Computational Neuroscience of Vision

Edmund T Rolls

Gustavo Deco

Lecture 1

Categories of applications of ANN:

- **Function approximation, or regression analysis, including time series prediction and modeling;**

- **Classification, including pattern and sequence recognition, novelty detection and sequential decision making.**

- **Data processing, including filtering, clustering, blind source separation and compression.**

Application areas include :

System identification and control (vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications (automated trading systems), data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

Introduction to Artificial Neural Networks

Categories (architecture, learning, application) of ANN:

- **Feed-forward with BPNN learning (perceptron, ADALINE, MADALINE**

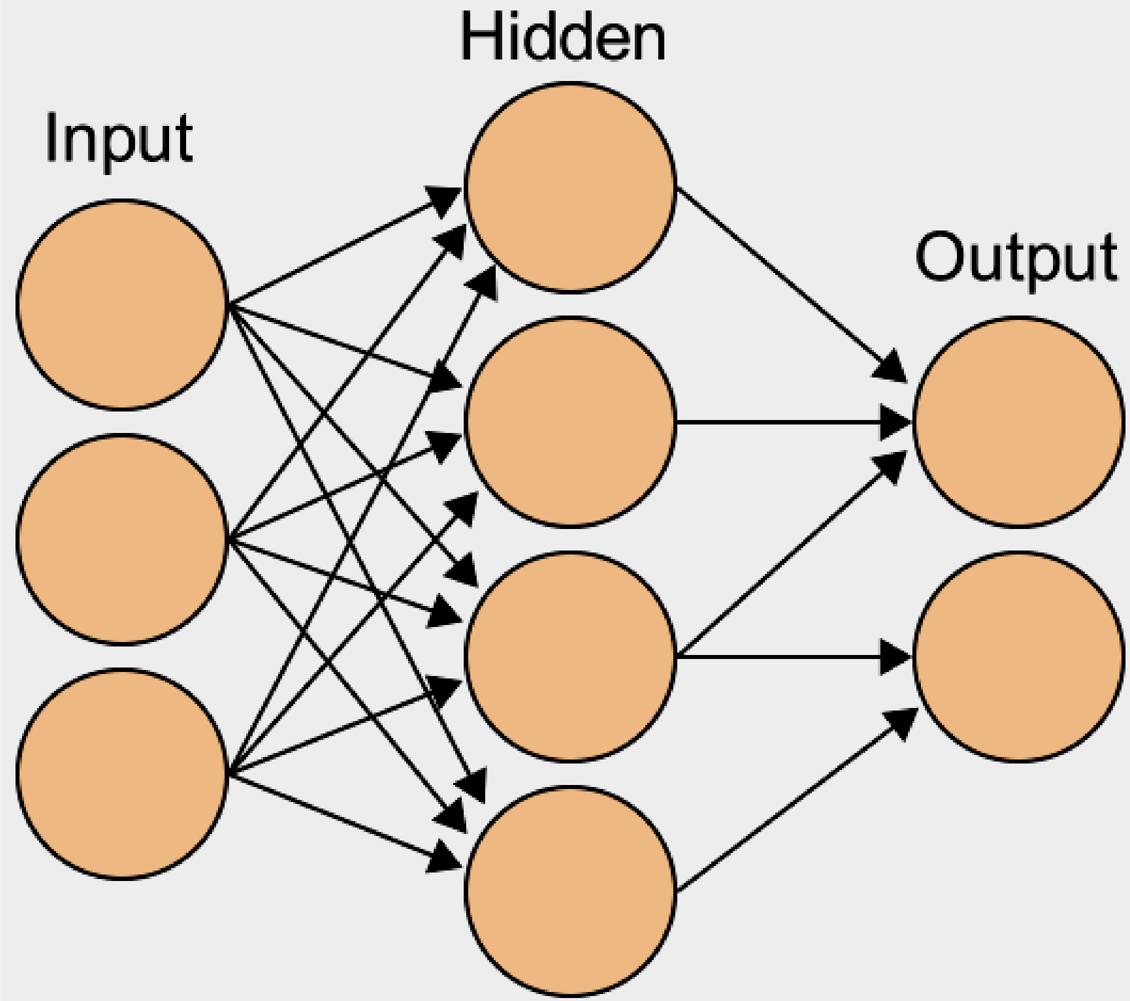
- **Associative nets (CAMs, BAMs, Hopfield net and Boltzmann machine); Auto- & hetero-associative nets and recurrent NNs**

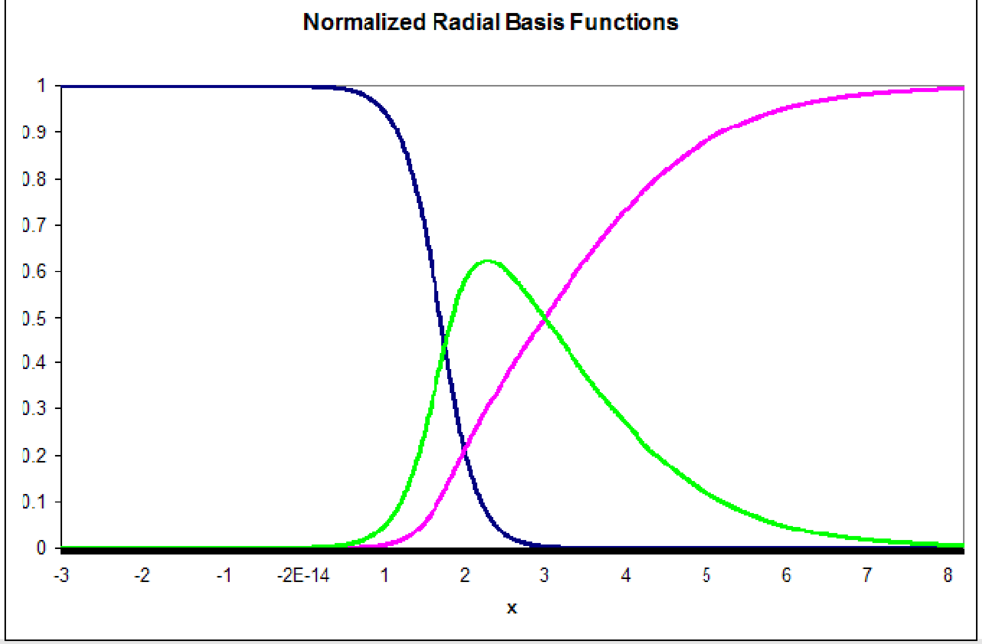
- **Competitive (SOM, counter propagation)**

