Now think of a situation where let us say suppose I have to model you know an XOR gate right the functioning of an XOR or XOR right. Now for an XOR right what do you have I mean let us say suppose I go back to this example X1, X2, Y what is the XOR thing 0 0? 0 I thought right 0 0 is 0, 0 1 is 1, 1 0 is 1 and 1 1 is 0 right this is the XOR right. Now if you again plot it okay the same way okay as we did before now suppose I say that right my 0 0 I am going to indicate with a 0 because the output label is a 0 then 0 1 right is somewhere here which I will indicate with a cross because my label for that input is 1 okay. So, this is X1 again okay this is X2 sorry okay 0 1 is here okay 0 1 is here okay. So, this is 0 1 this guy is 1 0 and 0 1 and 1 0 I am going to indicate by a cross because the label for that is 1 right. So, I am going to indicate this as 1 and this is 1 0.

So, also has a value 1 and then 1 1 right is out here and that I am going to indicate by a circle because the output corresponding to that output label is 0. So, now right if you ask with a with a and and of course right I mean you know remember that that an extension to a higher dimension for example right from on a on a 2D plane it looks like a line will line will do the separation and a higher this one it will be a hyper plane and so on right. So, this is the simplest that you can illustrate right on a on a this one. Now if you think about this right I mean whichever you way you try to draw a line right I mean you would not be able to do a classification right because because suppose I suppose I try drawing a line okay suppose I try drawing a line like that then then again I am not able to classify because because right along with this guy there is another label which is right 2 2 kind of different labels are coming in on this side.

Suppose I draw like this then again there is a problem because on this side there is there is again a confusion if I go there then again there is a confusion because there is one label out there. So, whichever way you try right whatever you try I mean try like that whichever you try right with a single line you would not be able to solve this right and that is where I think you know that is where people sort of you know got got a little whatever right I mean you know I mean they kind of understood that right this is not so great and okay right this you can also show that right this is a condition that you cannot satisfy in any case. So, if you just go back and and try to do the same thing okay which we did before just just write down the conditions right what do you have so let us say w1 0 and the plus w2 0 what do you say is less than theta right this is what you want then you have w1 0 plus w2 1 is greater than or equal to theta then you will say w1 1 okay reality that one does it solve one does not solve it like this okay there is something called the perceptron learning rule but for the time being right we will just do brute force so w1 into 1 plus w2 into 0 this will begin be greater than or equal to theta and then w1 1 plus w2 into 1 will be less than theta right this is how you want the labels to be. Now which means that theta is of course greater than 0 from the first condition the second one says that w2 should be what is it greater than or equal to theta right and then the say the third condition says that w1 should be greater than or equal to theta but in the last condition says w1 plus w2 should be less than theta right and this you cannot satisfy right which means even that is also evident I mean which is a look at it graphically. So here is where right came sort of a lull period right for a perceptron because people felt that well I mean right you can probably do only some things with it right and to sort of right go kind of further right I will just I will introduce what is called a general form of a perceptron okay this will just make the math a little more easy I mean in terms of matrix vector notations.

So a general form of a perceptron okay how does that look so we again kind of go back to the same condition let us say that y is equal to 1 if summation wi xi but right note this index okay I am making it run from i equal to 1 to n greater than or equal to theta is 0 otherwise okay and so what we can do is right this can this we can alternately write it as summation wi xi i equal to 1 to n minus theta is actually greater than or equal to 0 okay. Let us write down a vector w right to be this let us write this as w0 w1 w2 all the way up to wn so this w1 to w1 are the same way that we are all sitting in that original summation there is this w0 that have additionally introduced and then for x right I am going to write this as 1 okay x0 let me write it as x0 x1 x2 all the way up to xn where x1 to xn are again the inputs and then x0 is something additional right which have which have which which right we have introduced. Now if you really look at this expansion right so what you have is suppose I wanted to write this right in terms of so the idea is that right you want to be able to express this entire operation as w transpose okay x is actually greater than or equal to 0 for your y to be 1 and right y is 0 otherwise right. So I mean you do not want to kind of mess with this theta and all separately right you just want to put them all together. So now what should be your I mean right if you want to rewrite right this okay this one in this form okay so what do you think should your w0 and x0 be I mean what should be w0 and what should be x0? x0 is 1 right I mean okay that is all because you see your w should become the unknown okay always I mean right you do not want to mix up I mean you might argue can I not put you know that is the theta under x0 that is not the idea.

