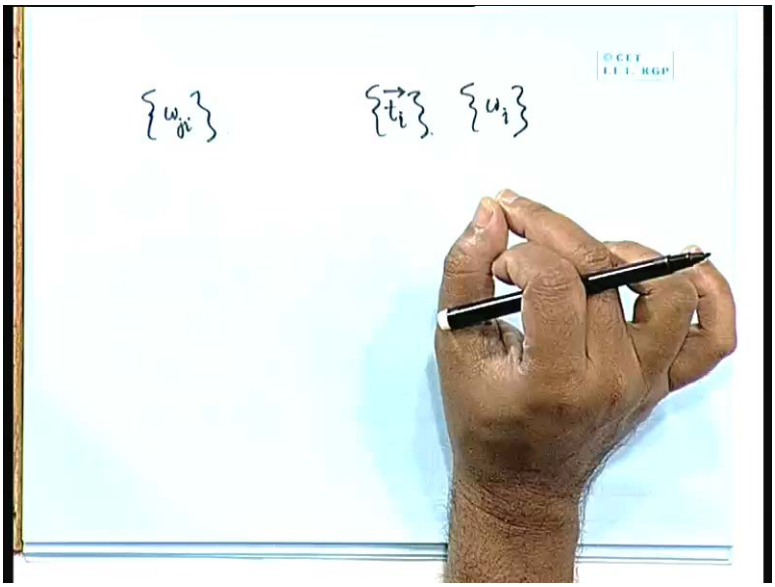
Neural Network and Applications
Prof. S. Sengupta
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 31
Learning Mechanisms in RBF

The topic that we are going to cover is the Learning Mechanisms in RBF. This is the topic, which we have already introduced in the last class and in fact, this is the last topic that we wish to cover under the chapter of the radial basis functions. One thing, which you have noted during our last lecture is that. Whereas in the case of the multi layered perceptrons or single layered perceptrons, as the free parameters what we have is the set of all the w j i's where, the weights are the free parameters rates and the bias.

(Refer Slide Time: 01:44)



Of course, and then we have to adjust this weights and the bias in order to satisfy the constraints of the problem, which is given to us through the form of the training patterns. Whereas, in the case the RBF what we have, as the free parameters is not only the set of w i's which are the synaptic weights from the hidden layer output from the hidden layer to the output layer.

But, also the centrals of the radial basis functions which we call as the t i's, those t i's is also will be regarded as the set of this t i's will be also regarded as a free parameter to the system. Now that means, to say that in accordance with the training patterns, we have to first adjust this centre's of the radial basis functions and when and once that is adjusted,

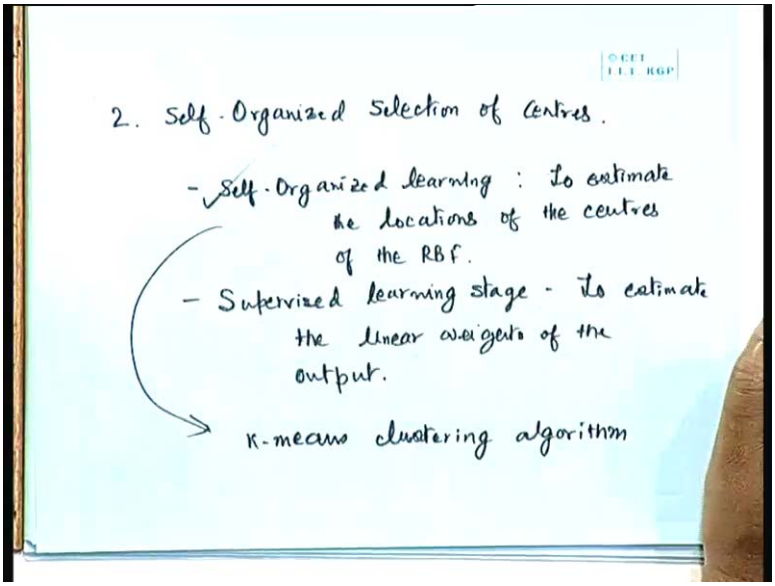
as a second step to it this w i's will be adjusted. So, it is not that everything can be done together, because the adjustments or updating of the w i relatively requires, that at first all the t i's have been obtained.

So, this is a two step learning process, first for the centrals of the radial basis functions and then for the synaptic weight connections. Now, the first case that we had taken for the choice of the centre's was, where we choose fixed centers. In fact, when we are choosing fixed centers which are randomly located, there is no learning mechanism that is involved in it. Only thing that we argued is that yes it is performance can be in doubt in some cases where, we have got dense population of training patterns in one of the regions and we have sparse population of training patterns in another region.

And then, if we uniform this space the centre's of the radial basis functions, then it is performance will not be good. In fact, here because there is no learning that is involved in the t i's and the only learning that we are making it is through this adjustment of the w i's. That is why one requires say very large number of training patterns, in order to achieve a satisfactory level of performance in the radial basis function networks.

