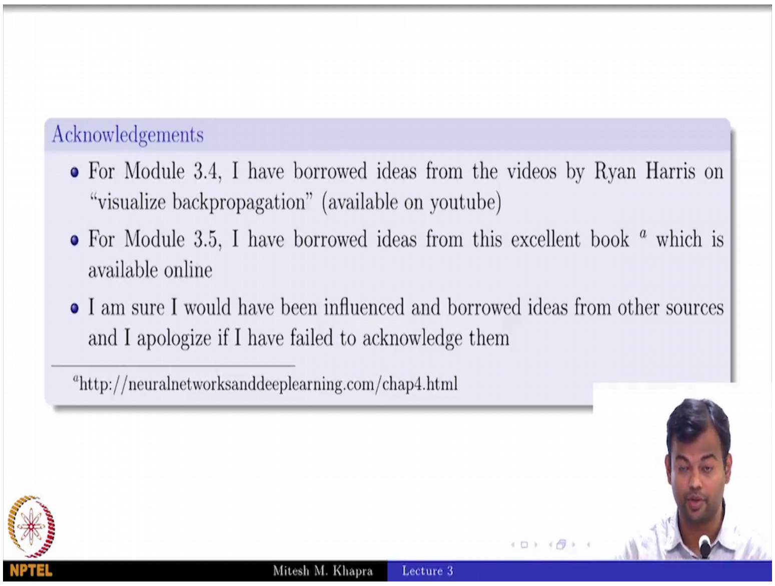
Deep Learning
Prof. Mitesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 03
Sigmoid Neurons, Gradient Descent, Feedforward Neural Networks, Representation Power of Feedforward Neural Networks

We are in lecture 3 of CS7015. And today we are going to cover the following modules, we are going to talk about Sigmoid Neurons, Gradient Descent, Feedforward Neural Networks, Representation Power of Feedforward Neural Networks.

(Refer Slide Time: 00:31)



Acknowledgements

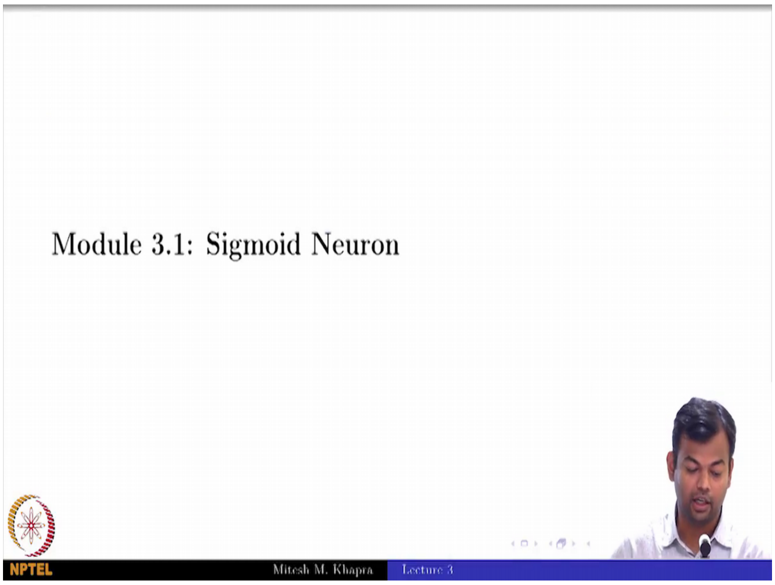
- For Module 3.4, I have borrowed ideas from the videos by Ryan Harris on "visualize backpropagation" (available on youtube)
- For Module 3.5, I have borrowed ideas from this excellent book [a] which is available online
- I am sure I would have been influenced and borrowed ideas from other sources and I apologize if I have failed to acknowledge them

[a] http://neuralnetworksanddeeplearning.com/chap4.html

So, let us start; so, here are some acknowledgments. So, for one of the modules I have borrowed ideas from the videos of Ryan Harris on "visualize back propagation" they are available on YouTube, you can have a look if you want. For module 3.5, I have borrowed ideas from this excellent book which is available, online it is the URL as mentioned in the footnote.

