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

Lecture - 20
Practical Consideration in Back Propagation Algorithm

In the last class, we had considered the Back Propagation Algorithm derivation from mathematical point of view. And today we are going to see some of the practical aspect of back propagation algorithm. Now, you have seen in the last class that the local gradient that we are having for the output as well as for the hidden layer neurons. That local gradient is very much dependent upon the derivative of the activation function, because it is proportional to phi dash of v j.

So, this necessarily means that phi dash v j should be computable; that means, to say that one of the fundamental conditions. That we are imposing for back propagation algorithm to work is that, the activation functions that we consider must be continuous and they should be differentiable. Unless we have that the back propagation algorithm cannot work, because in such cases phi dash v j will simply become undefined phi dash v j will not exist, if that is not the case.

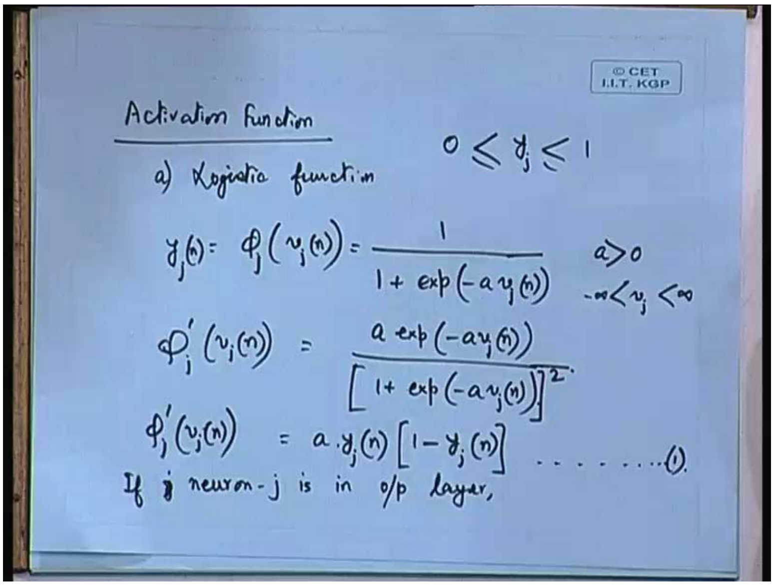
So, coming to the question of activation functions to be used in the back propagation algorithm, certainly the activation functions like McCulloch and Pitts model or the signal function they are ruled out. Because, they are necessarily becoming discontinuous function at the point v equal to 0, the function becomes discontinuous, if we are considering any one of those kind of functions.

So, instead we should again concentrate our attention to the continuous activation functions for which the typical examples, that we consider where the sig model non linear functions. And for sig model functions the two kind of modules that we are taking or the logistic sig modal function and the tan hyperbolic sig modal function. Logistic is in the range of 0 to 1, where as the tan hyperbolic is in the range of minus 1 to plus 1.

So, let us consider first the case of logistic activation function and we will derive, that what is the local gradient that we can compute considering a logistic function. And then, we will go over to some of the other practical considerations, like what is the effect like every time we are presenting any neural network algorithm. We have to debate upon the point, that what should be the right kind of choice for the etas that is the learning range.

We have been telling about compromise that it should be too large, it should be too high, but some research efforts were put in during the late 80s whereby one can make the algorithm stable, as well as cause is a quicker convergence rate. So, that is that will be discussed immediately thereafter and then, we will try to compare about the two modes of learning, that is the batch mode and what you can call as the sequential mode of the learning for the case back propagation. So, these are that thing which I intent covered in today’s class.

(Refer Slide Time: 05:14)



So, let me first give consider the activation functions for back propagation network, in which I am going to consider the example of the sig modal function and especially the logistic function. So, we consider that as the phi function we have the logistic function defined as follows, since we are considering the neuron j. So, for the neuron j the logistic function phi of j as a function v j n is going to be 1 by 1 plus exponential to the power minus a times v j of n.

Where, you know that as before a is controlling the slope of the activation function and v j n is nothing but, the induce local field. That is, summation of the inputs times the weight, weighted summation of all the inputs will be the v j. So, the definition remains the same only, in this case certainly the conditions that we are imposing is that a is greater than 0. And v j that we are considering can be anywhere in the range of minus infinity to plus infinity, we are not putting in a restriction on the bounce of v j in this case.

