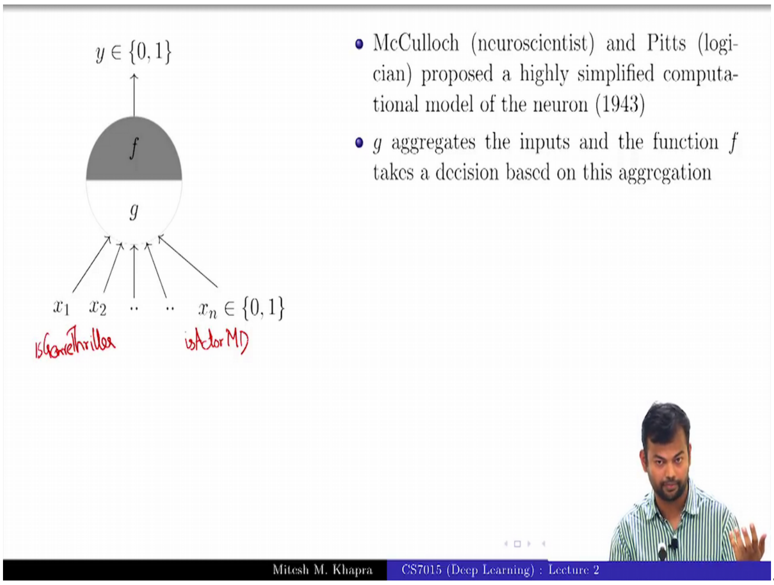
Deep Learning
Prof. Mitesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module 2.2
Lecture - 02
McCulloch Pitts Neuron

Let us start in module 2.2 which is about McCulloch Pitts Neuron.

(Refer Slide Time: 00:17)

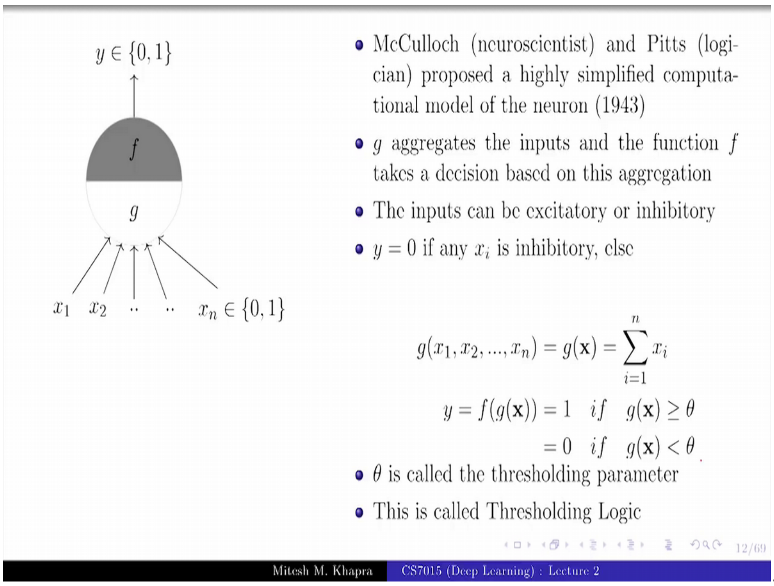


So, as we are done this during the history lecture way back in 1943 McCulloch and Pitts, they proposed highly simplified computational model of the neuron, right. So, now let us see what the motivation is. We know that our brain is capable of very complex processing, it is capable of taking a lot of inputs from various sources and then, helpers take various decisions and take various actions, right. Now, what if you want a computer to do this, right; we want a module which is very similar to how the brain works or at least how we think the brain works which takes a lot of inputs and then, does some processing and helps us take a decision, right.

So, what they proposed is this model which will take a lot of inputs and these inputs are all binary, ok. All these inputs that you see here these inputs are fed to this McCulloch Pitts neuron which is an artificial neuron and it is divided into two parts, right. So, the first part collects all the input. So, remember you had these dendrites which were taking all the information from everywhere, right. So, this just collects all the information and then, the second part sees what this aggregation is, right. I have collected a lot of information from all the sources. Now, the second function will decide what this aggregation is and based on that it will take a decision whether to fire or not, right.