And I am sure I would have been influenced in borrowed ideas from other places and I apologize if I am not acknowledge them probably properly. If you think there are some other sources from which I have taken ideas and let me know I will put them in the acknowledgments, ok.

(Refer Slide Time: 01:02)



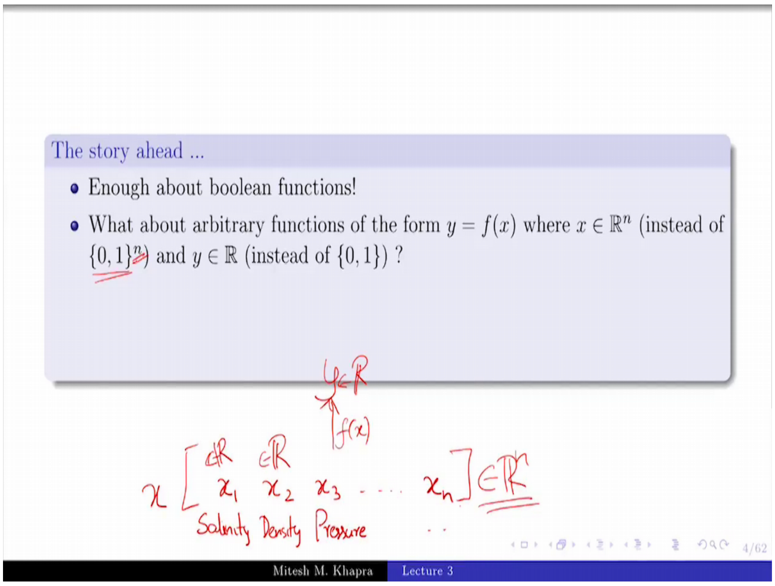
Module 3.1: Sigmoid Neuron

NPTEL

Mitesh M. Khapra Lecture 3

So, with that we will start with module 3.1 which is on sigmoid neurons. So, the story I had is that it is enough about Boolean functions, right?

(Refer Slide Time: 01:10)



The story ahead ...

- Enough about boolean functions!
- What about arbitrary functions of the form $y = f(x)$ where $x \in \mathbb{R}^n$ (instead of $\{0,1\}^n$) and $y \in \mathbb{R}$ (instead of $\{0,1\}$) ?

$y \in R$
$f(x)$
$x \begin{bmatrix} \in R & \in R & & & \\ x_1 & x_2 & x_3 & \dots & x_n \end{bmatrix} \in R^n$
Salinity Density Pressure . .

Mitesh M. Khapra Lecture 3

4/62

Now, we have done a lot of Boolean functions, but now we want to move on to arbitrary functions of the form y is equal to f of x; where x could belong to R n and y could belong to R. So, what do I mean by this? So, let me just explain this with the help of an example. So, I will again go back to our oil mining example oil drilling example; where we are given a particular location say in the ocean and we are interested in finding how much oil could I drill from this place, and that is what I would base my decision alright whether I want to actually invest in this location or not.

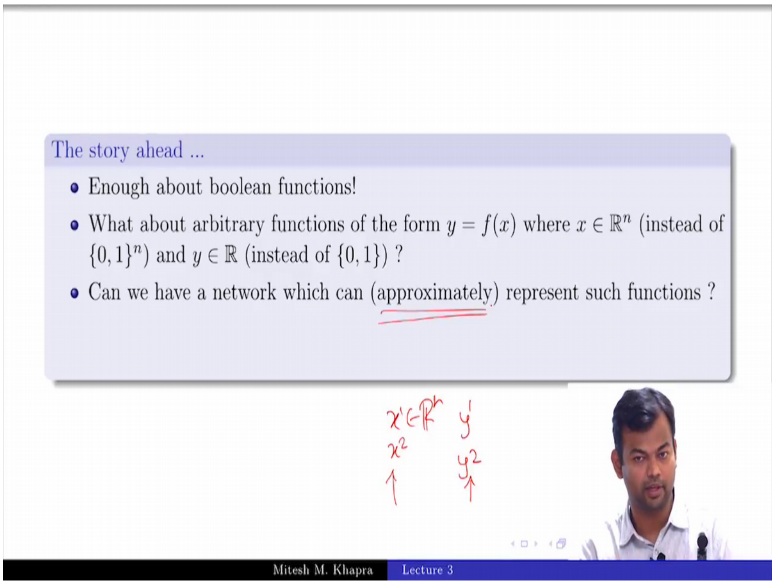
And then what we are saying is that this could depend on several factors. So, we could have x 1, x 2, x 3 up to x n, right where this could be the salinity of the water at that location. So, this could be a real number, this could be the density of the water it is average density. This could be the pressure on the surface of the ocean bed and so on and so forth, right?

