial neural networks

f) <u>VLSI Implementation Ability</u>
- It is possible to integrate a large number of artificial neurons using VLSI technology
- Neurons do independent computations giving a good degree of parallelism

g) <u>Neurobiological Analogy</u>
- These properties of a biological neuron can be imparted to the artificial neuron

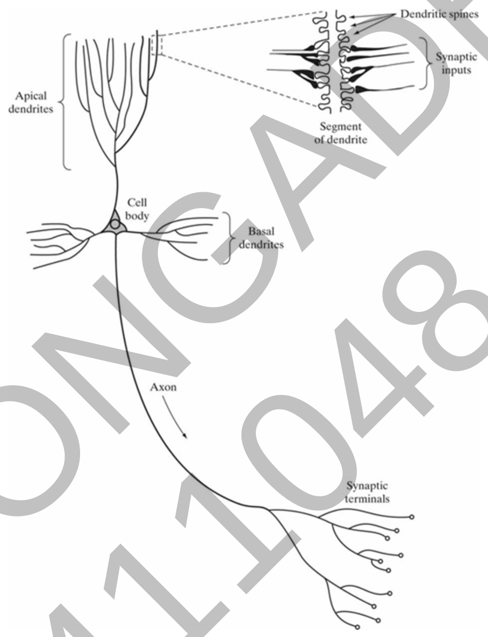
Nervous System

- Human nervous system may be viewed as a three-stage system
  - The ***receptors*** convert stimuli from the human body or the external environment into electrical impulses that convey information to the neural net (brain)
  - ***Neural (nerve) net***, which continually receives information, perceives it, and makes appropriate decisions
  - The ***effectors*** convert electrical impulses generated by the neural net into discernible responses as system outputs
- Arrows pointing from left to right indicate the *forward* transmission of information-bearing signals through the system
- The arrows pointing from right to left signify the presence of *feedback* in the system



Structure of a Nerve Cell

- *Synapses* are elementary structural and functional units that mediate the interactions between neurons
- *Axons,* the transmission lines, and *dendrites,* the receptive zones, constitute two types of cell filaments that are distinguished on morphological grounds
- An axon has a smoother surface, fewer branches, and greater length
- A dendrite (so called because of its resemblance to a tree) has an irregular surface and more branches
- It is assumed that a synapse is a simple connection that can impose *excitation* or *inhibition,* but not both on the receptive neuron



Operation

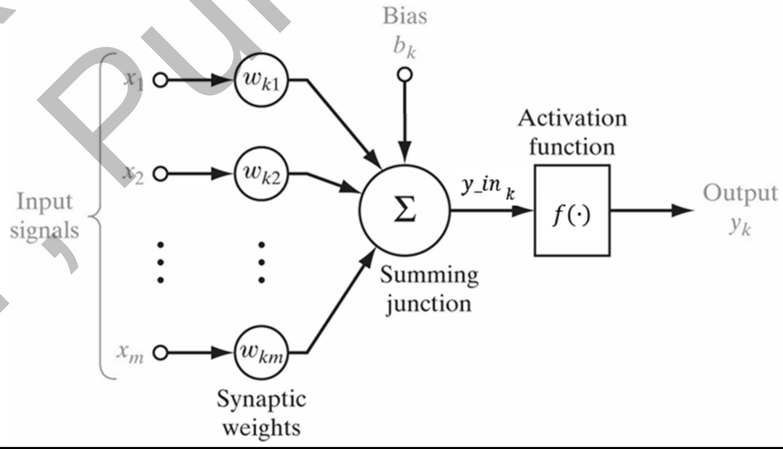
- Strength of synaptic connections decide the net signal coming to the cell body
- Free parameters refer to the strength of the synapse
- Every input is associated with synaptic strengths
- Assuming some a-priori synaptic strength a cell computes the response
- If the response differs from the desired value, the synaptic strengths are adjusted
- This process may take many iterations

Model of a Neuron

Three basic elements of the neuronal model:

- A set of *synapses* or *connecting links,* each of which is characterized by a *weight* or *strength* of its own
  - Specifically, a signal $\boldsymbol{x_j}$ at the input of synapse ***j*** connected to neuron ***k*** is multiplied by the synaptic weight $\boldsymbol{w_{kj}}$
  - Unlike a synapse in the brain, the synaptic weight of an artificial neuron may lie in a range that includes **negative as well as positive values**
- An *adder* for summing the input signals, weighted by the respective synapses of **the neuron; the operations described here constitute a *linear combiner***
- An *activation function* for limiting the amplitude of the output of a neuron

Electrical Model of a Neuron

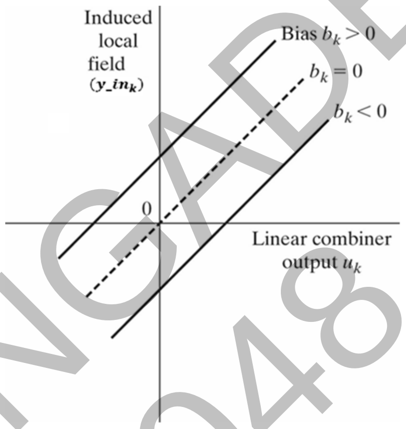


$$\boxed{\boldsymbol{y\_in_k = b_k + \sum_{i=1}^{m} x_i \cdot w_{ki} = b_k + x_1 \cdot w_{k1} + x_2 \cdot w_{k2} + \cdots + x_m \cdot w_{km}}}$$

