**Chapter 18**

**Artificial Neural Network**

**The chapter consist of Short type Questions &Answers , Descriptive Question & Answer and MCQs & answers.**

Contents

[Short type Questions & Answers 1](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869683)

[Q Define Artificial Neural Network 2](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869684)

[Q What is data normalization and why do we need it? 2](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869685)

[Q Why it is an advantage to have threshold set to zero 2](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869686)

[Q What is the role of the activation function? 2](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869687)

[Q Which are Basic Models of Artificial Neural Networks 2](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869688)

[Q What is deep learning? 3](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869689)

[Descriptive Question & Answer 3](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869690)

[Q Explain Features Of Artificial Network (Ann) 3](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869691)

[Q. What is a backpropagation? 4](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869692)

[Q Explain the following three variants of gradient descent: batch, stochastic and mini-batch? 5](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869693)

[Q How neural network works? 5](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869694)

[**Q Explain the concept of Multiplayer perceptron** 6](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869695)

[Q Below is a diagram if a single artificial neuron (unit) 7](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869696)

[**Q Consider the unitshownon Figure 1.Supposethattheweightscorrespond- ing to the three inputs have the followingvalues:** 8](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869697)

[Q Explain Delta Learning Rule 9](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869698)

[Q For each of the truth tables below say whether it is possible for a perceptron to learn the required output. 10](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869699)

[Q Write a short note on Backpropagation 11](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869700)

[Q A perceptron with two inputs has a threshold level set at the point at which it will fire (i.e. output a one). It is sometimes convenient to always set the threshold level to zero. Show how this can be achieved by describing two perceptrons which act in the same way but one has its threshold set to a non-zero figure and the other perceptron has a zero threshold. Why might it be a good idea to build a perceptron with a zero threshold figure? 13](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869701)

[Q Explain the concept of Back Propagation neural network 17](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869702)

[MCQs & answers. 21](file:///C:\0.%20RNDas_WORKING%20FOLDER\D%20Drive%20Data%20from%20Desktop%20PC\0.%20RNDas_WILEY\Intelligent%20System\AI_RESOURCES\AI_Students'%20Resources\AI_App%20B_Testing\Chapter_18.docx#_Toc14869703)

# Short type Questions & Answers

## Q Define Artificial Neural Network

The inventor of the first neurocomputer, Dr. Robert Hecht-Nielsen, defines a neural network as −

"...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.”

## Q What is data normalization and why do we need it?

**Answer:**  
Data normalization is used during backpropagation. The main motive behind data normalization is to reduce or eliminate data redundancy. Here we rescale values to fit into a specific range to achieve better convergence.

## Q Why it is an advantage to have threshold set to zero

When learning, the algorithm only has to adjust weights and not thresholds and weights.

## Q What is the role of the activation function?

**Answer:**  
The activation function is used to introduce non-linearity into the [neural network](https://www.educba.com/neural-networks-vs-deep-learning/) helping it to learn more complex function. Without which the neural network would be only able to learn linear function which is a linear combination of its input data.

## Q Which are Basic Models of Artificial Neural Networks

There are various Artificial Neural Network Model. Main ones are

* Multilayer Perceptron – It is a feedforward artificial neural network model. It maps sets of input data onto a set of appropriate outputs.
* Radial Basis Function Network – A radial basis function network is an artificial neural network. It uses radial basis functions as *activation functions.*

Both of the above are being supervised learning networks used with 1 or more dependent variables at the output.

* The Kohonen Network – It is an unsupervised learning network used for *clustering.*

## Q What is deep learning?

**Answer:**  
The area of [machine learning](https://www.educba.com/course/machine-learning-python-basic-tutorials/) which focuses on deep artificial neural networks which are loosely inspired by brains. Alexey GrigorevichIvakhnenko published the first general on working [Deep Learning network](https://www.educba.com/course/deep-learning-tutorials/). Today it has its application in various fields such as computer vision, speech recognition, [natural language](https://www.educba.com/course/natural-language-processing-nlp-tutorials/) processing.

