

**Deep Learning**  
**Prof. Prabir Kumar Biswas**  
**Department of Electronics and Electrical Communication Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture - 46**  
**Normalization**

Hello, welcome back to the NPTEL online certification course on Deep Learning. For last few classes, we were discussing about the challenges which are being faced by the backpropagation learning algorithm or the gradient descent algorithm. And in last few classes, we have seen that how the gradient descent algorithm can be made faster using different optimization techniques like momentum optimization, Nesterov accelerated gradient, RMS prop, Adagrad and all that.

And we have also compared the performance of those different algorithms when we apply those algorithms to make the gradient descent algorithm faster. In today's lecture we will start another topic, again for improvisation of the learning of the gradient descent learning algorithm.

(Refer Slide Time: 01:23)



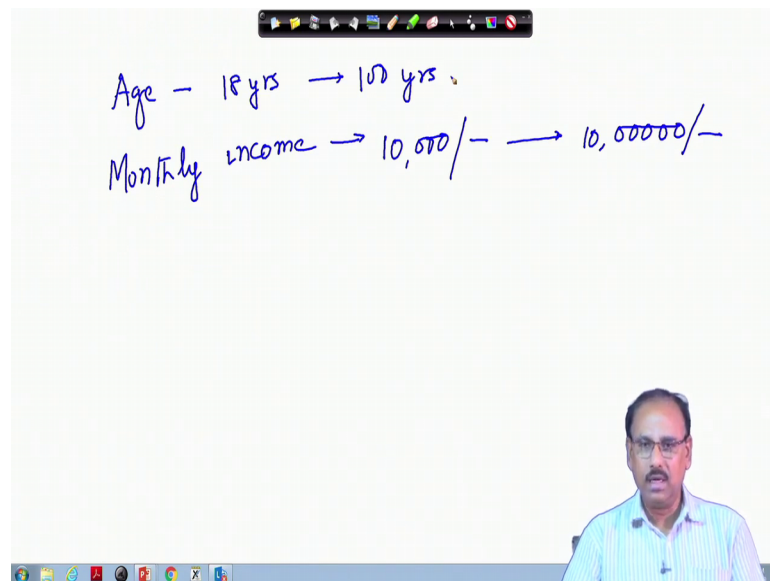
And we will talk about the normalization techniques, and we will see that how this normalization techniques makes the learning stable as well as the learning operation faster. And under normalization we will have different normalization techniques like

batch normalization, layer normalization, instance normalization and group normalization.

So, before we start all that let us see what does a normalization mean. Let us just take an hypothetical example. Say for example, a bank gives loan to the individuals and for giving the loans to the individuals the bank collects a different information.

And suppose, the loan amount of loan or the tenure of loan which is sanctioned to a particular customer of a particular individual is decided based on two parameters, one is the age of the individual and the other parameter can be what is the monthly income of that individual. And depending upon these two parameters the bank decides that what is the amount of loan that can be approved to an individual and what is the tenure of that loan that is in how many instalments the loan has to be paid back.

(Refer Slide Time: 03:05)



So, what we are saying is we are taking two different parameters or two different attributes, one is the age of the individual and other one is the monthly income. You will find that the monthly income of an individual has a wide range. For example, the individual's monthly income may vary from say something like 10,000 rupees a month to even 10 lakhs rupees a month.

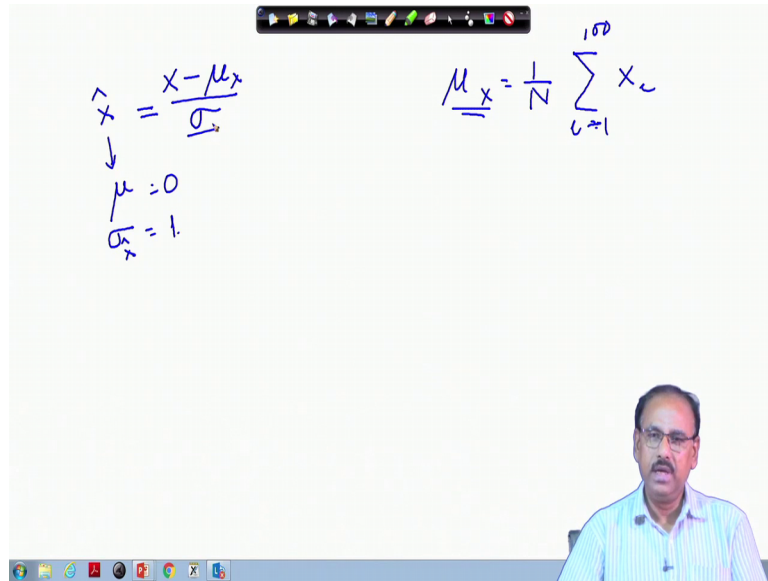
So, the variation in monthly income is very large whereas, if you consider the age of an adult I mean the age at which a loan can be approved that may vary from say 18 years to

