Data Mining
Prof. Pabitra Mitra
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

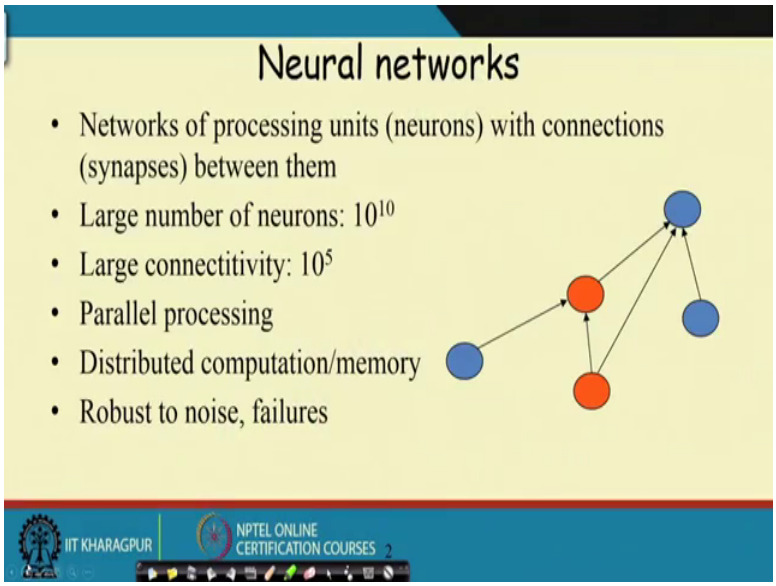
Lecture – 28
Artificial Neural Networks I

We will continue our discussion on classification and prediction models and previously, we had seen that he we had used an support vector machine a linear discriminant and its extension the kernel machine where we have the non-linear surfaces as our classification bounded.

So, we had kind of a 2 classes or more classes and we want to separate them out by drawing a class bounded and one of the way, we saw was to use the support vector machine a linear discriminant which is hiding the highest margin. So, what we will do in our next three lectures is to use another paradigm where we tackle the same problem of classification of putting an object into one of the k classes.

In fact, you can actually use this to a more broader class of problems known as the prediction problems where it is not just a class levels that we can predict, but also if some real valued. So, for example, the temperature of a place tomorrow some real value not class that we can also do with this paradigm I will explain how to do that. So, this new paradigm we are talking of is false under a general umbrella of what is called a artificial neural network.

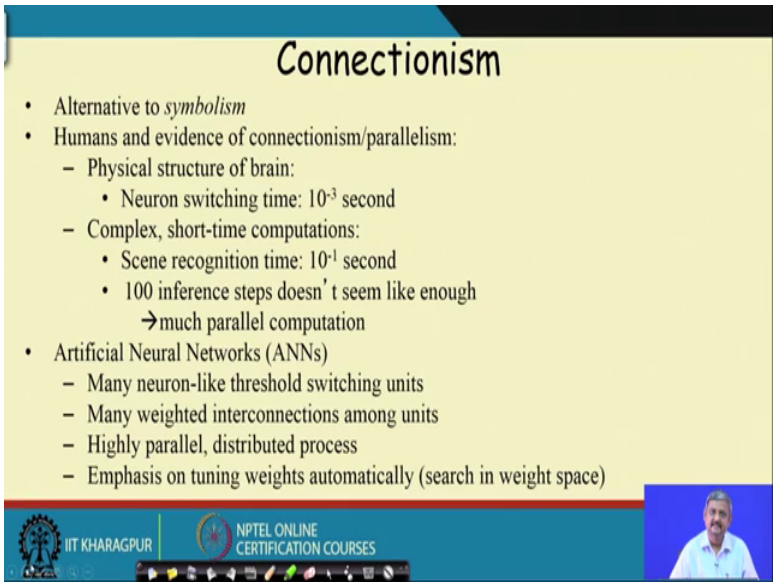
(Refer Slide Time: 01:56)



The main reason for that is that it came from inspiration from the biological neurons. So, it was one of the popular artificial intelligence algorithm in very early in computer science in that 50s and 1950s-1960s is become popular. So, the motivation was motivation was you can perform a complex prediction task by connecting together a number of simple units simple computer processing simple units known as neurons with connections between them known as synapses.

