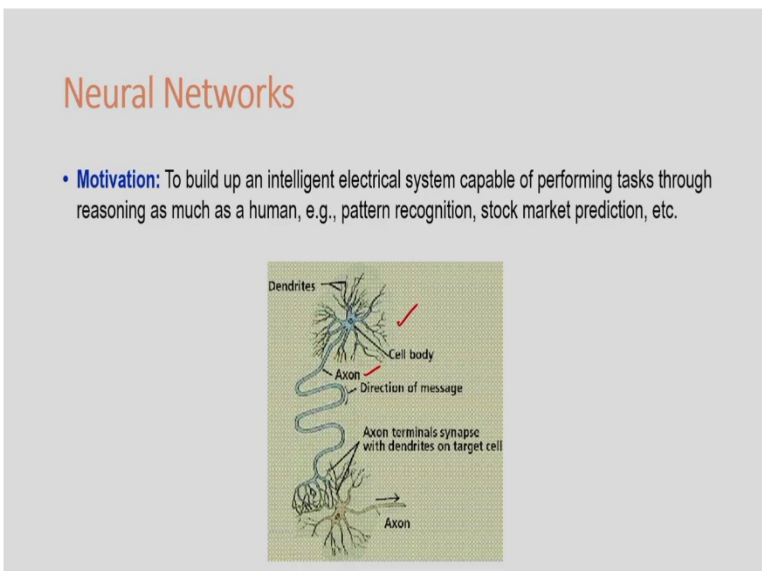
Computer Vision and Image Processing-Fundamentals and Applications
Professor Doctor M.K Bhuyan
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati, India
Lecture 35
Artificial Neural Network for Pattern Classification

Welcome to NPTEL MOOC course on Computer Vision and Image Processing- Fundamentals and Applications. In this week I will be discussing the concept of artificial neural networks. How it can be use for pattern classification. There are two types of artificial neural networks one is supervised artificial neural network and another one is unsupervised artificial neural networks. After this I will discuss the concept of deep networks. And how it can use for object classification, object recognition, image recognition that concept I will be discussing.

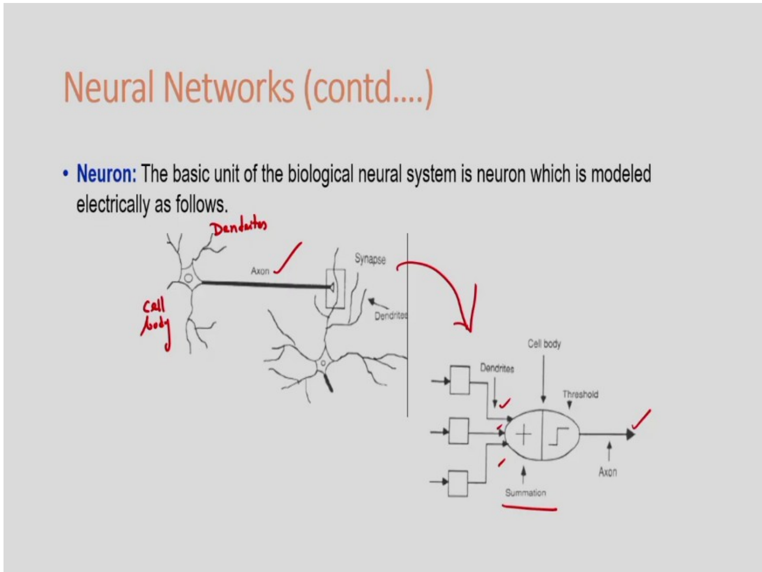
So, what is artificial neural network? So, artificial neural network the concept is very much similar to biological nervous system. So, that is why I can say it is a biologically inspired system. Now, let us see the concept of the artificial neural network.

(Refer Slide Time: 1:30)



In this figure you can see I am showing one biological nervous system you can see cell body, dendrites, axon. So, that concept I will be explaining in my next slide. So, idea is to deal an electrical system which is capable of performing tasks through reasoning just like human being. So, this is the biological nervous system. Now, you can see how it is related to the electrical system.

(Refer Slide Time: 2:02)

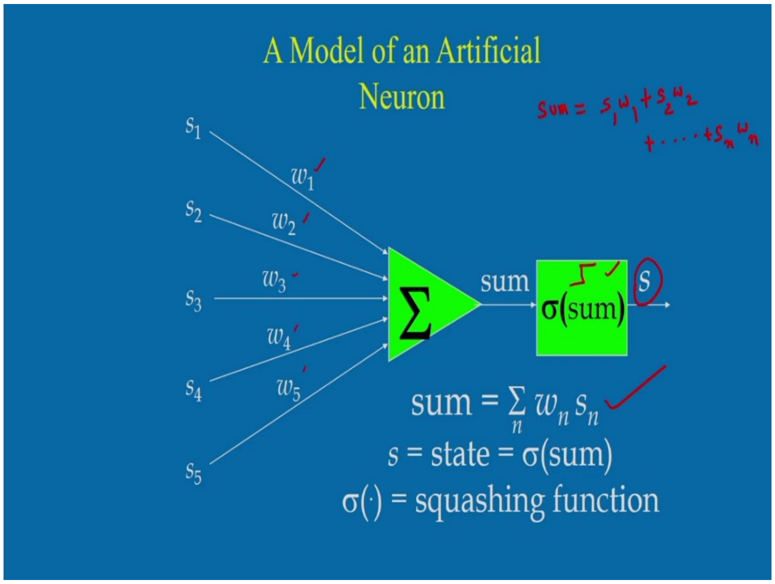


So, here you can see the concept of the biological nervous system, the biological neural system and here you can see the axon and the dendrites. So, if you see if I consider these are the dendrites. So, the dendrites collects signal from the cell body. So, suppose this is the cell body. So, the dendrites collects the signal from the cell body and all the signals are sum up if the signal value the sum value is greater than a particular threshold.

Then the signal will be transmitted via axon. So, you can see this axon and axon terminates synapse. So, this is the concept of the biological neural system. These dendrites collect signal from the cell body. And all the signals are sum up if the signal value is greater than a particular threshold then the signal will be transmitted via axon and the axon terminates synapse. So, corresponding to this biological neural system.

You can see what is actually going in the cell body that concept I am going to explain. So, that means dendrites collects the signal from the cell body. And all the signals are sum up here, you can see all the signals are collected. If the signal value is greater than a particular threshold then the signal will be transmitted via axon. So, this is the concept of the biological neural system. So, I am repeating these so dendrites collect the information from the neurons. And all the signals are sum up and if the sum value is greater than a particular threshold then I will be getting a signal in the axon. So, that is concept of the biological neural system.

(Refer Slide Time: 4:03)

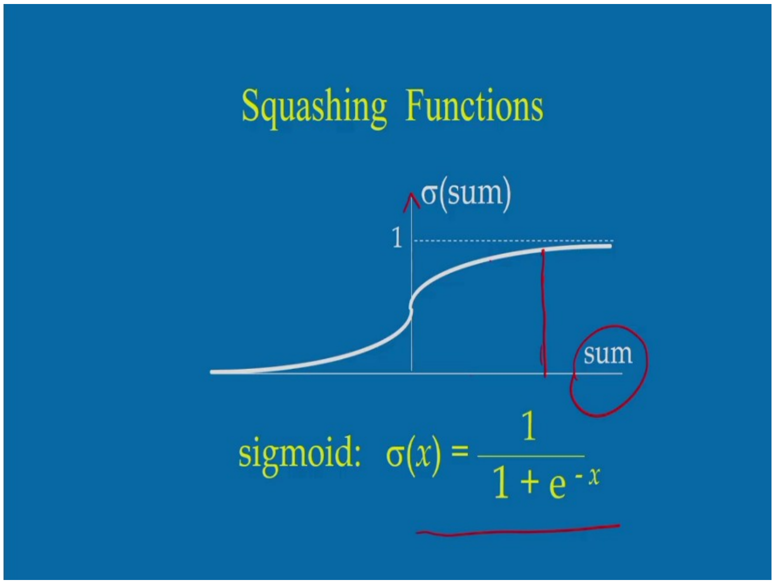


And if I consider the equivalent, the elliptical system suppose that is nothing but the artificial neural network. So, corresponding to that artificial neuron you can see here I have the signals the input signals are S1, S2, S3, S4, S5. And I am considering the Weights, the Weights are W1, W2, W3, W4, W5 these are the Weights. Now, in this case what I am doing I am just doing the summation of the signals the input signal that means I am adding all the signals

The signals are added like this S1W1 a signal is multiplied with the weight; the weight is W1. Another signal is S2 that is multiplied with W2 like this I will be multiplying and suppose if I consider Sn and Wn so like this I am determining the sum. So, sum I am determining like this, so this sum value is determine like this. and after this I am considering one thresholding function. So, this is the thresholding function I am considering one thresholding function.

And this function is called as squashing function. And this is used to compare the sum value with the threshold. A threshold is the squashing function. If the sum value is greater than a particular threshold then this threshold then I will be getting the signal in the output. So, I will be getting the signal S in the output that is the state. So, I am repeating this if the sum value is greater than a particular threshold then I will be the signal S in the output. So, this is the model of an artificial neuron.