# Descriptive Question & Answer

## Q Explain Features Of Artificial Network (Ann)

Artificial neural networks may by physical devices or simulated on conventional computers. From a practical point of view, an ANN is just a parallel computational system consisting of many simple processing elements connected together in a specific way in order to perform a particular task. There are some important features of artificial networks as follows.

(1)    Artificial neural networks are extremely powerful computational devices (Universal computers).

(2)    ANNs are modeled on the basis of current brain theories, in which information is represented by weights.

(3)    ANNs have massive parallelism which makes them very efficient.

(4)    They can learn and generalize from training data so there is no need for enormous feats of programming.

(5)    Storage is fault tolerant i.e. some portions of the neural net can be removed and there will be only a small degradation in the quality of stored data.

(6)    They are particularly fault tolerant which is equivalent to the “graceful degradation” found in biological systems.

(7)    Data are naturally stored in the form of associative memory which contrasts with conventional memory, in which data are recalled by specifying address of that data.

(8)    They are very noise tolerant, so they can cope with situations where normal symbolic systems would have difficulty.

(9)    In practice, they can do anything a symbolic/ logic system can do and more.

(10)        Neural networks can extrapolate and intrapolate from their stored information. The neural networks can also be trained. Special training teaches the net to look for significant features or relationships of data.

## Q. What is a backpropagation?

**Answer:**  
Backpropagation is training algorithm used for multilayer neural network. In this method, we move the error from an end of the network to all weights inside the network and thus allowing efficient computation of the gradient. It can be divided into several steps as follows:-

¬Forward propagation of training data in order to generate output.  
¬Then using target value and output value error [derivative](https://www.educba.com/course/derivatives-2/) can be computed with respect to output activation.  
¬Then we back propagate for computing derivative of error with respect to output activation on previous and continue this for all the hidden layers.  
¬Using previously calculated derivatives for output and all hidden layers we calculate error derivatives with respect to weights.  
¬And then we update the weights.

## Q Explain the following three variants of gradient descent: batch, stochastic and mini-batch?

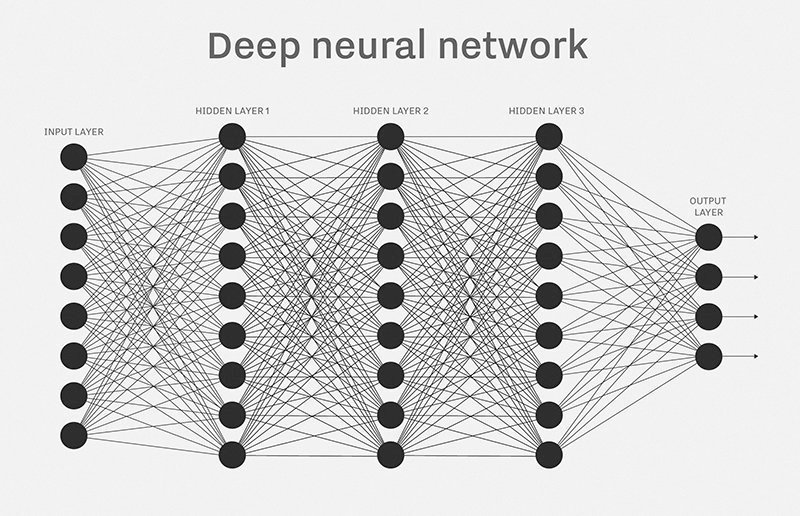
**Answer:**  
**Stochastic Gradient Descent**: Here we use only single training example for calculation of gradient and update parameters.  
**Batch Gradient Descent**: Here we calculate the gradient for the whole dataset and perform the update at each iteration.  
**Mini-batch Gradient Descent**: It’s one of the most popular optimization algorithms. It’s a variant of Stochastic Gradient Descent and here instead of single training example, mini-batch of samples is used.

## Q How neural network works?

A neural network sometimes involves an oversized range of processors operational in parallel and organized in tiers.The first tier receives the raw input data analogous to optic nerves in human visual process.Each serial tier receives the output from the tier preceding it, instead of from the raw input – withinthe same manner neurons away from the second cranial nerve receivesignals from those closer to it.The last tier produces the output of the system.