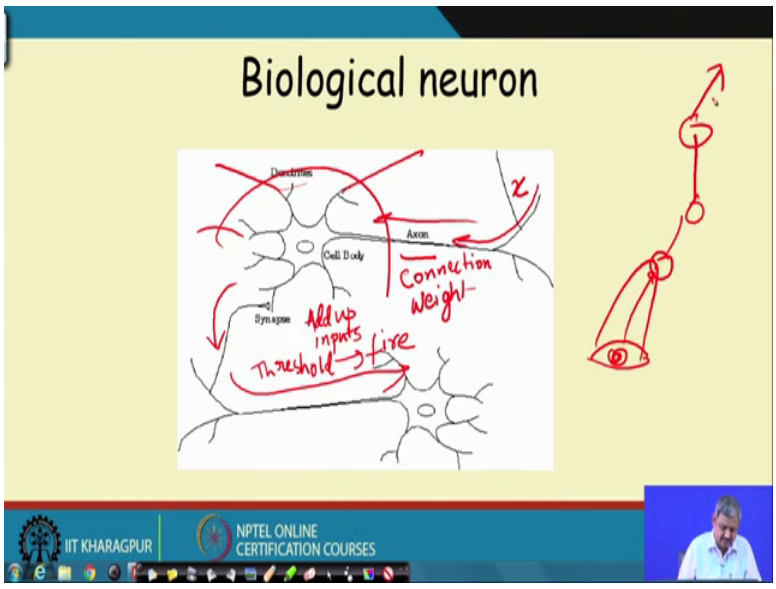
And this was kind of considered to be large scale parallel processing you thought of units come together simple units come together parallel you do it for the task and in a distributed manner solve a problem and it had some nice properties like it is robust. It was low was to both noise and failures and that a whole group of algorithms came to be known as neural networks. So, as I have said that it is kind of a mimic of the natural neuro.

(Refer Slide Time: 03:15)



The biological neuron develop we which has lot of connectionism, it can do very complex talks like face recognition audio recognition we will sort of try to replicate it in a in a computer model by I will explain by something called a simple you neuron unit and you can sort of tune the connection weights so that we can perform a desired task.

(Refer Slide Time: 03:56)



So, let me to motivate come to the structure of a basic biological neuron. So, it has a structure like this. So, it has 2 parts axons and dendrites and these are the neural cells. So, what happens is that as a input signal a stimulus come say a from the I am light; cells

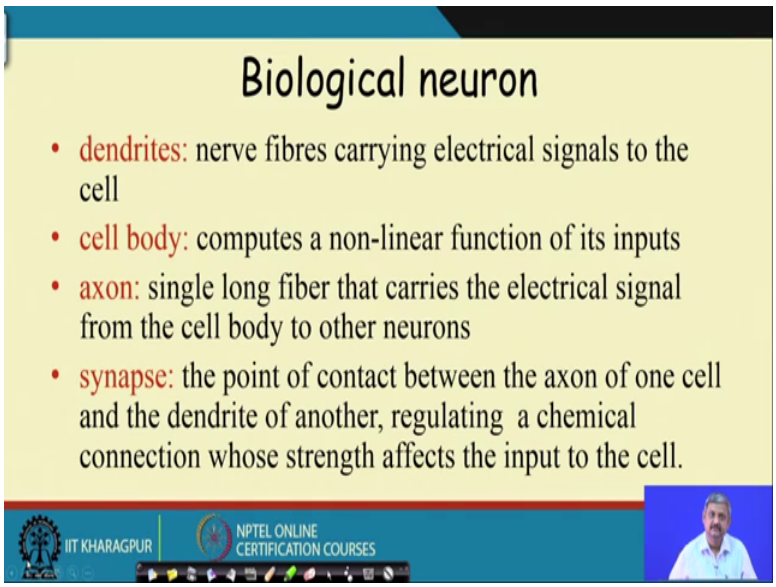
some visual stimulus comes it travels through this axons and where it gets inhibited or enhanced by some factor. So, here is the stimulus let me call it as input and this is the connection and its connection as a weight it increases or decreases x.

And then what happens is that this cell body it receives a lot of such connection it does 2 operation it adds them up of add up inputs aggregates the inputs. So, it is a multiple such neurons connecting to it. It will add up all these things and then it will apply some kind of threshold; that means, if it exceeds certain value it will fire. So, it will have a threshold and if the sum of the inputs exceed that it will fire.

