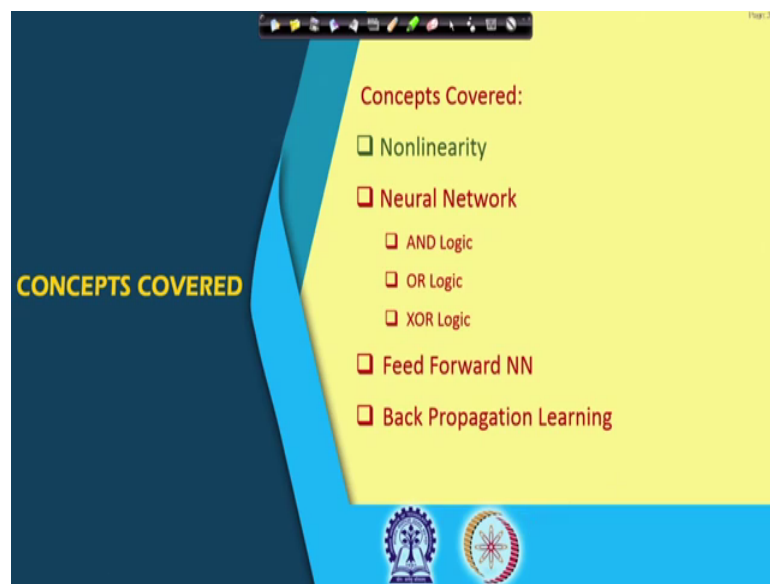


Deep Learning
Prof. Prabir Kumar Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 19
Introduction to Neural Network

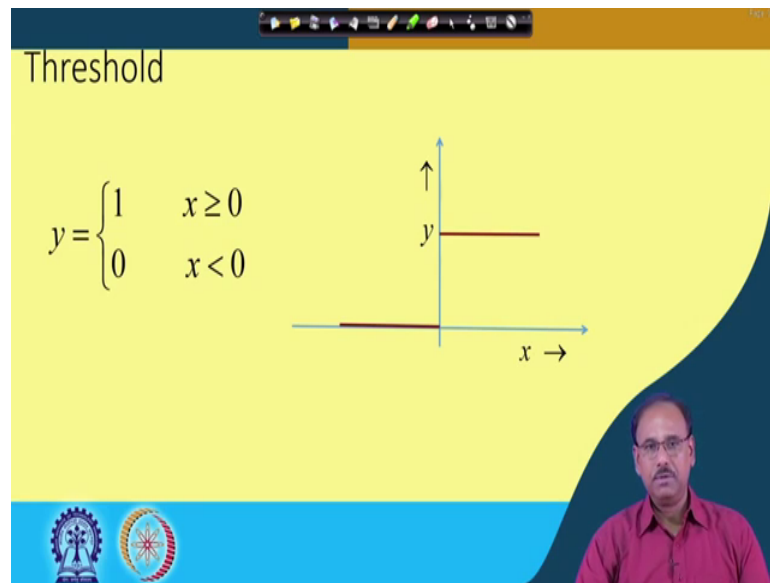
Hello welcome to the NPTEL online certification course on Deep Learning. In our previous lecture we have talked about the various non-linearity functions.

(Refer Slide Time: 00:43)



In today's lecture what we are going to talk about is the Neural Network. And, when we talk about neural network, we will initially see that how different logic functions. The simple functions like AND function, OR function, or XOR function can be implemented using the neural network, then we will talk about the feed forward neural network or multi-layer perceptron. And, we will also talk about the learning or the training mechanism of the feed forward neural network which is known as back propagation learning.

(Refer Slide Time: 01:22)

