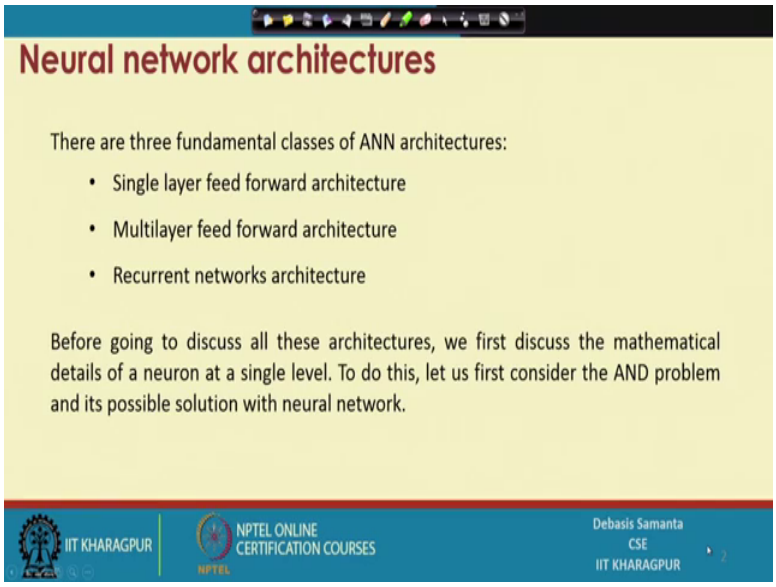
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 35
ANN Architectures

In Artificial Neural Network the basic unit is Neuron. And there are many neurons are interconnected to each other forming the network. That is why it is called the neural network. Now, today we will discuss in this lecture, what are the different architecture that can be used to build the neural network.

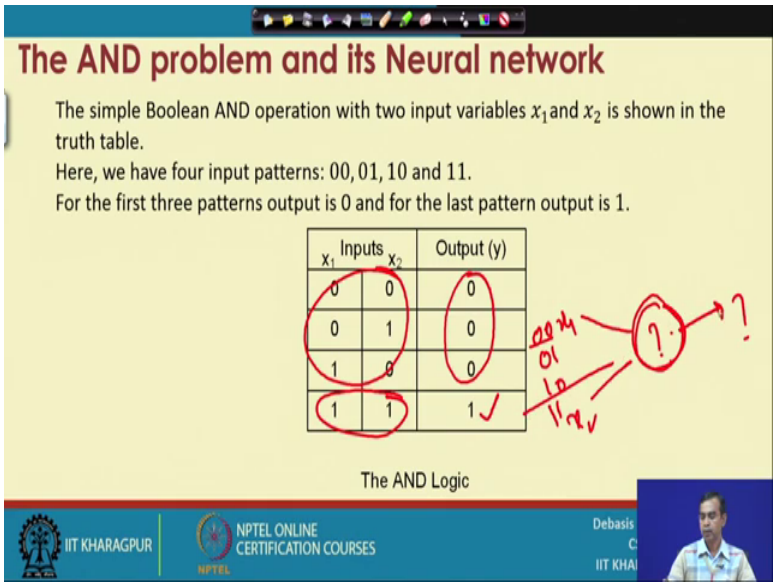
(Refer Slide Time: 00:34)



Now, all the architectures which are there in Artificial Neural Networks can be divided into broad 3 categories called the Single layer feed forward architecture, then the Multilayer feed forward architecture and finally, the Recurrent network architecture. Now, so, these are the 3 different architectures.

And then, before going to learn these architectures, we will just quickly go through the mathematical details of a neuron at a single level. So, to do these things, let us first consider a problem, we can term this is an and problem and is the one problem is basically very much popular in Boolean logic and the AND problem like this.

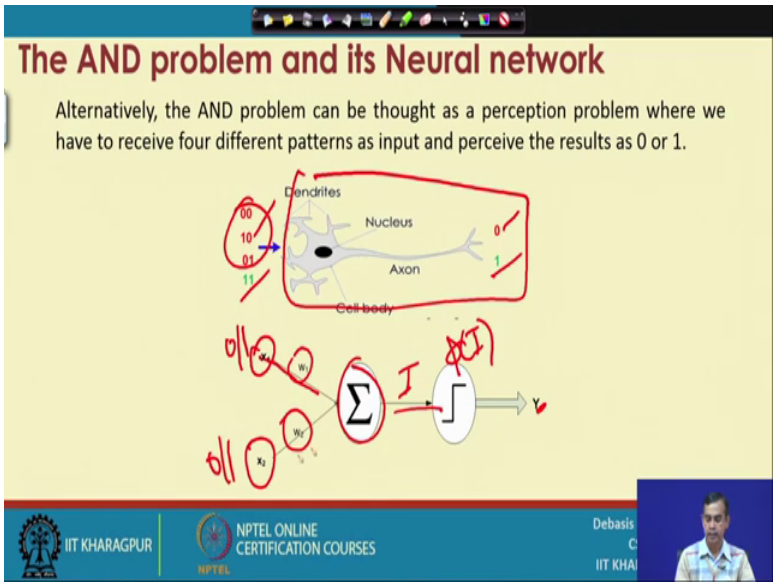
(Refer Slide Time: 01:25)



So, this is the truth table of an AND problem. Now, this AND problem can be considered as a Pattern Matching Problem or Pattern Recognition Problem. Now, let us see how it can be considered as a Pattern Recognition Problem. Now, we can consider if this is this is the neuron, that this neuron has the pattern consisting of two bits like say this one and this one so, x 1 and x 2. Now, these bits can be 0, 0 or 0, 1 or 1, 0 or 1, 1 ; that mean, x 1 can be 0 or 1. Similarly x 2 can be 0 or 1 and these are 2 bits are can be considered pattern. Now, what is the pattern recognition, that this you this system will do and it will give an output line.

So, pattern recognition problem like, so, if these are the patterns, if it is fed to this neuron, then it will give output 0. On the other hand, if this is the pattern, if it gives to this system, it will give that output 1. So, this way we can recognize the pattern whether it is 0, 0, 0, 1, 1, 0 or it is 1, 1. So, the that neural network can be right can correctly recognize this pattern either in the form of a 0 or 1. So, this is the one pattern recognition problem like and it is basically is nothing but a AND logic problem that is there in our Boolean algebra. Now, so, this AND logic and if this is a pattern, now let us see how the how the Neuron can be designed to solve this problem.