So, that is why one has to see for a better way whereby even the learning mechanism should include the proper choice of this t i's. So, the second methodology of the selection of the centre's of the radial basis function is what is known as the self organized selection of centre's.

(Refer Slide Time: 05:00)



Now, in this case two stages are involved, first is what is called as the self organized learning. And in the case of this will be first step, where we estimate the locations of the centre's of the RBF. And then, the second step is the supervised learning stage and this will be used for estimating the linear weights of the output layer. So, we will be we know that how the supervised learning stage in the second one is carried out.

Because, that is our very standard LMS algorithm, which will be used for the second one. But, what is important for us to know is the mechanism of this self organized learning, that is to estimate the locations of the center and this in fact, is done by the K-means clustering algorithm. So, for this mechanism to work what we have to do is to fit a set of training patterns and then, in accordance with the training patterns we have to properly adjust the positions of this t i's.

Now, here intuitively we can first talk about the approach, that what is it that we are looking for. Now, supposing we are having we have selected that there are m 1 number of such centre's of the RBF we have chosen m 1 number of RBFs in the hidden layer. So, we have got t 1 t 2 up to t m 1 as the centre's, let us say that initially we randomly distribute those centre's.

In fact, initially we do not have any proper mechanism to direct that, where exactly are we going to place our centre's initially. So, let us say that it is random to start with and then we will be fitting the training patterns one by one, let us say that we fit the first step in pattern say x 1 as a vector we are fitting x 1. And now, when the x 1 is fit then all these centre's of the RBFs, they will be competing with each other to see that if they are closest to the pattern x 1, that we have fit.

That means to say, that when x 1 is fit then out of the m 1 number of t i centre's that we have got, there will be one which will be closest to it. And there will be m 1 minus 1 others who will be relatively at a distant away from it, although it does not necessarily that there is no question of having a tie. There can still be tie in some cases, where may be that more than one centre's will be having the same distance.

And if that is the case in that case anyone can be arbitrarily declared as the winner. Because, there it would not really matter, that whether we declare out of the computing whether we declare this as the winner or some other centre's winner. But, one of the centre's will be the winner based on the proximity to the training input that you have fit.

Now, what we have to do is that the center which emerges as the winner, what we do is that we move that center closer to the pattern.

We do not exactly make it equal to the position of the input, we just make it closer, why because it is not that a particular center will respond to only one pattern. The center is not really patterned to one particular pattern only, that center may be responding to a group of patterns. So, ultimately what happens is that, if those group of patterns are organized in some manner, then possibly the centroidal position of that group, the center will try to live there.

So, what we do essentially is that once the first pattern let us say x 1 is fit we will be moving the center which emerges as the winner in the computation, we will be moving that center closer to the position of the pattern. And that we do by a simple mechanism of learning, just like the way we did for the earlier case. That means to say, that we choose a particular learning rate and then in accordance with the learning rate we will move it closer.

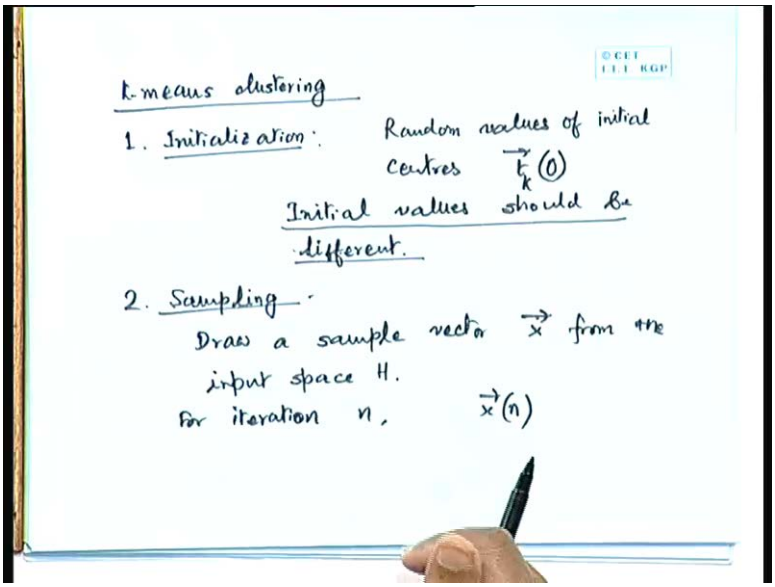
Next, we fit x 2 and for x 2 there may be some other center that emerges as the winner. So, we move that in fact, it is only the position of the winner that is changed all the losers the centre's which are losers in the competition, they remain in the same position we do not update their positions. So, it is a kind of a winner takes all type of an algorithm, where the winning centre's position will be updated.

