Unit:1
Introduction to ANN

ANN

- An artificial neuron network (neural network) is a computational model that mimics the way nerve cells work in the human brain.
- An artificial neural network has three or more layers that are interconnected.
- The first layer consists of input neurons. Those neurons send data on to the deeper layers, which in turn will send the final output data to the last output layer.

History of Neural Network

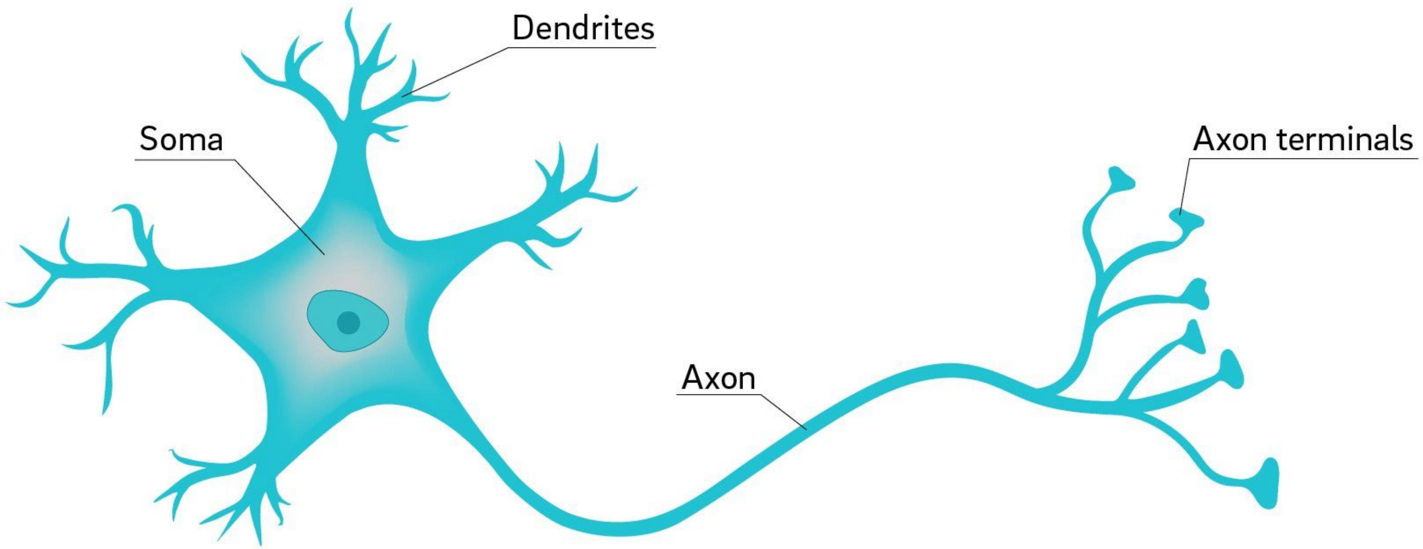
- Neural networks were **first proposed in 1944 by Warren McCullough and Walter Pitts**, two University of Chicago researchers

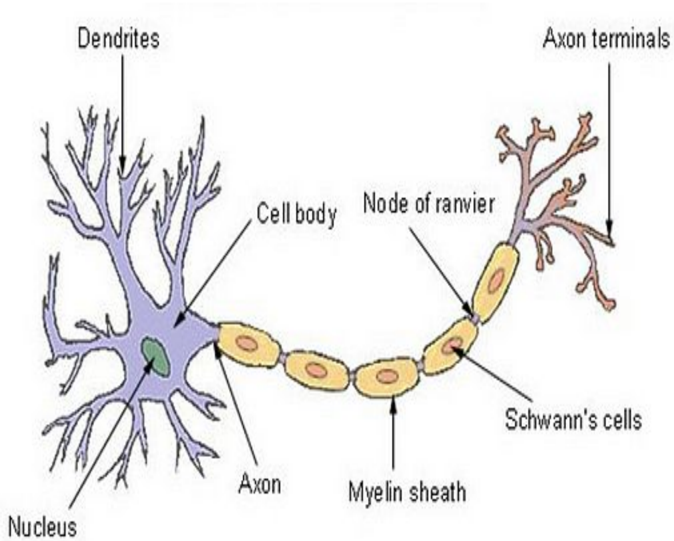
- This model is segmented in two parts

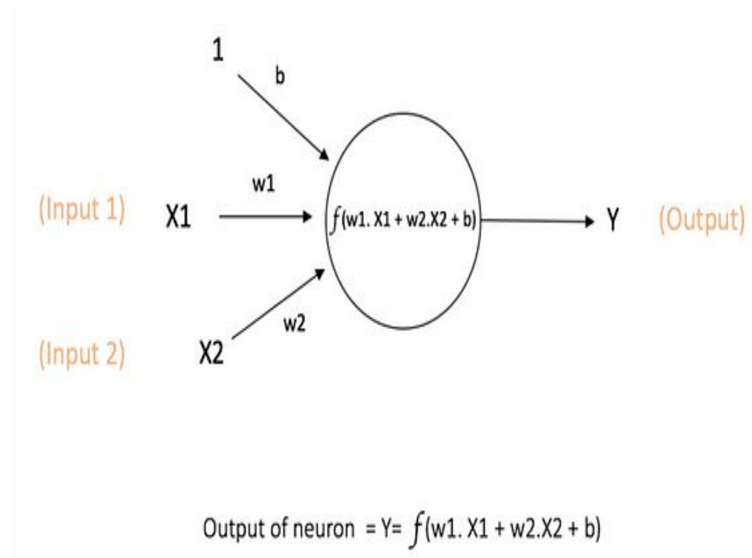
1. A summation over-weighted inputs.
2. An output function of the sum.

Biological Neural Network

Neuron







BNN

ANN

The input to each neuron are like the dendrites

Dendrites
- It takes input from other neurons in form of an electrical impulse

Cell Body
- It generates inferences from those inputs and decide what action to take