So, the output is again Boolean. If it is 0 the neuron does not fire. If it is 1, the neuron fires, right. So, let us take a concrete example, right. So, suppose I am trying to make a decision whether I should watch a movie or not, right. So, x 1 could be is the genre of the movie thriller. Similarly, there could be another variable say x n which says is the actor Matt Damon, right. So, these are all various such factors that I could take is the director Christopher Nolan, the music given by someone and so on, right. So, all these are probably factors which help me decide whether I want to watch this movie or not, right and you want this neuron to help us make that decision, ok.

(Refer Slide Time: 02:21)



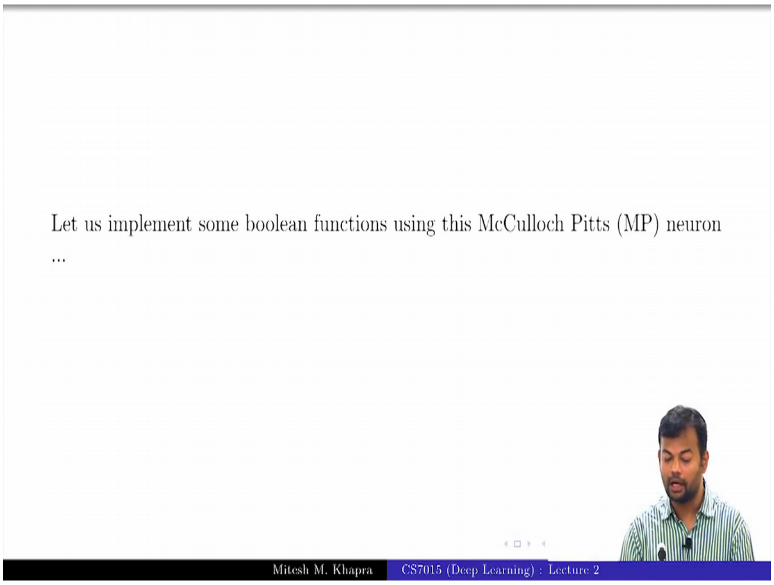
So, now what is happening here is these all inputs right, they can be either excitatory or inhibitory. So, let me tell you what inhibitory is first, right. So, you are taking input from a lot of sources. Now, see one of these sources or one of these inputs is am I ill today? Am I down with fever right? So, if that input is on irrespective of who the actor, director or whatever is, I am not going to watch the movie right because I just cannot leave from my bed, right. So, these are known as inhibitory inputs irrespective of what else is on in your input features. If this input is on, your output is always going to be 0. That means, the neuron is never going to fall a fire, right. So, you could think of it as suppose my mood is not good today, I do not feel like getting up or if I injured my leg or anything right, if any of these conditions is on irrespective of what the other factors are, I am not going to watch the movie, right.

So, that is an inhibitory input and excitatory input are on the other hand is not something which will cause the neuron to fire on its own, but it combine with all the other inputs that you have seen could cause the neuron to fire and how. So, this is how, right. So, these are all the inputs that your neuron is taking. All I am going to do is I am going to take a sum of these, right. I am going to take aggregation of all of these. So, what does this count actually give me the number of inputs which are on, right the number of inputs which are value 1. That is all. This aggregate, this is a sum of all the one's in my input, right.

Now, this is what g does. This is a very simple function is taking a sum of my inputs. Now, the function y takes this as the input. That means, it takes this sum as the input and if the sum is greater than a certain threshold, then it fires. If the sum is less than the certain threshold, then it does not fire, right. So, again see what is happening here is it is same as now if you depend on the actor, director, genre and so on and you fine, ok. At least two of these three conditions are satisfied. At least I am happy with the actor and the director even though the genre is not something that I care about.

I will watch the movie, right or you might be a very niche go movie watcher who only goes to a movie if the actor matches your requirement, the director matches your requirement and the genre and the music and everything matches your requirement, right. So, you are threshold. In that case, it should be high. So, this is how it is going to help you make decisions, ok. Now, again a very simplified model and this is theta is called the thresholding parameter that is the value which decides whether the neuron is going to fire or not and this over all thing is known as the thresholding logic. So, this is what a McCulloch Pitts neuron looks like.

