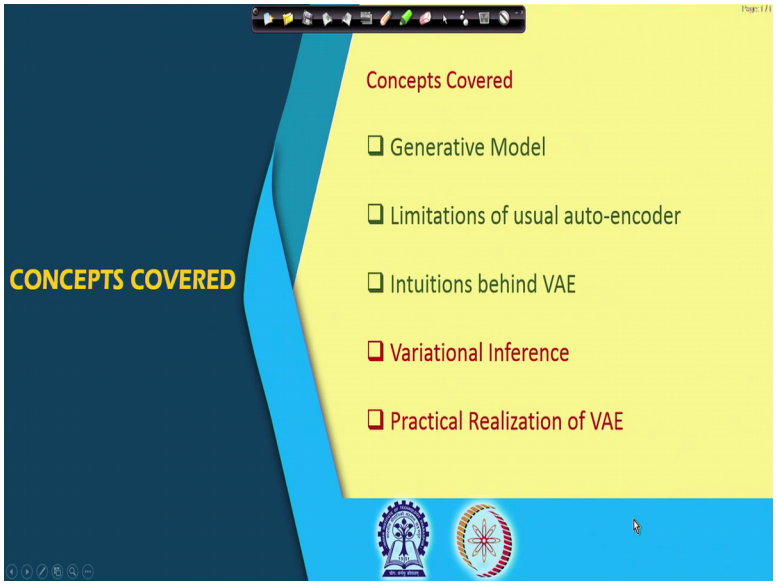
Deep Learning
Prof. Prabir Kumar Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 58
Variational Autoencoder – II

Hello, welcome back to the NPTEL online certification course on Deep Learning. So, from our previous lecture we have started discussion on the generative model and a particular model that we are discussing about is what is known as Variational Autoencoder.

So, unlike in case of discriminative models in case of generative model what you need is that given a latent description of the object of the image, the generator network is expected to generate an image or an impression of that particular object. So, that is what generative network does. Unlike in case of discriminative network where the input to the network is an image or an object and the network has to classify that object into one of the known categories. So, that is the difference between a generative network and a discriminative network.

(Refer Slide Time: 01:25)

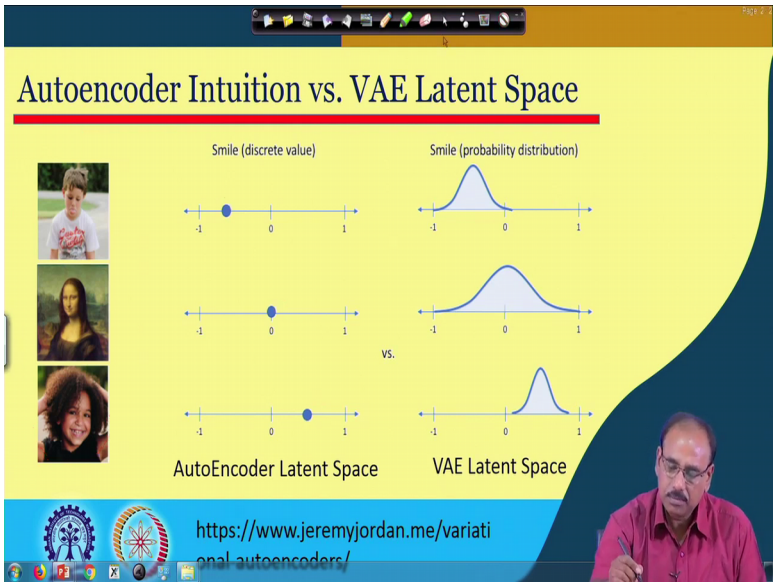


So, in our previous lecture, we have given a brief introduction to the generative model and we have said that what we have discussed previously that is traditional autoencoder, the decoder part of the traditional autoencoder can act as a generative model, but we

have discussed about what is the limitation of that autoencoder or traditional autoencoder why it cannot be used as a generative network.

Then, we have also talked about the intuitions between the variational autoencoder. Today, we will talk about variational inference and we will see that tactically how of variational autoencoder can be implemented. So, let us briefly recapitulate what we have done in our previous lecture.

(Refer Slide Time: 02:15)



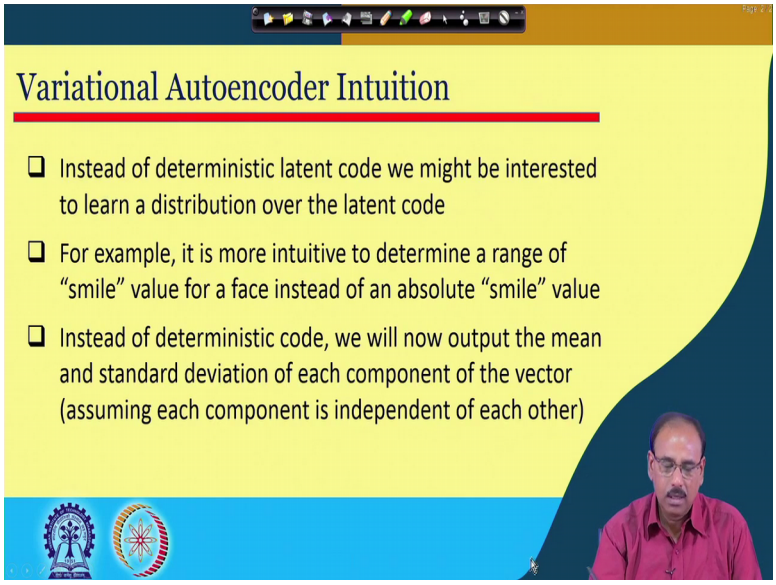
So, this diagram shows we have shown how the latent variable is generated by our traditional autoencoder and a variational autoencoder. So, given these images as shown over here a very variational autoencoder for each attribute which is learnt during the training of the autoencoder, the variational autoencoder for every descriptor or every attribute tries to find out the position of the attribute in the latent space as well as the range of that attribute or the variance of that attribute in the latent space as shown over here.