And then, accordingly in the next iteration when we fit the training pattern for the next epoch. Then, there is a better chance of finding that neuron, which emerged as the winner in the earlier epoch, the chances are that this time it will be even closer to the pattern that we have fit. And then, it will be move even further closer and in fact, there will be a stage when the positions will not change much from iteration to iteration.

And from epoch to epoch if the positions more or less remain the same; that means, to say that has achieved some kind of stability. So, all these centre's that we have chosen have already moved to the centre's of the patterns, which emerges the cluster. So, we will be having several clusters, each one cluster will correspond to one of the centre's. So, that is why this is to be done using what is called as the k-means clustering algorithm.

So, having understood the philosophy of it I think we can present the algorithm, which should not be difficult for you to understand. So, the self organized learning mechanism that we are talking of will be consisting four basic steps.

(Refer Slide Time: 13:29)



The step, so in the first step k-means clustering which will do the self organize learning of the selection of such centre's, the first step is the initialization. So, in this process we choose random values of initial centre's, so we choose t k, where we choose k as a variable as the index. And because, it is the initial estimate, so that is why we are putting here within the parenthesis 0.

So, here the number that we write down under the parenthesis is that is the iteration number. So, the iteration number 0 means it is initial, so we choose some random position, so then accordingly later on this t k's will be updated as t k 1, t k 2 like that. So, the n th iteration we will be calling that as t k n. So, this can be chosen as a random value, but only restriction is that the initial value should be different.

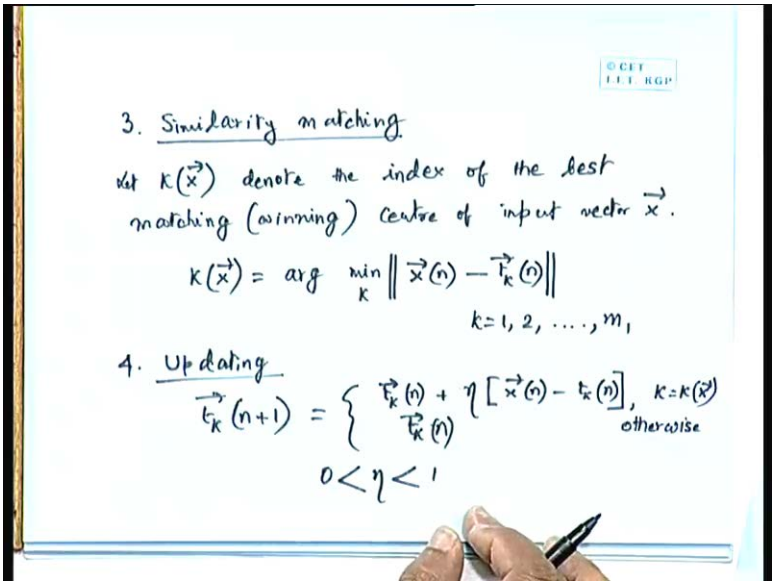
So, it should not be that in the random positioning of the initial centre's, we happen to place two or more than one centre's in the position. That is the only restriction that we imposed that is not allowed, the second step that is involved in the k-means clustering is a process of sampling. And sampling will mean, that to draw a sample vector from the input space, so here what we do is that we draw a sample vector, let us say that is the x vector from the input space H.

Now, this supposing we are in the nth iteration, so for iteration n the sample that we have drawn we can write it as x of n. Even in the nth iteration we have picked up the sample vector x, we will fitting picking the sample vector for every iteration from the input space. Now, next the third step will be what I think you can guess that the third step will be a computation of...

Student: distance.

Distance yes a computation of distance or a computation of similarity in the distance.

(Refer Slide Time: 16:49)



So, the computation of distance or in other words we can call this as the similarity matching. Now, in this case what we do is that, we have to determine that which is the index of the winning sector. So, supposing we denote that k as a function of x, y as function x because the index that is there for the winning RBF position is definitely a function of x that is the input, because if we feed a different input, we can get a different centre's.

So, the index will be different, so the index should always be written as a function of the input that we are feeding, function of the input vector that we are feeding. So, we choose that let k x denote the index of the winning centre denote the index of the best matching as to say the winning centre of input vector x. Now, here we can write that this k x, k x will be mathematically expressed as what, k x will be the argument of the minimum value minimum over k of what the Euclidean distance between x at iteration n and it is difference with t k at iteration n and this minimum we have to find out over k.

That must to say that for all the competing centre's given a pattern x at the iteration n we have to find it is distance with all the competing centre's whose index is k. So, it is the minimum over that k that we find out, so this in effect will give us the minimum distance. But, we are not interested in the minimum distance value as such we are interested in finding out the index of the RBF, which has got the minimum, which is closet to this x n.