may be 100 years. I mean it is very optimistic that if we assume that a person lives beyond 100 years of age. So, considering just these two attributes, you will find that one attribute has a very small range very narrow range from 18 years to approximately it is a 100 years or it may be slightly this side or that side. Whereas, the other attribute which is monthly income that has an wide range may vary from say 10,000 rupees a month to say even 10 lakh rupees a month or may be this side or that side.

So, as a result of this considering just these two parameters, if the bank decides that what is the amount of loan or the what is the tenure of loan that can be sanctioned to an individual you will find that because this monthly income has a wide range in most of the cases the decision will be biased by the attribute which is monthly income because this is many times larger than the other attribute which is range. So, because in most of the time that your decision is going to be biased by an attribute whose values are very large or the range is very large compared to the values and range of other attributes. So, in order to make your decision unbiased you need some sort of normalization techniques.

So, there are different types of normalization techniques that can be used. What one of the normalization techniques that can be that is a I may decide that, all the attribute values can vary from say 0 to 1, right. So, I will normalize all the attribute values in such a way that no attribute will have a value less than 0 and no other attribute will have a value greater than 1. So, that the contribution of all those attributes in the final decision making process is more or less same or all the attributes are equally weighted.

(Refer Slide Time: 06:25)


$$\hat{X} = \frac{X - \mu_x}{\sigma_x}$$
$$\mu_x = 0$$
$$\sigma_{\hat{X}} = 1$$
$$\mu_x = \frac{1}{N} \sum_{i=1}^{100} X_i$$

So, to have this type of normalization techniques, the normalization that we have to use is, suppose  $X$  is the population of different instances of attributes. Say for example, I have different customers or different individuals having ages ranging from say 18 to 100 and I may have say 100 such customers. So, for this customers I have to compute what is the average age. So, the average age is nothing but  $1$  by  $N$  or I can put it as  $\mu_x$  which is nothing, but  $1$  by  $N$  into  $X_i$ , where  $i$  will vary from  $1$  to  $100$ , if I have  $100$  such customers.

So, one of the attributes or one of the normalization techniques can be I can make it  $X$  minus  $\mu_x$ , where  $\mu_x$  is the mean of these attributes divided by, I compute what is the standard deviation of all these attribute values that I have or all these instances that I have. So, I can normalize it with respect to the standard deviation of the all the attributes. So, that is what becomes  $X$  normalized or  $\hat{X}$ .

And if you do this, you will find that because you are subtracting mean for every attribute value. So, this  $\hat{X}$  will have a mean value or  $\mu$  which is equal to  $0$  and it will have an standard deviation say  $\sigma_{\hat{X}}$  which will be equal to  $1$  because you are normalizing with respect to standard deviation.

So, this is one form of normalization where you are making the mean of the attributes which will be equal to  $0$  and the standard deviation or without standard variance is nothing but square of the standard deviation. So, variance or standard deviation will be

equal to 1 and this will be done for all the attributes. So, even the attribute of age will have a mean 0 and standard deviation 1 and the attributes which are income the different instances of income that will also have mean 0 and standard deviation 1. So, this is one form of normalization that can be done.

The other form of normalization can be that suppose I want that the values the attribute values will vary from 0 to 1. So, the minimum attribute value will be 0 and the maximum attribute value will be 1.

(Refer Slide Time: 08:55)

$$\hat{X} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

So, how I can do that? So, given all those attributes what I have to do is I have to find out what is the minimum of all the attributes, what is the minimum value of all the attributes and I also have to compute what is the maximum value of all the attributes.

Then, if I normalization is something like this that I put  $\hat{X}$  is equal to  $X$  minus  $X_{\min}$  divided by  $X_{\max}$  minus  $X_{\min}$  you will find that this  $X_{\max}$  minus  $X_{\min}$  is nothing, but the range of values of the attribute. So, this normalization that  $\hat{X}$  equal to  $X$  minus  $X_{\min}$  upon  $X_{\max}$  minus  $X_{\min}$ , this will make the minimum value of the attributes to be equal to 0 because the minimum value is  $X_{\min}$ .