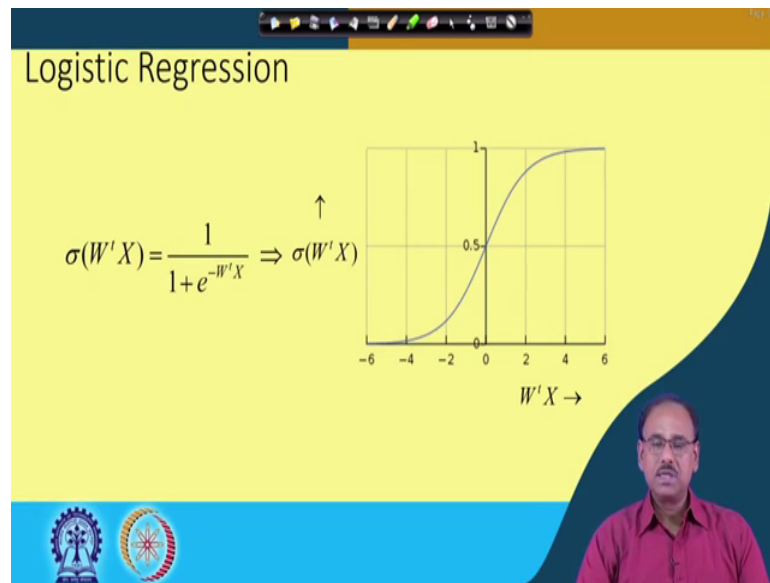


So, before we go to the neural network let us quickly recapitulate that what are the different types of nonlinearities or non-linear functions, that we have discussed in our previous lecture. So, we have talked about the very simple type of non-linearity which is the threshold non-linearity, that is if y is a function of x , then y will be equal to 1, if x is greater than or equal to 0 and y will be equal to 0, if x is less than 0.

So, this is a simple threshold function or a non-linear function, where the threshold value is equal to 0. I can also have a threshold function, where the threshold value can be nonzero may be say I take the threshold value to be equal to 5.

So, in that case value of y will be 1 if x is greater than or equal to 5 and it will be 0, if x is less than 5. So, this is the simplest kind of non-linearity then I can have which is a threshold function.

(Refer Slide Time: 02:32)



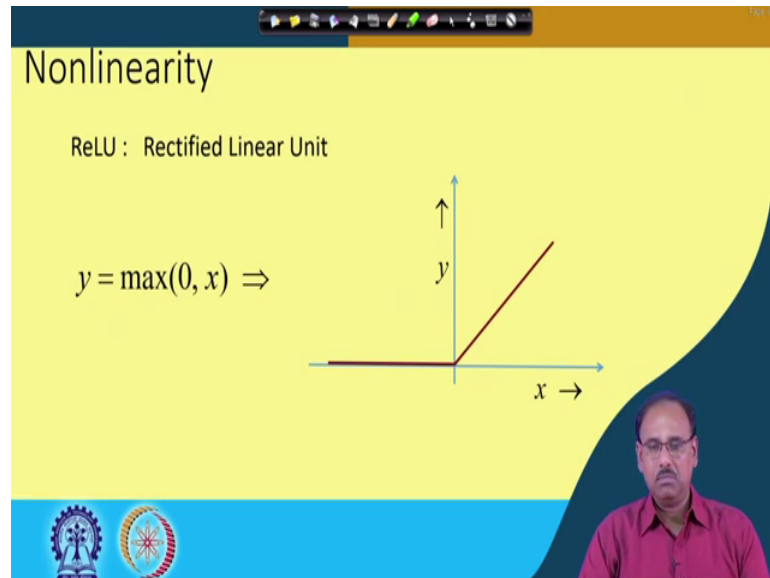
The other kind of non-linearity that we can have is what is known as sigmoidal function, which is used in logistic regression. So, the sigmoidal function is actually given by $1 / (1 + e^{-S})$, which is a sigmoidal function of the argument S . Now, in this case since we are talking about the classifications or machine learning techniques, where we will be frequently talking about the dot product of 2 different vectors W and X , where W is the weight vector and X is the sample vector, then our argument S becomes $W^T X$.

So, the sigmoidal non-linearity or logistic regression will be given by $\sigma(W^T X) = 1 / (1 + e^{-W^T X})$. So, as you find in the right hand side the sigmoidal function has been shown graphically. And, you see that at $W^T X = 0$, the value of the sigmoidal function is half and as $W^T X$ goes on increasing, the sigmoidal function asymptotically reaches a value equal to 1. Of course, it will never reach the value equal to 1, but asymptotically you can say that it reaches the value of 1. And, as $W^T X$ becomes negative, as it increases from the negative side or in other terms $W^T X$ goes on reducing on the negative side.

Then, the sigmoidal function asymptotically reaches a value equal to 0. So, this logistic regression actually gives an output a limit on the output, where the output is limited between 0 and 1. And between 0 and 1 we have a smooth transition, where at the center that is at the value of $W^T X = 0$, the sigmoidal function passes

through 0.5. So, this is another type of non-linearity, which we will see that this can this is widely used in implementation of neural network.

