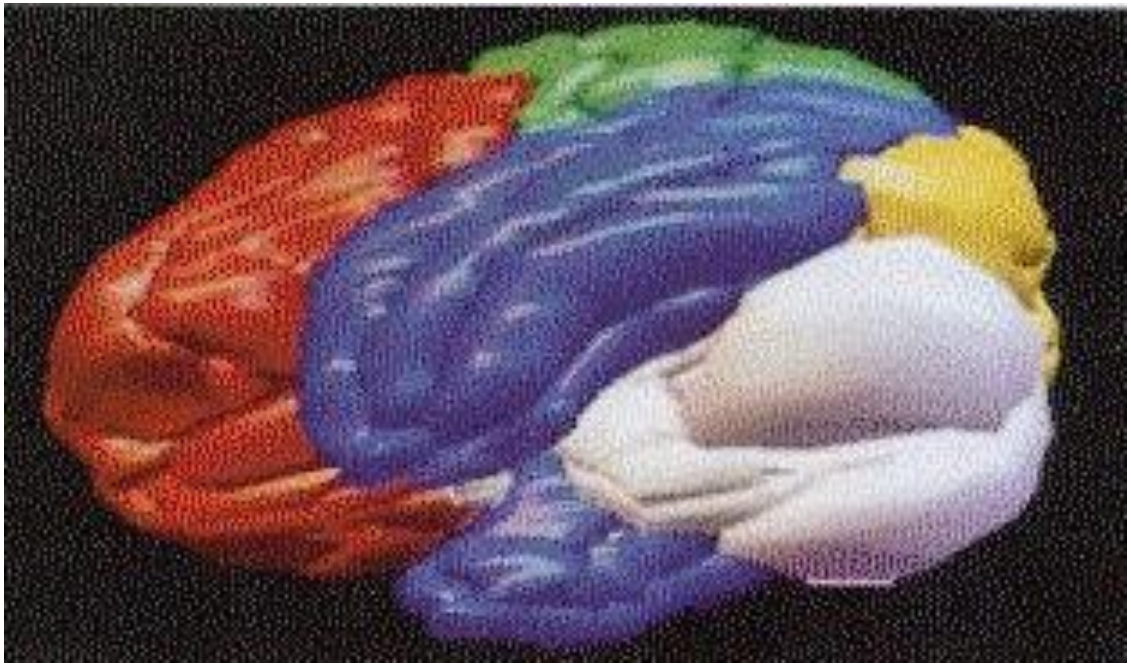
Foundations of Artificial Intelligence



Neural Networks

Building Artificial Brains

Background

Can machine ever be intelligent?
What machine CAN do?

Machine learning

Give machines the ability to learn, make decision, and behave not explicitly as what they are programmed

Improve performance of intelligent systems over time by examples and analogy

Background

Machine learning

A branch of artificial intelligence, is a scientific discipline that is concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases.

Background

Machine learning

A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data.

The difficulty lies in the fact that the set of all possible behaviors given all possible inputs is too large to be covered by the set of observed examples (training data).

Neural Networks

Objectives

Show how the human brain works.

To introduce the concept of neural networks.

Demonstrate different types of neural networks.

Introduce some neural network software.

Show some ‘real’ neural network applications.

Neural Networks

Session 1

Introduction

The Human Brain (How a neuron works)

Building Artificial Neurons

Network Architecture and Learning

Session 2

Software Demonstrations

Real World Applications

Artificial Neural Networks

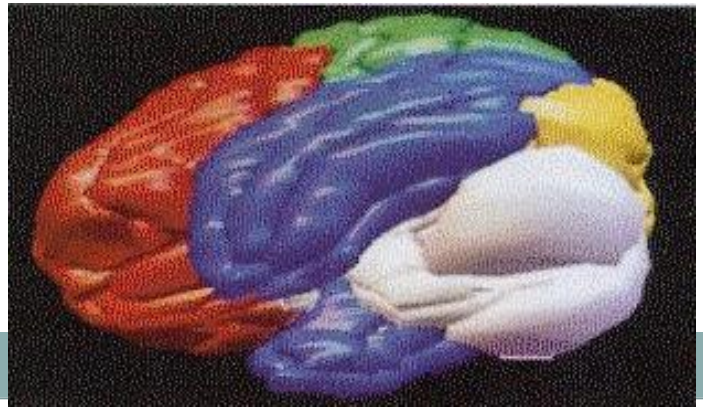
… consists of interconnected processing elements called nodes or neurons that work together to produce an output function. The output of a neural network replies on the connection of the individual neurons within the network to operate.

Wiki

Relationship between Artificial Neural Networks & the Human Brain

Neural networks are conceptually modelled on the human brain metaphor.

The general structure of a neural network tries to mimic what we know about the structure and operation of the human brain.



Relationship between Artificial Neural Networks & the Human Brain

The human brain consists of, among other things, a highly interconnected system of neurons.

The neuron is the basic building block of the brain and the nervous system.

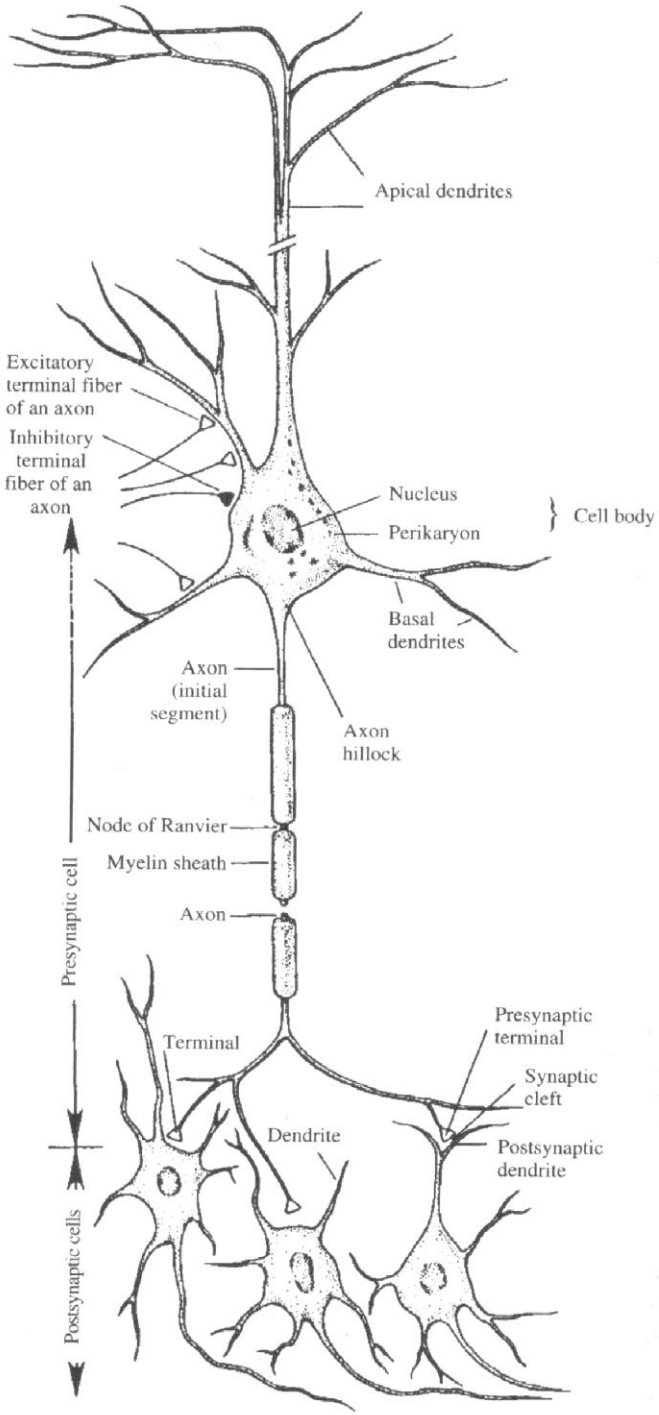
Signals are passed between neurons by means of electrochemical pulses.

The Human Brain

The neuron is the basic building block of the brain and the nervous system

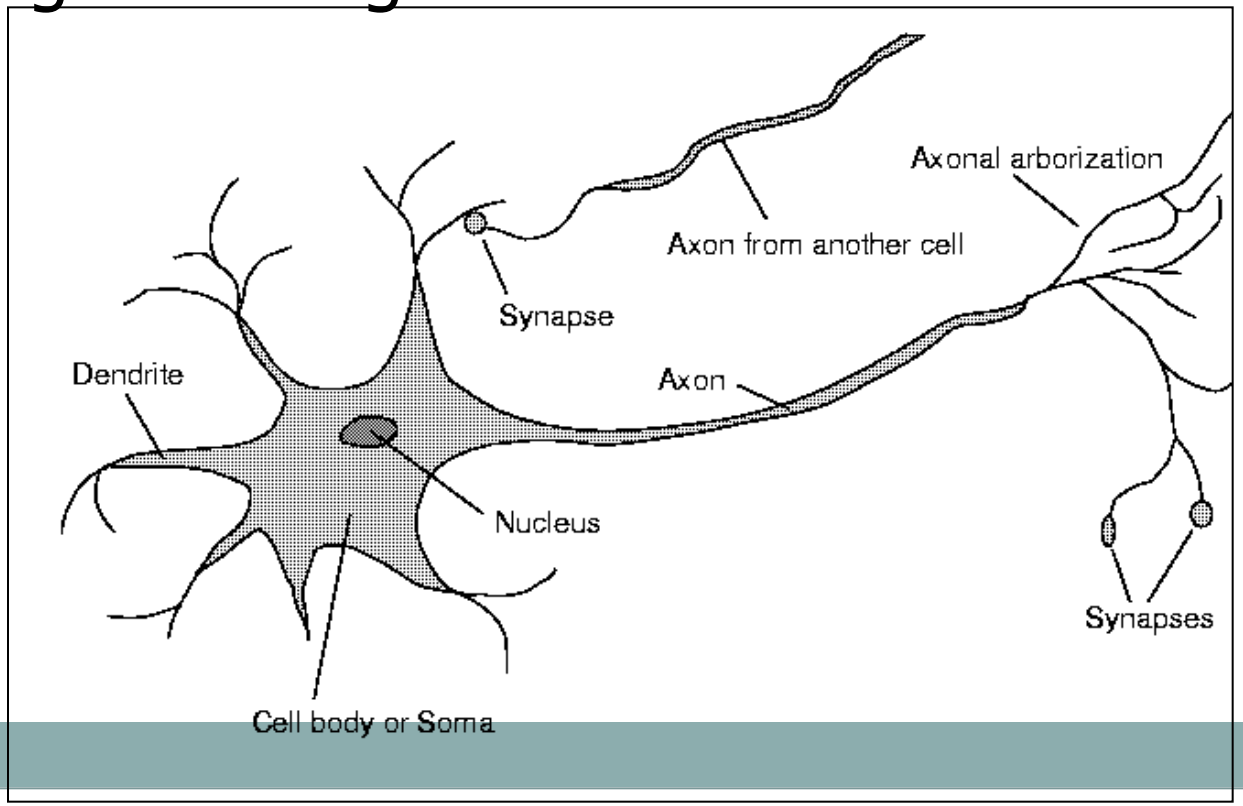
The human brain has around 10^{11} neurons *(between 10 and 500 billion)*

Each is connected to a large number of other neurons (about 1000 on average).



