Course Name: **Machine Learning and Deep learning - Fundamentals and Applications**

Professor Name: Prof. M. K. Bhuyan

Department Name: Electronics and Electrical Engineering

Institute Name: Indian Institute of Technology, Guwahati

Week-4

Lecture-18

Welcome to NPTEL MOOCs course on machine learning and deep learning fundamentals and applications. In my last class, I discussed the concept of perceptron criterion. And I have explained how to determine the best weight vector based on the information of the samples. So I have the samples for all the classes. And based on these training samples, I have to determine the weight vector. If I can determine the weight vector, I can determine the decision boundary between the classes

because the weight vector is orthogonal to the decision boundary.

After this, I discuss the concept of gradient descent algorithm. Today I am going to continue the same discussion that is the perceptron criterion and how to find the best weight vector from the training samples.

That concept I am going to explain again.So let us start this class.

So in my last class, I discussed the concept of designing of the weight vector W. So design of weight vector W. So what is the discriminant function?

$$g(\underline{X}) = \underline{w}^T \underline{x} + w_0$$

W is the weight vector, x is the input feature vector and W naught is the bias.

And this can be converted into the homogeneous form.

The homogeneous form is A transposed y. So this is the homogeneous form. So y is the augmented feature vector. So what is the y? y is the augmented feature vector.

So it is x1, x2 up to xd.

So d dimensional pressure vector and I am appending 1. So it is d plus 1 dimensional. So this is the augmented feature vector. And also we are considering the modified weight vector A. So we are considering all the components of the weight vector W.

That means it is W1, W2 up to Wd and also we are considering the bias W naught. So this is the modified weight vector. After this I discussed the concept of Perceptron criterion. So what is the Perceptron criterion?

So in the Perceptron criterion, we considered y. y is the augmented feature vector.

So that is already I have shown that is nothing but x1, x2, xd, 1. This is the augmented feature vector. So this actually what we are doing. This y is obtained from x. x is d dimensional.

We are getting y by appending 1 because I have to determine the homogeneous form. The homogeneous form is gx is equal to A transpose y. So how to take a classification decision? Suppose if A transpose y is greater than 0, the classification decision is y should be assigned to the class omega 1. And if A transpose y less than 0, so what is my decision rule? The y should be assigned to the class omega 2. This is my decision rule.

So in this case, these two conditions we are considering. So that means it is a two criteria we are considering for decision making, but how to make it one criterion. So for making one criterion, what we are considering all the training samples of omega 2, because we are considering two classes omega 1 and omega 2. So all the training samples of omega 2 are negated. So I can write all the training samples of omega 2 are negated.

So that A transpose y is greater than 0. So one condition I am getting is only one condition or one criterion and based on this condition, I can do the classification. And after this, I discussed how to determine the best weight vector. So what you have to consider? You have to consider A0 that is the initial weight vector. So arbitrarily we can select this one.

$$\underline{a}(K+1) = \underline{a}(K) - \eta \nabla J(\underline{a}(K))$$

In the k plus 1 iteration, Ak plus 1, my updation rule is like this Ak and eta is the learning rate and we are considering the gradient of the criterion function. That is the perceptron criterion function Ja. So what is the perceptron criterion? What we have considered? So move to the next slide. So I am repeating it again. The initial weight vector I have to select arbitrarily A0 and after this at the k plus 1 iteration, I can determine like this, the weight vector at the kth iteration Ak minus eta that is the learning rate and we are taking the gradient with respect to A and we are considering the criterion function Ja.

So what is the perceptron criterion? The criterion function is JpA and we have considered like this in the last class it is minus A transpose y and we are only considering the misclassified samples because the problem is because of the misclassified samples. So we are considering only the misclassified samples for all y which are misclassified. Based on this perceptron criterion how to do the iteration? So A0 we have to select arbitrarily weight vector and at k plus 1 iteration, so I can write like this Ak that is the weight vector at kth iteration plus eta summation y and for all the y's which are misclassified for all y's which are misclassified. So we discussed in this technique there is a problem. If you see here we are considering the summation of all the misclassified samples.