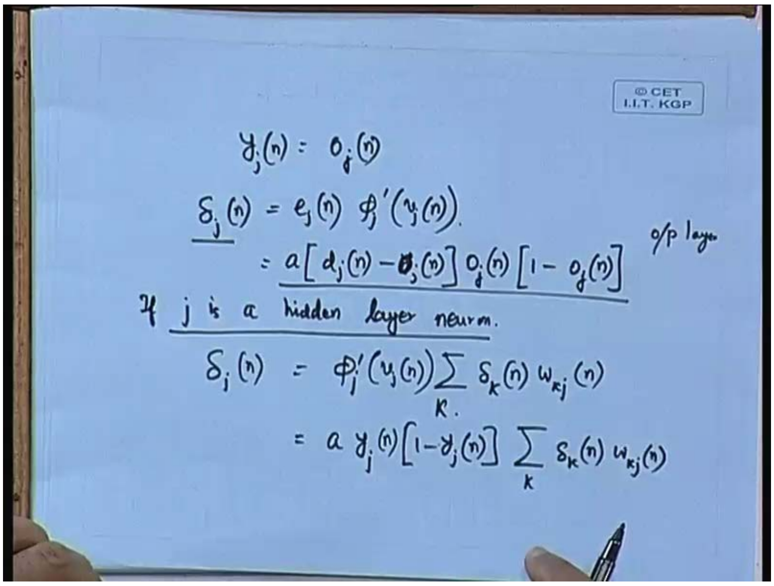
Now, if we consider a logistic function like this, in that case we can take the derivative of this logistic functions. So, let us compute this phi dash j v j n, which means to say that what we have to do it, is to take the derivative of this function with respective v j. So, that is going to be fairly simple and easy enough, what will happen is that the denominator term will now become a square term.

The denominator will become a square term and in the numerator we will be getting the differential of the denominator term. And then, the numerator term which is 1 product of that, so ((Refer Time: 07:32)) the differential of the this term only will be there. Which means to say, that we are going to get the derivatives as a times exponential to the power minus a v j n by 1 plus exponential to the power minus a v j n this term square.

So, this is going to be the derivative in fact, this phi j that we are writing can also be written as y j, because that is going net output. So, if this is y j in that case 1 minus y j if we consider 1 minus y j will become exponential to the power this term by 1 plus exponential of this term wholes square. So, no by 1 that is exponential of this by this, so that is going to be 1 minus y j in fact, this whole thing could be expressed as a times y j n multiplied by, so this also is y j n, so a times y j n times 1 minus y j n.

So, this is going to be our equation number 1, so this is phi dash j v j of n which is equal to this. So, this we are calling as equation number 1 for today, so now for neuron that is located in the output layer.

(Refer Slide Time: 09:36)



So, if neuron j is in output layer in that case we can simply write that y j n is equal to o j of n o j of n being the output. So, y j is the final output and in that case the local gradient that is delta j n will be equal to what e j times phi dash j n, so this is e j at iteration n times phi dash j v j n. And what is phi dash n, if we are considering the logistic function.

Student: ((Refer Time: 10:16))

So, first of all that e j n could be expressed as simply.

Student: d j n.

D j n.

Student: minus y j n.

Minus y j n that is right. So, it is d j n minus y j n that is going to be e j n not y j n, because we are considering y j n to be the output only. So, which is o j n, so it is d j n minus o j n times it will be y j n into 1 minus y j n, that is what we had already derived. So, this is going to be instead of y j o j we can write, so it is o j n into 1 minus o j n please verify is it correct.

So, this is the phi dash j term that we writing in fact, phi dash j will be a times o j n into 1 minus o j n and e j n is going to be d j minus o j n. So, this is what we got is the correct expression for delta j n that is the local gradient. And if j is a hidden layer neuron, in such case what will be the expression for our delta j n, delta j n is will be equal to what remember our discussion from the last class.

Student: ((Refer Time: 11:52))

Phi j n or phi dash j n.

Student: phi dash j n.

Phi dash j n multiplied by.

Student: ((Refer Time: 12:01))

Summation of what, summation of all the delta terms the weighted summation of the delta terms in the output layer and that weighted summation will be over k, where k is the output layer neurons. So, k is the index for output neuron, so what we are going to have is phi dash j; that means, to say this neuron j which is under consideration v j n times summation of delta k in w.

Student: k j

Student: K j

That is correct delta k n w k j n and this summation is over k. Simply what we can substitute is, that instead of phi dash j v j n, we can write it as a multiplied by in this case y j n only we have to write. We cannot write o j n, because o j n is the output, but in this case y j n will be the output of the j th neuron, the hidden neuron. So, we have to write here y j n only into 1 minus y j n, so that is the term that we are writing in place of the derivative of the activation function.

And this summation of the weighted summation of the all the local gradients of the output will remain as before. That means, to say that it will be summation of delta k n w k j n, where it is summed up over k and this is what the case of hidden layer neuron. Whereas, for the case of the...