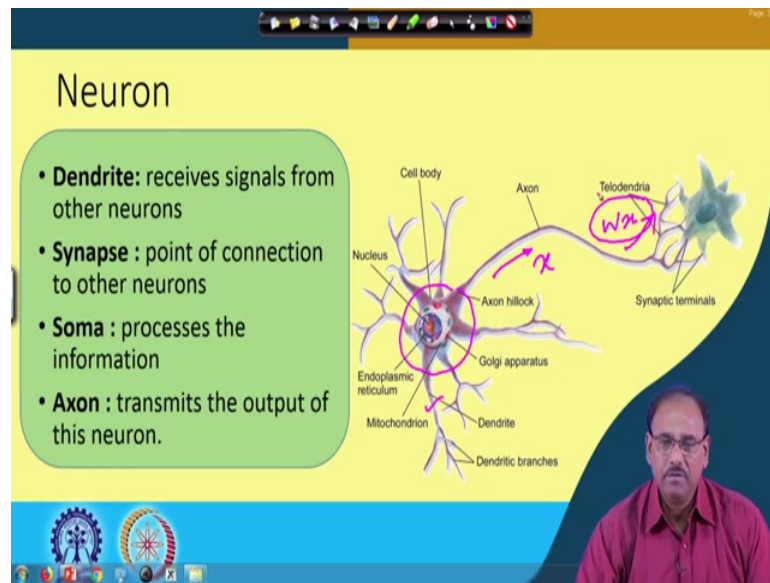
(Refer Slide Time: 04:47)



The other kind of non-linearity is what is known as rectified linear unit or ReLU, which is given by y is equal to maximum of 0 or x . So, if x is greater than 0, then value of y is equal to x , if x is 0 or less than 0, then value of y will be equal to 0. And, the representation the graphical representation of this ReLU function is also shown on the light right hand side in this figure. So, ReLU is also our non-linearity, which is widely used in modern neural networks particularly when we talk about in deep neural networks or deep learning.

So, we will come across all these different types of nonlinearities as we proceed in our discussion.

(Refer Slide Time: 05:44)



So, let us come to the neural network now. The heart of the neural network is neuron. So, when we talk about neural network. The concept of neural network is actually inspired from the way or we believe our brain works. Of course, till now nobody has been able to say with certainty, how the brain actually functions, but this is what till now what we believe how our brain actually functions? So, in our brain we have a network of neurons and if you look at every neuron as is shown in this figure on the right hand side, the neuron consists of a cell body.

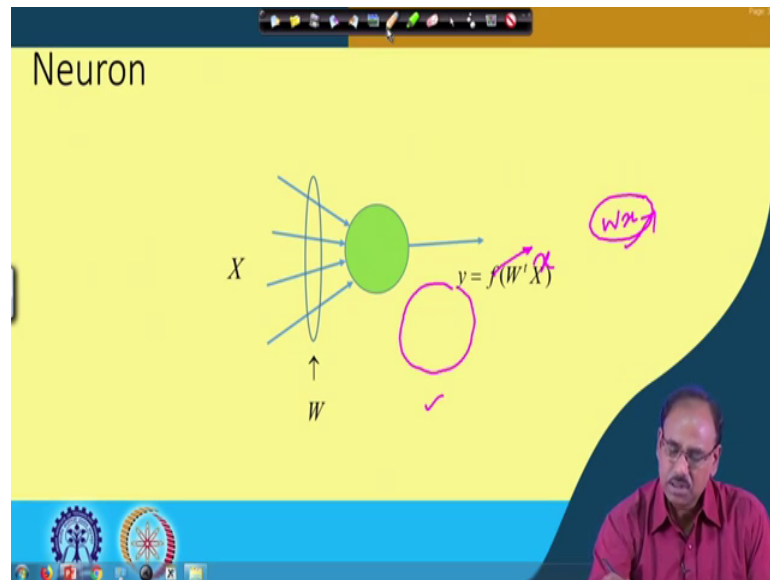
The cell body so, this is the center of this which is the cell body or the nucleus under cell body collects information, it receives information through a number of sensors coming to the cell body, which through a connectors, which are known as dendrites. Cell body processes this information and the information is outputted through a connection which is named known as action.