- **ART, TDNN**

- **SVM, RBFN**

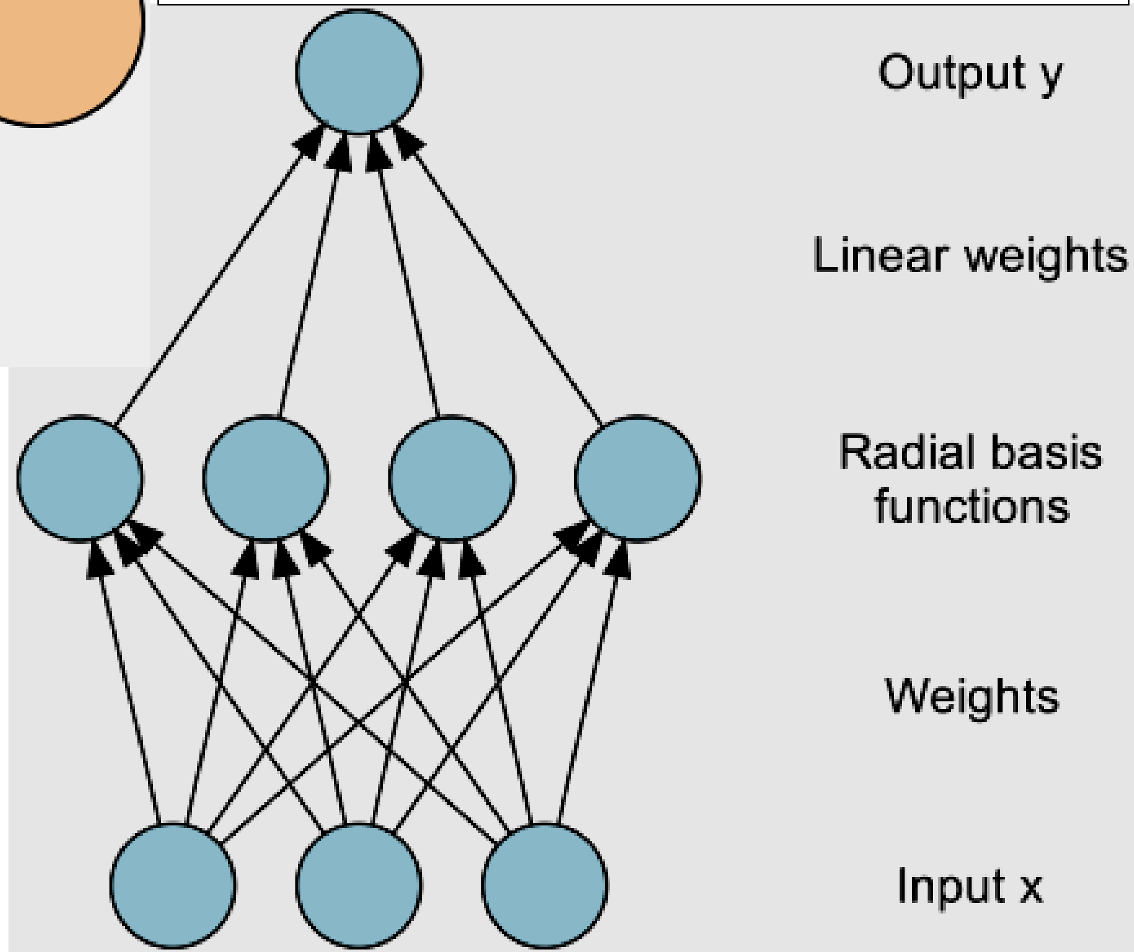
- **Prob-NN, Pulse-NN (Spike-NN) etc.**



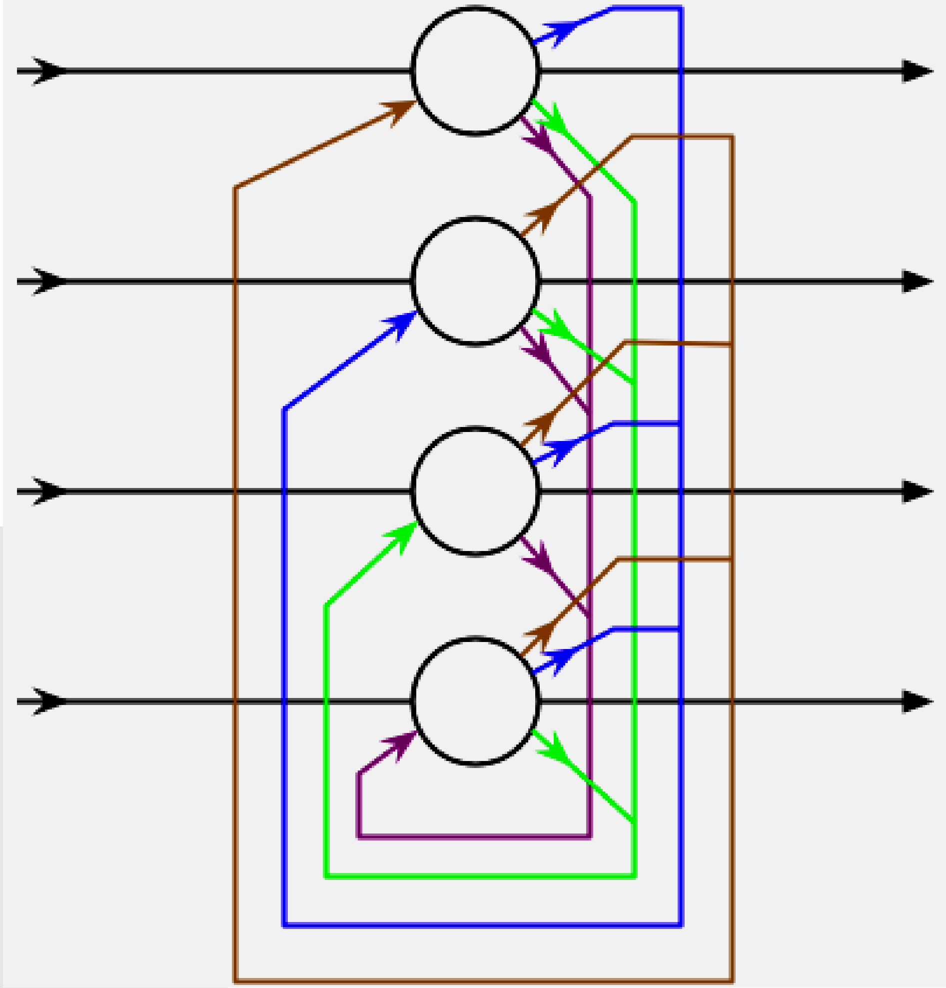


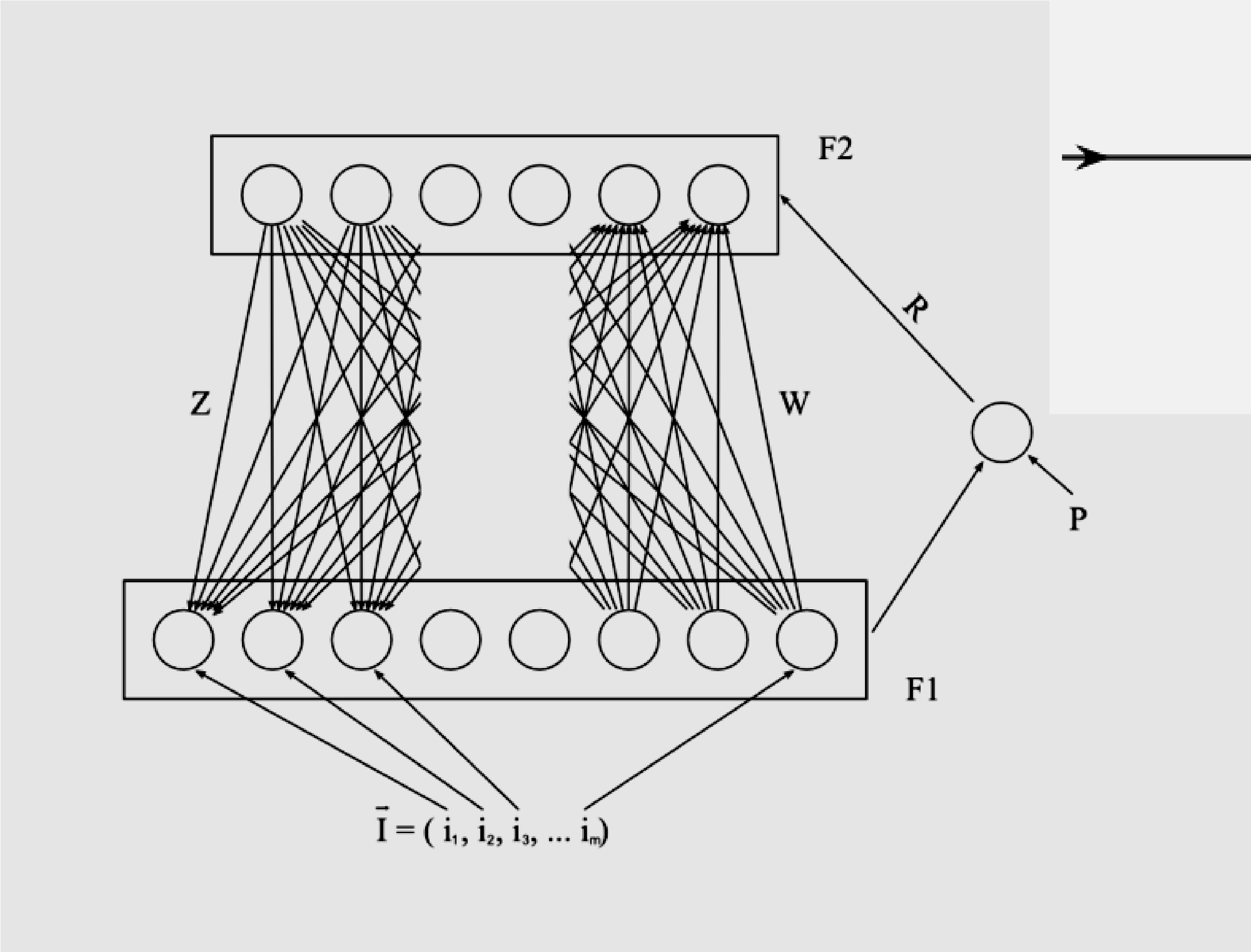