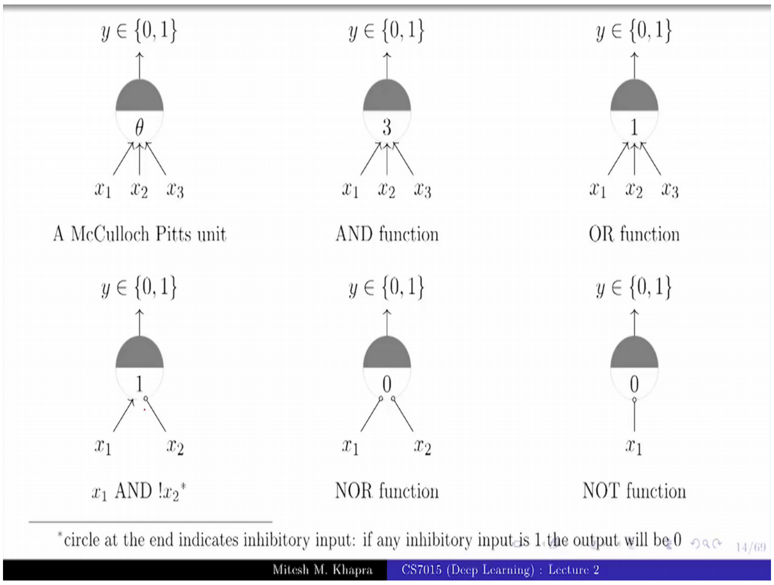
(Refer Slide Time: 04:52)



Now, let us implement some Boolean functions using this m p neuron. So, from now on I will just called it m p neuron and we will try to implement some Boolean functions using it. So, now why are we interested in Boolean functions? It is because we have only simplified the way we take decisions, right. We are saying that the way we take decisions is we take a lot of Boolean inputs is actor Matt Damon and genre thriller and so on and based on that we produce a Boolean output, right.

So, an input is all Booleans. So, we have x 1 to x n which are all Booleans and your output is also Boolean, right. So, that is a Boolean function that you are trying to learn from x to y. Is that clear? You have x just happens to contain n different variables here, and lot of decision problems you could cast in this framework. You can just imagine right whether to come for lecture today or not. Again is you could cast in it depending on various Boolean inputs, right.

(Refer Slide Time: 05:43)



This is a very concise representation of the McCulloch Pitts neuron. What it says is it takes a few Boolean inputs and it has certain threshold. If the sum of these inputs crosses this threshold, then the neuron will fire otherwise it will not fire. That is the simple representation of the m p neuron. Now, suppose I am trying to learn the function, when would the function fire?

All the inputs are on. So, what should be the value of the threshold in this case?

Student: 3

3. Everyone agrees, right. What about other function?

Student: 1

1. Let us see a few more this function. So, let me tell you what this function is, right. So, you see this circle here, right. So, that means that this input is an inhibitory input. If that is on, then the neuron is not going to fire, right. That is how I am representing it. So, now tell me what should the threshold for this be. It is not so hard.

See if x 2 is on, it is not going to fire, right. So, you have four rows; 0 0 0 1 1 0 1 1. So, two of those are ruled out and it is not going to fire. Now, out of the remaining two, when do you wanted to fire?

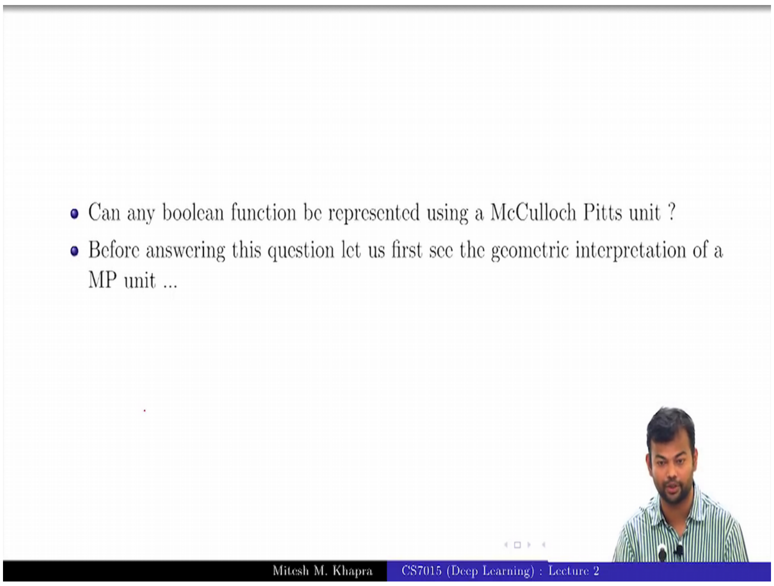
Student: 1

So, what should be the threshold?