The Human Brain

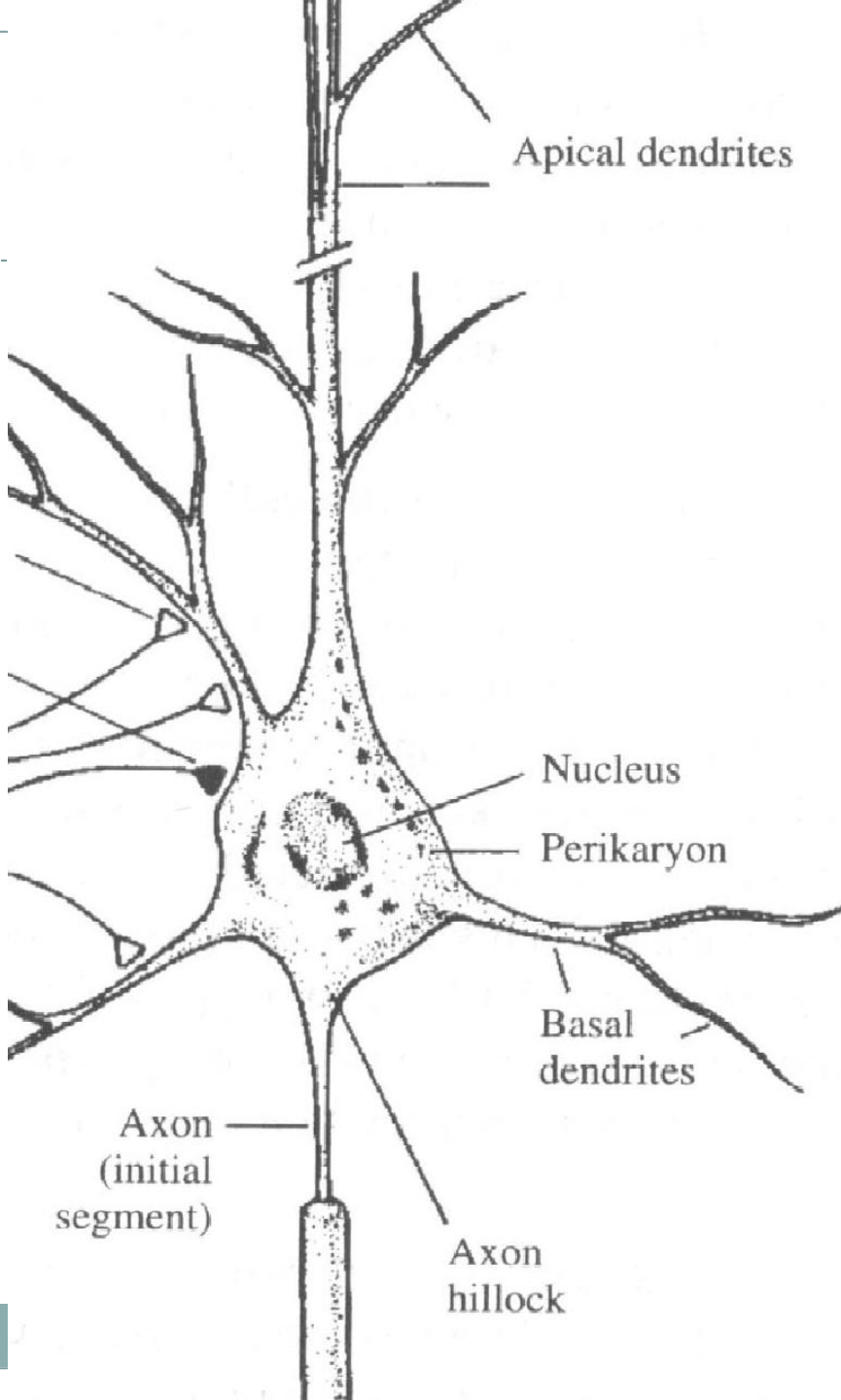
Each neuron is connected to other neurons by **axons** which end in **synapses**. These synapses send signals along **dendrites** into the neuron.



The Human Brain

The dendrites can be regarded as the *inputs* to the neuron and the axon can be regarded as the neuron's *output*.

In the brain axons (outputs) from many neurons will connect to the dendrites (inputs) of many other neurons.



The Human Brain

The strength of a signal can be modified by the strength of the *synapse* or connection between one neuron's axon (output) and another neuron's dendrite (input).

The synapse can be regarded as the chemical equivalent of an electrical variable resistor.

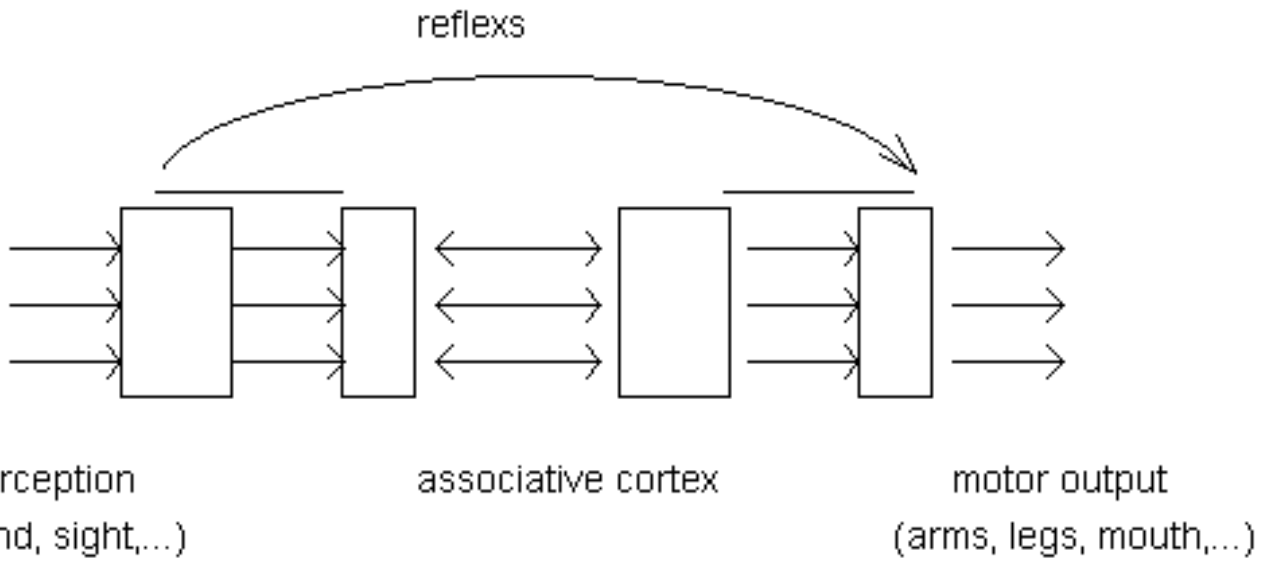
The brain learns by modifying the strength of its synapses.

The Human Brain

Neurons are arranged in a rough layer-like structure.

The early layers receive input from the sense organs.

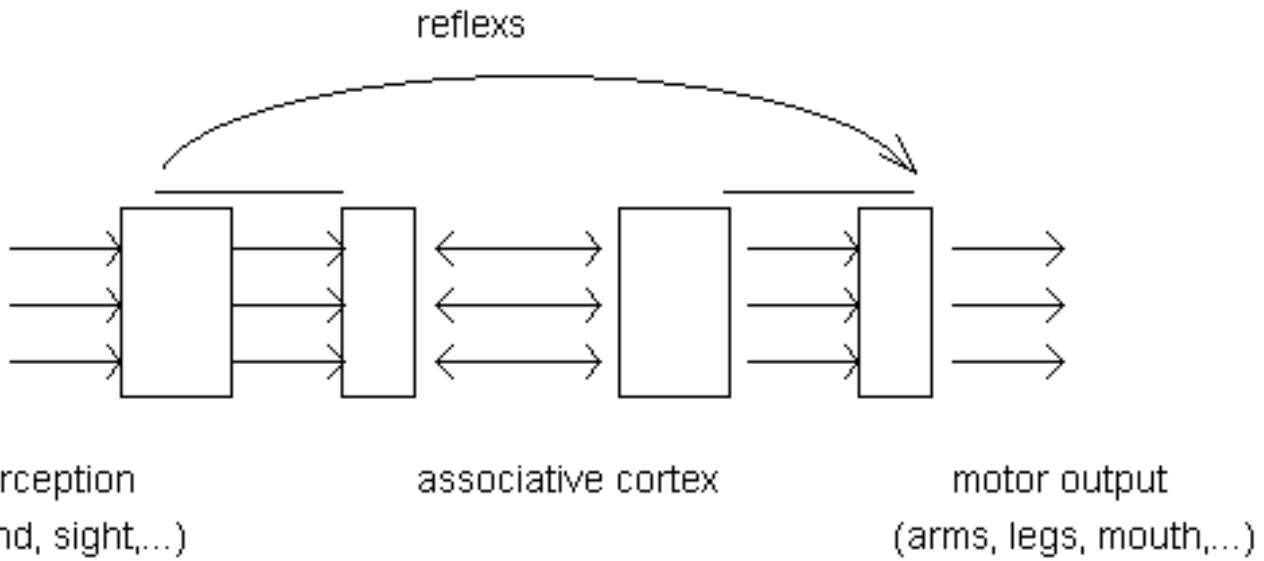
The final layers produce motor outputs.



The Human Brain

The middle layers form the **associative cortex**.

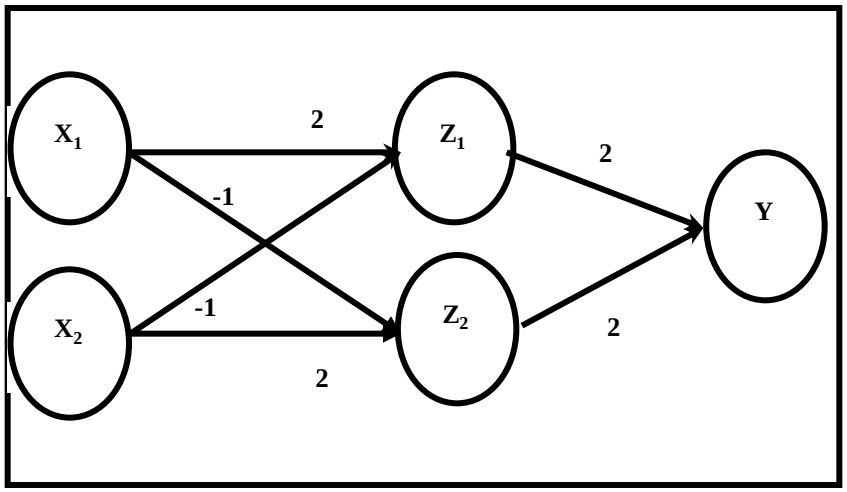
This part of the brain is little understood, but is almost certainly the most important part of the brain in humans.



An Artificial Brain

An artificial neural network consists of a number of simple and highly connected neurons.

Neurons are connected by weighted links passing signals from one neuron to another.



An Artificial Brain

In 1942 McCulloch and Pitts designed the first model of an artificial neuron.

It is still the basis for most neural networks today.