Student: Output layer.

Output layer, this is the expression that we have got, this is for the output layer. Whereas, this for the hidden layer, anybody having any doubts confusion pertaining to this no. So, now let us look at this, that after all we have derived our expression for phi dash j n, phi dash j. So, let us have a look at this expression ((Refer Time: 14:26)) in fact, we have computed phi dash and phi dash in term is controlling our delta j, that is the local gradient.

Now, first of all that what it going to be the range of y j according this activation function can anybody say, what is the range of y j.

Student: plus 1 minus 1.

Plus 1 to minus 1 everybody you agrees.

Student: ((Refer Time: 14:52))

0 to plus 1, 0 to 1 it is, because you see that it is very easy to see that, you do not have to remember it. Even, if you just simply have look at this expression, you can see that what is the bounce on this term v j could be minus infinity to plus infinity as a say. Now, if I considered the case of v j n tan into minus infinity. Then, it is exponential to the power infinite, which means to say other the denominator term is becoming infinite.

So, it is 0 y j n equal to 0, where as the other extreme as when v j becomes infinity tends to infinity. In that case it is exponential to the power minus infinity, which means to say the this term is 0, this exponential term is 0, so it is 1 at the output. So, here the way we have written a formula our y j that is to say considering this kind of logistic function our y j s bound is going to be in the range of 0 and 1.

So, certainly you all agree about the bounce of y j now if I ask you, that you try to plot consider this expression one. And try to find out that how this phi dash j function changes with the y j definitely phi dash j is dependent upon y j. So, how is dependent, where are we going to have the maximum value of phi dash j.

Student: ((Refer Time: 16:43))

1 by 2 yes; that means, to say that when y j equal to 0.5, in that case we are going to get the maximum value of phi dash j. And what is the value of phi dash j, when y j equal to 1.

Student: 0.

0 simply substitute here it is 0, what happens when you take y j equal to 0.

Student: 0.

Then also it is zero, so; that means, to say that for y j equal to 1 of a y j equal to 0, this equation one gives as value of 0 for the derivative. So; that means, to say that in those places the derivative of the activation function becomes 0 and when we have y j equal to 0.5. That means, to say that exactly at the mid range of y, there we are going to have the value of phi dash v j to be the maximum.

You can simply differentiate this phi dash j with respective y j and say it yourself by equating the derivative to 0, you can find out the point of maximum is certainly becoming at y j n equal to 0.5. So; that means, to say that the phi dash v j is definitely being maximum at the mid range of y. That means to say, that the weight adjustments that we are going to do that is to say ultimately we are going to do a delta w j i based on this computation is not it.

Based on phi dash j or based on the delta j, we are going to find out the delta w k k j. And that weight changes those weight changes will be maximum when we have y j n equal to 0.5 or rather to say at the mid range of the output values the weight adjustment is becoming maximum. In fact, this is one thing about it which definitely adds to the stability of using such kind of logistic functions, in the back propagation network.

Now, very similarly way we can show that instead of taking a logistic function, if I had taken a tan hyperbolic function. I could have derived a very similar expression to that only thing is that, there instead of taking 1 by 1 plus exponential to the power minus a v j n I should have taken tan hyperbolic of a v j n. And then, I could have computed a very similar expression.

So, I am not going into the computation of that, they are should be something that should be left to the students, as well as the viewers for them to solve. So, the tan hyperbolic case you can consider and solve it yourself, it would be quite easy following the same lines you can derive the expression for phi dash j and correspondingly the delta j considering j to be the output layer neuron and j to be a hidden layer neuron.

So, that is as for as the activation function considerations are there, which we have now derived out of this. And then, the next point to consider is the rate of learning, because in the expression that we had got as before remember the expression that we had got in the last class about the change of weight, that is delta w k j, delta w k j is dependent upon what terms remember it is dependent upon eta. So, that equal to eta times delta j times.

Student: ((Refer Time: 21:02))

Times y, so now, what we have to do is that definitely; that means, to say that it is dependent on eta. Now, we have been discussing about etas, all that time in this case you see that the back propagation algorithm, that we have got in this case is definitely and approximation to the steepest descent problem, this is also a solution of this steepest descent only. But, in this case it is an approximated version of the steepest descent.

And approximated version of the steepest descent means, that in the case of a perfect steepest descent we know that, see ultimately what is that is happening in the case of the steepest descent algorithm, we start with some value of weight vector. So, we are in some m dimensional space initially, we are in some point in the m dimensional space with the initial set of weights.