So that means all the samples are considered together all the samples we are considering. The problem is the memory requirement we have 1000 samples and we have 1000s of misclassified samples. So that is why we need large amount of memory. So that is the problem the problem is due to the memory requirement. So how to minimize the memory requirement? So for this we are considering the sequential version of this algorithm.

So the sequential version of the algorithm because we have to minimize the memory requirement. So what is this sequential version of the algorithm? Suppose we are considering the samples y1 like this suppose y2, y3, yk minus 1, yk and yn. So we are considering these samples. So we have to see my condition is satisfied or not corresponding to the samples. So what is the condition?

The condition is A transpose y should be greater than 0.

Suppose this condition is satisfied for the first sample y1 then we need not consider y1 and we have to go to the next sample and we have to see corresponding to y2 whether this particular condition is satisfied or not. And in this case what we are considering all the samples belonging to the class omega 2 are negated already in the sequential version of this algorithm. First I have to consider y1 and we have to see whether this condition the condition is A transpose y greater than 0 that condition is satisfied or not that we have to see. If it is satisfied then we need not consider y1 then we have to go to the next sample like this we have to go suppose up to yk minus 1 and this condition is satisfied what is the condition A transpose y greater than 0. So this condition is satisfied.

So that means no need to consider all the samples and no need to modify the weight vector because all the samples are correctly classified that means no need to modify the weight vector. So this is up to the sample yk minus 1. So move to the next slide suppose the next sample is yk that is after yk minus 1 and corresponding to this the condition is not satisfied the condition is A transpose yk that is not satisfied. So not satisfied means the misclassification then we have to consider only yk and we have to modify the weight vector. So how to modify the weight vector the formula already you know A0 we have to select arbitrarily and Ak plus 1 we can determine that is Ak plus eta yk we are updating the width now because corresponding to the sample yk and

the condition is not satisfied the misclassification is taking place.

So I have to get the new weight vector. So now by considering this I am getting the modified weight vector and it should classify all the samples correctly the new weight vector should classify all the samples correctly. So that means I have to consider all the previous samples again because I am modifying the weight vector then I am getting the new weight vector is Ak plus 1 suppose because I have the misclassification corresponding to the sample yk. So that means I have to modify the weight vector as per this formula. So Ak plus 1 is equal to Ak plus eta yk and after modifying the weight vector I am getting a new weight vector and it should classify all the samples correctly.

That means again I have to check all the previous samples and also we need to shake whether it is converging or not converging because the weight vector should move to the solution region. So in my last class I have explained what is the solution region. So I should get the best weight vector and the weight vector should lie in the solution region. So that we have to see whether the convergence is taking place or not.

So in my last class I have shown what is the solution region.

So I am showing it again suppose if I consider a two dimensional feature space. So it is suppose x1 and this is x2 suppose I have the samples. So these are the samples belonging to the class Omega 1 and I have also samples belonging to the class Omega 2. So suppose these are the samples belonging to the class Omega 2. So this is Omega 2 after is what we have to consider the samples of class Omega 2 should be negated.

So if I do the negation of these samples I will be getting the negated samples like this. Now I have to draw the decision boundary. So which one is the best decision boundary you can see. So corresponding to the class Omega 1 this is my limiting case.

So if I move the decision boundary suppose in the anti-clockwise direction

beyond this point the misclassification will take place.

And similarly if I rotate this decision boundary in the clockwise direction this is the limiting case. So beyond this I cannot move the decision boundary because the misclassification will take place. So corresponding to these two limiting cases what will be my weight vector. The weight vector is perpendicular or weight vector is orthogonal to the decision boundary. So that means corresponding to the number one decision boundary my weight vector is orthogonal weight vector and corresponding to the second decision boundary that is number two my weight vector will be like this maybe something like this.

So I have the conical region this is the conical region or I can say it is the solution region so it is a conical region solution region and within this region my weight vector should lie. So this is the concept that I have explained. So my weight vector should lie within this solution region. So this concept already I have explained in my last class. Now corresponding to this sequential algorithm let us see how we can determine the best weight vector.

