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-10

Lecture-35

Welcome to NPTEL online course on machine learning and deep learning fundamentals and applications. In my last class, I briefly introduced the principles of artificial neural networks. And also, I highlighted the importance of the bias input and the activation function. In that lecture, I explained the concept of XOR logic, which is a non-linearly separable pattern classification problem. And for this, I design one artificial neural network. Today, I am going to discuss about another popular artificial neural network that is the radial basis function, artificial neural network.

In case of the RBF, the output depends on the distance between the input vector and the stored vector. And today's class, I will be again discussing the XOR problem that is the XOR logic, which is a non-linearly separable pattern classification problem and how to handle that problem by considering the radial basis function. And in the first class, I explain one important concept that is the Cover's theorem.

If the training samples are not linearly separable in a low dimensional space, then what I can do, I can apply some non-linear transformation techniques to project the low dimensional data into a high dimensional feature space.

If I do these projections into a high dimensional feature space, then the training samples will be linearly separable. That is the concept of the Cover's theorem. So now let us discuss about the radial basis function, artificial neural network.

So today I am going to discuss about the radial basis functions networks. So already I told you in case of the RBF, the radial basis function neural network, the output depends on the distance of the input vector from a given stored vector.

In the figure also I am showing that principle. So suppose if I consider the vector, this vector is 1. This vector is approximated by 3 vectors. So I am showing 3 green colored vectors. The vector 1 is represented by considering these 3 green vectors.

So also I can write in a RBF neural network, we have the hidden layers and the hidden layer uses neurons with RBF activation functions. So I can write the hidden layers uses neurons with RBF. So, that means the radial basis function neural networks has hidden layers, hidden layer uses neurons with RBF activation function and one output node which is used to combine one output node is used to combine linearly the outputs of the hidden neurons. So you can see in the RBF networks, we have a hidden layer which uses neurons with RBF activation function and we have one output node which is used to combine linearly the output of the hidden neurons.

In this figure what already I have explained, I have a vector, vector is 1.

The output of the vector 1 is interpolated using the 3 green vectors. So in the figure you can see I have 3 green vectors. The vector 1 is interpolated using the 3 green vectors and each vector gives a contribution that depends on its weight and on its distance from the point 1. So that means again I am repeating each vector gives a contribution that depends on its weight and on its distance from the point 1. In this case the weights are like this $w_1 < w_3 < w_2$.

So the weights are like this in this figure. So this is the fundamental concept of the radial basis function. So the output depends on the distance of the input from a given stored vector. So let us now discuss about RBF architecture, the radial basis function artificial neural network architecture. So move to the next slide the RBF architecture.

So suppose these are the input nodes $x_1 x_2 ... x_d$. So this is for actually the vector X-. X- is a d dimensional vector. So I have d number of input nodes. After this I am considering the activation functions ϕ_1 ϕ_2 like this.

These are the activation functions corresponding to the hidden layer. I have a hidden layer and I have also one output layer. So this is the output node and from the output node I am getting the output Y. So now I have to show the interconnections between these neurons. So these interconnections I have to show.

So, all the interconnections I am showing and I have to show the interconnections between the hidden layer and the output node. So I am combining the outputs of the hidden layer in the output node and the weights are already I have explained the weights are $w_1 w_2 ... w_M$. These are the weights connecting weights. So this is one structure of the RBF artificial neural network. So in the RBF activation function you can see we are considering this is a hidden layer so this is the hidden layer and we are considering the activation functions ϕ_1 ϕ_2.

So these are the RBF activation functions and what will be the output layer we are considering this is the output layer. In this case we are considering the linear activation function. So corresponding to this network I can get the output Y that is nothing but the linear combinations I have to consider.

So the distance between X and t_1 that is the input vector is X and t_1 is the target vector that is the stored vector $w_M \phi_M(||X - t_M||)$ X I am not giving the vector sign here X is a vector t is also a vector. So I am not giving the vector sign here.

So in this expression this is $||X - t||$ that is a distance so we are considering the Euclidean distance the distance of X, X is a vector the d dimensional vector from the target vector the target vector is t the target vector is t. So you can see how we can determine the output corresponding to this RBF architecture and we are considering the distance between the input vector and the target vector. So what type of ϕ we can consider so let us move to the next slide the types of ϕ so what types of ϕ we can consider one popular is multi quadrics multi quadrics activation so that is $\phi(r) = (r^2 + c^2)^{\frac{1}{2}}$ so c > 0 and the $r = ||x - t||$ that is the r. So this activation function we can use that is the multi quadrics another one is inverse multiquadrics inverse multiquadrics so activation function will be $\phi(r) = \frac{1}{(r^2+c^2)^{\frac{1}{2}}}$ for c > 0 and number 3 that is a Gaussian function. So in most of the cases it is used it is a very popular activation function so $\phi(r) = exp(-\frac{r^2}{2\sigma^2})$ and $\sigma > 0$.