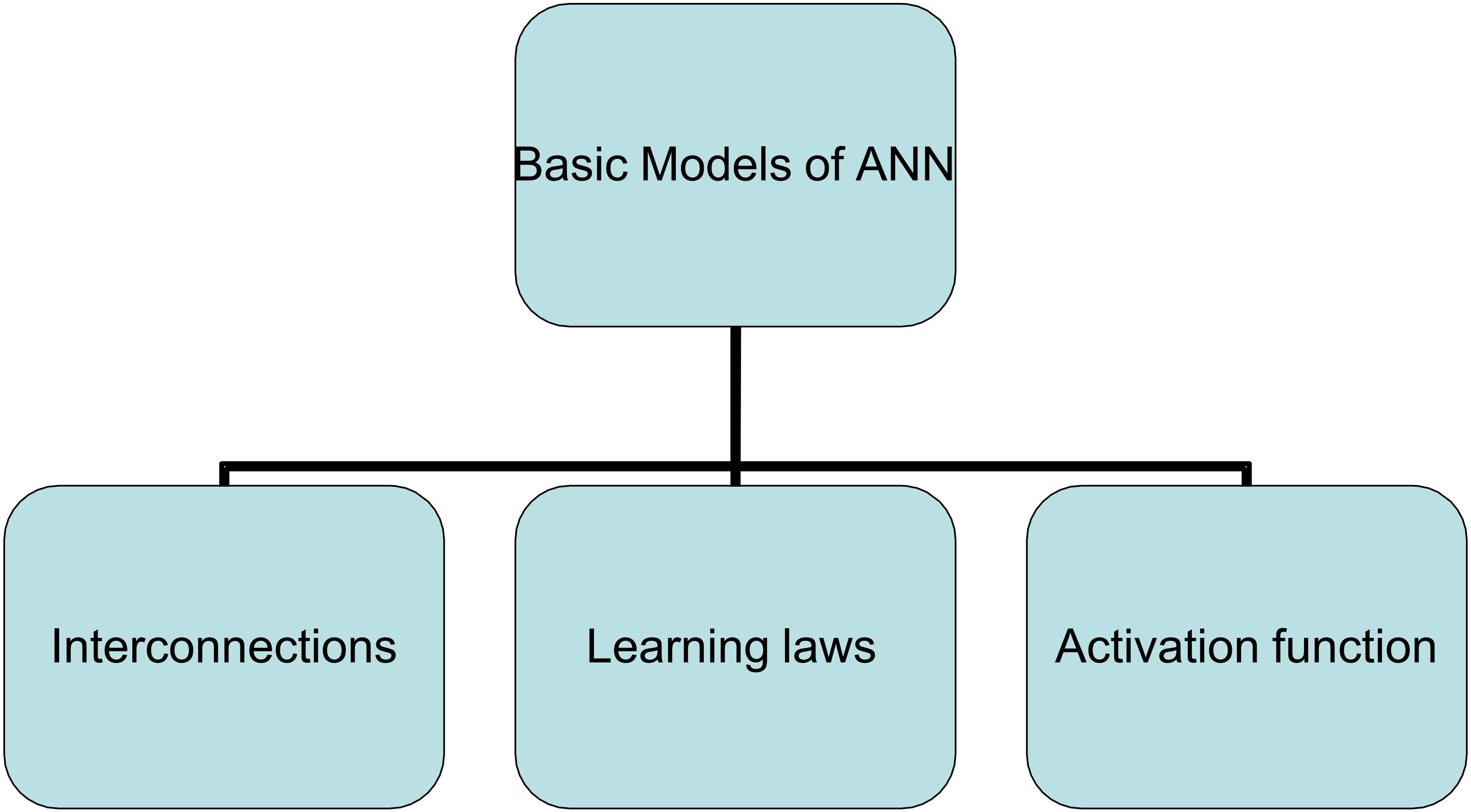
Axon terminals
- It transmit outputs in the form of electrical impulse

Input Layer
- The training observations are fed through these neurons

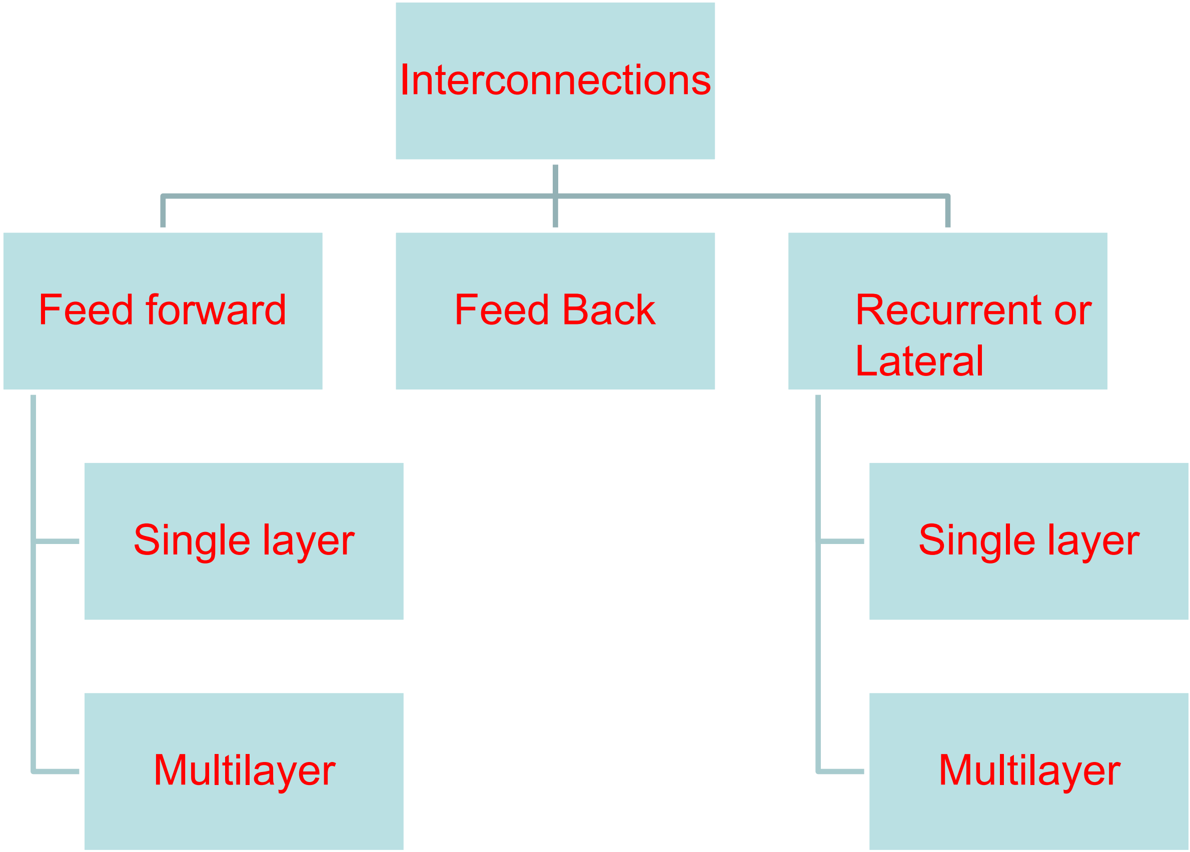
Hidden Layers
- Intermediate layers between input and output which help the Neural Network learn the complicated relationships involved in data