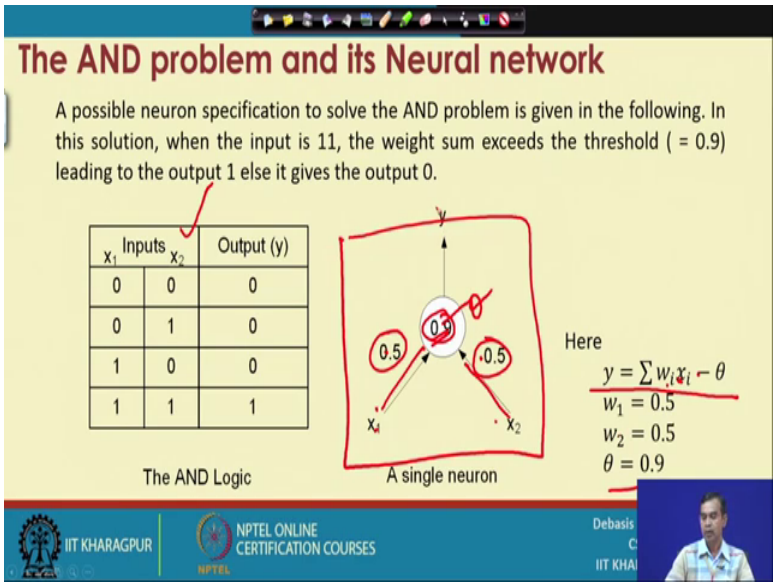
(Refer Slide Time: 03:10)



Now, so, here basically if we consider this is the biological neuron. So, these patterns whenever gives to us like see if we see this pattern, we can say 0 and if we see this pattern, we can say 1.

So, is a Pattern Recognition; Now again, so far this mimic of this neuron is concerned; that means, in the perceptron, it has 2 input x 1 and x 2 and the summation unit and; obviously, w 1 and w 2 is there and it goes there and this is the transfer function phi, who is basically take the input I and then gives the output Y. Now, we will just see exactly what are the different weights values and then for this input? This means either 0 or 1 and here also 0 and 1. It can recognize the pattern. Now, so the idea about; that means, how the neural network is there to solve this problem, it is shown here.

(Refer Slide Time: 04:04)

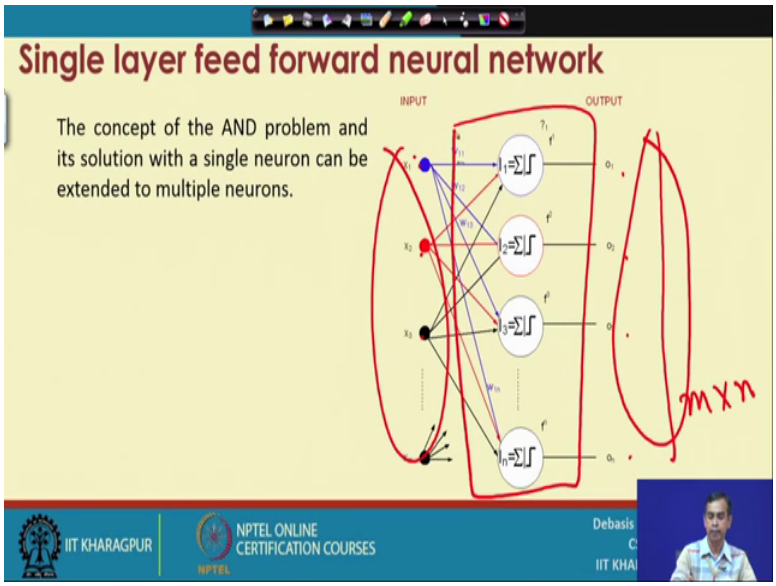


So, here we can see this is the patterns there needs to be recognized and this is the our simple neuron or we can they this neural network consists of only 1 neuron and this is the only one neuron, it takes the x 1 is the input x 2 is the input and this 0.5 and 0.5 are the 2 weights in this case. And 0.9 is basically the theta, the threshold values and then transfer function can be like this y w i x y, it is visually w 1 x 1, w 2 x 2 summation minus theta; theta is 0.9, then it will give the value either 1 or 0.

So, this is just like a idea about how this pattern recognition problem like this can be can be completed using a single neuron as it is shown here. Now, in the single neuron, so, few characteristics are important. These are the weights are there, we have to learn it. I gave you these weight values for an example, but you can ask that how these weights are calculated.

So, we will see how these weights can be calculated. Similarly, 0.91 threshold value, that we have discussed. Then, how this threshold value is known to us so, we learn about how these threshold values is known to us. Now, this is the idea about how the neural it waits for us.

(Refer Slide Time: 05:26)



Now, this is a simple one pattern matching problem; that means, 2 input and problem that we have considered. The idea can be again extended, 3 input and problem and so, on. Now, like this AND logic which is there in Boolean algebra, other logic like NAND, NOR all these things also can be implemented using this neural network. And you know, Boolean logic is basically, the basic things that is used there in VLSI circuit to develop our chips or computers processing unit computing processing unit.

Basically, it is the way it works, it is the same way it is also working here; that means, the way the (Refer Time: 06:08) size chips are basically designed. And basically same way our neural network can be designed to solve the problem. Now, so the last example that we have considered, it is on a very simple one and problem, but if it is a complex problem; that means, if it consists of many inputs and then many outputs are to be considered, then that architecture will look like this. So, here we can see, these are many input x 1, x 2, x 3 dot dot x ns are to be fed into the computing system and it will produce o 1, o 2, o 3 and o n, the n number of outputs are here.

So, it is basically m versus n one combination, that m input and n output is there. Now, so, this kind of input this kind of what is called a mapping from this input to this output can be managed by this kind of architecture. Now this is a one architecture if you see, there a number of neurons are stacked one after another. So, this is the one neuron, second neuron, third neuron and then nth neuron. So, if the number of output is n, then it

basically require n number of neurons, are to be stacked. Now, each neurons if we see. So, all the inputs that is are there is basically connected to all the neurons. So, x 1 is connected to this neuron, this neuron, this neuron, and this neuron.

The likewise, this neuron also connected to this neuron and this neuron. So, all inputs are connected to all neurons, which are there in this series. And another important thing that you can see also that, there are all the inputs are connected to this neuron by means of some weights are there. So, x 1 if this is the input to this one connected to this neuron, then it has weights w 1, 1 w 1, 2 and w 1, 3 and w 1, n. Similarly, this is the input if it is there, then w 2, 1 w 2, 2, w 2, 3 w 2, n. And for the nth neuron, so, it is w n 1, w n 2, w n 3 and w n m n. So, this way all inputs are connected to all neurons and this is basically the network of connectivity that we can check.

