INTRODUCTION:

What is intelligence?

Real intelligence is what determines the normal thought process of a human.

Artificial intelligence is a property of machines which gives it ability to mimic the human thought process. The intelligent machines are developed based on the intelligence of a subject, of a designer, of a person, of a human being.

Now two questions: can we construct a control system that hypothesizes its own control law? We encounter a plant and looking at the plant behavior, sometimes, we have to switch from one control system to another control system where the plant is operating. The plant is may be operating in a linear zone or non- linear zone; probably an operator can take a very nice intelligent decision about it, but can a machine do it? Can a machine actually hypothesize a control law, looking at the model? Can we design a method that can estimate any signal embedded in a noise without assuming any signal or noise behavior?

That is the first part; before we model a system, we need to observe. That is we collect certain data from the system and How do we actually do this? At the lowest level, we have to sense the environment, like if I want to do temperature control I must have temperature sensor. This data is polluted or corrupted by noise. How do we separate the actual data from the corrupted data? This is the second question. The first question is that can a control system be able to hypothesize its own control law? These are very important questions that we should think of actually. Similarly, also to represent knowledge in a world model, the way we manipulate the objects in this world and the advanced is a very high level of intelligence that we still do not understand; the capacity to perceive and understand.

What is AI ?

Artificial Intelligence is concerned with the design of intelligence in an artificial device.

The term was coined by McCarthy in 1956.

There are two ideas in the definition.

1. Intelligence
2. artificial device

What is intelligence?

- Is it that which characterize humans? Or is there an absolute standard of judgment?
- Accordingly there are two possibilities:
  - A system with intelligence is expected to behave as intelligently as a human
  - A system with intelligence is expected to behave in the best possible manner
- Secondly what type of behavior are we talking about?
  - Are we looking at the thought process or reasoning ability of the system?
  - Or are we only interested in the final manifestations of the system in terms of its actions?

Given this scenario different interpretations have been used by different researchers as defining the scope and view of Artificial Intelligence.

1. One view is that artificial intelligence is about designing systems that are as intelligent as humans. This view involves trying to understand human thought and an effort to build machines that emulate the human thought process. This view is the cognitive science approach to AI.

2. The second approach is best embodied by the concept of the Turing Test. Turing held that in future computers can be programmed to acquire abilities rivaling human intelligence. As part of his argument Turing put forward the idea of an 'imitation game', in which a human being and a computer would be interrogated under conditions where the interrogator would not know which was which, the communication being entirely by textual messages. Turing argued that if the interrogator could not distinguish them by questioning, then it would be unreasonable not to call the computer intelligent. Turing's 'imitation game' is now usually called 'the Turing test' for intelligence.

3. Logic and laws of thought deals with studies of ideal or rational thought process and inference. The emphasis in this case is on the inference mechanism, and its properties. That is how the system arrives at a conclusion, or the reasoning behind its selection of actions is very important in this point of view. The soundness and completeness of the inference mechanisms are important here.

4. The fourth view of AI is that it is the study of rational agents. This view deals with building machines that act rationally. The focus is on how the system acts and performs, and not so much on the reasoning process. A rational agent is one that acts rationally, that is, is in the best possible manner.

Typical AI problems

While studying the typical range of tasks that we might expect an “intelligent entity” to perform, we need to consider both “common-place” tasks as well as expert tasks.

Examples of common-place tasks include

- *Recognizing* people, objects.
- Communicating (through *natural language*).
- *Navigating* around obstacles on the streets

These tasks are done matter of factly and routinely by people and some other animals.

Expert tasks include:

- Medical diagnosis.
- Mathematical problem solving
- Playing games like chess

These tasks cannot be done by all people, and can only be performed by skilled specialists.

Now, which of these tasks are easy and which ones are hard? Clearly tasks of the first type are easy for humans to perform, and almost all are able to master them. However, when we look at what computer systems have been able to achieve to date, we see that their achievements include performing sophisticated tasks like medical diagnosis, performing symbolic integration, proving theorems and playing chess.

On the other hand it has proved to be very hard to make computer systems perform many routine tasks that all humans and a lot of animals can do. Examples of such tasks include navigating our way without running into things, catching prey and avoiding predators. Humans and animals are also capable of interpreting complex sensory information. We are able to recognize objects and people from the visual image that we receive. We are also able to perform complex social functions.

Intelligent behaviour

This discussion brings us back to the question of what constitutes intelligent behaviour. Some of these tasks and applications are:

1. Perception involving image recognition and computer vision
2. Reasoning
3. Learning
4. Understanding language involving natural language processing, speech processing
5. Solving problems
6. Robotics

Practical applications of AI

AI components are embedded in numerous devices e.g. in copy machines for automatic correction of operation for copy quality improvement. AI systems are in everyday use for identifying credit card fraud, for advising doctors, for recognizing speech and in helping complex planning tasks. Then there are intelligent tutoring systems that provide students with personalized attention.

Thus AI has increased understanding of the nature of intelligence and found many applications. It has helped in the understanding of human reasoning, and of the nature of intelligence. It has also helped us understand the complexity of modeling human reasoning.

Approaches to AI

Strong AI aims to build machines that can truly reason and solve problems. These machines should be self aware and their overall intellectual ability needs to be indistinguishable from that of a human being. Excessive optimism in the 1950s and 1960s concerning strong AI has given way to an appreciation of the extreme difficulty of the problem. Strong AI maintains that suitably programmed machines are capable of cognitive mental states.

Weak AI: deals with the creation of some form of computer-based artificial intelligence that cannot truly reason and solve problems, but can act as if it were intelligent. Weak AI holds that suitably programmed machines can simulate human cognition.

Applied AI: aims to produce commercially viable "smart" systems such as, for example, a security system that is able to recognise the faces of people who are permittedto enter a particular building. Applied AI has already enjoyed considerable success.

Cognitive AI: computers are used to test theories about how the human mind works--for example, theories about how we recognise faces and other objects, or about how we solve abstract problems.

Limits of AI Today

Today"s successful AI systems operate in well-defined domains and employ narrow, specialized knowledge. Common sense knowledge is needed to function in complex, open-ended worlds. Such a system also needs to understand unconstrained natural language. However these capabilities are not yet fully present in today"s intelligent systems.

What can AI systems do

Today"s AI systems have been able to achieve limited success in some of these tasks.