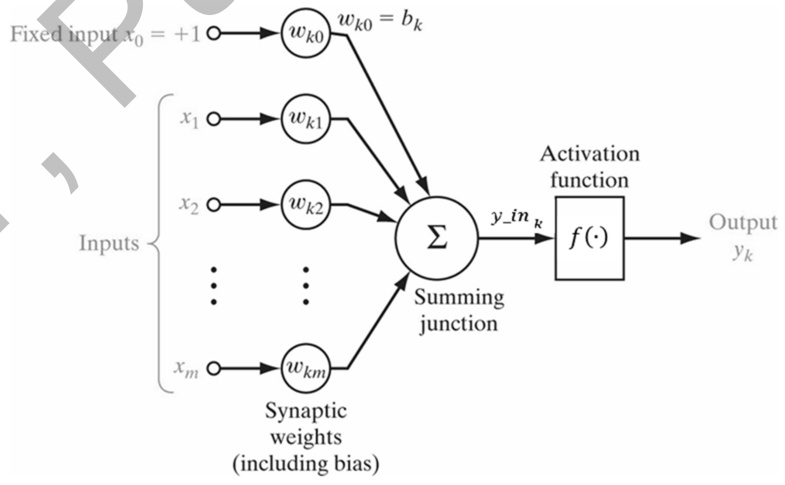
$$\boxed{\boldsymbol{y_k = f(y\_in_k)}}$$

Effect of bias

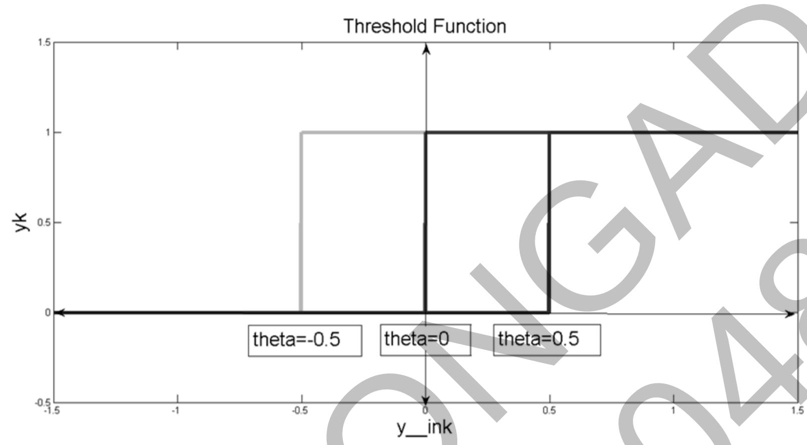
- The neuronal model also includes an externally applied *bias,* denoted by b_k
- The bias b_k has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively
- The use of bias b_k, has the effect of applying an *affine transformation* to the output y_in_k of the linear combiner in the model
- An *affine transformation* is any transformation that preserves collinearity (i.e., all points lying on a line initially still lie on a line after transformation)
- Depending on whether the bias b_k is positive or negative, the relationship between the *induced local field* or *activation potential* y_in_k of neuron k and the linear combiner output u_k is modified



Another nonlinear model of a neuron; w_{k0} accounts for the bias b_k.

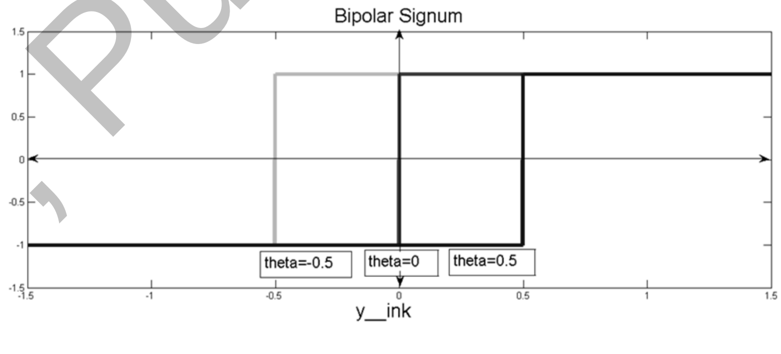


Activation Functions- Threshold Function



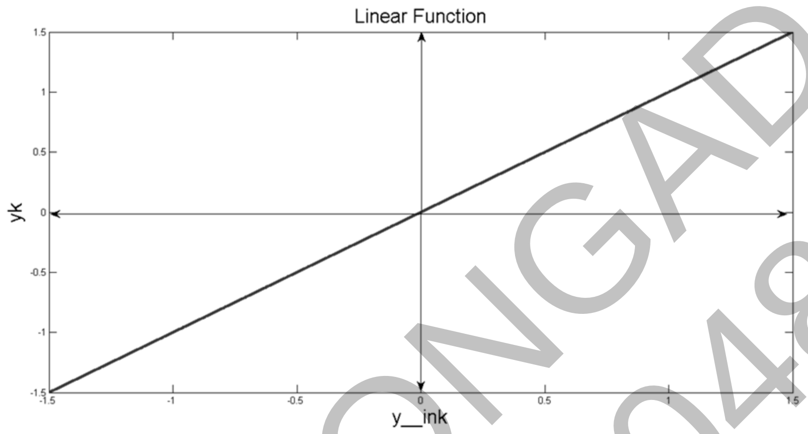
$$\text{If} \quad y\_in_k \geq \theta \qquad y_k = 1$$
$$y_k = 0 \quad \text{otherwise}$$

