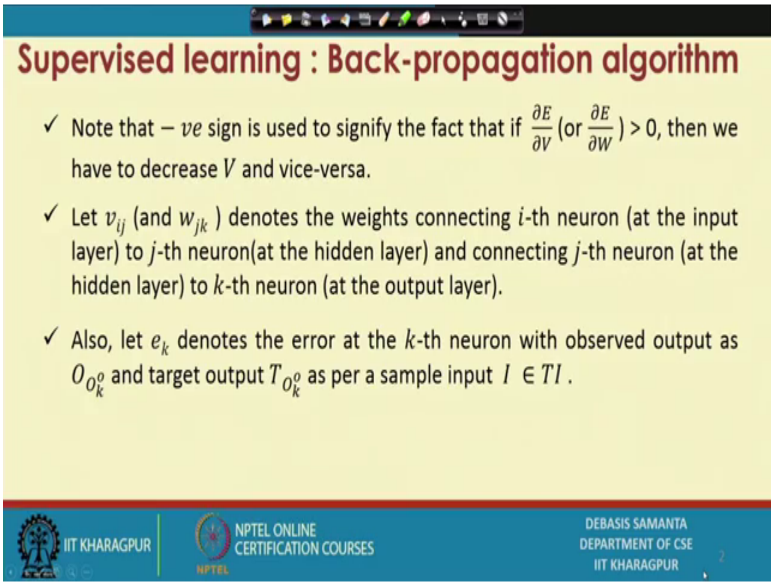
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 39
Training ANNs (Contd.)

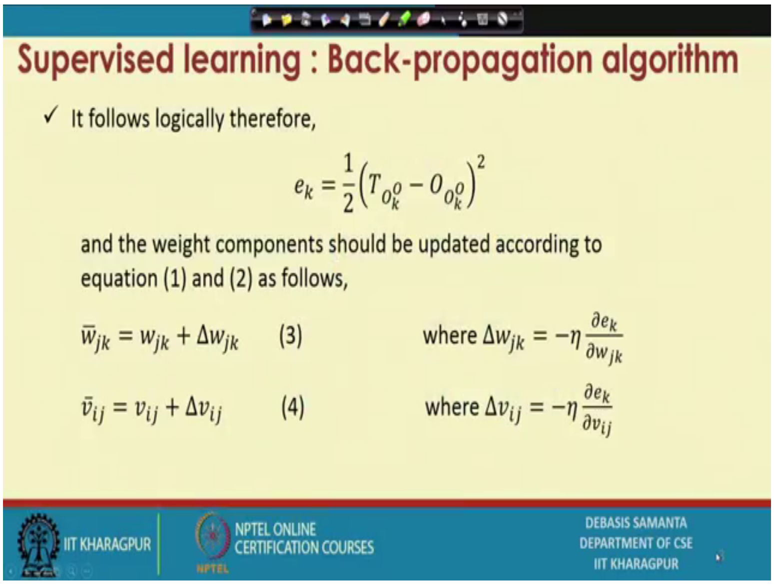
So, you are discussing about training multilayer feed forward neural net and training procedure that we are going to follow is basically supervised learning and in a special case of supervised learning we are learning back-propagation algorithm.

(Refer Slide Time: 00:39)



Now, in the last lectures we have discussed about the chain rule based on the method of gradient descent. So, chain rule basically tell about if we can compute the error of any neuron and then using this error calculation how we can decide the updated value of the neural network parameters. So, these are the concept that we have learned in the last slides.

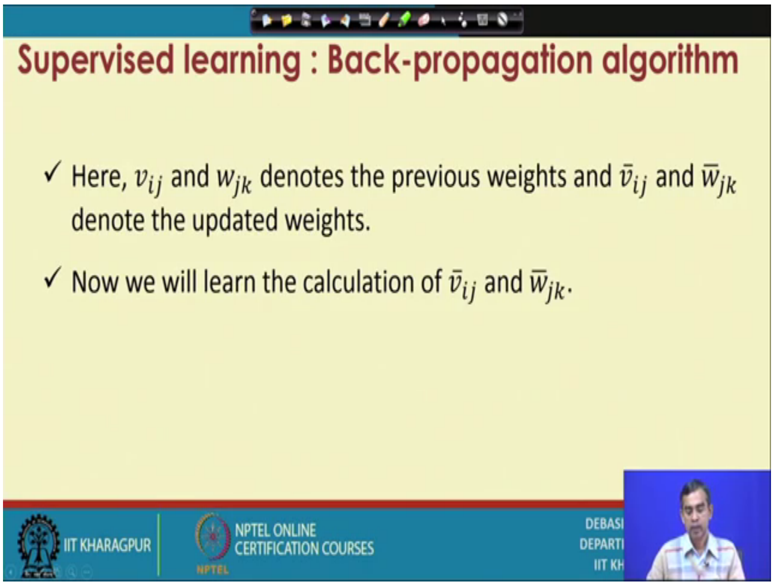
(Refer Slide Time: 01:01)



And, also we have discussed about how the error can be computed and thereby what will be the delta rules. So, the delta rule that we have discuss in the last is these are the delta rule. So, the delta rule is there and based on this delta rule the updated values of the neural network parameter that can be obtained.