Output Layer
- The final output is extracted from previous two layers

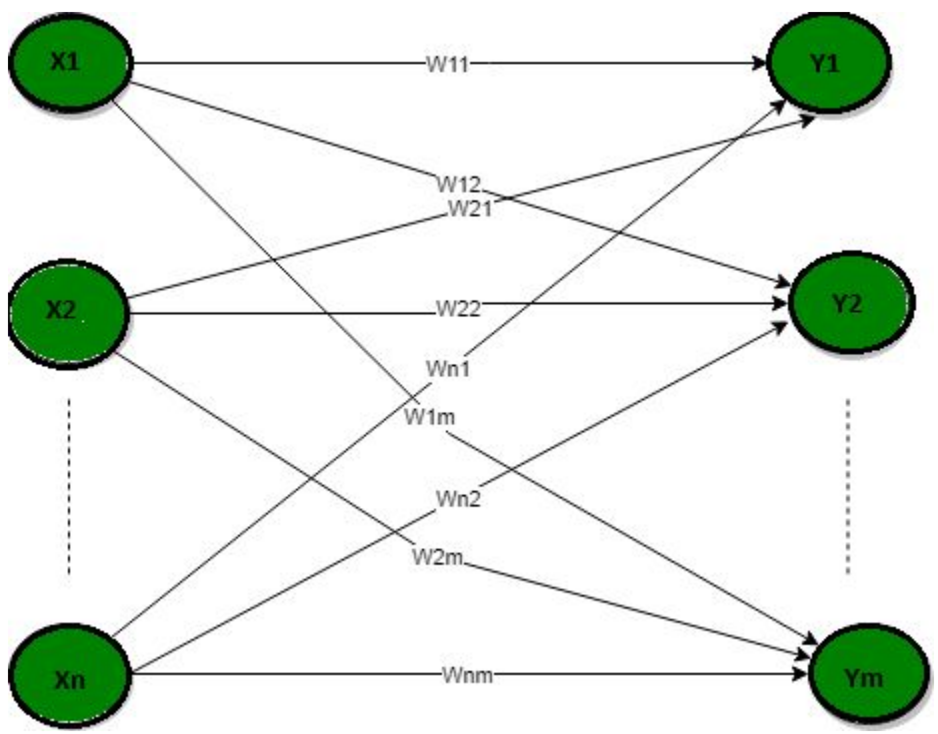
Basic models of ANN



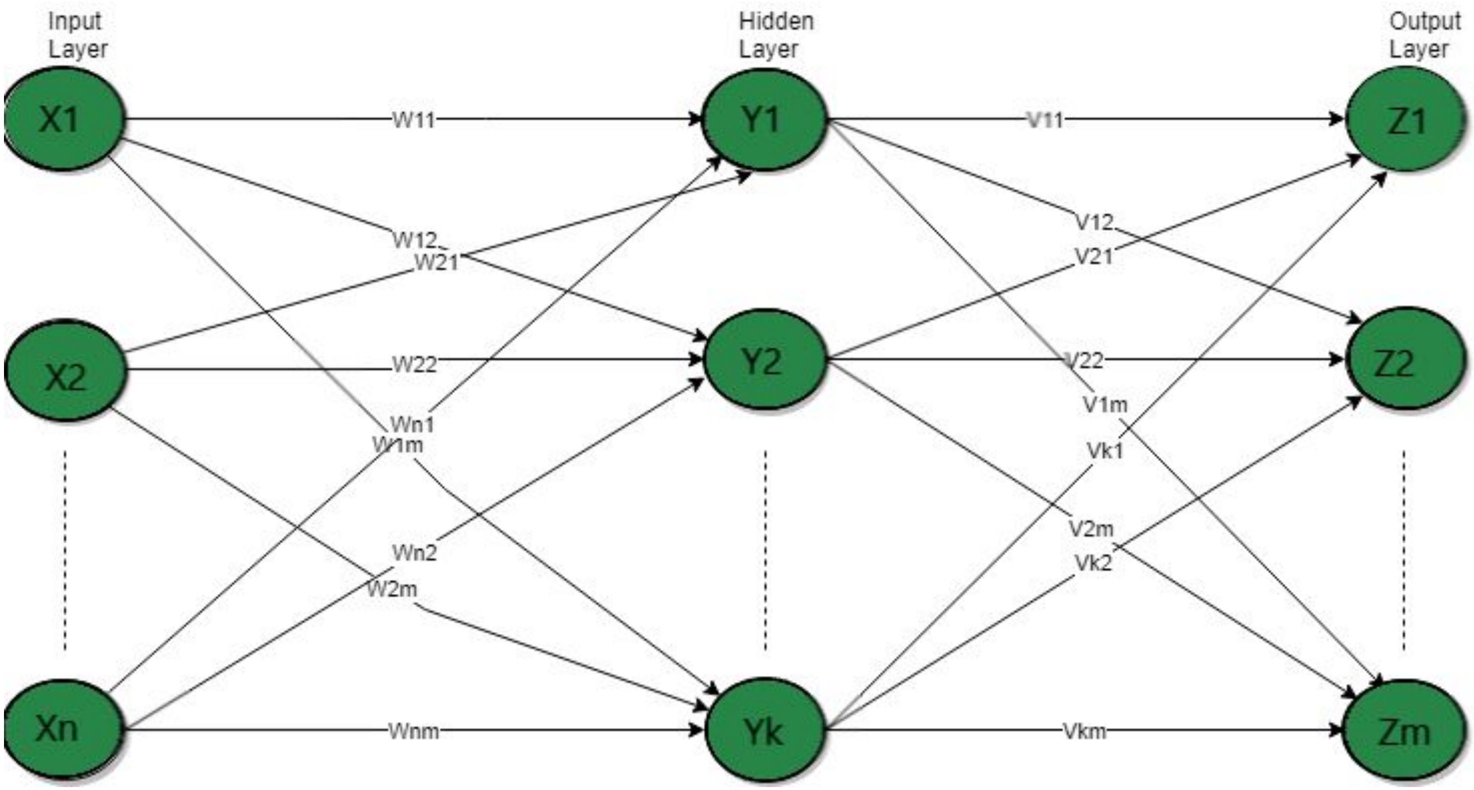
Topology of neural network architecture



Single layer Feedforward Network