FFNN Architecture

RBFN:



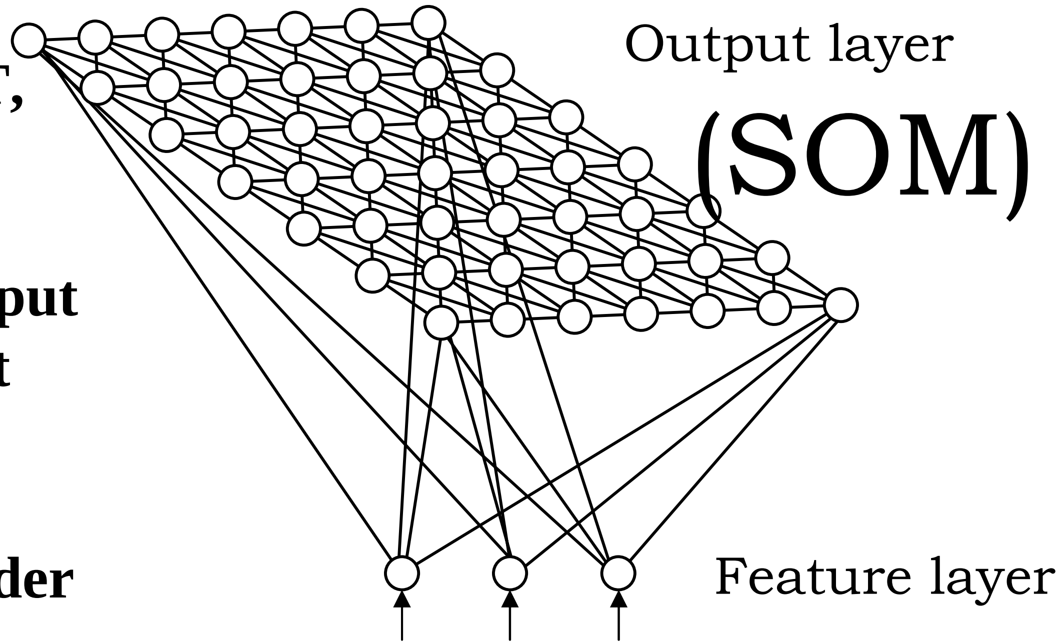
Hopfield net (recurrent/attractor); Boltzmann has multiple layers of the same (nodes and convergence – stochastic)