What is meant by fire; that means, it will send another signal to the next level of neuron which will again travel like this? So, for example, our eye say we see some stimulus or I and some signal goes to some neuron which processes, it may be from lot of other neurons sends it to the next neuron in the brain which sends to other neuron in the way and finally, some task is say we recognize, it as a person as a face of the person that happens ok.

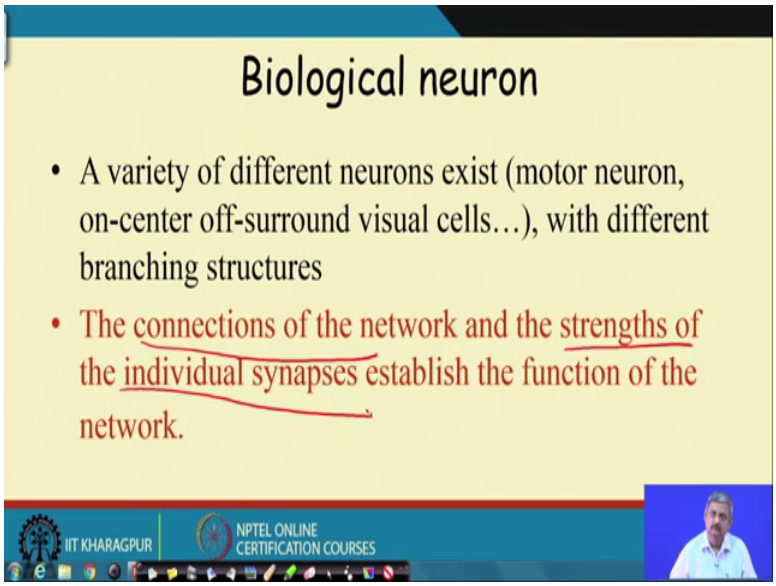
So, this is kind of there; there are much more complicacy in this I am only giving a simplified view it is much more complex, but this is the broad picture.

(Refer Slide Time: 06:49)



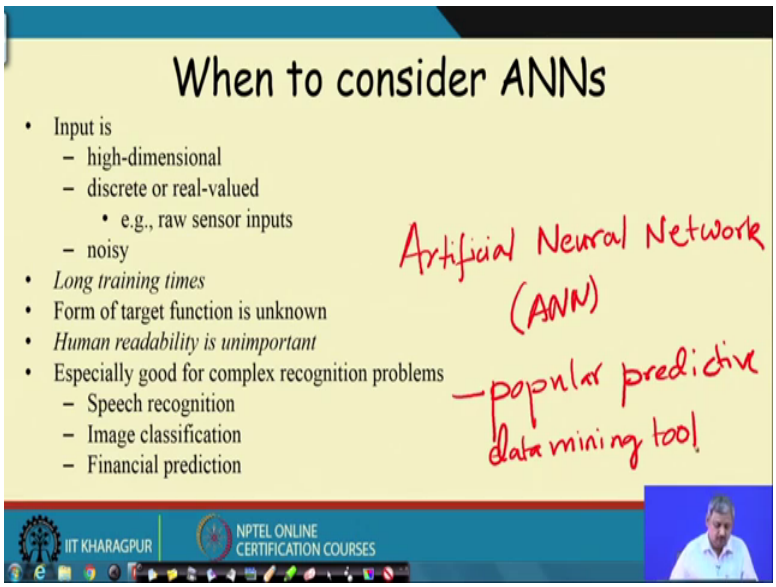
So, as I have said these are dendrites cell body axons and synapses where they connect to each other. So, they carry electrical signals in a non-linear way and affect the next level.

(Refer Slide Time: 07:07)



So, this is this nature of the connection and the strength of the individual synapse which I call as weight which decide finally, what action it will take ok.