Activation Functions- Signum Function



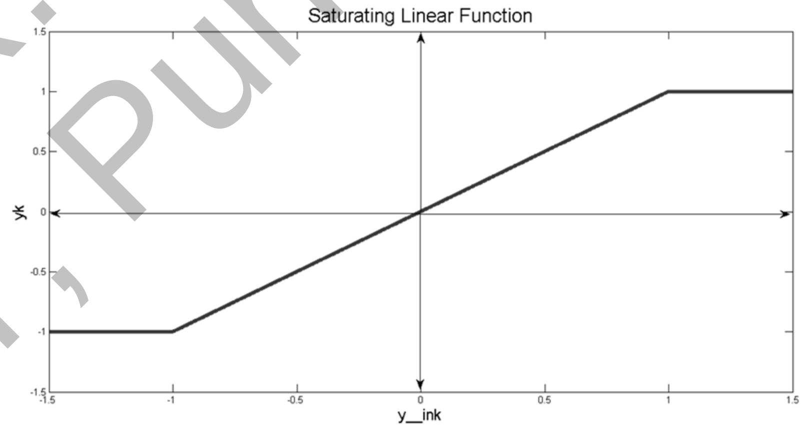
$$\text{If} \quad y\_in_k \geq \theta \qquad y_k = 1$$
$$y_k = -1 \quad \text{otherwise}$$

Activation Functions- Linear Function



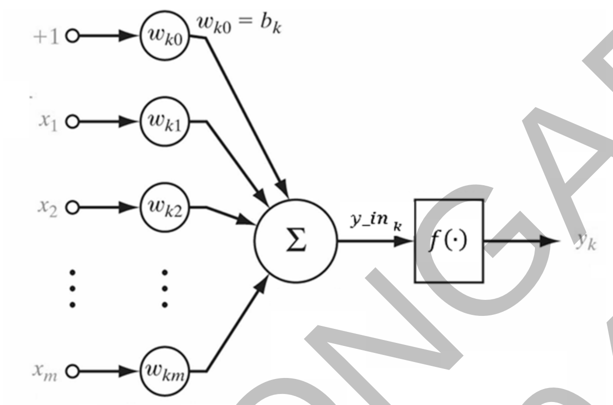
$y_k = y\_in_k$

Activation Functions- Saturating Linear Function



$$\begin{array}{lll} If \quad y\_in_k \geq 1 & y_k = 1 & \\ & y_k = y\_in_k & \text{otherwise} \\ If \quad y\_in_k \leq -1 & y_k = -1 & \\ & y_k = y\_in_k & \text{otherwise} \end{array}$$

Artificial Neuron Model



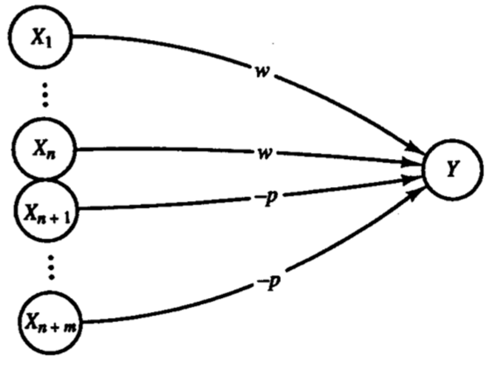
$$y\_in_k = \sum_{i=0}^{m} x_i \cdot w_{ki} = (1) \cdot w_{k0} + x_1 \cdot w_{k1} + x_2 \cdot w_{k2} + \cdots + x_m \cdot w_{km}$$

$$\boxed{y_k = f(y\_in_k)}$$

y_in_k is called the local induced field

Mc Culloch-Pitts Neuron

- McCulloch-Pitts neuron *Y* may receive signals from any number of other neurons
- Each connection path is either excitatory, with weight *w* > 0, or inhibitory, with weight - *p* (*p* > 0)



$$f(y\_in) = \begin{cases} 1 & \text{if } y\_in \geq \theta \\ 0 & \text{if } y\_in < \theta \end{cases}$$

- The condition that inhibition is absolute requires that θ for the activation function satisfy the inequality

$$\theta \geq nw - p$$

- *Y* will fire if it receives *k* or more excitatory inputs and no inhibitory inputs, where

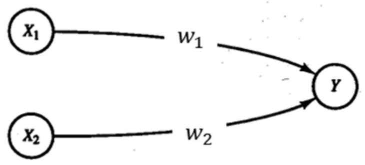
$$kw \geq \theta > (k-1)w$$

- Although all excitatory weights coming into any particular unit must be the same, the weights coming into one unit, say, Y_1, do not have to be the same as the weights coming into another unit, say Y_2