Whereas, a traditional autoencoder assigns a fixed value to different attributes. Say for example, in this particular set of figures if you look at the first diagram the first image you cannot say that the boy is smiling which is not very apparent. So, a traditional autoencoder assigns a very small value and in this case it is a negative value to that particular attribute smile. Similarly, over here coming to the same attribute smile we cannot say that this is a smiling face, right. So, the traditional autoencoder has assigned

will assign a value something around 0 to that particular attribute smile. Whereas, this is a smiling face. So, traditional autoencoder will assign a high value to this particular attribute smile.

Against, this if you look at what kind of descriptors or the values the variational autoencoder is expected to give it is not expected to give a specific value to a particular attribute, but it is expected to give the position of the mean of that value and around that around that mean what is that range over which that particular attribute may vary. So, that is what is given in case of variational autoencoder as given over here.

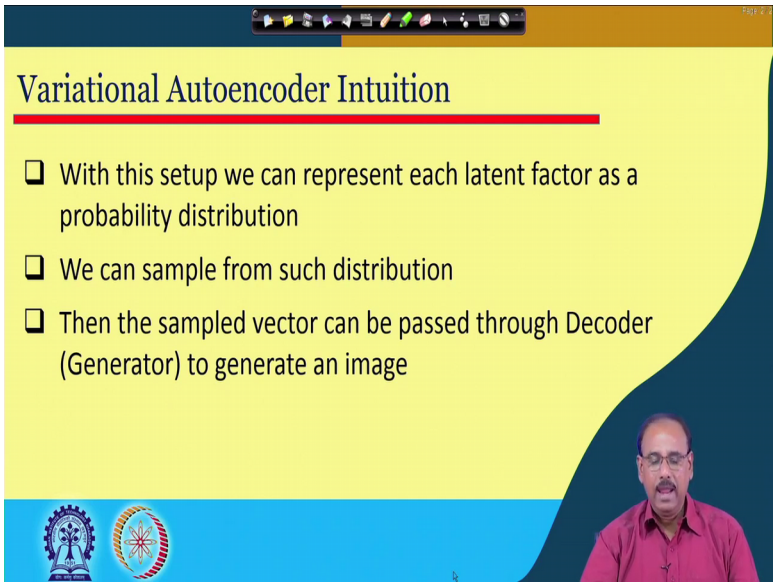
(Refer Slide Time: 04:38)



So, given this, so variational autoencoder instead of giving a deterministic latent code which is given by traditional autoencoder. In case of variational autoencoder, we are interested to learn the distribution of the latent code that is coming if I say that this distribution is a Gaussian distribution or a normal distribution. We are interested to learn what is the mean of that distribution and what is the variance of that distribution.

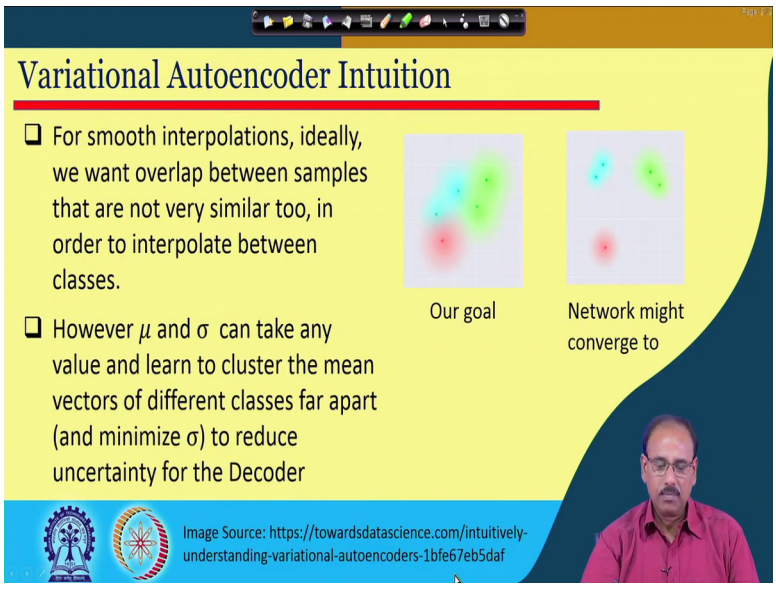
So, for example, as we have just seen that for that particular attribute smile, it is more intuitive to determine the range of smile value for a face instead of an absolute smile value as is given by our traditional autoencoder. So, in case of variational autoencoder instead of a deterministic code the variational autoencoder will output the mean and standard deviation of each component of the latent vector or each attribute which is learnt during the training process.

(Refer Slide Time: 05:41)



So, now as we said that once we have a distribution of the variables of the distribution of the latent vector you can sample a latent vector from that distribution and pass that latent vector to the generator model or the decoder and using that the generator can generate an image. So, that is our aim of a variational autoencoder.