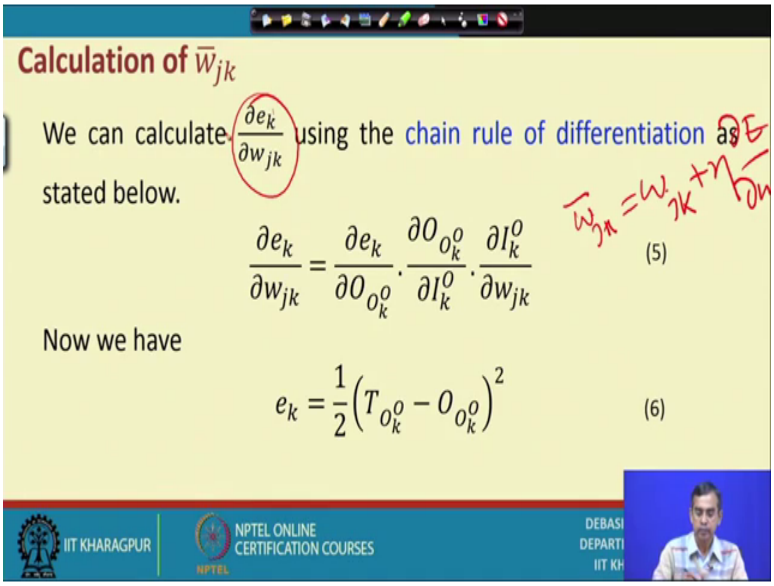
Now, that things we have these are the things we have discussed in the last lectures.

(Refer Slide Time: 01:28)



Now, in this lecture we will discuss about how the updated value can be computed and there is a systematic method or step and this systematic step is basically called the back-propagation algorithm which is a main point of discussion in today's in this lecture.

(Refer Slide Time: 01:52)



Now, so, back-propagation algorithm, it is basically the idea about is that ok, first we will discuss about the calculation of the chains weight values in between the j-th neuron in the hidden layer to the k-th neuron in the output here. So, it basically we have to calculate w jk this is updated value this is equals to w jk the previous value plus it is basically and delta E by delta w jk. So, this is the things that we have to consider it and this basically to essentially this is a calculation of del e k by del w jk. Now, let us see how this calculation can be obtained.

Now, here so, basically how the error is increases with the increased of the values of w jk if it is represented by del e k by del w jk, then we can say that this error e k is basically represented by the output of the output of the k-th neuron in the output layer. So, it is basically e k is a function of output of the k-th neuron in the output layer. So, it is basically del e k by del k.

Now, this output of the k-th neuron in the output layer again is depends on the output of the k-th neuron in the input layer. So, it is basically we can denote that I k o. I k o denotes that what is the input to the k-th neuron at the output layer. So, input to the k-th neuron at the output layer is bes basically influenced the output of the k-th neuron at output layer and now, again this values; that means, input to the output layer k-th neuron of the output layer is basically see influenced by the weight of the weight from the j-th layer to the k-th j-th perceptron in the hidden layer to the k-th layer.

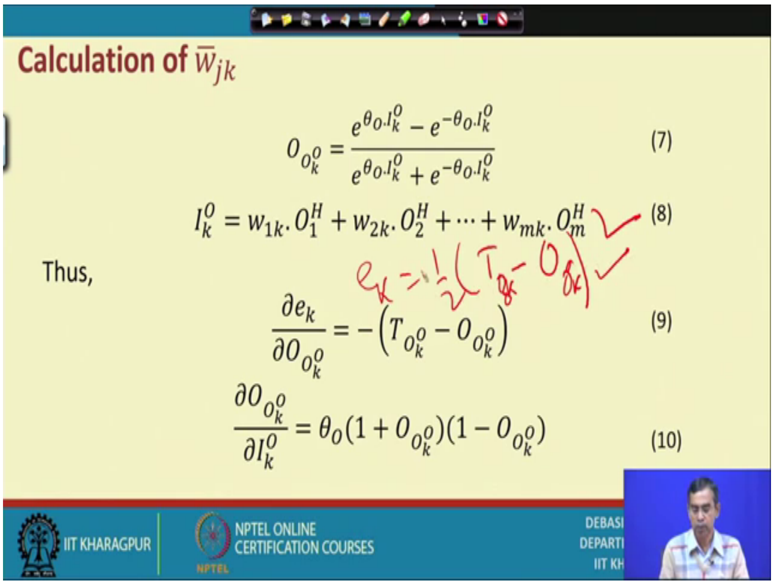
Now, so, this basically is the idea about this is the chain rule. So, if we know this then this and then this. So, it is basically chain rule. Now, this chain rule chain rule of differentiation rather, is basically to compute the error and then the relation between error and w jk. I hope you have understood this chain rule of differentiation, it basically gives is basically the chain chaining of differentiation or a dependency parameters; what I can say in other words if we move from input to the output direction.