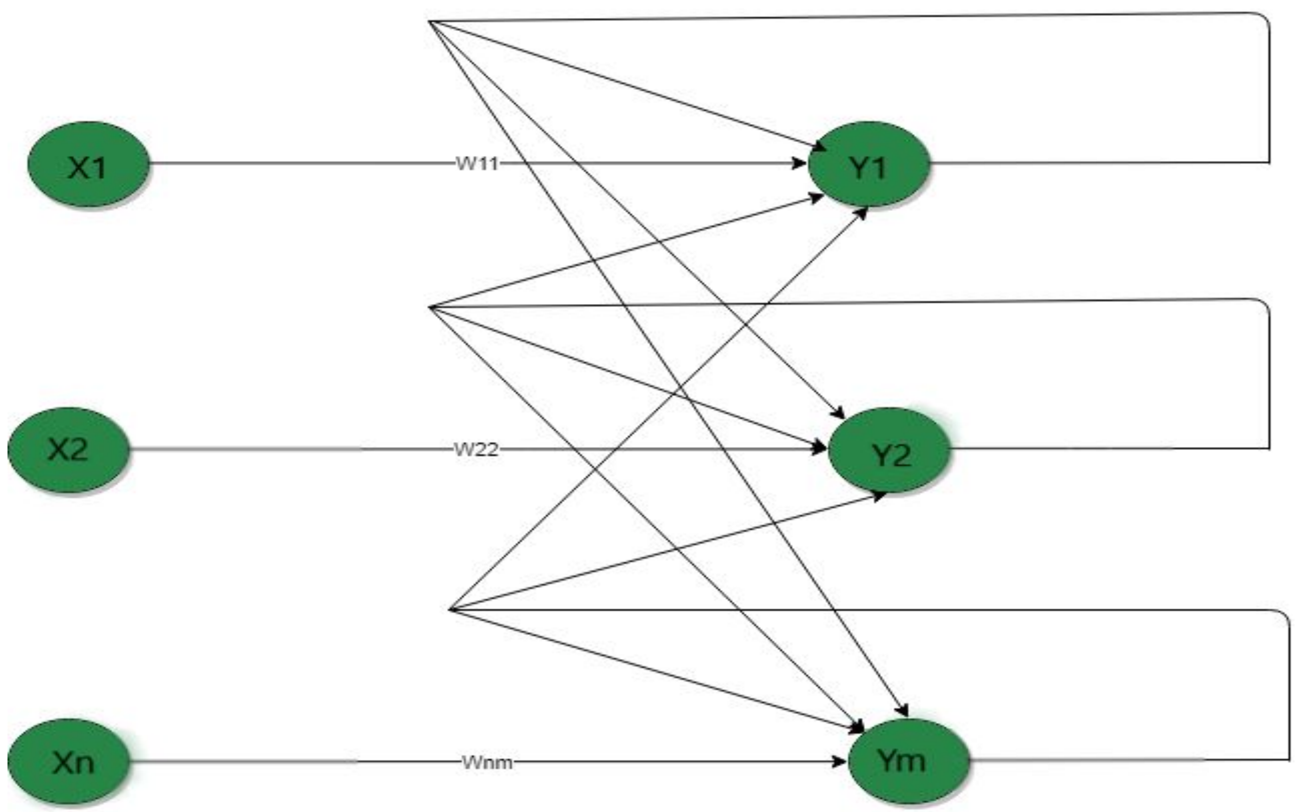


Multilayer feed forward network



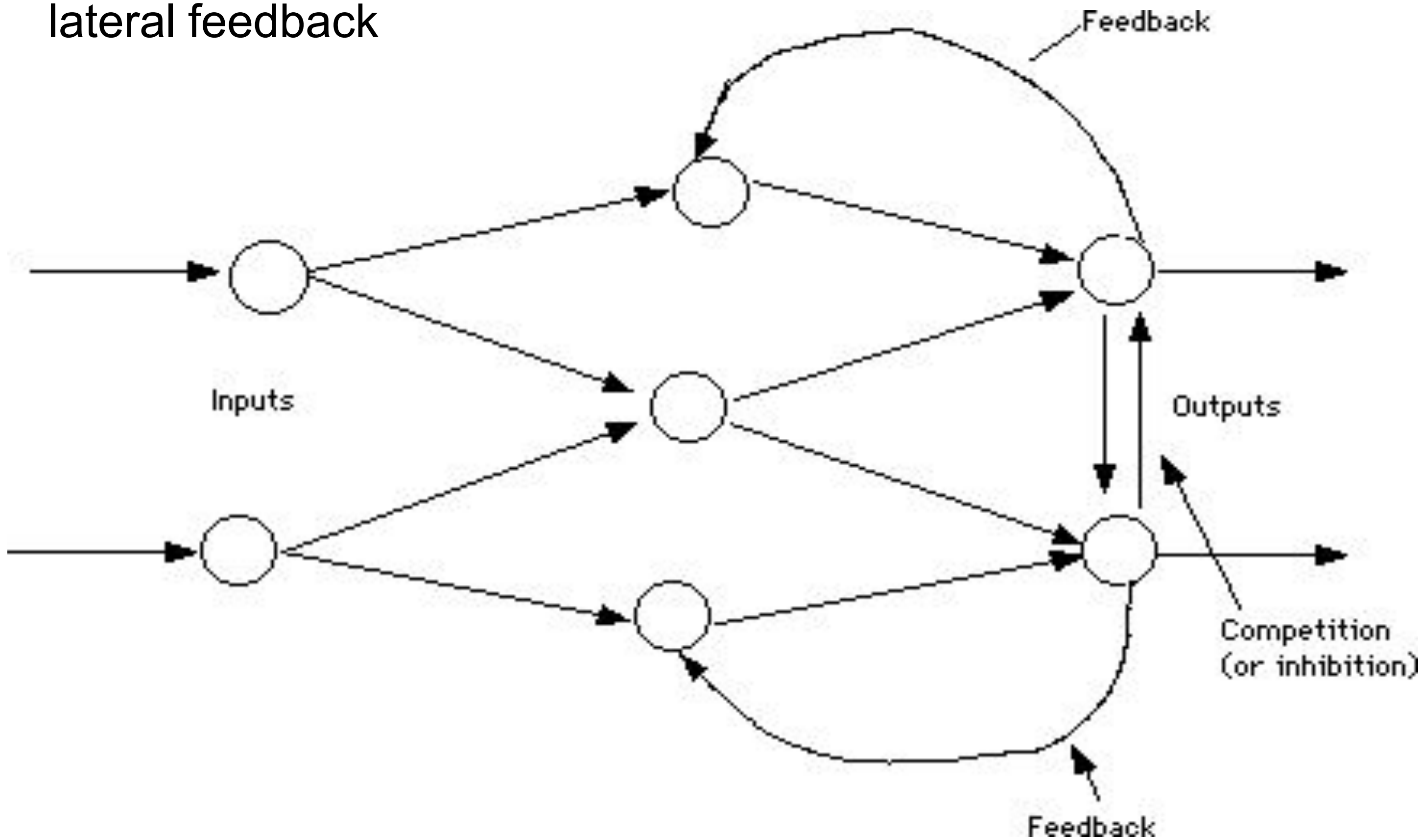
Feedback network

When outputs are directed back as inputs to same or preceding layer nodes it results in the formation of feedback networks