The idea is that you want to keep the w as the unknown right it contains the bias and the weight so right that becomes your unknown vector then x will not have any unknowns x0 will become 1 okay so if you think about it this will become 1 and w0 will then be minus theta right so this w0 is actually equal to minus theta which is what we call as a bias right so we can also call this as a simple term b instead of trying to remember some minus theta and all that think of it as a some quantity that is b and right what you are sort of right effectively saying is that solving this problem is equivalent to solving something like right w transpose x greater than I mean find a w such that for that rule right that you have for example this x1 x2 y kind of rule you should find this w which involves the bias as well as the weights right and this is going to be the case whatever be it right whether you have a million neurons connecting you know with each other and so on except that there the scale will be much higher right you are going to look at millions of these weights and so on and here you know it is at a very very what you call you know at a very very small scale but the idea is still the same right you still have to compute these unknowns the weights and the bias okay that will always be the case and irrespective of whether you are solving a regression problem or you know or right whether you are doing a classification problem does not really matter eventually it will all boil down to computer what will be the loss function could change right depending upon whether you are doing a classification kind of a problem or whether you are going to do a regression kind of problem okay. Now right what happened was then right then it was actually shown that okay then right what followed was actually was actually an interesting exercise right where they showed that instead of instead of just having 1 neuron right and then of course you know being able to show that you cannot use that to model as something like a non-linear sort of a classification if you think about it the 1 line you are not able to do it right so they said right why do not you why do not you why do not you have right extra this 1 right I mean why do not you introduce more number of say neurons to solve the problem but then how do you introduce them and where do you introduce them right that is where this notion of a hidden layer right came okay. So this notion I mean if you if you think about think about think about this right I mean so these are the layers so for example right when you say that you have a deep network right what you really mean is you have got several of these layers right you know coming okay in between. Now so what was done was you again sort of have so right one of the simplest things right that you can do is okay you have the inputs again okay your X1 and X2 I am going to solve the same XOR problem now but now what has been done is right you have introduced an additional layer in between so to say okay what by layer I mean I mean right this thing right which is come in between and then again you have an output neuron. So earlier right I mean you had these 2 inputs you had this output neuron right that is the only thing you had now in between okay you have added a layer of again neurons.

Now you can ask how many neurons you need and so on but right now I am just going to show that right with 2 neurons okay we should be able to solve this problem and right this kind of an uncertainty will always exist I mean for a given problem how many layers do you need how many neurons do you need effectively and all right nobody knows okay. So that is why all these GPUs and all are needed so that you can keep running them and then try to see which loss function fits the best which architecture fits the best. So really there are no easy answers but then but then you know certain kind of what you call this results are available that actually indicate what these things can do okay and I will actually talk about them right as we go along but for the time being right just kind of think about being able to again solve the XOR problem but now right by actually introducing 2 more neurons in between okay what we call is an say additional layer and you know whenever somebody talks about a deep network with a certain number of layers for example this one we would not call it a 3 layer network we will call it a 2 layer network because the hidden layer see the layers are the places where there is a computation going on. See X1, X2 is just an input right you are not really computing anything there whereas in the middle one we will compute something at the output neuron we will actually compute something so in that sense right we will call this as input layer this will be the hidden layer this will be the output layer now you can have bunch of these you do not need just one hidden layer many a time you will have many many hidden layers coming in between and all that along with the output layer is what will sort of tell you what is the kind of rate depth of your this one network. So in this case right it has a depth of 2 it is got you know so it is got actually 2 layers right and those are the layers where there is a computation going on okay.

Now let us just label things like you know so let us say that right here we have got theta 1 okay for this let us say we have got you know theta 2 for this and then we have got a theta 3 and like I said right a neuron gets gets right you know inputs from all the all the right previous things. So similarly each neuron should get should get input should get an input should receive you know input from all the all the right previous layers and in this case it is just X1 and X2 so right so basically this guy will get inputs from here then this other neuron will also get inputs from X1 and X2 and similarly this this neuron right will get an input from here as well as it will actually get an input from here and then you have an Y and the Y is still the same old XOR right that you had there okay that is the that is the one that you are still interested in. But now right what we will do is we will just follow a nomenclature okay we will write this as we will call these weights now as instead of calling W1 W2 we will call them as W11 that means from the first neuron or whatever the let us call this as neuron number 1 neuron number 2 neuron number 3 the neuron number 1 to input 1 neuron number 1 to input input 2 that will be W12 okay so that is this that is this weight on this arm okay and then you have like W21 okay that will be the weight that that gets applied to X1 when that goes as input to the neuron number 2 and then this let us call this as W22 okay I think this is just just this one notation. And similarly here we will call this as W31 okay and this will call us W32 and then you have a theta 3 there. So if you if you kind of yeah so so for the for the same XOR right so this we are trying to solve the XOR problem right by by introducing additional bunch of neurons there.