So, the w jk will influence the input, this input will influence the output, this output will influence the error and therefore, it is basically del e k by del j k, that means, how the error is influenced by the this one. So, error is basically influenced by this one, but it is in the form of a chain. So, this chain can be propagated in a back direction, it is basically output to the input to the w value. So, it is called back-propagation.

Now, so, this form this calculation is easy once we know the e k value. e k can be obtained in terms of true output and then the observe output. So, this differentiation also can be calculated if we know these functions.

Now, we will see exactly how this can be calculated in a more mathematical way or in a very systematic way. So, so this is the chain rule of differentiation and I have discussed the chain rule of differentiation for the w jk calculation or updated value of w jk calculation.

(Refer Slide Time: 05:59)



Now, we will discuss about more details that how this w jk value can be calculated as we know the output for the perceptron in output layer we also know what is the input to any k-th perceptron in the output layer and also we know what is the output from any perceptron in the hidden layer and then this one. So, basically here all the values if it is known to us then we will be able to calculate this differentiation form first one is derivatives e k with respect to w jk.

Now, let us follow. So, this basically we have already learned about. So far the output layer comp computation that we have discussed there this basically denotes the output of the k-th neuron at the output layer. So, it is basically output of the k-th neuron at the output layer and it is the function that it is here where I k o denotes the input to the k-th neuron at the output layer.

Now, and theta O is basically the threshold function of the perceptron in the output layer. So, this is the formula for deciding or a representing output of any k-th perceptron in the output layer and then the I k o this is the output layer come computation where the input to the output layer we have computed and in this output layer computation.

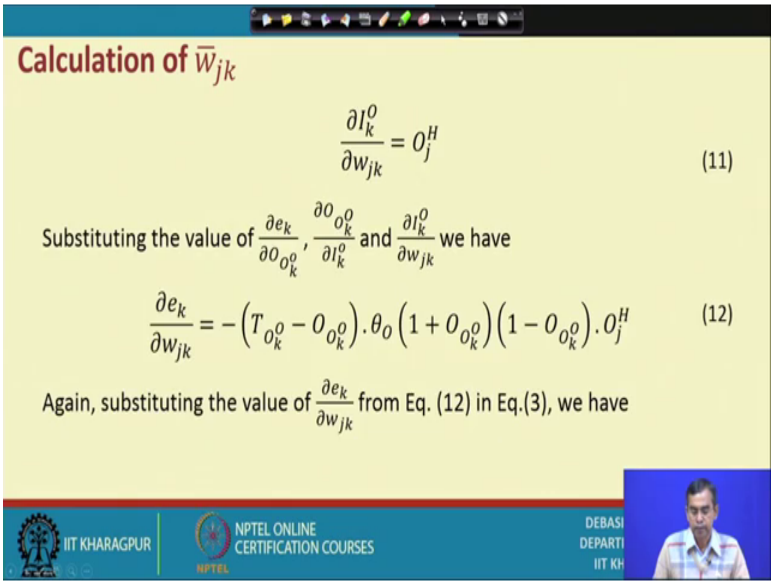
We know that this is the input to the k-th neuron at the output layer and which can be dep expressed in terms of this summation sum of the product form where w 1k is basically the weight value from the neuron perceptron one to the k-th perceptron the this is a perceptron 1 in the hidden layer to the k-th perceptron, perceptron 2 in the hidden layer to the k-th perceptron and corresponding to the output of the hidden layer. So, this is basically output of the first perceptron in the hidden layer, output of second perceptron in the hidden layer and so on so on.

So, this I k o therefore, can be obtained by this one and you have discussed about how the matrix representation of the same thing is there we will see exactly how this matrix representation can be ultimately to be used. Then using this formula so, first we have this del e k by del O k and e k is basically e k that we have discussed about half and T o o k minus object O o k whole square. Now, this del e k del O k, that means, it taking this first order derivatives of this one we will be able to obtain this formula. So, this is basically del e k del O k calculation.

Now, so, for our chain rule of differentiation this is the first calculation and then the second differentiation is del O k by del I k. So, this can be obtained by differentiating this expression with respect to I o k. Here, I o k is a parameter, if we differentiate this one O o k with respect to I k o then we can see after a lot of simplification this expression can be obtained so, theta o into 1 plus O o k, 1 minus O o k.

So, the detailed calculation I have avoided you can try yourself. So, that from these things we can represent the this expression. So, this basically gives the, this is the first differentiation in the chain rule of differentiation. It is the second differentiation in the chain rule of differentiation for w jk and then finally, we will be able to obtain the del I o k del I o k by del w jk.

