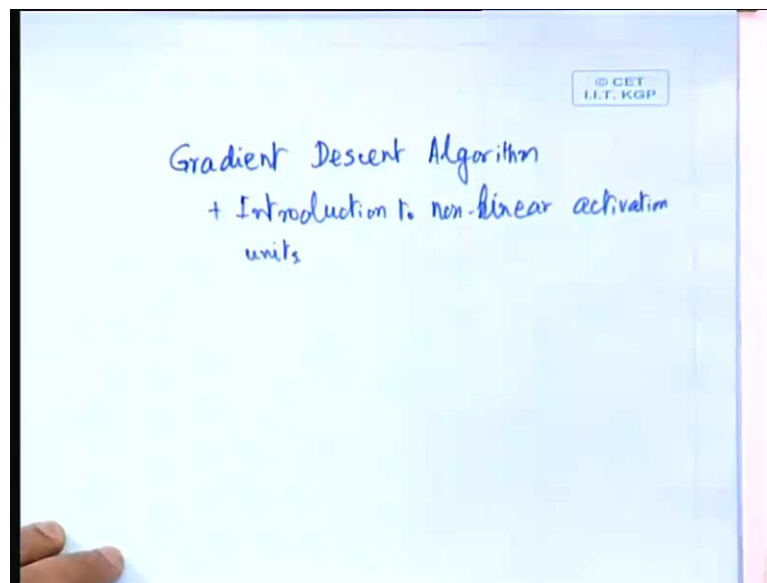


**Neural Network and Applications**  
**Prof. S. Sengupta**  
**Department of Electronics and Electrical Communication Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture - 03**  
**Gradient Descent Algorithm**

Today's, lecture is mostly a continuation of what we were doing in the last one. That is, when we discussed about the linear neuron models, considering the linear activation unit.

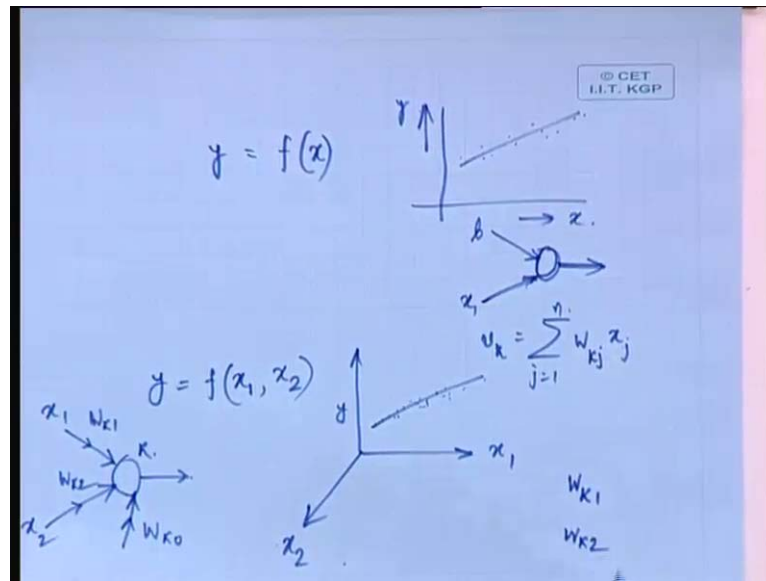
(Refer Slide Time: 01:18)



And today we are going to continue mostly with that, but more stress on the Gradient Descent Algorithm. And later on in the class today we will try to introduce the non-linear activation units. So, although it depends on how much of progress we make today about these two topics, but we will try to cover both these things in this particular lecture. If I mean anything is spill over that will continue in the next one.

Now, what I want to say is that in the last lecture. We had actually presented the case of a linear neuron model, where essentially they interpretation that we had tried to focus upon is that. As if to say that given a set of data, supposing you have got essentially a function.

(Refer Slide Time: 02:41)



Let us say a function  $y$ , supposing an output  $y$  which is function of some variable  $x$ , a simple  $y$  is equal to  $f(x)$  type of a function where we had certain set of observations, may be that these are the set of observations that we had got. And then, essentially the problem was to fit a straight line through this data; and it is essentially that fitting of the straight line which we had modeled using the linear neuron.

So, what we essentially had was that we had a single neuron, which worked on the linear activation unit. Linear activation unit means, essentially it was having  $v_k$  is equal to summation of  $w_{kj} x_j$  and where  $j$  is equal to 1 to  $n$ . Now, for a simple case we had only one such  $x$ , let us say  $x_1$  and we had the bias which was  $b$  or  $b$  called it as  $x_0$  and then, we got the output. And this could be extendable today multi-dimensional case.

Now, one thing which we need to interpret is that, as long as it is a single variable, like  $y$  is a function of  $x$ , we know that it is a line fitting. And essentially we have to tune the two parameters, that is the slope and the intercept these two parameters only. So, essentially it is a two dimensional fitting problem, when the variable is one, that means to say that in this axis we had  $x$  and in this axis we had  $y$ . And we had assumed that the function is going to be  $y$  is equal to  $f(x)$ , and it is a linear function that is what we considered.

The function could be non-linear as well, but that case we are considering later on. Now, what is important to think of is that the extension of the side here, that if instead of single

variable  $x$ , supposing  $y$  is dependent upon several variables. Let us say that  $y$  is a function of just to make the problem little more complex we make it  $f$  of  $x_1 x_2$ . So, it is dependent on two variables, so in that case the line fitting problem that we are looking at it is essentially becoming a three dimensional line fitting problem.

In a sense that, in that case we can consider that we have got two axis like this, let us say this one is our  $x_1$  axis and this one is our  $x_2$  axis. And this is where we are plotting the function that is to say to  $y$ . So,  $y$  is a function of  $x_1 x_2$ , so which means to say that essentially in the 3 D we will be getting some points, which will be obtained to us which will be given to us from the experiments that we perform. Or rather the training that we impart to this network and essentially it is a problem of fitting a 3 D straight line.

So, and how is it getting model into the neural network, simply we will be having a neuron with two inputs  $x_1$  and  $x_2$ ; and the third adjustment will come from the bias. So, essentially we will have three synoptic weights to tune, if this is our neuron  $k$  in that case it will be  $W_{k1}$ , this one will be  $W_{k2}$  and this one will be the bias that is  $W_{k0}$  and finally, it will be the output.

