Then right going forward, we just look at yeah let us just kind of look at the activation function. So, now that we have a condition that now that we know that right gradient descent is a way out for us to be able to right minimize. And we still have not talked about you know just looks like we can compute dou L by say dou theta irrespective of how many layers you have and so on. There is an elegant way to do that which is what actually propped up you know all this activity in the say deep network what is called back propagation algorithm which I think we will at least be able to give the schematic today and then next class I think we should be able to you know we should be able to show how it works. But the activation functions are now right are now sort of you know important because these are the ones you know through which you are going to introduce a non-linearity right. So, until now we saw a step kind of thing right what is called actually a heat side heat side signal right.

So, where you had some like 0 and then and then it jumps up to 1. So, you have x and then you write f of x and for x let us say less than 0 it is 0 and f and f of x and when x is greater than 0 it is equal to 1 and that x equal to 0 you can say it is undefined some people might say if it is a Fourier series what would that be if you approximated this by a Fourier you know transform not Fourier series Fourier transform what would you get at x equal to 0 half right. But here right people do not do not worry so much and all right about that some people will say that will say that right at f of you know at x equal to 0 I will take it to be 0 or somebody will take it to be 1 ok. Now, this is this is actually heat side, but then, but then the thing is right because of the fact that you need to be able to take dou f by dou x right along the way and and right and because at this point right at this point it is not differentiable.

What will it be I mean if you if you actually went ahead and can you do f dash of x at 0 if you kind of did that at what would you get delta right, but, but then right delta and all is not something that you can implement in an architecture right and impulse is to just understand mathematically right, but, but not, but not here ok. Now, so which is why this this heat side and all is not used right that we used in the right in the beginning just to make the argument for some simplicity right, but really this is not the one that is used then there is something called sigmoid right which has been around for a long time, but again it is not the one that is most commonly used ok. Initially I think another lot of hype around it, but, but then at all that kind of settled down ok. Now this guy is like f of x is equal to 1 by 1 plus e raise to minus x. So, this is 1 plus 1 by 1 plus e minus x right and what do you have.

So, so then right I mean if you if you plot this guy right. So, at x equal to 0 it is half right this is x versus f of x. So, at x equal to 0 it is 1 by 2 1 by 2 right. So, it is half and then at x as x tends to right negative you know a large value then this will go down and then as you as you then go up right x becomes a positive quantity you are hitting 1 right ok. This is actually a this is called a sigmoid and then it is nice because you know it gives you all values between right 0 and 1 therefore, it looks like a probability kind of thing right this looks like a nice thing to use because takes values between 0 and 1 and f dash of x right what will that be.

So, f dash of x right what is it. So, minus e power x right you know my no e power x e power x by 1 plus e raise e power x by 1 plus e raise to minus x the whole square right. And then so we can write this as 1 plus e power no minus x right minus x. So, 1 plus e raise to minus x. So, let me just add and add and subtract and then you have 1 plus e raise to minus x the whole square.

So, you get 1 minus no no yeah. So, you have 1 plus e raise to minus x right. So, you have 1 by 1 plus e raise to minus x for this and then minus 1 by 1 plus e raise to minus x the whole square right. So, 1 by 1 plus e raise to minus x is actually f of x right itself that is this guy what happened did I make a make a mistake it is ok no. So, then f of x minus this is what 1 by 1 plus e even.

So, this is f square of x right. So, this is like f of x into 1 minus f of x. So, f dash of x why we are why we are talking about it is because that is what which use gradient descent and derivative and all right we will actually enter enter the picture therefore, we should know what it is. So, if you try to if you try to intuitively see right it looks like a derivative is initially increasing right and then and then after you you hit x equal to 0 it starts to fall off right eventually goes to 0 and on the left again it goes all the way to 0. So, what do you what do you expect I mean it will look like I mean.

So, f dash of x and if you try to see how it will look like and so on that is going to. So, at 0 at 0 what is it 1 minus 0.5 0.5 into 0.5 that is like 2.

25 right. So, at 0 it is 0.25 and then and then it is got to be like that on either side it will go down go down to 0 and then right when the why this matters is because you know when you when you take this when you take this gradient right which is what is involved in the step when you move forward or whatever right during your gradient descent you know this one right iterations. Then if the slope really falls off right then it means that for large values of x right you would not even move much right that is what it is showing you know the gradient is very low. So, if you take if you take an x which is which is which is which is very low which means on the on the other side right negative side it is very low or on the positive side if it is very high it looks like the gradient right for those points are going to be very low. Therefore, it then means that you could you could get stuck somewhere right you would not be able to you would not be able to move forward much because the gradient value itself is very low.