So move to the next slide. So corresponding to this sequential algorithm let us again consider a two dimensional feature space and we are considering some samples belonging to the class omega 1 and omega 2 two classes we are considering. So suppose the samples are like this. So these are the samples corresponding to the class omega 1 and I have some samples corresponding to the class omega 2. So these are the samples corresponding to the class omega 2.

So first in the sequential algorithm we have to select the initial weight vector arbitrarily.

So suppose I am selecting my weight vector suppose here this is the initial weight vector and suppose it is A0. So I have the samples suppose I am just labeling the samples the sample number is suppose 1 2 3 4 5 these are the sample number I am putting 1 2 3 4 5 corresponding to this initial weight vector. So this is nothing but the initial weight vector this is the initial weight vector corresponding to this initial weight vector my decision boundary is something like this suppose it is orthogonal to the weight vector. So you can see if I consider the weight vector A0 so you can see the misclassification the three samples are misclassified number three it is misclassified four is misclassified and five is also misclassified. So that means the three samples are misclassified and already I told you the weight vector is orthogonal to the hyper plane and

in this case we are considering only the two dimensional Feature vector.

So that means we have the 2D Feature space that means the decision boundary is a straight line. So after this we have to apply this updation rule updation rule is Ak is equal to Ak minus 1 plus eta y so that already I have explained so that means the y is scale and it is added with the previous value of the weight vector. So you can see the vector y is scale by eta and it is added with the previous value of Ak that is the weight vector. In the next iteration what will be the updated weight vector because it is the addition of the previous weight vector with the eta y that means I will be getting my next weight vector maybe something like this that is A1 this weight vector I am getting corresponding to the second iteration in this case also if I draw the decision boundary it will be something like this. So in this case also the two samples are misclassified so that means again I have to go for this iteration Ak is equal to Ak minus 1 plus eta y so that means with each and every iteration pushing the weight vector towards the misclassified samples.

So I am pushing the weight vector like this the misclassified samples. So the sample initially the 3, 4, 5 they were misclassified and after the updation of the weight the sample 4 and 5 they are misclassified and what actually we are doing with each and every iteration pushing the weight vector towards the misclassified sample. And suppose in the next iteration my weight vector is A2 so corresponding to this A2 my decision boundary will be something like this. So you can see all the samples are classified correctly the 1, 2, 3, 4, 5 and obviously the samples belonging to the second class they are also correctly classified corresponding to A naught you can see the decision boundary corresponding to A1 you have seen the decision boundary and corresponding to A2 you can see the decision boundary. So that means what we are doing with each and every iteration pushing the weight vector towards the misclassified sample and we have a solution region.

So already I have explained what is the solution region. So we have a solution region suppose this is my solution region this is the solution region and you can see now A2 is within the solution region this is my solution region. So I am pushing the weight vector into the solution region because of this iteration. In the feature space actually I should write here this X1 and X2 the two dimensional feature space suppose the learning rate is very high after the first iteration this A1 may be here A1 may be here if the eta is very high if the learning rate is very high the A1 may not be in the initial position. So it may jump to or it may lean to that position the green colored one so A1.

So that means all the samples will be misclassified that is why the learning rate is important. So it should not be very very high and if I consider the small value of the learning rate the convergence will take time. So this is the concept of getting the best weight vector by considering this iterative equation. So in this case there is a problem the problem I will mention here so better to draw one new figure. So suppose showing the same thing here what is the problem in this case.

So we are again considering the two dimensional feature space X1 and X2 and we have the samples already I have shown the samples for two classes. So the samples are so these are the samples for one class and similarly I have the samples for the second class. So for two classes these are the samples suppose I am getting the decision boundary the decision boundary is suppose somewhere like this or maybe the decision boundary maybe somewhere like this and corresponding to a decision boundary 1 corresponding to the decision boundary 2. So I think better to so the second decision boundary by different color.

So the second decision boundary suppose the second decision boundary we are considering that is number 2.