So, we take the argument of this minimum function, so that will give us the index of the winning center for the pattern x. So, this value of k in our case can be varying from 1 to m 1, because we have assumed that there are m 1 number of such different centre's that we have chosen. Now, what will be the fourth step, we have to move the yes updating, the fourth step is; obviously, updating where we have to move the winning center closer to the pattern.

So, how we do that the center of the RBF is adjusted according to the updating rule, because we have already got t k n and supposing k is the index of the best matching neuron, so best matching center. So, what we have to do is, that we have to have t k n plus 1, n plus 1 is the updated center position, t k n is the present center position and t k n plus 1 is going to the updated center position.

And where, will t k n plus 1 be equal to t k n that is to say the present position plus we have some learning rate let us say eta again and x n minus t k n. So; obviously, depending upon the distance between these two that is x n minus t k n, we will be giving the vector a push. That means, to say the whole idea will be to move it closer to the pattern x of n.

So, this updating we will be doing only if k is equal to k x, k x is assumed to be the winning position k x is the winning position. So, if the kth center that we have chosen to update happens to be the winning center itself, then we will be applying this updating rule. And if that is not the case what are we going to do I have told you that, what will be t k n plus 1.

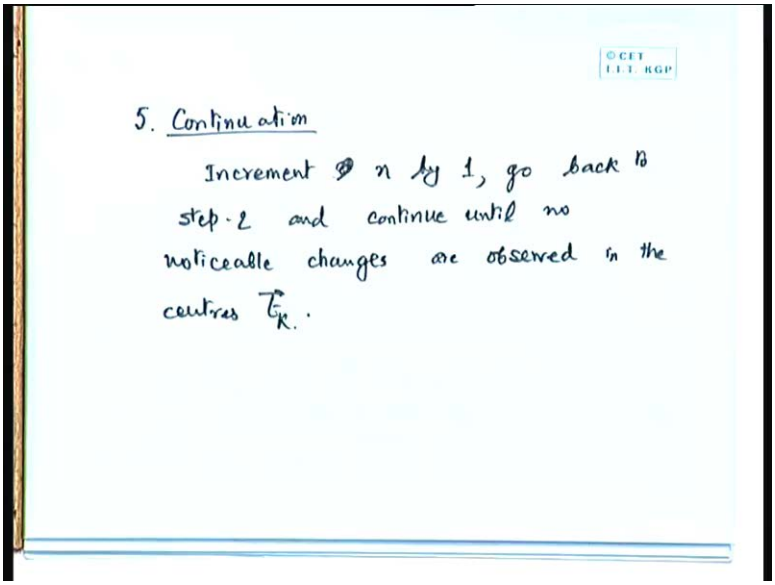
Student: Remain the same.

Remain the same, so; that means, to say in that case t k n plus 1 will be t k n, so this is the case otherwise. And obviously, as before what we have to do is to choose some learning rate eta which lies between 0 and 1. So, this is the fourth step, which is the

updating, but mind you updating we have done for only one this thing, for only one pattern.

Now, similarly we have to do it for all the patterns and then once that is done for the all the patterns we have to repeat it all over again. And so what we have to do is that, the next step will be the continuation.

(Refer Slide Time: 23:11)



So, as a continuation step what we have to do is to increment n by 1, so in fact, by implementing n by 1 what we are actually doing is that yes. We are going over to the next iteration and in next iteration we are drawing a different sample. So, this n is not an index by epoch, but rather n is an index, n we update and for every iteration we pick a new vector.

So, it is not like the way the learning mechanism is there for the multi-layer perceptron, where we said that, you have to finish of x 1 to x n and that constitutes an epoch. And then, again you have to present x 1 to x n., here it says that you pick up the iteration number n, pick any pattern from the input space. Update it make it n plus 1 and then again pickup something at random.

So, if you are doing it randomly several times, then there will be a stage when every pattern will be picked up sometime or the other. And then, the learning will be carried out, so what we do as a continuation step is to increment n by 1. And then, where we have to go, we have to go back to the step number 2 that is the sampling ((Refer Time: 24:41)) after updating it we have to draw a fresh sample from the input space.

So, increment n by 1, then go back to step 2 and we have to continue until no noticeable changes are observed in the centre's t k. So, this algorithm looks pretty impressive, so that must to say that, the first step of learning will be the adjustments of all these t i's. And once that is done, then only we will be in a position to going for the second step of adjusting the weights.

Now in fact, if you look at the k-means clustering algorithm, I think you have already started feeling. That it is a competitive learning process, we have already discussed something about the competitive learning mechanism, when we were talking as the generalized learning mechanisms towards the beginning of this course. So, this is the case of the competitive learning and in fact, in this case what is happening is that the centre's are getting organized.

And they are getting organized themselves without the involvement of any supervisor or a teacher, they are adjusting themselves just by the process of feeding the patterns. So, that is why since they are organizing, themselves on their own without a teacher, we are calling that this is a form of self organization. And this self organization is a mechanism, which is a form of the competitive learning network and those networks are called as the Self Organizing Maps or SOMs.