Now here, the input that is given to here will be feed forward to the output line. So, that is why it is called the feed forward neural network. And it is also called single layer ; because if this is the one layer of neurons, so, that is why it is called the single layer feed forward neural network. And in this network, so, there are many weights. So, basically the weights that is there it is like this. So, all weights are there. So, they are maybe; so, weights is basically m cross n, a number of weights basically involved in this network and this is called single layer and then feed forward because of is the one layer. And then this input is connected to one layer only. And one important thing that I want to mention here again; so, that in each perceptron so; this is the perceptron 1 and this is a perceptron 2 and so on.

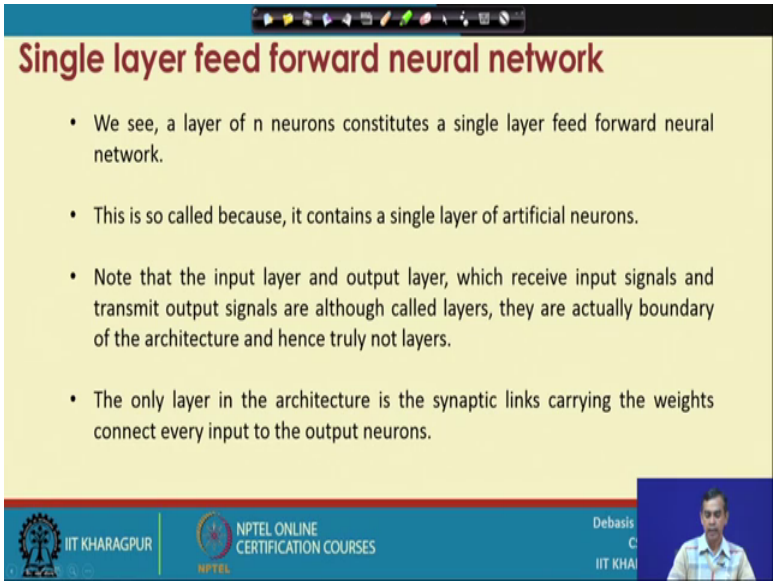
In each perceptron, there are what is called the thresholding function. So, it may be the different thresholding function or transfer function or may be all perceptron having the same transfer function. So, if it contains the different transfer function, then the learning that is required is very difficult. But if it contains only one transfer function, learning will be simple and straightforward. Further, in each transfer function there is a threshold in value. So, that threshold events also varies from one neuron or one perceptron to another perceptron. If we each perceptron contents the different thresholding value, then again also learning will be there and that learning will take much time.

So here, so further, Neural network is concerned and more precisely the single layer feed forward neural network is concerned, all the weights that is there to this layer or the

parameters to be learned. And all the transfer functions that is there and with that thersholding values are to be learned. If we learn all and also number of what is called the perceptrons in the layer also one factors to be learned. So, if we learn for a given problem all these parameters, then we can say the our neural network is trained perfectly. And once the neural network is trained then if we give any input to this it will produce the corresponding output. So, here only the matter of how a network can be trained or it can learn from the input to their output and then once it is build a network, we can use for solving the our problem.

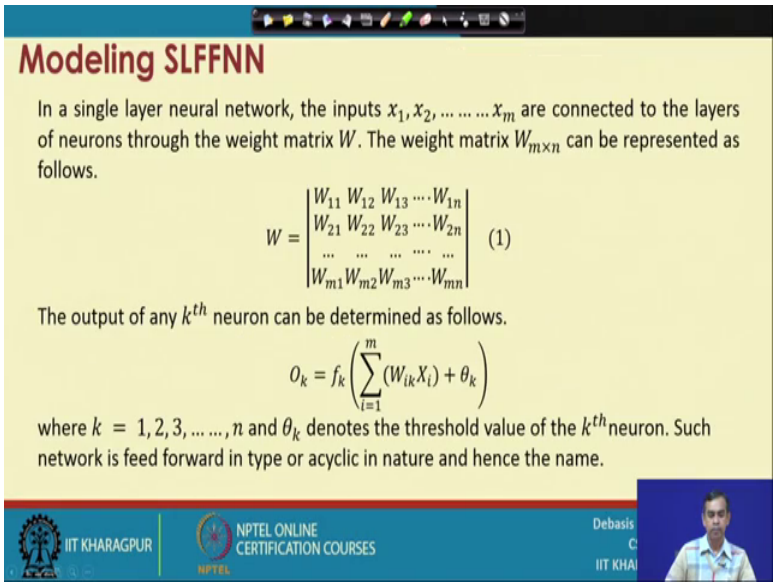
So, this is the concept that we follow in Artificial Neural Network. Now, we have started with first perceptron and we check that how and problem can solve it. Perceptron is the very simple most on problem solving neural network. After that, just now we have learned about Single layer feed forward neural network.

(Refer Slide Time: 11:33)



Next, we learn about little bit different and complex whether neural network architecture, which is called the multi multiple Multilayer feed forward neural net.

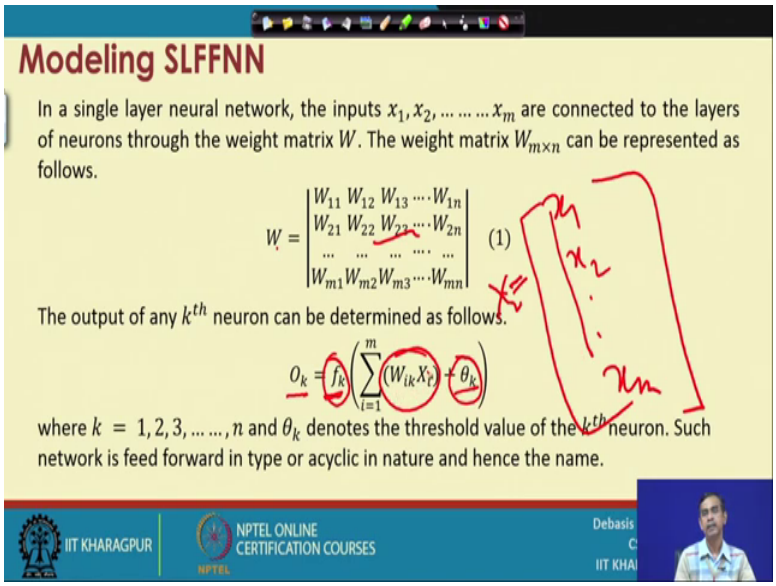
(Refer Slide Time: 11:37)



Now, again I just want to mention that I forgot to mention it there. So, in this particular neural network single layer feed forward network, the weight matrix W is there. It is basically the collection of all weights from any input to any neuron, any perceptron. So, these are the all weight values from the first input x 1 to all the neurons 1, 2, 3 to the n neurons. And see, this is the input from the nth weighted by these values are there.