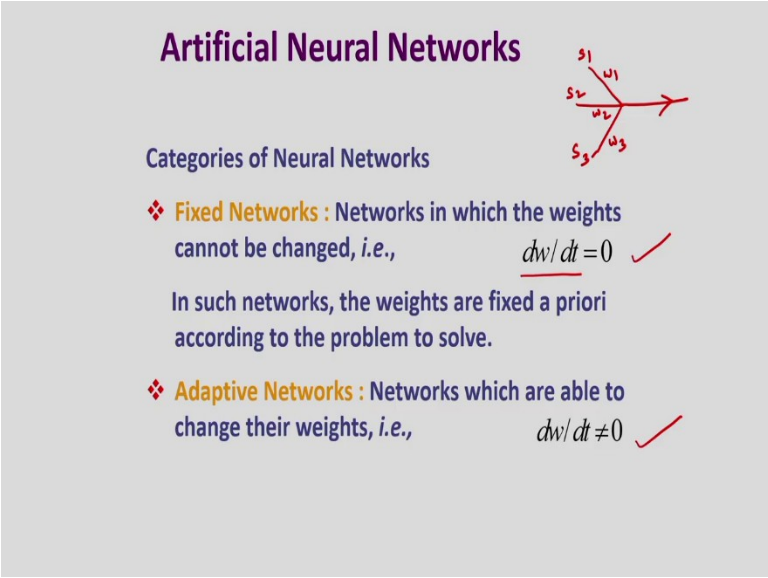
(Refer Slide Time: 5:56)



So, this the squashing function, this squashing function is the sigmoid function as generally use in the artificial neural network. So, $\sigma(x)=\frac{1}{1+e^{-x}}$. So, in the x axis you can see this is the sum I am considering and in the y axis I am considering the sigmoid function. So, that is sigma x, so sum value is compared with the sigmoid function and if the sum is greater than sum value then I will be getting 1 in the output.

So, suppose corresponding to this one the sum value is greater than this particular threshold then I will be getting one in the output. That means I will be getting the signal in the output terminal of the artificial neural network. So, that is the function of the squashing function and here I am considering the sigmoid function.

(Refer Slide Time: 6:50)

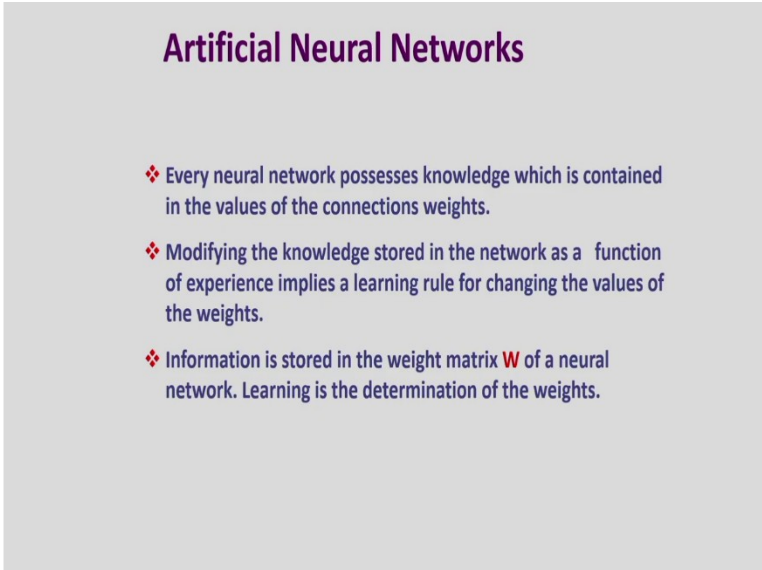


In artificial neural networks already I have shown you if the Weights, the Weights if you see W1, so W1 is the weights and another signal is suppose S2 and W2. So, W1, W2 another signal I am considering S3 W3 is the weight. And this is the structure of the artificial neural network. So, W1, W2, W3 these are the Weights of the artificial neural network. So, I have I may have fix networks in case of the fix network the weights cannot be changed.

So, that means the $\frac{dw}{dt}$= 0. So, I cannot changed the weights of the network. But in case of the adaptive network, I can changed the weights that means the $\frac{dw}{dt}$is not equal to 0 that is the adaptive networks. So, mainly I will be considering adaptive networks because the knowledge of the artificial neural network is stored in the weights. In case of the fix network the weights are fixed I cannot change the weights.

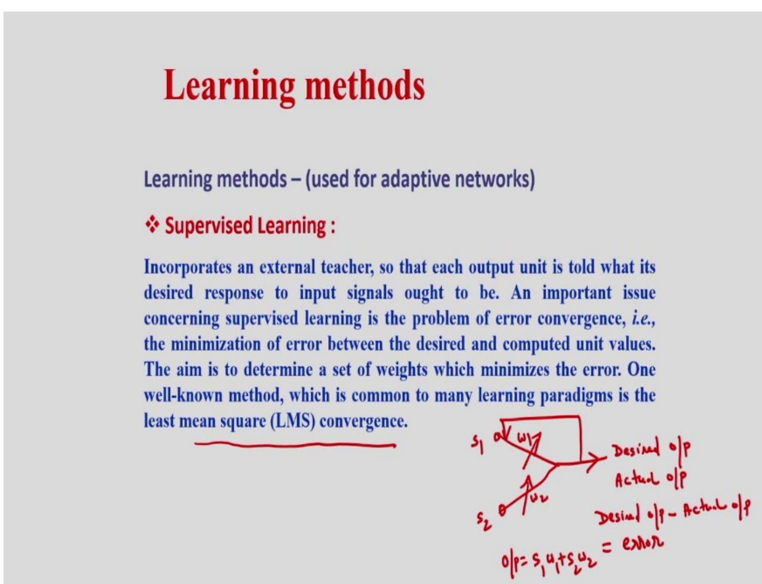
But in case of the adaptive networks, I can change the weights and that is nothing but the training of the artificial neural network. Because the knowledge of the artificial neural network is available in the weights. So, I have to change the weights based on the training samples. And this is called the learning of the artificial neural network.

(Refer Slide Time: 8:26)



So, every neural network has knowledge which is contained in the values of the connected weights that I have already explain that term knowledge of the artificial neural network is available in the weights. And that means I can modify the weights and, in this case, the modifying the knowledge stored in the network as a function of experience implies a learning rule. So, that means what is the training of the artificial neural network? That is nothing but the changing the weights corresponding to the training samples. So, information is stored in the weight matrix of the artificial neural network.

(Refer Slide Time: 9:10)



Now in case of the artificial neural networks we have to consider supervised learning and the unsupervised learning. So, what is the supervised learning? In a particular artificial neural network we know the desired output that information we have. So, corresponding to particular training sample we know what is the desired output. And also, we can calculate the actual output. The difference between the desired output and the actual output is called the error.

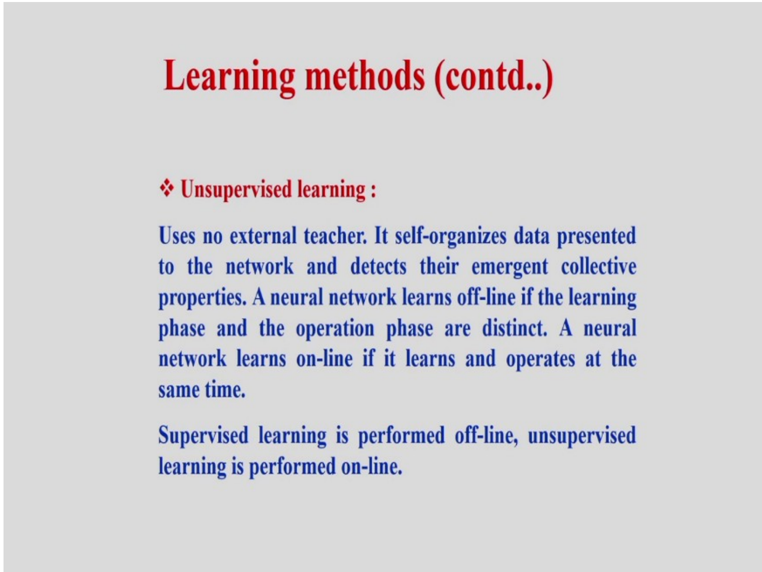
And the error is back propagated to the input so that we can changed the weights. And the objective is to minimize the error. The error is the desired output minus actual output. So, I can give one example suppose if I consider this is the artificial neural network and these are the weights W1, W2 and I am considering the input S1, S2. So, in this case corresponding to the input training samples I know what is the desired output.

That information is available so that is why it is called the supervised learning. So, corresponding to a particular class the training samples are available. And corresponding to this training samples I know what is my desired output that information I know. And from the input I can calculate the actual output. So, that also I can determine the actual output I can determine. And the difference between the desired output and the actual output is nothing but the error.