It consisted of:

- A set of inputs - (dendrites)
- A set of variable resistances - (synapses)
- A processing element - (neuron)
- A single output - (axon)

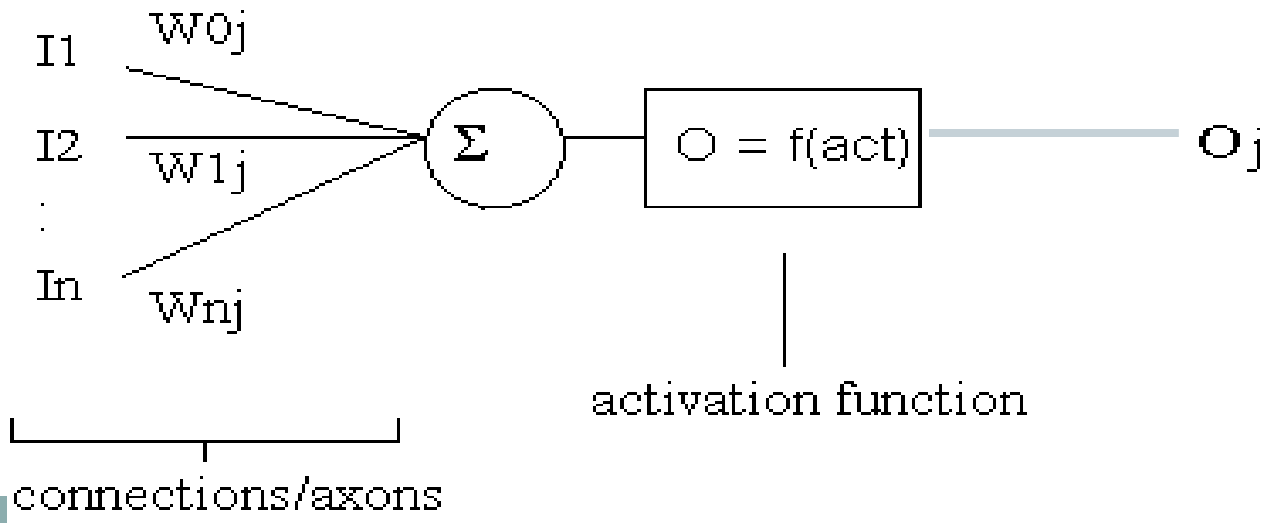
An Artificial Brain

A set of inputs - (I_1, I_2, I_n)

A set of variable resistances - (W_1, W_2, W_n)

A processing element - (summation and activation)

A single output - (O_j)

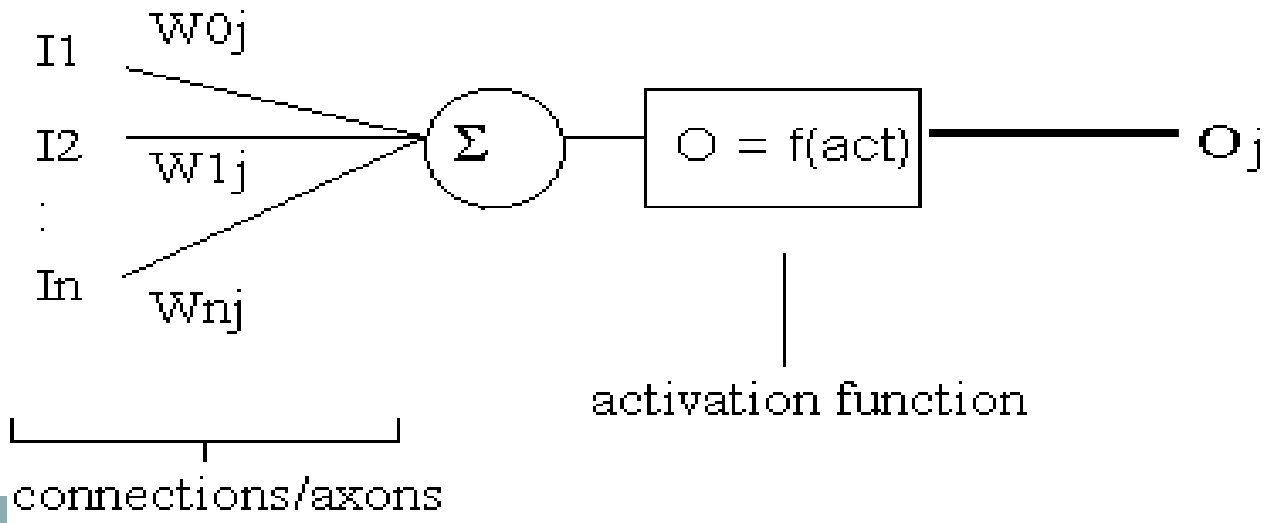


An Artificial Brain

The main cell body consistss of:

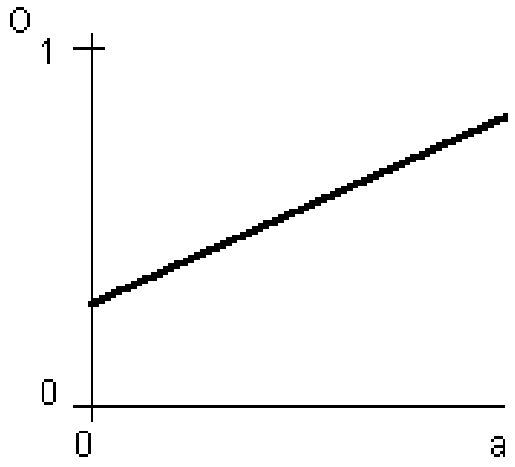
•a summation component that adds together all of the input signals to the neuron

•a transfer function (also called activation function) that modifies the results of the summation before sending it to the single output.

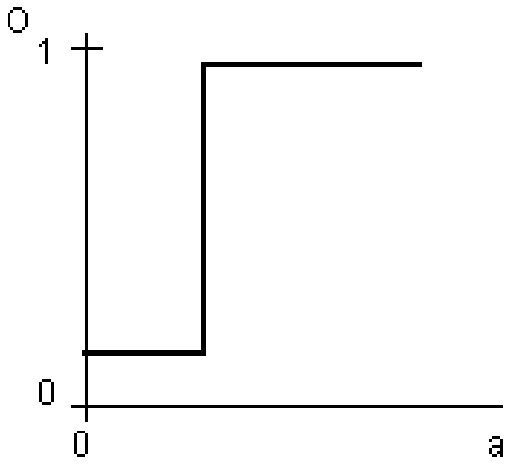


An Artificial Brain

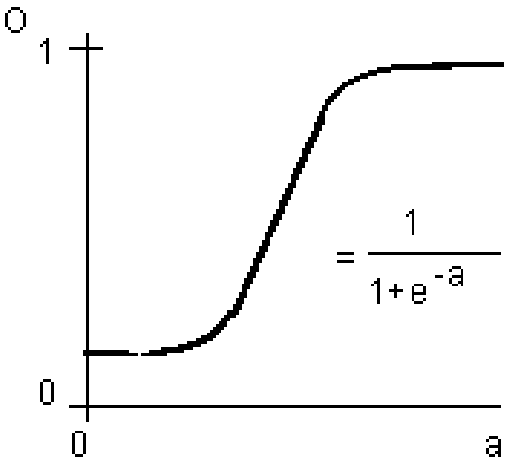
Basic transfer or activation functions used in processing elements can vary.
Only a few are found of practical use.
Think of the transfer function as an electronic gate.



4.1 Linear function



4.2 Nonlinear function



4.3 Semilinear function(sigmoid

An Artificial Brain

Characteristics that seem to be common to both biological neural networks and most of the advanced neural networks.

- Massively parallel computation.
- Adaptation to changing environment, and emergence of "intelligent" information processing functions by self-organisation, in response to data.

Background

McCulloch & Pitts (1943) are generally recognised as the designers of the first neural network

Many simple units combine to give an increased computational power

The idea of a threshold

Many of their ideas still used today

Background

Hebb (1949) developed the first learning rule

McCulloch & Pitts network has fixed weights

If two neurons were active at the same time the strength between them should be increased

Background

During the 50's and 60's

- Many researchers worked on the perceptron with great excitement
- This model can be proved to converge to the correct weights
- More powerful learning algorithm than Hebb

Background

1969 saw the death of neural network research

Minsky & Papert

Perceptron can't learn certain type of important functions

Research of ANN went to decline for about 15 years

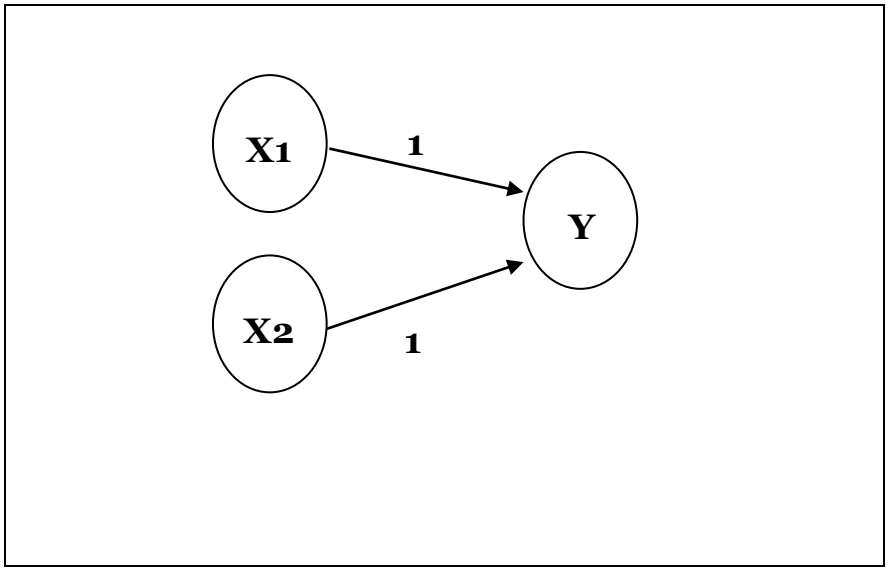
Background

Only in the mid 80's was interest revived

Parker (1985) and LeCun (1986) independently discovered multi-layer networks to solve problem of non-linear separable

In fact Werbos discovered an algorithm in 1974, Bryson & Ho developed another algorithm in 1969