That is also the reason why sigmoid and all right is not really a prominent I mean if you look at look at these look at these hidden layers right you would not find sigmoid and all right in the first initial layers and all if at all probably right they come much later because if you throw them right at the beginning then you will encounter issues of this kind where if your x is very large either way right if the let us say magnitude of x is large you are you are in trouble because the gradient then is small for you know large magnitudes of x. But there was a time right when let us say right people thought sigmoid was right it was it was everything of course, then right that does not mean that people do not use it in the initial I am saying generally not yeah you may find some network where they have done it and then you should not tell me that hey look you said that, but then they are not. Then the next one is what is called tan hyperbolic ok, when tan hyperbolic is what sin h x by cos h x right. So, it is some e power x minus e power minus x by e power x by e power x plus e power minus x ok. This is another sort of an activation ok.

Let me just ask you a question right can you have can you have any I mean right I do not expect a straight forward answer because you know this kind of say tricky tricky question can you have any kind of any kind of non-linear function as an activation. I mean that I am showing a few know I am going to show you sigmoid heapsite we saw right and then maybe tan hyperbolic it will be like minus 1 to 1, but right in general of course, one thing is that I think you know maybe we will again ask this question right after we do this ok. We will just finish this then we will come back then what happens is so as x goes negative all right. So, as x goes negative so you have like minus 1 right. So, as x goes negative that terms are wise whereas, the first first two terms will drop off a numerator and denominator and that x equal to 0 it is 0 by 2 right.

So, it goes like that right. So, it goes like 1 to minus 1 f of x and it goes via 0. So, this is f of x versus x right and that x equal to 0 it is 0 it is 0 and I will just leave it to you as actually and as a exercise that derivative of this I mean. So, if you call this as f of x then f dash of x is 1 minus f of x whole square these are all simple things right you guys know how to do this and again if you try to plot this right f dash then at x equal to 0 it is 1 right and yeah and you know whether you go this way or that way because you are squaring it right. So, it will again go down on either side and you get this.

So, this is your f dash of x again it is also has a problem that is somewhat similar to sigmoid right that means, for large magnitudes of x yeah right. So, slope will be will be small that means, you have a gradient of course, I am not drawing it correctly it is all symmetric right. So, this is right a gradient will then be small therefore, not really a great thing to use. Then the then the third one which is the most commonly used is ReLU this is called rectified linear unit rectified linear unit ReLU I mean R e taken from here L and U taken from there rectified linear unit and this looks like this. So, you have x you have f of x and f of x is actually max of x comma 0.

So, which means that on the positive side you have a slope 1. So, whatever is x right that is that is also the value of say f of x negative side it is simply 0. So, it means. So, it means that right. So, it means it would not even allow a negative values to kind of go forward right because it is simply clamped them to 0.

But in the nice thing about this is that for let us say positive x right it will I mean you do not have a gradient. So, if you try to plot f dash of x what will that look like will be a step right. So, it will be like this and then has a value 1. So, it is f dash of x versus x so on.

So, at. So, at. So, at strictly speaking at x equal to 0 f dash is kind of undefined, but then people just you know take it to be 0 right. So, f dash of x is 1 for x greater than 0 and 0 for x less than or equal to 0 or some people may say the other way again right that is simply because you want to be able to go forward and kind of use it right otherwise I mean you would be stuck. Now, there is also something called leaky ReLU right which means that which means the rate I mean if you do not like the fact that you are you are completely clamping all the values to the positive side. So, what you can do is you can have what is called a leaky ReLU that means that means you let you let something leak through and that will look like this. So, you have a small slope right on this side and then of course, then you have the slope of 1 let us say this is x and then this is f of x right.

So, you. So, if you allow for you know this a positive slope right, but then a small value. So, that. So, that you if x takes some negative values you still allow them to survive right you do not just knock them off and this will be like suppose let us say suppose I take this to be some beta then f of x is max of what is that beta x comma x and of course, if you try to take the gradient. So, this will be like what beta constant here and then jumps up to 1 and this is f dash that is what it will be. So, this leaky ReLU is used quite often.