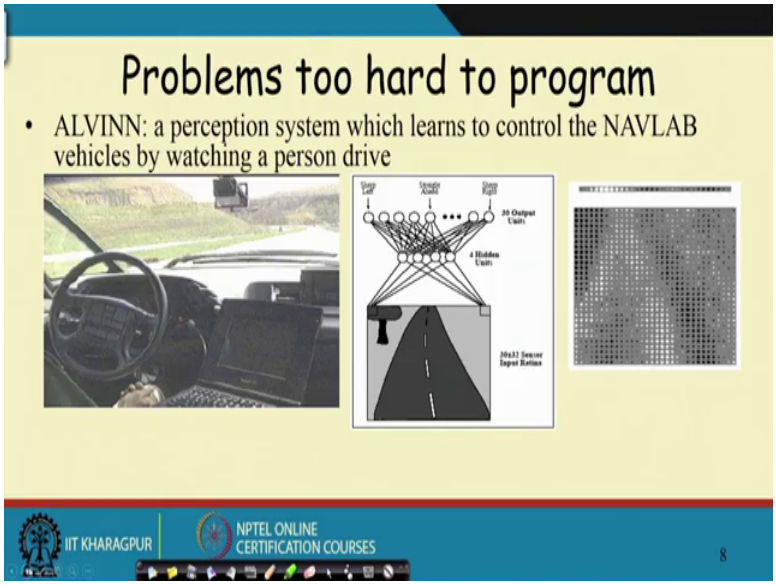
(Refer Slide Time: 07:32)



So, this model is found that it is a good thing to be meek. So, it can do well in lot of tasks where you have lot of high dimensional input for example, in so many visual inputs are coming. So, many pixel information are coming all type of input different type of complex functions they are this kind of model is powerful. So, we will henceforth not refer to the biological neuron, we will refer to something called a I will I will explain

what you say what we shall do relative what is the exact computer model network or ANN, it is a popular determining tool determining tool, right.

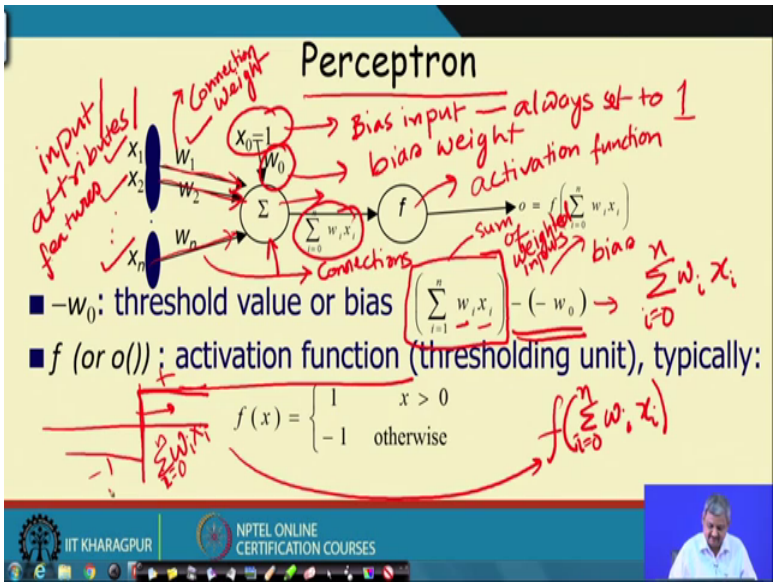
(Refer Slide Time: 08:54)



So, we will explain what it is; as I mentioned you can use it. In fact, it has been it has been particularly you should. So, this is a another sort of advice that only when you have a really complex problem because neural networks will usually require more computational time than other algorithms and that is why I am treating it towards the end of my this is the last classification algorithm that we will discuss.

You use it only in the problems which are the hardest most complex like driving a car and other things only there you use it.

(Refer Slide Time: 09:34)



Given this now let us start with the simplest artificial neural network model which we call as a Perceptron; Perceptron; it kind of mimics the earlier biological neuron that we talked about. So, let us see what this is this is the entire structure of a single capacitor step by step let us see what it is it takes a number of inputs x 1, x 2, xn as many inputs as you want which can be anything real valued usually this real valued real valued.

There are some models for symbolic values are taken, but usually it is real valued x 1 x 2 xn some input values. So, this input values may be length of a person, suppose, I am categorizing a person weight of a person or say credit cut for detection what is the income of that person. So, these are or we can also call it as attributes or these are also called features ok.

Then each of this input is multiplied by a weight of the connections each input is connected by this arrow to the main neuron main neuron each of these are connected these we call as connections and each connection has a weight which we call as connection right there is one kind of dummy input x naught which is always set to one which is always set to one which is called a bias and also similarly there is a weight w naught which is the bias weight.