And, action finally, it branches out and connects to other neurons through synaptic connections. And, it is believed that the information as it passes through actions and then finally, branches out and then it is passed on to other neurons in the network through a synaptic connections in this process there is a multiplicative interaction. What is that multiplicative interaction?

If the signal outputted by the cell body is the X then when it reaches the other neurons, through this multiplicative interaction coming through the synaptic connections, the

value which reaches the other neurons is W times x . So, this is the kind of multiplicative interaction, which is given in the neural network or in the network in the brain. So, when you talk about neural network, we will also see that the neural networks are derived from this particular concept.

(Refer Slide Time: 08:16)



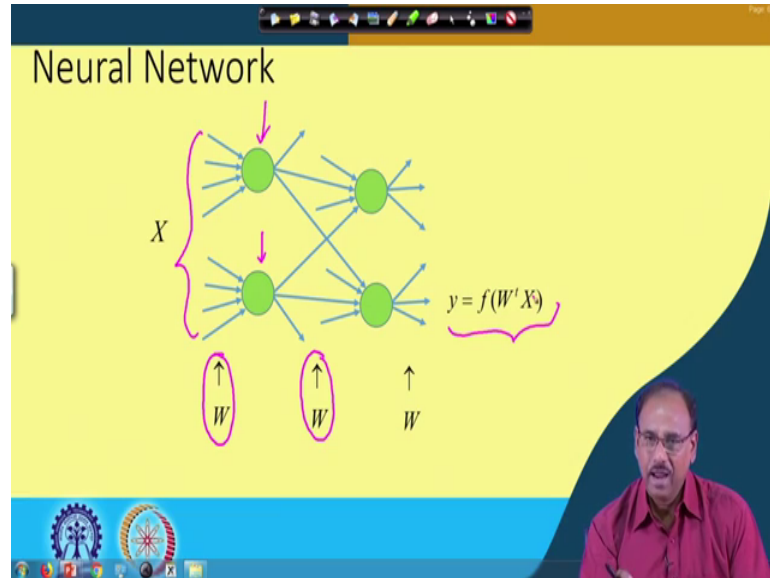
So, what we have in neurons? In neurons we have sales, which receives signals through dendrites and it pass it passes this signals after processing to the other neurons in the network through synaptic connections.

So, the processing is done in an unit in the cell which is known as soma and action is the connecting path, which transmits the signal from one neuron to another neuron. So, this is what is the concept of a neuron in human brains. So, when we talk about a neuron in our neural network, you find that here also every neuron consists of a functional unit, which is the cell body given by this unit. This collects information X or the vector X through a number of inputs, which if we are equivalent to dendrites.

And, when this inputs are coming to the cell body, they passed to our weighting function given by W or weight values given by the weight vector W . And, the output of the neuron is of the form some function of W transpose X , where W is the weight vectors, X is the input vector and output y of the neuron will be a function of W transpose X . And, when you talk about neural network this function f in most of the cases is a non-linear function like the non-linear functions that we have discussed before.

So, we will come to the use of those non-linear functions in neural network in our discussions.

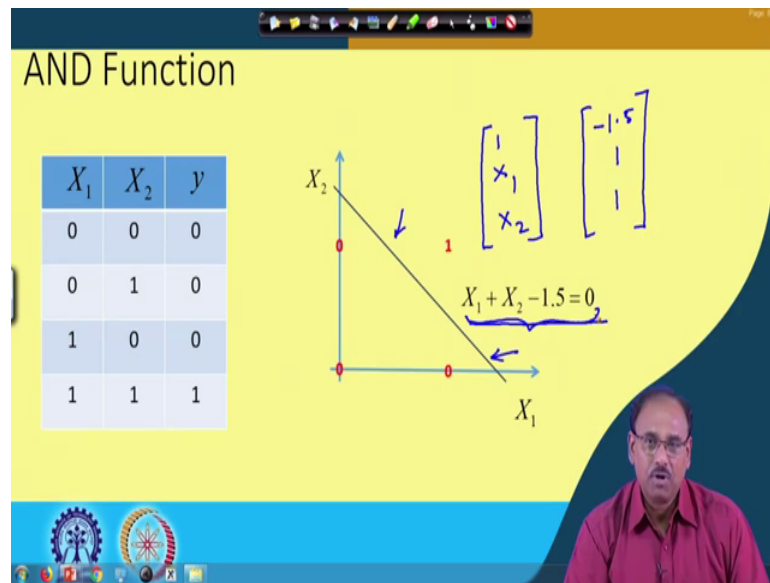
(Refer Slide Time: 10:17)