Now, all these things is basically can be stored by means of a m cross n matrix. So, it is called the weight matrix. Now, like this weight matrix, this is the transfer function that also needs to be learn and this is the threshold value that needs to be learn and this is for the kth perceptron and it basically for the kth perceptron these are the kth transfer function with the threshold value.

(Refer Slide Time: 12:39)

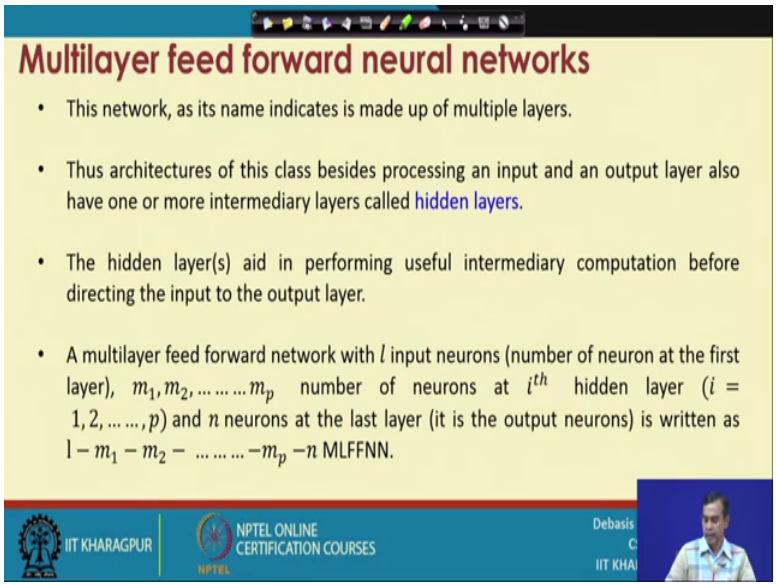


And given the input x 1, x 2 dot dot x n and that we can consider as a matrix X. So, any ith input if it is given there. So, it is basically, W dot x is basically, the matrix product of the 2 weights and then this one, then it will gives the ith what is called the summation unit to the ith perceptron. So, this way it can solve and as you know the matrix operation is the one simple most and very fast operation.

So, computing in a neural network is very fast and then not a time timing issues are there ok. So, this is the idea about Single layer feed forward neural network and. So, for modelling such a network is basically model this is the mathematical form or the mathematical model that can be considered or that that is used to that is used to solve problem.

So, this is basically model of a Single layer feed forward neural network. And then, model can be expressed in a mathematically in terms of matrix and simple computation as it is here. So, so modelling is there and then. So, basically we have to build this model, means we have to find this is the W value and this is the value of this function. So, modelled means basically, we have to learn these W matrix and f k values, f k functions for each neutron.

(Refer Slide Time: 14:09)



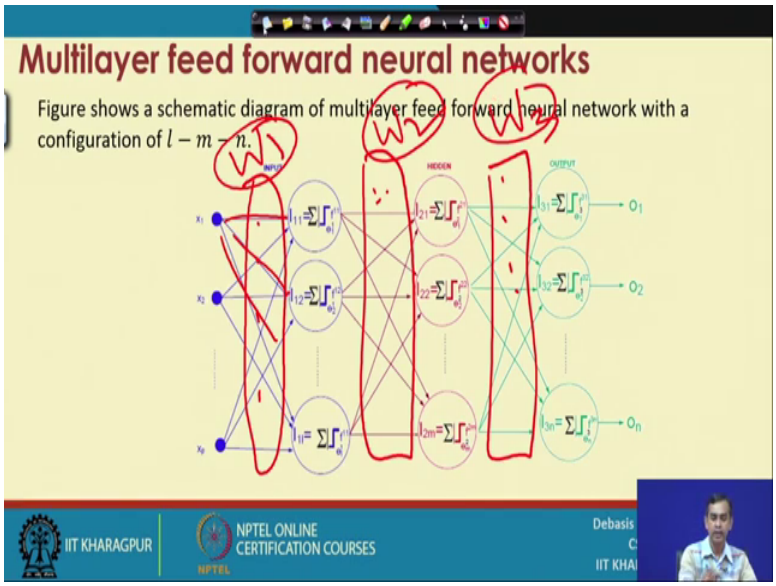
Now, so, this is the idea about Single layer feed forward neural network and we can extend the same concept to a little bit complex neural network. Now, in case of Single layer feed forward neural networks, only one layer of perceptrons are there.

On the other hand, in case of Multilayer feed forward neural network, instead of one there are many; out of which one layer which is connected to input data, it is called the input layer. And there is another layer which is connected to output is called the output layer. And in between input and output layer, there are some layers of perceptrons is called the hidden layers.

So, if l number of neurons in the input layer ; that means, there are l number of inputs. If n numbers of outputs are there, then in the output layer, n number of perceptron should be there. And therefore, so, it is called the l and n combination. So, for the input and output is concerned and in between this input and output layer, they are may be m number of hidden layer say m 1, m 2 dot dot m m n. So, these are the hidden layers. Then number of neurons in these hidden layers again can be divided by that number.

Say in m 1, m 1 number of neuron, in m 2 layer m 2 number of neuron and so on, so on. So, a typical look of such a multi layer neural network can be shown here.

(Refer Slide Time: 15:34)



We say here in this figure, I show one simple layer, a simple Multi layer feed forward neural network or 3 layers are there. So, this is the input layer, this is the output layer and this is the only one layer in the hidden layer. Instead of only one, are there may be like this many layers also can be considered. So, it will just increase the complexity of the network, that is all. Now, all the inputs are connected to this input layer and all the outputs are connected to be input layer. Here we can see, p number of inputs and n number of outputs and this network is called the Multilayer network because multiple layers of neurons are there.