So, the objective is to minimize the error. So, that means what I can do to minimize the error. So, for this I have to adjust the weights, so I can adjust the weight so that I can reduce the error. So, because the signal output is if I consider a signal output what is the output, output is nothing but $S1W1+S2W2$. So, that means I can change the weight so that I can reduce the error. So, that is the concept of the supervised learning.

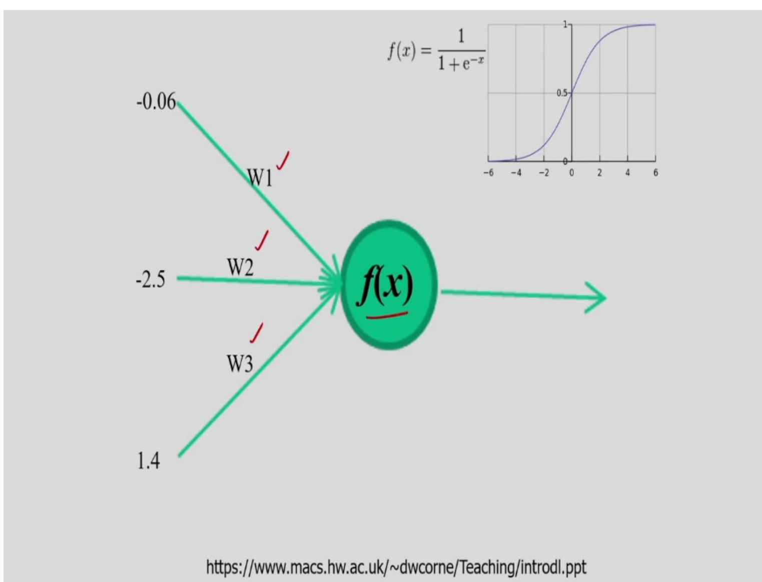
The error is back propagated to the input so that I can change the weights of the artificial neural networks. And that is called the supervised learning of the artificial neural network. For this we can use the concept like the least mean square convergence. Because I have to change the weights to minimize the error.

(Refer Slide Time: 11:50)



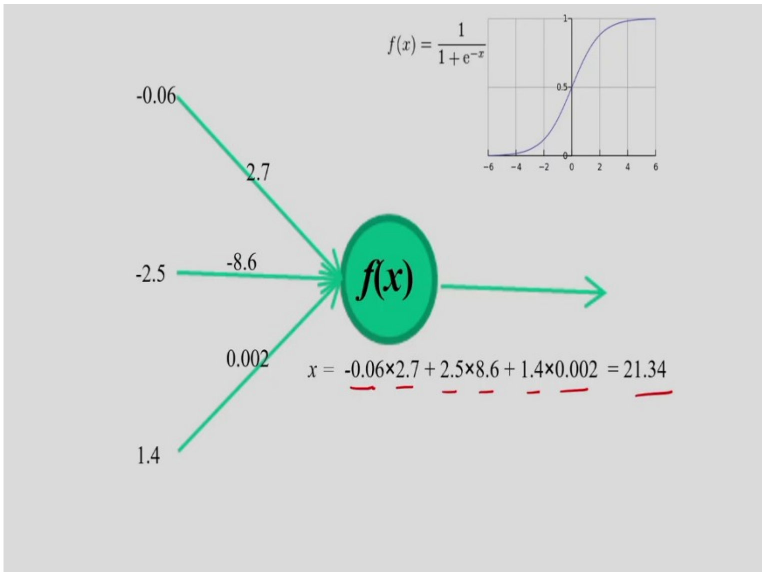
And in case of the unsupervised learning what actually it is nothing but the clustering the grouping of the training samples. So, in case of the machine learning algorithms I explain the concept of the K-mean clustering. Similar, to the k-mean clustering in case of the unsupervised artificial neural networks I can do clustering. I can do the grouping of the training samples. So, that is the unsupervised learning.

(Refer Slide Time: 12:22)



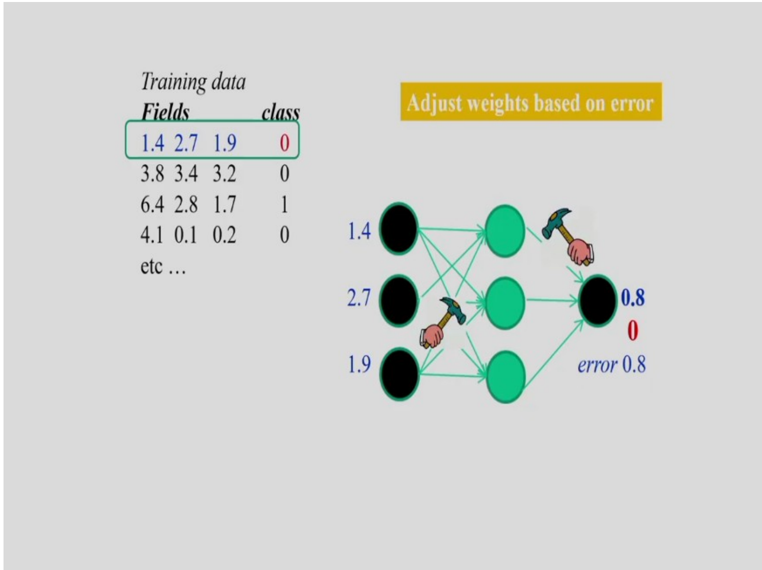
Now, let us consider one artificial neural network here I am considering the weights W1, W2, W3 and suppose my input is - 0.06, - 2.5, 1.4 and I am considering the function is the sigmoid function. So, $f(x)=\frac{1}{1+e^{-x}}$. So, this neural network I am considering now.

(Refer Slide Time: 12:53)



And corresponding to this neural network if you see, if I want to calculate the value x that is the sum value. So, I can determine the sum value like this. So, input that is the input value is multiplied with the weight value the weight is 2.7 plus I am considering the second input the second input is 2.5 and it is multiplied with 8.6 and 1.4 into 0.002 corresponding to this I can determine the value x, x is the sum value.

(Refer Slide Time: 13:27)



Now, let us see how to do the training in case of the supervised artificial neural networks. So, what we have to do? I have to determine the actual output and also we know the desired output. The difference between these two is called the error. So, I have to minimize the error for minimization of error I have to adjust the weights of the artificial neural network. So, in this example I am considering one artificial neural network.

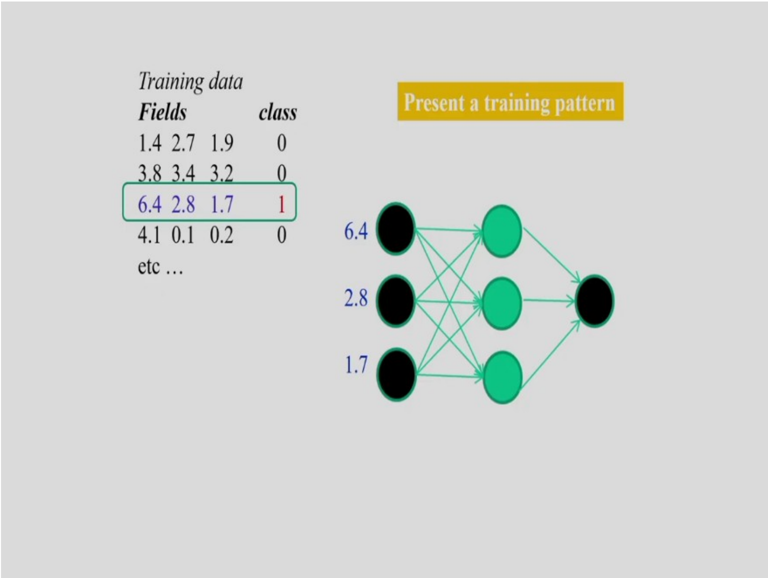
And you can see I have the input layer, suppose one layer is there, there is the input layer another layer is there suppose these is the hidden layer. And another layer suppose I have the output layer. So, in this case I am considering two classes that is pattern classification problem. So, corresponding to the first class the output will be 0 0 and corresponding to the second-class output will be 1 and I am giving the input data set.

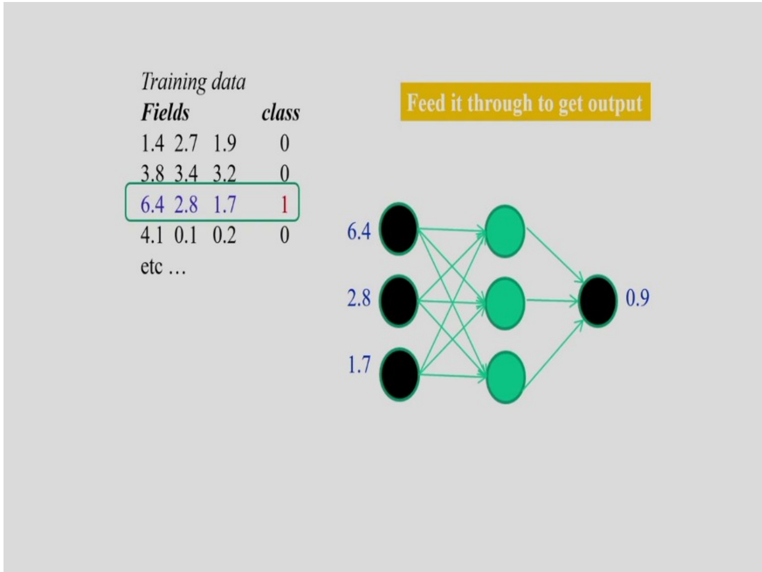
The input data set is 1.4 2.7 1.9 and corresponding to this the class will be 0. So, that information is available that is why it is called the supervised learning. Now how to do the training here you can see so first I have to initialize the random weights. So, random weights I have to select in the artificial neural networks. So, randomly I am selecting the weights and let us consider the first input, the first input is 1.4, 2.7 and the 1.9.