(Refer Slide Time: 06:05)



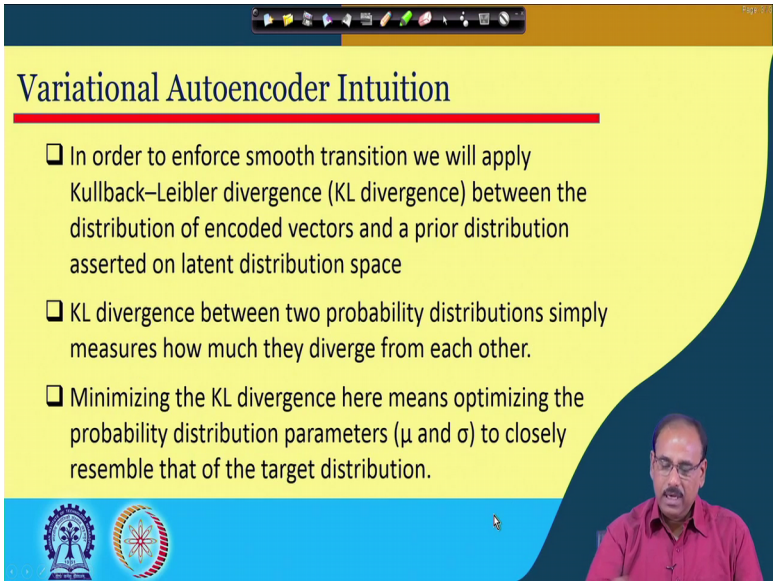
However, because now we are going to learn the distribution we want that to the distribution should be as compact as possible and as we also want to interpolate between

different objects to have different combinations. We also like to have that distributions as generated by different objects should be as close as possible.

So, actually, the kind of distribution that we would like to have is something like this as shown here. But while training the network may converse to a situation of this form, where the means of different categories of images or different classes of images this means become widely different from each other. So, we not only want that the distribution of every category should be compact, we also want the distributions of different categories this should be as close as possible.

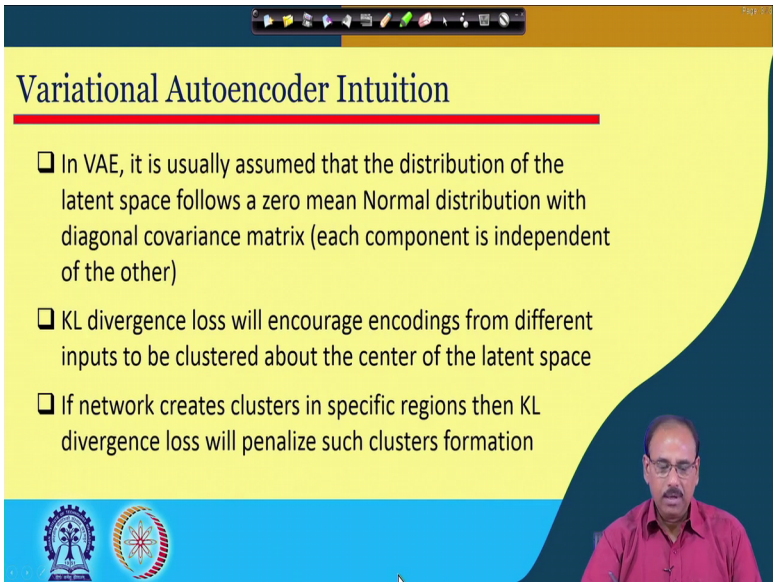
So, in order to do that what you do is you impose a prior distribution. We say that each of the distributions of individual categories should be as close as possible to a prior distribution and this prior distribution is usually taken as a zero mean Gaussian with variance of 1 or unit variance.

(Refer Slide Time: 07:37)



And then you try to find out the KL divergence between each of the distributions; distributions of the latent vectors from each of the categories and the prior distribution that we want to impose. And this KL divergence has to be minimized. So, if we want to minimize this KL divergence basically every distribution tries to be as close as possible to the center in your latent space and all the distributions become very very close.

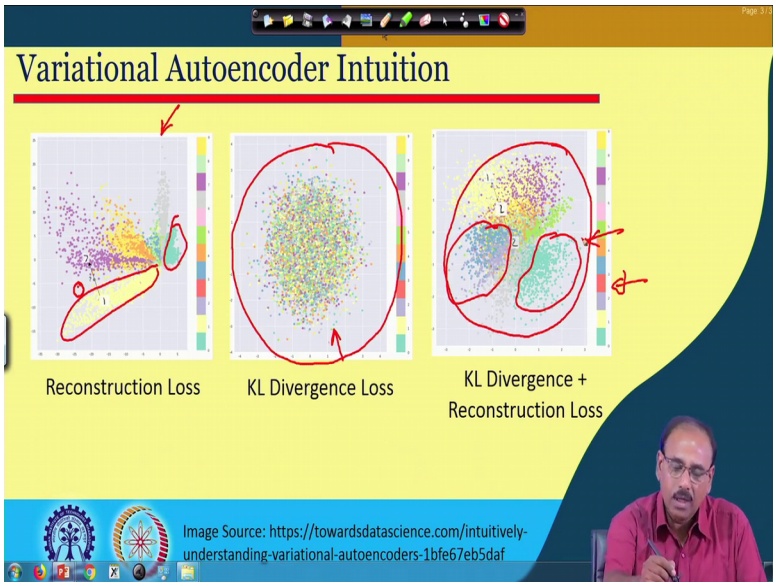
(Refer Slide Time: 08:08)



So, effectively what we get is something like this that for while training the network for every category we have our data loss component and we also have a KL divergence loss component. So, the data loss component or the reconstruction loss component will try to make the distribution of every category very compact and minimization of the KL divergence component will try to encourage encodings from different inputs to be clustered about the center of the latent space.