Logical AND NOT Implementation

x1	x2	Y
0	0	0
0	1	0
1	0	1
1	1	0

$$y = x_1\overline{x_2}$$



- Assume that w_1, w_2 are excitatory and $w_1 = w_2 = 1$
- Output is given by $y_{in} = x_1w_1 + x_2w_2$

x1.w1	X2.w2	Yin
0×1	0×1	0
0×1	1×1	1
1×1	0×1	1
1×1	1×1	2

- We see that for no threshold value we can separate the third combination
- Change the weights to
  $w_1 = 1$, $w_2 = -1$

x1.w1	x2.w2	yin
0×1	0× (−1)	0
0×1	1×(-1)	-1
1×1	0× (−1)	1
1×1	1×(-1)	0

- Now we can separate the third combination by setting $\theta \geq 1$
- So with w_1=1 , $w_2 = -1$ and $\theta \geq 1$ we can have the following response

$$If \;\; y\_in \geq 1 \qquad y = 1$$
$$y = 0, otherwise$$

x1.w1	x2.w2	yin	θ	y
0×1	0× (−1)	0	1	0
0×1	1×(-1)	-1	1	0
1×1	0× (−1)	1	1	1
1×1	1×(-1)	0	1	0

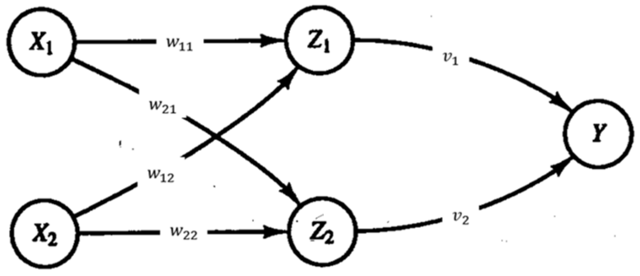
All the weight and threshold setting has to be done with trial and error!

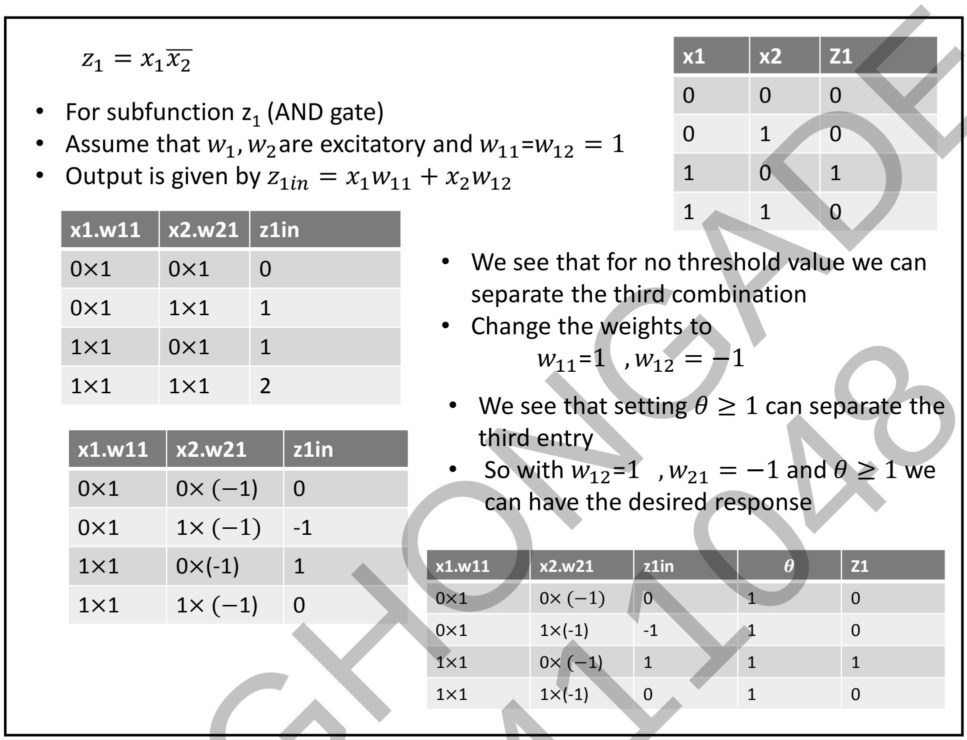
Logical XOR Implementation

x1	x2	Y
0	0	0
0	1	1
1	0	1
1	1	0

$$y = x_1\overline{x_2} + \overline{x_1}x_2$$
$$y = z_1 + z_2$$

We observe that we need two AND gates and one OR gate





$z_1 = x_1\overline{x_2}$

x1	x2	Z1
0	0	0
0	1	0
1	0	1
1	1	0

- For subfunction z_1 (AND gate)
- Assume that w_1, w_2 are excitatory and w_{11}=$w_{12} = 1$
- Output is given by $z_{1in} = x_1 w_{11} + x_2 w_{12}$