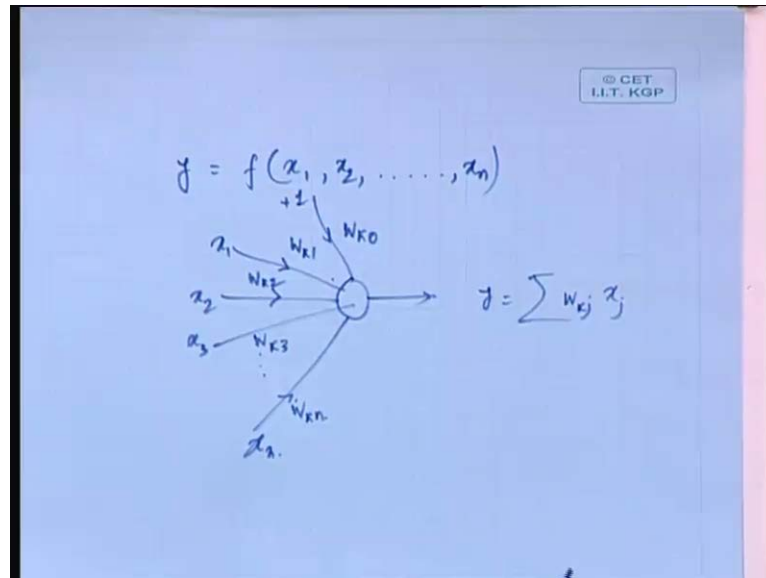
Now, how to interpret the fitment of a 3 D straight line, essentially if you look at a 3 D straight line this has got three components, it has got slopes. In fact, it has got a single 3 D slope you can imagine, but that 3 D slope that we can result it to two components. One component of that slope is along the  $x_1 y$  plain and the other component is along the  $x_2 y$  plain.

So, essentially we will be having two such projections, let us say that in the  $x_1 y$  plane the projection of the fitted 3 D straight line it is going to be let us say  $W_{k1}$ . And the projection of the 3 D straight line onto to the  $y x_2$  plain is going to be  $W_{k2}$ . So, essentially again we are having these two parameters  $W_{k1}$  and  $W_{k2}$  to be 0, so although it is a 3 D straight line fitting, we are essentially breaking it up into the fitment of two slopes.

One is  $W_{k1}$  which is the slope component along the  $x_1 y$  plain and the other is the component of the slope along the  $x_2 y$  plain, which is  $W_{k2}$ ; and on top of it we have got the bias also added to it. So, essentially it is just tuning of three such parameters, if we tune those three parameters we can fit a 3 D straight line. Now, up to 3 D bring into

our visualization, but anything more than 3 D, we cannot bring in into our visualization. So, we have to build up this concept, so what I would like point out is that...

(Refer Slide Time: 08:22)



If in that case, we take a function  $y$  is equal to  $f$  of  $x_1, x_2$  up to let us say  $x_n$  that means, to say that making the variable itself in dimensional. In that case the neuron model will be looking like this, where we will be getting the inputs from  $x_1, x_2, x_3$  and so on, up to  $x_n$ . And this will have their respective adjustment parameters  $W_{k1}, W_{k2}, W_{k3}$  etcetera up to  $W_{kn}$  and then, on top of it we are also having the bias, which we can imagine that a bias of let us say plus 1 and then, we have here  $W_{k0}$  where  $W_{k0}$  will indicate the weight which essentially stands for the bias itself, just giving at fixed input of plus 1.

And then, the output which will be simply the  $y$  is equal to summation form a linear summation form  $W_{kj} x_j$ . Now, in this case the interpretation is like this, that we have fitted and  $N$  dimensional straight line. And essentially that  $N$  dimensional straight line is having so many gradients, gradients along  $x_1$   $y$  plane, gradient along  $x_2$   $y$  plane, gradient along  $x_3$   $y$  plane and so one.

It can be rather projected into all these  $n$  different type of planes and then, these will be there individual slopes and then, we have the bias. So, this parameters if we adjust combined, then it is the problem of fitting and  $N$  dimensional straight line. Now,  $N$  dimensional straight line we cannot bring into our visualization process, but we can

certainly build up this concept, just by extending our thoughts from what we have already learn for  $y$  is equal to  $f(x)$  and  $y$  is equal to  $x^1 x^2$  just extending that part.

So, this is the interpretation that you can keep in mind and then, the problem we had looked at it this way, that we were defining the error that we are doing in the process of fitment. So, the problem was simply decided like this that, for this  $y$  is equal to  $f(x)$  function let us say, we had fitted straight line and then, we had a number of observations where naturally the fitted straight line is not exactly passing through all the observations, that we had made that fitted straight line indeed make some errors.

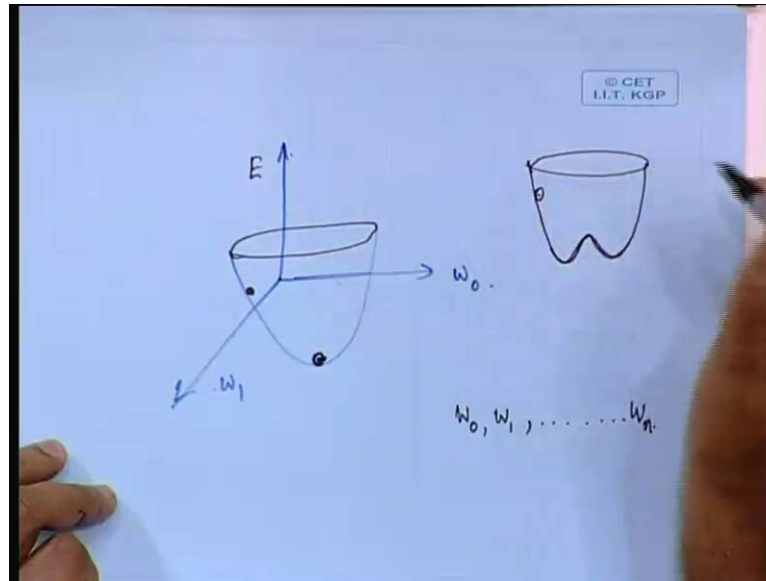
Some error values are positive, some error values are negative, but in order to make a combine estimate of the error, we simply made the square of those errors and we just added it up, so that we get a combined error. Overall the points through which we are fitting this, so that was the combined error measure that we had defined.

And if we have the error, in this case it is the error which we could plot as a function of two variables,  $W_1$  and  $W_0$  where  $W_1$  is the slope and  $W_2$  happens to be the intercept and then, we had considered a 3 D surface through which we could imagine that if we place anything on a 3 D surface. Let us just place an object and then, it is natural tendency will be to come to the point of local minima. Now, there was question that can come into our mind very easily, that is there any guarantee that it will really have such a kind of a minima.