Lateral feedback

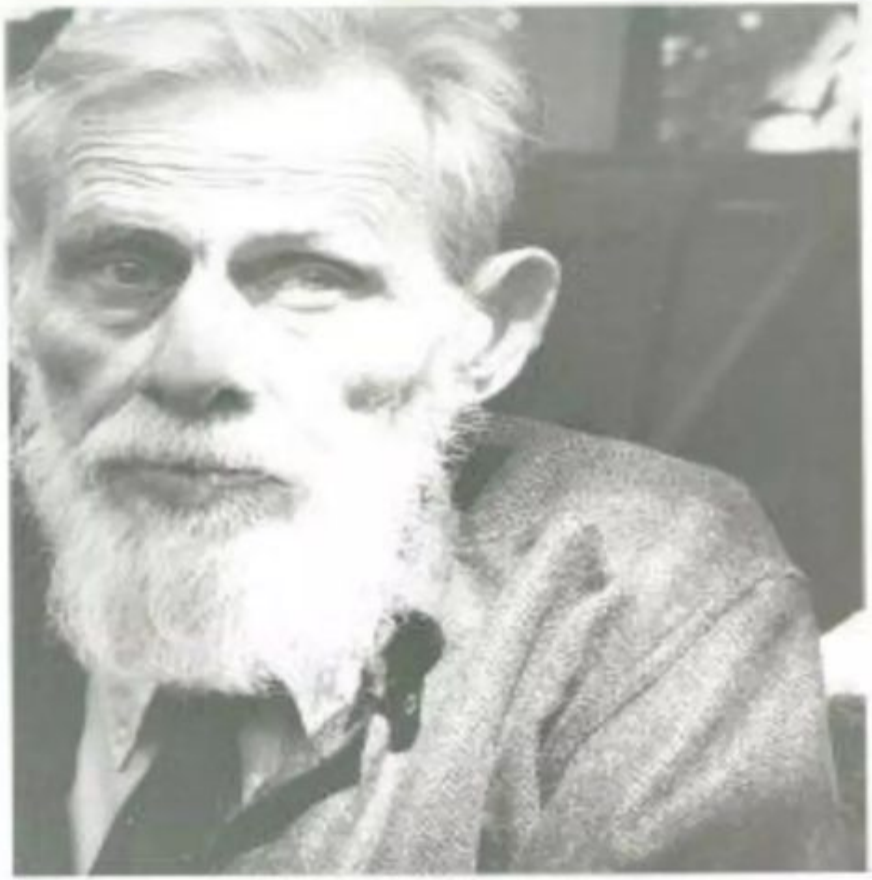
If the feedback of the output of the processing elements is directed back as input to the processing elements in the same layer then it is called lateral feedback



Activation Functions

- Refer to document attached Activation functions.pdf

Mc Culloch & Pitts model



Warren S. McCulloch

The McCulloch-Pitts Neuron was the earliest neural network discovered in 1943.

The weights associated with communication links may be excitatory (weights are positive) or inhibitory (weights are negative)

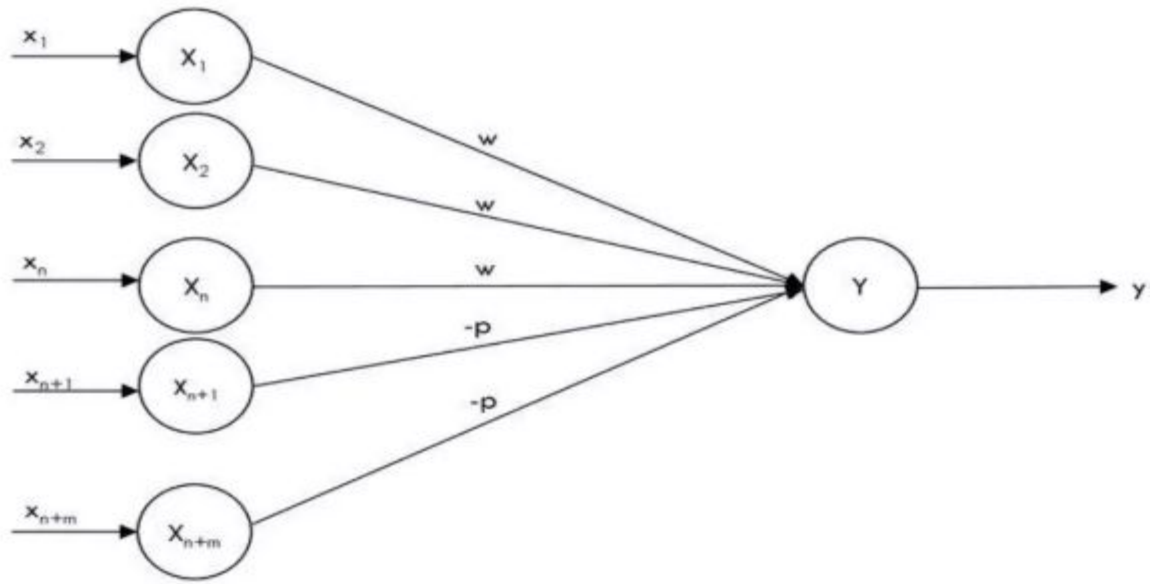
A threshold activation function plays a major role in the M-P neuron. If the net input is greater than the sum of the input signals, the neuron fires.

Mostly used in the logic functions.

The activation of M-P neuron is always binary in nature.

It consists of a number of input units connected to a single output unit.

McCULLOCH-PITTS NEURON: ARCHITECTURE (MP NEURON MODEL)



Salient features of McCulloch-Pitts artificial neuron

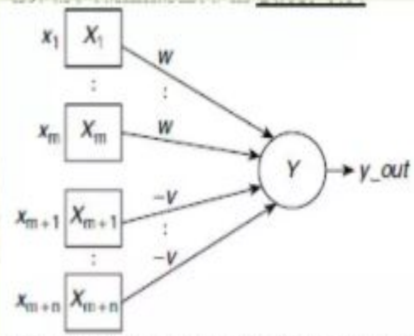


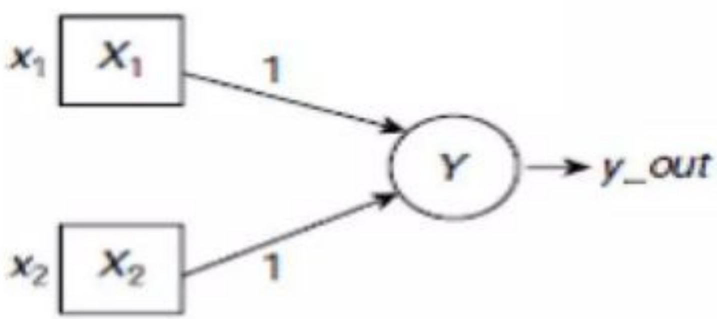
Fig. 6.11. Structure of a McCulloch-Pitts neuron

1	There are two kinds of input units, *excitatory*, and *inhibitory*. In Fig. 6.11 the excitatory inputs are shown as inputs $X_1, \ldots, X_m$ and the inhibitory inputs are $X_{m+1}, \ldots, X_{m+n}$. The excitatory inputs are connected to the output unit through positively weighted links. Inhibitory inputs have negative weights on their connecting paths to the output unit.
2	All excitatory weights have the same positive magnitude w and all inhibitory weights have the same negative magnitude $-v$.
3	The activation $y\_out = f(y\_in)$ is binary, i.e., either 1 (in case the neuron fires), or 0 (in case the neuron does not fire).
4	The activation function is a binary step function. It is 1 if the net input y_in is greater than or equal to a given threshold value θ, and 0 otherwise.
5	The inhibition is absolute. A single inhibitory input should prevent the neuron from firing irrespective of the number of active excitatory inputs.