(Refer Slide Time: 09:41)

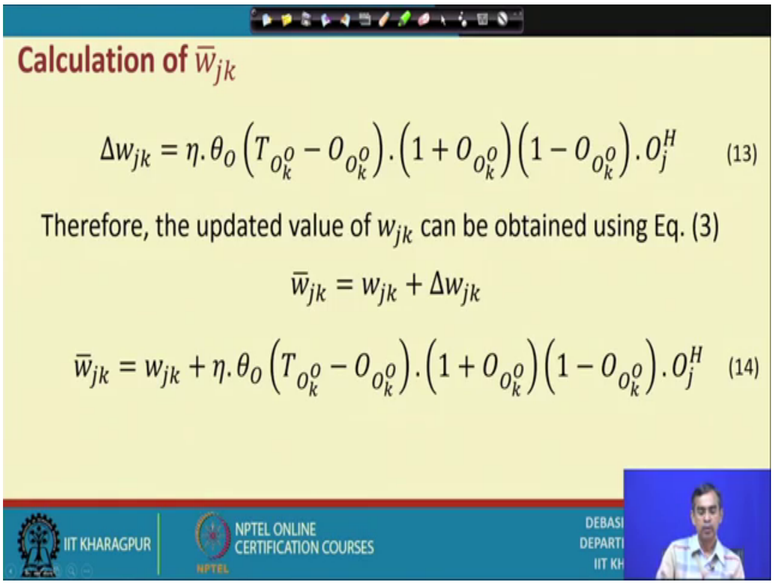


So, it is the formula it is there. Here del I o k del w jk is basically O o h. So, O j H, that can be obtained from the previous calculation, ok.

So, now, we have learned about that the differentiation values of these, this and this one and then combining this by means of chain rule or differentiation we can obtain this one the expression. So, this basically gives the calculation of del e k with respect to del w jk, that means, what will be the error at the k-th perceptron if we change the weight value between the jth and k-th perceptron in the two layers, hidden layer and perceptron layer. So, it is basically the expression that can give you the value of this change.

Now, this value when you substitute it to the modified value of this one we will be able to calculate the modified value.

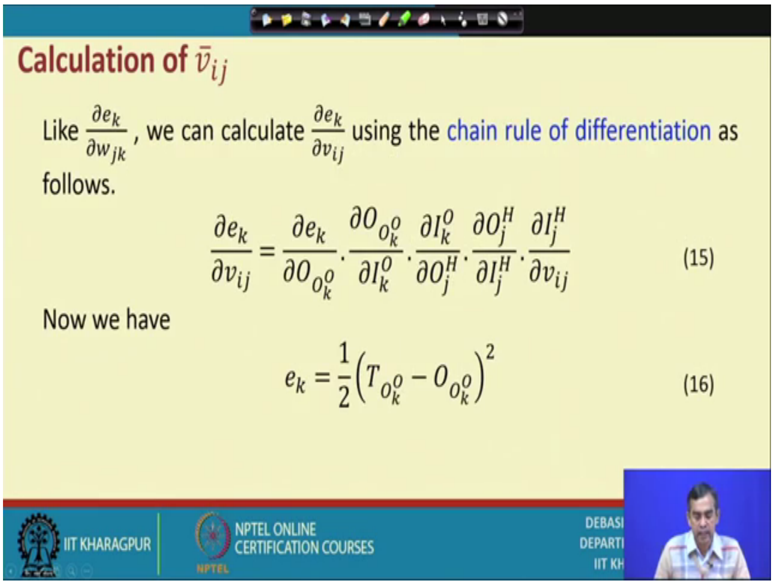
(Refer Slide Time: 10:43)



So, modified value will be like this. So, del w jk it is basically the modified value, del w jk it is the modified value and this basically using the chain rule of that is the delta rule. So, eta into del e k by del w jk. So, once we know this incremented value we will be able to obtain the updated value and this is the updated value.

So, straight away we can write this is the updated value is basically if this is the current value and this is the chain value and then updated value can be obtained. So, you can see that all the updated value can be obtained in star in terms of training data and this training at the this is the true output and this is the observe output, observe output is basically decided by the v and w value as well as the input.

(Refer Slide Time: 11:45)



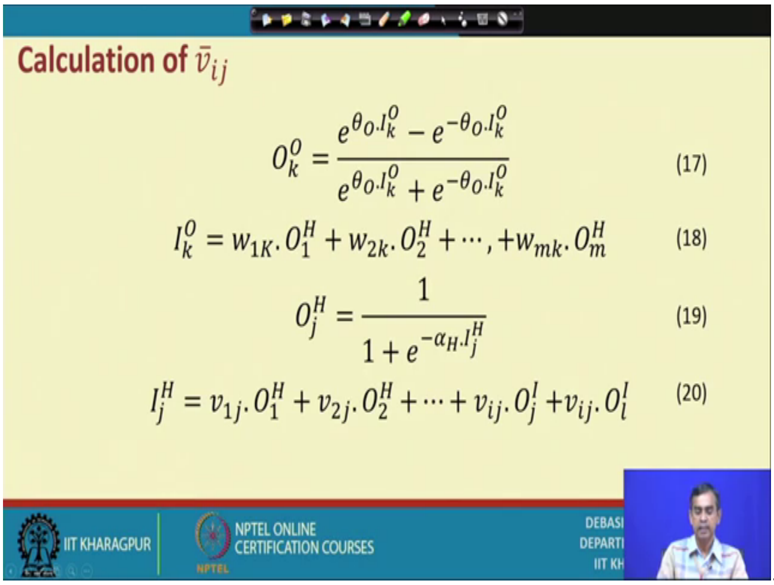
Now, we have learned about how the updated value so far the w matrix is concerned now, we will discuss about the same calculation, but for the v matrix. So, v matrix calculation is like this, it is the same way the chain chaining rule of differentiation that we have followed to calculate the w jk we will follow the same rule, but little bit different way. Now, let us see first discuss about the chain rule of differentiation with respect to v ij; that means, here e k error k is influence when the v ij changes; that means, del e k by del v ij gives basically if we change the v ij then how the error will change and this change can be expressed by means of chain rule of differentiation.