And if at all if it has got a minima, is that minima how confident we are that it will be global minima only. Because, ultimately what our objective it is, our objective is to fit a curve in this case line of course, but our objective is to fit a curve, which is actually giving the minimum error. That is the position that we are looking for, we are searching all over the place for the best combination of  $W_1$  and  $W_0$ . And for any dimensional problem, we are finding out the best combination of  $W_1$  to  $W_n$  and the bias.

So that the best type of fitment is there, what is the best fitment minimum error, but what is the guarantee and what is that minimum. Now, likewise we could imagine, our imagination can work only up to three dimensional surface as we had said that if we plot that the error.

(Refer Slide Time: 13:15)



And if we have the two parameters to be adjusted, let us say  $W_0$  and let us say  $W_1$ . And we had surface of this nature, where we were placing some object and this is a 3 D surface that we had considered. And this object was, if it is initial position is here, then we want to object finally, to be here; and for the object to come and rest over here whatever the corresponding positions of  $W_0$  and  $W_1$  are that is the solution.

So, this is the error that we have plotted, so it is a 3 D surface, but again when we are extending the problem, so that we have to adjust  $W_0$ ,  $W_1$  up to  $W_n$ , all these  $n$  parameters simultaneously. Then the problem is extended to an  $N$  dimensional error surface, it is essentially an  $N$  dimensional let us surface that we are looking for. And there we have to find out that what the minima is, now look at the way whereby I have drawn this diagram.

I have drawn this diagram in a very simply way, in a sense that we can clearly see from this object that, this particular surface has got only one minima, only a unit position of minimum. And that is where it is going to be, but it is not to guarantee that, that a surface will have only one minimum. Just try to imagine it like this that supposing, let us hypothetically imagine that a surface has got a shape like this a 3 D surface only, but it has got a shape like this and it has got a minima, it has got another minima.

Now, these two minima may be different in their magnitudes, out of the several minima's which a surface can exhibit. Only one of the surfaces, one of such positions

will be the global minima, that the actual minima exists. And all other places will be having all other minima's that we consider around it, there are all the local minima's. So, there is a possibility that, if you start with some initial state, supposing the initial state is here, the initial state could be there also.

And we allow the system to adopt that is what it is doing, it is ultimately adopting itself, learning and correcting itself, so that it gets the best fitment. So, while doing that, in doing that process it could either come to the absolute global minima and that is what we want. Or it could come to the to one of local minima's and get trapped in the local minima's. So, really speaking it is very much problem dependent, it depends on the problem and also it depends upon certain neural network configurations.

So, right now at this stage, let us keep this aspect open, let us not immediately come to the conclusion that any neural network and any problem that we consider is going to have the global minima only. It may not be, let us take it that way, that it may not always guarantee a global minima, it could be a local minima also and that is what we have to keep in mind. But, if there is a global minima solution, ultimately we have to reach for a global minima and how we reach it, that is what something that we have to think of.

Now, once again we take the same problem, in the sense that we have got these kind of a surface and we have kept some initial position and a object happens to roll down to this. Now, how is it roiling down, it is following the direction of the gradient, rather the gradient is pointing up and the ball or the roller that is coming down. So, it is against the gradient, against the opposite to the directional of the gradient where it is moved. So, this is what we call as the gradient decent and we are going to consider the gradient decent algorithm.

(Refer Slide Time: 18:16)

© CET  
I.I.T. KGP

Gradient Descent Algorithm.

$$E = \sum_p E^p$$
$$E^p = \frac{1}{2} \sum_p (t_p - y_p)^2$$

Desired Actual

$$G_j = \frac{\partial E}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_p E^p$$
$$= \sum_p \frac{\partial E^p}{\partial w_{ij}}$$

So, let us look at the gradient decent algorithm what it means, now let us consider that we have got  $p$  number of points,  $p$  number of points means we have  $p$  number of observations. Observations for which the data is already available, observations which we are using for the actual training of the network. So, our combined error measure if  $E$  is a combine errors, then  $E$  can be written as the summation of  $E^p$ .

So, it is summation of  $E^p$  over  $p$  and I have been indicating this  $p$  as a superscript, indicating that it is the error for the point  $p$ . So, it is the combine error that we are considering and if we consider the every individual error  $E^p$ , that can be defined as what as the summation of the square difference. So, what we can do is that this  $E^p$ , that is going to be  $\frac{1}{2} (t_p - y_p)^2$ , where  $t_p$  happens to be the target minus  $y_p$ , so that is for the point  $p$ , so  $\frac{1}{2} (t_p - y_p)^2$  whole square.

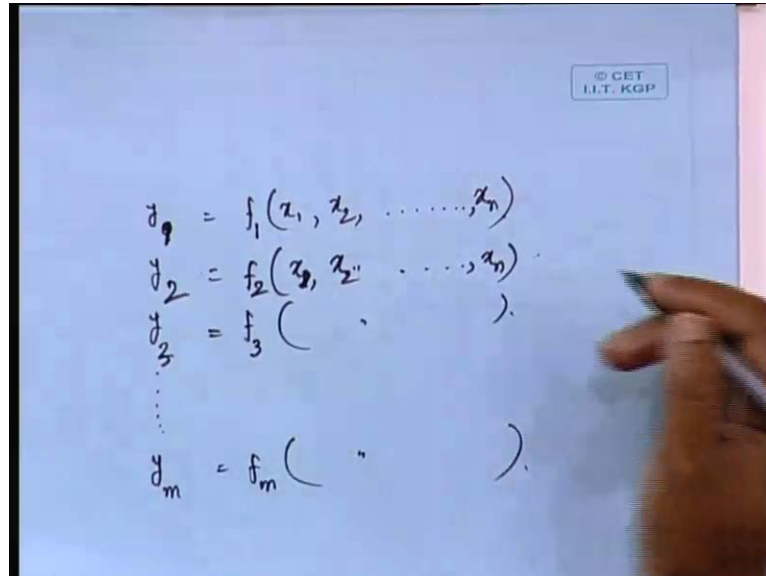
If this is the output, this is the target output and this is the output that we have got from the system, so it is what is desired. So, this is the desired and this is what we had and got as actual and there is a indeed departure between the desired and the actual and this is for one output unit. But, mind you typically a neural network is not going to have a single output, it is going to have several outputs.

If we have like say for example, you imagine any decent control system, any typical control system that you can imagine for any industrial application. That is certainly going to have multiple parameters to ultimately control, it has got several outputs to



control. So, we can indeed pose a problem like this, that we are going to have several such outputs.