So corresponding to the number 1 and decision boundary I can draw the weight vector the weight vector is orthogonal to the decision boundary. So this is the weight vector and corresponding to the number 2 and decision boundary I have the weight vector like this. This is my conical region that is the solution region. So what is the problem in this case suppose in this case only we are showing 5 samples for class omega 1 and 5 samples or samples for the class omega 2. So there may be many samples and if it is the position of the decision boundary what will happen there may be some misclassification suppose some of the samples may be here some of the samples may be here.

So these samples will be misclassified there are thousands of thousand samples and similarly if I consider some of the samples here these samples will be misclassified. So that is why better to consider one offset some tolerance we can consider maybe this decision boundary we can consider here by considering the offset and this decision boundary the second decision boundary maybe we can consider here by considering the offset some tolerance then the misclassification will be less. Otherwise if I consider this limiting cases there will be misclassification. So that is why we are considering some margin and based on this I am modifying the equation for the decision making.

So what is my modification A transpose Y greater than B. So earlier it was 0 now I am considering the margin and that is the positive constant. So this B is the margin and that is the positive constant we are considering this margin so that all the samples will be correctly classified that is actually the safety margin. So that means I am restricting the solution region within the main sub regions I can show the solution region now considering this my solution region maybe here now. So this is my new solution region that is within the main solution region. So that means I am restricting the solution region within the main solution region that means I am getting a new sub region.

So this is actually new solution region solution region I am getting. So this margin we are considering so my criteria is now A transpose Y greater than B. So we are considering the margin B and B is the positive constant. So up till now I discussed the concept of Perceptron criterion and this Perceptron criterion function is not only the single criteria for designing of a linear classifier.

So there are different algorithms

so I will be discussing some of these algorithms how to design the weight vector.

So move to the next slide so another criterion function is relaxation criterion. So this criterion function is defined like this J r that is the relaxation criterion for the weight vector

$$J_r(\underline{a}) = \frac{1}{2}\sum_{\forall \underline{y}} \frac{\underline{a^T y} - \underline{b}^2}{||y||^2}$$

So this is whole square and this summation that is for all Y which are misclassified. So this is the criterion function and that is for the relaxation criterion.

So I have to do the minimization of this function. So maybe we can apply the gradient descent algorithm. So what is the gradient of this so gradient with respect to the vector A. So if I take the gradient of this the gradient with respect to the vector A the summation that is for all Y which are misclassified. This is A transpose Y minus B mod of Y square and Y.

So we are taking the gradient with respect to the vector A. So this is the gradient of the relaxation criterion function. Now how to implement this so I can consider the iterative algorithm like we considered in case of the perceptron criterion. So what is this algorithm we can consider A naught that is arbitrarily we can select. So in the k plus 1 iteration Ak plus 1 it can be represented like this Ak plus eta eta is the learning rate and summation and again we are considering for all Y which are misclassified.

So this is the B minus A transpose Y Y square Y. So at k plus 1 iteration this is the expression Ak plus 1 is equal to Ak eta and summation for all Y which are misclassified and B minus A transpose Y divided by mod Y square into Y. So with the help of this iterative algorithm I can determine the weight vector Ak plus 1. So what is this misclassified samples the samples which are misclassified by the weight vector Ak. So what is the Y actually the Y is the misclassified sample and these are the misclassified sample by the weight vector Ak. So based on this I am getting updated with vector the updated with vector is Ak plus 1.

So now we are considering the sequential version of this algorithm. So move to the next slide because the problem is the memory requirement. So as we did in case of the perceptron criterion in this case also we can consider the sequential version of this algorithm. So what is the sequential version of this algorithm? So again we have to select A0 arbitrarily and what is the value of A in the k plus 1 iteration that is Ak that is the weight vector at the kth iteration. This is the learning rate eta is the learning rate B minus A transpose Yk. So this Yk is nothing but these are the samples which are misclassified by the weight vector Ak.