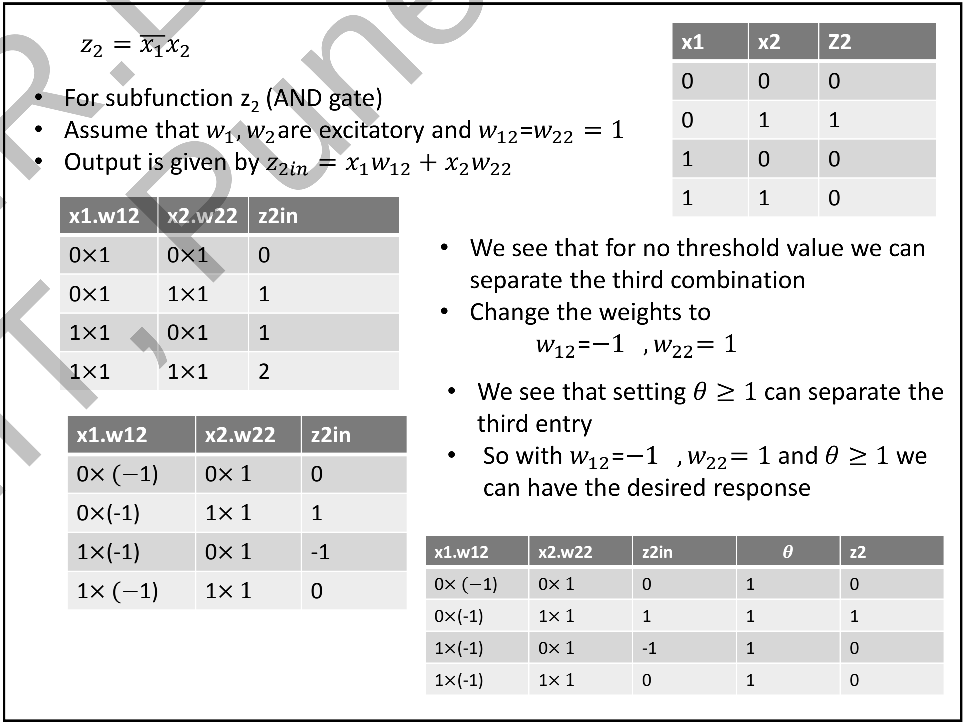
x1.w11	x2.w21	z1in
0×1	0×1	0
0×1	1×1	1
1×1	0×1	1
1×1	1×1	2

- We see that for no threshold value we can separate the third combination
- Change the weights to
  w_{11}=1 , $w_{12} = -1$

x1.w11	x2.w21	z1in
0×1	0× (−1)	0
0×1	1× (−1)	-1
1×1	0×(-1)	1
1×1	1× (−1)	0

- We see that setting $\theta \geq 1$ can separate the third entry
- So with w_{12}=1 , $w_{21} = -1$ and $\theta \geq 1$ we can have the desired response

x1.w11	x2.w21	z1in	θ	Z1
0×1	0× (−1)	0	1	0
0×1	1×(-1)	-1	1	0
1×1	0× (−1)	1	1	1
1×1	1×(-1)	0	1	0



$z_2 = \overline{x_1}x_2$

x1	x2	Z2
0	0	0
0	1	1
1	0	0
1	1	0

- For subfunction z_2 (AND gate)
- Assume that w_1, w_2 are excitatory and w_{12}=$w_{22} = 1$
- Output is given by $z_{2in} = x_1 w_{12} + x_2 w_{22}$

x1.w12	x2.w22	z2in
0×1	0×1	0
0×1	1×1	1
1×1	0×1	1
1×1	1×1	2

- We see that for no threshold value we can separate the third combination
- Change the weights to
  w_{12}=-1 , w_{22}= 1

x1.w12	x2.w22	z2in
0× (−1)	0× 1	0
0×(-1)	1× 1	1
1×(-1)	0× 1	-1
1× (−1)	1× 1	0

- We see that setting $\theta \geq 1$ can separate the third entry
- So with w_{12}=-1 , w_{22}= 1 and $\theta \geq 1$ we can have the desired response

x1.w12	x2.w22	z2in	θ	z2
0× (−1)	0× 1	0	1	0
0×(-1)	1× 1	1	1	1
1×(-1)	0× 1	-1	1	0
1×(-1)	1× 1	0	1	0

$$y = z_1 + z_2$$

- For subfunction y (OR gate)
- Note that the combination $z_1 = z_2 = 1$ **NEVER OCCURS**

x1	x2	z1	z2	y
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

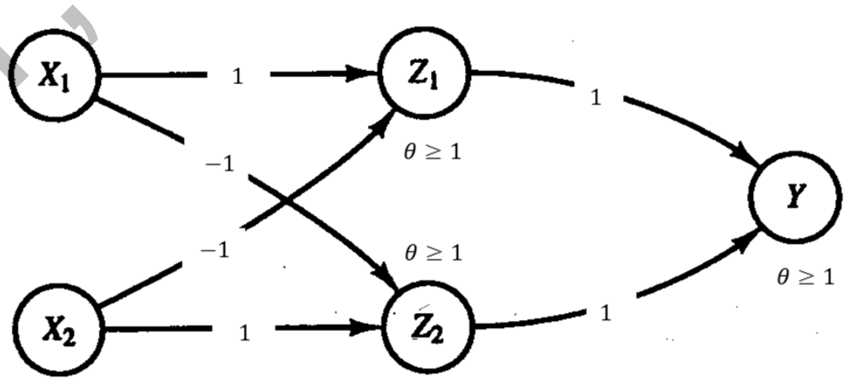
- Assume that $v_1, v_2$are excitatory and v_1=$v_2 = 1$
- Output is given by $y_{in} = z_1v_1 + z_2v_2$