So, with this now the network creates clusters in, if the network tries to create clusters for different categories in different regions in the latent space then KL divergence loss will penalize that and we will bring try to bring those distributions, those clusters as close as possible towards the center of the latent space. So, as a result the kind of data distribution or the latent vector distribution that we get is something like this.

(Refer Slide Time: 09:14)



If you use only reconstruction loss or only the data loss component in that case the latent vectors coming from the images of the same category they try to form the clusters as shown in this particular image, as shown in this image. Where, you will find that the data coming from different categories they become their form clusters, but these clusters are spread in your feature space of the latent space.

If I used only the KL divergence loss; what we said that the KL divergence loss we try to minimize the KL divergence between a prior probability distribution which is zero mean Gaussian with unit variance and the distribution that we get from the training data. And if I do not impose any reconstruction loss then all these distributions will try to be same as the distribution the normal distribution with zero mean Gaussian with unit variance. And the data distribution for all the classes will be something like this.

So, you will find that one both the cases the generator or the decoder find it difficult to reconstruct the image. In this particular case, if I have if I sample a data point over here, somewhere here, then as this data point was never seen by the decoder; the decoder will not be able to reconstruct what is to be reconstructed out of here, out of this particular latent distribution. So, what the decoder will generate may not be meaningful.

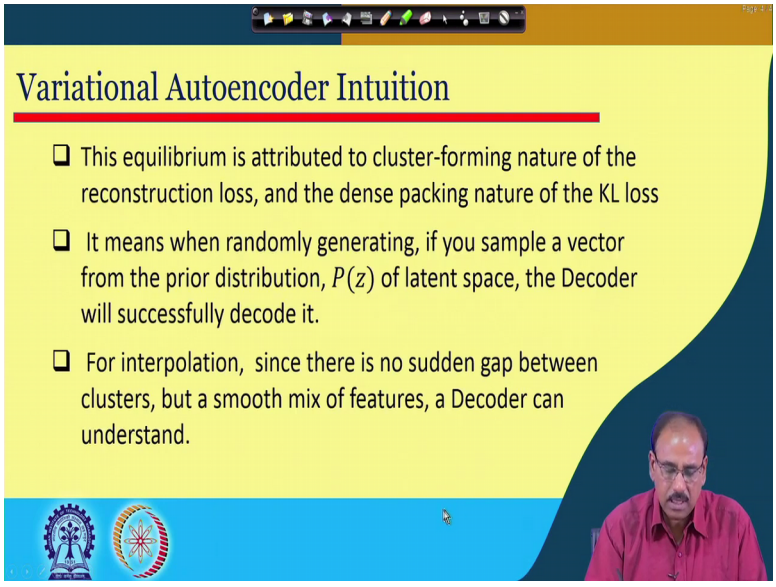
Whereas, if I have a situation like this here you find that the latent vectors from all different categories they are zero mean Gaussian with unit variance. So, as a result you have lost the structural information of the data or the class identity of different data. And

because we have lost the class identity or the structural information of your data again the decoder does not know what to generate because every latent vector coming from all different categories their same to the decoder.

So, here if you find that on the right hand side what we have shown is if you apply both the KL divergence and the reconstruction loss, in that case you find that the data belonging to different categories they have from their own cluster, at the same time globally all these distributions come closer and give a compact cluster of the latent vectors.

And this is the one which is ideally suited for the reconstruction purpose because now my latent bases or the distribution of the feature vectors is very compact. So, if I take any point or sample any point from any of the distributions and pass it to the decoder or the generator the generator will be able to generate a image out of that particular sampled latent vector. So, this is what we had discussed in our previous class.

(Refer Slide Time: 12:34)

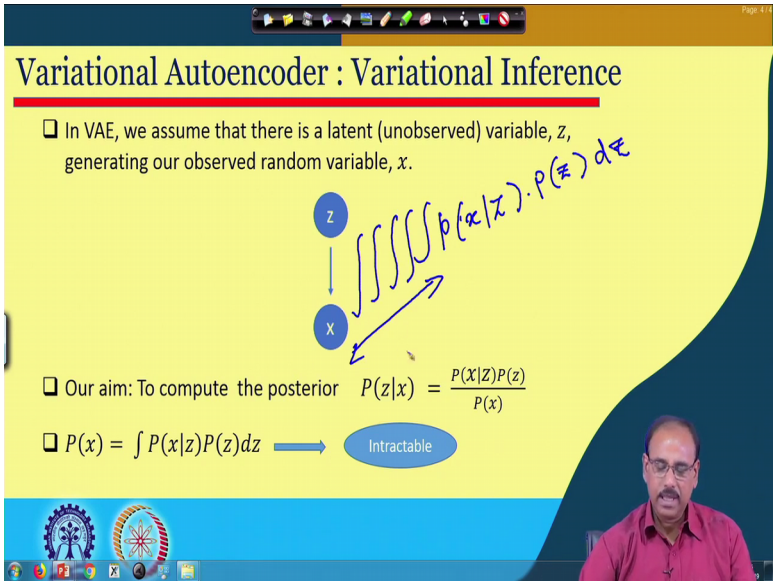


So, here, you find that we have two different components, one is clustering or forming compact clusters of the latent vectors coming from a category of images and this can be attributed to the reconstruction loss or minimization of the reconstruction loss component. And the other pushed part is all the distributions they have to be very close to each other. And this is the part which can be attributed to the minimization of KL divergence loss component.