So, given this model of the neuron a neural network is nothing, but an interconnection of all those neurons. So, here you find that in this figure what we have shown is we have a number of neurons, which collects information X , that is our information vector or sample vector. And, in every level it is passed through or multiplied by an weight vector W . So, I will have a set of weight vectors over here. This processed information from every neuron is passed to the other neurons through the dendrites or synapses, and while it passes through these dendrites or synapses while passing they are also multiplied by another set of weights or weight vectors W and it continues.

And, finally, when you get the output? The output of every neuron or every unit in this neural network is given by this function W transpose X and usually a non-linear function f of this W transpose X , that is what is the output of every neuron right. So, this is how is the architecture of a neural network looks like.

(Refer Slide Time: 11:44)



So, given this let us now see that how these neural networks can be used to implement various functions. So, the first function which is a very simple function, that we discussed is an AND function. And, the and operation that we are going to consider which is a logical operation on 2 units or 2 inputs the inputs are X_1 and X_2 and; obviously, these inputs are binary inputs.

So, I have the input vector, which is given by $X_1 X_2$ and my output function, which is an and function is given by y as shown in the table on the left hand side. So, as you all know that if the input vector is 0 0, then given the function to be and function the output is; obviously, 0 if the input is 0 1 the output is also 0, if the input is 1 0 output is 0 only when input is 1 1 that is both the inputs, both the variables on the input which are binary variables are 1, then only the output of the and logic will be equal to 1. So, if I consider this X_1 and X_2 which are inputs to this and get to be the features or $X_1 X_2$ given together become a binary feature vector, then I have a feature space or a 2 dimensional feature space.

So, if I plot these outputs in this 2 dimensional feature space as given on the right hand side of this figure over here, you see that when X_1 is 1 and X_2 is 0 the output is 0, which is shown here. If, X_1 is 0 and X_2 is 1 then also output is 0. If, X_1 is 0 X_2 is 0 output is 0 only when both X_1 and X_2 they are 1 the output is 1. So, this is how the

functional values will be distributed in the feature space given by the features X_1 and X_2 .

Now, I can consider this to be a classification problem that is when I am considering the input to be a binary feature vector, then I can consider the output to belong to 1 of the 2 process or the input vectors belonging to 1 of the 2 classes. In, 1 class which is class 1 and the other class which is class 0. So, all the feature vectors $(0, 0, 0, 1)$ and $(1, 0)$ they will belong to 1 class when the output should be equal to 0 and only when the feature vector is $(1, 1)$ it should belong to another class and output will be equal to 1.

And, those distributions of the feature vectors are as shown in this plot on the right hand side. Now, consider and now find that considering this to be a binary classification problem. I have to find out a classifying boundary or a classifier which classifies these 2 classes. And, as you see that this is a linear problem as I can separate these 2 classes by using linear boundaries. And, over here though there are multiple boundaries possible that is I can have this as a linear line which separates these 2 classes, this can also be a linear separator which is which separates these 2 classes, but 1 of the option is as shown over here.

And, you find that equation of the straight line in this 2 dimensional space is given by $X_1 + X_2 - 1.5 = 0$. And, as I said that this is one of the many possible linear boundaries that I can have between these 2 classes. So, considering this now you find that I can consider this to be a feature vector, where my feature vector is (X_1, X_2) and I have an weight vector which is given by $(-1.5, 1, 1)$. So, equation of this straight line in that case becomes $W^T X$ or $X^T W$ whichever way I put it because the value of $W^T X$ and $X^T W$ is same.

So, the equation of this straight line is given by $W^T X = 0$ or $X^T W = 0$. And, the feature vectors $(0, 0, 0, 1)$ and $(1, 0)$ that will fall on one side of the straight line and the feature vector $(1, 1)$ will fall on the other side of the state line. And, incidentally if you analyze you find that this particular equation, the classifier that I get this is nothing, but a 2 class support vector machine or a binary support vector machine, because it maximizes the virtual margin. And, as I said that though there are many possible straight lines that I can draw for them the margin will be less than the margin which is given by this.