And we will be covering the self organizing maps, because it is very important and quite interesting self competitive learning mechanism. So, we will be covering the self organizing maps in our future lectures, but one thing that one has to say is that, let is not thing that this k-means clustering algorithm is free from it is draw backs. Because, one thing what we have said is that, the very first step is an initialization ((Refer Time: 27:45)).

And because, we do not know any priory idea about where the center should be located, we chose random values. Now, that is the most logical thing that we could do initially, but this has got one drawback, that if by chance our initial choice of centre's is poor. It could be that the patterns will get stuck, if the positions of the radial basis function centre's will get stuck to the local optimum solution.

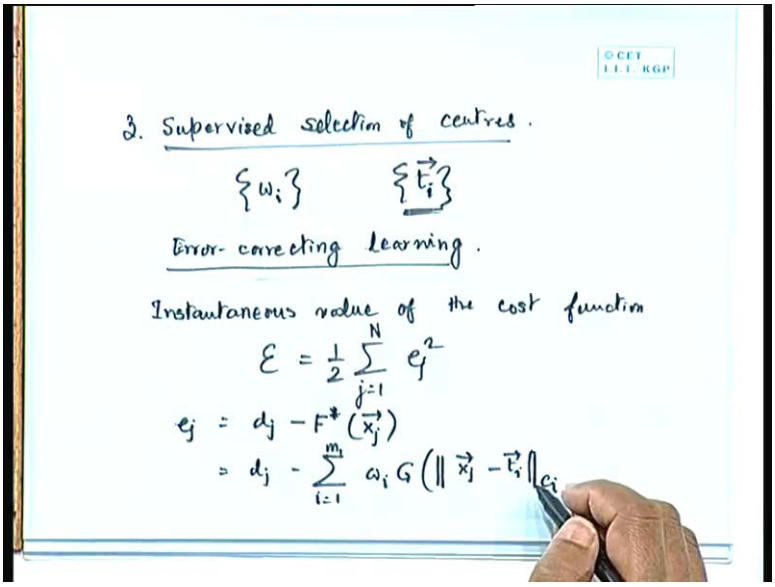
And if that is the case, then we will not be having the proper learning of the centre's, because instead of pointing to a globally optimum. Because, ideally what it should do is that for the group of patterns, which a radial basis function should present its position should be at the centroid of that. So, if instead it just locally adjusts to a few of the

patterns and it does not consider the others, which we should have then that is not a proper learn.

So, this; obviously, has got some draw back in the process of it is self organization. So, that is why, it was felt that may be that can we incorporate any supervised learning mechanism in this moment of the centre's. Because, here the only supervised learning process, that is involved in this sort of a self organization of the centre's of the RBF is, in the second step there is a supervision that can be incorporated.

Because, there we can compute the error between the desired and the actual response and then, accordingly we can adjust the w i's accordingly. But, not for the adjustment of the centre's, because they are organized in a self planning manner. So, we now come to a third methodology, where we will be talking about supervised selection of centre's.

(Refer Slide Time: 30:10)



So, this is the third mechanism third alternative process to learning, which is supervised selection of centres. So, in this case what is going to happen is that, it is not only the w i's, but also the t i's which will go through a supervised learning process. So, naturally for a supervised learning process, we know that the most obvious choice is the error correcting learning, what we have already studied.

So, if we can introduce a form of error correcting learning, which is most conveniently used implemented using gradient decent algorithm. And in that case, we can achieve this supervised way of learning for the centres t i's also. So, in order to have that we will be

talking in terms of the cost function, so we can define the instantaneous value of the cost function as.

So, in this case actually in the third alternative that is supervised selection of centre's, we choose that the learning mechanism will be now actually epoch wise. Epoch wise in the sense that we will be first feeding the set for n patterns and then we will be keeping the weights same for all the n patterns, like the batch learning mechanism, weights the centre's everything will be kept the same.

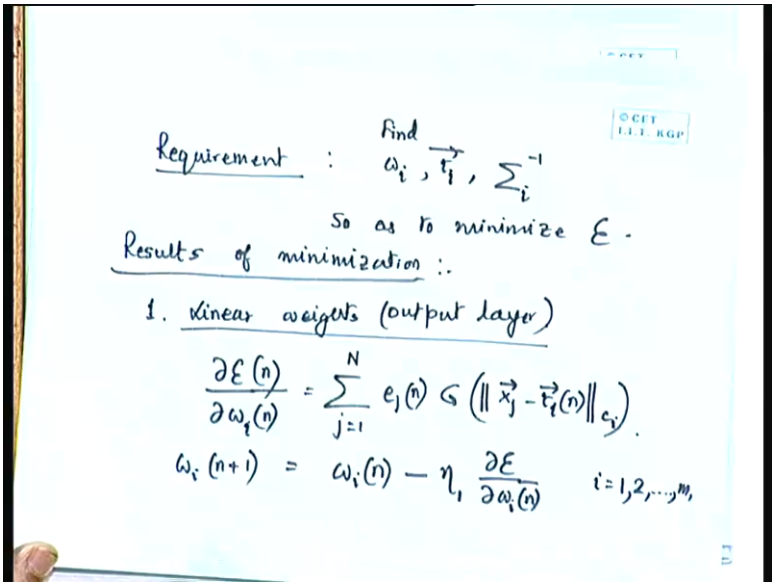
And then, depending upon the error that we obtain we will be making adjustments, so what we will be having is that. So, this will constitute an epoch and for an epoch the cost function that we are considering is, we are calling that as the instantaneous value of the cost function, which we are writing as half of e j square and this is j is equal to 1 to n. So, the sum of all the e j square from 1 to n we are calling as the instantaneous value of the error.