So, when I subtract its mean from the minimum  $X$  that is nothing equal to 0 or nothing, but 0. So, the minimum value of this  $\hat{X}$  will be equal to 0. And what is the maximum value? If you replace  $X$  by  $X_{\max}$  that is the maximum value of the attribute that I have

you will find that your denominator and numerator both of them become X max minus X min. So, the maximum value of the attributes will also be equal to 1. So, this sort of normalization, normalizes your different attribute values in such a way that the attribute values varies from 0 to 1. It cannot be more than 1, it cannot be less than 0.

So, by doing this sort of normalization you are bringing all the attributes irrespective of its range, irrespective of minimum value, irrespective of maximum value, irrespective of mean all of them are brought to the same scale, so that based on these normalized attributes when you take your decision your decision will not be biased whether it is positively or negatively biased, it will not be biased in anyway by any of the attribute values. So, that is the reason that you need all these different types of normalization.

(Refer Slide Time: 11:33)

Local Response Normalization (Inter-Channel)

$$b_{x,y}^i = \frac{a_{x,y}^i}{\left( k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^j)^2 \right)^\beta}$$

So, earlier we had also talked about one sort of normalization which we said that local response normalization or LRN. And when we discussed about LRN we talked about two different LRN techniques, one was inter channel local response normalization and the other technique was intra channel local response normalization.

So, the purpose why these LRN or local response normalization techniques were used was to limit the activation value from any of the nodes in the neural network. So, the non-linear activation function that are used in deep neural networks are Relu that is rectified linear unit unlike earlier days where the non-linear activation function which were used where the sigmoidal function or tan hyperbolic function. So, if you use

sigmoidal or tan hyperbolic functions, your output is limited to 0 to 1 if you use sigmoidal or the output will be limited from minus 1 to plus 1 if we use tan hyperbolic sigma, tan hyperbolic activation function as the non-linearity. But you know that to take care of the gradient descent problem where the gradient descent suffers from vanishing gradient problem we have preferred Relu as the non-linear activation rather than the sigmoidal or the tan hyperbolic.

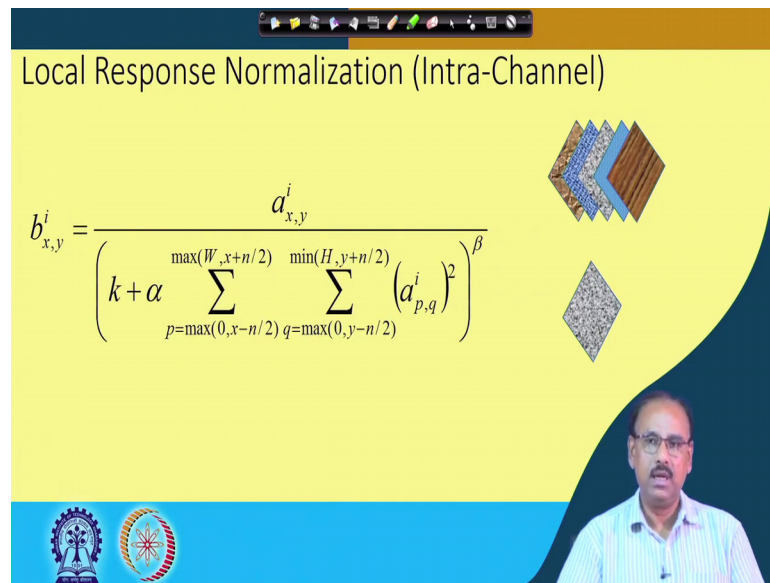
And when it is Relu, then the Relu output is maximum of 0 or  $x$ , where  $x$  is nothing but the linear combination of all the activations that it is receiving from the previous layer and because the output of Relu is same as the input if it is greater than 0 if it is 0 or less; obviously, it will be clamp to 0. So, if  $x$  is greater than 0 then output of the Relu activation function or linear activation function will be same as  $x$ , ok. So, whatever be the value of  $x$ , if  $x$  is very large the output of Relu activation function will also be very large and as a result the output of Relu operation may become indefinite.

So, to make it limited we go for local response normalization. So, the purpose of local response normalization is, one purpose is to make the output value bounded and the second is we have seen earlier that local response normalization can also give a sort of local enhancement of the features.