Student: 1

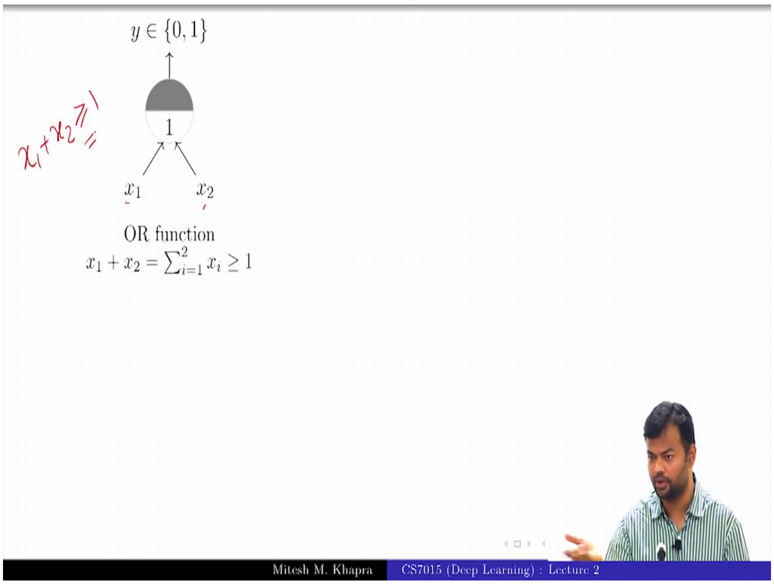
1 everyone gets that. Anyone who does not get that? Ok good. Now, what about this function 0 or 3. 3 is not even a valid option, 0. Everyone agrees to that fine and what about this? 0, ok. So, you get this? So, now if you have a certain number of input variables and the function that you are trying to model the decision that you are trying to make is a Boolean function, then you could represent using these MP neurons whether all Boolean functions can be represented in this way or not that is still not clear. I am just showed you some good examples. We will come to the bad examples later on, ok. Here is the question, right.

(Refer Slide Time: 07:37)



So, can any Boolean function be represented using a McCulloch Pitts neuron? So, before answering this question, we will see a bit of a geometric interpretation of what MP neuron is actually trying to do, ok

(Refer Slide Time: 07:49)

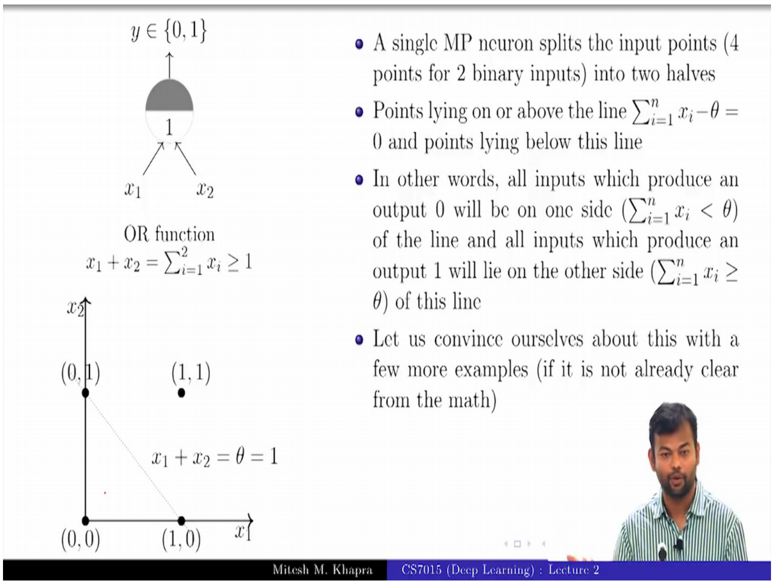


So, let us take OR function where you have two inputs x 1 and x 2 and this neuron is going to fire. If x 1 plus x 2 is greater than equal to 1, right. That is clear. That is how the definition is. Now, if you look at this right x 1 plus x 2 greater than equal to 1. Now, let us ignore the greater than part first. So, we will just talk about x 1 plus x 2 equal to 1. What is this equation of a?