- In Computer vision, the systems are capable of face recognition
- In Robotics, we have been able to make vehicles that are mostly autonomous.
- In Natural language processing, we have systems that are capable of simple machine translation.
- Today"s Expert systems can carry out medical diagnosis in a narrow domain
- Speech understanding systems are capable of recognizing several thousand words continuous speech
- Planning and scheduling systems had been employed in scheduling experiments with the Hubble Telescope.
- The Learning systems are capable of doing text categorization into about a 1000 topics
- In Games, AI systems can play at the Grand Master level in chess (world champion), checkers, etc.

What can AI systems NOT do yet?

- Understand natural language robustly (e.g., read and understand articles in a newspaper)
- Surf the web
- Interpret an arbitrary visual scene
- Learn a natural language
- Construct plans in dynamic real-time domains
- Exhibit true autonomy and intelligence

Applications:

We will now look at a few famous AI system that has been developed over the years.

1. **ALVINN:** Autonomous Land Vehicle In a Neural Network

In 1989, Dean Pomerleau at CMU created ALVINN. This is a system which learns to control vehicles by watching a person drive. It contains a neural network whose input is a 30x32 unit two dimensional camera image. The output layer is a representation of the direction the vehicle should travel.

The system drove a car from the East Coast of USA to the west coast, a total of about 2850 miles. Out of this about 50 miles were driven by a human, and the rest solely by the system.

2. Deep Blue

In 1997, the Deep Blue chess program created by IBM, beat the current world chess champion, Gary Kasparov.

3. Machine translation

A system capable of translations between people speaking different languages will be a remarkable achievement of enormous economic and cultural benefit. Machine translation is one of the important fields of endeavour in AI. While some translating systems have been developed, there is a lot of scope for improvement in translation quality.

4. Autonomous agents

In space exploration, robotic space probes autonomously monitor their surroundings, make decisions and act to achieve their goals.

NASA's Mars rovers successfully completed their primary three-month missions in April, 2004. The Spirit rover had been exploring a range of Martian hills that took two months to reach. It is finding curiously eroded rocks that may be new pieces to the puzzle of the region's past. Spirit's twin, Opportunity, had been examining exposed rock layers inside a crater.

5. Internet agents

The explosive growth of the internet has also led to growing interest in internet agents to monitor users' tasks, seek needed information, and to learn which information is most useful

What is soft computing?

An approach to computing which parallels the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision.

It is characterized by the use of inexact solutions to computationally hard tasks such as the solution of nonparametric complex problems for which an exact solution can"t be derived in polynomial of time.

Why soft computing approach?

Mathematical model & analysis can be done for relatively simple systems. More complex systems arising in biology, medicine and management systems remain intractable to conventional mathematical and analytical methods. Soft computing deals with imprecision, uncertainty, partial truth and approximation to achieve tractability, robustness and low solution cost. It extends its application to various disciplines of Engineering and science. Typically human can:

1. Take decisions
2. Inference from previous situations experienced
3. Expertise in an area
4. Adapt to changing environment
5. Learn to do better
6. Social behaviour of collective intelligence

Intelligent control strategies have emerged from the above mentioned characteristics of human/ animals. The first two characteristics have given rise to Fuzzy logic;2^{nd} , 3^{rd} and 4^{th} have led to Neural Networks; 4^{th} , 5^{th} and 6^{th} have been used in evolutionary algorithms.

Characteristics of Neuro-Fuzzy & Soft Computing:

1. Human Expertise
2. Biologically inspired computing models
3. New Optimization Techniques
4. Numerical Computation
5. New Application domains
6. Model-free learning
7. Intensive computation
8. Fault tolerance
9. Goal driven characteristics
10. Real world applications

Intelligent Control Strategies (Components of Soft Computing): The popular soft computing components in designing intelligent control theory are:

1. Fuzzy Logic
2. Neural Networks
3. Evolutionary Algorithms

Fuzzy logic:

Most of the time, people are fascinated about fuzzy logic controller. At some point of time in Japan, the scientists designed fuzzy logic controller even for household appliances like a room heater or a washing machine. Its popularity is such that it has been applied to various engineering products.

Fuzzy number or fuzzy variable:

We are discussing the concept of a fuzzy number. Let us take three statements: zero, almost zero, near zero. Zero is exactly zero with truth value assigned 1. If it is almost 0, then I can think that between minus 1 to 1, the values around 0 is 0, because this is almost 0. I am not very precise, but that is the way I use my day to day language in interpreting the real world. When I say near 0, maybe the bandwidth of the membership which represents actually the truth value. You can see that it is more, bandwidth increases near 0. This is the concept of fuzzy number. Without talking about membership now, but a notion is that I allow some small bandwidth when I say almost 0. When I say near 0 my bandwidth still further increases. In the case minus 2 to 2, when I encounter any data between minus 2 to 2, still I will consider them to be near 0. As I go away from 0 towards minus 2, the confidence level how near they are to 0 reduces; like if it is very near to 0, I am very certain. As I progressively go away from 0, the level of confidence also goes down, but still there is a tolerance limit. So when zero I am precise, I become imprecise when almost and I further become more imprecise in the third case.

When we say fuzzy logic, that is the variables that we encounter in physical devices, fuzzy numbers are used to describe these variables and using this methodology when a controller is designed, it is a fuzzy logic controller.

Neural networks :

Neural networks are basically inspired by various way of observing the biological organism. Most of the time, it is motivated from human way of learning. It is a learning theory. This is an artificial network that learns from example and because it is distributed in nature, fault tolerant, parallel processing of data and distributed structure.

The basic elements of artificial Neural Network are: input nodes, weights, activation function and output node. Inputs are associated with synaptic weights. They are all summed and passed through an activation function giving output y. In a way, output is summation of the signal multiplied with synaptic weight over many input channels.