Example-4: M-P Neuron to implementation the logical AND operation

(a) Truth Table

x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1



(b) Neural structure

$$y\_in = x_1 + x_2$$

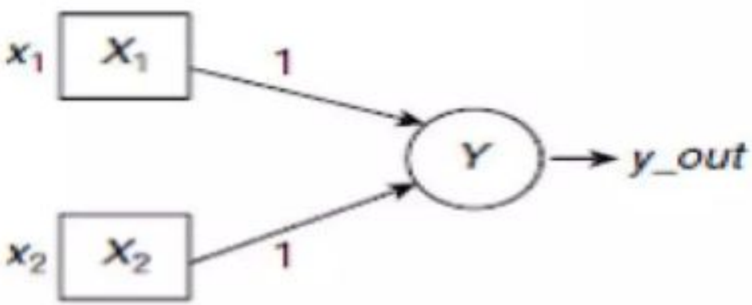
$$y\_out = f(y\_in) = \begin{cases} 1, & \text{if } y\_in \geq 2, \\ 0, & \text{otherwise.} \end{cases}$$

(c) Activation function

Example-5: M-P Neuron to implementation the logical OR operation

(a) Truth Table

x_1	x_2	x_1 OR x_2
0	0	0
0	1	1
1	0	1
1	1	1



(b) Neural structure

$$y\_in = x_1 + x_2$$

$$y\_out = f(y\_in) = \begin{cases} 1, & \text{if } y\_in \geq 1, \\ 0, & \text{otherwise.} \end{cases}$$

(c) Activation function

Features of McCulloch-Pitts model

- Allows binary 0,1 states only
- Operates under a discrete-time assumption
- Weights and the neurons’ thresholds are fixed in the model and no interaction among network neurons
- Just a primitive model

What is Perceptron?

Perceptron is a type of neural network that performs binary classification that maps input features to an output decision, usually classifying data into one of two categories, such as 0 or 1.

- Perceptron consists of a single layer of input nodes that are fully connected to a layer of output nodes. It is particularly good at learning linearly separable patterns.
- It utilizes a variation of artificial neurons called Threshold Logic Units (TLU), which were first introduced by McCulloch and Walter Pitts in the 1940s.

Basic Components of Perceptron

- **Input Features:** The perceptron takes multiple input features, each representing a characteristic of the input data.

- **Weights:** Each input feature is assigned a weight that determines its influence on the output. These weights are adjusted during training to find the optimal values.

- **Summation Function:** The perceptron calculates the weighted sum of its inputs, combining them with their respective weights.

- **Activation Function:** The weighted sum is passed through the **Heaviside step function**, comparing it to a threshold to produce a binary output (0 or 1).

- **Output:** The final output is determined by the activation function, often used for **binary classification** tasks.

- **Bias:** The bias term helps the perceptron make adjustments independent of the input, improving its flexibility in learning.

- **Learning Algorithm:** The perceptron adjusts its weights and bias using a learning algorithm, such as the Perceptron Learning Rule, to minimize prediction errors.

How does Perceptron work?

- A weight is assigned to each input node of a perceptron, indicating the importance of that input in determining the output. The Perceptron's output is calculated as a weighted sum of the inputs, which is then passed through an activation function to decide whether the Perceptron will fire.
- The weighted sum is computed as:

$$z = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n = X^T W$$

- The step function compares this weighted sum to a threshold. If the input is larger than the threshold value, the output is 1; otherwise, it's 0. This is the most common activation function used in Perceptrons are represented by the Heaviside step function:

$$h(z) = \begin{cases} 0 & \text{if } z < \text{Threshold} \\ 1 & \text{if } z \geq \text{Threshold} \end{cases}$$

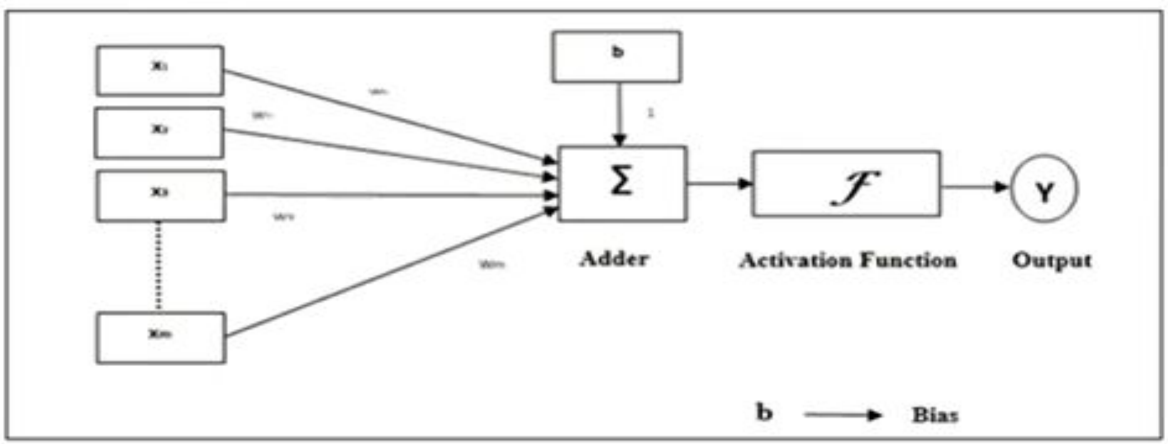
- A perceptron consists of a single layer of Threshold Logic Units (TLU), with each TLU fully connected to all input nodes

Example: Perceptron in Action

- Let's take a simple example of classifying whether a given fruit is an apple or not based on two inputs: its weight (in grams) and its color (on a scale of 0 to 1, where 1 means red). The perceptron receives these inputs, multiplies them by their weights, adds a bias, and applies the activation function to decide whether the fruit is an apple or not.
- Input 1 (Weight): 150 grams
- Input 2 (Color): 0.9 (since the fruit is mostly red)
- Weights: [0.5, 1.0]
- Bias: 1.5
- The perceptron's weighted sum would be:
- $(150*0.5)+(0.9*1.0)+1.5=76.4$
- *Let's assume the activation function uses a threshold of 75. Since 76.4 > 75, the perceptron classifies the fruit as an apple (output = 1).*

characteristics of the perceptron:

- It consists of a single neuron with an arbitrary number of inputs along with adjustable weights.
- The output of the neuron is 1 or 0 depending upon the threshold.
- It also consists of a bias which is always 1.