Now, here again see the chain rule of differentiation it is basically we want to find how the error will change if we change the v ij value; that means, the weight values from the i-th perceptron in the input layer to the j-th perceptron in the hidden layer. Now, here this e k as we know is basically depends on output now this output again depends on input to the output layer this input to the output layer depends on output of the hidden layer. Now, output of the hidden layer depends on input of the hidden layer and this input of the hidden layer depends on the values of the v ij.

So, it is basically del e k. So, del e k is del e k is a function of del v ij by it is basically chain from output layer of the out perceptron in the output layer output of the output layer to the input to the output layer to the output of the hidden layer to the input to the hidden layer and finally, to this v ij. So, this chain rule is basically here and the differentiation takes place this form.

Now, we have the all values for example, we have already known the values of O o k, I o k then O j H and then I j H. So, knowing all these values we will be able to calculate all the differential form. Now, let us see how the differentiation form can be calculated. So, for the del e k del O k is concerned we will use this form this is the error calculation and differentiating this e k with respect to O k, we will obtain this one. Now, for this one we will consider I o k. Now, let us see how the I o k can be represented that we have already representing when we are discussing about the hidden layer computation.

(Refer Slide Time: 14:14)

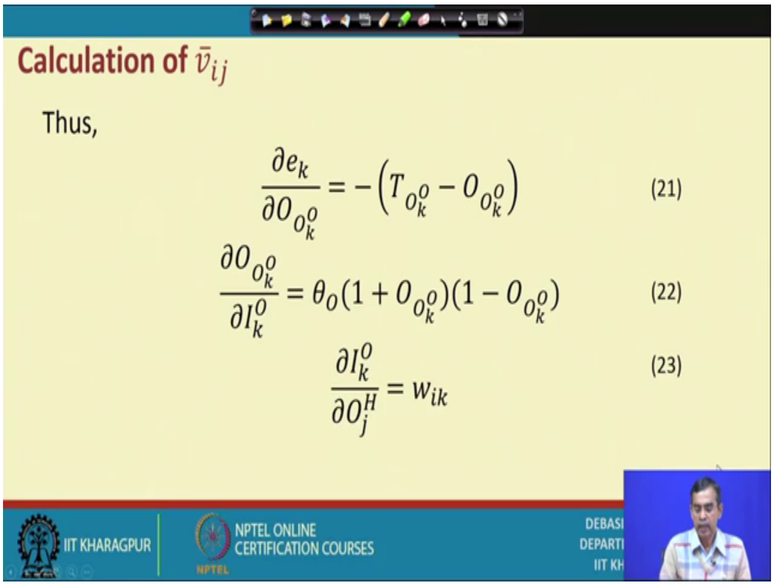


So, hidden layer computation says that this is the I o k or this is the basically input layer combination of the output, right. So, input combination computation in the output layer computation that we have discussed. So, it is I k u, I k o; I k o it can be expressed this one. Once knowing this I k o and then we will be able to calculate that is del a i del O o k, I k by del I k o.

Now, so, so this basically expression is useful for the second differentiation in the chain rule for the v ij this expression is required to know the third differentiation in the chain rule and this is the expression that is required for the first differentiation in the chain rule and finally, this is the expression that is required for the last difference in the chain rule.

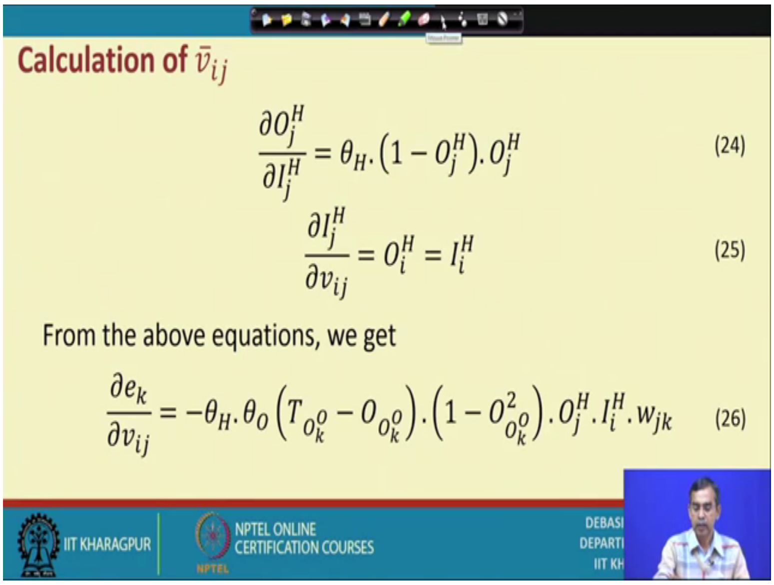
So, all the expression that can be obtained can be used to calculate the final value of del e k by del vj.