And in this case the error e j will be simply defined as the desired d j minus the actual response and how do we make the actual response. Now, in this case this is definitely an approximating network that we are taking, because we are feeding n number of patterns. But, we are having choice of m 1 number of centre's, where m 1 is less than that of n, so because this is an approximating function, we have used f star, so this is f star of x j.

So, this is the error and in fact, we can represent this f star x j as follows. So, we can rewrite this expression as d j minus summation over i is equal to 1 to m 1 of what this can be expressed as w i times G, the greens function of with the argument x of j minus t i and instead of choosing the Euclidean norm as an argument, we choose the weighted norm as an argument.

So, this is the i th, so we choose this as c i, so this is the i th greens function. So, we choose the weighted norm as the c i matrix, so we write it as c i, so this is the weighted norm of this function and weighted norm of this argument. So, this is our instantaneous error, so what we have to do in this process is, so this is e j and we will be accordingly finding the e j square and then the instantaneous value of the cost function. And what all our requirement will be to find all the free parameters.

(Refer Slide Time: 34:55)



So, our requirement is to find w i's definitely w i's are the free parameters t i's, they are also the free parameters, anything else that is the free parameters.

Student: ((Refer Time: 35:11))

No, yes there are you see that we are choosing the weighted norm, so weighted norm means there is a c i matrix that is involved weighting matrix. So, even the weighting matrix elements that is also a free parameter to it. So, weighting matrix is a free parameter, in other words what we have as a free parameter is the sigma, sigma is what, sigma we had chosen as the.

Student: ((Refer Time: 35:48))

Yes ((Refer Time: 35:50)) as the covariance matrix that, so as a free parameter we can also have sigma i or rather what we actually require in our case is the sigma i inverts. Because, actually sigma i inverts if you remember our last time's expression, that is essentially indicating the spread of the greens function, so even that is also a free parameter. So, it is not only the centre's, but also what size are you choosing for the function, that is also made a variable and as a free parameter.

So, everything can be controlled or everything can be updated, so our requirement is to find the best set of w i t i sigma i inverts. So, that the error is minimized, so as to minimize the error e, so the result of this minimization we can summarize as follows. In fact, that I am writing the results, but you can do it as an exercise yourselves, you see

that the first result that we present is that of the updating of the liner weights, linear weights of the ones, that we are having at the output layer.

And in order to find the linear weights, what we have to do is to differentiate the error with respect to come on do not keep quiet.

Student: With respect to w i's.

With respect to w i's, because when our objective is to update the linear weights, we should differentiate the e with respect to w i. And what are we going to obtain are there can anybody make a good guess ((Refer Time: 38:01)) you see look at the expression of e, e is summation of e j square j is equal to 1 to n. And individually this e j is like this, so if you look at the e j's expression here, there is a d j which is of course, independent of w j.

And only here we are having some terms that is involving w i. Now, out of all these things up to all these i is equal to 1 to m 1, if we try to differentiate it will be only the term w j that will have a nonzero component. And that we have to find out for all this j is equal to 1 to n. So, in effect if we differentiate this, we are going to obtain a solution of this form, we are going to obtain it as summation j is equal to 1 to n e j n.

And then, it will be multiplied by the greens function and the greens function will be G as an argument x j minus t i at iteration n having the weighted norm c i. And here, the w i n plus 1 that is to say the updated weight, the updated weight will be ((Refer Time: 39:39)) given by w i n that is a first weight minus eta 1 dau e dau w i n is a simplest gradient decent type of technique.

At where it will be learning rate times the derivative that we have obtained and that becomes our updated weight. And this mind you we have to carry out for i is equal to 1 to m 1 we have to do it for all the i's for 1 to m 1. So, this is the yes please any doubts.

Student: ((Refer Time: 40:23))

Derivative no.