And then, we are going to move in the weights space m dimensional weights space and we will follow a trajectory that ultimately takes us to some value of weight, which we have been calling, so long as the w star vector. And w star vector is something, where it gives the minimum value of the cost function that we have been considering. And the trajectory in the case of a steepest descent algorithm is going to be a smooth trajectory.

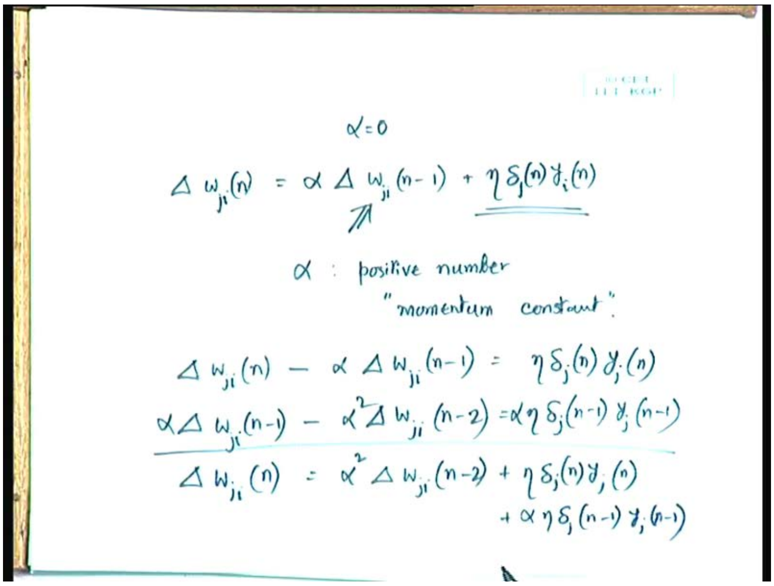
So, since the back propagation network back propagation algorithm is an approximation to the steepest descent, certainly that trajectory that we are going to follow out there in the m dimensional space is not going to be a smooth trajectory. The trajectory is very much dependent upon eta and as our discussion has gone in the case of the single layer perceptron also.

That, there is trade off that is definitely going to exist you make eta too small. Then, what happens is that the learning is getting too small, but the trajectory will be smooth and you will not run the risk of having a and instability. The system will be always stable, whereas if you a making eta to be too large, then the learning definitely will be faster, but the risk is that we can it can lead to then instability.

So, now there were a such as to find out, that if there could be any solution to this. That means, to say having a faster learning and also a better stability condition or rather to say it guaranteed stability condition is that possible. So, now, what happens is that in this effort some researches were done and as I was telling you that most of the developments in the multilayer perceptron and so to say about the back propagation algorithm, say etcetera where done in the 80s.

And it was the book by Rumor Hart in 1986, the book on Parallel Distributed Processing which we had mentioned the few class back. In that book it was presented that one of the solutions to have both together; that means, to say stability as well as faster learning is to add what is called as a momentum term to the learning and using the momentum term which was actually suggested by Rumor Hart.

(Refer Slide Time: 25:19)



There the delta w j i, that we are going to have delta w j i at iteration n is going to be equal to Alfa times delta w j i n minus 1 plus our usual learning term that exist. That means, to say eta, eta being the learning rate times delta j n which is the local gradient times y i n, y i n a bring the input to the neuron j. So, here this term is very familiar to us, this is the normal learning term that way always have.

Whereas, this is something which is new to us, way have not come across this term earlier. So, this is an addition and just see that here, what is the significant are that; that means, to say that the change of weight that you are doing is in some way proportional to the change of weight that you had done in the earlier iteration. That means, to say in the n minus 1 at iteration whatever delta w j i you had consider, you are take a proportion of that to be added up.

So, in discuss Alfa is going to be a constant in fact, Alfa is normally take in to be a positive number. And Alfa is refer to as the momentum constant in fact, y is it called momentum, if you spend a bit of time on it I think it is very easy to understand. That means, to say that over and above your normal leaning, you can a put to have eta to be small. Because, we know that eta small is good from the stability considerations.

So, despite having small eta we can have quick a learning, as if to say giving it a push; that means, to say that applying and external force. And force as we know will be causing a momentum, so definitely there is every reason to call it as a momentum term. So, which is added to this delta w j i now this in fact, this equation that we have got over here could be return in terms of this.