Now, suppose I suppose I did come back to that sort of a question and then and then you know and then there are some there are some other variations also no problem right we do not have to do right I mean every one of them, but let me ask you this question. So, now, there is see some of them have you know have what are called you know some of these are called squashing functions in the sense that they do not allow for example, if you look at sigmoid and then tan hyperbolic they are called squashing because they do not let the output go beyond let us say minus 1 1 and so on and so that is called squashing. ReLU is not like that right on the positive side you can go as much as you want with the slope 1. The slope is equal to 1 right not x I do not know how I wrote slope to x the slope is 1. Yeah I mean if I write something I mean sometimes I have something in my mind and then I may not write the same thing you know writing there.

So, if I make a mistake let me know now. So, what was I saying. So, now, can we use any kind of a non-linear function as an activation function. Suppose that is a sin can I use cos sin why do not you use that again I mean see the again that is not going to theoretical in the right answer and all right which I have and I have not seen one, but there is there is there is some reason right which seems to be a reasonable reason what do you think what what is the I mean now. So, the idea is that right I mean there is there is there is something about these functions right which is common.

I mean. So, the answer is this right. So, the activation function should be a monotonic function otherwise I mean right you can ask I mean can you not use a sin or cos, but then the problem is the output will start fluctuating right which you do not want again right I am saying right theoretically it is not like somebody has proven that because I in fact, I read somewhere that somebody has shown that sin can all still be used get it I mean right empirical evidence is you cannot somebody say that look I have used it and then I still get something, but normally you do not see it you do not see in sin you do not see sin or you know or you know cosine or something you do not see any of that the right reason being that you you kind of you would write the activation function to be to be a monotonic function. In fact, in fact, in fact, right that is why I wanted to wait for this right before I again went back to the right universal approximation theorem actually the the the original form was was such that right you should have should be a squashing function the original UAT right no though the activation function that you used had to be squashing functions, but then but then it is only a sufficient condition that is why you have relu right it is not a squashing function right relu. Boundary. Yeah, yeah I mean the squashing means you have to kind of bound the boundary value somewhere they cannot just just go on that is relu will allow it to go to go to go to any value right, but then, but I think as recent as I think 2, 3 years ago somebody then they came up with actually a proof for proof for relu that the universal approximation holds, but then before that right if you read the actual theorem it would say that it should be a squashing function right and and then the and the other thing is that regarding the value itself that you wanted the output I think you know yesterday yesterday I was trying to point out that at the output layer whether you should have an activation.

So for example, this is f of x what should it be. So for example, if it is a classification problem very likely that you will have a sigmoid sitting there at the output layer. So, so like I said you have sort of know in between you could have many layers where probably you have relu and all that, but then towards the end if you have a classification problem typically expect a sigmoid to be sitting there. If it is a if it is a regression problem typically expect a expect a relu to be sitting there or it could be a simply a simply a simple linear unit it does not have any other activation it can also happen at the output right again as I said universal approximation theorem says nothing about what should be in the output layer. The activation and all is for the is for this for that single single layer right which is a this one hidden layer it is all about that, but in our examples we took of course, you know activation to be in the output neuron also right we took all that, but, but that is why the output neuron right whether it will be simply linear it can also be totally linear it does not have to have an activation at all.

It can have activation of the type sigmoid, it can have activation of the type relu it does not have to have activations at all also right. So, you can have you can have networks of all all types. So, now, so now, the now the point is right now to sort of go back and say that now I have I have some input right X 1 to X n and then and then I have let us say right hidden layer 1, hidden layer 2, hidden layer 3 whatever I got let us say right 50 layers and then and then out comes my output layer. So, this is my input layer then this is a bunch of hidden layers and I have output layer right. Now, what do I want I want to be able to solve for my weights and and the biases which are all sitting here right.

So, weights and biases are here. So, the biases are sitting in the neurons and the weights are all those interconnections that are going from the output of the previous layer to the input of the right next layer. So, you want to find all of that and you want to be able to able to solve. So, at the output right you have a cost function L theta L of theta where theta is the weight and the biases that is like the whole network weights and biases right. So, so the idea was that how does one solve it then right I mean you know you may you may you may love to have a deep network, but then you should have a way to train it right.

So, so this. So, this back propagation which is basically you know basically a derivative chain rule that is what it is back back propagation algorithm is the one that. So, back propagation right. So, this is 1986. So, so this back prop is actually a systematic way to and you know see these is nobody even even writes down these equations right because if you see the way these things are structured these packages that you just define the cost function it does everything else for you. If you use pi torch or use tensor floor I think, but I think we should at least in our life we do it at least once.