Then let us see what this; what this threshold what this neuron does. So, what does neuron does; it is a sigma operation it just adds up this weighted inputs. So, it just computes this term weight into the input weight into the overall the inputs in and it. In

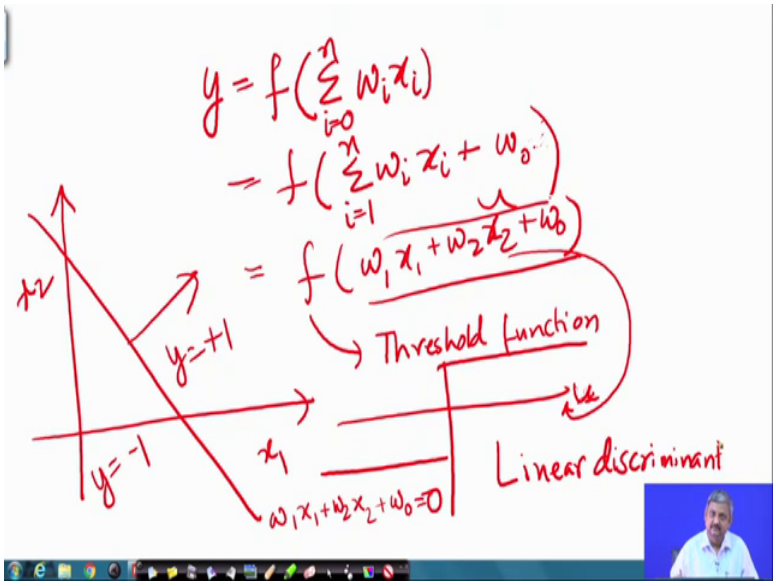
fact, actually you can write it down and it adds up the bias term bias weight w naught. So, actually you could have. So, this is the bias.

In fact, you could have also thought of it like this that in a summation I equal to 0 instead of 1 0 to n w i which includes w naught and x I; x i x naught is always one took it up also written it like this. So, this is the output of the segregation unit this is the output of this. So, then there is another unit f or the activation function the input to this unit is the summed up input weighted sum of the weighted inputs including the bias ok.

And the output is some function of this summation i equal to 0 to n that is the sum function sum f something it is as if some function it applies on this summed up inputs weighted inputs. So, this activation function is very important it can take different forms here is one particular form. So, its value of the output of the application activation function is one if the input is greater than positive 0 minus one otherwise.

So, it looks like this if summation w i x i; i equal to 0 to n if it is positive it gives plus one negative which gives minus 1. So, this particular form has a special significance; I will explain what the significance is the significance is like this.

(Refer Slide Time: 15:57)

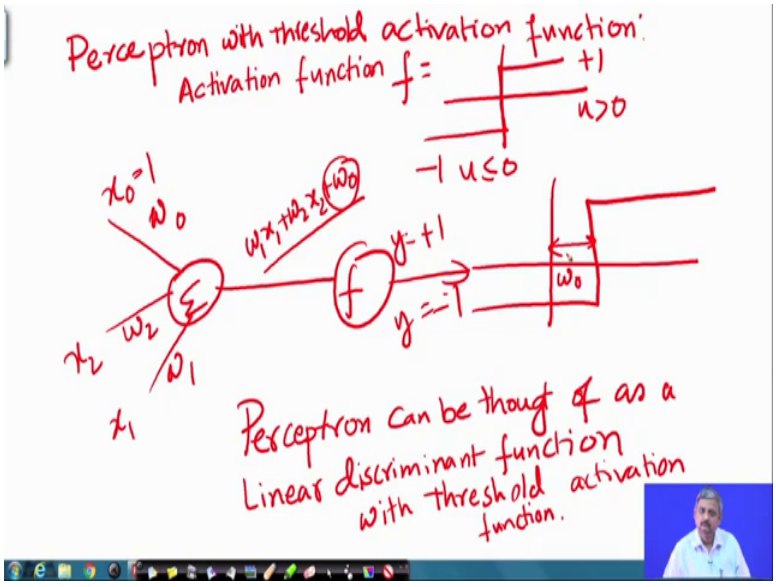


Let me call as the output as in fact, let me write it this way x naught is 1. So, into 1. In fact, I can address that like this. So, if there are 2 terms say 2 inputs I will have things like and y is sum. So, this f is a threshold function how does it look like if this quantity is

negative it is minus one positive plus one this quantity it looks at the x not x rather this looks at some kind of I some input let me call it as u. So, this quantity if I call as u; so, this u is on the x axis and it thresholds it.