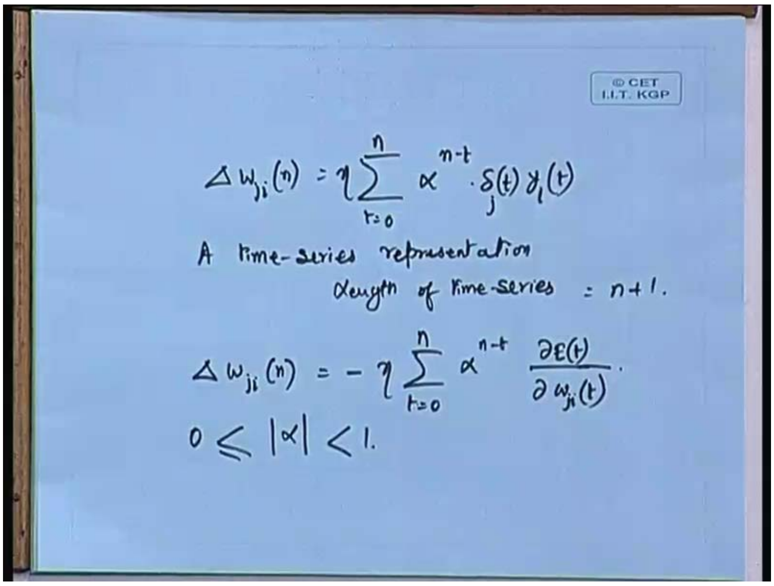
That delta w j i n in the form of a difference equation, if we write then we have to write it has delta w j i n minus delta minus Alfa times delta w j i n minus 1 which should be equal to eta times delta j n y j n. Which means to say, that very similarly we are going to have delta w j i n minus 1 minus Alfa times delta w j i n minus 2 is equal to eta times delta j n minus 1 y j n minus 1 is not it, if we had written the same expression for the earlier iteration, the difference equation for earlier iteration would have been like this.

So, which means to say that if somebody it tells me like this, we can go over is not it like this we can go on for n minus 2, n minus 3 and so on up to n minus n that is w j i 0, which means to say that initial. We can go there and if we have to eliminate all this intermediate terms, let us say that if we have to eliminate this delta w j i n minus 1 term. Simply what we have to do is to multiply the second equation by Alfa is not it.

So, if I multiply this whole equation this second equation by Alfa, then what I am getting is Alfa delta w j i minus here it becomes Alfa square. So, if I add of this two on then left hand side, it is delta w j i n minus Alfa square times delta w j i n minus 2. Which is going to be equal to eta delta j n y j n plus Alfa times eta delta j n minus 1 into y j n minus 1. We can shift the delta w j i term on the delta w j i n minus 2 term on the right hand side from left hand side we can take it to the right hand side.

And in that case, delta w j i n will become Alfa square delta w j i n minus 2 plus it is going to be eta times delta j n y j n, here it is going to be eta Alfa plus eta Alfa this term. In fact, if we keep one doing it, then we have expressed it in terms of n minus 2, but ultimately if we add of everything n minus 2, n minus 3, n minus 4 and all that in that case the delta j i n is going to be expressible as a time series. So, you can verify this yourself by proceeding with the difference equation solving approach.

(Refer Slide Time: 31:32)



That we are ultimately going to get delta w j i n to be equal to eta times summation t equal to 0 to n Alfa to the power n minus t into delta j t times y i t. This is something that is going to express it, eta is a constant eta is not varying with iteration, Alfa is also constant. But, here y i I had to put Alfa term inside the summation is because it is Alfa to the power n minus something n minus t. And this case t is quantity that is changing; that means, to say that considering t equal to n it is Alfa to the power 0.

So, it is delta j n eta delta j n y i n the faster, then t equal to n minus 1 if you look get it form a reverse way t equal to n minus 1 means it is Alfa to the power 1 delta j n minus 1 y i n minus 1, that term that viewer a writing out here for the second equation. And like this it is a summation series, that is ultimately going to be, so this is a time series representation of delta w j i s n. So, this is a time series, so this time series representation and what is the length of the time series, length of time series is in this case n plus 1.

Now, simply what we can do is that we already know that delta j times y i is going to be equal the negative of dau e dau w j i is not it the rate of change of the error function with respect to the cost function, with respect to then change of weights. So, I can express this same equation as delta w j i n equal to minus eta times summation t equal to 0 to n Alfa to the power n minus t into dau E t dau w j i t.