z1.v1	z2.v2	yin
0×1	0×1	0
0×1	1×1	1
1×1	0×1	1
0×1	0×1	0

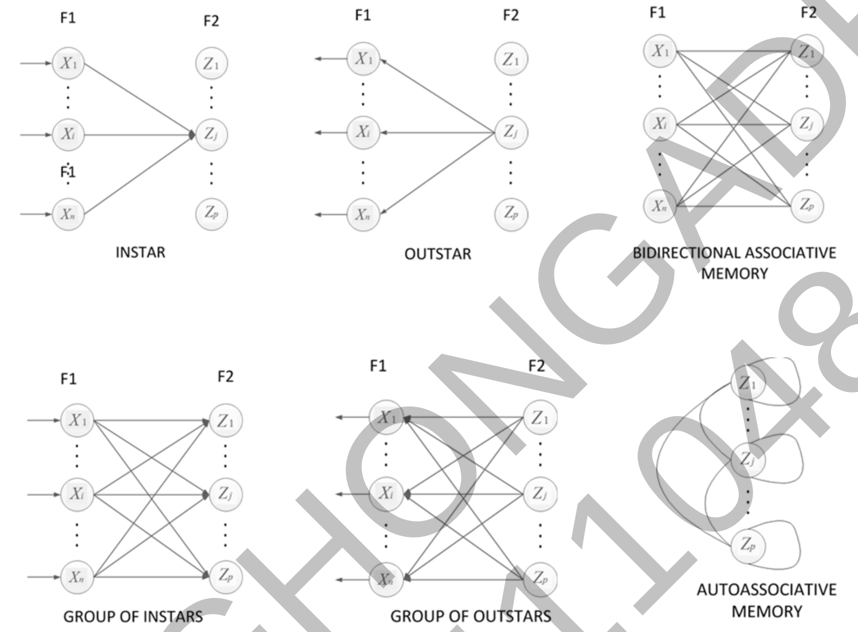
- We see that for $\theta \geq 1$ we can separate the second and the third combination
- With v_1=1 , $v_2 = 1$ and $\theta \geq 1$ we can have the desired response

x1	x2	z1	z2	yin	θ	y
0	0	0×1	0×1	0	1	0
0	1	0×1	1×1	1	1	1
1	0	1×1	0×1	1	1	1
1	1	0×1	0×1	0	1	0

- Hence the complete solution to XOR implementation is

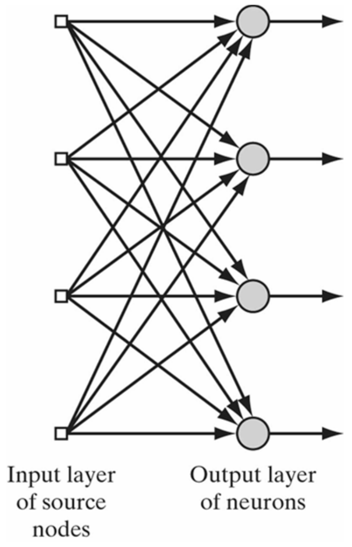


ANN Topologies

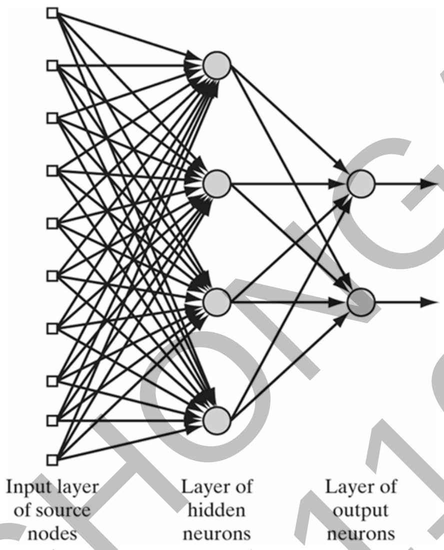


ANN Architectures

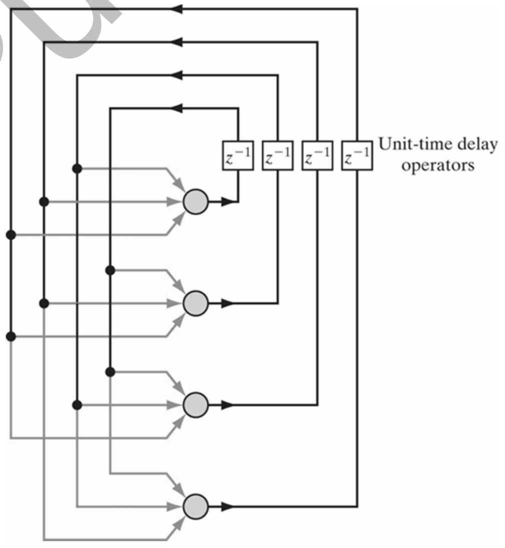
- Feedforward network with a single layer of neurons



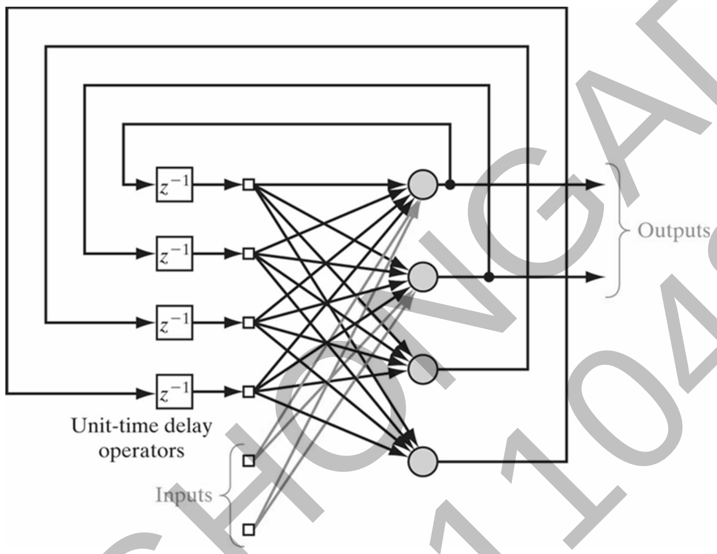
- Fully connected feedforward network with one hidden layer and one output layer