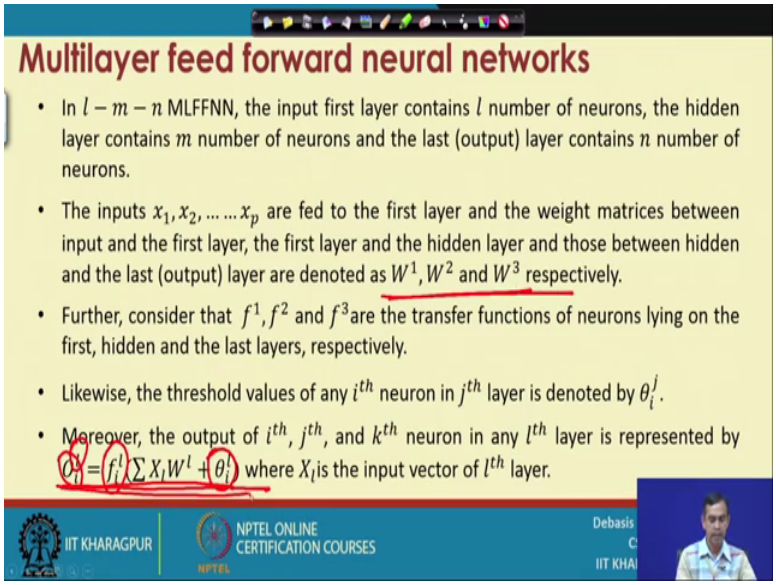
And it is also feed forward neural because input pass to this. It produced the output; whatever the output produced by a particular perceptron, gives the input to all other perceptrons into the next layer. Then this output also take this input from the different perception in the previous layers and produced the output gives to the this one. So, it is the same thing. So, it is basically in each every layer, we can say it is a single layer line. So, there is a what is called the stack of a number of single layers.

All are highly connected to each other and then it form the Multilayer feed forward neural networks. Like the modelling of Single layer feed forward networks, the Multilayer feed forward networks also can be modelled in same way; but here, more what weight matrix. more transfer functions and the different thresholding values are there.

For example, so, there will be one weight matrix is required to define all the weights here, another weight matrix is required to denote here, another weight matrix is required to denote here ; because all the inputs are passed through the different perceptron by via via the different weighting values there. So, here you can say the weight matrix w 1, here w 2 and w 3. So, in order to model this one, we have to know what is the weight matrix, what is the weight matrix at the different level. So, these are the modelling issues and then for each perceptron in each layer, the transfer function. So, it is f 1 1, f 1 2, f 1, f p 1 like this one. Similarly, here also this kind of transfer functions are there.

And in each transfer function, thresholding values also to be considered. So, in order to model such a neural network, it basically modelling these are the weights and the difference thresholding transfer function and then threshold values in each perceptron. Now, so all these things can be done again in a simple, in a compact and mathematical form by means of different weight matrix and then different what is called the functions that it can conserve. For example, this is the weight function that can be generalized to represent each perceptron in each layer.

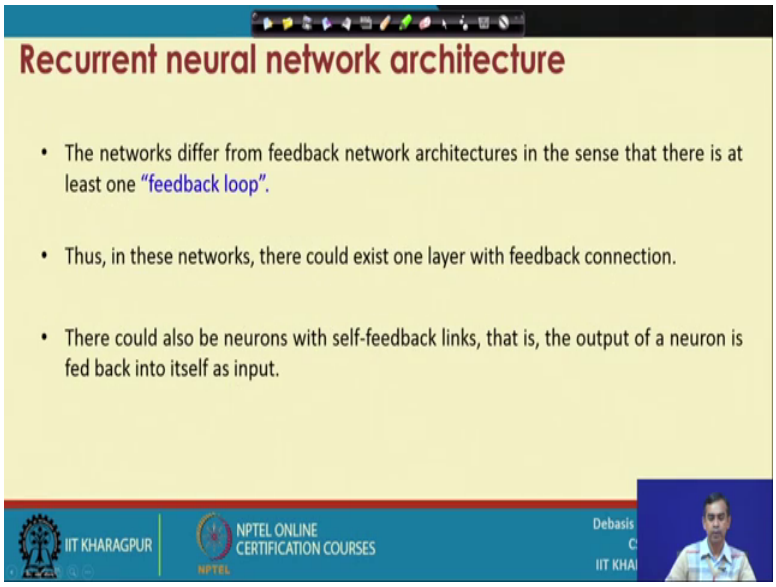
(Refer Slide Time: 18:17)



So, it is basically the output of the ith perceptron lth layer and it is defined by the f i function, the transfer function and this is the threshold function. So, this is from this is the unique values unique functions in each perceptron that is there in addition to these are the different weights that needs to be considered there.

So, modelling just like a Single layer feed forward network, if it was modelled using only one weighting matrix, it will be modelled using 3 different weighting matrix. If it is modelled by a series of transfer function and threshold value, but it will be a large correction of transfer functions and thresholding values. So, modelling only will be little bit complex than the single layer forerunner in case of Multilayer feed forward neural network.

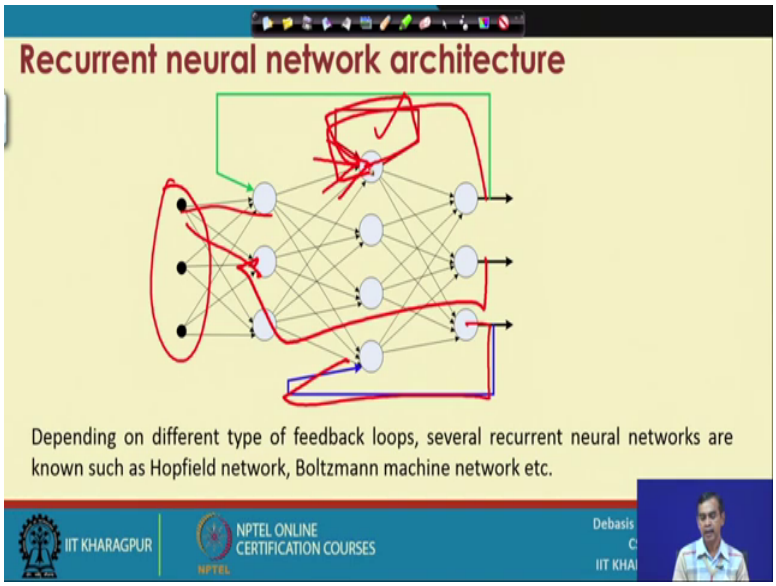
(Refer Slide Time: 19:23)



So, this is the idea about Multilayer feed forward neural network. Next type of neural network that it is called Recurrent neural network architecture. Now, the difference of this network architecture compared to the previous two architecture is that the feedback will be there.

That means, there will be a loop. So, it is called the feedback loop. So, there may be at least one feedback, a feedback loop from the next layer to the previous layer. So, if feedback is there, then it is called the Recurrent neural network. And now, so, let us see how the such a network looks like.