Difference between MP-neuron and Preceptron

- It overcomes some of the limitations of the M-P neuron by introducing the concept of numerical weights (a measure of importance) for inputs, and a mechanism for learning those weights.
- Inputs are no longer limited to boolean values like in the case of an M-P neuron, it supports real inputs as well which makes it more useful and generalized.

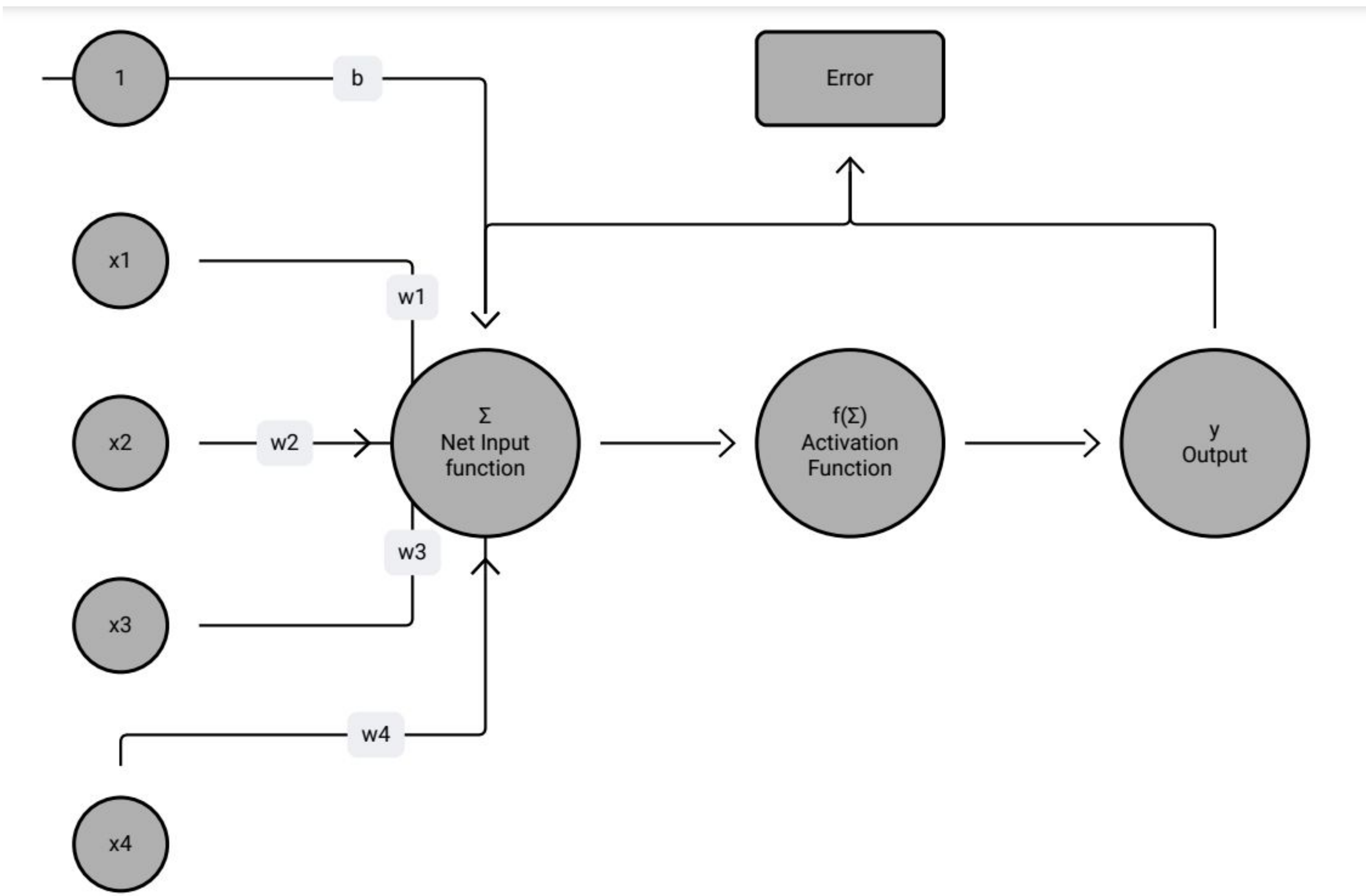
Adaline model

- ADALINE (Adaptive Linear Neuron) is **an early single-layer artificial neural network** and the name of the physical device that implemented this network.

- In Adaline, there is only one output unit and output values are bipolar (+1,-1).
- It uses the delta rule $w_i(new) = w_i(old) + (t - y_{in})x_i$

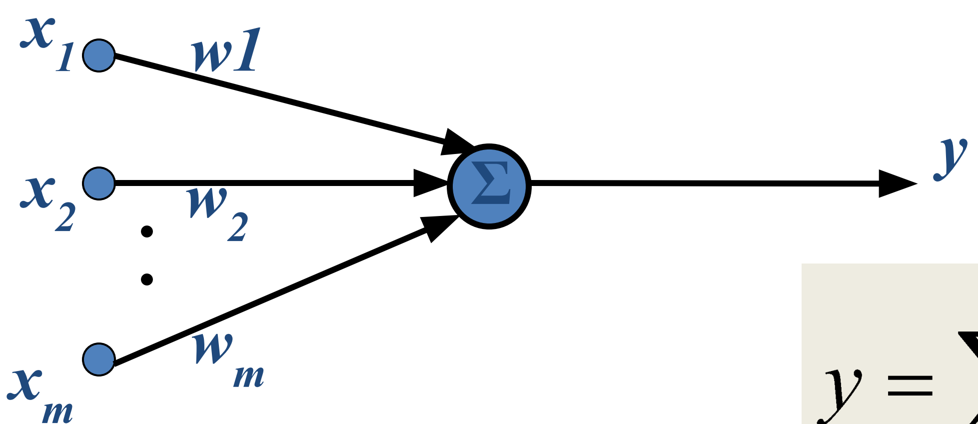
where w_i y_{in} and t are the weight predicted output, and true value respectively.