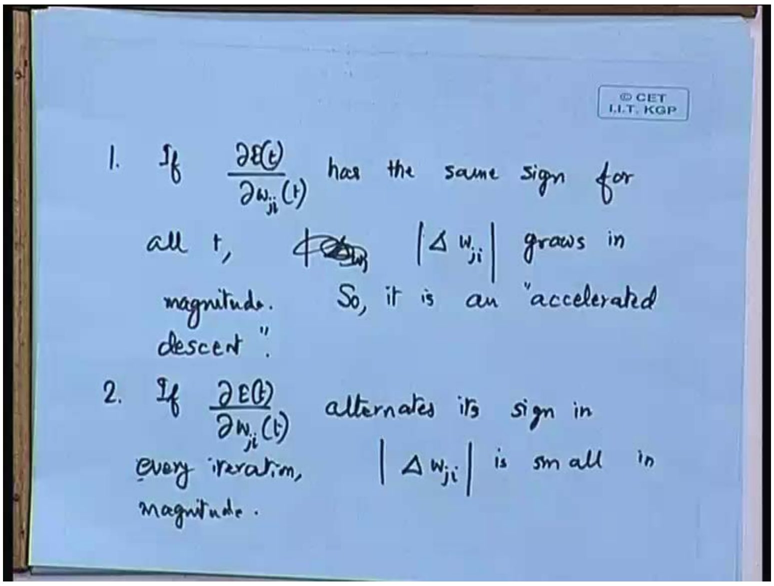
So, this is going to be our series representation, now we can spend little bit up time on it and find out about the convergence aspect of this equation. You see, we are considering the Alfa to be a positive quantity, now the thing is that we can take we never said that we have to put the Alfa term to be a Alfa term means greater than 1 or Alfa term is less than 1 that one be have not real if specify.

So, firstly that here we have to know one consideration which should be therefore, us is that, the Alfa term that we have to take should be definitely within 0 and 1. Because, one thing is there that you can make out from this series itself, that if I put Alfa to be greater than 1. Then, delta w j i term will be uncontrollable it will increase, because as the power of Alfa increases if Alfa is greater than 1. Then, the contribution this terms which are getting summed up will increase and definitely that will be uncontrollable.

So, that is not some change that be looking for in this case, so definitely we restrict the range of Alfa to be this, that within in 0 and 1. Now, if it is within 0 and 1, then one thing is very showed that this expression that we are getting Alfa to the power n minus t into dau e dau w j i, this is surely going to be a convergence series. Because, with Alfa less than 1 modules of Alfa being less than 1, it is going to be convergent and there is some consideration that we can have on this dau E dau w j i term.

Now, this term definitely we are having n plus 1 such term is not it and they are getting added up with the value this Alfa to the power n minus t. Now, if it, so happens now Alfa is positive in this case, now in this case if it is dau E dau w j i is having the same sign in every iteration. For every t if it is having the same sign, in that case the modules of delta w j i is going to be a higher value is not it although it will be convergent. But, modules of delta w j i grows magnitude.

(Refer Slide Time: 37:44)



So, I can say that if dau E dau w j i has the same sign for all t, then the modules of delta delta w j i. That grows in magnitude correct, look at this expression to confirm this ((Refer Time:38:20)) every dau E term is having the same magnitude. So, either it is increasing in the positive direction or it is increasing in the negative direction, this is the summation term that is existing.

So, grows in magnitude delta w j i growing in magnitude means what, that it is it refers to an accelerated descent. So, it is an accelerated descent it is descent to no doubt, it is following as steepest descent, but in an accelerated with we are coming down after somebody has given as a said push. So, we are coming down little faster and the other thing could be the other extreme could be that dau E dau w j i is alternating it is sign with every iteration.

What happens in that case, in that case if dau E dau w j i is alternating it is sign with every iteration, then mod of delta w j i is going to be small in magnitude is not it. Because of one term is tending to adapt in the positive direction, the next term is going to subtract it in the negative direction. So, addition subtraction, addition subtraction continuously happening, so that restricts the mod of delta w j i.

Now, if it is something in between; that means, to say that few terms are having same sign, few terms are going to have opposite signs. That means, to say it will be within this to extremes, but one of the extremes is going in the form of an accelerated descent. And the other extreme that if dau E dau w j i alternates it is sign in every iteration in that case delta w j i mod of that is small in magnitude and this certainly has got a more stabilizing effect.

So, what we have got as the momentum term is not something which is risky in that sense. Even if it is an accelerated descent ((Refer Time: 41:11)), but still you remember here that because Alfa is restricted within 0 and 1 we are not going to have an un bounded increase of delta w j i. So, definitely it is stable, so with stability it is possible to for us have an accelerated descent.

That of course, depends on what your derivative of in this e term is with respective w j i. And the other possibilities that which is small in magnitude and if is small in magnitude then I am definitely it is stabilizes. Now, the momentum term has got two f x, not only can it contribute to faster learning as we have just now shown. That means, to say that possibility of some form of an accelerated descent.