(Refer Slide Time: 15:14)



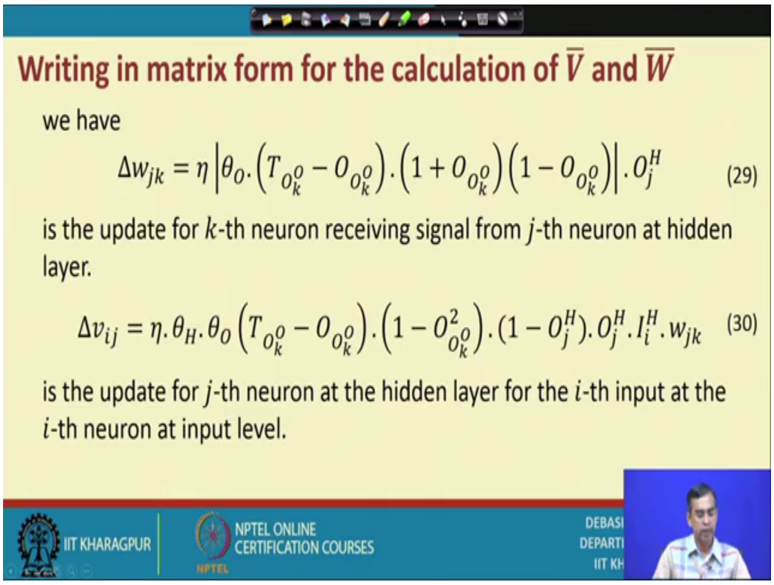
Now, here is the total composition it is like this. So, del e k del O o k can be obtained from the first rule first what is called the e k versus the output this one and so this one and this del O k by del I k can be obtained from the input of the output layer computation and this is basically the output of the hidden layer combination and.

(Refer Slide Time: 15:49)



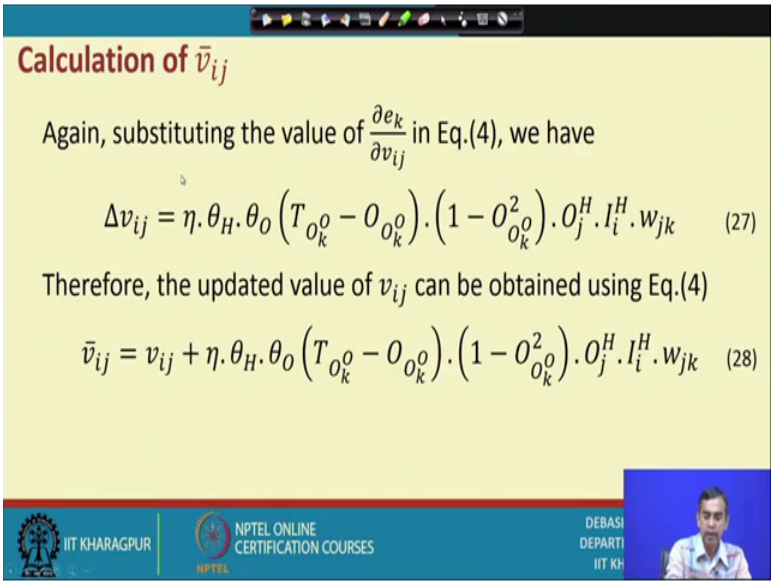
And, this is basically the computation from the this is basically the computation this is basically computation from the hidden layers and finally, this is the computation with respect to v ij. Now, all this expression can be obtained and then finally, putting all the calculations in the chain rule of differentiation ultimately we will be able to calculate this del e k by del v ij and which takes this form like this. So, this is the final form represented with the calculations of all the differentiations for the different layers computation.

(Refer Slide Time: 16:34)



So, this basically the way and then once we know this values del e del v del e k by del v ij we will be able to calculate the updated value, updated value v ij.

(Refer Slide Time: 16:37)



So, finally, using the chain using the delta rule of steepest descent method we will be able to calculate this is the del v ij. Once the del v ij known then the modified values of the weights from the i-th to j-th neuron can be expressed this one.

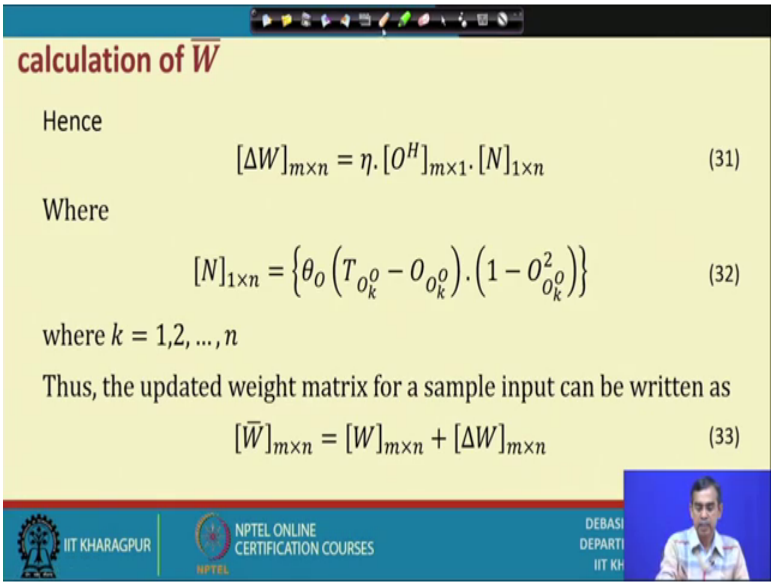
So, we have learned about with respect to some training input then how the v ij and the w ij can takes it change values. Now, this change values is basically following the steepest descent method that mean it will decide the next value so that it will minimize the error.