So, this is also a support vector machine. Now, given this now let us see that how I can implement this using a neural network.

(Refer Slide Time: 17:28)

AND Function

$$W = \begin{bmatrix} -1.5 \\ 1 \\ 1 \end{bmatrix}$$
$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

So, as I said before all the feature vectors taken together I can put that in the form of our matrix and we are also putting this in unified form that is I am adding in each of the feature vectors 1 additional element which will be equal to 1. So, my feature vectors are 1 0 0. So, 1 is added which is an additional element as we have shown over here.

So, this is one of the feature vector, which is 1 0 0 1 0 1 is another feature vector, 1 1 0 is another feature vector and 1 1 1 is the fourth feature vector. And, we also said so, all these feature vectors are represented are put together in the form of a matrix ok. And, out of this we know that this first 4 feature vectors they belong to say class omega 1 for which output will be equal to 0, and for this it belongs to class omega 2 for which output will be equal to 1.

And, I also have this weight vector W which is minus 1.5 1 1. So, given this representation representing all the feature vectors in the form of a matrix and that weight vector now how my classifier will work.

(Refer Slide Time: 18:56)

AND Function

$$X^t W = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1.5 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1.5 \\ -0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \Rightarrow \text{Step Function} \Rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The slide shows the calculation of the AND function output. It starts with the matrix multiplication of the input matrix X^t and the weight vector W . The result is a column vector of values: $[-1.5, -0.5, -0.5, 0.5]^T$. These values are then passed through a step function (represented by a green box with a blue 'S' curve), which outputs a binary vector $[0, 0, 0, 1]^T$. Blue arrows point to the input matrix, the weight vector, the resulting vector, the step function, and the final output vector.

Let us see this one. So, I can put it in the form of X transpose W, where I just put it this way.

(Refer Slide Time: 19:16)

AND Function

$$W = \begin{bmatrix} -1.5 \\ 1 \\ 1 \end{bmatrix} \quad X^t = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

The slide shows the weight vector W and the input matrix X^t . The weight vector is $[-1.5, 1, 1]^T$. The input matrix X^t is $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$. A blue circle highlights the first row of X^t , which is $[1, 0, 0]$, indicating it is a column vector.

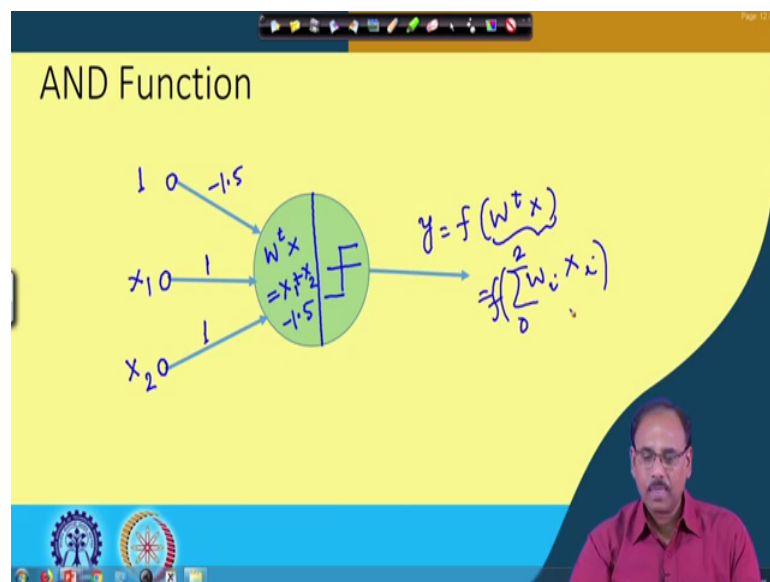
Here instead of writing this as X I will write this as X transpose, because whenever we talk about a vector we usually talk the vector as a column vector. So, this 1 0 0, which is go over here it is actually 1 0 0 that is a column vector. So, instead of writing this matrix as X let us put this as X transpose. So, that every row in this matrix is actually transpose of our feature vectors right.

So, with this understanding you find that the way the classified will actually work is if I compute the W transpose X sorry X transpose W where X is this matrix X transpose is this matrix.