Now, this is a geometrically if I look it is a familiar form. So, greater than 0 is plus 1. So, if this equation of this line is 0. So, all this here we will have y equal to plus 1 and all this here equal to y equal to minus 1. So, this thresholded activation function is nothing, but a linear discriminant.

(Refer Slide Time: 18:37)

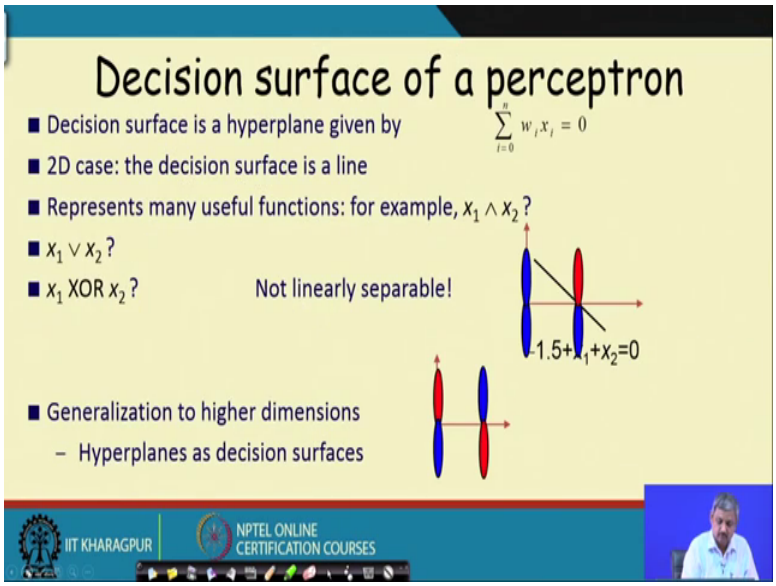


So, let me again write down this. So, we remember it f take this form and you have I am expanding for 2 input features you can extend it to multiple features also. So, this is not this is this on this I apply f which gives either plus 1 y equal to plus 1 or y equal to minus one depending on the sides sign of this quantity greater than 0 or less than 0 this function.

So, Perceptron can be thought of as a linear discriminant function with threshold activation function. In fact, there is a slightly different way of looking at it instead of having w naught as a separate term you can also have it like this a shifted threshold shifted threshold where this is w naught your bias term bias effect this amount of shift all right.

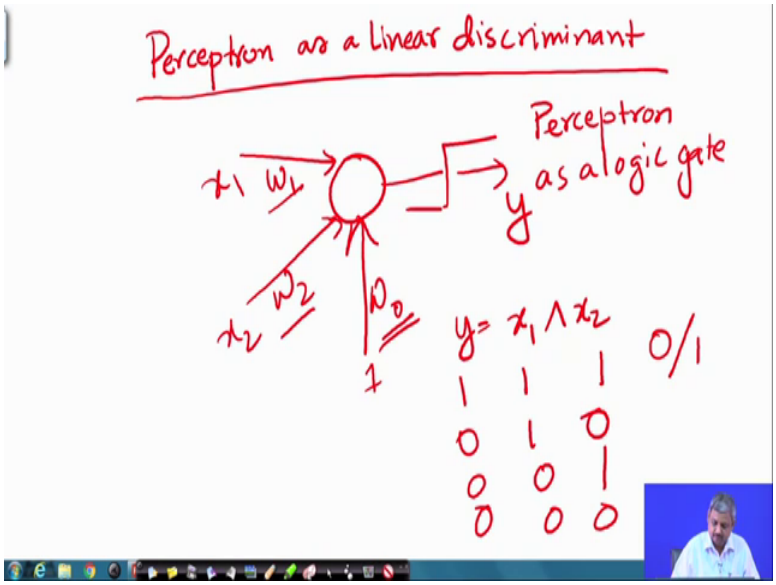
But in general in general you can have you can have other forms of activation function.

(Refer Slide Time: 21:36)



So, this is what I have seen. In fact, this particular way let me explain you this particular way of writing.