- Recurrent network with no self-feedback loops and no hidden neurons



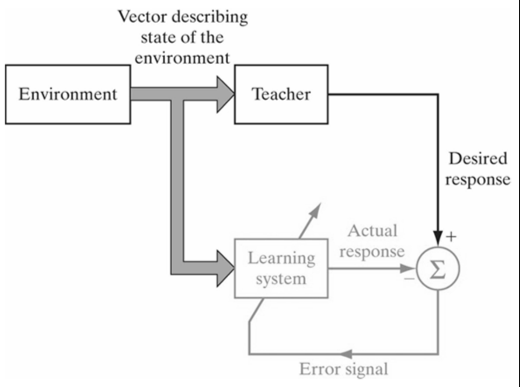
- Recurrent network with hidden neurons



Learning Paradigms

LEARNING WITH A TEACHER (SUPERVISED LEARNING)

- Teacher as has knowledge of the environment, with that knowledge being represented by a set of *input-output examples*
- The environment is, however, *unknown* to the neural network of interest
- Suppose now that the teacher and the neural network are both exposed to a training vector
- By virtue of built-in knowledge, the teacher is able to provide the neural network with a desired response for that training vector



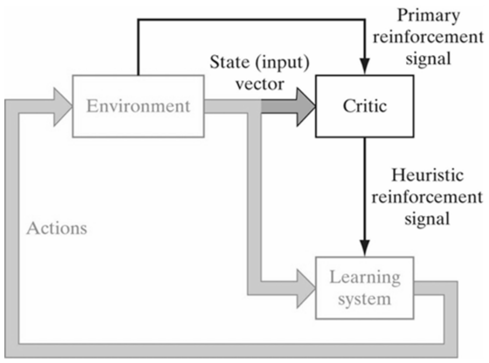
- The network parameters are adjusted under the combined influence of the training vector and the error signal, iteratively
- The *error signal* is defined as the difference between the desired response and the actual response of the network

LEARNING WITHOUT A TEACHER

- There is *no* teacher to oversee the learning process
- Hence there are no labeled examples of the function to be learned by the network
- Two sub types:
  - Reinforcement learning / Neurodynamic programming
  - Unsupervised or self-organized learning

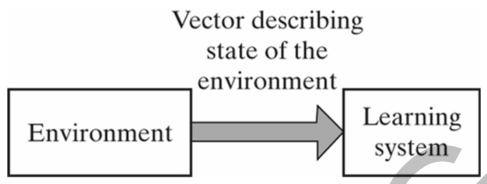
Reinforcement learning

- Learning of an input-output mapping is performed through continued interaction with the environment in order to minimize a scalar index of performance
- Built around a *critic* that converts a *primary reinforcement signal* received from the environment into a higher quality reinforcement signal called the *heuristic reinforcement signal,* both of which are scalar inputs



- The system observes a temporal sequence of stimuli (i.e., state vectors) also received from the **environment, which eventually result in the generation of the heuristic reinforcement** signal
- The goal of learning is to minimize a *cost-to-go function,* defined as the expectation of the cumulative cost of *actions* taken over a sequence of steps instead of simply the immediate cost

Unsupervised Learning



- There is no external teacher or critic to oversee the learning process
- Provision is made for a t*ask independent measure* of the quality of representation that the network is required to learn, and the free parameters of the network are optimized with respect to that measure
- Once the network has become tuned to the statistical regularities of the input data, it develops the ability to form internal representations for encoding features of the input and thereby to create new classes automatically

Revision of Some Basic Maths

- Vector and Matrix
  - Row vector/column vector/vector transposition
  - Vector length/norm
  - Inner/dot product
  - Matrix (vector) multiplication
  - Linear algebra
  - Euclidean space
- Basic Calculus
  - Partial derivatives
  - Gradient
  - Chain rule

Revision of Some Basic Maths

- Inner/dot product

x = $[x_1, x_1, \ldots, x_n]^T$ **,** ***y*** = $[y_1, y_1, \ldots, y_n]^T$

Inner/dot product of *x* and *y*, x^Ty

$$x^T y = x_1y_1 + x_2y_2 + \cdots + x_ny_n = \sum_{i=1}^{n} x_iy_i$$

- Matrix/Vector multiplication

Revision of Some Basic Maths

- Vector space/Euclidean space
  - A vector space V is a set that is closed under finite vector addition and scalar multiplication.
  - The basic example is n-dimensional Euclidean space, where every element is represented by a list of n real numbers
  - An n-dimensional real vector corresponds to **a point** in the Euclidean space.