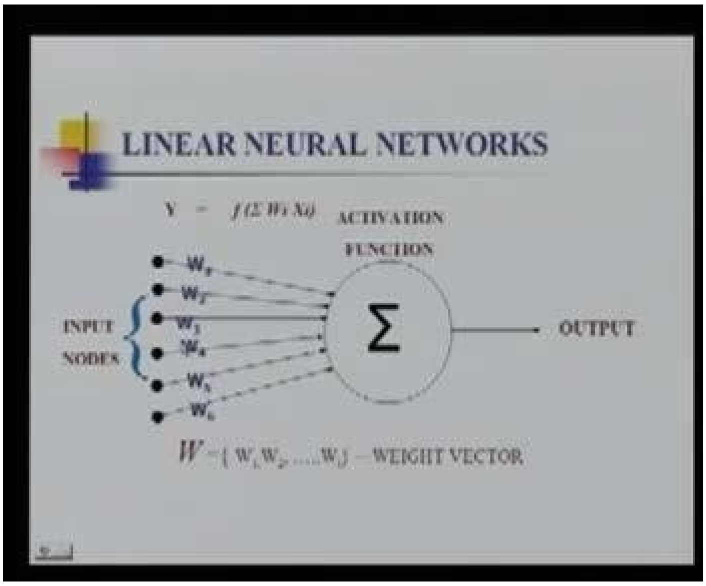


Fig. Basic elements of an artificial neuro

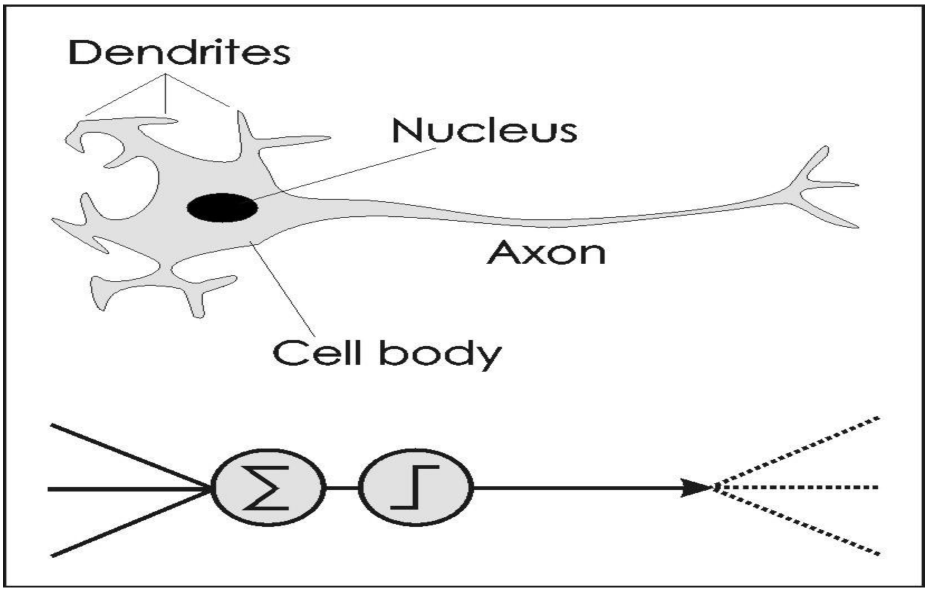


Fig. Analogy of biological neuron and artificial neuron

Above fig. Shows a biological neuron on top. Through axon this neuron actuates the signal and this signal is sent out through synapses to various neurons. Similarly shown a classical artificial neuron(bottom).This is a computational unit. There are many inputs reaching this. The input excites this neuron. Similarly, there are many inputs that excite this computational unit and the output again excites many other units like here. Like that taking certain concepts in actual neural network, we develop these artificial computing models having similar structure.

There are various locations where various functions take place in the brain.

If we look at a computer and a brain, this is the central processing unit and a brain. Let us compare the connection between our high speed computers that are available in the market today and a brain. Approximately there are 10 to the power of 14 synapses in the human brain, whereas typically you will have 10 to the power of 8 transistors inside a CPU. The element size is almost comparable, both are 10 to the power minus 6 and energy use is almost like 30 Watts and comparable actually; that is energy dissipated in a brain is almost same as in a computer. But you see the processing speed. Processing speed is only 100 hertz; our brain is very slow, whereas computers nowadays, are some Giga hertz.

When you compare this, you get an idea that although computer is very fast, it is very slow to do intelligent tasks like pattern recognition, language understanding, etc. These are certain activities which humans do much better, but with such a slow speed, 100 Hz contrast between these two, one of the very big difference between these two is the structure; one is brain, another is central processing unit is that the brain learns, we learn. Certain mapping that is found in biological brain that we have studied in neuroscience is not there in a central processing unit and we do not know whether self awareness takes place in the brain or somewhere else, but we know that in a computer there is no self-awareness.

Neural networks are analogous to adaptive control concepts that we have in control theory and one of the most important aspects of intelligent control is to learn the control parameters, to learn the system model. Some of the learning methodologies we will be learning here is the error-back propagation algorithm, real-time learning algorithm for recurrent network, Kohonen"s self organizing feature map & Hopfield network.

Features of Artificial Neural Network (ANN) models:

1. Parallel Distributed information processing
2. High degree of connectivity between basic units
3. Connections are modifiable based on experience
4. Learning is a continuous unsupervised process
5. Learns based on local information
6. Performance degrades with less units

All the methods discussed so far makes a strong assumption about the space around; that is, when we use whether a neural network or fuzzy logic or …. and …. any method that may have been adopted in intelligent control framework, they all make always very strong assumptions and normally they cannot work in a generalized condition. The question is that can they hypothesize a theory? When I design all these controllers, I always take the data; the engineer takes the data. He always builds these models that are updated. They update their own weights based on the feedback from the plant. But the structure of the controller, the model by which we assume the physical plant, all these are done by the engineer and also the structure of the intelligent controller is also decided by the engineer. We do not have a machine that can hypothesize everything; the model it should select, the controller it should select, looking at simply data. As it encounters a specific kind of data from a plant can it come up with specific controller architecture and can it come up with specific type of system model? That is the question we are asking now.

You will see that in the entire course we will be discussing various tools. They will only be dealing with these two things; behaviour. These tools are actually developed by mimicking the human behavior, but not the human way of working. An intelligent machine is one which learns, thinks and behaves in line with the thought process. That we would like but we are very far from it. At least, at the moment, we are very far from this target of achieving real intelligence.

We perceive the environment in a very unique way, in a coherent manner. This is called unity of perception and intelligence has also something to do with this unity of perception, awareness and certain things are not very clear to us until now. So an intelligent machine is one which learns, thinks & behaves in line with thought process.

Evolutionary algorithms:

These are mostly derivative free optimization algorithms that perform random search in a systematic manner to optimize the solution to a hard problem. In this course Genetic Algorithm being the first such algorithm developed in 1970"s will be discussed in detail. The other algorithms are swarm based that mimic behaviour of organisms, or any systematic process.