(Refer Slide Time: 20:51)



The image shows a whiteboard with handwritten mathematical equations. In the top right corner, there is a small logo that reads "© CET I.I.T. KGP". The equations are written in black ink and represent a set of functions:

$$\begin{aligned} y_0 &= f_0(x_1, x_2, \dots, x_n) \\ y_1 &= f_1(x_1, x_2, \dots, x_n) \\ y_2 &= f_2(x_1, x_2, \dots, x_n) \\ &\vdots \\ y_m &= f_m(x_1, x_2, \dots, x_n) \end{aligned}$$

Like  $y_0$ ,  $y_1$ ,  $y_2$  etcetera, up to let us say  $y_m$ . And each of these are going to be a function of  $x_1, x_2$  up to  $x_n$ ,  $y_1$  also is going to be the same, so all these there are different outputs. So, what we have to do is that in order to measure this combined error for a point  $p$ , we have to find out that for this point  $p$ , what is the combined error from all the outputs.

So, I should sum it up over all outputs, that is where  $o$  stands for the index of the output and this is  $t_{0p} - y_{0p}$  squared. In fact, for our convenience, we are not exactly taking it this way, we are just multiplying it by another factor half. And this is only for a computational convenience, because the reason why I tell you is that ultimately, we are going to find out the gradient of the error. And this brings a squared quantity the gradient is naturally going to be two times of this.

So, having half outside is always convenient, so that you get half multiplied by 2 and you get that part eliminated. So, that is why it is only for the mathematical convenience that we define the  $E_p$  to be half of this summation of this squared quantity. So, that given this, so I hope that the things are clear, this summation over  $p$ ,  $p$  is the number of points over which we are observing.

And  $o$  is the number of outputs, number of different output units that is existing in our neural network system. Yes please is there any doubt that.

Student: ((Refer Time: 22:56))

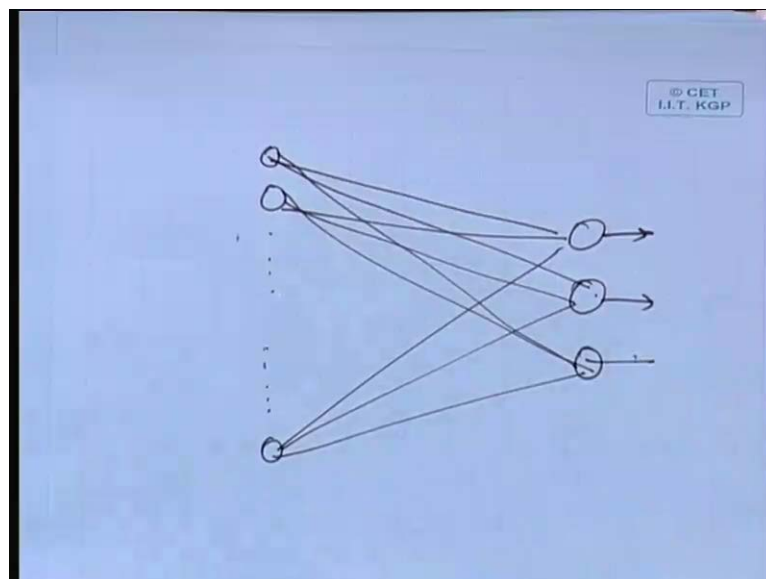
No, there is there is not one function in fact, that is the good thing to point out, what we have to consider is that  $y_0$  is a function  $f_1$  of this,  $y_1$  is a function  $f_2$  of this. So, again  $x_1$   $x_2$  like this up to  $x_n$  this is or just to put it in a consistent manner  $y_1$  is  $f_1$ ,  $y_2$  is  $f_2$ ,  $y_3$  is  $f_3$  and  $y_m$  is  $f_m$ . So, essentially it is all different functions, if we have  $m$  different outputs, then we are going to ultimately approximate  $m$  different functions.

So, it is a common function ultimately we are, ultimately our objective is to generate  $m$  outputs.

Student: ((Refer Time: 23:56))

So, the input parameters could be common and then, it could have a number of observations obtained out of it, it could be having number of conclusions drawn out of it. Let us say for example, I give you a simple example, let us say that the input is a satellite image.

(Refer Slide Time: 24:28)



Satellite image means what, that we have got several such inputs what are the inputs, those are the pixels. Now, these are the inputs to the system and then, ultimately we are

wanting a classification to be done that means, to say that which areas are the vegetation areas, which are the water areas. And like this we may be classifying the satellite image into three different classes, but mind you this classification will be based upon all the pixels that you are receiving.

So, if you are looking at the classification number 1, the classification number 1 is a function of is sum of all these pixels. Classification number 2 that is also a function of all these things, classification number m that is also a function of all these things different functions. So, ultimately this is our network that we will be having m different inputs, m different outputs, this n and m could be anything. So, we are looking at the problem that way, any other doubts.

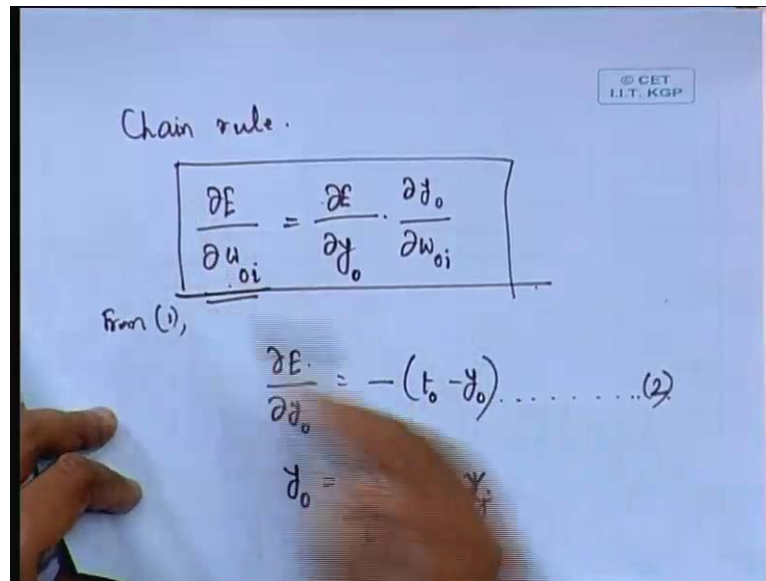
If there is none I can continue, but so this is the definition of the error that we are considering. So, now we have to do what, we have to compute the gradient of this error, so what is the gradient, the gradient we can define as follows that the gradient we call as G. G will be the defined as the  $\frac{\partial E}{\partial W_{ij}}$ , so what is  $W_{ij}$ ,  $W_{ij}$  is nothing but, the synaptic interconnection weight between what from neuron j to neuron i.