[1, 3] is a point in 2-dimensional space
[2, 4, 6] is point in 3-dimensional space

Revision of Some Basic Maths

- Vector space/Euclidean space
  - Euclidean space (Euclidean distance)

$$\|X-Y\| = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 \dots + (x_n-y_n)^2}$$

  - Dot/inner product and Euclidean distance
    - Let *x* and *y* be two normalized vectors, ||*x*||= 1, ||*y*||=1, we can write

$$\|X-Y\|^2 = (X-Y)^T(X-Y) = 2 - 2X^TY$$

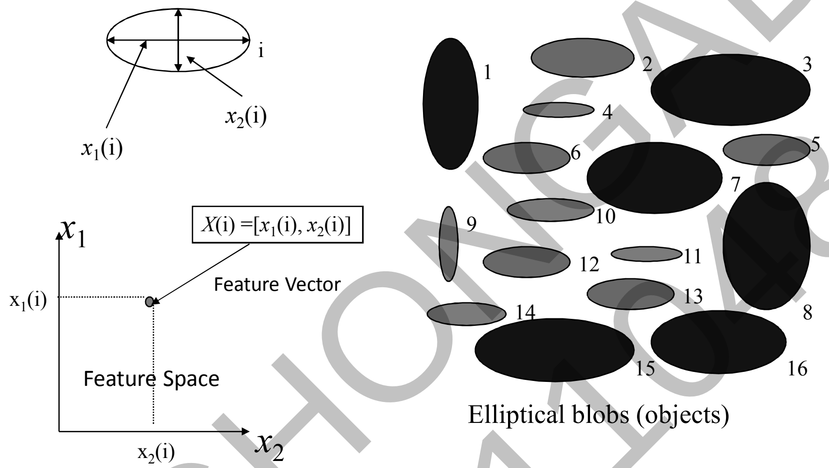
    - Minimization of Euclidean distance between two vectors corresponds to maximization of their inner product.
  - Euclidean distance/inner product as similarity measure

Revision of Some Basic Maths

- Basic Calculus
  - Multivariable function: $y(x) = f(x_1, x_2, \dots, x_n)$
  - Partial derivative: gives the direction and speed of change of *y*, with respect to *x*i
  - Gradient $\nabla f = \left[\frac{\partial f}{\partial x_1}, \dots\dots \frac{\partial f}{\partial x_n}\right]$
  - **Chain rule**: Let *y* = *f* (*g*(*x*)), *u* = *g*(*x*), then $\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}$
  - Let *z* = *f*(*x*, *y*), *x* = *g*(*t*), *y* = *h*(*t*), then $\frac{dz}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}$

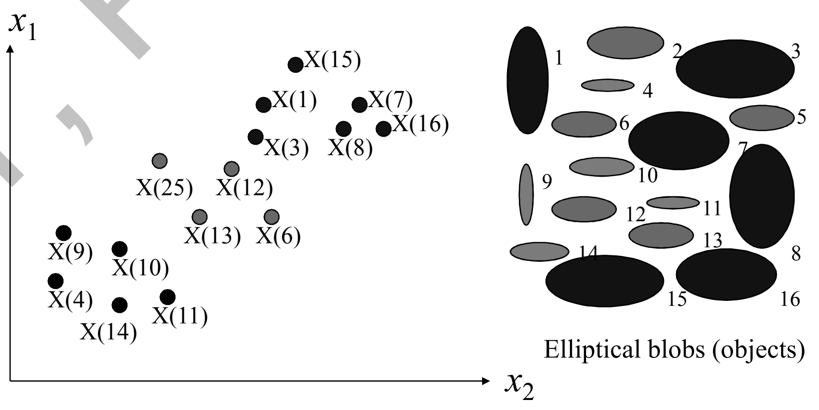
Feature Space

- Representing real world objects using **feature vectors**



Feature Space

✪ From Objects to Feature Vectors to Points in the Feature Spaces



Linear Neuron as classifier

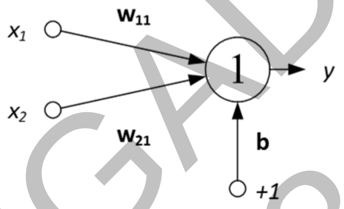
- Consider the line represented by a single neuron, with equation: $x_1 w_{11} + x_2 w_{21} + b = 0$
- Re-arranging we get:

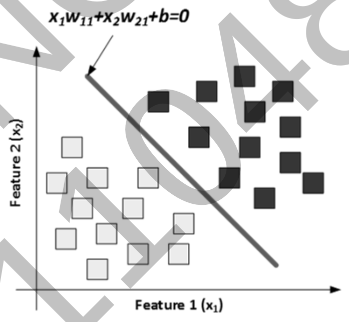
$$x_2 = \left(-\frac{w_{11}}{w_{21}}\right) x_1 + \left(-\frac{b}{w_{21}}\right)$$

- Is of the form: $y = mx + c$

Where $m = -\frac{w_{11}}{w_{2\ 1}}$ and $c = -\frac{b}{w_{2\ 1}}$

- If we set appropriate values for w_{11} , w_{21} , b (such that yellow objects produce output <0 and red objects produce output >0)then we can use this line as a decision boundary separating objects
- Now if a new object $\vec{T} = [t_1\ t_2]$ produces output of neuron $y > 0$, it simply means the object belongs to the red object class
- Thus the linear neuron works as a classifier!


x_1
$\mathbf{w_{11}}$
1
y
x_2
$\mathbf{w_{21}}$
b
+1


$x_1w_{11}+x_2w_{21}+b=0$
Feature 2 (x_2)
Feature 1 (x_1)