And, W is a weight vector which is this, then the output of this multiplication this matrix multiplication is minus 1.5 then minus 0.5 minus 0.5 and 0.5. Now, if I pass it through a non-linearity. So, if you remember we said that this nonlinearities are non-linear functions are widely used in neural networks. So, this vector that I get, if I pass it through a non-linearity, which is a threshold function my output becomes 0 0 0 1. So, the threshold function is when the input is less than 0 the output should be equal to output should be 0, if the input is greater than 0, then output will be 1.

So, in all these cases here it is minus 1.5 which is less than 0; obviously. So, I will have an output 0 here it is minus 0.5, again I will have an output 0 here it is minus 0.5, again I will have an output 0 here it is plus 0.5 which is greater than 0. So, here I get an output equal to 1. So, you find that this matrix multiplication followed by this threshold operations actually perform an AND operation which is a logical operation.

(Refer Slide Time: 21:23)



So, given this now how a neural network I can design a neuron to perform this particular task. So, in case of neuron as it inputs the feature vectors and I say that the feature vectors are inputted through the dendrites.

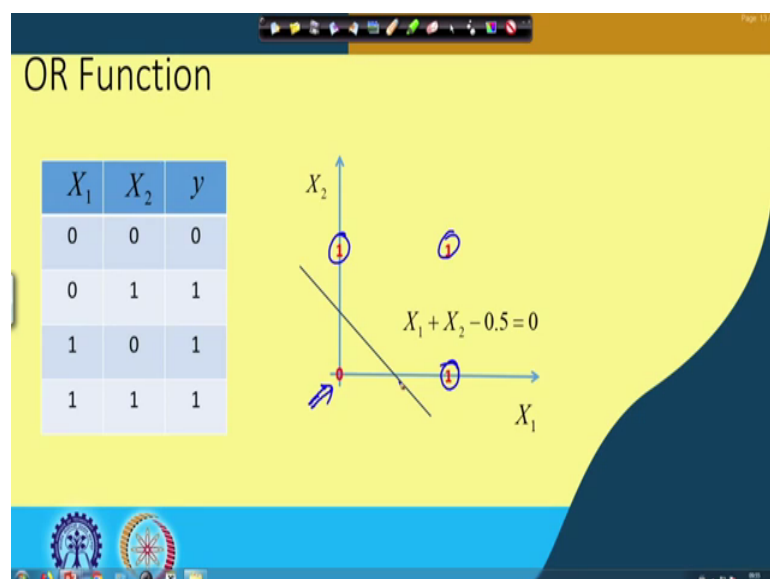
One of the input I will put it as 1. Because the feature vector $X_1 X_2$ we are converting that to $1 X_1$ and X_2 , we are adding an additional component and making that equal to 1, which is you know a unified representation. So, I have one over here, I have X_1 over here, and I have X_2 over here, which are my input vectors. Then, the weight vector I put it as minus 1.5 here it is 1, here it is 1.

So, this function of the neuron I can put it in 2 forms to 2 parts. The first part computes $W^T X$ and what is this $W^T X$? $W^T X$ is nothing, but X_1 plus X_2 minus 1.5. So, it becomes X_1 plus X_2 minus 1.5. And, then the second part of the neuron that gives you the threshold function. This threshold is and at the output what I get is function y is equal to f of $W^T X$.

So, I can put this either in the form of $W^T X$ or I can also write it as $W_i X_i$, i varying from 0 to 2 and function of this. And, in this particular case this function with these weight vectors will be an AND function. So, this is one of the ways you will find that, I can implement an AND logic, which I can pose as a classifier problem as a binary classifier problem with binary inputs.

So, we have 2 dimensional binary inputs X_1 and X_2 and that classifier can easily be implemented by a single neuron. So, I do not need multiple number of neurons or in neural network for that purpose. So, using a simple single neuron, neuron having a threshold non-linearity can implement an AND logic.

(Refer Slide Time: 24:07)



In the same manner let us consider that whether I can have some other logical functions to be implemented in the same manner. So, I consider here the other logical function which is an OR function. Again in case of or function I again consider the inputs to be 2 dimensional binary vectors having components X_1 and X_2 . So, my function will be a both one X_1 , X_1 and X_2 , then output should be 0 in all other cases that is if the inputs are only when X_1 and X_2 both of them are 0.