NEURAL NETWORK INTRODUCTION:

What is a neuron? A neuron is the basic processing unit in a neural network sitting on our brain. It consists of

1. Nucleus-
2. Axon- Output node
3. Dendrites-Input node
4. Synaptic junction

The dynamics of this synaptic junction is complex. We can see the signal inputs from the action of a neuron and through synaptic junction an output is actuated which is carried over through dendrites to another neuron. Here, these are the neurotransmitters. We learned from our experience that these synaptic junctions are either reinforced or in the sense they behave in such a way that the output of synaptic junction may excite a neuron or inhibit the neuron. This reinforcement of the synaptic weight is a concept that has been taken to artificial neural model.

The objective is to create artificial machine and this artificial neural networks are motivated by certain features that are observed in human brain, like as we said earlier, parallel distributed information processing.

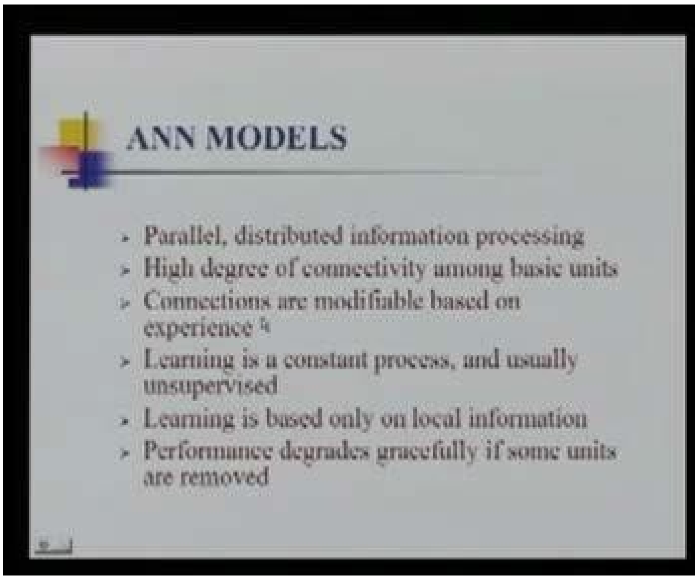


Fig. ANN model

Artificial neural networks are among the most powerful learning models. They have the versatility to approximate a wide range of complex functions representing multi-dimensional input-output maps. Neural networks also have inherent adaptability, and can perform robustly even in noisy environments.

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected simple processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. ANNs can process information at a great speed owing to their highly massive parallelism.

A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

Advantages of ANN:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.

2. Self-Organisation: An ANN can create its own organisation or representation of the information it receives during learning time.

3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Table- Difference between the brain and a digital Computer

Property	Computer	Brain
Shape	2d Sheets of inorganic matter	3d volume of organic matter
Power	Powered by DC mains	Powered by ATP
Signal	Digital	pulsed
Clock	Centralized clock	No centralized clock
Clock speed	Gigahertz	100s of Hz
Fault tolerance	Highly fault-sensitive	Very fault-tolerant
Performance	By programming	By learning

Differences human brain & ANN:

1. Computer has such fast speed of GHz, a traditional computer, however, when it comes to certain processing like pattern recognition and language understanding, the brain is very fast.
2. Intelligence and self-awareness, are absent in an artificial machine.

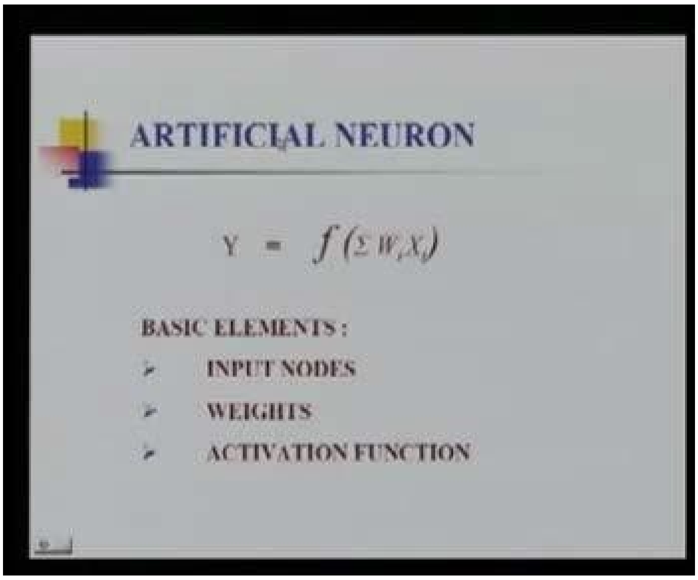


Fig. An artificial neuron

Basic computational unit in an artificial neural network is neuron. Obviously, it has to be an artificial neuron.

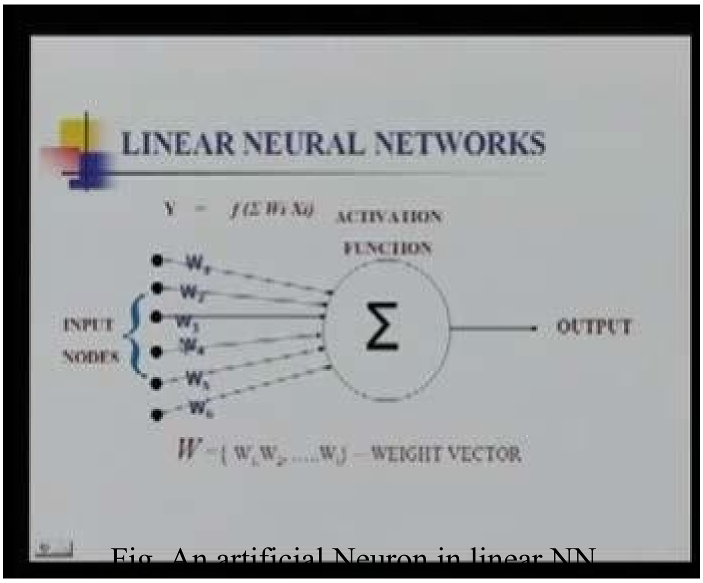


Fig. An artificial Neuron in linear NN

This artificial neuron has three basic elements:

1. Nodes,
2. Weights and
3. Activation function.

Between input nodes and output nodes, there are synaptic weights w_1, w_2, w_3, w_4, w_5 and w_6. There can be as many weights and these weights are multiplied with the signal as they reach the output unit, where the output is simply sum of the signal multiplied with the weights and then this output goes to an activation function f.