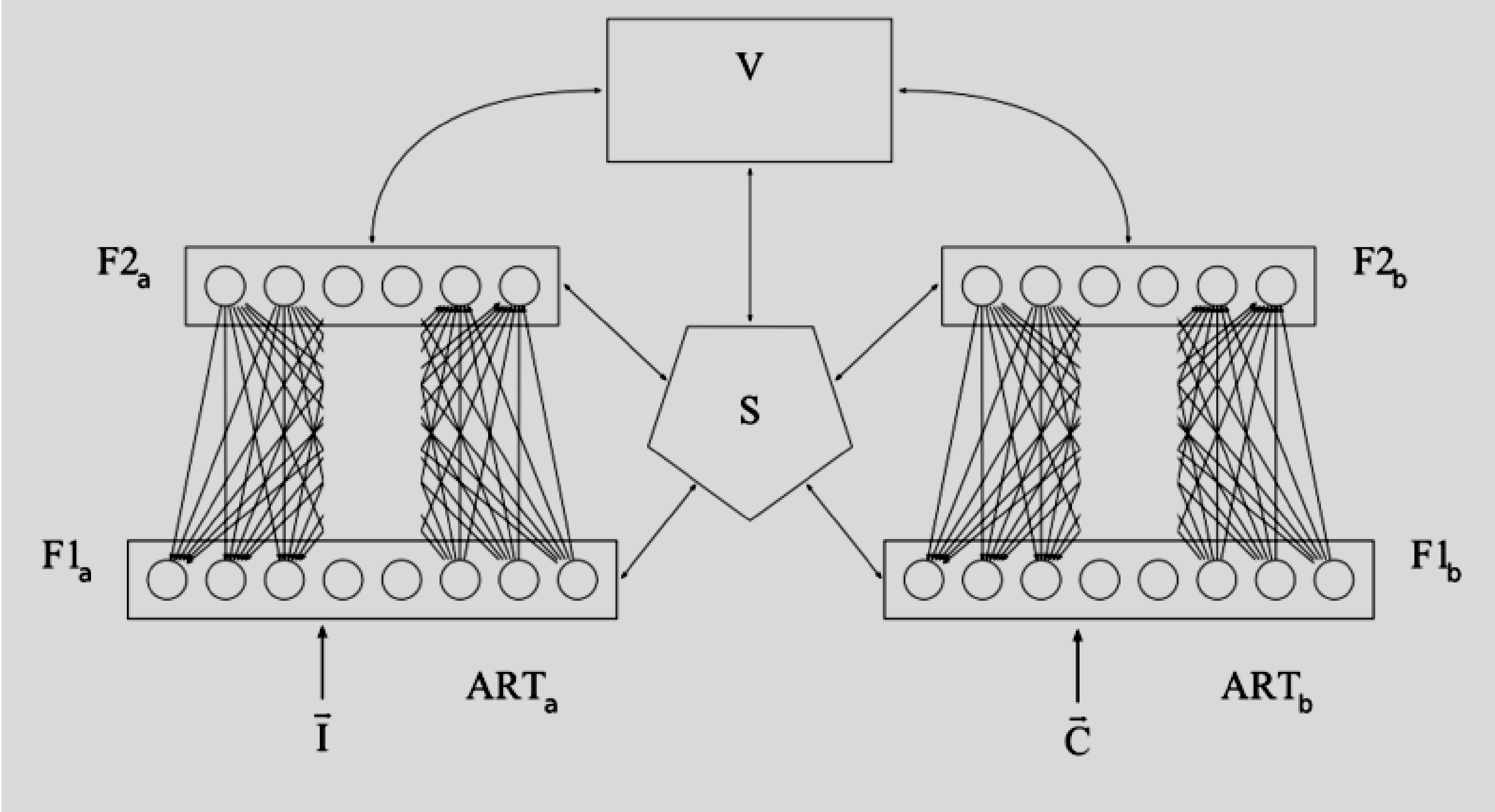




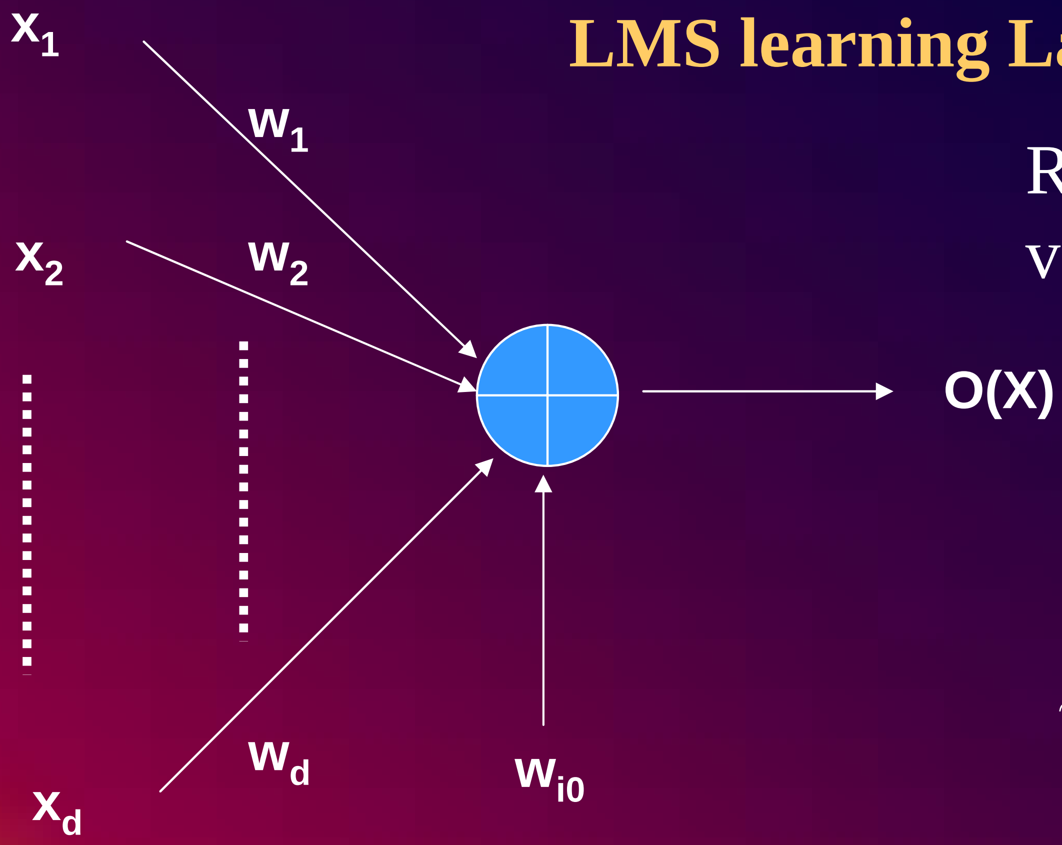
ART-NET (Grossberg model)

ARTMAP, also known as Predictive ART, combines two slightly modified ART-1 or ART-2 units into a supervised learning structure where the first unit takes the input data and the second unit takes the correct output data, then used to make the minimum possible adjustment of the vigilance parameter in the first unit in order to make the correct classification





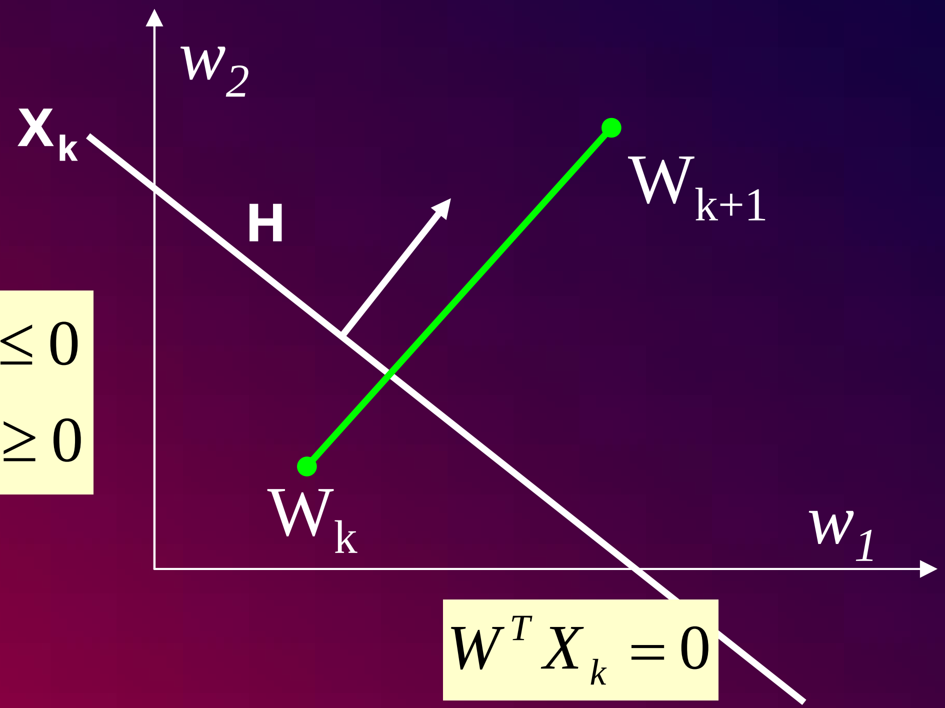
LMS learning Law in BPNN or FFNN models



Read about perceptron vs. multi-layer feedforward network

$$W_{k+1} = \begin{cases} W_k + \eta_k X_k & \text{if } X_k^T W_k \le 0 \\ W_k & \text{if } X_k^T W_k \ge 0 \end{cases}$$