Then output should be 0 that is it belongs to one class and in all other cases that, when $X_1 X_2$ is 0 1 or $X_1 X_2$ is 0 1 0 or $X_1 X_2$ is 1 1 then output should be equal to 1 indicating that these feature vectors belong to the other class. Again as before if I plot these feature vectors in the 2 dimensional feature space as given over here, you will find that when X_1 , X_2 both of them are 0 the output is 0.

In all other cases the output is 1. Here again it becomes a linearly separable problem. Again you can see that I can have multiple number of straight lines, or infinite number of straight lines, which separates these 2 different classes. One of all these straight lines; one of the straight lines is given by $X_1 + X_2 - 0.5$ which is equal to 0.

And, here you can easily verify that both if both X_1 and X_2 are 0s then output becomes minus 0.5. However, if any of or both X_1 and X_2 they are equal to 1, then output becomes output becomes plus 1.5, clearly indicating that when it is 0 0 the output is negative in all other cases the output is positive ok. So, when it when both of X_1 and X_2 are 1 1, the output is 1.5. If, one of them is 1 and the other 1 is 0 the output is 0.5. However, in all these 3 cases the output is positive.

So, given this so, this becomes a simple linear classifier and when I have this simple linear classifier, you find that a single straight line in the feature space can separate these two different classes.

(Refer Slide Time: 26:55)

OR Function

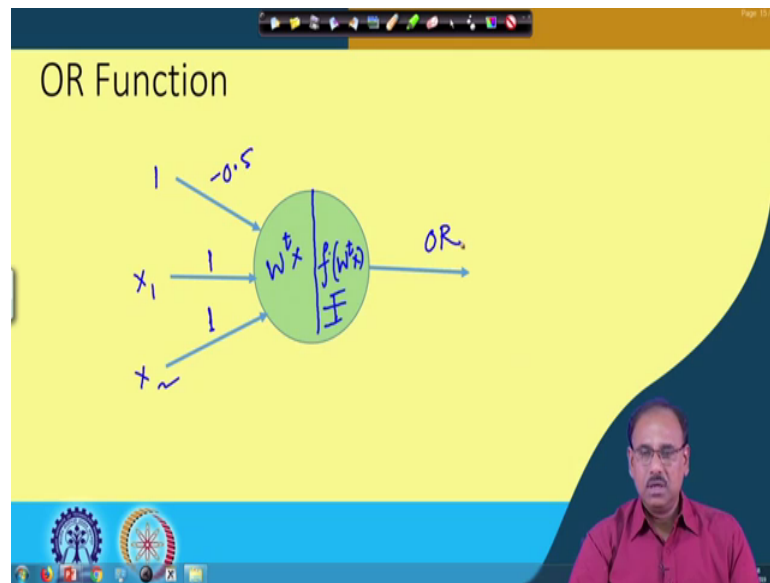
$$X^T W = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.5 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.5 \\ 0.5 \\ 1.5 \end{bmatrix} \Rightarrow \begin{matrix} \text{Step Function} \\ \Rightarrow \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

How can I put it in the form of a neuron? Can, I how can I implement it in the form of a neuron? So, here as we have shown before in the same manner I can do it as X transpose W for the matrix X is the matrix which is formed from all those 2 dimensional feature vectors, W is the weight vector which indicates, what is the separating plane between the two different classes. If, I perform W transpose X , then this is the output vector that I get again you pass through this threshold non-linearity. So, output becomes 0 1 1 1.

So, when my input vector is 0 0 output is 0, when the input vector is 0 1 output is 1 1 1 0 again output is 1 1 1 the output is 1.

So, this simple operation implements the OR logic. And, how do I implement it in neural network again a very simple.

(Refer Slide Time: 27:55)



I put the input vector to be $1 \times 1 \times 2$ and the weight vectors will be minus 0.5 1 1, this neuron computes W transpose X and I have this threshold non-linearity, which performs f of W transpose X , where this f is nothing, but the threshold non-linearity. And, at the output what I get is an OR function. So, again you find that using a single neuron I can implement a OR function. So, it has been possible to implement these logical functions using a single neuron because the problem that have considered they are linearly separable problems both and or functions they are linearly separable.

But, if the problems becomes non separable, what will be our situation and how we can solve those problems using neurons or neural networks that we will explain in our next lecture.

Thank you.