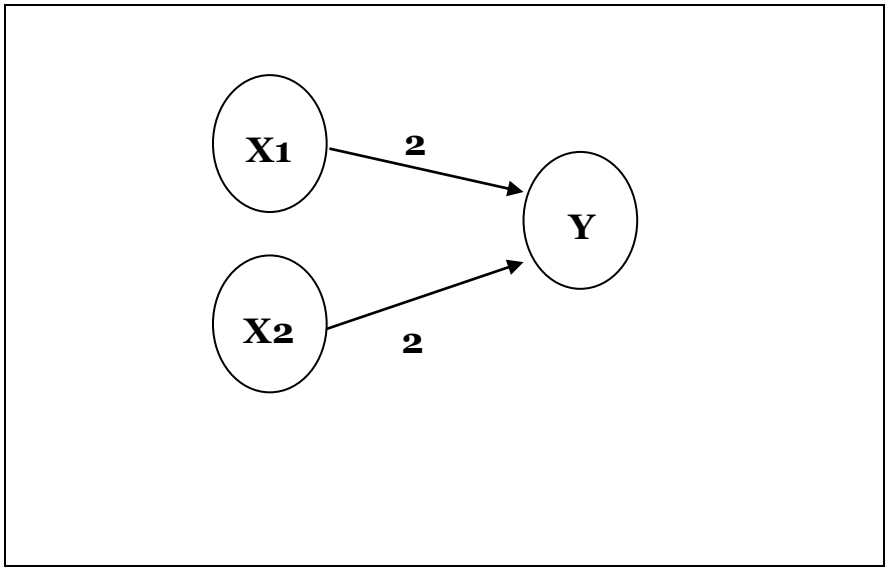
The First Neural Networks



AND		
X1	X2	Y
1	1	1
1	0	0
0	1	0
0	0	0

Threshold(*Y*) = 2

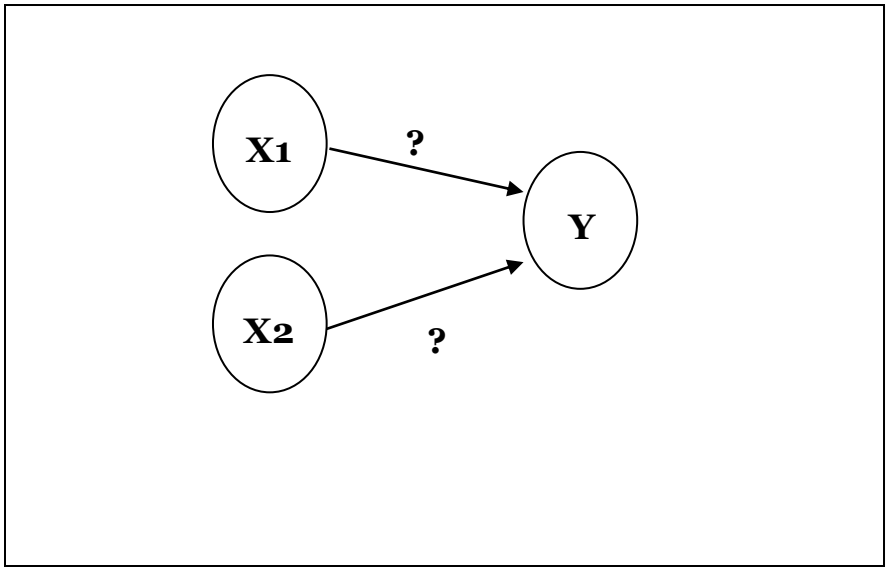
The First Neural Networks



OR		
X1	X2	Y
1	1	1
1	0	1
0	1	1
0	0	0

Threshold(*Y*) = 2

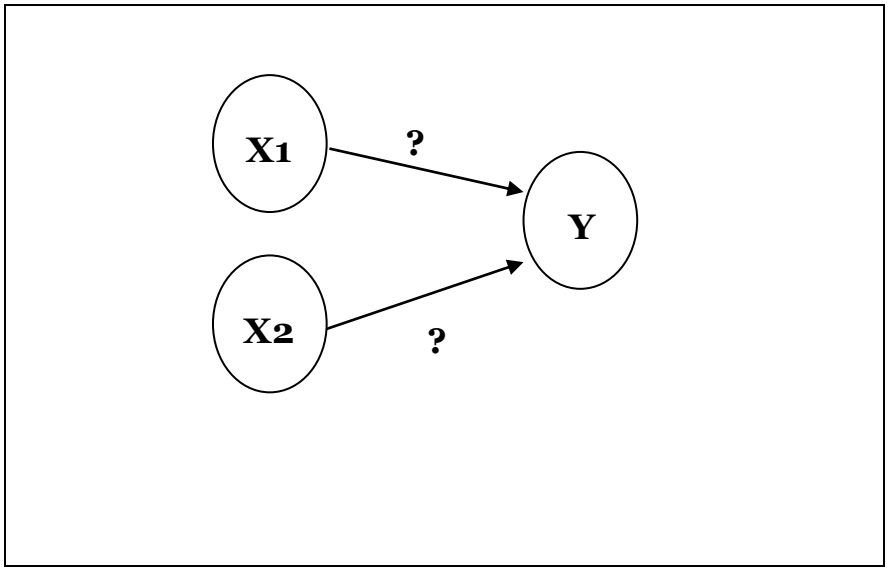
The First Neural Networks



AND NOT		
X1	X2	Y
1	1	0
1	0	1
0	1	0
0	0	0

Threshold(*Y*) = ?

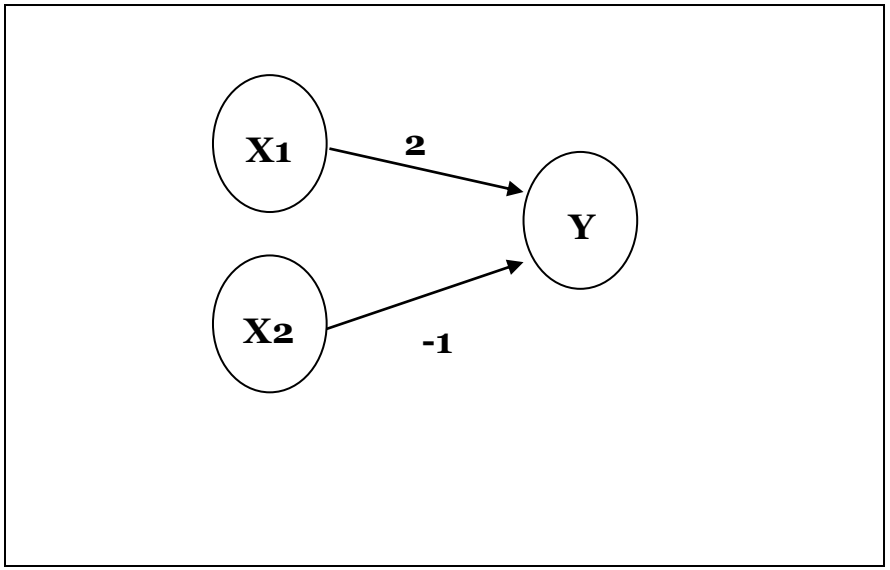
The First Neural Networks



XOR		
X1	X2	Y
1	1	0
1	0	1
0	1	1
0	0	0

Threshold(*Y*) = ?

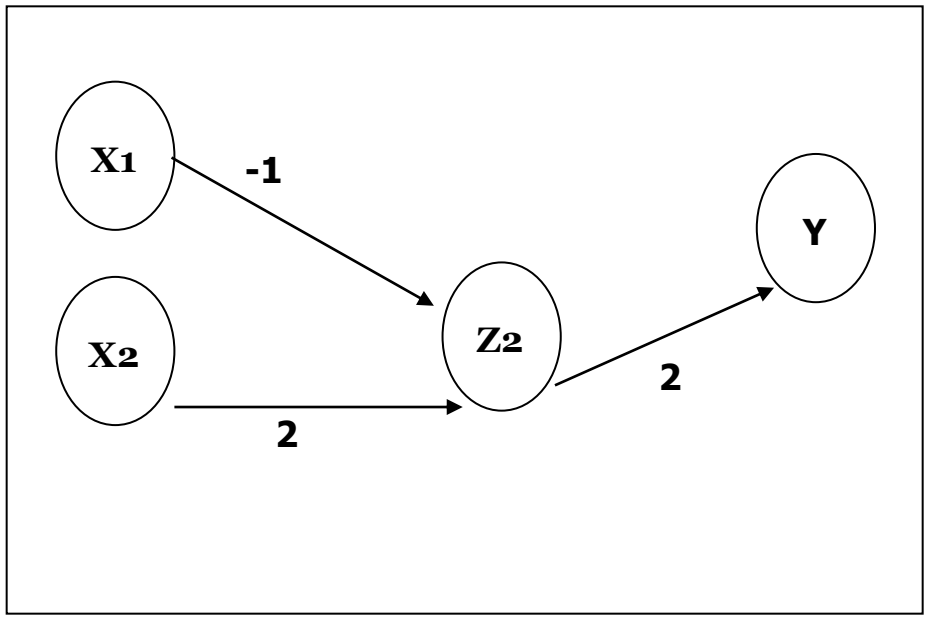
The First Neural Networks



AND NOT		
X1	X2	Y
1	1	0
1	0	1
0	1	0
0	0	0

Threshold(*Y*) = 2

The First Neural Networks

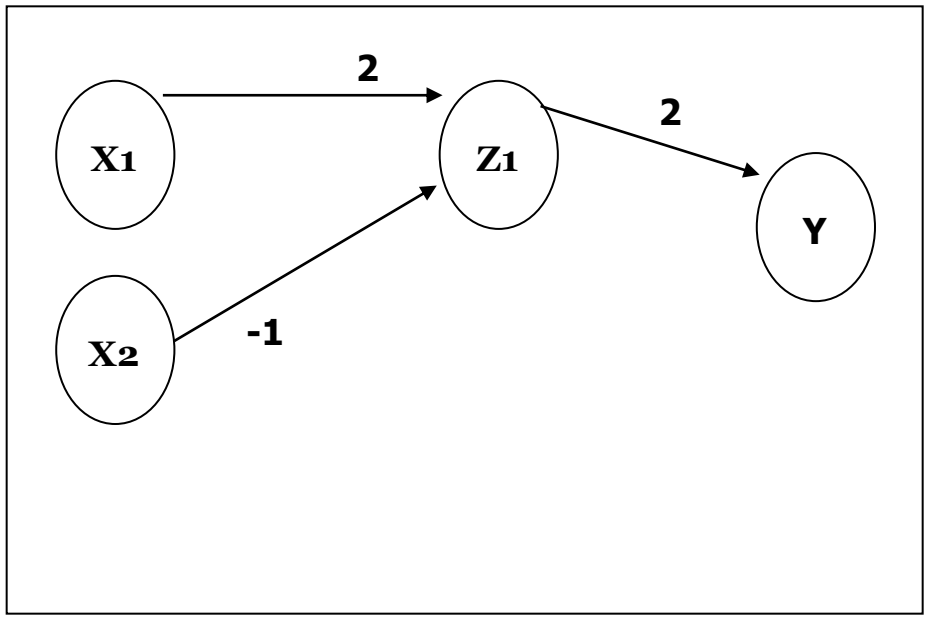


XOR		
X1	X2	Y
1	1	0
1	0	1
0	1	1
0	0	0

AND NOT		
X1	X2	Y
1	1	0
1	0	1
0	1	0
0	0	0

X1 XOR X2 = (X1 AND NOT X2) OR (**X2 AND NOT X1**)

The First Neural Networks

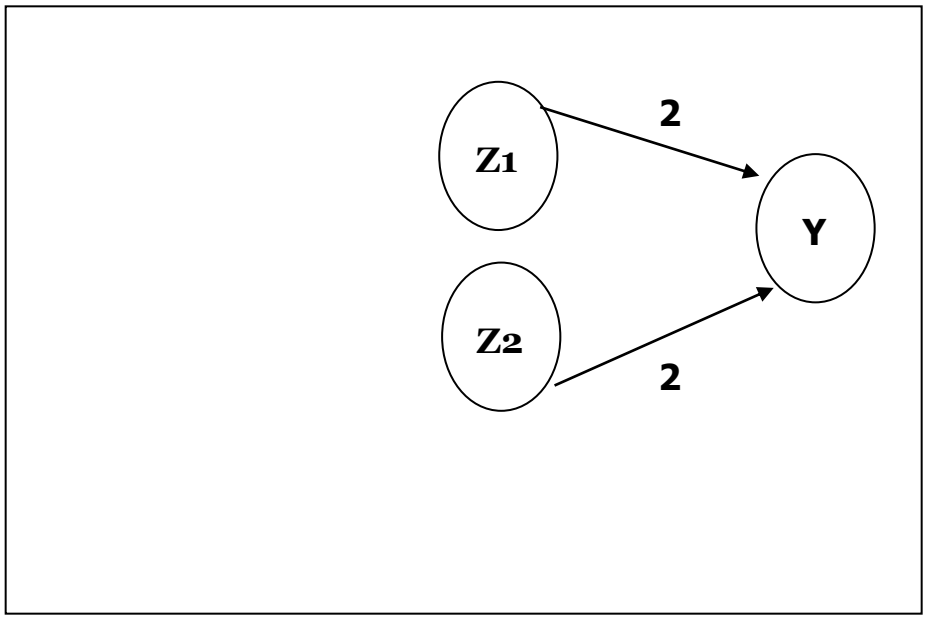


XOR		
X1	X2	Y
1	1	0
1	0	1
0	1	1
0	0	0

AND NOT		
X1	X2	Y
1	1	0
1	0	1
0	1	0
0	0	0

X1 XOR X2 = (**X1 AND NOT X2**) OR (X2 AND NOT X1)