(Refer Slide Time: 20:05)

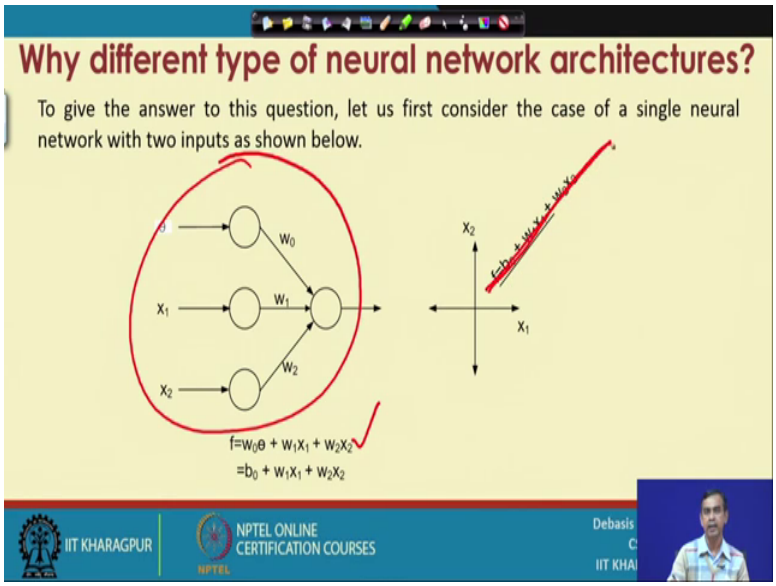


Here is the one picture here. So, here basically if we see, the output of this is feed back to this one. The output can be feedback to this one also. And here also output to the same neural this is called a self loop and this is the previous loop that ; that means, it is more complex because in addition to the conventional input there which will be connected plus the output from any near to the previous perceptron is there.

So, number of what is called the output to each will number of inputs to each perceptron will be enormously high. And it does leads to a very complex network architecture called the Recurrent neural architecture. So, if there is a self feedback or recurrent things then they are call the Hop field neural network, Boltzmann machine network like this on. So, different networks are there. Whether there will be a self loop or not, the loop from the next layer to just previous layer not or the loop from one perceptron in a layer to any perceptron in any other layer or not.

So, this way the different architectures can be thought and then it is there. Now again, so, for the modelling is concerned the same concept, it can be applied here and then same modelling will be there. Only the thing is that, all the matrix that is there, they will have very larger size compared to the simple feed forward neural network.

(Refer Slide Time: 21:39)



Now, so, this is the Recurrent neural network architecture and there is the; obviously, the question is that, which network architecture is suitable to which application ? Now, we quickly gone through this concept there [noise, in which case the Single layer network is required in which case the Multilayer network is required or in which case the Recurrent neural network is there. Now here, let us consider this figure. In this figure we can see so, this is the input layer and this is the one processing on output layer. So, neuron is there. So, it is just like a Multilayer feed forward network sort of thing or we can say this input is directly come to here this also we can consider then this layer is not there.

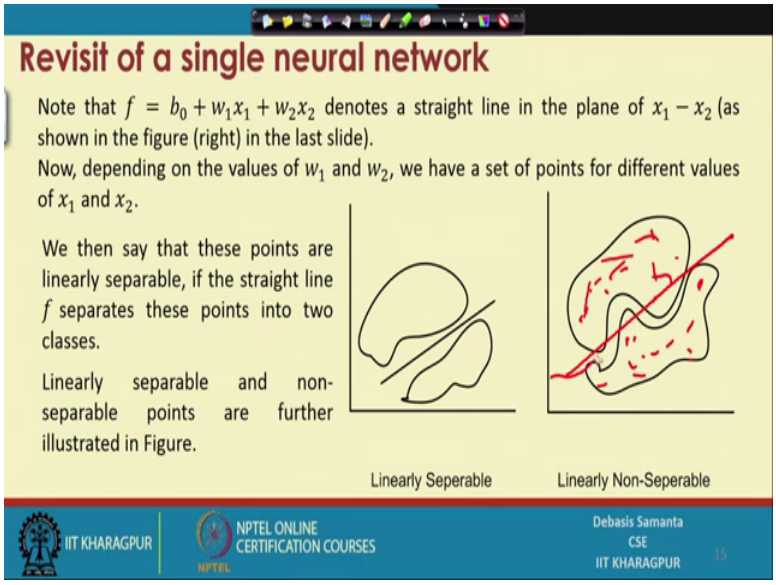
So, it is basically a Single layer feed forward neural networks are there. We will consider 3 o s; w 0, w 1, w 2 and this theta is called the bias input sometimes use there. Mainly there are 2 input; x 1 and x 2 and this can be x 0 also can be written like there. Anyway, so, if this is the neural network, then we can say the transfer function will be look like this. So, basically the transfer function basically summation of the I input into their weights plus this is the threshold value.

So, this is basically nothing but the threshold function it is here. Now, so, this basically f will return depending on the different values of theta and weights it will return one output. Now, this basically if we see, this is this is a an expression of a straight line in a 2 dimensional phase x 1 and x 2, if we consider the 2 dimensions there.

So, it is a 2 dimensional data space, x 1 and x 2 and for any input values having x 1 and x 2, it basically decide either this input is in these sides or in these sides so, this basically way. It basically classify the data it belongs to these sides or not these sides.

So, it is also a classification or is a we can say prediction concept like. And this kind of prediction concept we can see how it can be solved using this Single layer feed forward neural network. And in this case, if this kind of expression is possible, then this is basically straight line. So, here the predictor or the classifier looks like a linear and straight line is there. Now, so, this is a linear classification or linear prediction is there.

(Refer Slide Time: 24:09)

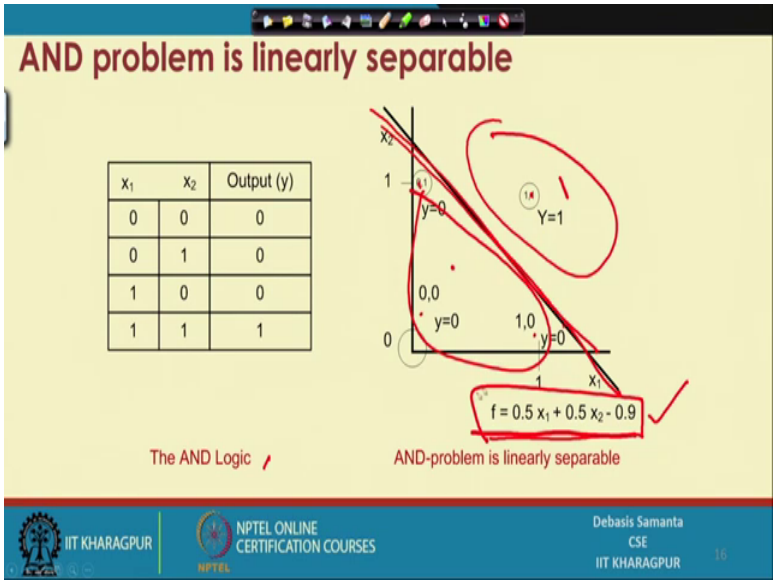


Now, if these kind of predictions are there, then we can implement these kind of things using a Single layer forward neural network. But there are some classifications are there, which where the data cannot be linearly separable. Now, exactly I can discuss about what is the linearly separable data is there? Now, so, suppose these are the data and these are the another type of data, so, 2 patterns; one pattern is this one and another pattern is this one. Then we can think about a linear line to separate all the patterns into 2 parts. So, this is the one pattern type, another pattern type. So then, we can say that these data is linearly separable, but like here if this is the patterns like this and these are 2 different patterns, then no linear line can be thought of.