So this is the sequential version of the previous algorithm and this is mainly done to reduce the memory requirement. So this is called the sequential version of the relaxation algorithm. So I will be discussing another algorithm for determining the weight vector that is the minimum squared error criterion. So what is the minimum squared error criterion? So let us discuss about this minimum squared error that is called MSE a minimum squared error criterion. So in case of the perceptron criterion and also if I consider the relaxation criterion they can perfectly classify when the classes are linearly separable.

So in case of the perceptron criterion what we considered that we considered that the classes are linearly separable and similarly in case of the relaxation criterion also we considered that the classes are linearly separable. But suppose if we do not know this classes are linearly separable or not then the best designing technique is the mean squared error criterion. So this minimum squared error criterion we have to consider if I do not have the knowledge whether the classes are linearly separable or not and based on this MSE criterion I can design a linear classifier. So this is the fundamental concept of the MSE criterion. So I have to reduce the error I am repeating this so earlier we considered that the classes are linearly separable.

But if the classes are not linearly separable or if I not sure if the classes are linearly separable then I have to design a linear classifier based on the minimum squared error criterion. That means I have to reduce the error and based on this I can design a linear classifier. So now the decision rule that we obtained previously so what is the decision rule? So decision rule already I have defined so A transpose y should be greater than B so B is the margin. So if this condition is satisfied then y will be properly classified. So this condition I have to consider for the classification if this condition is satisfied then y will be properly classified.

And what is the decision boundary or decision surface I can represent like this A transpose y is equal to B. So this is the equation of the decision boundary. So B is the margin vector the B is the margin we are considering the margin vector. So the equation of the decision surface is A transpose y is equal to B.

So we are considering the margin the margin is nothing but the B. So solution of this equation which is obtained by mean squared error criterion. So I have to solve this equation and this solution can be obtained by minimum squared error criterion. So suppose if I consider this expression A transpose y i what is y i that is the i th sample we are considering and suppose B i. So different margins for different y i we are considering.

So that means what is B i actually different margins for different y i. So what is y i that is the i th sample. So we are considering different margins for different y i. So in this case I will be getting n number of equations. So if you see here I will be getting n number of equations.

So in the matrix form I can write this equation also. So move to the next slide. So from the previous slide what we have shown A transpose y i is equal to B i. So we have n number of equations. So in the matrix equation I can write like this

$$[y_{10}\ y_{11} \dots y_{1d}\ y_{20}\ y_{21} \dots y_{2d}\ \vdots\ y_{n0}\ y_{n1} \dots y_{nd}\ ][a_0\ a_1\ \vdots\ a_d\ ] = [b_1\ b_2\ \vdots\ b_n\ ]$$

because we are considering different margins for different y's. So the margins are like this B 1 B 2. So that means the B is a vector and different margins we are considering B 1 B 2 for different y i's.

So in a matrix form I can write like this. So in this case n number of simultaneous equation here which can be written like this y A is equal to B. So how to determine A because we have to determine the weight vector A. So that is nothing but y inverse B this y is a rectangular matrix if you see number of rows is less than the number of columns. So inverse we cannot determine. So it is a rectangular matrix that means the number of rows is less than the number of columns.

So I cannot get the exact solution of this. So that is why to solve this equation we have to define one error term. So how to define the error. So move to the next slide. So to solve this equation so we define the error y A minus B and based on this error we are considering the sum squared error criterion. So this criterion function I can write like this J S A is equal to mod y A minus B squared.

So which can be written like this summation from I is equal to 1 to n A transpose y I minus B I. So we can write in this form. So from I is equal to 1 to n. So if I solve this equation then I will be getting the weight vector. So that means I have to minimize the error and for this I can apply the gradient descent algorithm.

So we have defined the sum squared error criterion and if I solve this one I can get the weight vector. That means I have to minimize the error and we can apply the gradient descent algorithm. So now I have to determine the gradient. So gradient of this error function. So if I take the gradient of J S A the gradient will be summation from I is equal to 1 to n to A transpose y I minus B I into y I.

So in the matrix from I can write like this twice the y transpose y A minus B in the matrix form I can write like this. So with this gradient I can apply the gradient descent algorithm.