So, each of these values independently belongs to the set of real numbers, right? So, each of this is a real number and we have n of these. So, together they belong to R n right. So, I can read that I have n such real numbers, and I could just put them in a vector and say that I have a input x which belongs to R raised to n, ok.

So, we have this x which we can say belongs to R n. And in this particular case, we want to predict y, we want to take this as an input and predictor y, right? And what is y in this case? You want to predict the quantity of oil that we could mine. So, what does R y belong to again a set of real numbers, and it could be some gallons or litres or kill of water right. So, this again belongs to R. So, these are the kind of functions that we are interested in now.

We want a function which takes us from I am having this x, which belongs to R n right it is a vector of dimension n, and takes us to a value belonging to R right. So, you clearly see that this is different from the case when we had n variables each of this was just Boolean, right. So, these were only 0 one inputs now we have real inputs, and these are the kind of functions that we are interested in.

(Refer Slide Time: 03:35)



The story ahead ...

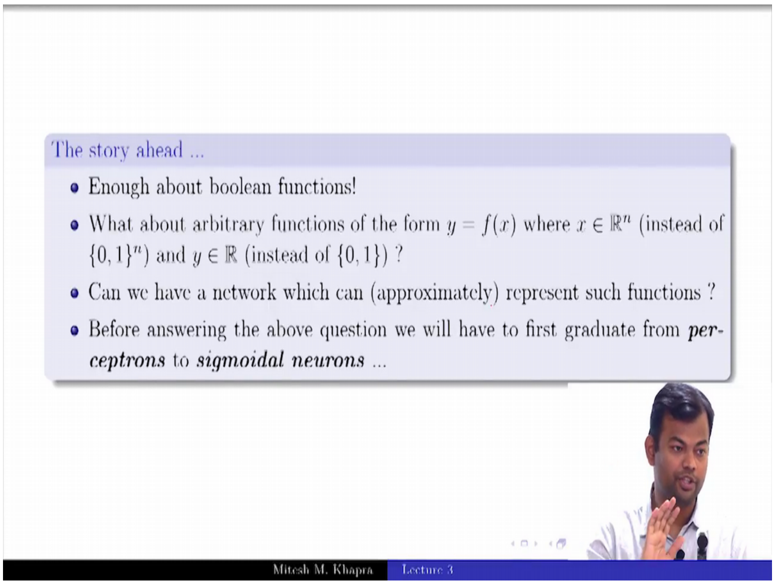
- Enough about boolean functions!
- What about arbitrary functions of the form $y = f(x)$ where $x \in \mathbb{R}^n$ (instead of $\{0,1\}^n$) and $y \in \mathbb{R}$ (instead of $\{0,1\}$) ?
- Can we have a network which can (approximately) represent such functions ?

Now, can we have a network which can represent such functions? Now, what do I mean by represent such functions? We already spoke about this when we were doing Boolean functions, ok. So, what do we mean by representing the function? We mean that if I am given a lot of training data, right so, I am given these x 1 x 2 each of these belongs to R n, right? And I am also given the corresponding labels. Now I want a network which should be able to give me the same predictions as is are there in my training data.

So, it should be able to take any of these x is as input, and it should give me the same y I corresponding to it. And I am saying approximately which means I am with some error rate, whether if it is within some to with as long as it is close to the actual value I am fine with it. So, that is what I mean by a network which can represent such functions, is that working definition of representing clear? Right, so, that is a very similar to the definition that we were used for Boolean functions, right? We had said that we should be exactly be able to get the truth table the network should be able to represent the truth table exactly.

So, that is very similar to the definition that I am using here, ok.