So, if we make it like this in that case this problem, this  $\frac{\partial E}{\partial W_{ij}}$ , now out of this E becomes the summation of this  $E_p$  and what we can simply write is that, this is  $\frac{\partial E}{\partial W_{ij}}$  of summation of  $E_p$  over this p. That is just a different way of writing this E and this effectively is nothing but, we can take this under the, we can take the summation under this, so this is  $\frac{\partial E}{\partial W_{ij}}$ .

So, this is the gradient, we have to find out the gradient of error for one point with respective to one set of weights. So,  $W_{ij}$  is only one set of weights and there are several such weights in the system, there are several synaptic weights in the system. So, it is the gradient with respect to this particular weight of  $W_{ij}$  that we are considering and in order to consider, that we have to sum it up over all the observations p.

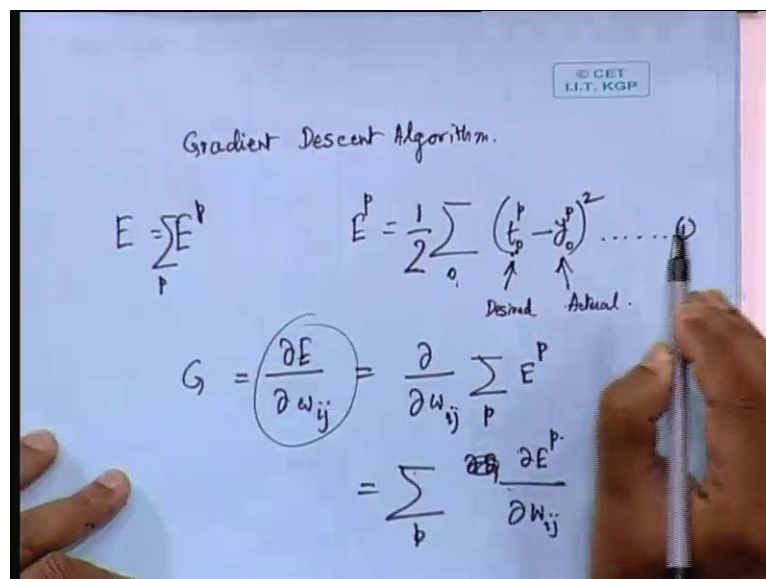
So, summation over p for  $\frac{\partial E}{\partial W_{ij}}$ , now what we have to simply do is to perform this differentiation. And to perform this differentiation we just apply the chain rule out differentiation to compute this. So, what we do is that we have to compute ultimately this thing.

(Refer Slide Time: 28:23)



So, to compute this we simply define  $\frac{\partial E}{\partial w_{oi}}$  let us say, and that should be equal to  $\frac{\partial E}{\partial y_o} \frac{\partial y_o}{\partial w_{oi}}$ . Let us say  $\frac{\partial E}{\partial y_o}$  where  $y_o$  is just to recollect,  $y_o$  is the output. So, this is derivative of  $E$  with respect to the output multiplied by  $\frac{\partial y_o}{\partial w_{oi}}$ . We can attract either this is just by applying the chain rule out differentiation and let us have a look at this expression, now this we can call as the equation number 1.

(Refer Slide Time: 29:15)



So, now if we have a look at the equation number 1 and we take the derivative of this. What do we get, we can take the derivative of this with respect to  $y_o$  cannot we, because

it is a function of this is here  $E_p$  has been clearly expressed as a function of  $y_0$ . So, we can indeed take the differentiation and if we do that, that means to say that if we take the derivative of one. So, we can write at from 1 what we obtain, we obtain  $\frac{dE}{dy_0}$  and that is equal to  $-\frac{1}{2} t_0 - y_0$ .

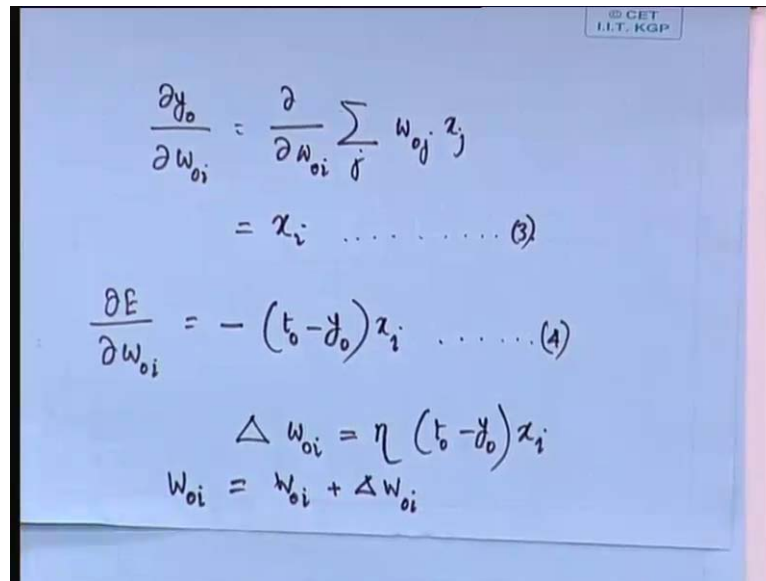
This is what we are getting, this is for any point  $p$  that we can take, so here because it is to the power 2 over here and because this is with a minus sign, that is why half gets cancel with 2. So, now, you can realize the quantity of having a half outside it, so that this looks pretty good, that it is  $\frac{dE}{dy_0}$  simply becomes equal to minus of  $t_0 - y_0$ .

So, that is the ultimate derivative that we have got, but that is the derivative with respect to  $y_0$  only, but ultimately we want to find out the derivative with respect to  $W_0$ . So, what I have to do is simply also to calculate this. How would I calculate this  $\frac{dE}{dW_0}$   $W_0$ , how would I calculate this is it easy difficult, which one do I apply, which particular equation do I take?

Student: ((Refer Time: 31:13))

Exactly, I simply take the summation equation, because this one is already defined with us that  $y_0$  is equal to summation of  $W_0^j y_j$  and we are summing at over  $j$ . Now, that is  $x_j$ , that is why, so that is  $x_j$  and we are summing it up over all the  $j$ 's. So, essentially what happen is that we have now, if we are applying this  $w y_0$ ,  $\frac{dE}{dW_0}$  what we get we have to take the derivative of this.