So we have determined the gradient and now we can get the close from of the solution the close from of the solution I can write like this. The close from solution that is nothing but this gradient of J S A I am equating it to 0.

So that is the close from of the solution. So we can move to the next slide. So what is the close from of the solution from the previous slide close from solution. So this gradient of J S A equating it to 0 and if I equate to 0 so it is nothing but 2 y transpose y A minus B B is the margin vector is equal to 0. So that is equal to y transpose y A is equal to y transpose B. So now how to determine this weight vector A the weight vector A is nothing but y transpose y inverse y transpose B.

So if you see the y is a rectangular matrix of dimension n cross D. So I can say y is a rectangular matrix. The dimension is n cross D. So what is the dimension of y transpose the dimension of y transpose is D cross n. So that means the dimension of y transpose y that will be D cross D and it is a non-singular matrix.

So if it is a non-singular matrix then y transpose y inverse I can determine. Because you can see y is originally the rectangular matrix of dimension n cross D and y transpose the dimension is D cross n and what is the dimension of y transpose y dimension is D cross D. So that means I am getting the squared matrix if I consider y transpose y I am getting the square matrix and if I consider this one y transpose y inverse into y transpose. So if I see this term from this to this and this is nothing but y plus that is actually the pseudo inverse it is called a pseudo inverse. This y transpose y is a square matrix of dimension D cross D and you can see this term y transpose y inverse into y transpose that is nothing but a pseudo inverse. So that means what I am getting the final expression the expression is a is equal to this y plus that is the pseudo inverse b.

So y plus is the pseudo inverse of y. So this pseudo inverse is same as that of the regular inverse for square and non-singular matrix. So that concept already you know that means I am repeating this a pseudo inverse is same as that of the regular inverse for a square and non-singular matrix. So in this case you can see y transpose y is a square matrix and it is a non-singular matrix. So that means we can determine the inverse and we can see the final expression that is a weight vector a I can determine like this a is nothing but y pseudo inverse b that is the expression for the a that is the weight vector.

So this also we can apply the iterative algorithm. So like in the previous cases we applied the iterative algorithms in case of the perceptron criterion also in case of the relaxation criterion also we applied the iterative algorithms in this case also we can apply the iterative algorithm. So let us move to the next slide. So this iterative algorithm is called the withdraw hop or the LMS learning rule. So what is this iterative algorithm withdraw hop or I can say LMS learning rule. So like in the previous case is a naught we have to select arbitrarily after this a k plus 1 we can consider like this a k plus 1 is equal to a k plus eta eta is the learning rate y transpose b minus y a k.

So in this case y is actually we are considering all the samples together that means we need the large amount of memory. So why we are considering all the samples together. So that is why the sequential version we can consider to reduce the amount of memory. So what is the sequential version sequential version of this algorithm is again we have to select a naught arbitrarily and we can determine a k plus 1 that is equal to a k plus eta b k minus a transpose k y k into y k.

So y k means the samples which are misclassified by the weight vector a k. So that samples we are considering. So this is the sequential version of the above algorithm because we have to reduce the memory requirement and you can see this is the LMS learning rule or it is called a withdraw hop learning rule LMS means the least mean squared learning rule. So this is about the minimum squared error criterion. So you can see if the classes are not linearly separable then we can go for mean squared error criterion. In this case we are not considering whether the classes are linearly separable or not but we are trying to reduce the error and based on this we can design the weight vector.

In this class I have discussed the concept of designing the best weight vector and for this first I discussed the concept of the perceptron criterion. After this I discussed the concept of the relaxation criterion and also I discussed the sequential version of these algorithms because the sequential version of this algorithm is quite important to reduce the memory requirement. After this I discussed the concept of MSC criterion if I do not have the information whether the classes are linearly separable or not then better to apply the MSC criterion. So after applying the MSC criterion I will be getting a linear classifier. So all these criterion functions are quite important to design the discriminant function because for the discriminant function I need the information of the weight vector. So let me stop here today. Thank you.