(Refer Slide Time: 09:45)



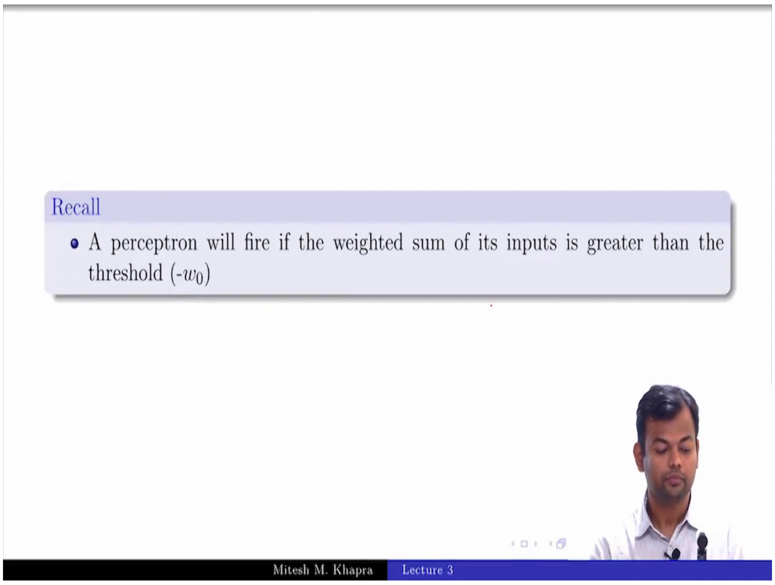
The story ahead ...

- Enough about boolean functions!
- What about arbitrary functions of the form $y = f(x)$ where $x \in \mathbb{R}^n$ (instead of $\{0,1\}^n$) and $y \in \mathbb{R}$ (instead of $\{0,1\}$) ?
- Can we have a network which can (approximately) represent such functions ?
- Before answering the above question we will have to first graduate from ***perceptrons*** to ***sigmoidal neurons*** ...

And then before we do this, right before we come up with a network which can do this for arbitrary functions, we have to graduate from perceptron's to something known as sigma neurons. So, please remember this overall context that we dealt with a lot of Boolean functions, we analyze them carefully and we saw that we could come up with these networks which could represent arbitrary Boolean functions, right?

And they could represent them exactly as long as we have one hidden layer. Of course, the catch was that that hidden layer could grow exponentially. Now we want to graduate from Boolean to real functions; that means, you have a real input of n variables, and one or more outputs and you should be able to represent this exactly right. So, that is where the transition is where so, that is the story that we are looking for, ok.

(Refer Slide Time: 05:30)



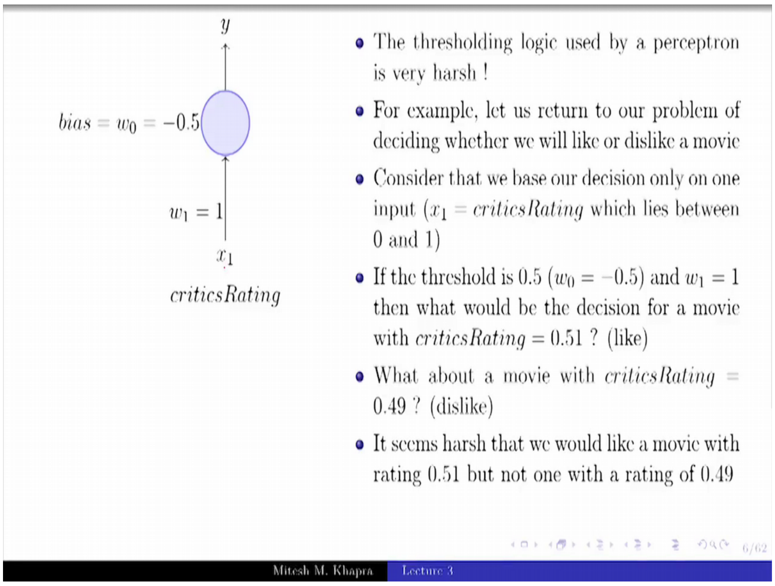
Recall

- A perceptron will fire if the weighted sum of its inputs is greater than the threshold (-w_0)

Mitesh M. Khapra Lecture 3

So, let us start so, recall that a perceptron will fire, if the weighted sum of it is inputs is greater than the threshold, right? Just recall that fine.