So, it is basically this is a very difficult to pattern by a linear line. So then, we can say that these are the data are linearly not separable. So, there are 2 types of data. So, for the

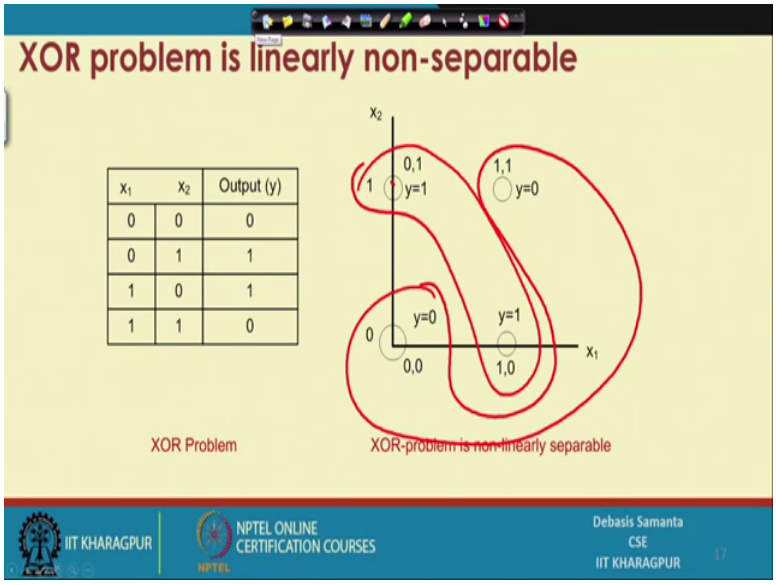
prediction or pattern recognition is concerned data is very linear; that means, it can be separate linearly and something is cannot be separate non-linearly. So, if the data is linearly separable, then we can use simple network the Single layer feed forward network. But if the data are not linearly recognizable or separable, then we should consider the network which is other than Single layer feed forward network ; that means, Multilayer or Recurrent near like this one. Now, so, there that so, these are the basically rational or genesis that in which situation, we should follow which kind of neural network architecture.

(Refer Slide Time: 25:42)



Now, here is an example that, I want to give it. If the AND logic that is the case, we have discussed about. So, this is the AND logic has that pattern. These are the different input and this is the another output. So, all the input can be patterned as a 0 and all the input in this size can be pattern at the 1. So, 2 patterns and there is a straight line like this one, which is the equation of this form that can be used to separate this one. So, this is the neural network and implementation of this neural network by means of a this kind of lines and then these data are linearly separable data and then a Single layer feed forward network can be used to train to develop this model.

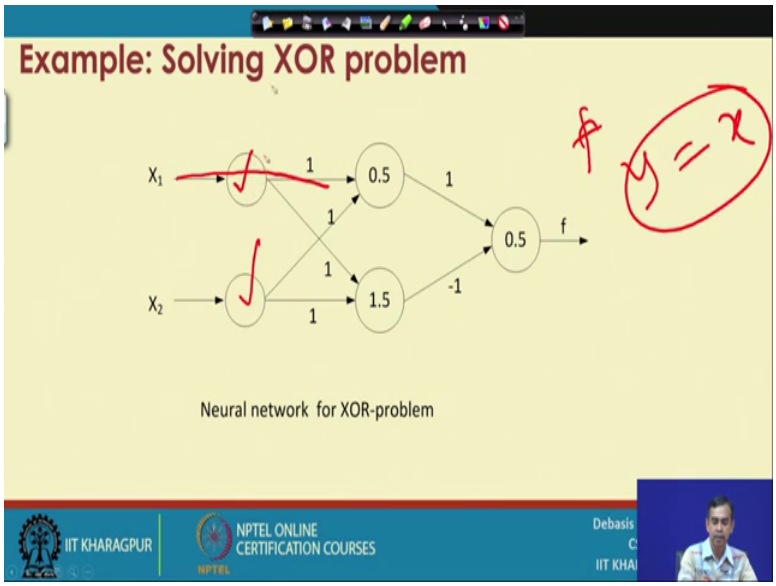
(Refer Slide Time: 26:25)



On the other hand, let us consider another problem. It is called the XOR problem. So, XOR problem has the pattern like this wherever the output will be like this. Now if it is like this, so, different patterns it is there. So, here basically, so, these are the one pattern and these are the another pattern. So, these are 2 patterns. Now, we can see that if this is the 2 patterns given to us, the data cannot be linearly separable, that data cannot be c linearly say means, we need something else then the single layer feeder. So, in this case, we can follow the Multilayer feed forward neural network. And let us see how the Multi layer feed forward networks can be planned to design this kind of architecture. So, their problem is like this.

So, it is linearly non-separable.

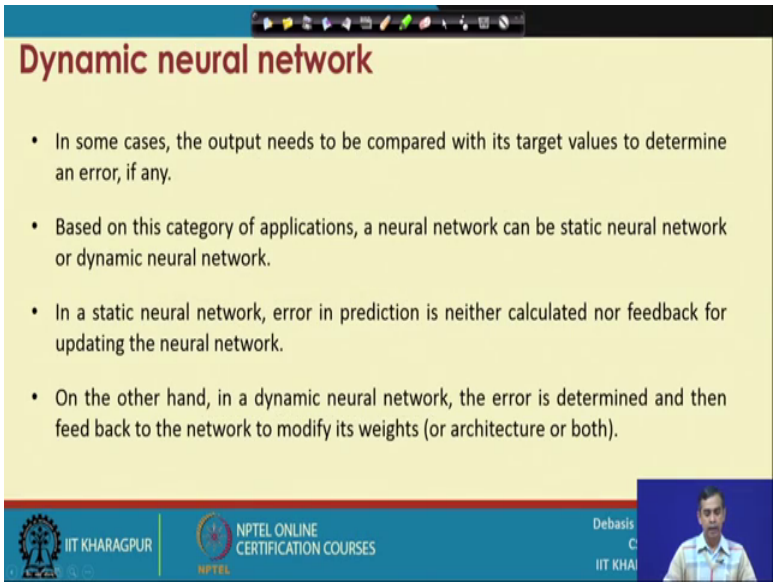
(Refer Slide Time: 27:12)



And here is architecture of the XOR problem. So, this is a one layer, the hidden layer and the output layer; input layer, hidden layer and output layer. So, it is a 3 layer. In this 3 layer 2 inputs are there and 2 perceptrons and in case of hidden layer, again 2 perceptron, in case of output layer, 1 perceptron. Now, here the different weights, here the different weights. Here, we can say the weights are this one and here the different weights we can consider. So, there are 3 weight matrix are there and here the 0 or 1 threshold value you can consider. The simple linear transfer function it has directly pass it ; that means, y y equals to x.