So, now, as every new cluster is compact and the distributions all of all different categories they are also very close to each other. So, that simply means that, if I simply take a vector a latent vector at random from this latent space then the decoder will be able to successfully decode or reconstruct the corresponding image. And because they are very close, all the distributions are a very close, so it is in a continuous form.

So, if I sample a vector from that latent space which was not used during the training process, but the decoder can give you a reconstruction which is basically interpolation of two different samples in individual distributions. So, when you go for variational autoencoder, the encoder part unlike in case of traditional encoder instead of giving a deterministic vector latent vector it gives you actually a distribution indicated by mean and variance of the distribution, we are assuming that these distributions are normal or Gaussian in nature.

(Refer Slide Time: 14:27)



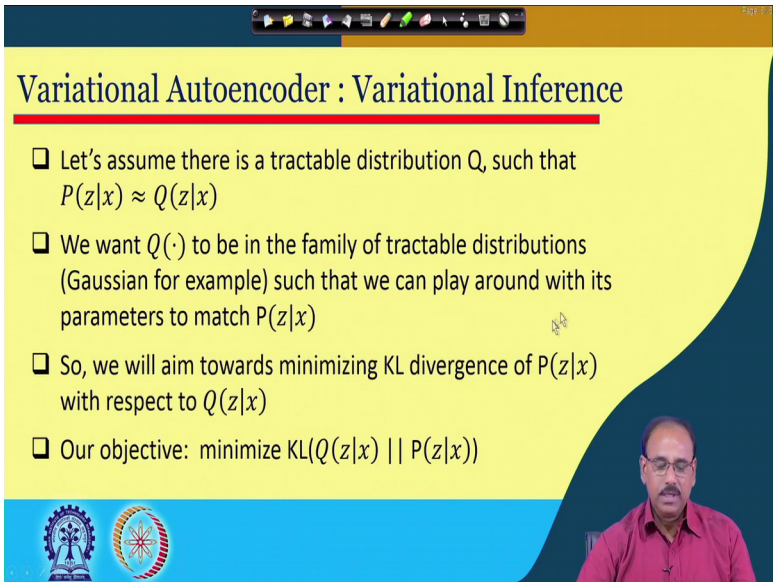
So, effectively what we want is, let us have a graphical model of this variational autoencoder. So, we have our input data which is x and we have a latent vector which is generated out of x which is z. Now, the generator part what will do it will do is this z is sampled from the distribution in the latent space and from this sample z the generator or the decoder is likely to generate my data, I expect that it will generate the data which is x.

Now, as we have just said that in the latent space we also want to maintain the class identity or the structural information of all the data. So, data for different categories will have different structural information and they will have different identities. So, as a result we want to compute a posterior probability which is P z given x, because for x of different categories the probability of getting a latent vector z we will be different. So, we want to compute P z given x and you find that P z given x is nothing but P of x given z into P of z upon P of x and this is what you get from Bayes rule.

Now, over here you will find that if I want to compute P of x which comes in that denominator P of x is simply given by P of x given z into P z into d z you integrate this over z. Now, here comes the first problem of the first challenge that we face, the challenge is that this integral that is integration of P of x given z P z d z it is intractable. Why is it intractable? The reason being when I consider this vector z, the z is a multi-dimensional vector, it is the dimension maybe 10, it its dimension maybe 15, dimension maybe 100 and so on.

And if it is so, then this integration what we have shown over here. Let me change the color that integral P of x given z into P z d z as z is multi-dimension, multi-dimensional, so this integral also has to be taken over a multiple dimension. So, if the dimensionality of z the latent vectors z is 100, then this integration will be over a 100 dimensional space, and which is very very difficult. So, that is the reason we say that this integral is intractable. So, as this integral is intractable then what is the way out?

(Refer Slide Time: 17:33)

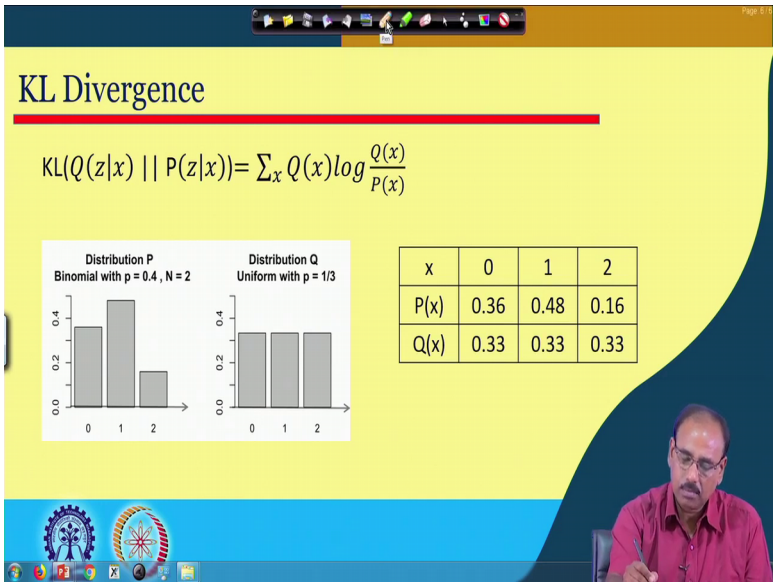


What I can do is we can assume that there is a tractable distribution Q and we want that this P of z given x has to be similar to P of z, a Q of z given x, where this Q is a tractable distribution, fully tractable distribution and we assume that this distribution will be a Gaussian and it will have a mean and a variance.