(Refer Slide Time: 32:14)


$$\frac{\partial y_0}{\partial w_{oi}} = \frac{\partial}{\partial w_{oi}} \sum_j w_{oj} x_j$$
$$= x_i \dots \dots \dots (3)$$
$$\frac{\partial E}{\partial w_{oi}} = - (t_0 - y_0) x_i \dots \dots \dots (4)$$
$$\Delta w_{oi} = \eta (t_0 - y_0) x_i$$
$$w_{oi} = w_{oi} + \Delta w_{oi}$$

So, by taking this derivative we can see that it becomes  $\frac{\partial y_0}{\partial w_{oi}}$  that is given by  $\frac{\partial}{\partial w_{oi}} \sum_j w_{oj} x_j$ . And there we will be having  $w_{oj} x_j$ . And what is it becoming  $x_i$  and what is  $x_i$  that is the input. So, now this is  $x_i$  only, so this if I say to be equation, we should have mark the equations little earlier, so if I mark this thing as let us say equation 2.

One we have already marked if this is number 2 and if this one is a number 3, then simply from this equation, if we look back into this it is possible for us to express this  $\frac{\partial E}{\partial w_{oi}}$ . So, we have in that case  $\frac{\partial E}{\partial w_{oi}}$  to be equal to minus of  $t_0 - y_0$  into  $x_i$ , so this is what this is nothing but, the gradient and this is gradient with respect to one particular weight.

That is considering  $o$  as an output unit considering  $i$  as the input unit and we are able to find out the derivative of the error with respect to this  $w_{oi}$ . Now, likewise we are having several  $o$ 's, that is to say several output units and several input units, which is index by  $i$ . So, we have got several  $o$ 's and several  $i$ 's and likewise we will be able to find out the derivative. Now, ultimately what is the direction in which we would like to move.

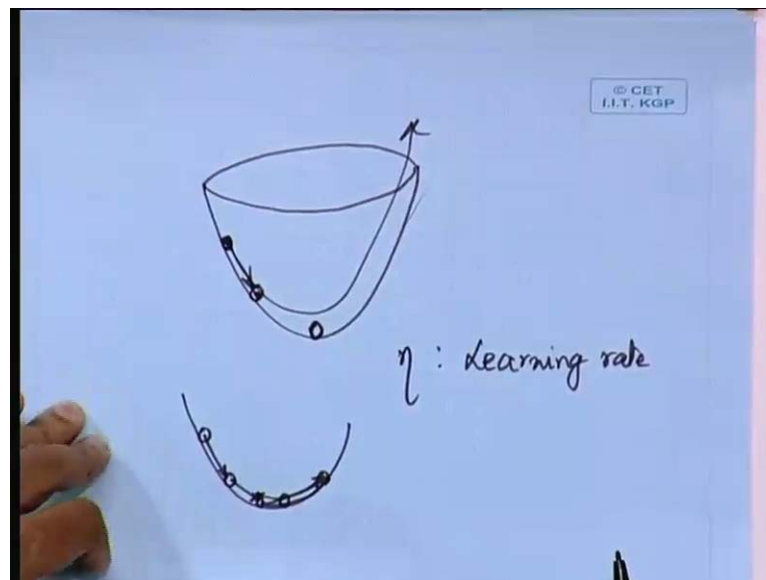
We have found out the derivative with respect to this weight. Now, as I told you that we have to move opposite to the direction of the derivative. So, the correction that we have to apply to the  $w_{ij}$ 's or the correction that we have to apply to the weights, that is going

to be what, that is going to be negative of this, that comes to say since this is already a negative quantity.

We have to consider minus of  $g$  that is to say minus of  $\frac{\partial E}{\partial W_{oi}}$ , so that plus  $i$  mean with the positive sign we will be having  $t_{0i} - y_{0i}$  into  $x_i$ . Now, the physical interpretation of this is quite easy, because what is  $t_{0i}$  target output what is  $y_{0i}$  the actual output, what is  $x_i$  our input. So, it is quite easy to interpret this and all that it means is that the error term that is  $t_{0i} - y_{0i}$ , simply multiplied by the input quantity.

And that is the correction that you have to apply, or rather you have to move in that direction.

(Refer Slide Time: 35:56)



Now, again the question that I had posed in the last class was that, now we have surface like this a 3 D surface. And this was our initial position and definitely by applying this kind of an algorithm, necessarily means that we have to apply the gradient descent. So, gradient descent means, it has to now slide down this surface and in that case we could accelerate the sliding down, we could retour the sliding down, on that we can exercise some control.

And control in what manner, if we this is the derivative that we have got, but ultimately what is the correction that we have to apply let us think of that. Now, this means to say that to this  $W_{oi}$ , to the existing value of  $W_{oi}$  we have to apply a correction which we

can write as  $\Delta W_{oi}$ . And what the  $\Delta W_{oi}$  is going to be, we can simply take  $t_0$  minus  $y_0$  times  $x_i$ , meaning what that if presently I have  $W_{oi}$  as the weight. Then the new weight is going to be, the new synaptic weight is going to be the present synaptic weight  $W_{oi}$  plus  $\Delta W_{oi}$ .

So, it is  $W_{oi}^{\text{new}}$  is going to be  $W_{oi}^{\text{old}}$  plus the correction that we are applying into it. So, now there is a choice that either we can apply this correction in its entirety, or we can decide to multiply this correction factor by something by some constant. Let us call this as  $\eta$ . If we multiply this gradient by some constant quantity  $\eta$ , that certainly does not affect our proceeding in the gradient descent direction.

It is not affecting the direction, only thing it is affecting is the rate at which we fall down. The rate at which we fall down will be controlled by this parameter  $\eta$ , now ultimately what we are doing out of it. We are in some initial position and this is the minima position, which is our destiny. We are going to reach that and through what process are we reaching, we are reaching that using a learning mechanism. We are initially here and then, what did we do we calculated that what error did we incur.

And based on the gradient of the error, we are applying a correction and correction in a correct way. Correction in a manner that if from this position we come to this position, naturally the error that we are having is much less than the error that we were having out there. So, we are reducing the error and slowly proceeding to the ultimate destiny. Now, how fast we go is dependent upon this parameter  $\eta$ .

And since it is a process of learning, the best name that we can attach to this factor  $\eta$  is the learning rate, so here  $\eta$  will be defined as the learning rate. So, if you are making the learning rate high, it learns faster. If you are making the learning rate slow it learns slower as simple as that, but think it is a simple way, because let us take the case of human learning.

A teacher can tell you that, you see this neural network course has to be covered within 5 lectures, I want finish one semester neural network course in 5 lectures. So, your learning rate has to be very very high and I am going to finish of 5 or 6 topics in every lecture, in 5 lectures the course will be over. If I tell you that and if you start attending my lectures this way, will you be able to learn that fast, you will have problems.