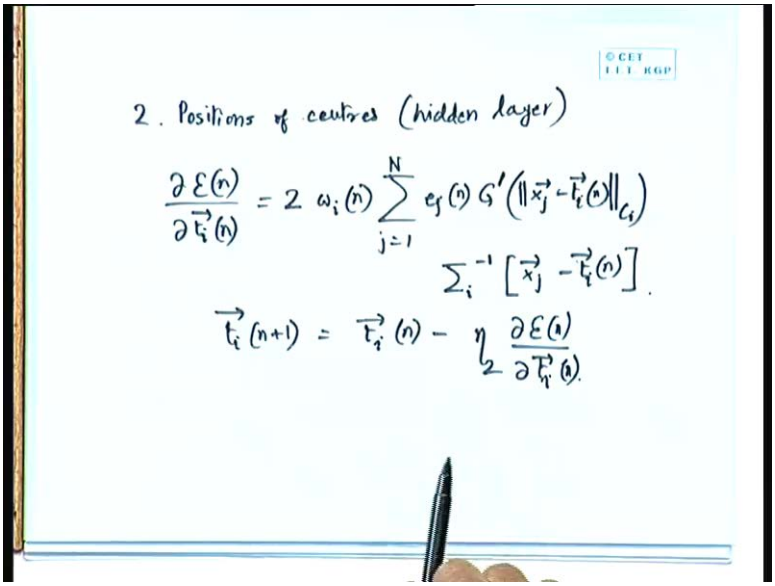
Student: ((Refer Time: 40:30))

But, actually speaking whatever dau e you get as the sign I you will have to reverse that sign over here. So, if it happens that this one ((Refer Time: 40:48)), now because of weights you do not know weights could be positive weights could be negative, but

whatever term we get we have to move in the opposite. Then, in a direction against the gradient, so naturally there is a minus that is involved in the learning process.

So, this will take care, so this is the adjustment of the linear weights which is pretty simple not any new knowledge that we had acquired. Because, this sort of a weight adjustment learning we earlier also did, only difference is the that in this case we are having greens function coming in the derivative term. But, once found out derivative it is easy to adjust the weights.

(Refer Slide Time: 41:45)



And next, what is involved is the positions of the centre's, because now that is also done in a supervised manner. So, let us see that how the positions of the centre's are organized, so this positions of the centre's will be learned for the hidden layer. So, in order to have that now what are we going to do for the how are we going to differentiate this function e with respect to.

Student: ((Refer Time: 42:19))

No.

Student: T i.

With respect to t i's yes that is right, so dau en in fact ((Refer Time: 42:28)), last time also we should have written here dau e n. Because, this is the instantaneous error at the nth iteration, so here we have to see dau e n dau t i n, we have to differentiate with respect to this. And if we do that in fact, you can just make a guess from here itself

((Refer Time: 42:51)) that e j square is and for also, e j square would necessarily mean that there will be a term that involves this w i square.

So, this in effect is two times in fact, that also tells us one thing, that somebody was pointing out to me that whether the derivative of e j with respect to w i becomes negative. In fact, here mind you that there is a square term that is involved, so this plus this square plus this square term. And then, if you would take a derivative of this, there is a half term that is involved over here.

So, what you get is e j w i that term, so the derivative was a positive term derivative was thinking was a negative. And here, the derivative that we will be obtaining with respect to t i is 2 times w i n summation over j is equal to 1 to n e j of n and this time what we are getting is the derivative of this G function. So, it is G prime that is what we are writing, so G prime means the derivative of G and as it is argument we will be having x j minus t i of n having weighted norm c i.

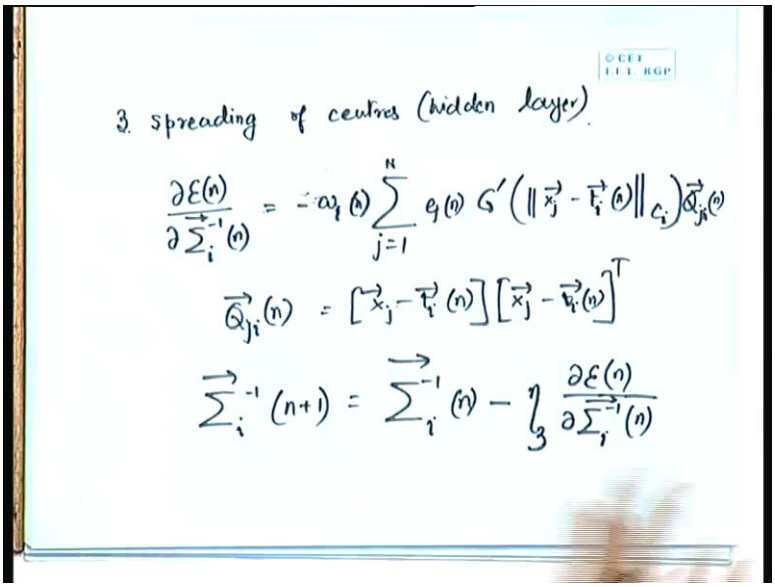
And then, this continues sigma i inverts and then we are having the term x j minus t i of n. And here, you say that our t i's t i n plus 1 becomes equal to t i n minus again some learning rate, but last time the last learning rate that is to say for the linear weights ((Refer Time: 45:07)) we were putting eta 1 as the learning rate. So, this time we have to put a different learning rate for the positions of the center.