Adaline model



Adaline: Adaptive Linear Element

- The output **y** is a linear combination of the input **x:**



$$y = \sum_{j=0}^{m} x_j(n) w_j(n)$$

Adaline: Adaptive Linear Element (Cont'd)

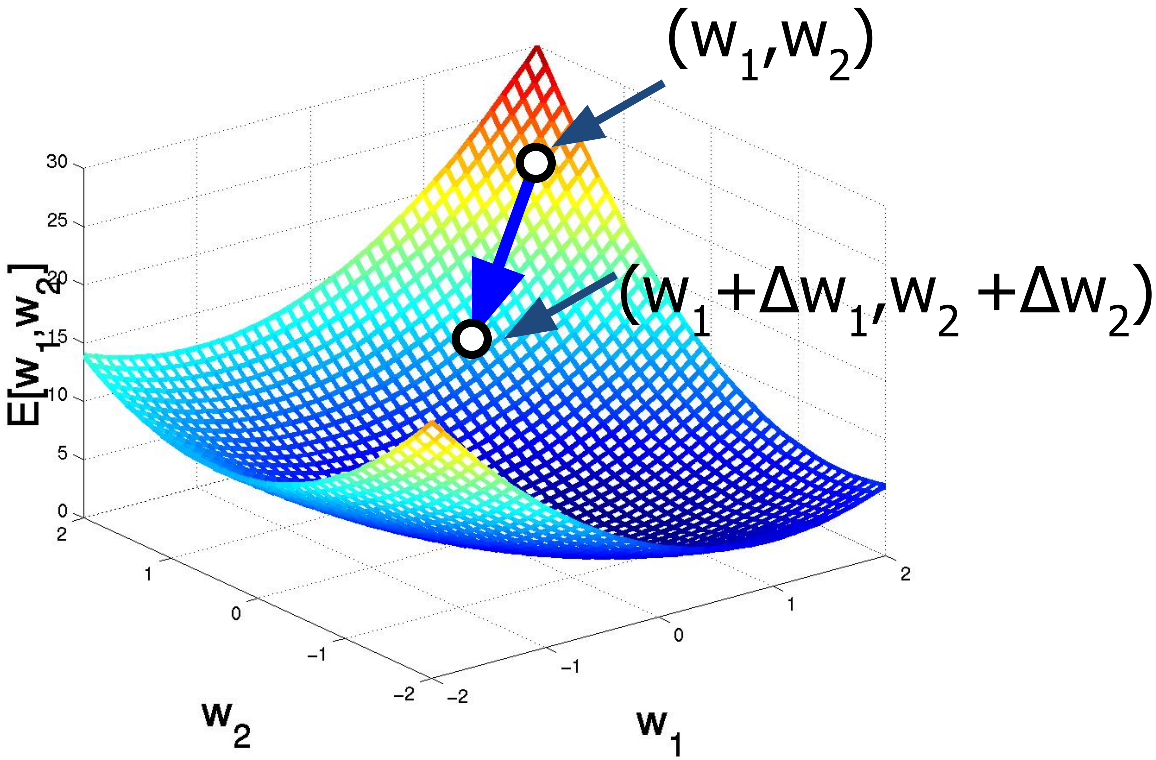
- Adaline: uses a linear neuron model and the Least-Mean-Square (LMS) learning algorithm

  The idea: try to minimize the square error, which is a function of the weights

$$\boxed{E(\mathrm{w}(\mathrm{n})) = \frac{1}{2}\mathrm{e}^2(\mathrm{n})}$$

$$\boxed{\mathrm{e}(\mathrm{n}) = \mathrm{d}(\mathrm{n}) - \sum_{\mathrm{j}=0}^{\mathrm{m}} \mathrm{x}_{\mathrm{j}}(\mathrm{n})\mathrm{w}_{\mathrm{j}}(\mathrm{n})}$$

- We can find the minimum of the error function *E* by means of the Steepest descent method (Optimization Procedure)



(w1,w2)
(w1+Δw1,w2+Δw2)
E[w1,w2]
30
25
20
15
10
5
0
2
1
0
−1
−2
w2
w1

Delta Rule Example

- Refer to document attached Delta Rule.pdf

Applications of Neural Networks

1. **Facial Recognition**

Recognition Systems matches the human face and compares it with the digital images. They are used in offices for selective entries. The systems thus authenticate a human face and match it up with the list of IDs that are present in its database.

CNN are used for **facial recognition and image processing.** Large number of pictures are fed into the database for training a neural network. The collected images are further processed for training.

2. **Stock Market Prediction**

- To make a successful stock prediction in real time a **Multilayer Perceptron MLP** *(class of feedforward artificial intelligence algorithm)* is employed.
- MLP comprises multiple layers of nodes, each of these layers is fully connected to the succeeding nodes.
- Stock's past performances, annual returns, and non profit ratios are considered for building the MLP model.

3. **Social Media**

Artificial Neural Networks are used to study the behaviours of social media users.

Neural networks duplicate the behaviours of social media users. Post analysis of individuals' behaviours via social media networks the data can be linked to people's spending habits.

4. **Aerospace**

Fault diagnosis, high performance auto piloting, securing the aircraft control systems, and modeling key dynamic simulations are some of the key areas that neural networks have taken over.

Neural Networks are used for **position independent feature recognition.**

5. **<u>Defence</u>**

Neural networks are used in logistics, armed attack analysis, and for object location. They are also used in air patrols, maritime patrol, and for controlling automated drones.

6. **<u>Healthcare</u>**

Convolutional Neural Networks are actively employed in the healthcare industry for **X ray detection, CT Scan** and **ultrasound.**

7. <u>Weather Forecasting</u>

Multilayer Perceptron (MLP), Convolutional Neural Network (CNN) and Recurrent Neural Networks (RNN) are used for weather forecasting. Traditional ANN multilayer models can also be used to predict climatic conditions 15 days in advance. A combination of different types of neural network architecture can be used to predict air temperatures.

Comparison of BNN and ANN

Artificial Neural Network (ANN) is a type of neural network which is based on a Feed-Forward strategy.

- It is called this because they pass information through the nodes continuously till it reaches the output node.
- This is also known as the simplest type of neural network.

Biological Neural Network (BNN) is a structure that consists of Synapse, dendrites, cell body, and axon.

- In this neural network, the processing is carried out by neurons.
- Dendrites receive signals from other neurons, Soma sums all the incoming signals and axon transmits the signals to other cells.

Parameters	ANN	BNN
Structure	Input, weight, output hidden layer	Dendrites, synapse, axon cell body
Learning	very precise structures and formatted data	they can tolerate ambiguity
Processor	Complex high speed one or a few	Simple , low speed ,large number
Memory	separate from a processor Localized non-content addressable	integrated into processor distributed content-addressable
Computing	Centralized sequential stored programs	Distributed parallel self-learning
Reliability	very vulnerable	robust
Expertise	numerical and symbolic manipulations	perceptual problems
Operating Environment	well-defined , well-constrained	poorly defined ,un-constrained
Fault Tolerance	the potential of fault tolerance	performance degraded even on partial damage