So, this is the back-propagation algorithm which follow the steepest descent method are there now. So, these methods that we have discussed about calculation of the differentiation values at every neuron in the network, but at a every neuron network if we do one by one and if there are large number of neurons in the network then it is computationally infeasible.

So, in order to make this I mean sorry address this problem so, there is a method by which the entire thing can be expressed in the matrix representation. Now, here so far the matrix representation is concerned so these are the calculation can be represent in one matrix these are the calculation can be represented on matrix and then finally, whole the w can be represented one matrix. So, this is for the w ijk similarly, for the v ij.

So, this is the one matrix representation this is the another matrix representation and putting all the things together a compact matrix representation can be obtained. I will not discuss about details computation, but at the final result I will explain that how this matrix form can be obtained for the entire w and v matrix there.

(Refer Slide Time: 18:44)



So, this is basically a little bit careful calculation a lot of steps if we can follow, you will be able to derive into this one where O H in our previous discussion we have discussed that it is basically the output of the hidden layer. So, it is in the form of matrix m cross n and n is the one matrix of the size 1 cross n, where n matrix is there in the previous expression that is there. So, del W m cross n that means, it will take care for any w jk from any j-th neuron in the hidden layer to the k-th neuron in the output layer.

So, this basically gives the matrix representation of the error changes or updated values of w j, that means, this will take all the updated values all the set, but for this we need only the output layer and then is multiplied by this form for every neuron. So, this can be obtained for k equals to 1 to n it is basically one row column matrix like.

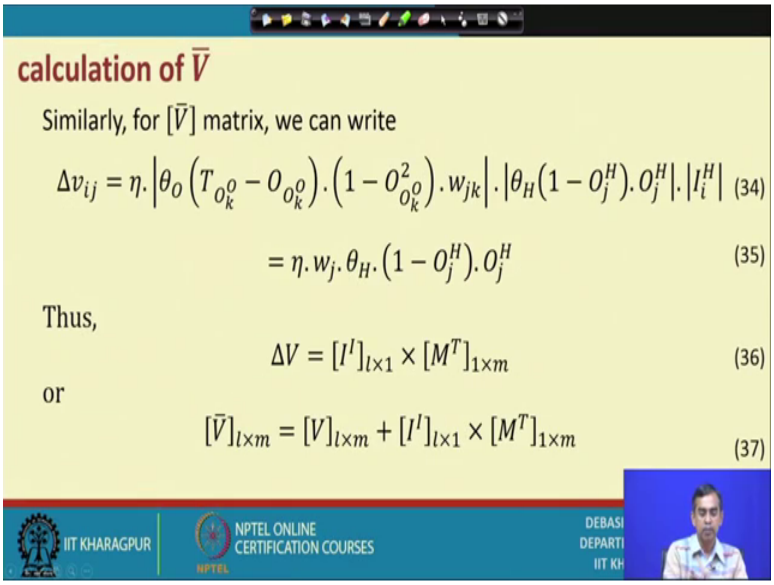
So, this is one column matrix and this is one row matrix and it is a product of row matrix and column matrix for all neurons that is there in the output layer in between hidden and output layer and it will calculate the w matrix in between hidden layer and output layer. So, this is basically is a compact matrix representation in terms of two matrix O H and then N, where N is this one that del W matrix can be obtained.

So, this is the idea about del w matrix, once the del w matrix is known we can know if this is the w matrix at any instant and this is the updated matrix according to this formulation then the updated matrix W bar m cross n can be obtained using this one. So, if it is known to us knowing this one we will be able to calculate what is the next value.

So, this is the concept and here I can say again that simple matrix calculation wants the different values of the neurons in different layers are known we will be able to find it quickly and then this matrix can be obtained. So, this way the network can be learned for the next matrix.

Now, this is the case of the W matrix the similarly, the V matrix also can be expressed.