Student: Line

Line everyone gets that. Now, in this case since we are dealing with Boolean inputs and we have two access x 1 and x 2, how many input points can we have ? 4, right 0 0 0 1 1 0 1 1, right.

(Refer Slide Time: 08:36)



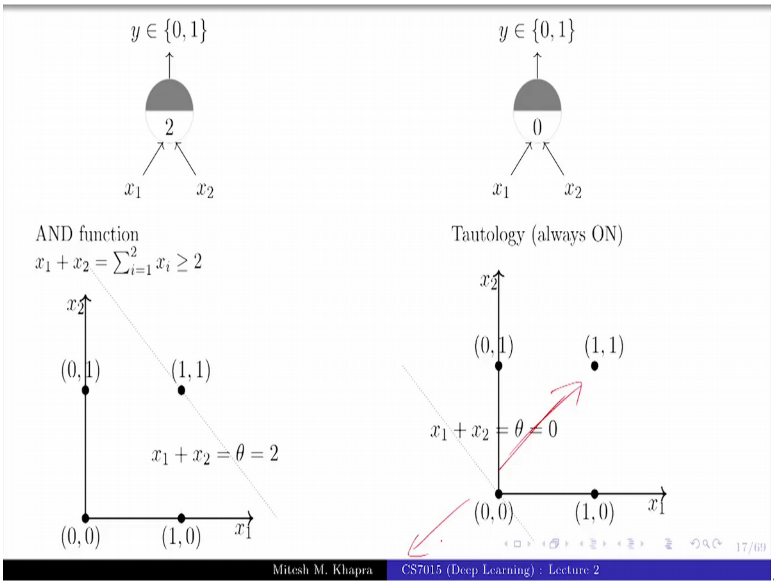
So, you could have these four points. So, just note that this is an x 1 and x 2 axis, but only four inputs are valid here, right. This is not a real numbered access, ok. This is only Boolean inputs possible here, ok.

Now, what is the line? X 1 plus x 2 equal to 1 tell you which line is that.

So, one which passes through 1, 0 here and 0, 1 here, right; so, this is that line ok; Now, what do we want that for all those inputs for which the output is actually 1, they should lie on the line or on the positive side on the line, right and all those inputs for which the output is 0, they should lie on the other side of the line. Is that happening? So, what is actually MP in unit actually learning linear decision boundary, right. It just what it is doing in effect is actually it is dividing the input points into two halves such that all the points lying on that line right are, sorry all the points for which the input should be 0 low below this line and all the points for which the output should be 1 right. Sorry in both cases, it should have been output.

So, let me just repeat it. All the points for which the output is 0 low below this line and all the points for which the output is 1 either lie on this line or above the line, ok. Is that fine? So, let us conveyance ourselves about this. Even it is not already clear from the equation. For how many of you it is already cleared from the equation that this is exactly what it does for a large number of period, but still we will just do a few examples and move ahead.

(Refer Slide Time: 10:15)



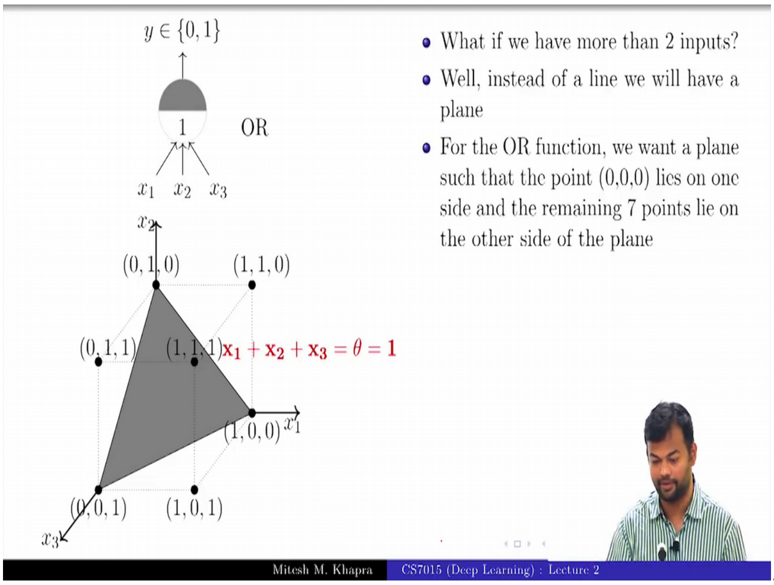
Now, for the function what is the decision boundary? It is x 1 plus x 2. No, that is the decision boundary.