The First Neural Networks



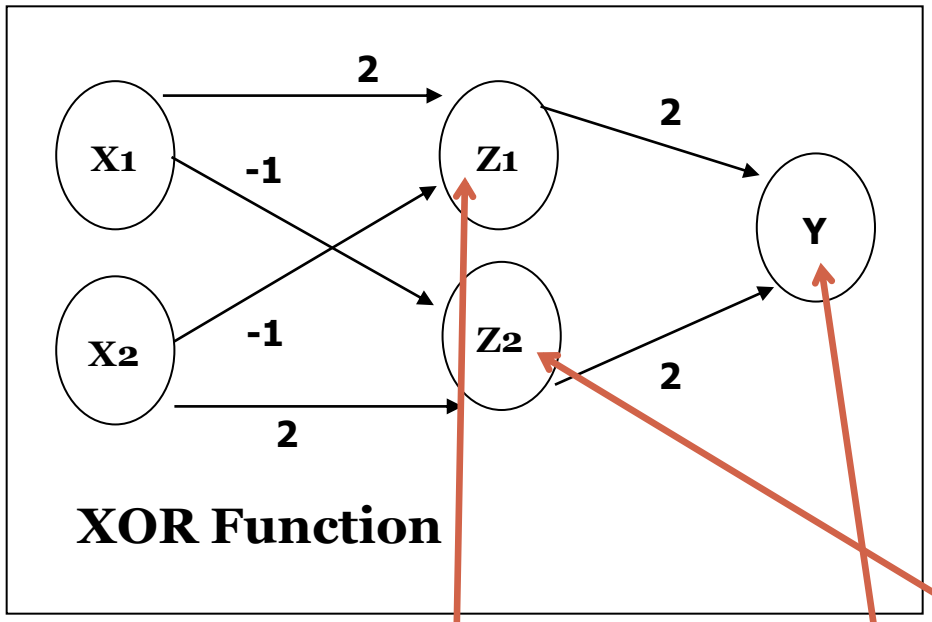
XOR		
X1	X2	Y
1	1	0
1	0	1
0	1	1
0	0	0

AND NOT		
X1	X2	Y
1	1	0
1	0	1
0	1	0
0	0	0

X1 XOR X2 = (X1 AND NOT X2) **OR** (X2 AND NOT X1)

The First Neural Networks

XOR		
X1	X2	Y
1	1	0
1	0	1
0	1	1
0	0	0

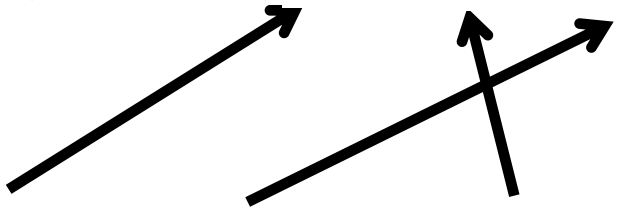


X1 XOR X2 = (X1 AND NOT X2) OR (X2 AND NOT X1)

The First Neural Networks

XOR	
X1	X2
1	1
1	0
0	1
0	0

Y
0
1
1
0



X1 XOR X2 = (X1 AND NOT X2) OR (X2 AND NOT X1)

Network Architecture

Multi-layer perceptrons can be trained to learn non-linear separable functions (1980s).

A typical neural network will have several layers

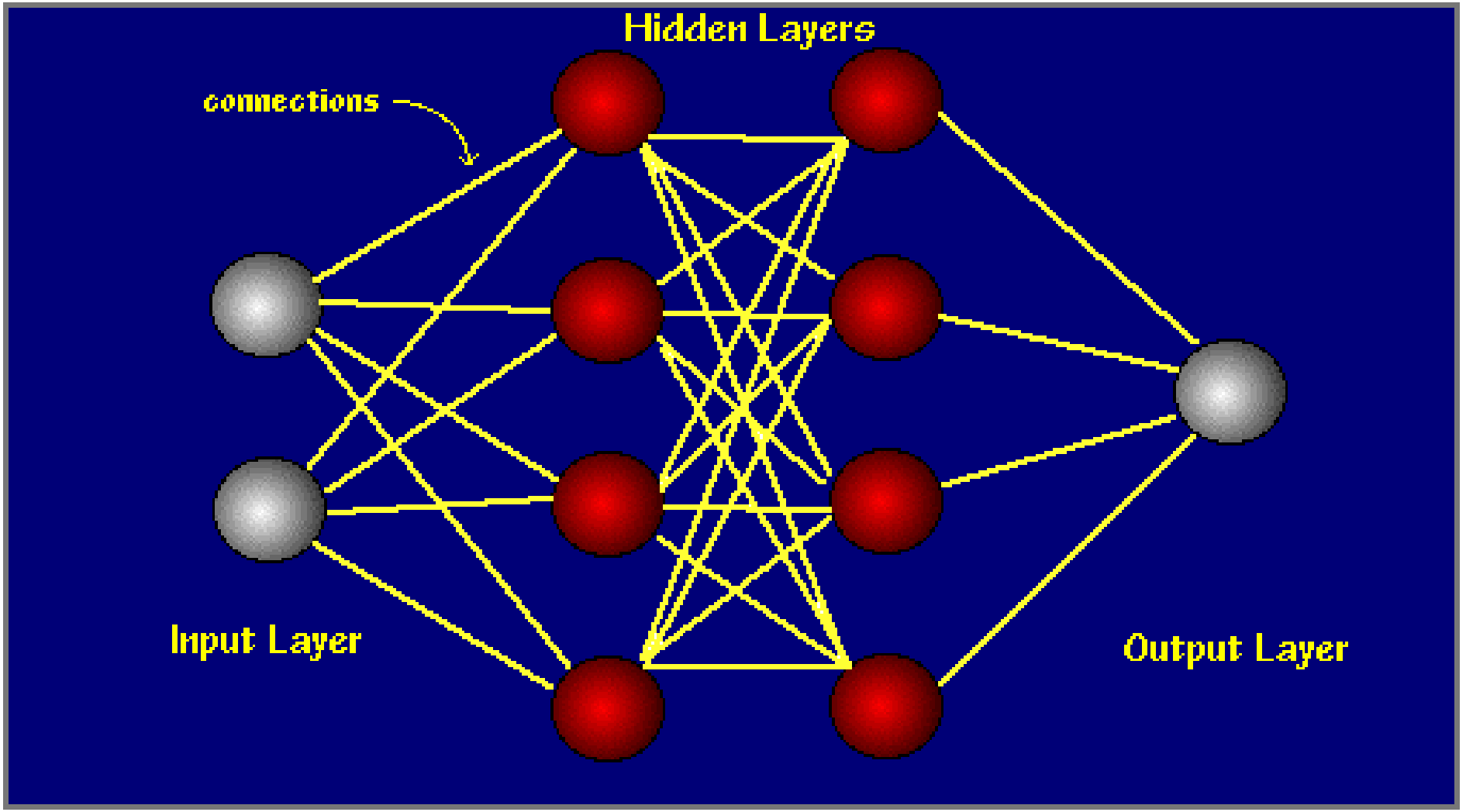
- an input layer,
- one or more hidden layers,
- and a single output layer.

In practice

- no hidden layer: cannot learn non-linear separable
- one-three layers: more practical use
- more than five layers: computational expensive

Network Architecture

Layers

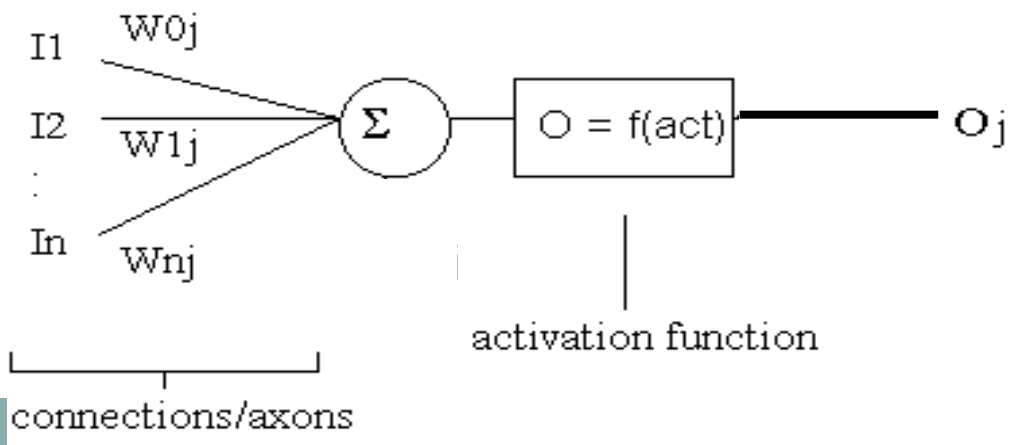


Network Architecture

The network learns by adapting its weights in accordance with a learning rule when presented with values at its inputs.

When a network has finished learning, the knowledge it has learnt is represented by the values of its connection weights.

There are many different types of network architecture.



Network Architecture

A perceptron can be trained to perform basic logic operations such as AND, OR.

- These functions are linearly separable.

Minsky and Papert (1969) reported the limitations of perceptrons.

- Perceptrons cannot learn the operations such as Exclusive-OR, which is non-linear separable.

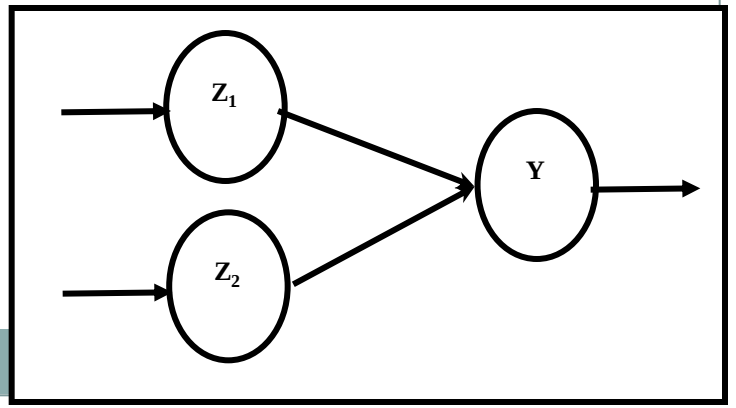
Network Architecture

- Multi-Layer Perceptron (MLP) Networks
- Radial Basis Function (RBF) Networks
- Learning Vector Quantisation (LVQ) Networks
- Recurrent Neural Networks
- Auto-Associative Neural networks
- Hopfield Networks
- Self Organising Maps (SOM)

Network Training - Perceptron

Rosenblatt (1958) introduced the first NN training procedure: a perceptron.