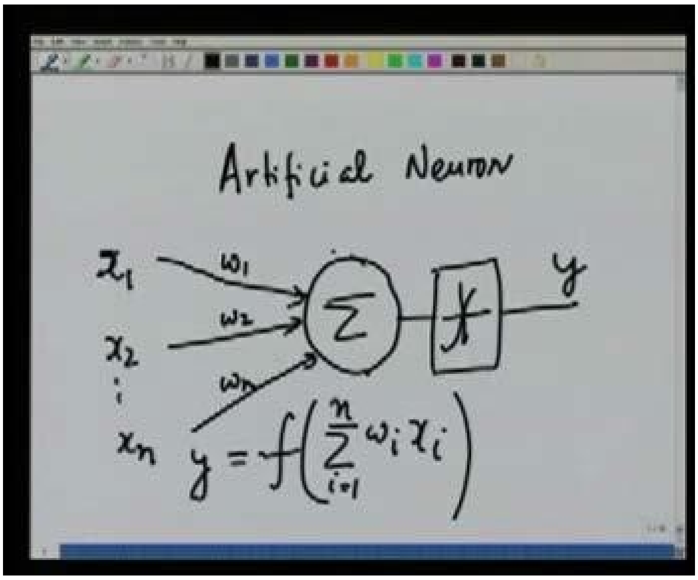


Fig. Basic processing unit- the neuron

What you are seeing is actually a nonlinear map from input vector x to output y. A single neuron has single output but multiple inputs. Inputs are multiple for a single neuron and the output is unique, y and this output y and the input bear a nonlinear relationship, by f. Neural networks can be built using this single neuron. We can use the single neuron and build neural networks.

Analogy to brain:

Artificial Neural Network (ANN) is a system which performs information processing. An ANN resembles or it can be considered as a generalization of mathematical model of human brain assuming that

1. Information processing occurs at many simple elements called neurons.
2. Signals are passed between neurons over connection links.
3. Each connection link has an associated weight, which in a typical neural net multiplies the signal transmitted.

ANN is built with basic units called *neurons* which greatly resemble the neurons of human brain. A neural net consists of a large number of simple processing elements called neurons. Each neuron applies an activation function to its net input to determine its output signal. Every neuron is connected to other neurons by means of directed communication links, each with an associated weight. Each neuron has an internal state called its *activation level*, which is a function of the inputs it has received. As and when the neuron receives the signal, it gets added up and when the cumulative signal reaches the activation level the neuron sends an output. Till then it keeps receiving the input. So activation level can be considered as a threshold value for us to understand.

In general, a neural network is characterized by

1. *Pattern of connections* between the neurons called its *architecture*
2. Method of *determining the weights* on the connections called its training or *learning algorithm*
3. Its *internal state* called its *Activation function*.

The arrangement of neurons into layers and the connection patterns within and between layers is called the net *architecture*. A neural net in which the signals flow from the input units to the output units in a forward direction is called *feed forward nets*.

Interconnected competitive net in which there are closed loop signal paths from a unit back to it is called a *recurrent network*. In addition to architecture, the method of setting the values of the weights called *training* is an important characteristic of neural nets. Based on the training methodology used neural nets can be distinguished into *supervised* or *unsupervised* neural nets. For a neural net with supervised training, the training is accomplished by presenting a sequence of training vectors or patterns each with an associated target output vector. The weights are then adjusted according to a learning algorithm. For neural nets with unsupervised training, a sequence of input vectors is provided, but no target vectors are specified. The net modifies the weights so that the most similar input vectors are assigned to the same output unit. The neural net will produce a representative vector for each cluster formed. Unsupervised learning is also used for other tasks, in addition to clustering.