Each process node has its own little sphere of information, as well as what it's seen and any rules it absolutely was originally programmed with or developed for itself.The tiers are highly interconnected, which means each node in tier n will be connected to many nodes in tier n-1its inputsand in tier n+1, which provides input for those nodes.There is also one or multiple nodes within the output layer, from that the solution it produces will be browse.

Neural networks are notable for being adjustive, which implies they modify themselves as they learn from initial coaching and ulterior runs offer a lot of dataconcerning the globe.The most basic learning model is focused on weight the input streams, which is how each node weights the importance of input from each of its predecessors.Inputs that contribute to obtaining right answers are weighted higher.



## Q Explain the concept of Multiplayer perceptron

As we tend to saw higher than, A multilayer perceptron may be a feedforward artificial neural network model.

It maps sets of input data onto a set of appropriate outputs.

In feed-forward neural networks, the movement is only possible in the forward direction.

An MLP consists of many layers of nodes in a directed graph, with each layer connected to the next one. Each neuron is a linear equation like linear regression as shown in the following equation

Yi = W0 + W 1 \* X1 + W 2 \* X2 + …..+ W n \* Xn

The equation is the transfer function in a neural network. This linear weight sum would be a threshold at some value so that output of neuron would be either 1 or 0.  
The multilayer perceptron networks are suitable for the discovery of complex nonlinear models.

On the chance of approximating any regular perform with a add of sigmoid its power based mostly.

MLP utilizes a supervised learning technique known as backpropagation for coaching the network.

This requires a best-known, desired output for each input value to calculate the loss function gradient.

MLP may be a modification of the quality linear perceptron and may distinguish information that don't seem to be linearly dissociable.

## Q Below is a diagram if a single artificial neuron (unit)

## 

The node has three inputs **x** = (*x*1*, x*2*, x*3) that receive only binary signals (either 0 or 1). How many different input patterns this node can receive? What if the node had four inputs? Five? Can you give a formula that computesthenumberofbinaryinputpatternsforagivennumberofinputs?

**Answer:**For three inputs the number of combinations of 0 and 1 is 8:

|  |  |
| --- | --- |
| *x*1 | 0 1 0 1 0 1 0 1 |
| *x*2 | 0 0 1 1 0 0 1 1 |
| *x*3 | 0 0 0 0 1 1 1 1 |

and for four inputs the number of combinations is 16:

|  |  |
| --- | --- |
| *x*1 | 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 |
| *x*2 | 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 |
| *x*3 | 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 |
| *x*4 | 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 |

You may check that for five inputs the number of combinations will be 32. Note that 8 = 23, 16 = 24 and 32 = 25 (for three, four and five inputs). Thus,theformulaforthenumberofbinaryinputpatternsis:

2n, whereninthenumberofinputs

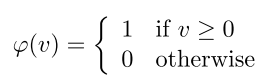
## Q Consider the unitshownon Figure 1.Supposethattheweightscorrespond- ing to the three inputs have the followingvalues:

*w*3 = 1

*w*2 =*−*4

*w*1 = 2

**and the activation of the unit is given by the step-function:**



Calculatewhatwillbetheoutputvalue*y*oftheunitforeachofthefollowing inputpatterns:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pattern | *P*1 | *P*2 | *P*3 | *P*4 |
| *x*1 | 1 | 0 | 1 | 1 |
| *x*2 | 0 | 1 | 0 | 1 |
| *x*3 | 0 | 1 | 1 | 1 |

**Answer:** To find the output value y for each pattern we haveto:

1. *Calculate the weighted sum:*
2. *Apply the activation function tov*

The calculations for each input pattern are:

*P*1: *v*=2*·*1*−*4*·*0+1*·*0=2*,* (2 *>*0)*, y* = *ϕ*(2) =1

*P*2:*v* = 2 *·* 0 *−* 4 *·* 1 + 1 *·* 1 = *−*3 *,* (*−*3 *<*0) *, y* = *ϕ*(*−*3) =0

*P*3: *v*=2*·*1*−*4*·*0+1*·*1=3*,* (3 *>*0)*, y* = *ϕ*(3) =1

*P*4:*v* = 2 *·* 1 *−* 4 *·* 1 + 1 *·* 1 = *−*1 *,* (*−*1 *<*0) *,y* = *ϕ*(*−*1) = 0

## Q Explain Delta Learning Rule

Developed by *Widrow* and *Hoff*, the delta rule, is one of the most common learning rules. It depends on supervised learning.  
This rule states that the modification in sympatric weight of a node is equal to the multiplication of error and the input.  
In Mathematical form the delta rule is as follows:

Mathematical Formula of Delta Learning Rule in Artificial Neural Network.