So these are some popular activation function used in the RBF neural network. Now let us discussed about the example that already I have discussed in my last class that is the XOR classification problem that is a non-linearly separable pattern classification problem. So how to consider that problem by considering this radial basis function artificial neural network. So move to the next slide so one example we are considering now and that is nothing but the XOR problem. So you know in case of the XOR input suppose is X and output is suppose Y so corresponding to this combination 0 0, 0 1, 1 0 and 1 1.

So you can see the output is 0 here 0 1 is 1 1 0 is 1 and 1 1 is 0. So these are the outputs corresponding to the combinations the input combinations 0 0 0 1 1 0 and 1 1. So this is the XOR logic so that can be shown in the figure also. So input space so how to show the input space so we are considering these two variables corresponding to X one is x_1 another one is x_2 and I have to consider the points so this is one point this is another point this is another point and this is another point. So that is actually the 0 0 this is 1 0 this is 1 1 and it is 0 1 this is the input space.

So what about the output space the output space so I can show the output Y you have seen that output we have two outputs only one is 0 another one is 1. So now our problem is what is the problem now construct or the design an RBF pattern classifier such that so the conditions are like this corresponding to 0 0 and 1 1 these input combinations that should be mapped to 0 and that is corresponding to the class C_1 and corresponding to this input combination that is 1 0 and 0 1 it should be mapped to 1 and that is corresponding to the another class that class is C_2. So that means it is a two class problem it is a two class pattern classification problem. So how this problem I can consider by considering the RBF network let us explain in the next slide. So in the feature space that is actually the in the hidden layer so that is in the hidden layer so $\phi_1(||X - t_1||) = e^{-||X-t_1||^2}$.

So in all these cases I am not giving the vector sign so I am not showing the vector sign and similarly the $\phi_2(||X - t_2||) = e^{-||X-t_2||^2}$. So this we are considering and what is t_1? t_1 is nothing but (1, 1) that is the target vector and the t_2 is (0, 0) so these are two target vectors. So corresponding to this I can show the decision boundary and also the outputs I can show. So it is in the feature space ϕ_1 and ϕ_2 so suppose this value is suppose 1.0 and this value is suppose 1.0 this is 0.5 this is also 0.5 and I can show the decision boundary the decision boundary will be so this is the decision boundary.

So I have to do the mapping so after the mapping what I will be getting the points you can see here I will be getting one point here another point I will be getting here maybe here and another point I may get here these three points I am getting after mapping. So this point corresponds to the input inputs are 0 1 and 1 0 so for these two combinations I am getting this output in the feature space this output I am getting corresponding to the input 1 1 and this output I am getting for the input 0 0. So all these outputs I am getting in the feature space and this is what this is the decision boundary. So you can see when we map into the feature space that is in the hidden layer these two classes C_1 and C_2 they will be linearly separable.

So a linear classifier with $\phi_1(x)$ and $\phi_2(x)$ as inputs can be used to solve the XOR problem so I am repeating this so when I map into the feature space these two classes C_1 and C_2 will be linearly separable. So what I can consider now so I can consider a linear classifier with $\phi_1(x)$ and $\phi_2(x)$ as inputs these inputs can be used to solve the feature problem so how to get these green colored points that is the output points I can show these computations in the next slide. So let us move to the next slide so we consider the activation function $\phi_1(x) = e^{-||x-t_1||^2}$ and $\phi_2(x) = e^{-||x-t_2||^2}$ so we considered these two activation functions and what are my target vectors $t_1 = [0,0]^T$ and $t_2 = [1,1]^T$. So now

I am considering all the four cases so number one case x is input combination is 0 and 0 how to do this computation $||X - t_1|| = ||[0,0]^T - [0,0]^T|| = ||[0,0]^T|| = \sqrt{0^2 + 0^2} = 0$ and $||X - t_2|| = ||[0,0]^T - [1,1]^T|| = ||[-1,-1]^T|| = \sqrt{1+1} = \sqrt{2}$ this is for the input combination 0 0 so I am getting the output 0 and $\sqrt{2}$ and corresponding to this what is $\phi_1(x)$ the $\phi_1(x) = e^{-0} = 1$ and what is $\phi_2(x)$ the $\phi_1(x) = e^{-2} = 0.135$ I am getting this one so that means I am getting 1 and 0.135. Let us consider the second input input is X =(0, 1) and corresponding to this you can determine $||X - t_1|| = 1$ that you can do the computation and$||X - t_2|| = 1$. So, corresponding to this $\phi_1(x) = e^{-1} = 0.367$ and $\phi_2(x) = e^{-1} = 0.367$ so I am getting this.