Activation function

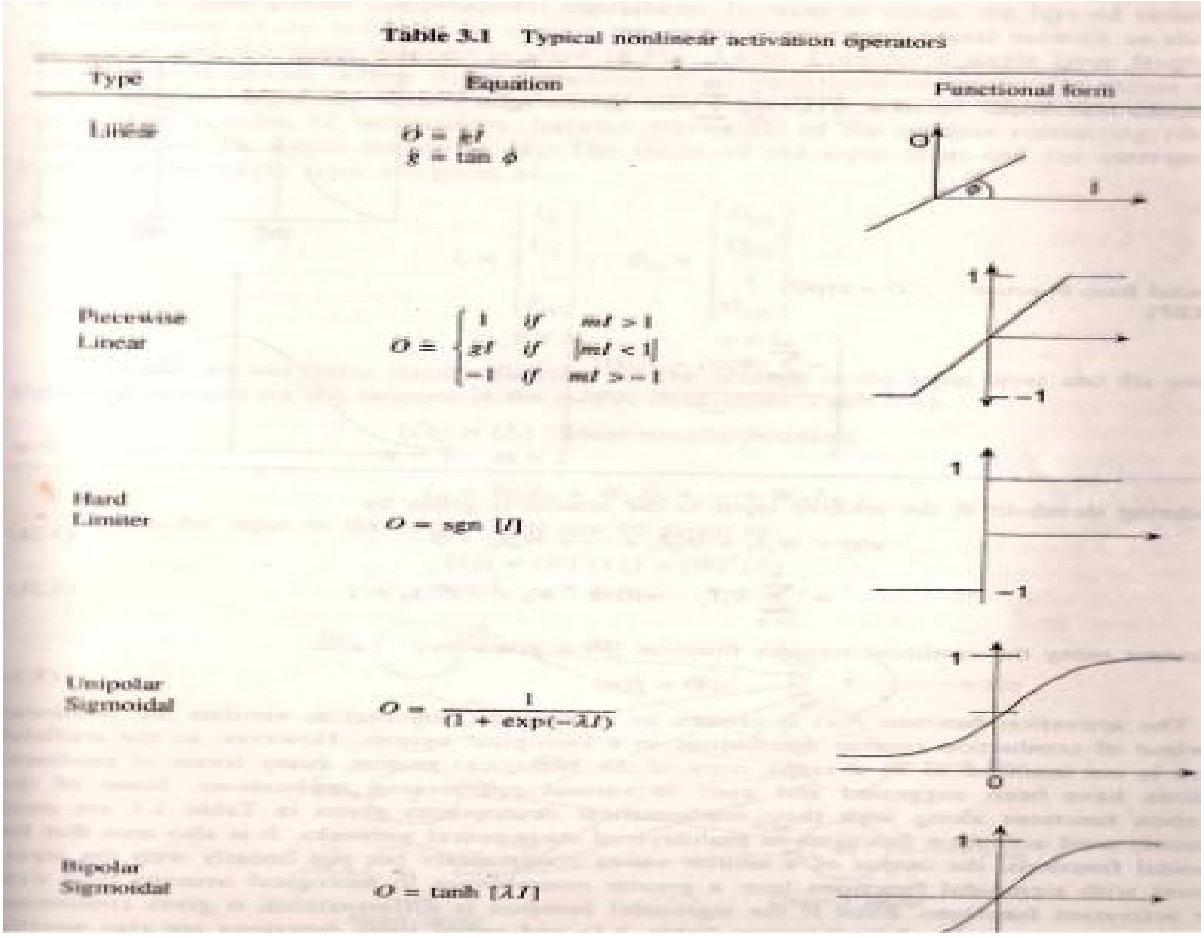


Table 3.1 Typical nonlinear activation operators

Type	Equation	Functional form
Linear	$O = gI$ $g = \tan\phi$	
Piecewise Linear	$O = \begin{cases} 1 & if \quad mI > 1 \\ gI & if \quad \lvert mI < 1 \rvert \\ -1 & if \quad mI > -1 \end{cases}$	
Hard Limiter	$O = \mathrm{sgn}\,[I]$	
Unipolar Sigmoidal	$O = \dfrac{1}{(1 + \exp(-\lambda I))}$	
Bipolar Sigmoidal	$O = \tanh\,[\lambda I]$	

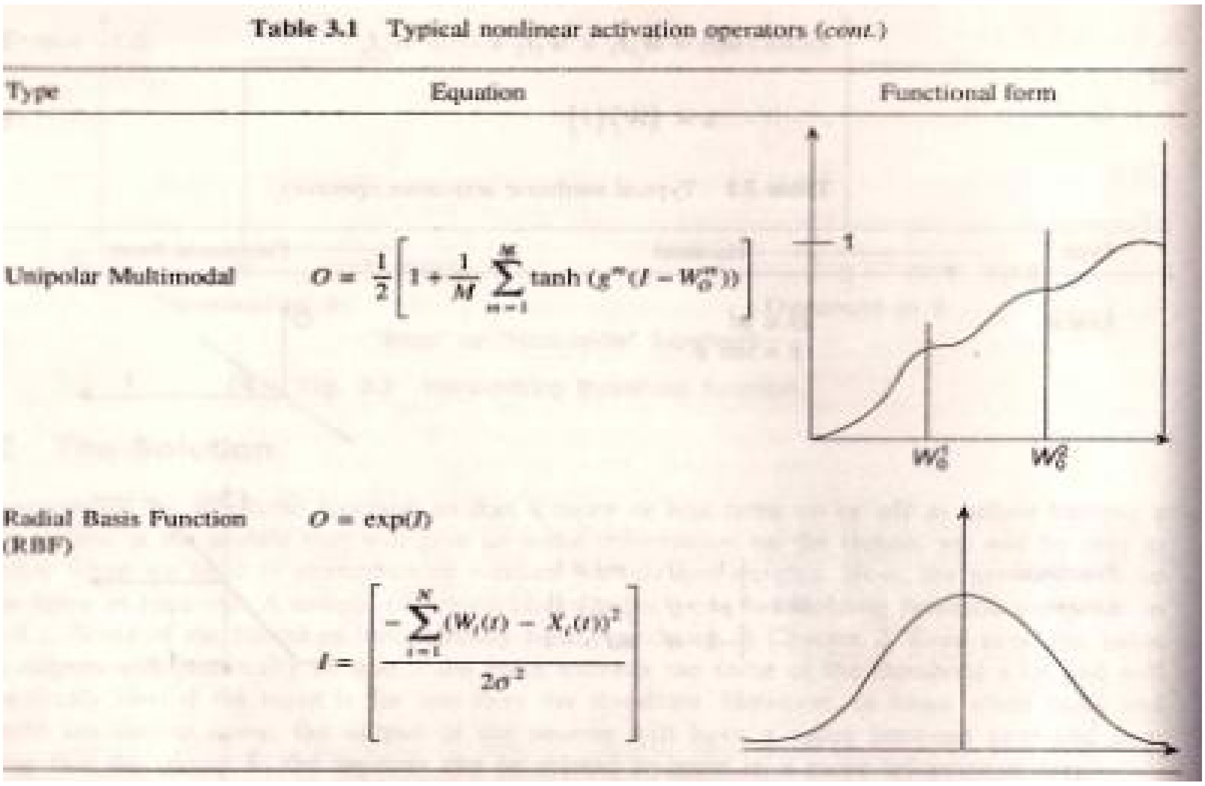


Table 3.1 Typical nonlinear activation operators (*cont.*)

Type	Equation	Functional form
Unipolar Multimodal	$O = \frac{1}{2}\left[1 + \frac{1}{M}\sum_{m=1}^{M} \tanh\left(g^m(I - W_0^m)\right)\right]$	
Radial Basis Function (RBF)	$O = \exp(I)$ $I = \left[\dfrac{-\sum_{i=1}^{N}(W_i(t) - X_i(t))^2}{2\sigma^2}\right]$	

Architecture:

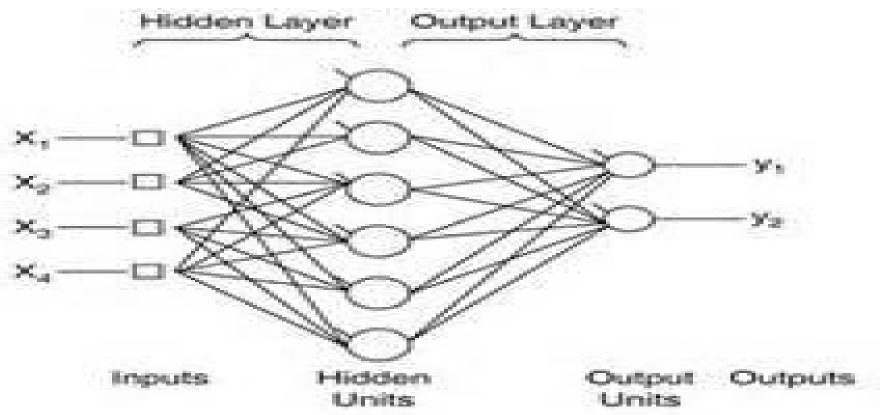


Fig. Architecture of multilayer Neural network

Artificial neural networks are represented by a set of nodes, often arranged in layers, and a setof weighted directed links connecting them. The nodes are equivalent to neurons, while the links denote synapses. The nodes are the information processing units and the links acts as communicating media.