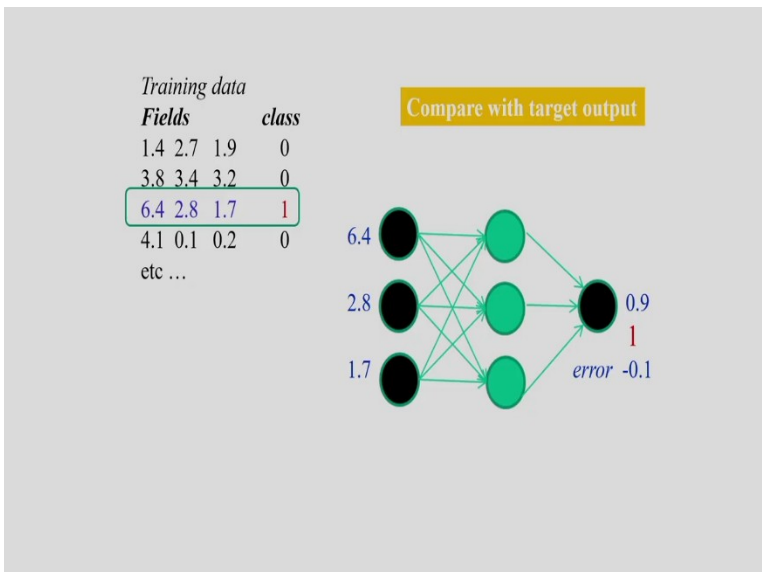
And corresponding to this the output should be 0 corresponding to a particular class that class is suppose W1. So, I am getting 0.8 corresponding to these inputs. So, input is 1.4, 2.7, 1.9 and what I am doing randomly I am selecting the weights of the artificial neural network. So, corresponding to this I am getting the output, output is 0.8 but my output should be 0. So, that is why the actual output minus desired output is 0.8 that is the error.

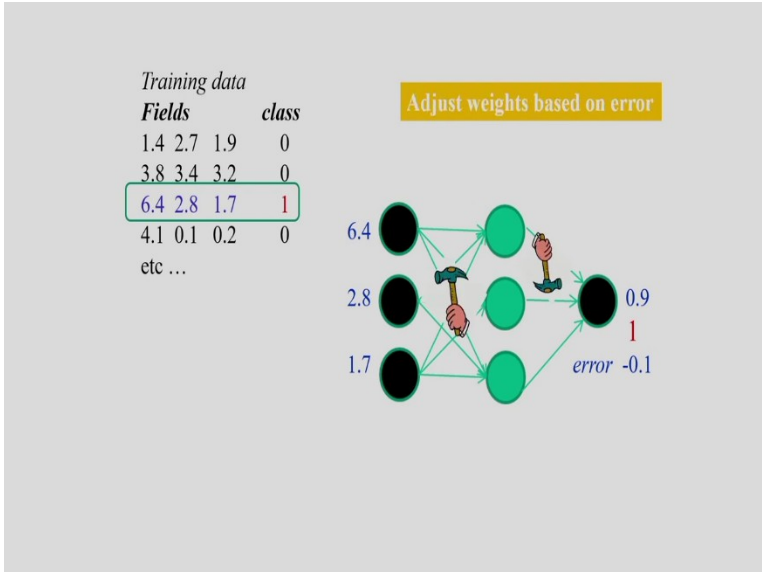
So, I know what is the desired output corresponding to a particular training data. So, that is why it is call the supervise network. I am repeating this so corresponding to a particular training data, training samples I know what will be the output. So, output should be 0 but I am getting the output 0.8. So, that is why the error will be 0.8. the error means the difference between the desired output and actual output. So, I have to minimize this error so for this I have to adjust the weights of the artificial neural network. So, that means I am adjusting the weights of the artificial neural network.

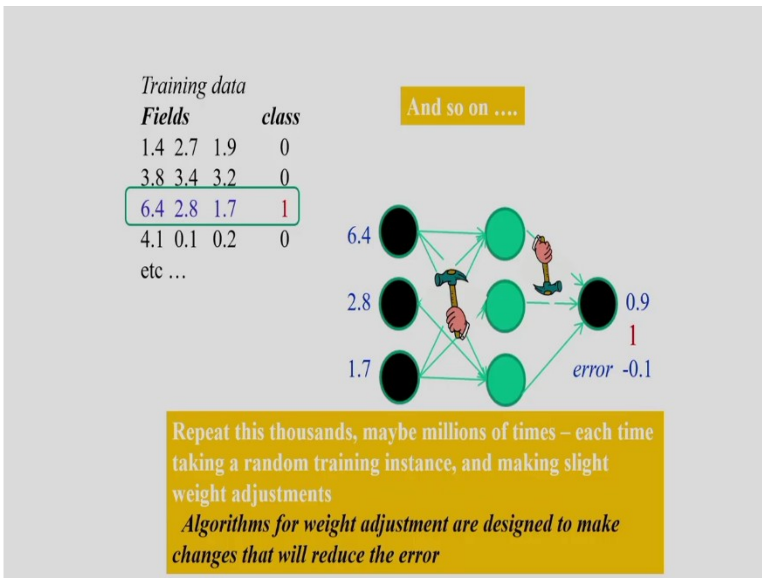
(Refer Slide Time: 16:17)




Training data
Fields class
1.4 2.7 1.9 0
3.8 3.4 3.2 0
6.4 2.8 1.7 1
4.1 0.1 0.2 0
etc …
Feed it through to get output
6.4
2.8
1.7
0.9


Training data
Fields class
1.4 2.7 1.9 0
3.8 3.4 3.2 0
6.4 2.8 1.7 1
4.1 0.1 0.2 0
etc …
Compare with target output
6.4
2.8
1.7
0.9
1
error -0.1

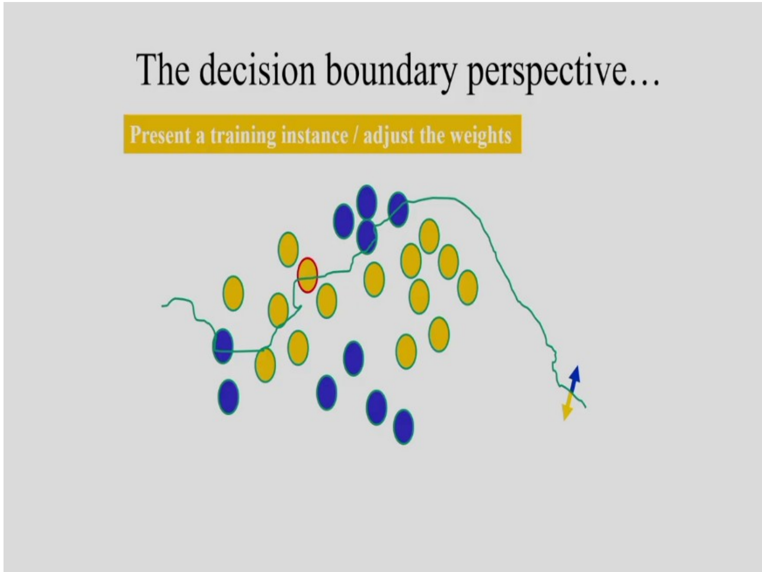




After this you can see let us considered the second input the second input is suppose 6.4, 2.8, 1.7 and corresponding to this my output should be 1, corresponding to another class so suppose the class is W2. And in this case if I give this inputs 6.4, 2.8 and 1.7 I will be getting the output, output is 0.9 but desired output should be 1. So, that is why the error will be - 0.1. So, to minimize the error I have to adjust the weights of artificial neural networks.