It also helps in avoiding the local minimum, because one thing is there you know that again going back to our original discussion. That the back propagation algorithm is an approximation to the steepest descent and in a steepest descent the convergence to a local minimum is guaranteed. Convergence to a minimum in fact, is guaranteed. But, whether that is local minimum or global minimum is not very clear to us.

Because, if the surface if the cost functions are face in the multidimensional space is such that it could be having local minimum as well as global minimum. Then, at least the steepest descent can bring us to a local minima and trap the solution in that manner. Whereas, in this case if we are adding a momentum term, the momentum term has got some effect in avoiding the local minimum.

In fact, more on the avoidance of local minimum we are going see when we discuss about the aspects of simulated annulling. So, any doubts that your having related to the learning rate aspect, what we have just now presented in terms of it is momentum, parameter any queries related this yes please.

Student: ((Refer Time: 43:42))

First we have propagating as.

Student: ((Refer Time: 43:47))

Right.

Student: ((Refer Time: 43:49))

After every back propagation yes, we are first doing a forward pass, then we are doing a backward pass and the backward pass is allowing us to calculate all the delta j's. And then, after calculating delta j's we are calculating the delta w j i, so only thing that we have modified this that the ((Refer Time: 44:21)) algorithm that we have been talking, so long is as if taking Alfa equal to 0, in this case you just substitute Alfa equal to 0.

Then, it is leads to the un modified back propagation algorithm. That means, to say un modified weight updating, in discuss what we have simply done is that instead of doing the weight modification simply according to eta delta j i y i we are adding a momentum term to it. So, that we are doing after the backward pass is done, that is only during the change of weight delta w j i that we are introducing this term. That certainly does not affect our forward pass of the backward pass.

Student: ((Refer Time: 45:06))

At output nodes as well as the hidden nodes be a doing the delta j computations.

Student: ((Refer Time: 45:18))

You see, this is this delta w j i that we are doing, we are doing it connection by connection. That means, to say that when the forward pass is done and then backward pass is also completed, when we are completing one of the nodes, then we are computing it is corresponding change of weights. We go to another node compute it change of weights.

So, we update the weigh as we go from one layer to another stating with the output layer, to the next layer, to the next layer, like that as we go on we will update the weights every time. So, we have to allow for that much of combination, but certainly since we have doing this competitions certainly I am certainly does not add to anything like a instability or anything like that.

We adjust taking this time, there is nothing different from the original back propagation algorithm except for the fact that computationally you are introducing one more momentum term, that is all any other doubts.

Student: ((Refer Time: 46:37))

You see, that is just to show you some steps about this, so this one.

Student: ((Refer Time: 47:02))

So, this is the difference equation involving n and n minus 1, likewise I can have a difference equation involving n minus 1 and n minus 2. So, which is going to be what Alfa times this minus square times this equal to Alfa times this. So, now if I add up this two, then what happens. Then, if I simply adapt, then we are getting delta w j i is equal to Alfa square times this plus this term.

Student: ((Refer Time: 47:33))

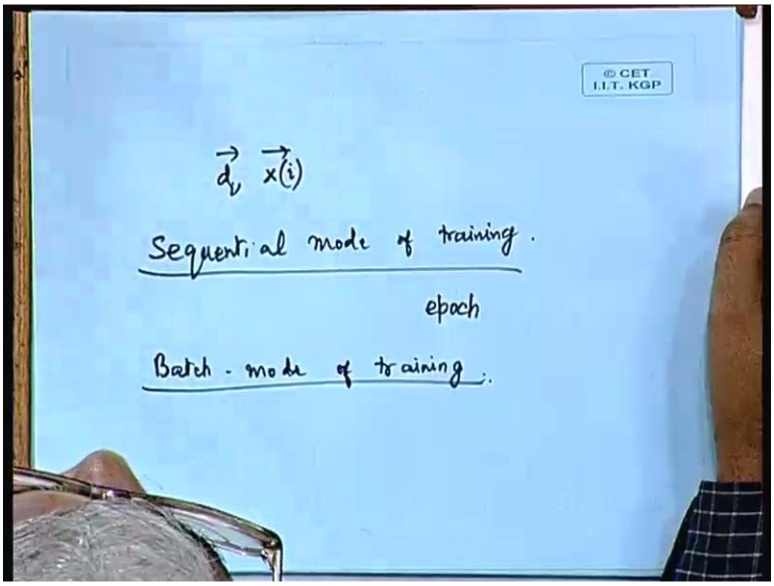
After this, this is I will tell you here what you are getting is delta Alfa eta delta j n minus 1 y j n minus 1. So, you see that this is for two term that I have got; that means, to say that if I had got this for three terms. Then, what would have been the case, then it would have been delta w j i n equal to Alfa cube of delta w j i n minus 3 plus eta terms this Alfa into eta terms this plus Alfa square into eta terms of this.