And when you do this normalization inter channel; that means, at the same location  $x, y$  you are taking the feature values over all the channels and normalizing the  $x, y$  location the feature value at location  $x, y$  in channel  $i$  with respect to the feature value at the same location  $x, y$  in all other channels. So, the expression of this normalization operation in inter channel normalization operation is as given over here. So, this is what is the equation of the inter channel local response normalization.

(Refer Slide Time: 15:11)

Local Response Normalization (Intra-Channel)

$$b_{x,y}^i = \frac{a_{x,y}^i}{\left( k + \alpha \sum_{p=\max(0,x-n/2)}^{\max(W,x+n/2)} \sum_{q=\max(0,y-n/2)}^{\min(H,y+n/2)} (a_{p,q}^i)^2 \right)^\beta}$$


In the other case, if you go for intra channel local response normalization then you are normalizing with respect to the feature values in the same channel that is at a location say  $x, y$  in channel  $i$ , I want to normalize the feature value.

So, for this normalization, I take in the same channel all the neighbouring feature values which are in the neighbourhood of location  $x, y$  and with respect to that neighbourhood you go for normalization of the feature value at location  $x, y$  and that is what is intra channel normalization. So, these are the two types of normalizations which can be used to enhance the features or locally announce the features and also to make the feature values bounded.



(Refer Slide Time: 16:01)

Normalization

- ❑ Normalization that address the problem of covariate shift.
- ❑ Makes learning process faster.
- ❑ Different layers learn independently of others.

What does a classifier learn?

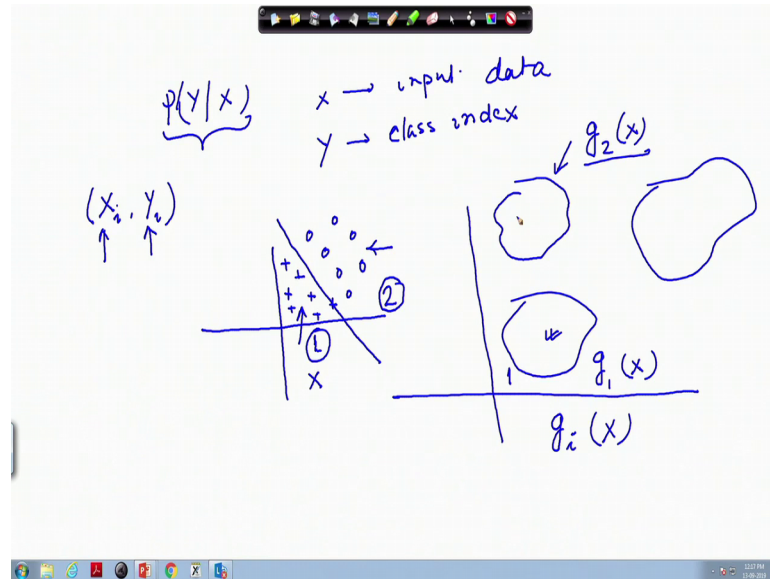
In today's lecture, we are going to talk about normalization techniques. It is the sort of normalization that addresses the problem of covariate shift. So, we will see in a short while on from now that what is this covariate shift.

And as a result this normalization techniques makes the learning process faster and also it makes the learning or the update of the parameters of one layer independent of the update of parameters in other layers. We will see that how all these objectives are made are fulfilled using the normalization techniques, and the kind of normalization that we will talk about are different types of like batch normalization, layer normalization, instance normalization and all that.

But before we go for all these, let us once again what does a classifier learn. So, let us see what does a classifier learn. We have seen that for training of a classifier or basically a classifier, classifies an input data, raw input data into one of the known classes. So, the kind of application domains that we are discussing about are the computer vision domains, where we assume that suppose you have been given an image as an input.

And I have to identify of the classifier or the machine learning machine identifies whether this input image is image of a bird or it is image of a car or image of a flower and all that and that is what is classification problem. So, as a result during training what are the classifier learns is nothing but  $P$  of  $Y$  given  $X$ , where this  $X$  is your input data and  $Y$  is the class index.

(Refer Slide Time: 18:05)



So, the way this input is provided for training of your classifiers or in our case it is training of the deep neural network is you have give input as appear say  $X_i, Y_i$  which indicates that this data  $X_i$  belongs to class  $Y_i$ . So, these are the labelled data which are fed for training of the classifier of training of the deep neural network.

And when you feed a large number of such labelled data and following the learning algorithm or backpropogation gradient descent algorithm. What the classifier learns is basically the distribution or a posteriori probability which is given by  $P$  of  $Y$  given  $X$ , where  $X$  is the distribution of the input data and the probability post a posteriori probability that  $P$  of  $Y$  given  $X$  that is what the classifier learns.