So, we put it as eta 2 as the learning rate, so this is eta 2 minus dau e n dau t i n. Now, in this case you see that t i n plus 1 becomes the updated positions of the center. One point that you should note at this stages, that how are we obtaining t i n plus 1, we have to take the derivative with respect to t i of course, derivative of the error. And error we are getting only because we know that what our desired pattern is.

So, definitely the positions of the centre's they are also adjusted in accordance with the supervision. Because, we know that what d i is, we know what errors are and because we know what errors are, we can compute the derivative of that error function. And we can move the center positions accordingly, so it is a supervise adjustment of the positions of the center and as I was telling you, that in addition to the determination of the positions of the centre's.

Even, the spreading of the centre's that also can be determined in this case, in which case what we have do to is to differentiate this is with respect to sigma i's. So, if we do that then we will be getting results of this nature.

(Refer Slide Time: 46:54)



So, the third learning that is involved with the supervised selection clusters is the spreading of centre's, this also is to be done in the hidden layer. So, there what we have to do is dau e n dau sigma i inverts at n, which will be equal to minus w i n summation j is equal to 1 to capital N e j of n G prime argument is x j minus t i n weight norm c i times Q j i n. So, q j i n is a new matrix whose definition is like this Q j i n is actually the out of product matrix, which is obtained using x j minus t i n.

So, because it is an outer product what we have to do is x j minus t i n and then have x j minus t i n transpose, see if you multiply this with the transpose, then you will be getting the outer product of x j minus t i n with itself. So, that is the composition of this Q i j n matrix which we will be requiring in the computation of the derivative. And once this derivative is computed, then the updated sigma matrix could be obtained like this.

So, sigma inverts n plus 1 is equal to sigma inverts at iteration n, this is for the ith center mind you. So, this is done for the ith center and we have to do it for i is equal to 1 to up to m 1 for all the m 1 centre's we have to do this. So, this is eta i n minus this is for sigma i n minus again it should be proportional to the derivative, so dau e n dau sigma i inverts n and this we have to multiply by it is learning rate.

So, there is a third learning rate that is coming into picture and that is why we are putting it as eta 3. So, this way we will be able to adjust the spreading of the centre's also, so again you can see that three things are getting involved, one is the adjustment of the linear weights, the second is positions of the centre's and the third is the spreading of the centre's, this three things are involved.

So, now you can see that the first one, that is to say the linear weight adjustment if you now concentrate on. Then, it can be shown although we are not showing the detailed analysis of this, but it can be shown that this error surface happens to be convex with respect to the weights. So, as a result it is possible for us to obtain a globally minimum solution as far as these linear weights are concerned.

Whereas, this surface is not exactly the cost function is not convex or rather it is non convex with respect to t i's and the sigma i inverts. So, since the cost function is non convex with respect to t i and sigma i inverts, the optimum values that we get for t i and sigma i inverts can get start in the local minimum. And one thing that you see that here also a supervised learning is in fault, in the case of this supervised selection of centres.

We have a supervised learning, but the supervised learning in this case is much different from that of the back propagation algorithm. Because, back propagation you see what happens is we start with the outer layer and then we adjust to the inner ones. In this case there is no such requirement that we have to first to the output. And then, we have to come to the center, then we have to do the spreading of the centre's, this three things are quite independent of each of other.

So, the parallelization of learning mechanism is also possible in this case. So, this is highly interesting and in fact, lot of good researches are now going on in the field of the radial basis functions. Because, they have really emerged as good alternative to the back propagation learning process. In fact, more of such research I think we will continue in the future and there are already several applications of the radial basis function, that one finds especially in the domain of speech processing, there is good amount of research that is going on.

And in the image processing we have not found too many applications of RBF ((Refer Time: 53:14)), but in the coming years I think we are going to use lot of such applications involving the radial basis functions. And the next topic, which we will be

covering from the next class is the principle component analysis, which is in fact, a very important, very useful and very efficient tool for the dimensionality reduction of the data.

In fact, you will be surprised that whereas, in the case of the RBFs, we were talking about some dimensionality increase. Because, what we had talked of is that we had to map it into higher dimensional space. So, that the separability of the function improves, whereas in the case of the principle component another system, whole motivation will be to obtain a data reduction, which we will be doing by simply doing a dimensionality reduction technique.

And it is done in a very optimal way finds lot of applications in the data compression be it speech or be it image and video compression applications. So, we will be talking about that from the next class, so till then good bye.