- Based on McCulloch and Pitts neuron model.
- Simplest NN: single neuron
- Training/learning: making small adjustments in the weights repeatedly to reduce the difference between the expected and actual output of the perceptron.



Network Training

Unlike conventional processing techniques which require complex programming, neural networks develop their own solutions to problems.

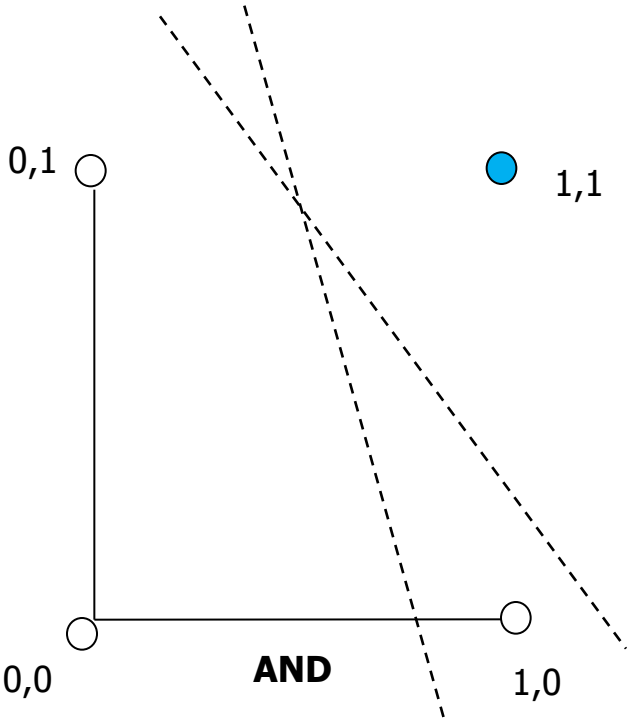
In effect, neural networks are **trained** rather than programmed.

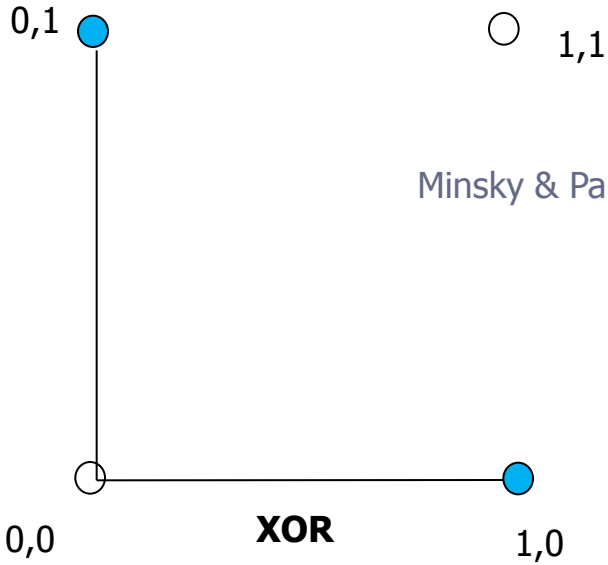
Weights are the basic means of memory, represents the importance of each neuron input.

Network Training

	AND			
Input 1	0	0	1	1
Input 2	0	1	0	1
Output	0	0	0	1

XOR			
0	0	1	1
0	1	0	1
0	1	1	0

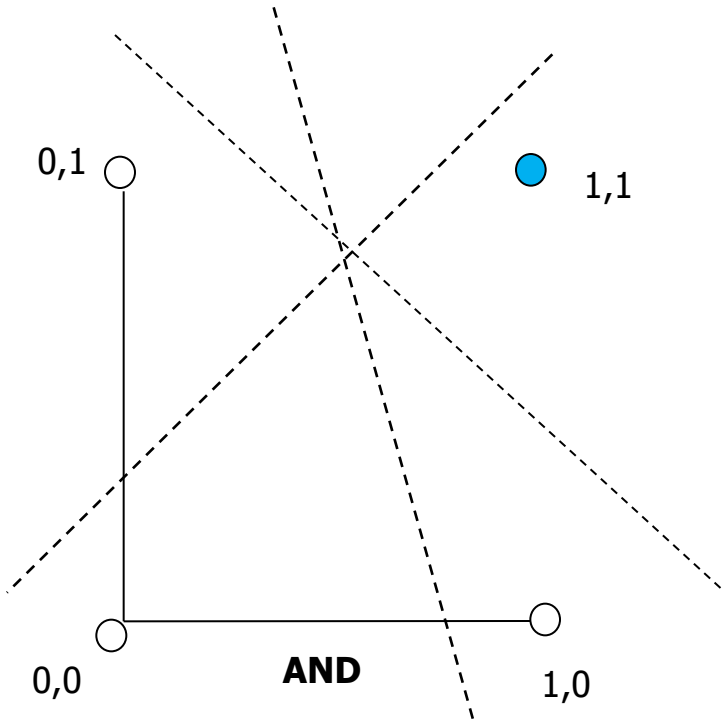




Minsky & Papert

Functions which can be separated in this way are called Linearly Separable
Only linearly Separable functions can be represented by a single layer NN

Network Training



	AND			
Input 1	0	0	1	1
Input 2	0	1	0	1
Output	0	0	0	1

Network Training

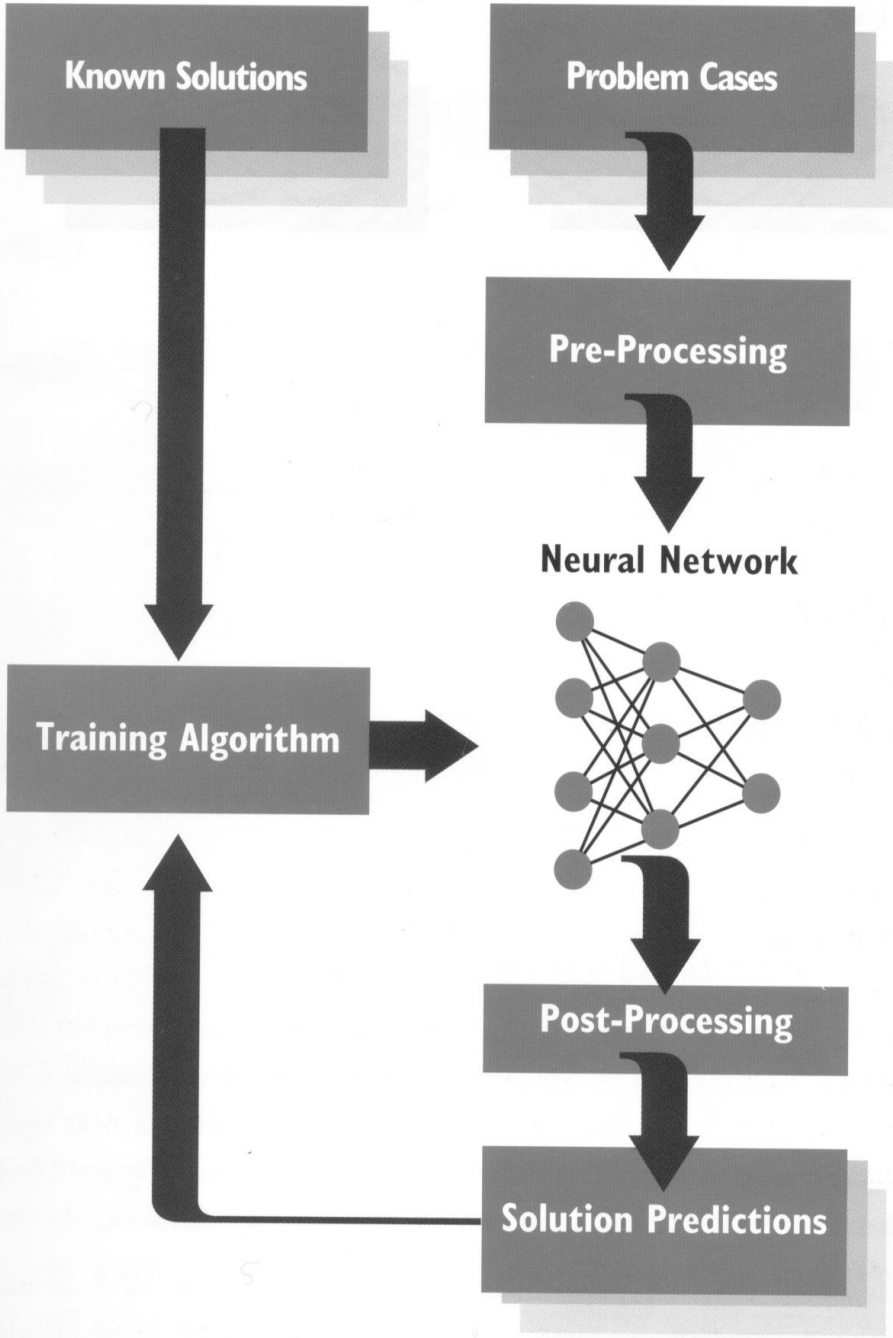
The training or learning process consists of presenting the neural network with example data and then adjusting the network's internal weights until the desired neural network response is obtained.

The method used to adjust the weights is known as the "training algorithm".

There are two basic classes of training algorithms: *supervised* and *unsupervised* training.