So, ultimately what you will feel is that, ultimately at the end of such 5 excessively high learning rate lectures you will finally, conclude that no, whatever we had learnt we have already started forgetting that. The teacher has not given enough of time to us for learning. Whereas, if I go in the other direction, if I cover for every topic if I spend 3 or 4 lectures, then ultimately at the end of 40 lectures you are going to feel, that neural network is such a fast subject, but the teacher has not covered much of that.

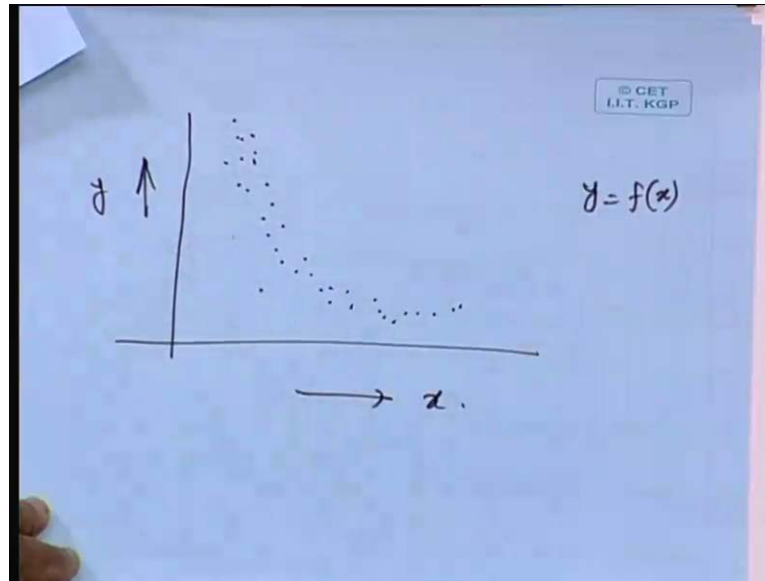
We were having very slow progress, but may be by making slow progress, you have come from here, but you are not lost completely, but if you try to learn too fast. You may finally, just go I may give you such a great push that you will cross the minima and ultimately you may fly away is possible. So, let us see that this learning rate, that we are talking of is well optimal, we should not have the learning rate too high, too high means it can lead to essentially what it means is that it could lead an instability.

Or oscillatory behavior of the system, let us say if we start conducting an experiment with this  $\eta$ . What can happen is that if this  $\eta$  is made too high, it can come from here to here, it can come from here to here and then, it can go from here to here it is remaining on the surface only. And then, again it comes back from here to here and the ultimate global minima position, it will be very difficult to reach. It will take more number of alterations or in this case the behavior will be oscillatory in nature.

So, that is not something that we are looking for. So, this learning rate is very important, in fact the success of the neural network convergence a lot depends upon this learning rate let us understand that. So, having talked of this gradient decent, essentially what we have done is that, in this process we have computed the gradient of the error based on the linear neural network model. Why linear, because this is essentially the equation that we have applied, the equation for the linear neuron, linear activation you need that is what we do.

Now, interpretation why it is what is this linear neural network leading us to as I told you, straight line fitting, a straight fitting in  $n$  dimensions that is what the problem is ultimately. But, the question is that, for any general data fitting or for any generalized curve fitting, is it that straight line fitting is always the best solution never. For most of the practical data's or practical situations that, we encounter we will be finding a non-linear characteristics.

(Refer Slide Time: 44:47)



Let us consider a case, that we have perform some experiment, let us say  $y$  is equal to  $f(x)$  that is what we are finding out. But, we have  $y$  in this direction and  $x$  in this direction and we have got the observations like this, say these are the set of observations that we have got. Now, if I have to fit the best curve, the best curve is settled in all the straight line, I can attempt to fit straight line, but whatever attempt I make.

Even the best attempt is going to give lot of errors, there will be a lot of points which will be deviated from the straight line. So, ultimately what we have to fit through is some form of a curve, in this case may be that the best solution will be to fit a curve like this, through all this set of points, if you could do that, that is the solution. So, we are ultimately looking at from a general point of view, we are looking at a non-linear curve fitting problem.

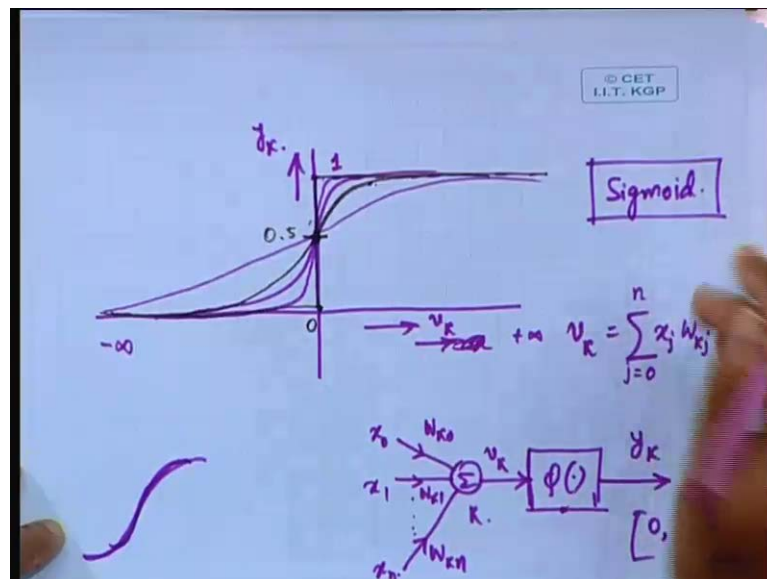
Now, there will arise a problem, that if the problem is a non-linear curve fitting can we do it using the linear neural network model is the answer yes or no. No, because we cannot approximate a there is a way whereby you can do it, that if it is a non-linear function you can break it up into piecewise linear components. And then, you could realize, but that is a very painful process whereby you just break it up into piecewise linear. And then, try to realize all these straight line using the neural network component.

So, that itself will be quite a complicated problem. In a sense in that case you will have to fit several such straight lines, with their individual slopes and intercepts, so many

parameters you have to tune. Rather, if you could a curve like this then that would have been a simpler solution. So, now what we have to look for, we cannot always have the linear neuron model, rather we should go in for some kind of a non-linear model.

Now, when such kind of problems cropped up, that ultimately a neural network has to approximate some kind of a function like this. People started, researching with different types of neural networks or different models of activation. And one of the activation models which people came up with is some activation function, if we could consider to be like this.

(Refer Slide Time: 47:40)



That supposing, this is the input let us call the input by  $x$  only, not the input, rather we should model it this way, that supposing we have got a neuron. Where we have got that is a inputs has  $x_0, x_1$  up to  $x_n$ . And no matter whether we are using a linear neuron or a non-linear neuron, this summer is essential. Because, ultimately what we are going to have is that after this summer we are following up this linear combiner with some activation function.