This kind of transfer function it is here. So, we have written blank. It is this is y. Now here, the different transfer function is to be followed. We will come to this and then the threshold value that can be considered in each unit, it is shown here in the hidden layers and also through some value that can be considered here, it can be shown here. So, this basically the architectures which basically takes any input pattern and then what is an output occurring the XOR logic and this kind of things is possible only. So, XOR logic cannot be designed using Single layer feed forward network. In fact, no one can design it. So, it can be designed only using this kind of simple Multilayer feed forward neural network.

(Refer Slide Time: 28:32)

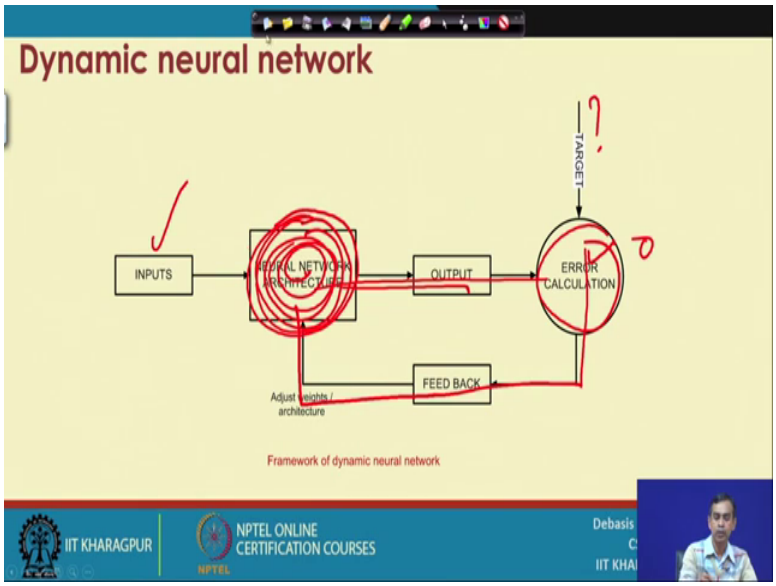


Now, so, these are the simple network that we have considered. Other than this network the Single layer, Multilayer, there is another one network also known in the theory of Artificial Neural Network is called the Dynamic neural network. Now, Dynamic neural network come into the way that if at any instant, if we decide that this is the output and then we can calculate it is error; that means, output should be x and it is coming at x dash then the error is x dash minus x.

Now, if we it this error to configure our neural network automatically, then it is called a Dynamic neural network. Now, so, there is a Static neural network versus Dynamic neural network. In case of Static neural network, no error computation takes place. On the other hand, in case of Dynamic neural network, error needs to be calculated at every instance and based on the error. That we have obtained the network parameter can be adjusted, then it leads to another neuron into or it is called the Dynamic neural network.

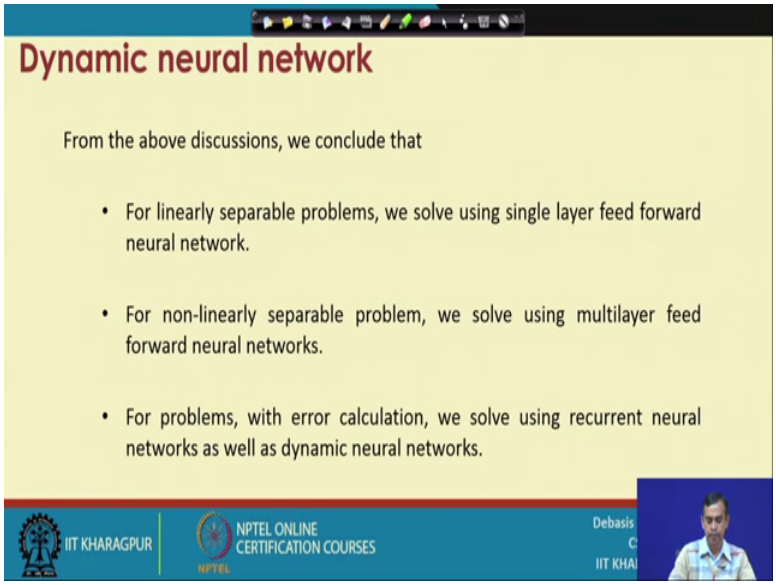
So, you can understand that it is not a; obviously, the feedback will be there. It is just Recurring neural network type, but error needs to be considered as the feed book feedback in each neuron. So, it is the concept and it is; obviously, too much complex. So, for the network architecture is concerned and so, further computation is concerned because in every instances, we have to calculate the error and if error can be propagated back to the previous layer neurons and then that neurons then adjust it it is weights or parameters values, so, that the error can be minimized.

(Refer Slide Time: 29:57)



So, this is the pictorial description, if the input is given to these and this is the Dynamic neural network architecture. Output will give there and there is a error calculation unit which basically recalculate the error and give a feedback to this one. And using this feedback, this neural network will automatically update or dynamically update so that, the error can be minimized or the 0 error or the target output can be precisely obtained. So, basically this a automatic adjusting this neural architecture will takes place and then neural net will give the best output and in that case it is very fault tolerant or robust system can be developed. So, this is the concept of Dynamic neural network.

(Refer Slide Time: 30:55)



Now, so, as a summary you can see that, for linearly separable problems which we can designed the Neural Network Architecture in the form of a Single layer feed forward neural network.

On the other hand, for non-linearly separable problems, we can use either Multilayer feed forward neural network or the higher configuration likes in a Recurrent neural network or Dynamic neural network. And Dynamic neural network in particular can be used for error calculation, we solve and then Recurring neural network and the Dynamic neural network, we can use for the um if we want to take into account the errors in the computation. So, these are the different architecture that we have learnt and next our idea next our task is basically how to model the different network architecture.

So, modelling the different networks teacher means, how to learn the different parameters with which a neural network can be composed of. So, that will be discussed in the next lectures.

Thank you.