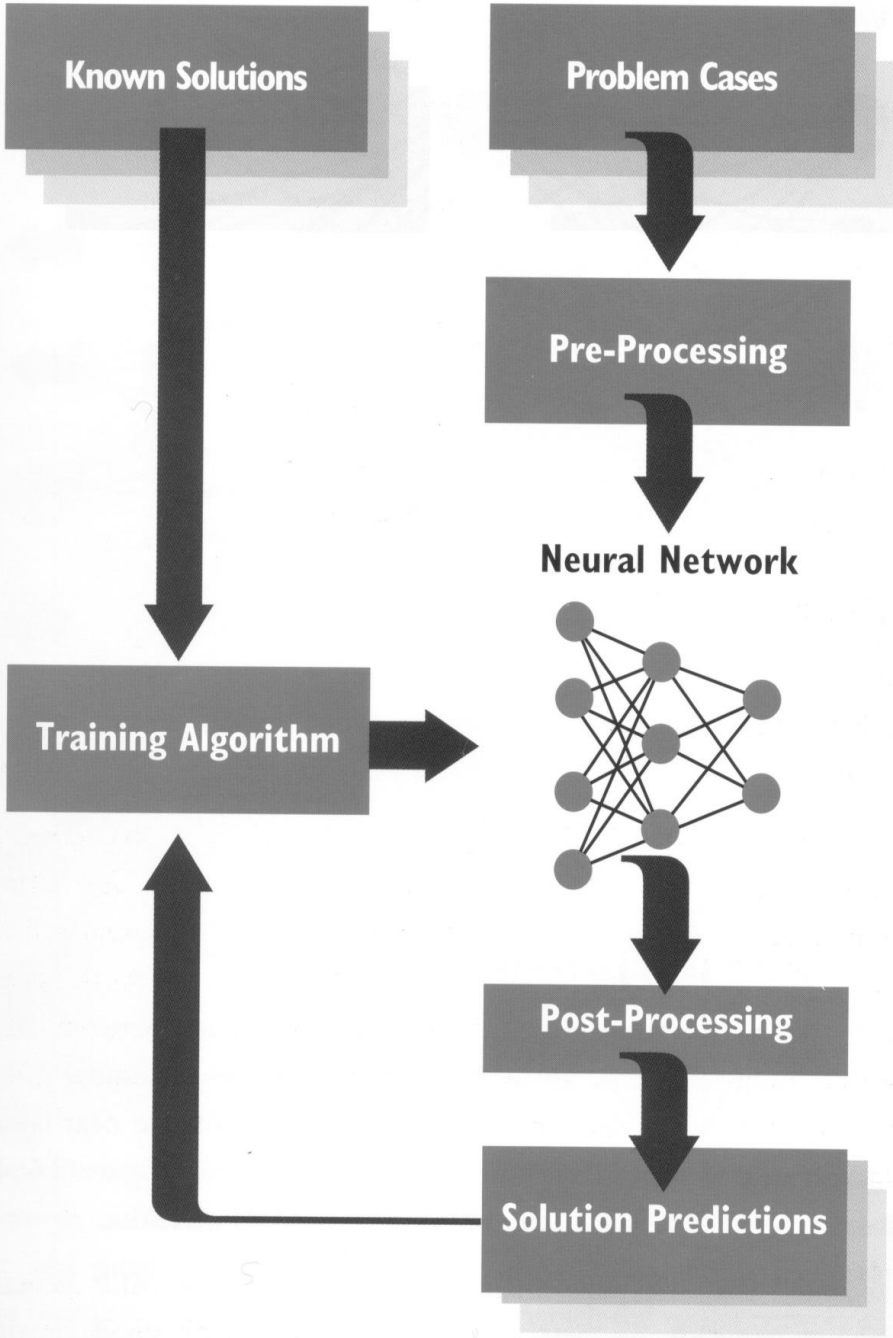
Supervised Learning

Supervised training or learning requires training data that contains both the network inputs and the associated outputs.



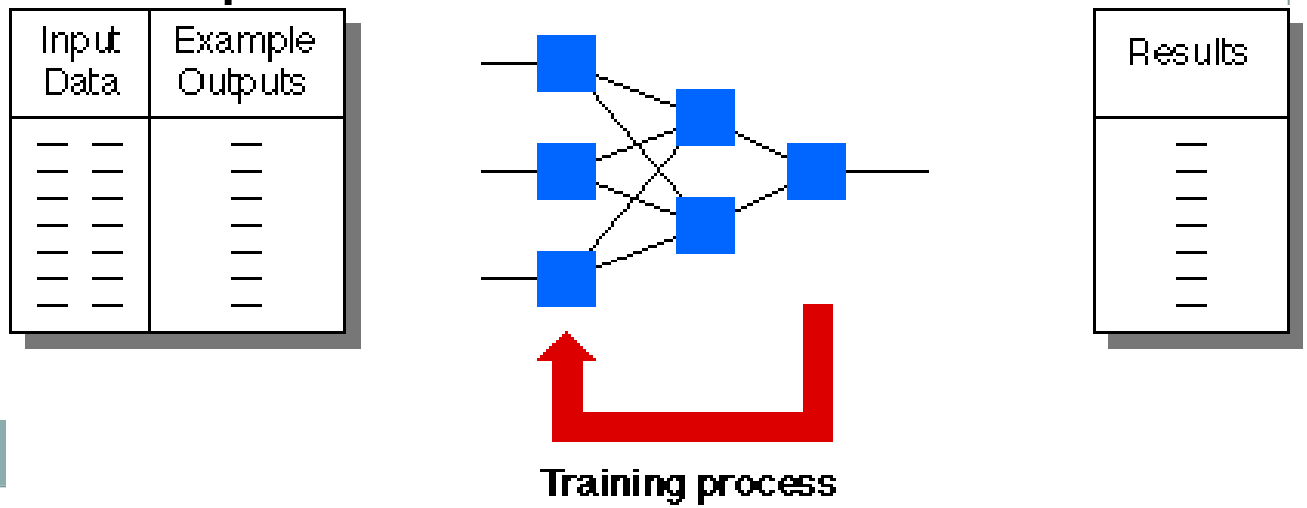
Supervised Learning

With supervised learning, for a given input pattern the network uses the associated output pattern to modify its weights to bring the network output closer to the target.



Supervised Learning

This means that to use this type of training one must have a set of training inputs for which the outputs are known.
Once this type of network performs satisfactorily on the training examples, it can be used with inputs for which the correct outputs are not known.



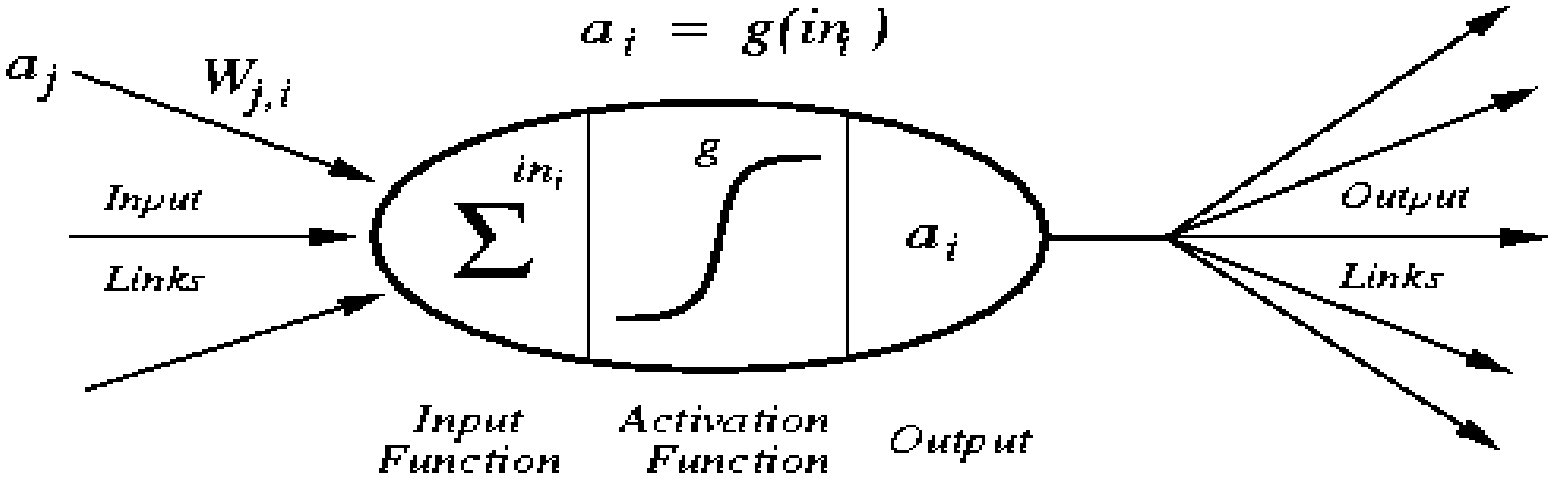
Unsupervised Learning

Unsupervised learning techniques use only input data and attempt through self organisation to divide the examples presented to the network inputs up into categories or groups with similar characteristics.

Unsupervised learning can act as a type of *discovery* process identifying significant features in the input patterns presented to it.

Do not require a teacher. It receives a number of different input patterns, and discover significant features in inputs and classify them.

Summary



$$in_i = \sum_j W_{j,i} a_j$$

aj : Input value (output from unit j)
wj,i : Weight on the link from unit j to unit i
ini : Weighted sum of inputs to unit i
ai : Activation value of unit i
g : Activation function

Neural Networks

Session 1

Introduction

The Human Brain (How a neuron works)

Building Artificial Neurons

Network Architecture and Learning

Session 2

Software Demonstrations

Real World Applications

Why Use a Neural Network ?

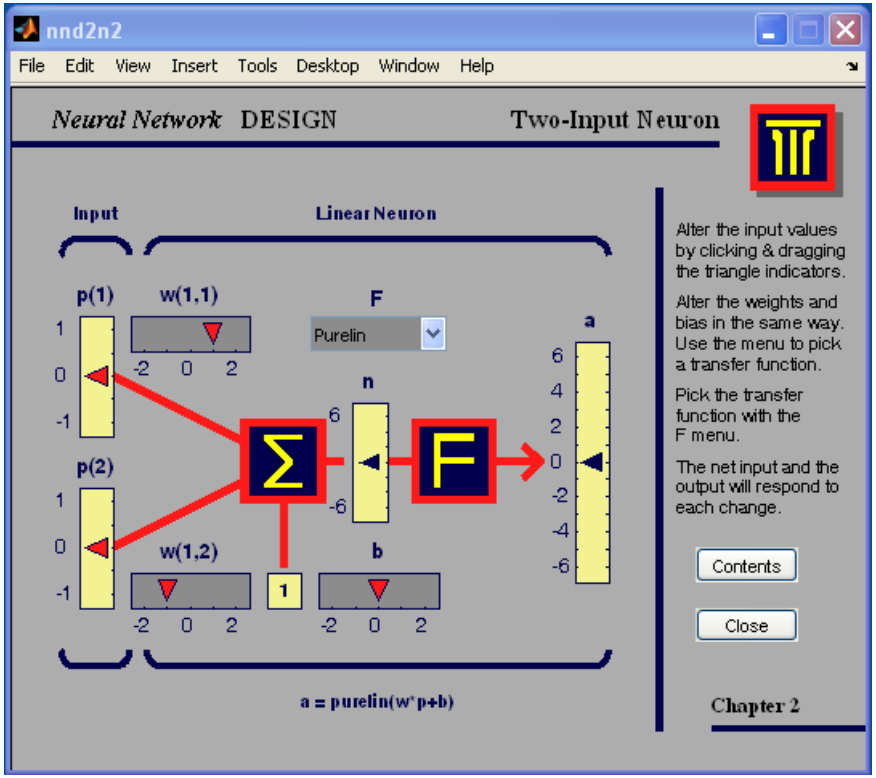
Reports of success on neural nets, particularly in financial markets, have begun leaking out to the world at large.

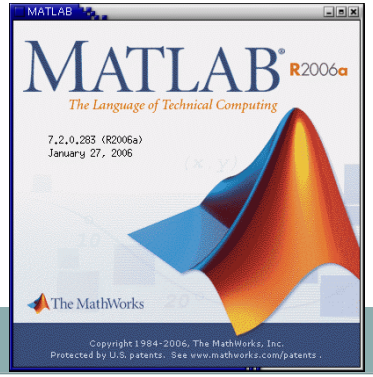
Why Use a Neural Network ?

- Learning from experience.
- Generalising from examples.
- Extracting essential information from noisy data.
- Developing solutions faster, and with less reliance on domain expertise.
- Computational efficiency.
- Adaptability.
- Non-linearity.