Now, in the earlier case we had considered that to be simply a linear function only. But, in this case this five function we are going to realize is going to be a non-linear function, that is the only difference. But, other things remain the same meaning that if this is the neuron  $k$  which is under our consideration. Then if this is  $W_{k0}, W_{k1}$  and this is  $W_{kn}$ ,

so ultimately your  $v_k$  that is what you are getting as the linear combiner output.  $V_k$  is equal to the summation of  $x_j W_{kj}$  and we are summing up for  $j$  is equal to 0 to  $n$ .

0 to  $n$  means there we have included the bias as we have discussed in the model that we had followed in the last class. Now, here we have the  $v_k$  and it is the function  $\phi$  of  $v_k$ , which we are thinking over and ultimately we are having the output  $y_k$ . Now, in earlier case we had got  $y$ , we had made it for the linear neurons we had made  $y_k$  equal to  $v_k$ , but in this case we are just going to pass this  $v_k$  through some non-linear function  $y_k$ .

So, how are we going to do that let us think of some model where, so what is the input to this function, the input to this function mind you is  $v_k$ . So, the input is  $v_k$  and the output is going to be  $y_k$ . So, this  $y_k, v_k$  characteristics, why do not we imagine like this that, let us consider a neuron whose response ultimately should lie in the range of the 0 to 1. So, 0 to 1 is the ultimate range that we are talking of meaning that when  $v_k$ , what is the range  $v_k$  can assume, let us say a value anywhere between minus infinity to plus infinity.

Supposing that is the range that we are following for  $v_k$  and in that range of  $v_k$ ,  $y_k$  could have a value only in the range of 0 to 1. So, to get a characteristic like this, I mean we can say, so that means to say that  $y_k$  is one value is 0, the other extreme is that it could take a value of 1. Now, we had that in the McCulloch and Pitts model which we discussed in the last class, where we had followed a threshold logics. So, there in the McCulloch and Pitts model, we had considered that up to here the response was 0.

Here there was a discontinuity and then, it was equal to 1. But, instead if we think of a non-linear function, where the characteristic could be define like this, that we do not exactly restrict  $y_k$  to binary values. But, we consider that  $y_k$  could take values in the continuous domain, but within 0 to 1 range. So, in that case why cannot we imagine a function of this nature.

We can imagine a function of this nature, where we can say that when  $v_k$  is equal to 0, the value of  $y_k$  will be exactly meet by between 0 and 1. That means, to say 0.5 the value could be and we are going to realize a function of this nature. Where at minus infinity it will be  $y_k$  will be equal to 0 at plus infinity the value will be equal to 1, but in between it should follow a smooth slope like this.

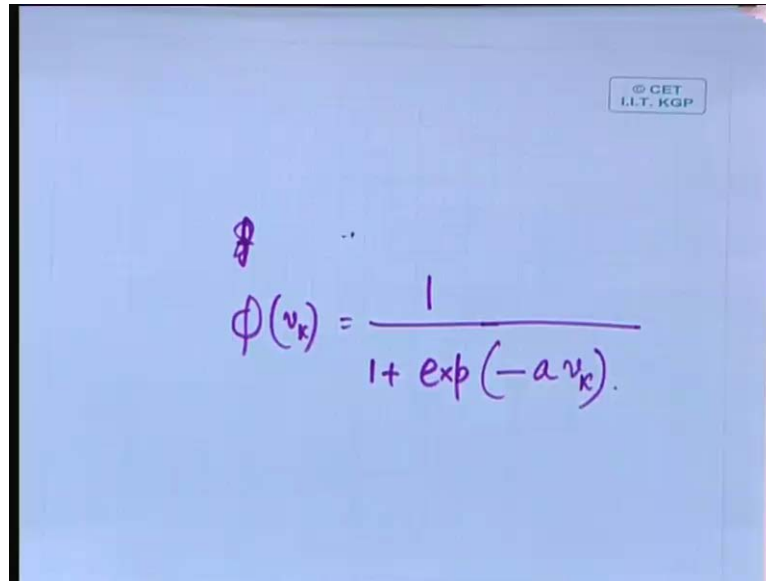
Now, mathematically we have to define this function, but let us look at some of the characteristics of this. Do not you find that this function that I have just now drawn, look at the characteristics of this function. First is that it is monotonically increasing, it is monotonically increasing function. And now if I ask you to find out that is it continuously differentiable it is, it is continuously differentiable. Mind you McCulloch and Pitts function which was a threshold logic was not continuously differentiable, but this function is definitely continuously differentiable.

So, this is having three characteristics non-linear, monotonically increasing, continuously differentiable, but I have drawn a shape like this. Now, you could tell me that now, I do not you like your shape exactly if you ask me to draw, I would have drawn it like this. Somebody can say that no, I should make it even steeper, somebody can say that no I want to make this function more flat.

So, if I use this sort of a function as an activation function  $\phi$  to this neuron, I should have some kind of a tunability. Tunability in the sense that, I should be able to control the shape of this curve and how am I controlling the shape of this curve. I mean this is what I am doing that, either I am making it more and more steep, the ultimate of that is going to be McCulloch and Pitts model.

If I make it excessively steep, it is ultimately going to approximate McCulloch and Pitts model, the threshold function. And if I go to the other extreme, then it is like a straight line. So, it is in between linear model and McCulloch and Pitts model, where I can play around. And the function that we can think of in order to realize this.

(Refer Slide Time: 55:14)



A handwritten equation on a blue background. The equation is  $\phi(v_k) = \frac{1}{1 + \exp(-a v_k)}$ . Above the equation, there is a small symbol that looks like a crossed-out 'f'. In the top right corner, there is a small logo that says '© CET I.I.T. KGP'.

We can write it this way that,  $f$  or rather the  $\phi$  function  $\phi$  of  $v_k$ , we can write as  $1$  by  $1$  plus exponential to the power minus  $a v_k$ . If you plot the function you will get some kind of a shape like this ((Refer Time: 55:40)). In fact, the shape of this function is somewhat like this, this is the kind of a shape is like shape. So, that is way this category of functions are called as the sigmoid function. So, these functions are quite popular they are known as sigmoid function.

So, sigmoidal functions could be used as the activation functions, for as the non-linear activation functions for the neurons. So, this is just a realization of a sigmoidal function, it is not that sigmoidal function can be realized in only one way. But, this is one way to realize this sigmoidal function, I think more of this we can see in the coming lecture, any quick questions.