(Refer Slide Time: 05:38)



- The thresholding logic used by a perceptron is very harsh !
- For example, let us return to our problem of deciding whether we will like or dislike a movie
- Consider that we base our decision only on one input ($x_1 = criticsRating$ which lies between 0 and 1)
- If the threshold is 0.5 ($w_0 = -0.5$) and $w_1 = 1$ then what would be the decision for a movie with $criticsRating = 0.51$? (like)
- What about a movie with $criticsRating = 0.49$? (dislike)
- It seems harsh that we would like a movie with rating 0.51 but not one with a rating of 0.49

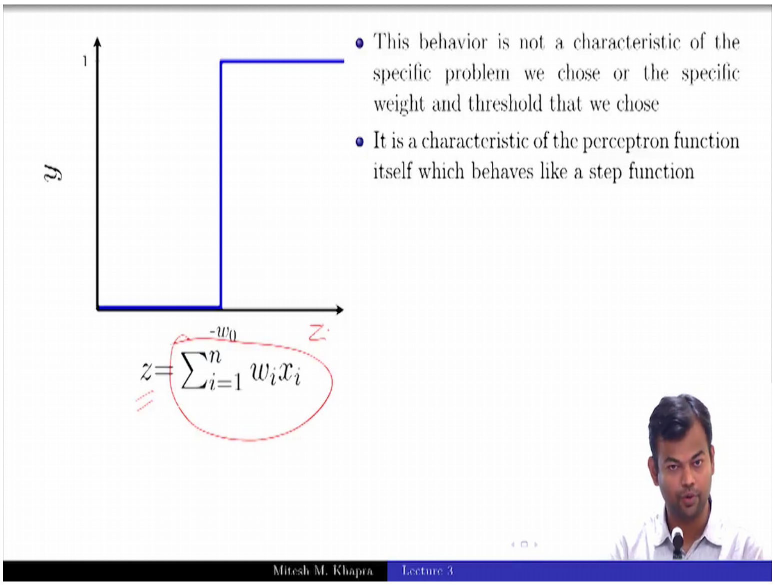
6/62

Mitesh M. Khapra Lecture 3

So now, I claim that the thresholding logic which is used by a perceptron is actually very harsh. Now what do I mean by that? Let us see. So, let us return to a problem of deciding whether we like or dislike a movie, right. That is the same problem that we have been dealing with. And now consider that we base our decisions only on one input; which is the critics rating which lies between 0 to 1, ok. And this is what my model looks like. It takes the input as the critics rating, I have learned some weight for it, and my threshold is 0.5,. What does this mean? It means that if for a given movie the rating is 0.51 will it predict like or dislike like. So, then I should go and watch the movie, what about a movie for which the critics rating is 0.49, dislike. So now, you see what I mean by harsh, right?

So, both these values are very close to each other, but for one I say I like it, for the other I say that I would not like it, right. So, it is not how we make decisions, right you would have probably said something equal for both the movies, right you would have not given such a drastic decision.

(Refer Slide Time: 06:52)

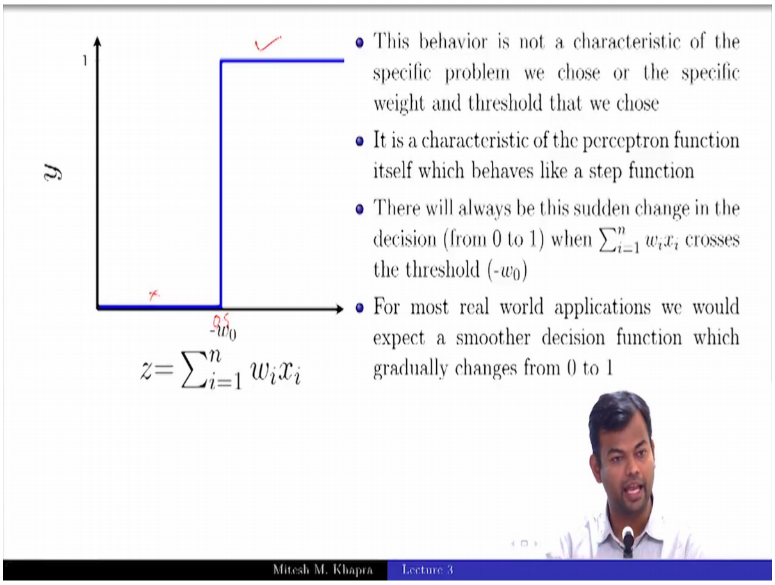


- This behavior is not a characteristic of the specific problem we chose or the specific weight and threshold that we chose
- It is a characteristic of the perceptron function itself which behaves like a step function