Neural Network Toolkits

MatLab Neural Network Toolbox

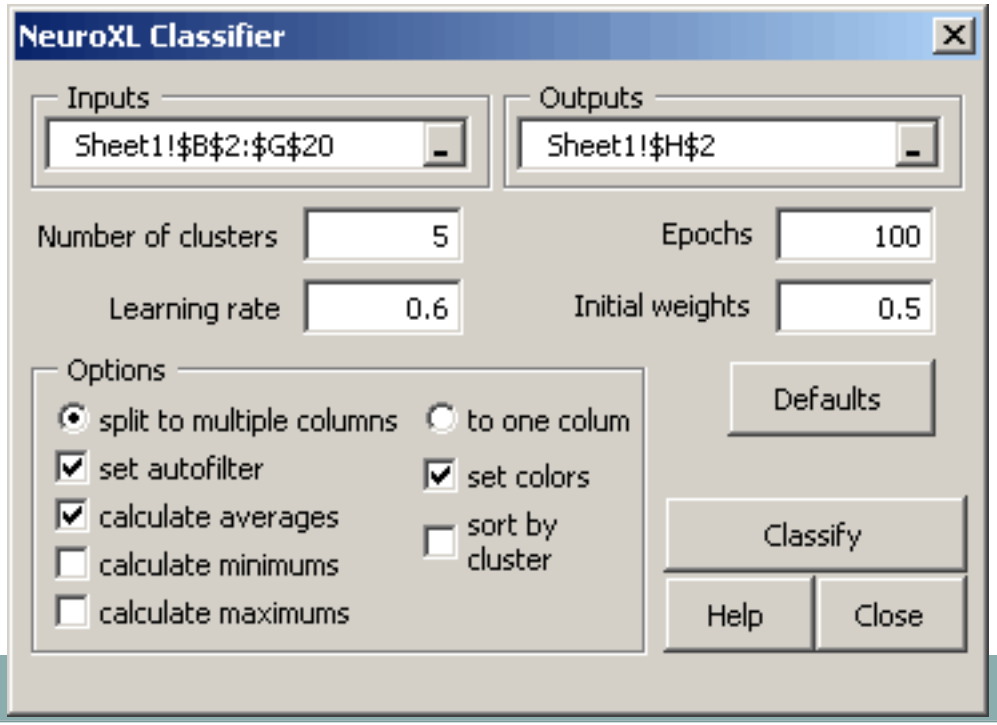




Neural Network Toolkits

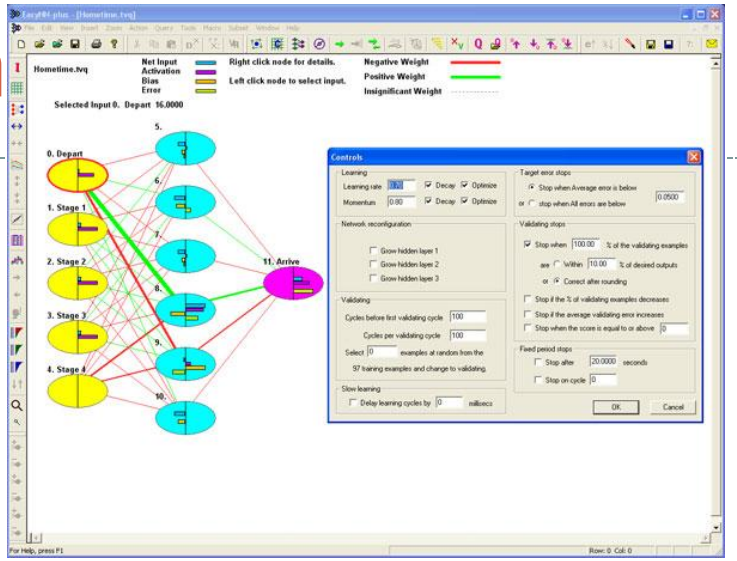
Neural Connection

NeuroXL Classifier

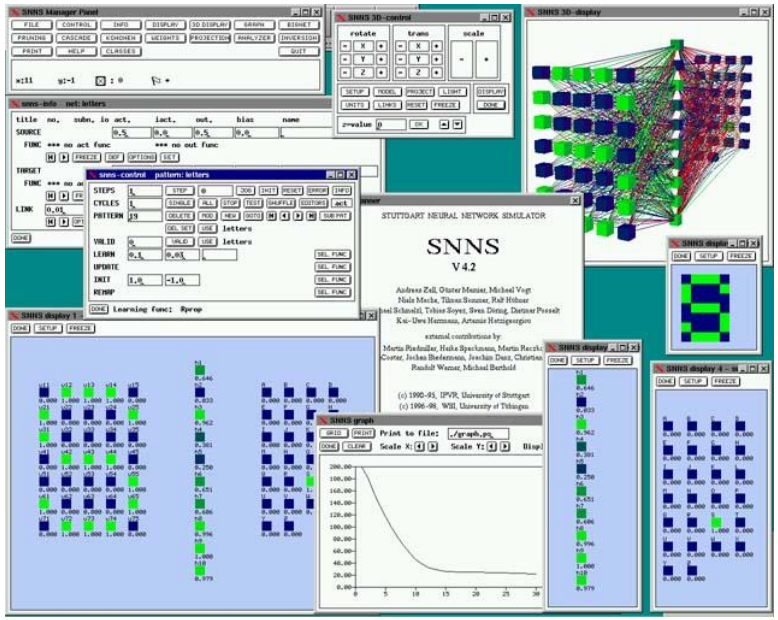


Neural Network Simula Packages

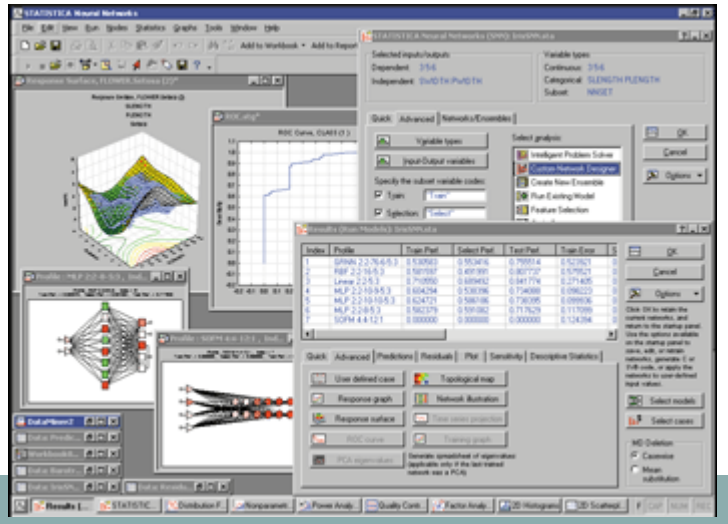
EasyNN-Plus



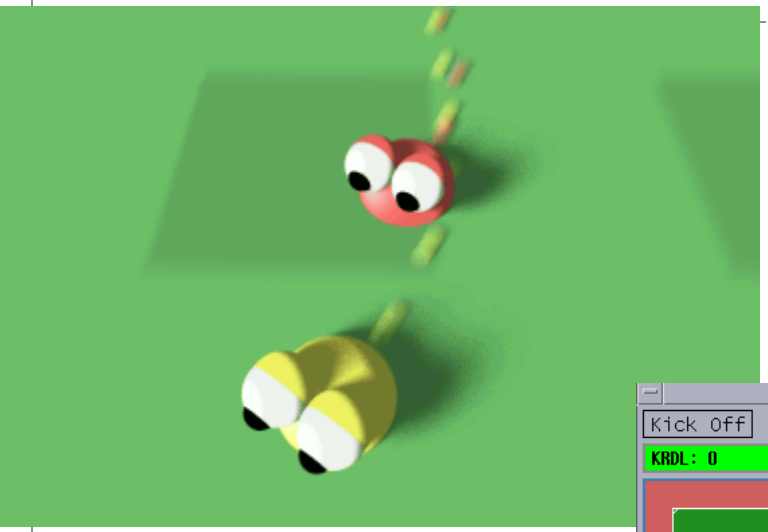
Stuttgart Neural Network Simulator

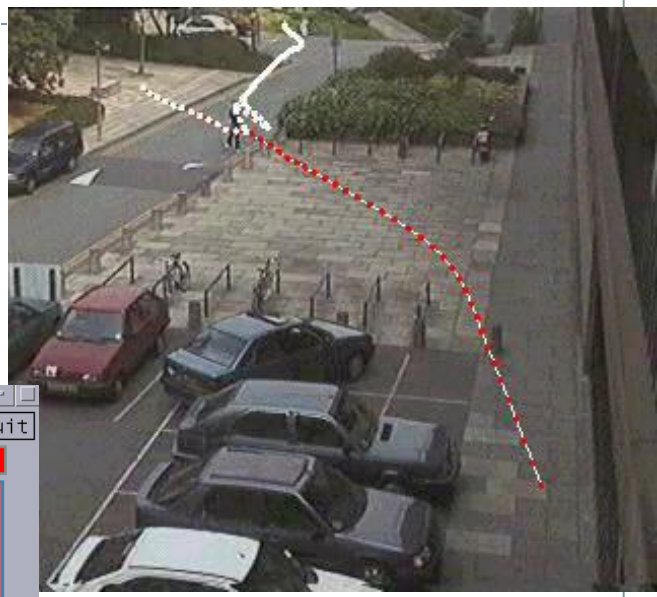


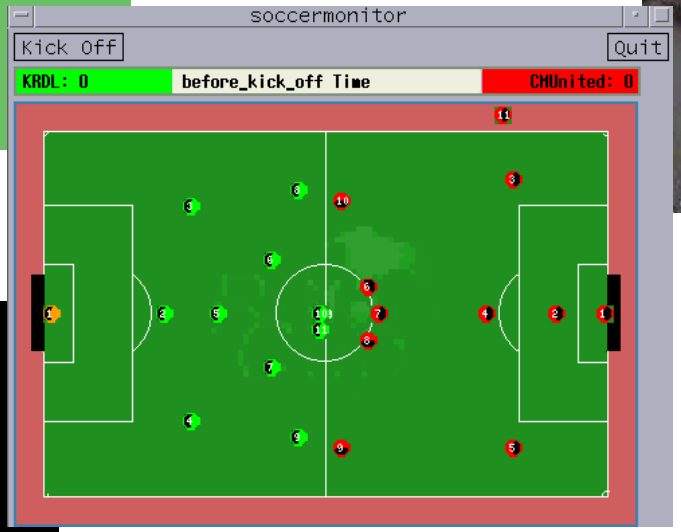
STATISTICA Neural Network

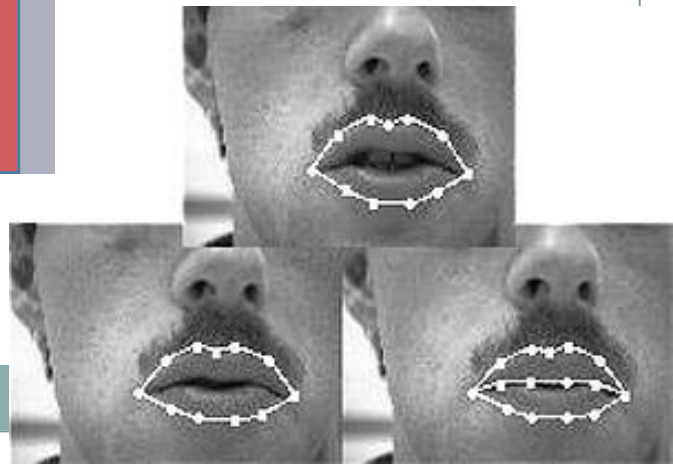


Neural Network Applications









Neural Network Applications

Time Series Prediction

- Currency price predictions
- Forecasting bond prices
- Cost prediction

Classification

- Credit scoring
- Breast cancer cell analysis
- Diagnosing heart attacks

Neural Network Applications

Science

- Protein sequence pattern recognition
- Weather forecasting
- Air quality testing

Pattern recognition

- Speech recognition
- Classification of text
- Numerical sequence prediction