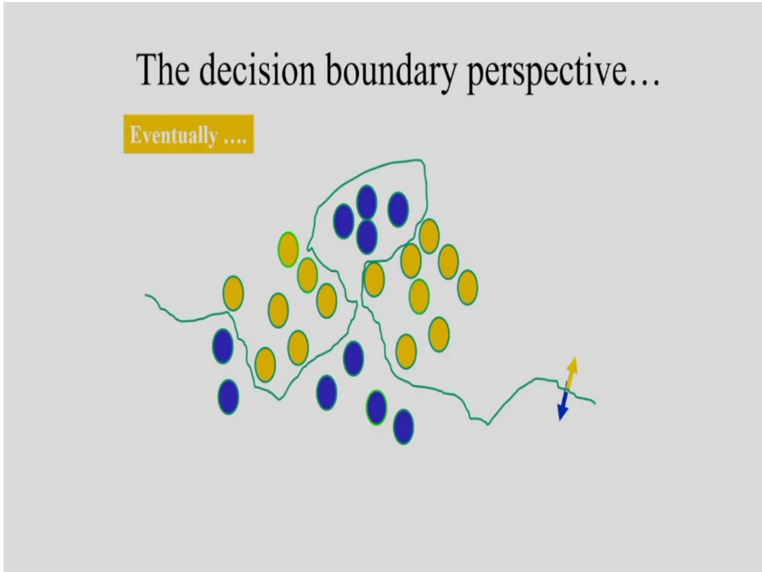
So, I am adjusting the weights of the artificial neural network. And this process I have to repeats all the training samples and finally I will be getting the train artificial neural networks. Train means the weights will be adjusted. So, this is the concept of the training of artificial neural network.

(Refer Slide Time: 17:16)



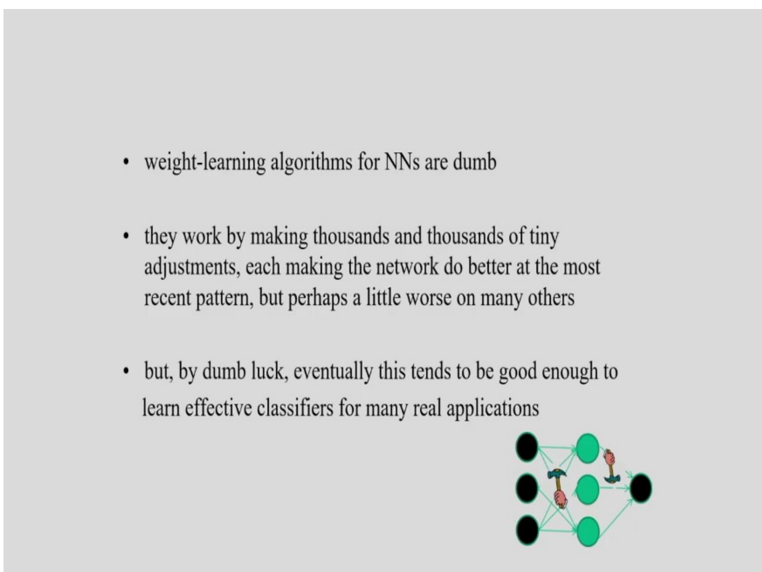
And you can see the concept of the decision boundary so in this case I am showing two classes one is the yellow another one is the blue. And randomly I am initializing the weights and corresponding to this you can see the decision boundary. Now, after this I have to adjust the weights that means I am doing the training and you can see the moment of the decision boundary. So, you can see the decision boundary it is moving because I am adjusting the weights because I have to minimize the error in the output. After this again I am adjusting the weights and you can see the position of the decision boundary. Like this I have to do the training.

(Refer Slide Time: 17:52)



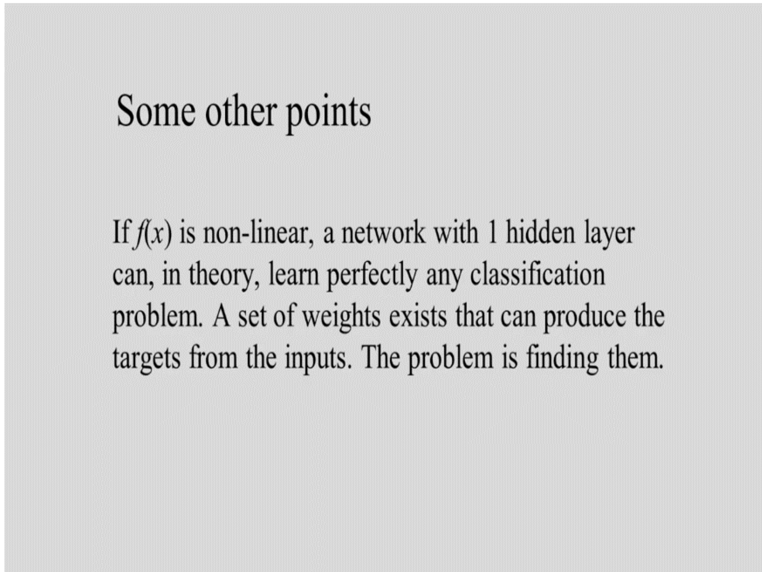
And finally you can see the decision boundary will be like this. So, I am getting a nonlinear decision boundary between these two classes. And it shows the separation between two classes that classes are one class is blue class another is the yellow class. This is the objective of artificial neural network training.

(Refer Slide Time: 18:22)



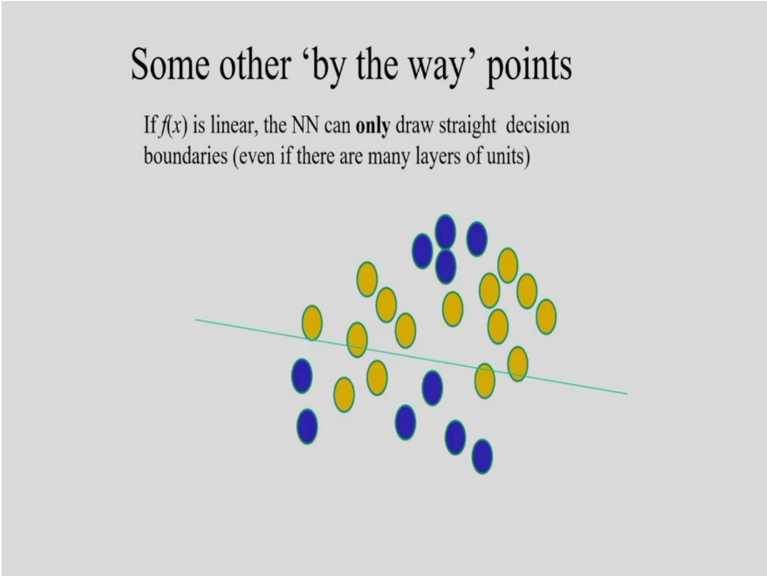
So, here you can see the weight learning algorithms for neural networks are dumb. And in this case we have to consider the thousands and thousands of adjustment and ultimately we will be getting the train artificial neural networks. The training procedural already I have explained.

(Refer Slide Time: 18:40)



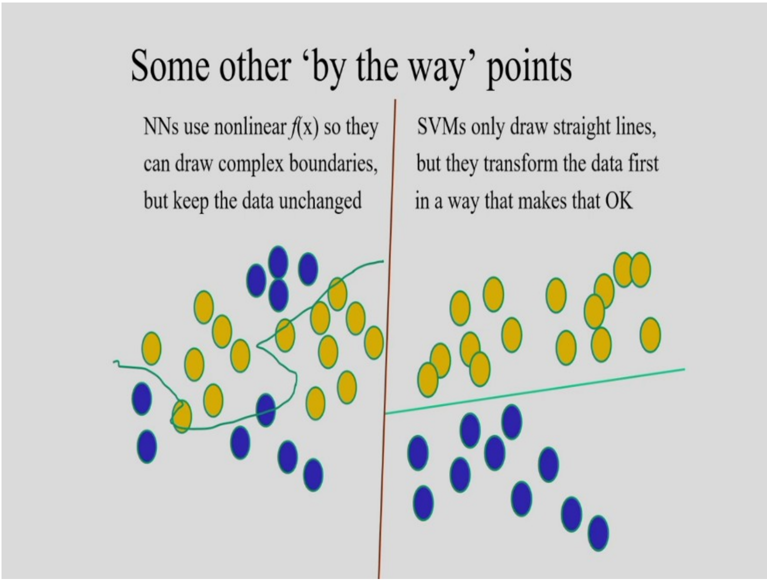
Now, in this case if the function fx get I have considered in my previous slide that is the sigmoid function. Suppose the function fx is not linear. The sigmoid function is nonlinear function then a network with one hidden layer can learn perfectly any classification problem. A set of weights exists that can produce the target from the input. The problem is finding them. So, that means corresponding to this fx if the fx is nonlinear then what I can consider may be the network with only one hidden layer will be sufficient for most of the classification problems.

(Refer Slide Time: 19:21)



And suppose the fx is linear that is the that sigmoid function I am not considering. A sigmoid function is also called as the activation function. Suppose the activation function is linear then in this case I can draw only the straight lines that is the straight decision boundaries I can draw between two classes. So, that means if I considered an activation function fx is linear then the neural network can only draw straight decision boundaries between the classes.