Mathematical Formula of Delta Learning Rule in Artificial Neural Network.

For a given input vector, compare the output vector is the correct answer. If the difference is zero, no learning takes place; otherwise, adjusts its weights to reduce this difference. The change in weight from ui to uj is: dwij = r\* ai \* ej.  
where r is the learning rate, ai represents the activation of ui and ej is the difference between the expected output and the actual output of uj. If the set of input patterns form an independent set then learn arbitrary associations using the delta rule.  
It has seen that for networks with linear activation functions and with no hidden units. The error squared vs. the weight graph is a paraboloid in n-space. Since the proportionality constant is negative, the graph of such a function is concave upward and has the least value. The vertex of this paraboloid represents the point where it reduces the error. The weight vector corresponding to this point is then the ideal weight vector.  
We can use the delta learning rule with both single output unit and several output units.  
While applying the delta rule assume that the error can be directly measured.  
The aim of applying the delta rule is to reduce the difference between the actual and expected output that is the error.

## Q For each of the truth tables below say whether it is possible for a perceptron to learn the required output.

**In each case, explain the reason behind your decision.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **i)** | **Input** | **0** | **0** | **1** | **1** |
|  | **Input** | **0** | **1** | **0** | **1** |
|  | **Required Output** | **1** | **0** | **0** | **1** |
|  |  |  |  |  |  |
| **ii)** | **Input** | **0** | **0** | **1** | **1** |
|  | **Input** | **0** | **1** | **0** | **1** |
|  | **Required Output** | **1** | **1** | **0** | **0** |
|  |  |  |  |  |  |
| **iii)** | **Input** | **0** | **0** | **1** | **1** |
|  | **Input** | **0** | **1** | **0** | **1** |
|  | **Required Output** | **1** | **1** | **1** | **1** |

Only i) cannot be learnt. This is because it is not ***linearly separable***. This can be shown on the diagrams below (where the outputs have been plotted and the filled circles represent a 1 and the hollow circles represent a zero.).

Those problems which are linearly separable can have a line dividing the "1" outputs from the "0" outputs. In the case of i) this is not possible.

**i) ii)**

1,0

0,1

1,1

0,0

1,0

0,1

1,1

0,0

**iii)**

1,0

0,1

1,1

0,0

## Q Write a short note on Backpropagation

Backpropagation is a supervised learning algorithm, for training Multi-layer Perceptrons (Artificial Neural Networks).

The Backpropagation algorithm looks for the minimum value of the error function in weight space using a technique called the delta rule or gradient descent. The weights that minimize the error function is then considered to be a solution to the learning problem.

**Why We Need Backpropagation?**

While designing a Neural Network, in the beginning, we initialize weights with some random values or any variable for that fact.

Now obviously, we are not superhuman. So, it’s not necessary that whatever weight values we have selected will be correct, or it fits our model the best.

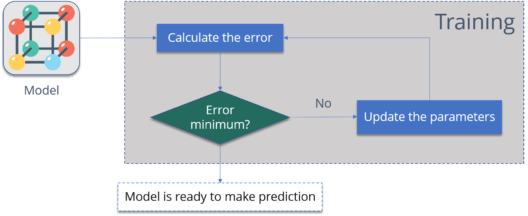
Okay, fine, we have selected some weight values in the beginning, but our model output is way different than our actual output i.e. the error value is huge.

Now, how will you reduce the error?

Basically, what we need to do, we need to somehow explain the model to change the parameters (weights), such that error becomes minimum.

Let’s put it in an another way, we need to train our model.