Now I am just I am just going to going to write down these values okay a possible choice for the biases and weights so the biases are theta 1 theta 2 theta 3 or minus of that okay so let us not worry about minus and all that just say that there is a bias that is that will that will relate to this is the theta sitting there and there are the weights which are all these you know interconnections and and you know these things are called fully connected. If you see right this is called fully connected because because you get I mean suppose I had X1 to Xn instead of just 2 Xn if I had X1 to Xn they would all come in so you so you can imagine right so you can imagine how many weights you eventually end up needing. So which is also the reason why later right we talk about a convolutional neural network and so on because that has that is much less weights just like your LTI sort of a sort of a system right where you can have local convolution I mean where you can have a convolution going on right which is which is like which is like invariant right I mean you do not change when you do an impulse response right when you apply it on a signal you do not change it right as you go along the signal it remains the same. Whereas here right you can think about this as being some weighted averaging that is actually changing right every time. So which means that it is computationally much more expensive lot of unknowns involved but that is the way it evolved right.

Now okay now I am just going to say that a possible choice right this is not this is not unique okay a possible choice for the weights for the weights and the biases possible choice could be this the biases is good right this is w1 I mean I just leave it to you to check this okay w11 is 2, w21 is equal to minus 1, w12 is equal to minus 1, then w22 is equal to 2, then the thetas right theta 1 is 1.5 like I said right these do not have to be had to be binary value that anything theta 2 is equal to 1.5 and theta 3 is equal to 1 and all this for still a Boolean sort of you know implementation right. Now okay now what okay now in this table right I will just write down so X1, X2 and then summation let us say what is this w1j Xj, w1j Xj j equal to 1 to 2, then I am going to write w summation w2j Xj, so j equal to again 1 to 2, then I have got Z1, then Z2, then summation w3j Xj, j equals 1 to 2, j equals 1 to 2 and then let us have what happened any doubt what okay yeah you are right w3j, w3j okay what are we calling the output set I have not even called them as whatever yeah I think I have to call them as certain input so I think let us call this as Z1 yeah you are right I mean I miss that Z1 and Z2 yeah you are right w3j Zj, j going from 1 to 2 and then Y which is let us say Z3 let us call this Z1, Z2, Z3, Y is Z3 okay. So now right if I had my X1 and X2 as 0 right I mean you have 0 0 0 1 1 0 1 1 right so if I had it as 0 0 then what will be right this summation that goes inside the first neuron so w1j so w1 1 is what 2 right so basically 2 anyway right so it is all going to be 0 X1 and X2 are both 0 and therefore this is just 0 what about what about it w okay that will also be 0 and then what will be your Z1 and Z2 all 0 right because your see theta 1 is all like you know is all more than 0 it is all crippled and this one positive therefore Z1, Z2 0 therefore w3j that summation is also 0 and therefore your Y right you know which needs to exceed so to get a Y equal to 1 you need to we need the sum to exceed theta 3 and theta 3 is greater than 0 therefore the output is 0 which is okay that is what we want for 0 0.

Now if you put 0 1 right then then what will happen w11 has no role but then w12 what is that minus 1 so minus 1 into 1 so that is actually minus 1 and minus 1 is being compared with theta 1 that is 1.5 therefore what will be the output so anyway right so I think here here here let us just write the number it is minus 1 what about what about w2j Xj minus 1 into w21 is minus 1 and hey what happened did I miss something w22 I have not written oh no w22 is 2 no okay so right w21 is minus 1 into into 0 that is 0 and then w22 right is actually 2 into X2 that is 2 right therefore the second guy is 2 therefore what do you think will your Z1 be Z1 will be 0 but Z2 right you have to compare 2 with theta 2 right and theta 2 is 1.5 it is greater than and therefore this will become 1 right then what will happen here w3j Xj right so w31 oh I have not told what w31 and w32 are okay w31 is actually 2 and w32 is also 2 okay that is what I have taken here so then what will happen so 2 into 0 it is gone and the other one is 2 into 1 that is 2 right and therefore this sums up to 2 and then this you have to compare with theta 3 that is 1 therefore the output is 1 right which is what you want and similarly 1 0 can can write somebody quickly tell or else I will just fill this up right let us not waste time so 1 0 we just go back and check 2 minus 1 1 0 right and then 2 1 okay and then for 1 1 you get okay so 1 1 right what do you get you have I will write down directly okay 1 1 0 0 0 0 okay you can actually check this out so the point is right so now okay you seem to have been able to get the output right that you wanted this is what you wanted for an XOR right now you can also graphically visualize what is going on so what we will do is you know we will actually look at look at this again let us go back to what we had so we said this and this and then we had this and this right as 1 the crosses are 1s the circles are 0 labels right and and let us kind of right look at look at look at this guy w 1 1 x 1 so w 1 1 is what 2 x 1 plus this is w 1 2 right for the for the first neuron w 1 2 x 2 so that is minus x 2 is equal to let us say theta 1 so theta 1 is 1.5 right and the second equation will be w 2 1 x 1 so w 2 1 is minus 1 so minus 1 so it is minus x 1 plus w 2 2 x 2 right so plus 2 x 2 this is equal to theta 2 and theta 2 is 1.5 okay this is what you get or 2 x 1 is equal to x 2 plus 1.