And if I want to minimize I mean how I can make this P of z given x similar to Qs of z given x I had to minimize the KL divergence between Q and P and by minimization of Q and P what I am effectively doing is, I am effectively playing with the parameters of the distribution P such that it becomes similar to Q, where Q is tractable. We know what it is distribution, right.

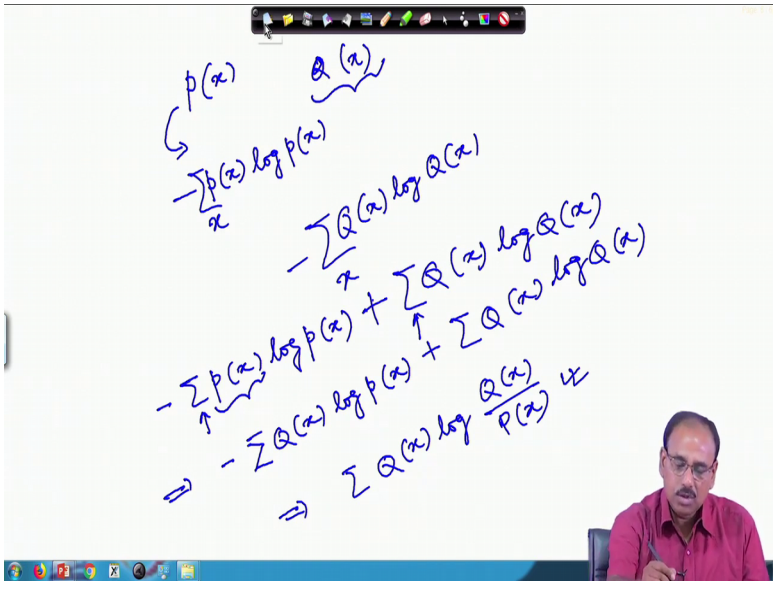
And we are assuming that it is a Gaussian distribution with certain mu mean and certain variance. So, our objective now becomes that we have to minimize the KL divergence between Q and P. So, now, let us say what is this KL divergence.

(Refer Slide Time: 19:00)



The KL divergence is defined as between Q and P is defined as sum of Q x log of Q x upon P x you take the summation over all x that becomes the KL divergence between Q and P. Before going further let us try to discuss that how this KL divergence or what is the genesis of this KL divergence.

(Refer Slide Time: 19:38)



So, suppose we have a probability two probabilities, one is p x, other one is Q x. So, when the probability distribution is p x, we know I can compute the information content. The information content is given by minus p x, then log of p x take the summation over

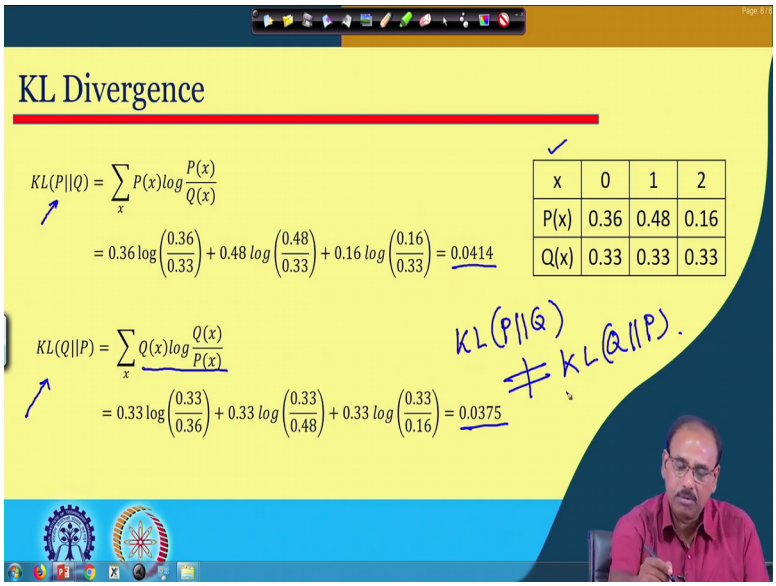
x. So, that gives you the information content when your probability density is P of x. Similarly, the information content when the probability density is Q of x is Q x log of Q x take the summation over x with a negative sign.

So, now if I want to find out what is the difference or the divergence between these two probabilities, I can estimate that using what is the difference of information content of these two probabilities. Or in other words, what I want to find out that minus p x log of p x minus minus Q x log of Q x summation of this, so that becomes plus sum of Q x log of Q x.

Now, in KL divergence, so what is what are these two components? This is nothing, but the expectation value of log of P x, where the expectation is taken with respect to P of x this is the expectation value of log of Q x, where the expectation value is taken with respect to Q of x. Now, in KL divergence this expectation of log of p x is taken with respect to Q x. So, if I take it with respect to Q x this simply becomes minus Q x log p x plus sum of Q x log of Q x. So, effectively this is nothing, but sum of Q x log of Q x upon P of x and this is what is the KL divergence and that is what we have just said.

So, let us come back to this, that the KL divergence between Q and P is given by Q x log of Q x upon P x take the summation over x. And just an illustration, that how can I compute the KL divergence given two different distributions. So, here we have taken two distributions, one distribution is P which is a binomial distribution with p equal to 0 and N equal to 2 and the other distribution is an uniform distribution with p equal to 1 by 3, and this table on the right shows you this distribution in the in a tabular form. So, now, let us see that how we can compute the KL divergence between these two distributions.