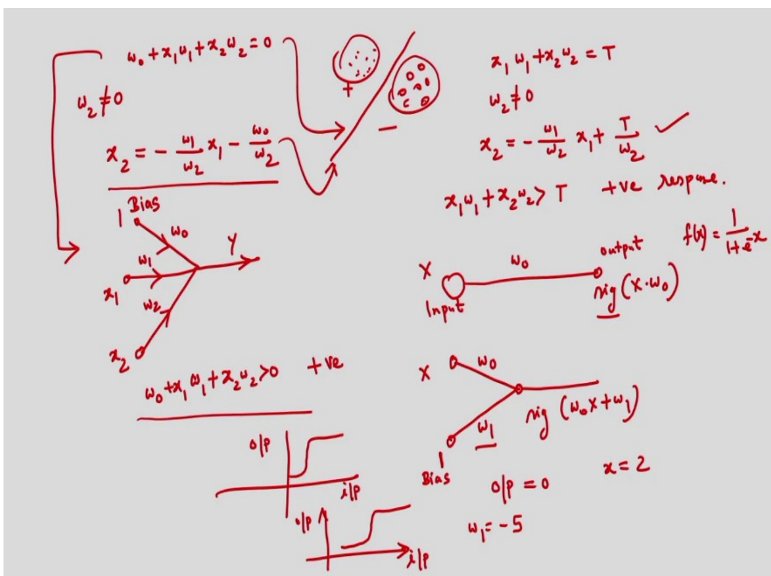
(Refer Slide Time: 19:53)



And that is why we should consider the nonlinear activation function that I have considered that is the sigmoid function is the nonlinear activation function to draw complex boundaries between two or more classes. So, that is the importance of the nonlinear activation function. And in case of the support vector machines only draw the straight lines. But they transform the data first in a way that makes that.

So, that means the support vector machine transform the data and after this transformation. The linear decision boundary will be sufficient to separate two classes or two or more classes. It will be sufficient the linear decision boundary we can consider in case of the support vector machine. But in case of the artificial neural network, we considered nonlinear activation function fx. So, that is why we can consider complex decision boundaries between the classes. Now, let us consider the concept of that artificial neural network. How it can be used for a pattern classification. I will give some simple examples suppose if I consider.

(Refer Slide Time: 21:15)



Suppose this equation I am considering $w_0+x_1w_1+x_2w_2=0$. So, I want to do simple pattern recognition tasks. And by considering two-layer artificial neural networks. And in this case, I have to consider the separation of the input space into regions. Where the response of the network is positive and the regions where the response of the network will be negative that I want to consider.

So, that means suppose if I consider two classes, suppose one class is this and another class is this. And I want to find a decision boundary between these two. And what will be the response of the artificial neural network. So, corresponding to the first class the response should be positive and the corresponding to the second class the response should be negative. And I want to find the decision boundary between this classes.

So, for this I am considering that equation the equation is suppose $w_0+x_1 w_1+x_2 w_2=0$. That is the equation of the separating line, so separating line I am considering this. So, that is the equation of the separating line. And in this case the w2 is not equal to 0. So, corresponding to this I will be getting the equation of a line. So, $x_2=\frac{-w_1}{w_2}x_1-\frac{w_0}{w_2}$.

So, that is the equation of a line. That is the separating line between two classes. corresponding to this equation I can draw the artificial neural network. So, the network will be something like this. 1 x1. So, this is my artificial neural network corresponding to depth equation the equation is w1 that equation is $w_0+x_1 w_1+x_2 w_2=0$. And I am considering the equation of the separating line.

The equation of the separating line is this. $x_2=\frac{-w_1}{w_2}x_1-\frac{w_0}{w_2}$. So, this is very similar to the equation y is equal to m x + c that equation is the equation of the straight line and that is the decision boundary. Now, in this case I have to get the positive response and the negative response. So, suppose this $w_0+x_1 w_1+x_2 w_2$ is greater than 0.

Then I will be getting the positive response. And the values of w1, w2 and a w_0 are determine during the training process. So, I am repeating this the values of w1, w2, and $w_0$are determine during the training process. So, corresponding to this if the $w_0 + x_1 w_1+x_2 w_2$ is greater than 0 then I will be getting the positive response otherwise I will be getting the negative response.

Because I am considering two classes. So, corresponding to the first class my response will be positive. And corresponding to the second class my response will be negative. The response of the network. And in this network, you can see I am considering one terminal that terminal is called the bias terminal. The input is one and weights corresponding to this a terminal is w_0. So, I am considering one bias terminal.

So, what is the importance of the bias terminal I will be explaining after sometime. And in this case I have shown the concept of the pattern classification. And I am considering two class problem here. So, for one class the response should be positive for another class the response of the network should be negative. And in between I have to draw the decision boundary and in this case you can see I am getting the equation of the straight line.

And that is the decision boundary between the classes. And suppose if I consider suppose this equation $x_1 w_1 + x_2 w_2$is equal to T. Now, I am considering the threshold, the threshold is T and w2 is not equal to 0 if I consider. Then in this case I will be getting the equation, equation is x2 equal to $x_2 = \frac{-w_1}{w_2} x_1 + \frac{T}{w_2}$I will be getting. And in this case if the net input that is the network input.

That is the net input means the sum input the network input x1 w1 if the net input x1 w1 x2 w2 is greater than threshold then in this case I will be getting the positive response. Otherwise, I will be getting the negative response. So, in this case I am considering the threshold. In the second case also I am getting the decision boundary I will be getting the decision boundary corresponding to depth equation.

$x_2 = \frac{-w_1}{w_2} x_1 + \frac{T}{w_2}$. The values of w1 and w2 are determine during the training process. And the response will be negative in this case when the input pattern is not a member of it is class. So, in case of pattern classification the desired response of a particular output unit is positive. If the input pattern is a member of it is class that means the response will be negative when the input pattern is not a member of its class.

The positive response can be represented by an output signal of 1 and the negative response by an output signal of minus 1. So, positive response is represented by a signal 1 and the negative response is represented by the output signal of minus 1. So, that means the response will be negative when the input pattern is not a member of its class. In case of the pattern classification problem the desired response of the particular output unit is positive.

If the input pattern is a member of its class so I have considered this case. So, what is the need of the bias of the artificial neural network. So, that concept I am going to explain suppose if I consider, suppose input is x I am considering and I am considering the weight suppose weight is w naught. And I am getting the output the output is the sigmoid function that I am considering sigmoid function x into $w_0$sigmoid function is nothing but $\frac{1}{1+e^{-x}}$ that is the sigmoid function.

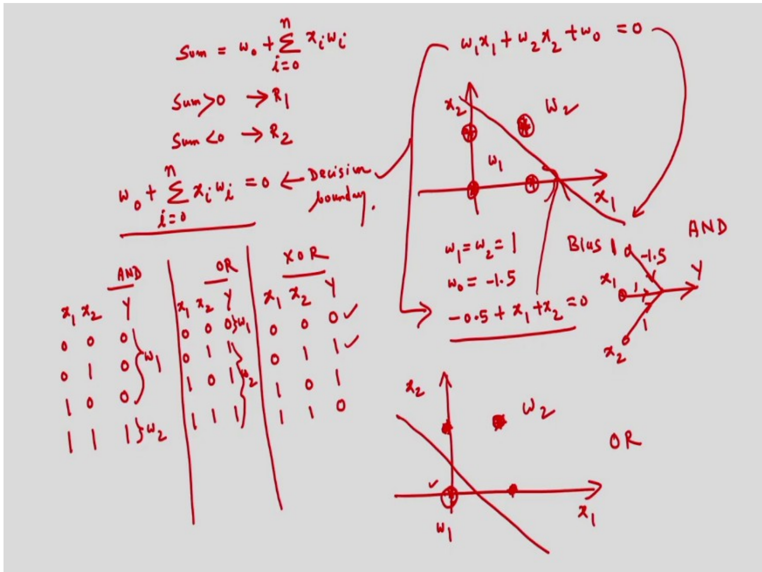
So, that function I considered fx, fx is the sigmoid function so here I am considering this sigmoid function is the fx. In this case I am not considering the bias. In the second case I am what I am considering suppose I am considering the bias input. So, x is the input the weight is w_0. I am considering the bias and corresponding to this input 1 the weight is w1 and this is nothing but the bias.

And in this case, I will be getting the output, output is the sigmoid function w_0x + w1. So, in the second case I am getting this one. Now, what is the importance of bias, so what is the requirement of bias. Based on w1 the activation function can be shifted towards right or left. Suppose we require output of the network for x is equal to 2 suppose. So, suppose we require output 0 for a value x equals to 2.

That means I need the output 0, output should be 0 corresponding to the x is equal to 2. So, that means in this case I have to shift the sigmoid function that means the activation function I have to shift. So, what was my activation function if you see. So, suppose this is my sigmoid function this sigmoid function this is the output and this is my input. This sigmoid function can be shifted the sigmoid function can be shifted now I am shifting the sigmoid function like this.

So, this is my input and this is my output the sigmoid function can be shifted based on the value w1. So, suppose the w1 is equal to - 5 suppose if I considered w1 is equal to - 5 that means the sigmoid function will be shifted towards right. So, that means the activation function can be shifted towards right or left based on the value w1. So, that is why we have considered the bias terminal. Now, let us consider how we can do the pattern classification so already I have explained about the decision boundary.

(Refer Slide Time: 31:17)



So, suppose the sum is equal to that is the sum of the network is $w_0 + \sum_{i=0}^{n} x_i w_i$. So, there is a sum I am determining. And in this case the decision boundary between two regions can be obtain. I am considering two classes and one region corresponds to suppose if I consider a sum is greater than 0 than I will be getting the one region suppose region is R1 and another is region, suppose the sum is less than 0.