So, why is this happening? So, you might say oh this is a characteristic of a problem that you have picked up, maybe that is the critics rating which is between 0 to 1 or something, but I want to convince you that this is not a characteristic of the problem that I have picked up. But this is something to do with the perceptron function itself. So, this is what the perceptron function looks like, right. So, this sum of all the inputs the weighted sum of all the inputs I am calling it by a quantity z, right? And this is what I am going to plot on the this axis, so, this is my z axis, ok.

Now, what does the perceptron say that? When this value of z becomes greater than w naught or minus of w naught it will fire, and when it is less than minus of w naught, it will not fire that is what it says. So, this is a characteristic of the perceptron function itself it is going to have this.

(Refer Slide Time: 07:43)

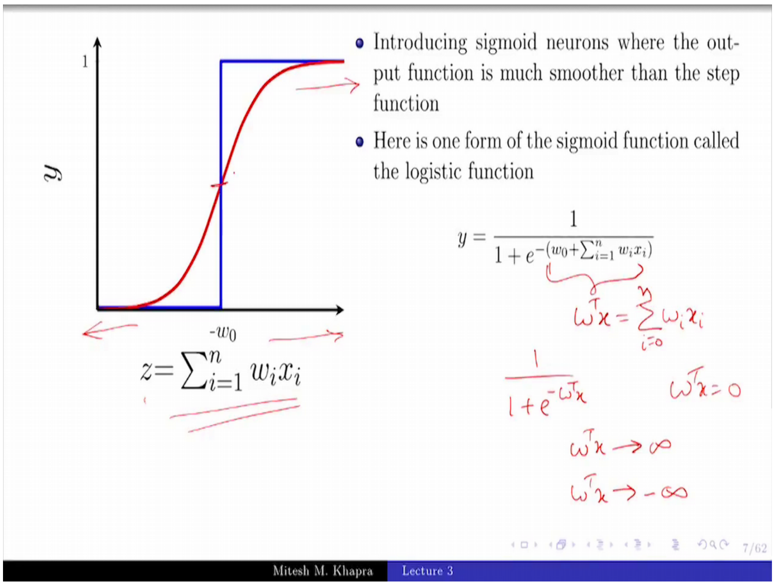


- This behavior is not a characteristic of the specific problem we chose or the specific weight and threshold that we chose
- It is a characteristic of the perceptron function itself which behaves like a step function
- There will always be this sudden change in the decision (from 0 to 1) when $\sum_{i=1}^{n} w_i x_i$ crosses the threshold ($-w_0$)
- For most real world applications we would expect a smoother decision function which gradually changes from 0 to 1

Mitesh M. Khapra Lecture 3

Sharp decision boundary that whenever your sum crosses this threshold you will say 1, and whenever your sum does not cross this threshold you will say 0. So, in this toy example over the movie critics it just happened that this was 0.5. And so, it was saying yes for 0.51, and it was saying no for 0.49 right. So, this will happen for any problem that you pick up, ok.

(Refer Slide Time: 08:06)



- Introducing sigmoid neurons where the output function is much smoother than the step function
- Here is one form of the sigmoid function called the logistic function

$$y = \frac{1}{1 + e^{-(w_0 + \sum_{i=1}^{n} w_i x_i)}}$$

$$\omega^T x = \sum_{i=0}^{n} \omega_i x_i$$

$$\frac{1}{1 + e^{-\omega^T x}} \qquad \omega^T x = 0$$

$$\omega^T x \to \infty$$

$$\omega^T x \to -\infty$$

Mitesh M. Khapra Lecture 3

So, to counter this we introduce something known as sigmoid neurons, and this is just a smoother function or a smoother version of the step function, you see that, ok?