And similarly 3 number 3 X =(1, 0) we can consider and also the number 4 we can consider corresponding to this X=(1, 0) the $\phi_1(x)$ will be 0.367 and the $\phi_2(x)$ will be 0.367 and finally for the combination X=(1, 1) $\phi_1(x)$ will be 0.135 and $\phi_2(x)$ will be 1. So, in the tabular form I can show like this so X is my input and this $\phi_1(x)$ and $\phi_2(x)$. So, in the tabular form I can show like this so for the combination 0 0 this is 1 and this is 0.135 for the combination 0 1 this is 0.367 and this is 0.367 for the combination 1 0 this is 0.367 and 0.367 and finally for the combination 1 and 1 this is 0.135 and this is 1. So, we are getting this combinations so corresponding to this as shown in my previous slide I will be getting the outputs in the feature space.

So, it is ϕ_1 ϕ_2 so if I consider this point as 1 0 and this is 1 this is 0.5 this is suppose 0.5 and this is the decision boundary. So, from this table corresponding to the first condition the condition is 0 0 so I am getting the output 1 and 0.135 so that means I will be getting a point somewhere here. Similar to these two conditions I will be getting one point in the feature space and that may be here because it is 0.367 and 0.367 so I will be getting this point. So, this point corresponds to the input 0 0 this point corresponds to the input 0 1 and 1 0 and finally for the last 1 1 1 I will be getting this is the point will be something like this here. So, $\phi_1(x)$ will be 0.135 and $\phi_2(x)$ is 1 so this corresponds to the input 1 1 so I can get like this.

So, that means after the mapping into the feature space the classes C_1 and C_2 will be linearly separable.

So, in this case and this will be the class C_1 and this side is class C_2 and you can see the decision boundary between these two classes. So, you can see how I can solve the XOR problem. So, corresponding to this case now I want to show the architecture of the RBF network. So, move to the next slide.

So, now I want to show the RBF neural network for the XOR problem.

RBF neural for XOR problem. So, ϕ_1 is nothing but $\phi_1(||X - t_1||) = e^{-||X-t_1||^2}$ that already I have explained and ϕ_1 is $\phi_2(||X - t_2||) = e^{-||X-t_2||^2}$ and corresponding to this you know the target vectors are t_1=(1, 1) and t_2=(0, 0) because it is a 2 class problem. Now the architecture will be because I have two inputs x_1 and x_2 two inputs we are considering and the target is t_1 t_2 and we have one output node. So, this is one output node. So, I want to show the interconnections here and we are also considering one bias input to the output node that is Y I am getting the output Y here. So, this weight is -1 and this weight is -1.

So, what is the output Y? $Y = -e^{-||X-t_1||^2} - e^{-||X-t_2||^2} + 1$. So, that will be my output. Now the conditions for the classification is if $Y > 0$ then +1 or I can say C_1 otherwise 0. So, this condition I can consider for the classification.

So, this is the final output I am getting. So, here you can see and how I can consider the RBF neural network for the problem of XOR logic. Now let us discuss about the RBF network parameters. RBF network parameters what are the parameters of a RBF network? So, what we have to learn for a RBF neural network with a given architecture. The first one I can say the first parameter is the center of the RBF activation function. The second parameter is the spread of the spread of the Gaussian RBF activation function.

So, for Gaussian RBF function we have to determine the spread also RBF activation function. And also we have to learn the weights from the hidden layer to the output layer. So, these are the parameters we have to learn.

So, there are different learning algorithms which can be used for learning the RBF network parameters. So, I will briefly explain how to learn these parameters.

Now I will explain how to learn the RBF network parameters. So, one is the center the another one is the spread of the Gaussian RBF activation function and the connecting weights. So, these three parameters we have to learn. The center of the RBF activation function and the spread of the Gaussian RBF activation functions can be learned using the k-means algorithm. So, one simple technique is the center of the RBF activation function and the spread of the Gaussian activation function that can be learned by simple k-means algorithm.

Now regarding the weights how to learn the weights. So, weights can be computed by one popular technique that is the pseudo inverse method. So, for this what we can consider the pseudo inverse method we can apply for learning the weights. So, this pseudo inverse method I have explained in my lecture of regression if you see my lecture on regression I have applied this principle the pseudo inverse method to learn the weights. So, let us discuss about how to learn the weights in my next slide.

Learning of weights. So now how to learn the weights by the pseudo inverse method. So, for an example for an example X_i is a vector input vector is X_i and d_i is the desired output. So, corresponding to this output of the network I can write in this form output of the network the output of the network is suppose$Y(X_i) = w_1\phi_1(||X_i - t_1||)+\ldots+w_M\phi_M(||X_i - t_M||)$. So, we are considering M number of weights that is that is the connecting weight between the hidden layer and the output node.