5 and 2 x 2 is equal to x 1 plus is 1.5 right so let us say that let us let us let us let us find out it what is this line okay so what is it so if I put if I put x 2 is equal to 0 I get x 1 is equal to 0.75 so some point here and then and then if I put x 1 x 1 as 0 then x 2 is minus 1 point correct minus 1.5 right so then then it means that right it is somewhere here and okay right I am not drawing it exactly but then I think about it as some line like that and for the other one right you have another line so which is like if I put x 1 is equal to okay x 2 is equal to 0 x 1 is minus 1.5 so it is somewhere here right and if I put x 1 is equal to 0 then then x 2 is 0.

75 right so it means that it is somewhere here because it because this top cross is what is it 0 comma 1 right so so the other line right and I think I think they actually meet or something right okay the parallel okay so so the point is this right so what you know now find is right with two lines okay now instead of instead of the earlier one at where he had varied only at the only one line right now you have actually two lines now and you can kind of right and which is what which is what has been able to which is what has enabled you to solve the right xor problem now okay so so so so the way to think about it is whatever is lying between the lines is one class and whatever is outside this area right which outside this region that is being that being covered by the two lines right whatever is outside is the other label okay. Now this right along with this came actually you know a result that actually said that said that you know if you I mean here here right for the simple xor problem yeah one more thing right there is just just one more thing right that I that maybe I should just mention in passing is that if you had if you if you had constrained this neuron to be linear right in the sense that right I mean this for example right I mean you know if you if you did not have a nonlinearity in this theta 1 and theta 2 right theta 2 then you can then you can show that right effectively I will just leave that to you as a homework right then you can effectively show that the whole thing becomes like like like a one neuron stuff okay so what so the idea is that without these nonlinearities coming this nonlinearity by which I mean this kind of an activation or a step right that you have which is what is introducing a nonlinearity if you did not have it right you would not be able to solve this right if you simply thought that you know I can just have some have something linear out there which means that right you do not have this kind of an activation simply have some weighted sum there and then you simply have a weighted sum okay under the second neuron but do not have a nonlinear activation then you can show that the whole thing will then boil down to doing just doing just a just a this equivalent to just having one neuron out there which means that you would not be able to solve solve this solve this problem okay. So, so this nonlinearity is very important okay that is what actually you know lends you the strength

to be able to do the classification and when they come together in a certain way like in this case right you brought in another hidden layer in between and you introduced a few neurons and that is what actually helped you solve the problem and well I mean there are there are other offset there are you know see right this is not the only way to do this by the way okay you can also you can also do it in other ways but what I showed you is just one way to do it okay there is there is nothing unique about this okay this is just to just to give a give a window into what is going on okay. And okay and then the idea is that okay yeah right so I think let me just just let us talk about this there is a theorem right that actually that is for a Boolean function and then and then I think then we will go on to I mean so which we just we just say is that what kind of a power right this kind of a representation has. So this theorem right what it says is that you know any this one Boolean function and later it will we will talk about real valued functions and so on any any Boolean function of n inputs n inputs can be represented exactly can be represented exactly by a network of perceptrons network of perceptrons so what you saw was something like a very simple network right just now the perceptrons.

So perceptrons containing one hidden layer okay we will not actually enter into the proof of all this right it is supposed to be a quick review right so one hidden layer with 2 power n perceptrons and output layer containing one this one perceptron and output layer containing one perceptron of course the but then the thing to note is that this is this is a sufficient condition okay the earlier case that we did not require for example we had 2 inputs right so you would think that in the hidden layer I should have had 4 neurons right how did I do it with 2 so this is only a sufficient condition okay and you can you can do you can you can actually do with fewer neurons and the the whole strength is that you can actually do it with with let us say much less neurons okay this is like the upper limit okay and the real strength of these networks comes because of the fact that you do not really have to satisfy this they can do with with with far fewer neurons and still be able to accomplish what you want like we did for that small example right we used only only 2. Now right I mean otherwise otherwise the whole thing will just right explode you know i right it exponentially.