So, if we were having a two class problem; that means, I have the data from two different classes the class 1 and class 2, and the situation may be something like this that for one class the data belongs to say in this region. So, this gives you the distribution of the data of one of the class, whereas in other class the distribution of the data is something like this.

So, in such case the classifier actually learns what is the boundary between these two classes, ok. So, on all this side your  $P$  of  $Y$  given  $X$ , for a given  $X$ , if  $X$  belongs to this positive class we will be at large than  $P$  of  $Y$  given  $X$ , if  $X$  belongs to on the other side where  $Y$  is equal to 1. Let us say this is the region for  $Y$  equal to 1, this is the region for  $Y$  equal to 2. So, if I have an  $X$  which is in this region then  $P$  of  $Y$  give 1 given  $X$  will be

much more than  $P$  of 2 given  $X$  and this is what the classifier learns. Whereas, if we have a multi class classifier then we have the training data from all different classes say in one of the class the data belongs to this region.

So, this is the distribution or range of data belonging to 1 class, this might be range of data belonging to another class, this might be the range of data belongs to belonging to another class and so on and what the classifier learns in that case is some discriminating function  $g_i$ ;  $g_i$  of  $X$ . So, based on the distribution of data the  $g_i$  of  $X$  will be maximum in this if the data comes from this distribution and this is a class 1. So, we call it  $g_1 X$ .

Similarly, for this class we define the discriminating function which is say  $g_2 X$  and  $g_2 X$  will be maximum out of these  $g_1$  and  $g_3$ , if  $X$  belongs to this particular distribution. So, this is what is learnt by the classifier. And based on this knowledge or in the distribution which is learnt by the classifier; the classifier finally, tries to classify the unknown data.

So, now let us come back to our case that how this knowledge can help us in normalization of the data or why I need such normalization and how this normalization process will finally make our learning process more effectively faster and it will be more efficient.

(Refer Slide Time: 22:39)

Why normalization ?

Batch 1

Batch 2

So, the kind of learning or the training that we are talking about is what is known as batch processing or batch learning techniques. In case of batch processing the training data are fed to the neural network or fed to the classifier during learning operation in the form of batches. And it is quite possible that the distribution of the data in one batch may be different from the distribution of data in another batch even though the data comes from the same class.

So, here we are taking an example of a two class classifier, where we want to classify say flowers from all other objects which are not flowers. So, basically we want to classify flowers against non-flowers. So, we are feeding the data for the training of the neural network in batches. So, you will find that on the left hand side the images which are given, so this is the set of images which are from the category flower and these are the images which are category which are non-flowers. Similarly, on in batch 2 these are the images belonging to category flower and these are the images which are from non-flower category. And based on this training data in batches we want to find out that what should be the classifier.

Now, if you look at this you find that in the first batch the flowers are mostly whitish whereas, in the second batch the flowers are very very colourful. So, from these flowers if you compute the feature vectors and then find out what is the distribution of those feature vectors, it is quite possible that in the first case for all the whitish images the features will be distributed somewhere in this region. Whereas, the features for the images which do not belong to flower category they may be distributed in a region something like this.

Going to batch 2, because now the colour has changed the feature distribution may be something like this. All the features which are taken which are computed out of the flowers may come somewhere over here whereas, the features from the non-flower category may be distributed something like this.

So, now, we find that when I try to find out that what is the boundary between these two classes and that is what our classifier is learning in this case. Over here the boundary between the two classes will be like this where on this side of the boundary I have feature vectors of flowers whereas, on this side of the boundary I have feature vectors of non-flowers. Whereas, in other case you find that the boundary between the two classes

will be somewhere over here where these are the features which belong to the flowers which are now coloured and these are the features which comes from the images for non-flowers.

So, as a result you find that depending upon the characteristics of the input data or in this case just the colour of the input data the distribution of the features of the images taken from flower category and the distribution of the features taken from images of the non-flower category they are different, and as a result the classifier which is learnt with the first batch is different from the classifier that is learnt from the second batch.

So, this is the problem this is what is known as covariate shaped and because of this covariate shaped now because the classifier has to hop from one classifier to another classifier during learning, your learning process eventually becomes very very slow. So, let us stop here today. And in our next lecture we will try to see that how the different types of normalization techniques can help to avoid or to address this covariate shaped problem.

Thank you.