So, we would like $Y(X_i) = d_i$. So, that means one is the actual output another one is the desired output. So, d_i is the desired output. So, we would like $Y(X_i) = d_i$ for each example that means the meaning is $w_1\phi_1(||X_i - t_1||)+\ldots+w_M\phi_M(||X_i - t_M||) = d_i$ that is the desired output corresponding to the input is X_i. So, this expression I can write in the matrix from. So, this expression I can write in the matrix from $[\phi_1(||X_i - t_1||)\ldots\phi_M(||X_i - t_M||][w_1w_2\ldots w_M]^T = d_i$.

So, that means this expression I am writing in the matrix from like this. So, if I consider all the examples now if I consider all the examples now then this equation I can write together by considering all the examples. So, let us move to the next slide for all the examples at the same time we are considering now. So, corresponding to this my previous equation I can write in this form $[\phi_1(||X_1 - t_1||)\ldots\phi_M(||X_1 - t_M||)]$. So, in between I am not showing so finally I am getting $[\phi_1(||X_N - t_1||)\ldots\phi_M(||X_N - t_M||)]$.

So, all the examples I am considering together the weight vector is $[w_1w_2\ldots w_M]$ and what are the outputs the outputs are$[d_1d_2\ldots d_N]$. So, this equation we can solve by pseudo inverse method that already I have explained in my lecture on regression. So, I can employ the pseudo inverse technique.

So, I can employ pseudo inverse technique to determine the weights. So, I can apply another technique that is the gradient method I can apply the learning by gradient method.

So, move to the next slide learning by gradient method. So, we can apply the gradient descent method for finding the centers spread and the weights by minimizing the squared error. So, I have to minimize the minimize the squared error $E = \frac{1}{2}(Y(X) - d)^2$, but and the actual output and d is the desired output. So, this we can determine the minimize the squared error after this I have to update by considering the gradient descent algorithm. So, update for update for the first for the center. So, $\triangle t_j = -\eta_{t_j}\frac{\partial E}{\partial t_j}$.

So, just we are differentiating the error with respect to t_j that is the center of the cluster and also we can determine the spread. So, η is nothing, but the learning rate parameter. So, it can the η can control the learning rate and simply we are doing the differentiation and finding the weights also $\nabla w_{ij} = \eta_{ij}$. So, you can see we can apply this gradient descent algorithm for learning the parameters.

This is another technique. So, now let us compare the RBF network with multilayer neural network. So, in case of the architecture point of view, RBF network have one single hidden layer. So, already I have explained in case of the RBF neural network, we have only one hidden layer. So, single hidden layer is there. But if I consider the feed forward neural network, they may have more hidden layers.

So, more hidden layers in case of the feed forward neural network. And what about the neuron model? So, in the RBF, the neuron model of the hidden neurons is different from the one of the output nodes. So, you can see that is a neuron model corresponding to the hidden neuron is different from the output nodes.

In case of the feed forward neural network, hidden and output neurons share a common neuron model. But in case of the RBF network, the neuron model of the hidden neurons is different from the one of the output nodes. Another point is the hidden layer of the RBF is nonlinear and output layer of the RBF is linear.

Because in case of the RBF network, we are considering the RBF activation function in the hidden layer, which is a nonlinear function. But the output layer we are considering is linear. In case of the feed forward neural network, hidden and output layers of the feed forward network are usually nonlinear, because we use the sigmoid function. So, if you remember that one, already I have discussed about the feed forward neural network. In that case, we considered the sigmoid activation function, which is a nonlinear activation function.

So, this is the comparison with the multilayer neural network. And another point is the activation functions. So, in case of the RBF neural network, we have to compute the Euclidean distance between the input vector and the center of the unit. So, that means in case of the RBF, we have to compute the Euclidean distance between the input vector and the center of that unit particular unit. But in case of the feed forward neural networks, what we need to consider we compute the inner product. So, we compute the inner product of the input vector and the weight vector of that particular neuron.

So, that is another difference between the radial basis function neural network and the feed forward neural network. And corresponding to this approximation, RBF neural network using Gaussian functions, construct local approximation to nonlinear I/O mapping, feed forward neural network construct global approximation to nonlinear I/O mapping. So, in case of the RBF neural network, we are considering the local approximation. So, local approximation, but in case of the feed forward and neural network, we are considering the global approximation to nonlinear I/O mapping. So, this is the difference between the feed forward neural network and the radial basis function neural network.

So, in this class, I explained the concept of the radial basis function neural network. And you can see the difference between the feed forward neural network and the radial basis function neural network. And for explaining the concept of the RBF network, I considered the problem of XOR classification that is a nonlinearly separable classification problem. So, let me stop here today. Thank you.