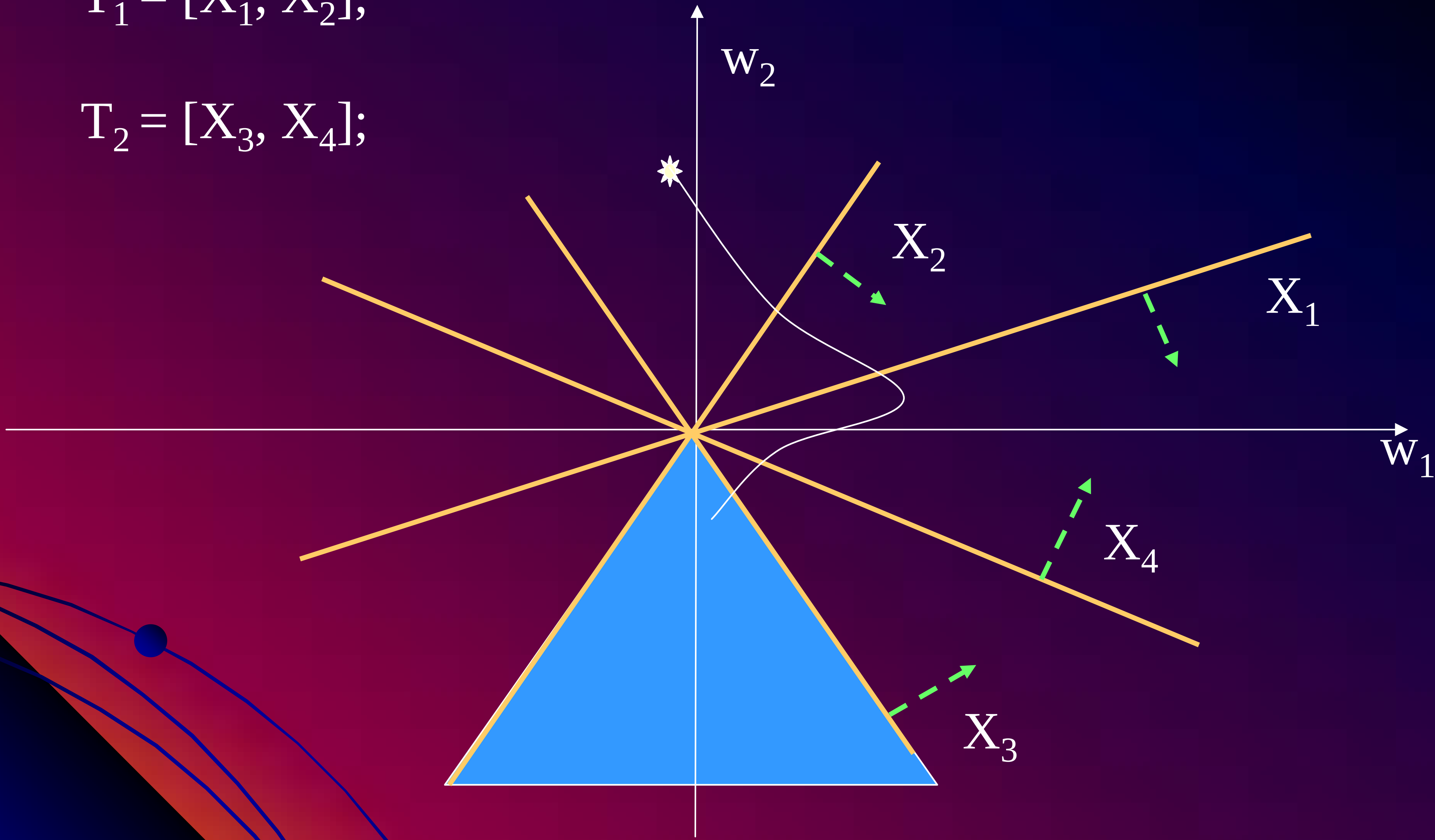
η_κ is the learning rate parameter



$$W_{k+1} = \begin{cases} W_k + \eta_k X_k & \text{if } \ X_k \in X_1 \text{ and } X_k^T W_k \le 0 \\ W_k - \eta_k X_k & \text{if } \ X_k \in X_0 \text{ and } X_k^T W_k \ge 0 \end{cases}$$

$T_1 = [X_1, X_2];$

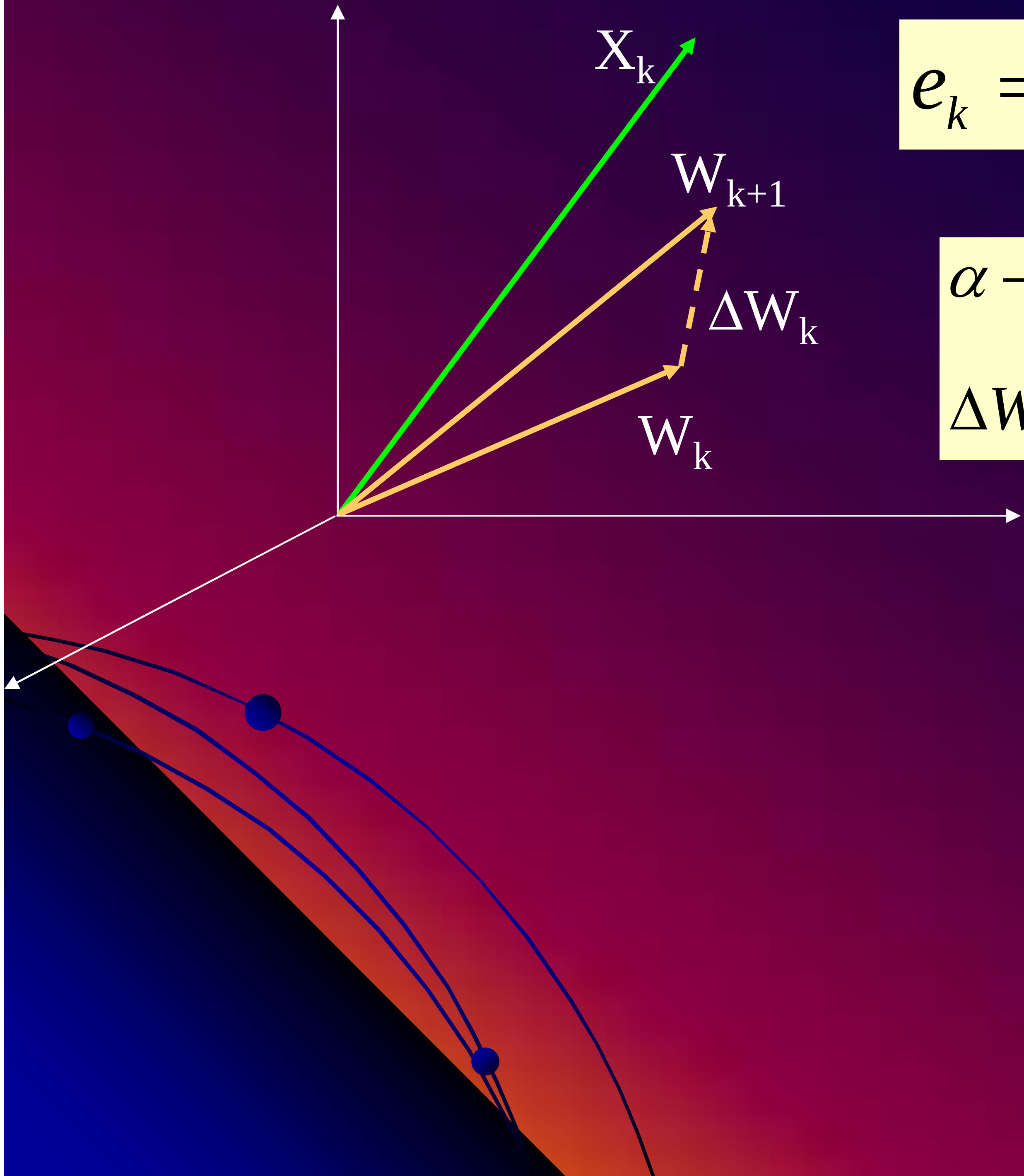
$T_2 = [X_3, X_4];$



η_κ decreases with each iteration

$$W_{k+1} = \begin{cases} W_k + \eta_k X_k & \text{if} \quad X_k \in \mathrm{X}_1 \text{ and } X_k^T W_k \le 0 \\ W_k - \eta_k X_k & \text{if} \quad X_k \in \mathrm{X}_0 \text{ and } X_k^T W_k \ge 0 \end{cases}$$