One way to train our model is called as Backpropagation. Consider the diagram below:



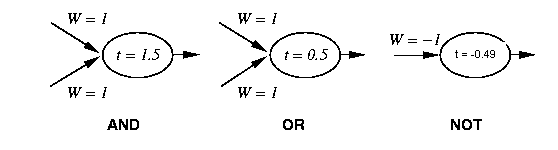
Let me summarize the steps for you:

* **Calculate the error** – How far is your model output from the actual output.
* **Minimum Error** – Check whether the error is minimized or not.
* **Update the parameters** – If the error is huge then, update the parameters (weights and biases). After that again check the error. Repeat the process until the error becomes minimum.
* **Model is ready to make a prediction** – Once the error becomes minimum, you can feed some inputs to your model and it will produce the output.

## Q A perceptron with two inputs has a threshold level set at the point at which it will fire (i.e. output a one). It is sometimes convenient to always set the threshold level to zero. Show how this can be achieved by describing two perceptrons which act in the same way but one has its threshold set to a non-zero figure and the other perceptron has a zero threshold. Why might it be a good idea to build a perceptron with a zero threshold figure?

#### Perceptrons with two inputs and the threshold non-zero

A perceptron (with two inputs) to act as a logic gate could be modelled as follows (three examples are shown – students would only need to show one example).



If we consider the AND function we can see that it acts correctly for the four possible inputs (see table below).

Note in the table Sum is defined as (Input 1 \* Weight 1) + (Input 2 \* Weight 2)

Step(t) is defined as returning 1 if Sum => t else 0 and the output of Step(t) is also the output (or activation level) of the perceptron.

AND

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input 1** | **Input 2** | **Weight 1** | **Weight2** | **Sum** | **Step(t)** |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 2 | 1 |

Similarly, OR and NOT can be shown as follows

OR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input 1** | **Input 2** | **Weight 1** | **Weight 2** | **Sum** | **Step(t)** |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0.5 | 1 |
| 1 | 0 | 1 | 1 | 0.5 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

NOT

|  |  |  |  |
| --- | --- | --- | --- |
| **Input 1** | **Weight 1** | **Sum** | **Step(t)** |
| 0 | -1 | 0 | 1 |
| 1 | 01 | -0.49 | 0 |

Perceptron with three inputs and the threshold set to zero

It is possible to have an extra input whose activation is set to –1 and the weight from that input unit to the output neuron is set to the required threshold level. Diagramatically this can be shown as follows

t = 0.0

y

x

-1

W = 1.5

W = 1

W = 1

It can be shown that this acts in the same way as the previous perceptrons.

To demonstrate the perceptrons act in the same way the following tables are given

AND (weight on extra neuron = 1.5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input 1** | **Input 2** | **Input 3** | **Sum** | **Step(0)** |
| -1 | 0 | 0 | -1.5 | 0 |
| -1 | 0 | 1 | -0.5 | 0 |
| -1 | 1 | 0 | -0.5 | 0 |
| -1 | 1 | 1 | 0.5 | 1 |

OR (weight on extra neuron = 0.5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input 1** | **Input 2** | **Input 3** | **Sum** | **Step(0)** |
| -1 | 0 | 0 | -0.5 | 0 |
| -1 | 0 | 1 | 0 | 1 |
| -1 | 1 | 0 | 0 | 1 |
| -1 | 1 | 1 | 0.5 | 1 |

NOT (weight on extra neuron = -0.49)

|  |  |  |  |
| --- | --- | --- | --- |
| **Input 1** | **Input 2** | **Sum** | **Step(0)** |
| -1 | 0 | 0.49 | 1 |
| -1 | 1 | -0.51 | 0 |

Again, I would not expect the students to show three examples. They are just shown for completeness.

## Q Explain the concept of Back Propagation neural network

Multilayer neural networks use a most common technique from a variety of learning technique, called the back propagation algorithm. In back propagation neural network, the output values are compared with the correct answer to compute the value of some predefined error function. By various techniques the error is then fed back through the network. Using this information, the algorithms adjust the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles the network will usually converge to some state where the error of the calculation is small.

The goal of back propagation, as with most training algorithms, is to iteratively adjust the weights in the network to produce the desired output by minimizing the output error. The algorithm’s goal is to solve credit assignment problem. Back propagation is a gradient-descent approach in that it uses the minimization of first-order derivatives to find an optimal solution. The standard back propagation algorithm is given below.