Then in this case I will be getting another region, the another region will be R2. And the decision boundary can be obtained what will be the decision boundary. The decision boundary will be $w_0 + \sum_{i=0}^{n} x_i w_i = 0$. So, that is the equation of the decision boundary. So, depending on the number of input units in the network. This equation represents a line a plain or a hyper plain.

So, if I consider this equation, so depending upon the number of input units in the network artificial neural network this equation represents a line or may be a plain or may be a hyper plain. And the classification problem would be linearly separable when all of the training input vectors for which the correct response is plus 1 lies on one side of the above mention decision boundary. And all of the training input vectors for which the correct response is minus 1 lie on the other side of the decision boundary that is call the linearly separable classification problem.

I am repeating this one that means the classification problem would be linearly separable when all of the training input vectors for which the correct response is +1 lies on one side of the above mention decision boundary. So, this is the decision boundary and all of the training input vectors for which the correct response is minus 1 lie on the other side of the decision boundary that is the concept of the linearly separable problem that is the classification problem is the linearly separable problem.

And also, I want to mention that, that a single layer neural network can learn only linearly separable classification problems. So, this is one important aspect of artificial neural network. So, if I consider only a single layer artificial neural network that can learn only linearly separable classification problems. But if I consider a nonlinearly separable classification problem. Then in this case I need more than two layers.

But if I consider a nonlinearly separable classification problem then I need more than one layer. So, that means I have to consider hidden layers in the artificial neural network. So, this concept I can explain like this suppose I am considering the AND logic, OR logic, and the XOR logic. So, you know about the these what is and, what is or, and what is x or logic. Now, in this case suppose my input is x1 x2 and output is y.

So, corresponding to and logic if the input is 0 and 0 output will be 0. If it is 0 and 1 output will be 0. If it is 1 and 0 the output is 0. And if it is 1 and 1 if x1 is 1 and x2 is 1 the output is 1. Similarly, in case of the OR logic x1 x2 y if it is 0 0 output is 0. If it is 0 1 output is 1. If it is 1 0 output is 1. And if it 1 and 1 output is 1. And if I considered XOR logic so again I considering x1 and x2 and output is suppose y. So, if it is 0 and 0 output is 0. If it is 0 and 1 output is 1. If it is 1 and 0 the output is 1. If it is 1 and 1 the output is 0.

So, XOR means it is the comparator for similar inputs the output is 0 but for a this similar input it is 1 so that is why it is a comparator. So, and logic, or logic, and x or logic I am considering as a classification problem. So, suppose if I consider this equation $w_0 + x_1 w_1 + x_2 w_2$is equal to 0. That is the equation of the decision boundary. So, by using this decision boundary I can implement the AND logic.

So, corresponding to the AND logic, if you see this is suppose my x1 and this is x2 if x1 is 0 x2 is 0 the output will be 0. So, that means I am considering the first point the x1 is 0, x2 is 0 that I am considering. Next one is x1 is 0 and x2 is 1 so that I am considering this one and next I am considering the x1 is 1 and x2 is 0 so that means I am considering this point. And finally, I am considering x1 is 1 x2 is 1 that means I am considering this point.

So, if you see this case, I am considering this as a two-class problem. So, corresponding to this suppose the class is w1 and corresponding to this the class is w2. That means if the output is 0 the class is w1 if the output is 1 the class is w2. There is a two-class problem. So, corresponding to this I can draw the decision boundary, the decision boundary will be like this, this is the decision boundary.

So, this will be in one class so this is the class w1 and that will be in another class. So, this by using this equation I can represent the decision boundary. So, corresponding to this point and the output should be 0, corresponding to this point the output is 0, corresponding to this point output is 0. But corresponding to this point if I consider this point corresponding to this point the output is 1.

So, that means by considering this linear decision boundary I can separate these two classes. This is about the AND logic. So, for implementation of this AND operation if I consider the weight w1 is equal to w2 is equal to 1. Suppose the bias terminal weight is 1.5 suppose then the equation will be 0.5 plus x1 plus x2 is equal to 2. So, that means this equation will be this if I consider w1 is equal to w2 is equal to 1. And w naught is equal to - 1.5.

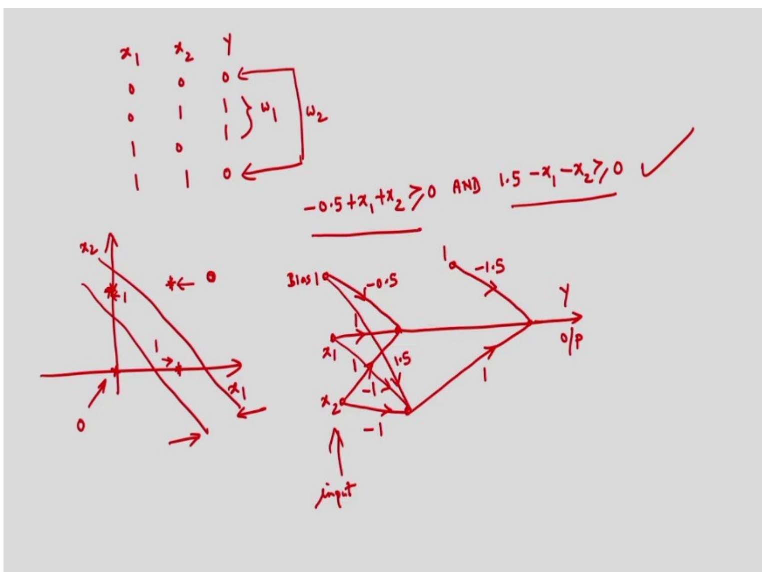
So, corresponding to this my network will be, my artificial neural network will be so one terminal that is the bias terminal is 1. And in this case, it is minus 1.5 and this is x1 this is x1, this is x2 and this weight is 1 and this weight is 1. So, this is the output the output is y. So, by considering this network I can implement the and logic. So, this is the network for and logic the bias terminal I am considering the bias is 1 and corresponding to this bias input my weight is - 1.5 that I am considering.

So, if I consider this equation, if I consider this equation - 0.5 + x1 +x2 is equal to 0 this is the equation of the straight line. That is the decision boundary I am considering. This is about the AND logic. if I considered OR logic what will be my implementation? So, if I consider input is x1 suppose input is another input is x2. So, in this case suppose x1 is 0, x2 is 00 then corresponding to this, this is the point.

x1 is 0, x2 is 1 corresponding to this, this is the point x1 is 1 and x2 is 0 corresponding to this, this is another point and corresponding to 1 and 1 this will be one point. So, if you consider this OR logic so you can see the output is 1 here corresponding to this 0 1 1 0 1 1 that means I can consider this as a class two w2 suppose. And corresponding to the input 0 0 output is 0 so corresponding to this I can consider this is the class w1.

So, I am considering a two-class problem, so I have two classes w1 and w2. And corresponding to this you can see I can draw the decision boundary; the decision boundary will be something like this. Because this will be my class w1 and that will be my class w2 for class w2 the output will be 1 1 1 you can see for class w2 the output will be 1. So, here if the output is 1 here the output is 1 here the output is 1 here the output is 1. But corresponding to this 1, corresponding to this 1 the output is 0. So, you can see the decision boundary for the OR logic and finally I want to show the XOR logic. So, in the XOR logic already I have mention that is.

(Refer Slide Time: 41:35)



If I consider input x1 and x2, if I consider the input x1 and x2 and output is suppose y. So, 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 and 1 0 1. So, this is the XOR logic I have considered. And corresponding to this XOR logic if you plot here x1 x2 so the first point is 0 0 x1 0 x2 0, second point is 0 1, third point is 1 0, and the fourth point is 1 1. And here you can see a corresponding to this condition that is 0 1 and 1 0 condition I will be getting 1.

So, suppose this is the class w1 and the corresponding to another two cases that is 0 0 and 1 1 the output is 0. So, that I can consider as the output corresponding to the class w2. So, corresponding to the class w1 the output is 1 1 and corresponding to this the input will be 0 1 or 1 0. And corresponding to the class w2 the output is 0 and in this case the input will be 0 0 or 1 1.

Then in this case how to draw the decision boundary? In this case it is you can see it very difficult to draw the decision boundary now not like the AND logic or the OR logic. So, I can draw the decision boundary like this. This a decision boundary so corresponding to this if I consider this point what is my output, the output is 0. And corresponding to this point if I consider this what is my output, the output is also 0.

So, that means 1 plus and corresponding to this point if I consider, corresponding to this point the output is 1. And corresponding to this point also if I consider the output is 1. So, you can see the decision boundary for the XOR logic. So, that means a two-layer network can classify data samples into two classes which are separated by hyper plains. However, a network having three layers is required when the problem is to classify samples into two decision regions.

Where one class is convex and another class is the complement of the first class. In case of the x or logic I am getting this one. That means one class is convex and another class is the complement of the first class. A convex set can be approximated by the intersection of finite number of half plains. So, I am repeating this a convex set can be approximated by the intersection of a finite number of half plains.