In case of FFNN, the objective is to minimize the error term:



$$e_k = d_k - s_k = d_k - X_k^T W_k$$

$\alpha - LMS$ Learning Algorithm:

$$\Delta W_k = \eta e_k \hat{X}_k$$

ADALINE (Adaptive Linear Neuron or later Adaptive Linear Element) **is a single layer neural network. It consists of a weight, a bias and a summation function.**

Original Delta Rule, for iterative ADALINE training :

$\Delta w_{ij} = \alpha(t_j - y_j)x_i;$

X - Input Training vector;

T - target Output for X;

α - Learning Rate.

$$y_j = \sum_i x_i w_{ij}$$

MSE error surface:

$$\xi_k = \frac{1}{2}[d_k - X_k^T W_k]^2 = E/2 - P^T W + (1/2) W^T R W.$$

$$P^T = E[d_k X_k^T];$$

$$R = E[X_k X_k^T] = E[\begin{bmatrix} 1 & x_1^k & & x_n^k \\ x_1^k & x_1^k x_1^k & & x_1^k x_n^k \\ & & & \\ x_n^k & x_n^k x_1^k & & x_n^k x_n^k \end{bmatrix}$$

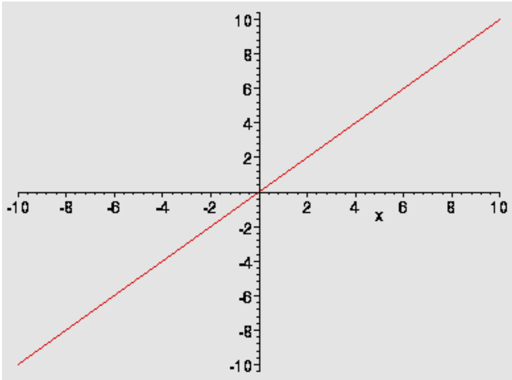
$$\nabla \xi = \left(\frac{\delta\xi}{\delta w_0}, \frac{\delta\xi}{\delta w_1}, \ldots\ldots, \frac{\delta\xi}{\delta w_n}\right)^T = -P + RW$$

Thus,

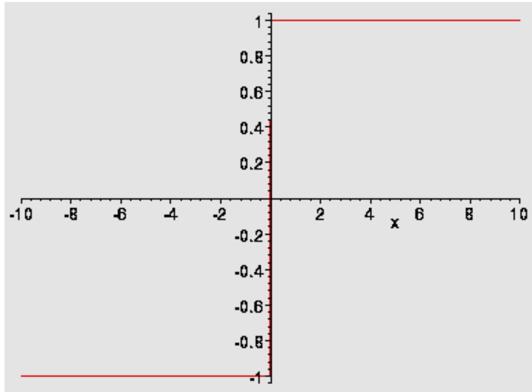
$$\hat{W} = R^{-1} P$$

Introduction to Artificial Neural Networks

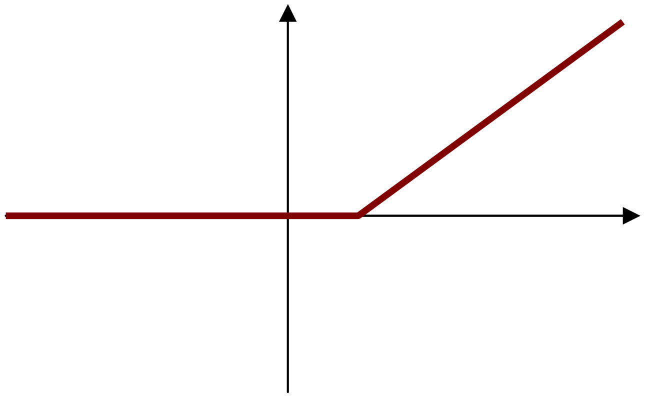
Different Types of Activation Functions



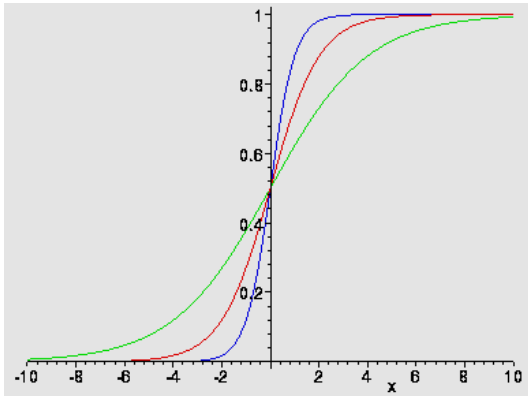
Linear Activation Function



Binary Activation Function



Threshold Linear Activation Function



Sigmoidal Activation Function

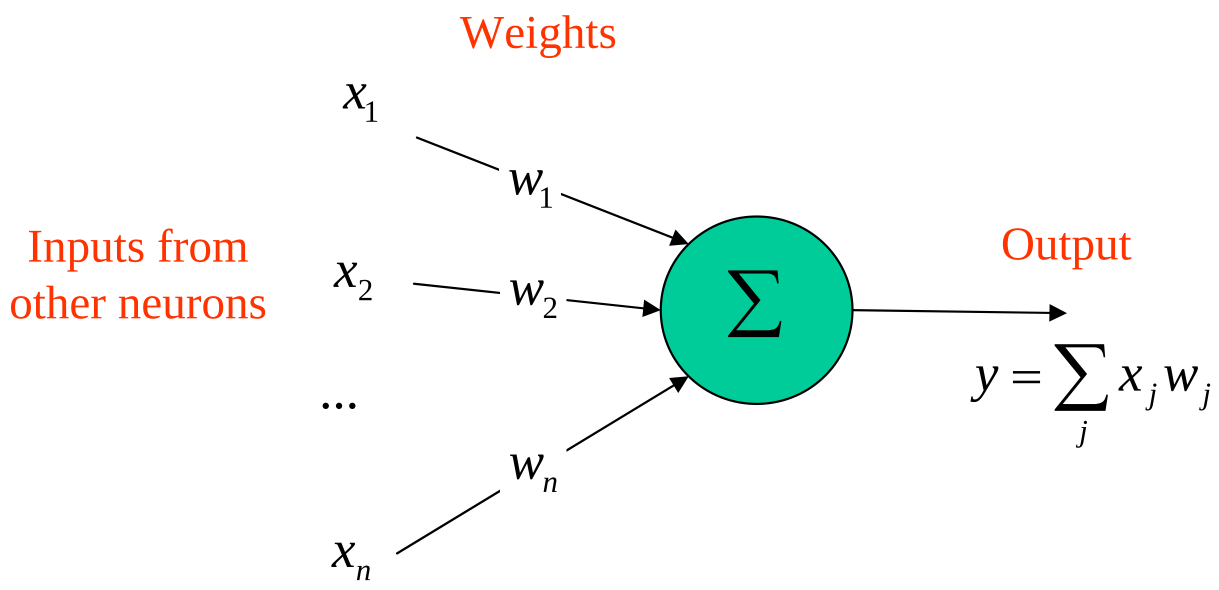
$$y_i = f\left(h_i\right) \quad \text{where} \quad h_i = \sum_j x_j w_{ij}$$

w_{ij} : strength of the synaptic from input axion(connection) j to output dendrite of receiving neuron i