(Refer Slide Time: 22:54)



So, given this the KL divergence KL P given Q is simply P x log of P x upon Q x which is if you compute this over here taking the values from this table that we have said the value comes out to be 0.0414. Whereas, if I compute the KL divergence between Q and P, so it was KL divergence P Q, this is KL divergence Q P following this equation. Again, taking the values from this table if you compute this the value comes out to be 0.0375.

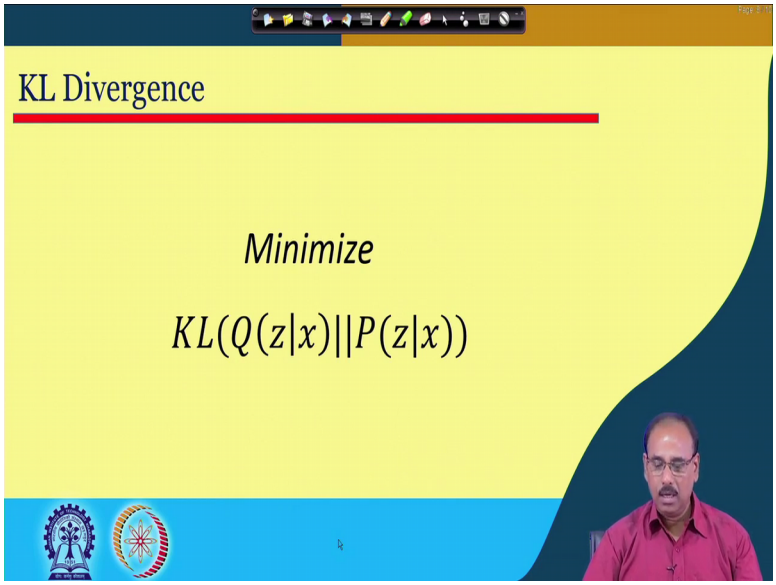
So, this gives you one information that KL divergence between P Q is not same as KL divergence between Q P or KL divergence is not symmetric. So, KL divergence P Q is not same as KL divergence Q P. So, truly speaking, I cannot use it as a distance metric because distance metric is usually symmetrical. And the other thing is the KL divergence is always greater than 0.

So, though it is not symmetric, but KL divergence is always greater than 0. So, using KL divergence I can estimate that what is the divergence or difference between two different distributions and if I want to make those two distributions very close, then what I have to do is I have to minimize the KL divergence and that is what is done in this particular case.

So, what we have seen before is that we have seen that this P of x is intractable. So, if it is intractable then there are two ways to deal with it, one of the option is you go for Monte Carlo simulation and the other option is you go for variational inference

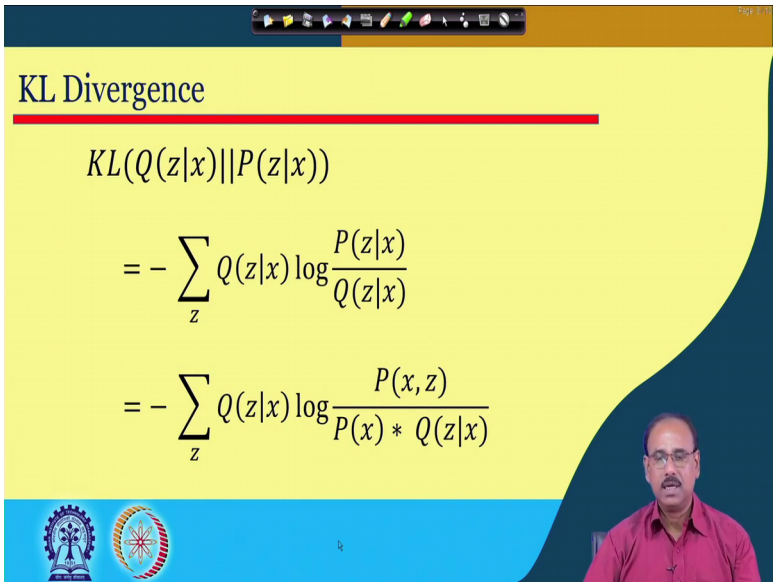
technique. So, what we are going to discuss in this lecture is what is this variational inference technique. So, let us see what is this variational Inference technique.

(Refer Slide Time: 25:15)



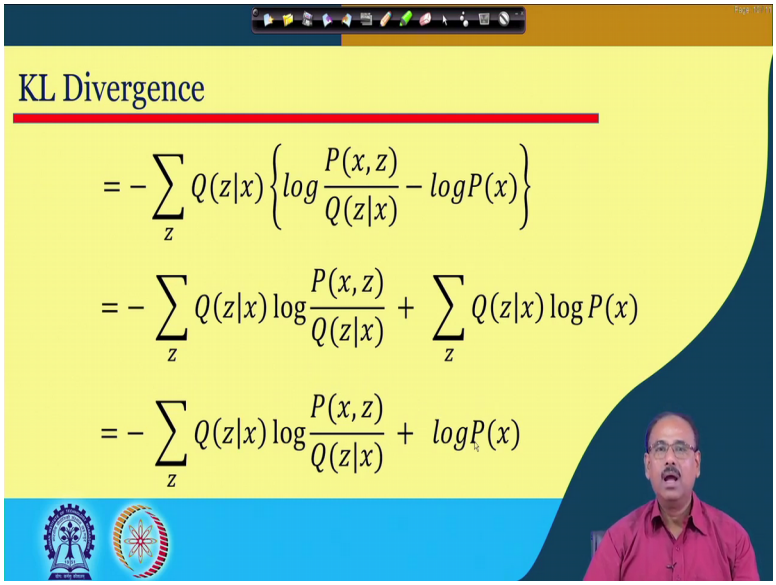
So, you remember that our aim is that we want to minimize the KL divergence Q P, because P is intractable we have assumed a tractable distribution Q z and we want to minimize the KL divergence, so that P becomes similar to Q, the distribution Q which is tractable.