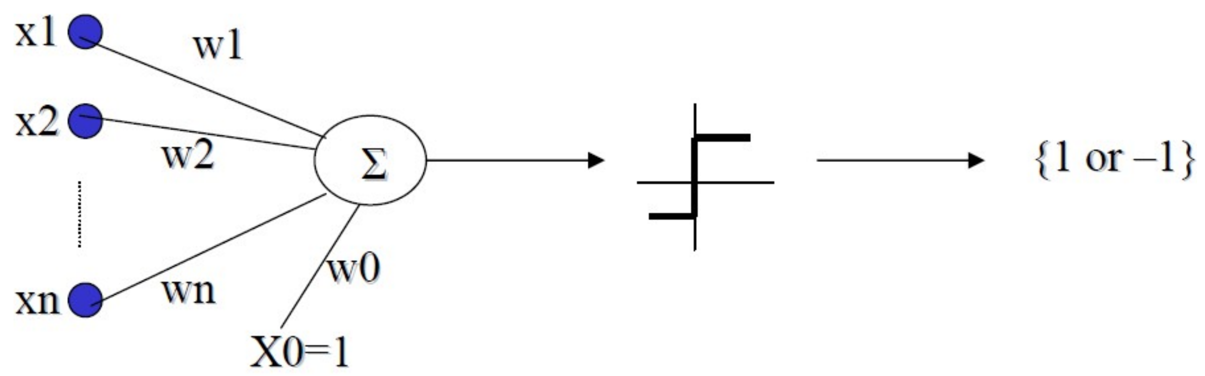
A neural network may have different layers of neurons like

1. input layer,
2. hidden layer,
3. output layer.

The input layer receives input data from the user and propagates a signal to the next layer called the hidden layer. While doing so it multiplies the weight along with the input signal. The hidden layer is a middle layer which lies between the input and the output layers. The hidden layer with non linear activation function increases the ability of the neural network to solve many problems than the case without the hidden layer. The output layer sends its calculated output to the user from which decision can be made. Neural nets can also be classified based on the above stated properties.

There are a wide variety of networks depending on the nature of information processing carried out at individual nodes, the topology of the links, and the algorithm for adaptation of link weights. Some of the popular among them include:

Perceptron: *Definition:* It"s a step function based on a linear combination of real-valued inputs. If the combination is above a threshold it outputs a 1, otherwise it outputs a –1. This consists of a single neuron with multiple inputs and a single output. It has restricted information processing capability. The information processing is done through a transfer function which is either linear or non-linear.



$$O(x1,x2,\ldots,xn) = \begin{cases} 1 \text{ if } w0 + w1x1 + w2x2 + \ldots + wnxn > 0 \\ -1 \text{ otherwise} \end{cases}$$

Fig. A perceptron

A perceptron can learn only examples that are called “linearly separable”. These are examples that can be perfectly separated by a hyperplane.

Perceptrons can learn many boolean functions: AND, OR, NAND, NOR, but not XOR

However, every boolean function can be represented with a perceptron network that has two levels of depth or more.

The weights of a perceptron implementing the AND function is shown below.

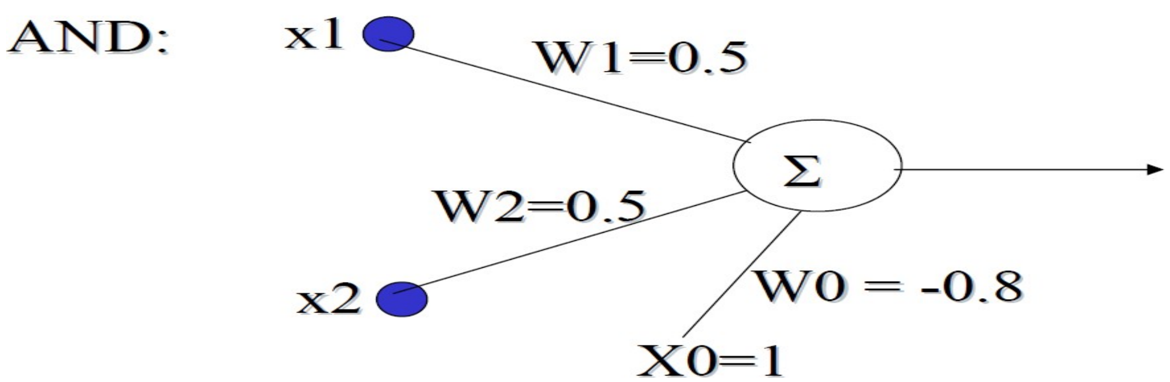


Fig. AND operation on inputs by a single perceptron

Multi-layered Perceptron (MLP): It has a layered architecture consisting of input, hidden and output layers. Each layer consists of a number of perceptrons. The output of each layer is transmitted to the input of nodes in other layers through weighted links. Usually, this transmission is done only to nodes of the next layer, leading to what are known as feed forward networks. MLPs were proposed to extend the limited information processing capabilities of simple perceptrons, and are highly versatile in terms of their approximation ability. Training or weight adaptation is done in MLPs using supervised backpropagation learning.

Adding a hidden layer:

The perceptron, which has no hidden layers, can classify only linearly separable patterns.

The MLP, with at least 1 hidden layer can classify *any* linearly non-separable classes also.

An MLP can approximate any continuous multivariate function to any degree of accuracy, provided there are sufficiently many hidden neurons (Cybenko, 1988; Hornik et al, 1989). A more precise formulation is given below.

A serious limitation disappears suddenly by adding a single hidden layer.

It can easily be shown that the XOR problem which was not solvable by a Perceptron can be solved by a MLP with a single hidden layer containing two neurons.

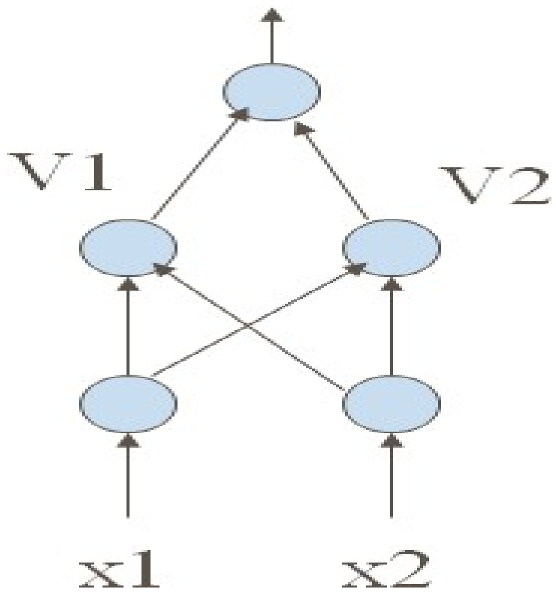


Figure 6.2.1.1: MLP for solving Xor

Recurrent Neural Networks: RNN topology involves backward links from output to the input and hidden layers. The notion of time is encoded in the RNN information processing scheme. They are thus used in applications like speech processing where inputs are time sequences data.