Neural Network Architectures

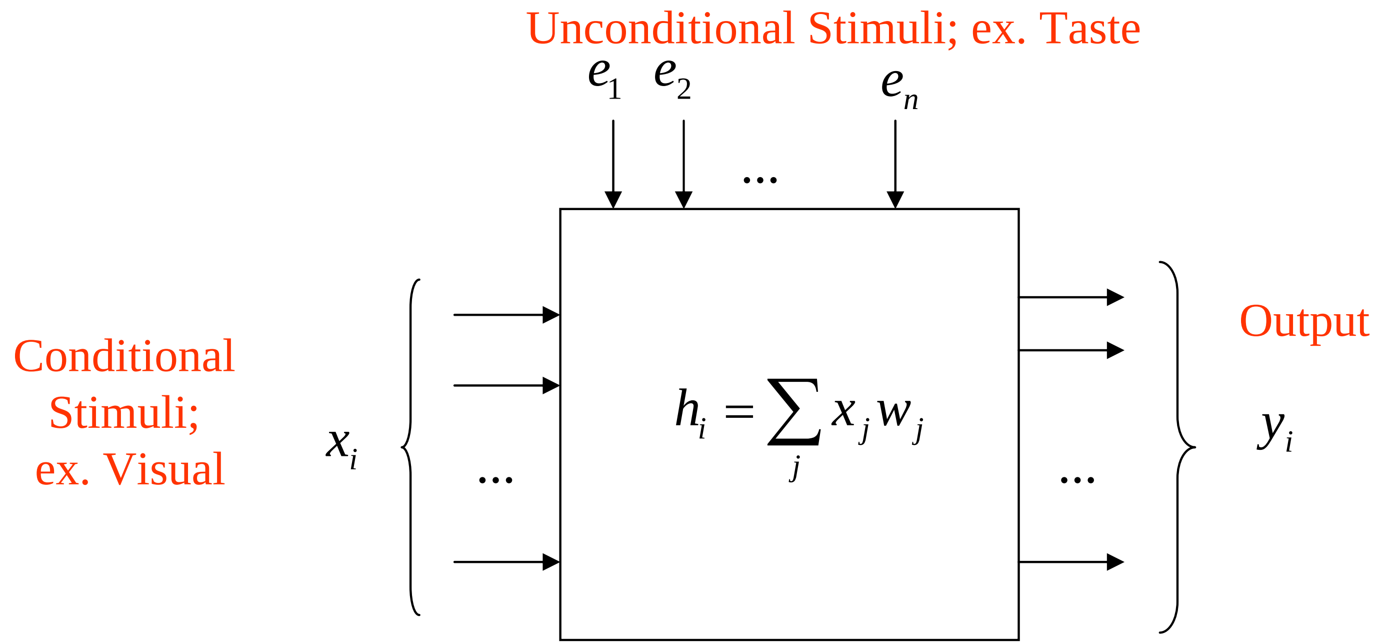
- **Perceptron Architecture**
- **Pattern Association Architecture**
- **Auto-Association Architecture**

Perceptron Architecture



Neural Network Architectures (Cont.)

Pattern Association Architecture



- **Associate first stimuli with a second**
- **Assumption : The two stimuli occur simultaneously**

Food $\longrightarrow$ ~~Taste~~ $\longrightarrow$ Saliva

After Learning: Retrieve the second stimulus when only the first is present

Synaptic Modification : Learning

- **Learning involves synaptic modification**
- **Must produce a desired output from a particular input**

HEBB's Rule

$$\delta w_{ij} = \alpha y_i x_j$$

x_i : pre-synaptic firing

y_i : post-synaptic firing

α : Learning Rate Parameter

- **Captures Long Term Potentiation (LPT) of the brain**
- **Applications : LAM, BAM, CAM, Hopfield, Cohen-Grossberg**

HEBB's Learning Law

- **Initially,** $y_i = f(e_i)$
- **Assume some initial random values for** $w_{ij}^{s} \cong 0$

$$h_i = \sum_j x_j w_{ij} \qquad \text{OR} \qquad h_i = \left\langle \vec{x} \,.\, \vec{w}_i \right\rangle$$

$$y_i = f(h_i)$$

- **To learn, now use HEBB's rule**

$$\delta w_{ij} = \alpha y_i x_j \qquad \text{OR} \qquad \delta \vec{w}_i = \alpha y_i \vec{x}$$

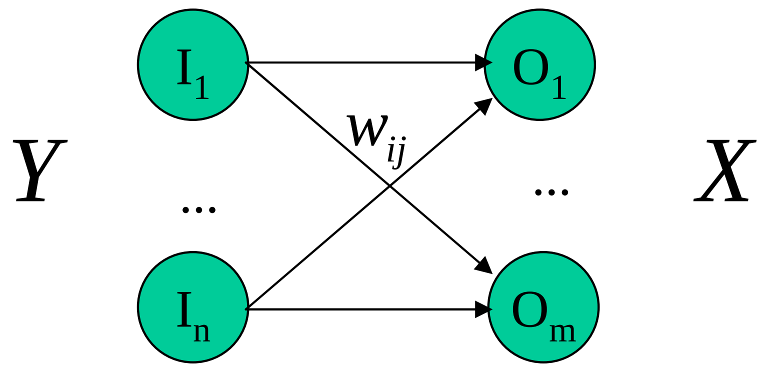
- **The inner product is max. (h_i), when both** $\overleftarrow{x}$ **and** $\overleftarrow{w}$ **are same**
- **Thus, *h* will be large, if** $\overleftarrow{x}$ **and** $\overleftarrow{w}$ **are similar**

ANN Theory

- **Let** $\alpha = 1$

Thus, $\delta \vec{w}_i = y_i \vec{x}$

Thus, $W = \left[w_{ij}\right] = YX^T$



$$X^T = Y^T W$$

- **From the above two equations,**

$$X'^T = Y^T W = Y^T Y X^T = \|Y\| X^T$$

- **Thus, if Y is normalized, then we have a *Perfect Recall***

$$X'^T = X^T$$

Hebb's Rule for multiple patterns:

Say, want to store a set of associations:

$$S(p):T(p),\ p=1,2,\ldots.,P;$$

$$S(p)=[s_1(p),s_2(p),\ldots\ldots,s_n(p)];$$

$$T(p)=[t_1(p),t_2(p),\ldots\ldots,t_m(p)];$$

The Weight matrix W_{n*m}, is given by (for training):

$$w_{ij}=\sum_{p=1}^{P}s_i(p)t_j(p)=\sum_{p=1}^{P}S^T(p)T(p)$$

Say, S(*p*)'s are orthogonal during training):

$$S(k).S^T(p)=0,\ \text{for k} \neq \text{p}$$

During testing for a vector, X=S(*k*), the response is :