So, that we know right how it works right and. So, the back propagation is what I want to do next. So, so it kind of looks like this. So, as an example I am going to take a simple example, but this example will help right illustrate how those whole back prop works. So, I am just going to take you know a 2 input case and you know the first this one hidden layer and I am trying to take a fairly general case.

So, that we understand I am going to take 4 neurons I think let me make them a little bigger just I am just blowing it up. Then I have a second hidden layer where I have got 3 neurons then I have an output layer where I have got 2 neurons right which is which is. So, that means, right we are not we are not taking a very simple case for something that you can explain in class right this is enough. So, x 1 x 2 and I am going to write down a few things here now. Now, I am not going to show all the all the you know weight connection, but let me say that right this will get kind of say connected to this.

So, this goes there this goes there this goes there and these weights right I mean I will just show for now let me also put something here I will tell you right what these things mean. And then so, this is input layer I am going to call this as l equal to 1 and all these things will matter and this will be l equal to 2 this will be l equal to 3 will be l equal to 4 and I am going to write this thing as Z 1. So, the first neuron I will write it as Z 1 2 because this l is 2. So, each is like this is like Z i l and B i l if you are kind of right think about it like that I am sorry A i l not B i this is A i. So, what this means is this is like A 1 2 and then this will be like Z 2 2 it is not square just Z 2 2 then A 2 2 Z 2 2 A 2 2 then Z 3 2 because this neuron is a third neuron this will be A 3 2 Z 4 2 A 4 2.

Then similarly here this will become Z 1 3. So, this will be like Z 1 3 because l is 3 Z 1 3 first neuron therefore, it will be A 1 3 then second neuron therefore, Z 2 then the superscript is for l Z 2 3 A 2 3 then Z 3 3 A 3 3 what will be the fourth one then. So, we should write it as Z 1 4 A 1 4 Z 2 4 A 2 4 as long as you can understand this sort of a notation we are fine. And then let us call this as output let us call this as Y 1 hat let us call this as Y 2 hat and then let this wait. So, this is the first neuron. So, let us call this as W now each of these branches weights will be like W i j l and B the bias we will indicate it as B what is the B j l no B i l B i l i because i is i is the first neuron second neuron and so on.

And you know what is then what will typically read people write is a plus 1 here there will be a plus 1 here plus 1 here why do you think you need that plus 1 because there is a bias term no. So, these weights will get weighted you have W 1 X 1 plus W 2 X 2 plus W 3 plus theta 1 and that is the theta 1 is this is plus 1 that means there is actually an arm going from what to say I mean you I mean see there is a there is a let me show it by a by different color. So, I mean you have like a one arm going like this plus. So, which means that you have B 1 say for example, it is l equal to 2. So, this bias will be B 1 2 this bias will be B 2 2 the bias here for this is a neuron will be will be will be B 3 B 3 2 and then B 4 2.

So, the Z is actually the Z is actually that that linear weighted combination plus the bias means Z is like you know summation W i j X j plus theta plus B B j anyway I mean we will revisit you do not have to worry too much, but I hope you understand this plus 1 is to just indicate the bias, but we would not show these arms going because you know then it will make it very complicated. So, I am just going to remove, but just believe remember that everywhere this plus 1. So, the plus 1 will mean that the bias for the for the for the neuron right going ahead and then. So, in that sense right this will be like you know W 1 W 1 1 1 this will be this is a second neuron first right input from the input from the first input. So, write W 2 1 this will be like W 3 1 1 this will be like W 4 1 1.

So, I am just following the same this one notation that we used earlier we are not changing that then what will be what will be the weight here then how will you how will you how will you how will you write this. So, this should be like W i j l. So, what will be the first weight W now W 1 1 2 right and this will be like this will be like W 2 2 2 and so on. So, you should understand now I mean if I had if I had given the other weight then it will have been W 2 1 2. So, I mean something from here to there it would have been W 2 1 2 right.

So, as long as you understand the notation right it is fine then finally, right I am just going to write something as delta delta 1 2 that is right. So, delta 2 2. So, all this like you know delta i l. So, this is like delta i l. So, I will stop here and then this would be like delta 3 2 will be like delta 4 2 and similarly this would be like delta 1 3 delta 2 3 delta 3 3 and then this would be like delta 1 4 and then this would be like delta 2 4 and all these things are still there.

So, this weight connection and all is there. So, this will be like W W 1 1 1 and then you will have three right W 1 1 3 and then W 1 2 3 what all that is still there and then the bias will be BIL.