Equal to 2, right; so, again I have these four points. Only these four points are possible. Now, where is my decision line?

Passing through that 1, 1 and intercepting this somewhere around 2, 0 and this around 0, 2 right. So, that is the line which I am interested in, ok. Now, again do you see that our condition is satisfied that all the inputs for which we want the output to be 1 are on or above the line and all the inputs for which we want the output to be 0 or below the line, right. Now, what about this function? What is the threshold?

Zero; so, what would the line be? X 1 plus x 2 equal to 0 which passes through the origin, right and again all the points are either on or above the line, right. So, this part we are going to call as a positive half space and this we are going to call as the negative half space. So far everything is fine.

(Refer Slide Time: 11:24)



Now, what if we have more than two inputs? In a two dimensional case, when we just had x 1 and x 2 we are trying to find a separating line in the three dimension case. What will we do?

Student: Play.

Play good and in the higher dimensions?

Student: Hyper Plane

Hyper plane very good,. So, this is now your three dimensional case, right. Again there are three axis here, but not all points are possible. How many points are possible? 8 points and which is the function that we are trying to implement.

Student: OR

So, for these eight out of these eight points, for how many is the output 1?

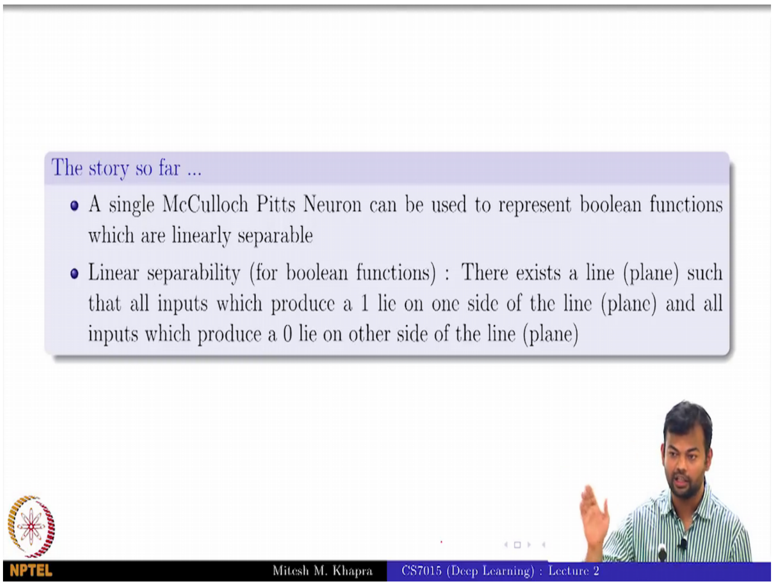
Student: 7

7 and for 1, it is 0. So, what is the kind of plane that we are looking at? We are looking for a plane such that 7 points lie on or above it and 1 point lies below it and which is that point.

Student: 0

Zero, zero, 0, right; so, now what is the equation of that hyper plane? X 1 plus x 2 plus x 3 is equal to 1, good. You see this. So, you see that all the seven points are visible, but the points 0, 0 is not visible because it is on the other side of the plane, right. So, this is doable in three dimensions also and again in higher dimensions also, right we could find in hyper plane.

(Refer Slide Time: 12:38)



So, the story so far is that single McCulloch Pitts neuron can be used to represent Boolean functions which are linearly separable. So, a linearly separable function is such that there exists a line such that for that function whichever points produce an output of 1 lie on one side of the line and whichever points produce an output 0 lie on the other side of the line. This is a very informal definition. Later on we will try to make it more formal. So, is that fine ok? So, this is where we will end the previous module.