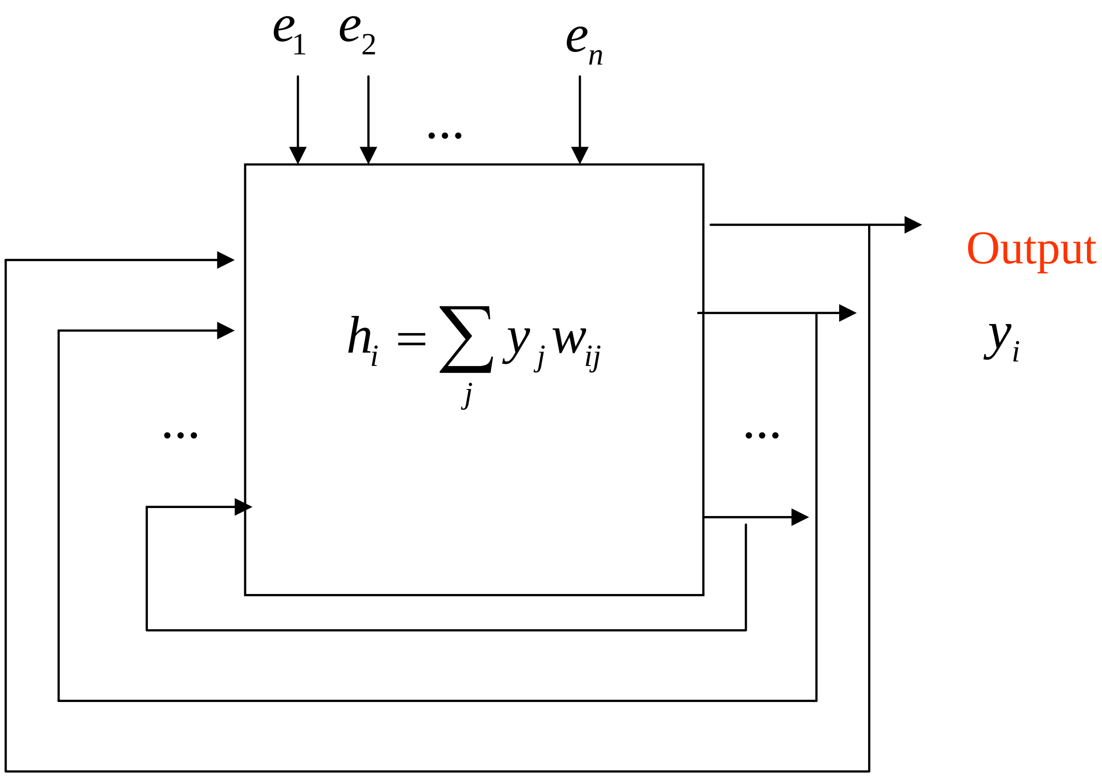
$$S(k)W=\sum_{p=1}^{P}S(k)S^T(p)T(p)$$

$$=S(k)S^T(k)T(k)$$

= T(*k*), if S(*k*)'s are othronormal

Else, we have "Cross-talk"

Neural Network Architectures (Cont.)



Auto Association Architecture

$$\delta w_{ij} = \alpha y_i y_j$$

Autoassociative memory, also known as auto-association memory or an autoassociation network, is a form of backpropagation or other neural networks that enables one to retrieve entire memories from only a tiny sample of itself.

e.g.: To draw a picture, present the fragments that's necessary to retrieve the appropriate memory.

Auto-association is a means by which a neural network communicates that it does recognize the pattern that was presented to the network.

A neural network that supports autoassociation will pass a pattern directly from its input neurons to the output neurons. No change occurs;

In case of successful recognition an autoassociative neural network simply passes the date from input neurons to the output neurons.

Failed pattern recognition results in anything but the input neurons passing directly to the output neurons. If the pattern recognition fails, some other pattern will be presented to the output neurons.

If the input-output pair of patterns are from different spaces, we need a **hetero-associative network/memory**.

Original Delta Rule,

for iterative ADALINE training :

$\Delta w_{ij} = \alpha(t_j - y_j)x_i;$

X - Input Training vector;

T - target Output for X;

α - Learning Rate.

$$y_j = \sum_i x_i w_{ij}$$

Extended Delta Rule,

for iterative ADALINE training :

$\Delta w_{ij} = \alpha(t_j - y_j)x_i f'(y\_in_j);$

$y_j = f'(y\_in_j)$

$$y\_in_j = \sum_i x_i w_{ij}$$

Most structures – SOM, RBFNs etc., use Hebb's law for training

Read about

- **Hetero-association**
- **Generalized Delta Rule**
- **CAM**
- **BAM etc.**
- **Neuro-dynamics**
- **Spike Neural models**

Other properties of ANN:

- **Generalization**
- **Graceful degradation**
- **Speed**
- **Capacity**

Gestalt Theory and Image Analysis

Key laws of Gestalt (revisited):

- **Grouping**

- **Illusions - Foreground/background distinction**

- **Collaboration in grouping (proximity, closure, continuity, similarity)**

- **Conflicts - in two groupings, one dominate**

- **Masking (information in noise, texture etc.)**

Basic principles in CV (see CS 636 notes):

Principle I (Shannon):

Any Image or Signal (including noise), is a band-limited function sampled on a bounded, periodic grid.

- Digital images are often considered as trigonometric polynomials (representation-wise).

Principle II (Wertheimer's Contrast principle):

Image interpretation does not depend on actual values of the gray levels, but only on their relative values.

- see work on: upper and lower level sets to be independent of contrast

Principle III (Helmholtz and D. Lowe):

Gestalts are sets of points whose (geometric, regular) spatial arrangement could not occur in noise.

No. of false alarms in a noise Shannon image:

Let, N_s be the no. of segments joining pixels of the image.

Let, $0 \leq p \leq 1$ be an angular precision (arbitrary).

Let, S be a segment with length *l* and with *k* sample points aligned at precision *p*. Then No. of False Alarms is:

$$NFA(l,k,p) = N_s \sum_{j=k}^{l} \binom{l}{j} p^j (1-p)^j$$

An alignment is meaningful, if: $NFA(l,k,p) \leq 1$

A perceptual boundary is defined as a level line whose points have a "large enough" gradient, so that no such line is likely to occur in a white noise with the same overall contrast.

Using NFA, a few principles of Gestalt and Hemholtz logics have been analyzed and algorithms designed:

- **Alignment in Digital images (area and no. of points in the strip)**

- **Maximum meaningful alignments and exclusion principle (No. of aligned points and no. of points in the segment)**

- **Modes, intervals and gaps in a histogram**

- **Vanishing points (no. of lines meeting)**

- **Contrasting and meaningful boundaries (minimum contrast and length of the curve; snakes vs. edges)**

- **Clusters (area of thick bounding curve, region and no. of points)**

- **Binocular Vision**

Read about (some of these are basics):

- **ε-meaningful;**
- **Stochastic Geometry**
- **Binomial law**
- **Gaussian and other probabilities**
- **Hypothesis testing**
- **K-L Divergence**
- **8-point DLT algorithm and Fundamental Matrix**
- **PCA, LDA and ICA**

Other properties of the brain:

- **Visual Attention**
- **Visual Search**
- **Outputs of Brain processing: to short-term and long-term memory; reward association; emotion; motivation; selection and action; search etc.**
- **Multimodal representation**
- **Conscious perception**
- **Integrated approached**