(Refer Slide Time: 25:44)



So, effectively this KL divergence Q z given x and P z given x, as you have just seen that this is nothing, but sum of Q z given x into log of Q z given x upon Q z given x. You can go further, it simply becomes log Q z given x into log of P x z, again coming from your base theory upon P x into (Refer Time: 26:15) that given x you take the summation over z. So, that is the KL divergence between Q and P.

(Refer Slide Time: 26:20)

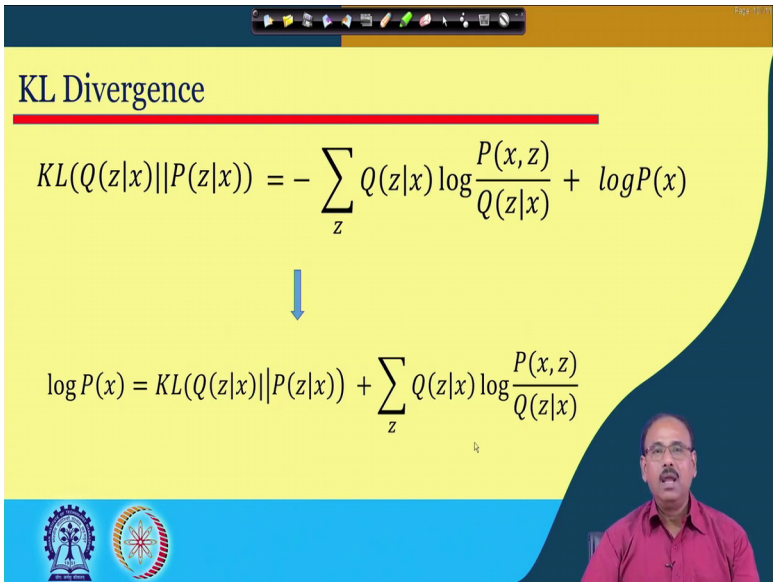


If you expand it further the same expression as you see over here that my expression was Q z given x log of P x z upon P x into Q z x. So, if I expand this simply further, simplify this the expression becomes summation of with a negative sign Q z given x into log of P x z upon Q z x, z given x minus a log of P x. You go on simplifying, it becomes minus sum of Q z given x into log of P x z upon Q z given x plus sum of Q z given x into log of P x summation over z.

Now, here you find that you are taking the expectation, I mean the second component of this expression which is this one, this is nothing but expectation value of log p x with respect to Q z given x and the summation is taken over z and P x is independent of z. So, I can take P x out.

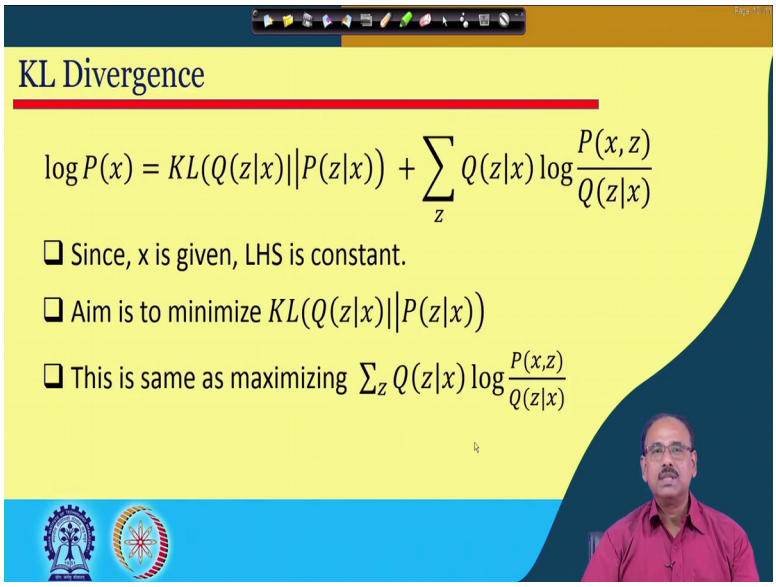
So, effectively what I will have is this, if I take log of P x out of the summation then sum of Q z given x over all z this is nothing but 1. So, the same expression is now simplified to minus sum of Q z given x into log of P x z that is the joint probability upon Q z given x plus log of P x.

(Refer Slide Time: 28:05)



I can derive this expression in this form. So, this is the expression that you have got. You can rewrite this as P of or log of P x is equal to KL divergence between Q and P plus sum of Q z given x into log of P x z upon Q of z given x.

(Refer Slide Time: 28:30)



You go on simplifying further. Here you find that since x is given, so the left hand side or log of P of x log of P given x is fixed because x is done. So, but our aim is to minimize the KL divergence Q and P; KL divergence between Q and P. So, given this expression that log of P x is equal to KL divergence between Q P plus sum of Q z given x into log of

x z upon Q z x and here since the left hand side is constant minimization of KL divergence amounts to maximization of some of Q z z given x into log of P x z upon Q z given x.

So, you find that we have come to a maximization problem, we started with a minimization problem of the KL divergence between two given distributions and while simplifying we have now seen that it is same as maximizing sum of Q z given x into log of P x given z upon Q z given x. We will stop today's lecture here. In our next class, we are going to start from this point onwards.

Thank you.