Now; that means, to say that ultimately you reach what, Alfa to the power n, delta j i 0 and what is delta w j i 0 it is 0. So, this term ultimately does not remain, what remains is the next term that is to say eta delta j y j plus Alfa eta delta j y j plus Alfa square eta delta j y j plus Alfa cube eta delta j y j like that it goes on. And we have to some of all these things together. And that is exactly what we have represented in the form of this time series is it clear to those who had any difficulty in following this.

So, this is about some form an accelerated learning using the momentum term. And now, there is one more debate that to be should initiate, that is to say that whether the sequential learning is good or the batch mode of learning is good. Now, the discussion that we have been having, so long is as if to say that we are presenting one pattern, which we are calling as by the index n.

That means, to say that when we are putting everything within bracket n means that, it is the nth iteration or rather the nth pattern. As if to say that we are presenting one pattern and then, we are adjusting all the weights and then, we are presenting the next pattern adjusting all the weights and so on followed. We will be given training set a training sample is something that we begin with.

(Refer Slide Time: 50:36)



That means, to say that we will be having set of d and a set of x i's, where x i's are going to be the input vector and d or rather to say d i is going to be the desired output. In fact, in this case I can say it is d i in the form of a vector or, so meaning that, since the output is going to be from several neurons I can describe it in the form of vector. So, I am feeding one pattern calculating it is errors at the outputs. And then, making the readjustment of the weights, then only feeding the next pattern.

That is the approach that we have been following and this is called as the sequential mode of training, this is sequential mode of training. Now, as we discuss last time that we defiantly have a fixed set of pattern fixed set of training pattern. And once, the system goes throw the training of one set of training patterns at the complete set of training pattern, when that is completed we are calling that, that is one epoch of learning is over.

And then, what we are doing is that, we are can we stop the network there certainly not, because this is only once we have gone through all the set of patterns. Now, we are going to repeat that epoch again, repeat that epoch means that is again start with the first pattern, second pattern, third pattern with every pattern feeding you update the weight feed the next pattern like that it goes on. So, again we will be complete in the second epoch, then we will be completing the third epoch and so on.

Now, were to stop that is also a question that we should answer, what should be the stopping criteria. But, coming to the question of sequential mode of training we are doing epoch by epoch, now one thing which you could have done in alternative way is that instead of adjusting the weights at every iteration if we could adjust the weight.

Once, all the patterns in the epochs are presented, then adjust the weight only once, in that case the difference would have been that you could not have in this case, the case of sequential mode of training, what is the cost function that you are taking it is simply the...

Student: ((Refer Time: 53:32))

Instantaneous value of the error energy, it is simply the instantaneous value of error energy that we were considering for a sequential mode of training. And for a batch mode of training what we would have considered.

Student: ((Refer Time: 53:47))

Simply the e average, e average that we had discussed few classes back, that would have been our cost function. Which means to say that after computing e average, we should have taken the gradient of that e average and then would have adjusted the weights according to that. Even, that way also we can derive a back propagation algorithms, so there what happens is that, the weights will be updated based on that e average once for all.

That means, to say once for epoch it will be adjusted. Now, the debate that one can come to our mind is that, whether the sequential mode of training or the one that I described just now that is to say the batch mode of training, which one of these to is better. Now, I can ask the students present over here to initiate, this kind of a debate, what is in your mind let me just hear from one or two of you about your feeling what one would you prefer.

Student: ((Refer Time: 55:04))

Sequential is it a kind of consensus in this house, everybody feels sequential one good reason and anyone good reason for choosing sequential and rejecting the batch mode as such.

Student: ((Refer Time: 55:23))

You see, if I try to counter your debate by say that you were saying that with every pattern you have learning and you are contributing that incremental learning. Now, whether you incorporate that learning immediately or whether you differ that learning how does it matter. Because, after all if you are considering e average, that e average also is containing the accumulated errors.

So, ultimately we are learning it is something like this that, whether you read one page and then you try to assess that how much you have read with every page you do that, update your learning. Or other thing is the other alternative is that you read the whole chapter and then only you think that what you have learnt I do not have much of a time to continue with this debate in this particular class.

So, in next class we will continue this debate and find out that between the sequential mode and batch mode, which one are we going to prefer and under what condition. And the second discussion that we wanted to make is that what should be the stopping criteria for that. So, that we will do in the next class.

Thank you very much.