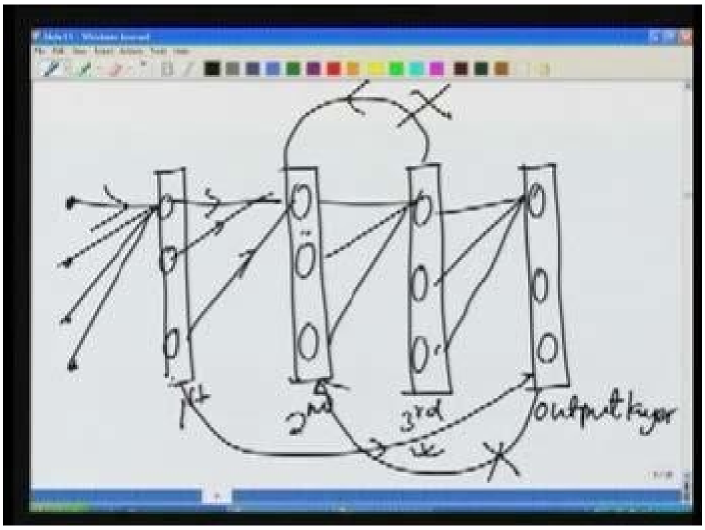


Fig. Multilayer feed back network (Recurrent Neural Network)

Self-Organizing Maps: SOMs or Kohonen networks have a grid topology, wit unequal grid weights. The topology of the grid provides a low dimensional visualization of the data distribution. These are thus used in applications which typically involve organization and human browsing of a large volume of data. Learning is performed using a winner take all strategy in a unsupervised mode. It is described in detail later.

Single layer Network:

A neural net with only input layer and output layer is called single layer neural network. A neural network with input layer, one or more hidden layers and an output layer is called a multilayer neural network. A single layer network has limited capabilities when compared to the multilayer neural networks.

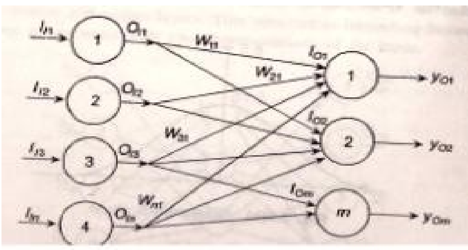


Fig. Single Layer feed foreward Neural Network

Steps in developing NN:

Network formation

Neural network consists of an input layer, an output layer and a hidden layer. While a neural network is constructed, the number of neurons in each layer has to be fixed. The input layer will have neurons whose number will be equal to the number of features extracted. The number of neurons in the output layer will be equal to the number of pattern classes. The number of neurons in the hidden layer is decided by trial and error basis. With a minimum number of neurons in the hidden layer, the neural network will be constructed and the convergence will be checked for. Then the error will be noted. The number of neurons for which the error is minimum, can be taken and will be checked for reduced error criterion.

Data preprocessing and normalization

Data selection and pre processing can be a demanding and intricate task. Neural net is as good as the input data used to train it. If important data inputs are missing, then the effect on the neural network‟s performance can be significant. The most appropriate raw input data must be preprocessed. Otherwise the neural network will not produce accurate results. Transformation and normalization are two widely used preprocessing methods. Transformation involves manipulating raw data inputs to create a single input to a net, while normalization is a transformation performed on a single data input to distribute the data evenly and scale it into an acceptable range for the network. Knowledge of the domain is important in choosing preprocessing methods to highlight the features in the data, which can increase the ability of the network to learn the association between inputs and outputs. Data normalization is the final preprocessing step. In normalizing data, the goal is to ensure that the statistical distribution of values should be scaled to match the range of the input neurons. The simplest method of normalization can be done using the formula

X normalized = $(X-\mu) / \sigma$ where μ and σ are the mean and standard deviation of the input data.

Perceptron Learning

Learning a perceptron means finding the right values for W. The hypothesis space of a perceptron is the space of all weight vectors.

The perceptron learning algorithm can be stated as below.

1. Assign random values to the weight vector

2. Apply the *weight update rule* to every training example

3. Are all training examples correctly classified?

a. Yes. Quit

b. No. Go back to Step 2.

There are two popular weight update rules.

i) The perceptron rule, and

ii) Delta rule

The Perceptron Rule

For a new training example X = (x1, x2, …, xn), update each weight according to this rule:

wi = wi + Δwi

Where Δwi = η (t-o) xi

t: target output

o: output generated by the perceptron

η: constant called the learning rate (e.g., 0.1)

Comments about the perceptron training rule:

Example means training data.

• If the example is correctly classified the term (t-o) equals zero, and no update on the weight is necessary.

• If the perceptron outputs –1 and the real answer is 1, the weight is increased.

• If the perceptron outputs a 1 and the real answer is -1, the weight is decreased.

• Provided the examples are linearly separable and a small value for η is used, the rule is proved to classify all training examples correctly (i.e, is consistent with the training data).

The Delta Rule

What happens if the examples are <u>not linearly separable</u>?

To address this situation we try to approximate the real concept using the delta rule.

The key idea is to use a *gradient descent search*. We will try to minimize the following error:

E = ½ Σi (ti – oi) 2

where the sum goes over all training examples. Here oi is the inner product WX and not sgn(WX) as with the perceptron rule. The idea is to find a minimum in the space of weights and the error function E.

The delta rule is as follows:

For a new training example X = (x1, x2, …, xn), update each weight according to this rule:

wi = wi + Δwi

Where Δwi = -η E‟(W)/wi

η: learning rate (e.g., 0.1)

It is easy to see that

E‟(W)/ wi = Σi (ti – oi) (-xi)

So that gives us the following equation:

wi = η Σi (ti – oi) xi

There are two differences between the perceptron and the delta rule. The perceptron is based on an output from a <u>step function</u>, whereas the delta rule uses the linear combination of inputs directly. The perceptron is guaranteed to converge to a consistent hypothesis assuming the data is linearly separable. The delta rules converges in the limit but it does not need the condition of linearly separable data.

There are two main **difficulties with the gradient descent method:**

1. Convergence to a minimum may take a long time.

2. There is no guarantee we will find the global minimum.

These are handled by using momentum terms and random perturbations to the weight vectors.