How many if you know what a sigmoid function, what is the formula for a sigmoid function? Quite a few good, and here is one such sigmoid function which is called the logistic function. So, remember that sigmoid is a family of functions, these are functions which have this s shaped, logistic function which I have shown here is one such function and the other function that we will see in this course is something known as the tan edge function right. So, let me just get into a bit more detail with this logistic function.

I just want you to understand it properly. So, this quantity here remember we were writing it as w transpose x, right? Which was summation i equal to 0 to n, w i x i remember this, right? So now, I am just going to consider this to be 1 over 1 plus e raised to minus w transpose x. Now I am going to ask you some questions and try answering those.

What happens when w transpose x tends to infinity. What happens to the sigmoid function?

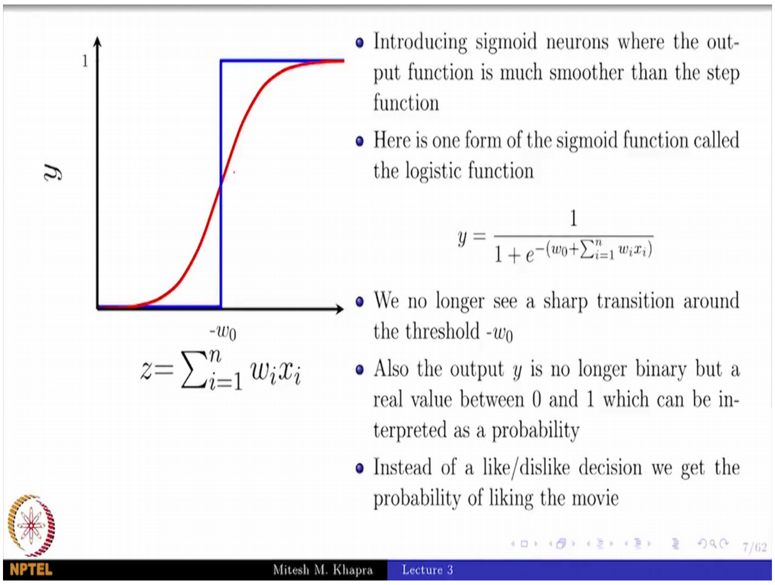
Student: 1.

One and that is exactly what is happening here as this tends to infinity as this keeps growing, right? So, remember this axis is z which is the same as w transpose x, right this is w transpose x, ok. So, as it tends to infinity, your sigmoid goes to 1, what happens if w transpose x is minus infinity.

Student: 0.

0 and that is exactly what is happening here, right. And what happens when w transpose x is equal to 0, half right? So, this is that value corresponding to half, is that clear? Ok so, that is how a sigmoid function behaves fine.

(Refer Slide Time: 10:09)



- Introducing sigmoid neurons where the output function is much smoother than the step function
- Here is one form of the sigmoid function called the logistic function

$$y = \frac{1}{1 + e^{-(w_0 + \sum_{i=1}^{n} w_i x_i)}}$$

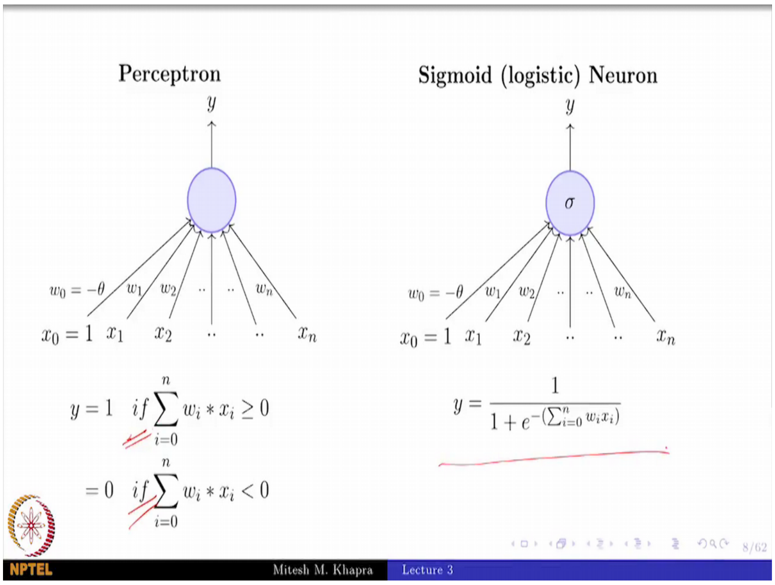
- We no longer see a sharp transition around the threshold -w_0
- Also the output y is no longer binary but a real value between 0 and 1 which can be interpreted as a probability
- Instead of a like/dislike decision we get the probability of liking the movie

Now, we no longer see a sharp transition, it is a very smooth function, and the sigmoid function lies between the values produced by the sigmoid function rate, what is the range that they lie between?