(Refer Slide Time: 21:50)



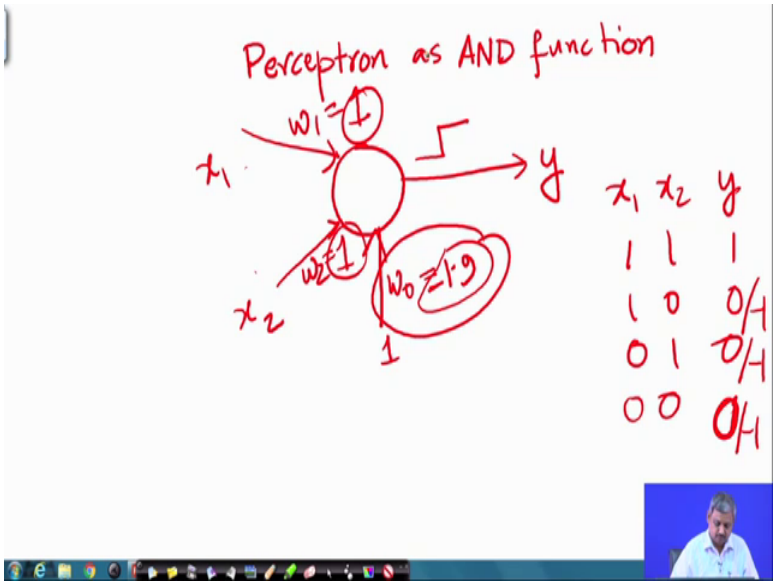
And it can be it was originally used to use Perceptron as a logic gate logic; logic gate, let us see how to do that suppose I have 2 inputs 2 weights and there is a bias term w naught and there is a threshold activation function.

So, what I want this final y as some logical function of this x 1 x 2 say x 1 and x two. So, if we say that x 1 x 2 takes binary values 0 or one values so; that means, when this is one

this is one output should be one all other cases output should be 0 is and gate. So, I can make this Perceptron act like an and gate by choosing w 1 w 2 and w naught properly ok.

So, can you tell how can you tell a value when this will happen let us see let me draw again why.

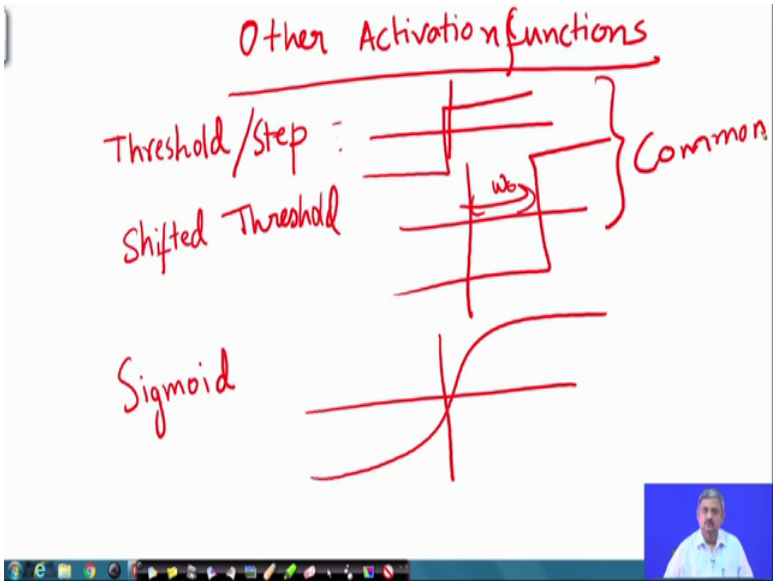
(Refer Slide Time: 23:51)



So, what I want. So, let me choose a value of w 1 w 2. So, let us say; let us try this if w 1 is 1 w 2 is 1. So, if this is say one point nine what happens when this is one this is one this is one when this is one this is 0; this is less than 1 sorry this is not 1.9; this is minus 1.9 less than 1 and so, the less than 0.

So, this will come to minus one side or say 0 side; this will become 0. So, let me actually let me. In fact, I can choose this at like this also by convention and I can do it. So, these values minus 1.911 makes it elegant you can realize other logical functions also.

(Refer Slide Time: 25:42)



Threshold is this threshold or sometimes called step also is like this shifted threshold just like this it is like this approximation of that ok.

So, you can choose, but this is the most common. In fact, the original definition of Perceptron mainly something; so, I hope the overall architecture. So, to say the function is clear next class we will talk about how to tune the weights to perform a task.

Thank you and breaking our last ago.