The nodes in layer one can determine whether a particular sample lies in each of the half plains corresponding to the convex region. And subsequently the layer two of the network perform a logical and to decide if the pattern is in the all of these half plains simultaneously. So, that means a convex set can be approximated by the intersection of a finite number of half plains. The node in layer one can determine whether a particular sample lies in each of the half plains corresponding to the convex regions.

So, this concept I am going to explain now. So, for implementation of the XOR logic. So, in case of the XOR logic I can implement like this suppose if I consider 0.5 + x1 + x2 greater than 0 and after this I am doing the and operation. And I am considering 1.5 - x1 - x2 greater than 0. If I consider this case I can implement the XOR logic. That means the XOR mapping can be implemented as the intersection of the two hyper plains.

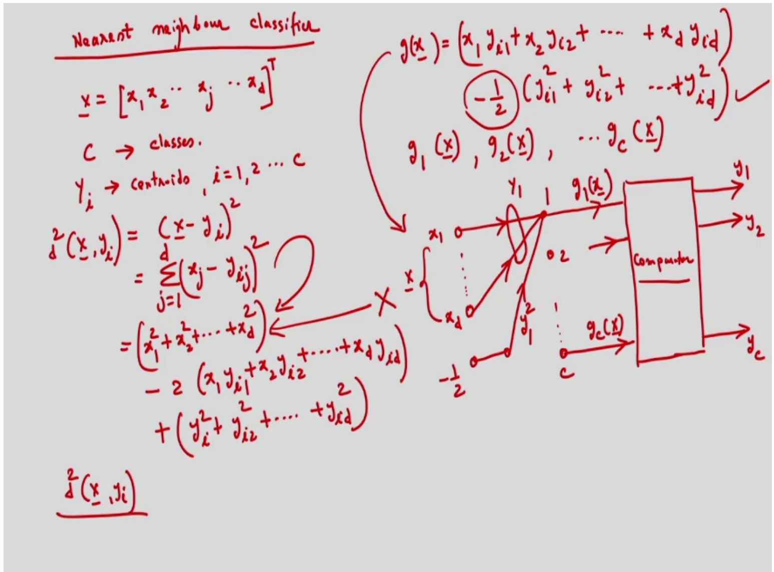
So, I am considering two hyper plains here I am considering and that x or mapping I am implementing as the intersection of the two hyper plains. Because you can see here two hyper plains, this is the hyper plain one and this is the hyper plain two corresponding to the XOR operation. And in this case corresponding to these hyper plains corresponding to these equations I can draw the network that is the artificial neural network I can draw.

So, one input is suppose x1 another input is x2 and the bias is suppose 1. This bias terminal I am considering, so bias is 1 and corresponding to this the weight is - 0.5. And x1 I am considering so this is 1 x2 1 I am considering. So, you can see I am drawing the network corresponding x or pattern classification problem. So, that means I am considering two layers I am considering and in this case, you can see this XOR mapping I am implementing as the intersection of two hyper plains.

And you can see the importance of the hidden layer. So, here you can see I am considering the input layer you can see this is the input layer and we have the output layer and in between you can see the hidden layers. So, for these problems we need the hidden layers. But if I consider and logic AND the OR logic then in this case only with the help of input and the output layer we can do the implementation.

But for the XOR logic we need hidden layers for the implementation because you can see I have two hyper plains. And for this I have to implement by considering suppose this equation and by considering this equation I can implement the XOR logic now I will show the concept of the nearest neighbor classifier. So, how to consider an artificial neural network for nearest neighbor classifier.

(Refer Slide Time: 49:19)



So, let us considered the concept of the nearest neighbor. So, let us consider this case the nearest neighbor classifier so how to implement this. And suppose if I consider artificial vector is x, x is the artificial vector and the elements of the artificial vectors are x1, x2 suppose xj,…, xd transpose. And in this case I am considering the d dimensional artificial vector. So, dimension is d and also I am considering C number of classes.

Now, I am considering C number of classes with the clusters center or the centroids. Suppose yi this is nothing but the centroids. And how many classes I am considering? I am considering c number of classes. So, that is i is equal to 1 to C. So, that means I have C number of classes and corresponding to these classes I have the centroid. And what is the principle of the minimum distance classifier?

So, that means the concept is to classify the input vector. What I have to do? The distance between the vector and the centroids I have to compute. And after this the input vector is assign to a particular class which gives minimum distance. So, that is why it is call the minimum distance classifier that is the nearest neighbor classifier. The distance between the input vector x and the centroid yi can be determine like this.

So, what I am determining the distance between the artificial vector x and the centroid, the centroid is suppose y I am considering that I am determining a distance. And this is nothing but the equality and distance I am considering. So, that I can write like this $\sum_{j=1}^{d}(x_j - y_j)^2$. I can write like this. So, I am considering input artificial vector, artificial vector is x and what is yi that is nothing but the centroid for the classes.

So, for class one the centroid is y1, for the class two the centroid is y2, so I have C number of classes. And this equation I can expand. So, it will be equal to $(x_1^2 + x_2^2 + \ldots x_d^2) - 2(x_1 y_{i1} + x_2 y_{i2} + \ldots x_d y_{id}) + (y_{i1} ¿¿ 2 + y_{i2}^2 + \ldots + y_{id}^2) ¿$. So, that means what I am doing just I am expanding this one that is nothing but a minus b whole square.

So, a square minus twice a b plus b square that like this I am expanding. So, if I consider the first term, the first term is $(x_1^2 + x_2^2 + \ldots x_d^2)$. So, if I consider this term, this term is common for all the classes and it plays no role in classification. So, this term the first term is ¿ like this. So, it is common for all the classes and it has no role in classification.

And in this case, I have to determine the minimum distance a minimum distance what is the minimum distance the minimum distance is $d^2(x, y_i)$ I have to determine. The minimum distance corresponds to maximum discriminate function. Because based on the discriminate function I can take the classification decision. So, what will be my discriminate function. The discriminate function g x will be equal to $x_1 y_{i1} + x_2 y_{i2} + \ldots x_d y_{id}$because I am considering the d dimensional artificial vector $x_d y_{id} - (y_{i1} ¿¿ 2 + y_{i2}^2 + \ldots + y_{id}^2) ¿$ I can consider.

So, that means I am determining the discriminate function and this the first term I am not considering this is I am neglecting because this is same for all the classes. And it has no role in classification the minimum distance between x and y corresponds to maximum discriminate function. So, I am considering the discriminate function, so this is the discriminate function I am considering.

I have to compute the discriminate function g1 x for the class one, g2 x for the class two like this I am determining the discriminate function for all the classes I am determining g_c x I am determining. And I have to find which one is the maximum. The maximum discriminate function corresponds to the desired class. Now, how to implement this gx in the artificial neural network that means how to implement this one.

Because I have to find the maximum discriminate function. So, you can see I am drawing the network. So, my input is x1 like this I have the d dimensional artificial vector. So, this is my input, input is x. So, I am considering the centroid the centroid y1 suppose corresponding to the class the class is one. And you can see if I implement this, this should be minus half because minus half term I have considered here, minus half.

And this weight is ${y_1}^2$ that weight I am considering. And for all this cases one class number two like this for all the classes I have to connect like this we have C number of classes. And corresponding to this I can determine the discriminate function I can determine. So, after doing this one I can determine the discriminate function, the discriminate function is g1 x I can determine corresponding to the class 1.

Corresponding to the class two also I can determine a discriminate function. Corresponding to the class c I have to determine the discriminate function gc x. And here I am considering the comparator one comparator I am considering. Here I am considering one comparator because I have to find the maximum discriminate function. And corresponding to this my output will be the corresponding class.

So, I am considering the class 1, 2 up to C number of classes. And for each and every classes I have the centroid is $y1, y2, \ldots, yc$ like this. So, this is the structure of the artificial neural network corresponding to the nearest neighbor classifier. So, you can see I have to compute $g1x, g2x, \ldots, gcx$ I have to compute. After computing I have to find the maximum discriminate function. So, that is why I am considering the comparator that comparator I am considering and based on this I can determine the maximum discriminate function.

And that corresponds to the class. So, this is the concept of the nearest neighbor classifier that can be implemented by using artificial neural network. So, in class I discuss the concept of artificial neural networks. And how we can adjust the weight in case of the supervised artificial neural networks. So, the training procedure is like this I know what is the desired output and also, I can compute the actual output.

The difference between the desired output and the actual output is the error. And I have to minimize the error. For minimization of error, I have to adjust the weights of the artificial neural network that is the training of the artificial neural network. After this I discuss the concept of the decision boundary, I consider three cases one is the AND logic one is the OR logic and another one is the XOR logic.

So, for the AND logic and the OR logic it is very simple the one line I can draw that is the separating line I can draw for these two classes but in case of the XOR logic I need two hyper plains that means the two lines I need and I have shown the implementation corresponding to the XOR logic. In my next class I will discussing different types of artificial neural networks also I will be discussing supervised and the unsupervised neural networks. So, let me stop here today. Thank you.