Student: 0 to 1.

0 to 1, what is another quantity of interest that you know which lies between 0 to 1? Probability; so, that is one advantage of sigmoid functions. So now, you can interpret the value given by a sigmoid function as a probability, right? So, what does it mean in our movie example again? So, it just tells me in those 2 cases, that with 50 one percent probability I like the movie or with 49 percent probability I like the movie. So now, this is not very drastic or very harsh, right I am not saying yes or no I am not committing myself, I am just giving you a number which is proportional to how much I like the movie. So, it can be interpreted as a probability, ok.

(Refer Slide Time: 10:59)



Perceptron

Sigmoid (logistic) Neuron

$$y = 1 \quad if \sum_{i=0}^{n} w_i * x_i \geq 0$$

$$= 0 \quad if \sum_{i=0}^{n} w_i * x_i < 0$$

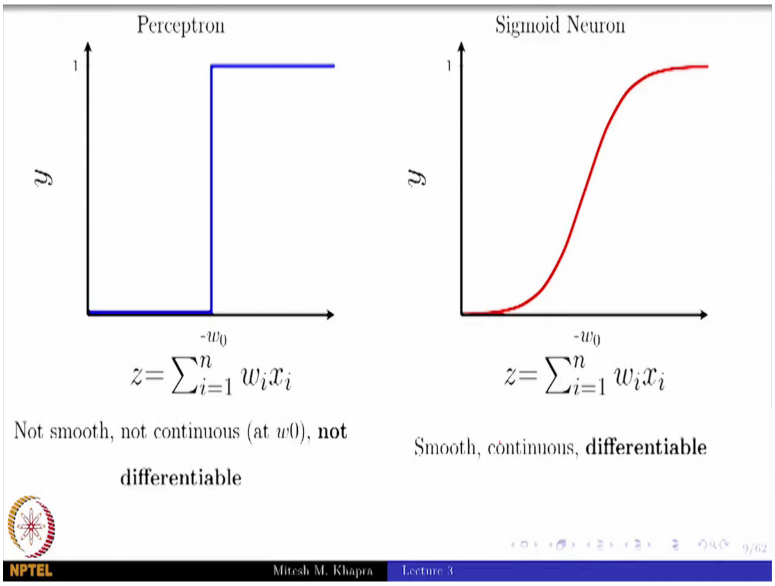
$$y = \frac{1}{1 + e^{-(\sum_{i=0}^{n} w_i x_i)}}$$

NPTEL Mitesh M. Khapra Lecture 3 8/62

Now here’s the overall picture it. So, this is the difference between the perceptron function and the sigmoid function. So, notice that here we had this if else condition, right which was leading to that sharp boundary.

Now, here we do not have that defence condition, we just have a function which is a smooth function, ok.

(Refer Slide Time: 11:18)



Perceptron

Sigmoid Neuron

$$z = \sum_{i=1}^{n} w_i x_i$$

$$z = \sum_{i=1}^{n} w_i x_i$$

Not smooth, not continuous (at $w0$), **not differentiable**

Smooth, continuous, **differentiable**

NPTEL Mitesh M. Khapra Lecture 3 9/62

And here is another picture, so, this is not smooth not continuous and not differentiable, everyone agrees with that? It is not smooth here, right it is not differentiable. Here whereas, this is smooth continuous and differentiable. And the contents that we covered today it will be very important to deal with functions; which are smooth continuous and differentiable, ok.

So, for lot of this course calculus is going to be the hero of the course lot of the things that we do will be based on calculus. And in calculus always if you have smooth and continuous and differentiable functions they are always good right. So, that is why we want to deal with such functions ok. So, with that we end module 1.