**Step1:**

Build a network with the choosen number of input, hidden and output units.

**Step2:**

Initialize all the weights to low random values.

**Step3:**

Randomly, choose a single training pair.

**Step4:**

Copy the input pattern to the input layer.

**Step5:**

Cycle the network so that the activation from the inputs generates the activations in the hidden and output layers.

**Step6:**

Calculate the error derivative between the output activation and the final output.

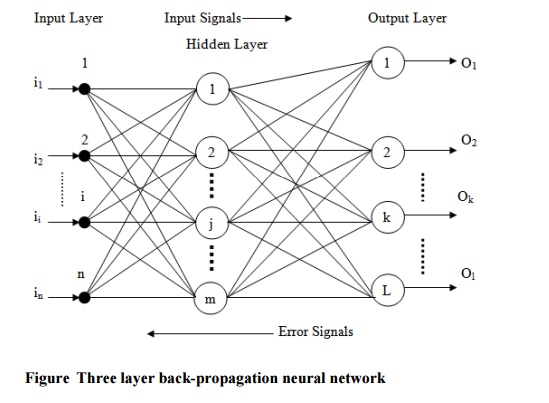
**Step7:**

Apply the method of back propagation to the summed products of the weights and errors in the output layer in order to calculate the error in the hidden units.

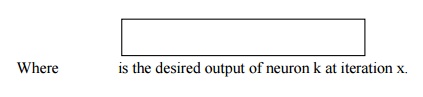
**Step8:**

Update the weights attached the each unit according to the error in that unit, the output from the unit below it and the learning parameters, until the error is sufficiently low.

To derive the back propagation algorithm, let us consider the three layer network shown in figure .



To propagate error signals, we start at the output layer and work backward to the hidden layer. The error signal at the output of neuron k at iteration x is defined as



Generally, computational learning theory is concerned with training classifiers on a limited amount of data. In the context of neural networks a simple heuristic, called early stopping often ensures that the network will generalize well to examples not in the training set. There are some problems with the back propagation algorithm like speed of convergence and the possibility of ending up in a local minimum of the error function. Today there are a variety of practical solutions that make back propagation in multilayer perceptrons the solution of choice for many machine learning tasks.

# MCQs & answers.

**1.** What is ART in neural networks?

(a) Automatic resonance theory

(b) Artificial resonance theory

(c) Adaptive resonance theory

(d) None of the above

**2.** What is an activation value?

(a) Weighted sum of inputs

(b) Threshold value

(c) Main input to neuron

(d) None of the above

**3.** What are the issues on which biological networks proves to be superior to AI networks?

(a) Robustness and fault tolerance

(b) Flexibility

(c) Collective computation

(d) All of the above

**4.** The fundamental unit of network is

(a) Brain

(b) Nucleus

(c) Neuron

(d) Axon

**5.** Activation value is associated with?

(a) Potential at synapses

(b) Cell membrane potential

(c) All of the above

(d) None of the above

6. Back propagation is a learning technique that adjusts weights in the neural network by propagating weight changes.

(a) Forward from source to sink

(b) Backward from sink to source

(c) Forward from source to hidden nodes

(d) Backward from sink to hidden nodes

7. Identify the following activation function:

φ(V) = Z + (1/ 1 + exp (– x \* V + Y) ),

Z, X, Y are parameters

(a) Step function

(b) Ramp function

(c) Sigmoid function

(d) Gaussian function

8. An artificial neuron receives n inputs x1, x2, x3............xnwith weights w1, w2, ..........wn attached to the input links. The weighted sum\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is computed to be passed on to a non-linear filter Φ called activation function to release the output.

(a) Σ wi

(b) Σ xi

(c) Σ wi + Σ xi

(d) Σ wi\* xi

9. Slots and facets are used in

(a) Semantic Networks

(b) Frames

(c) Rules

(d) All of these

**Answers**

**1. (c) 2. (a) 3. (d) 4. (c) 5. (b) 6. (b) 7. (c) 8. (d) 9. (b)**