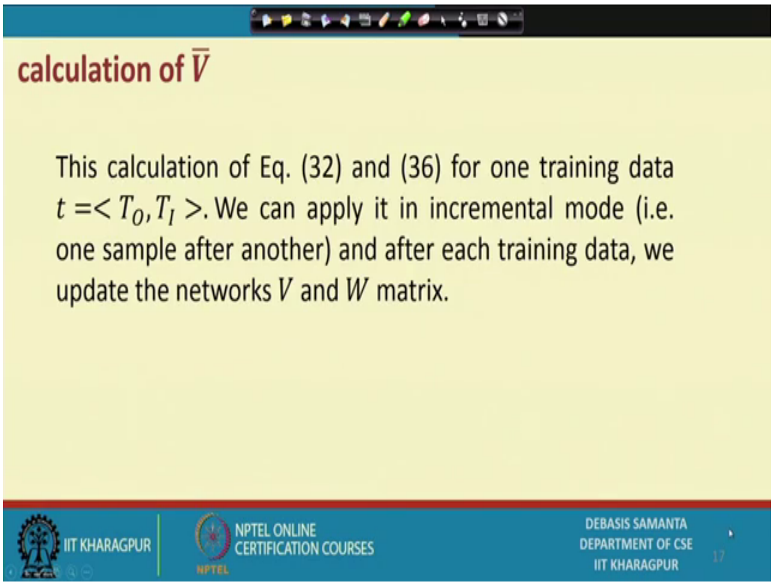
(Refer Slide Time: 21:12)



In case of V matrix as you have already discussed about this is the v ij calculation delta v ij which can be represented in this form. So, it is basically this one and this one like. So, this basically input I l and this is basically M T, M T is this one. So, del v ij can be expressed this way and finally, this is the updated matrix V just like a W bar and this is the current one and this is the implementation one and we will be able to calculate it.

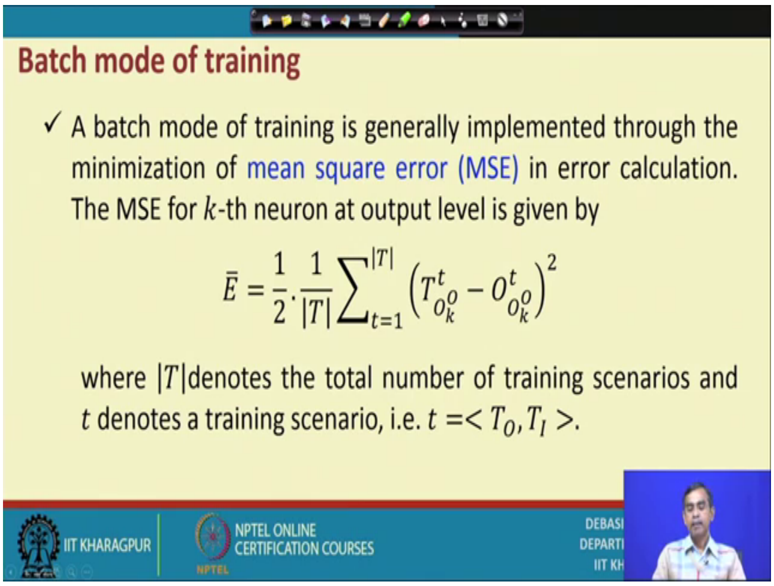
So, there are some steps has been jumped here so that we can give finally, so finally, this is very important one expression that if we know this matrix at the moment and if we know the input to the system and M T matrix which is basically this is the M T matrix M T matrix for all neurons in the k-th layer then we will be able to calculate the V bar matrix. So, it is just a matrix calculation so for the calculation of updated matrix is concerned. So, both W and then V matrix therefore, can be calculated using simple matrix product operation.

(Refer Slide Time: 22:30)



Now, so, this way for the different input the output if we know and then we will be able to calculate the V and then W matrix there.

(Refer Slide Time: 22:35)



Now, here is idea about it that we have considered about training data. In a training data there may be a large number of inputs and then outputs. So, all the training data can be processed in a different way, because if the training data is very large then we can apply this training data in the batch mode. So, we can basically consider out of the entire training data a sub set of training data. Then this subset can be applied and then it can be tested and then the V and W can be decided. Then some other subset can be used for the test data. So, we apply it and then calculate this one and then again we will see that how much error is there and if we repeat the same procedure again and again then we will be able to calculate the errors and then finally, the V and W the final form.

So, this is called the batch mode of training. There are many training the strategy is known to us like say cross validation method or is a k fold validation method like this, but those things are not possibly discuss in this in this discussion in this lecture.

Anyway so, I have just given an idea about how a network can be trained. So, that it can learn the network parameter V and W in this case the same idea which basically we have followed to learn the V, W can be extended to learn the other parametric values like say number of neurons in the hidden layer number of hidden layers in the hidden layers, number of layers in the hidden level then the transfer function.

We have considered that these are the transfer function that we have to follow in each layer, but that can be also can be updated the different transfer functions for different layer and then again see for which transfer function values it gives the better result that can be decided finally, after a lot of trial and error method.

So, the procedure it is there, but the programming environment which is required to find the good network to solve your problem for a given training data is seems to be very tedious job, but with the help of tool it is very simple job on. So, in the next lectures we will discuss about the different tools which are available which can be used to solve many problem using the different soft computing approaches like fuzzy logic, the genetic algorithm and neural network.

So, in the next in the next lecture we will discuss quickly about the different tools that is available in our so far the different tools that is available and how to use the tools all the different problems, ok.

Thank you very much.