PDeep Learning for Visual Computing
Prof. Debdoot Sheet
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 52
Principle of Generative Modeling

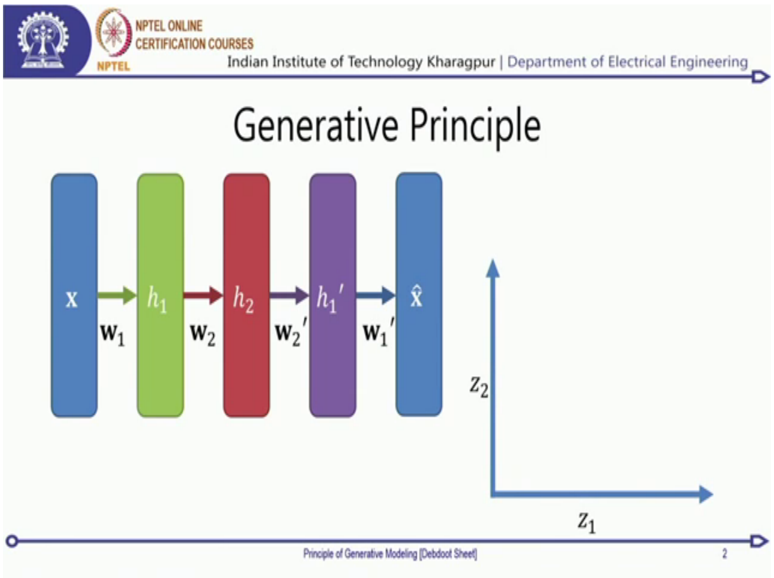
So, welcome in this lecture we are going to start doing with our proceeds of the principle of generative modeling. Now, in the last lecture I had covered down what is the concept of a latent space and based on the concept of a latent variable or a latent space over there how can we start with a generative principle.

Now, it was a very sort of a fuzzy introduction which we had in the last class, but that was more of to get you introduced you onto the factor that any kind of a network which we are dealing with. And taking down a very simple example of a fully connected neural network in order to form an auto encoder I had shown you that this bottleneck layer over there is something which can give you a latent space.

And in some way if you are able to so basically what happens in those kind of a network is you have the whole image which is now transformed in terms of a smaller 1 D tensor. And essentially now if you can replace this 1 D you can pull out any 1 D tensor and place to this decoder you are going to get down something which is equivalent to an image which comes up over there.

But then it is not such a straightforward way what we had seen is that you can have some weird kind of a results coming out over there so that is where I entered down today that now that you know exactly what is a latent space representation. So, today let us get into what is this principle of a generative modeling ok.

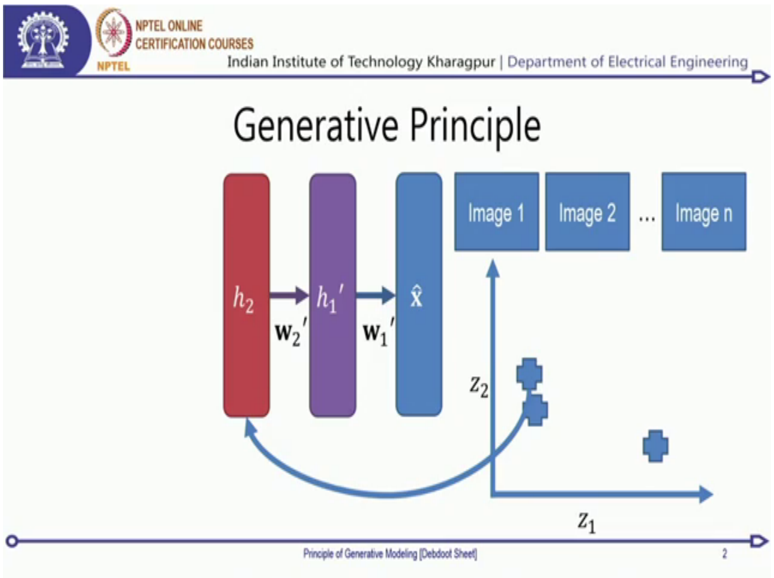
(Refer Slide Time: 01:31)



Now, how we are going to cover is quite straightforward. So, you have your input image X which is given down over here. Now, in your encoder part over there you have your first hidden layer and second hidden layer coming down and then you have two weights which come down over there.

So, this is a very straightforward way of forming down an auto encoder as we had seen in case of a stacked auto encoder building of principle. Now, the whole point was that once this encoder is built up the next part is basically weight to get down by decoder coming down and now in if I try to look into just the space of my decoder and fetch any of these points coming down from a latent space.

(Refer Slide Time: 02:06)



So, over here if I consider that my output over here in case of h 2, is just a two dimensional variable. So, it is just a 2 cross 1 sized a 1 D tensor. Now, if I take an any point over there and feed it over here. So, I am going to get down corresponding output coming over here which will look in terms of an image ok.

So, there is going to be some image because it is its going to be a 2 D form over there whether it makes sense as an image, or not that is not yet known to me but this is the basic principle of generating. So, what I want to essentially do is that I want to take a random number and feed it over here and whatever comes out over there I want to see whether that mimics an image or not ok.

Now, I can take down another number, I can take another number, and then keep on going down. So, what this will help me in doing is that if I am able to get down this sort of a neural network being created where putting a lower dimensional data always gives me a very high dimensional output over there.

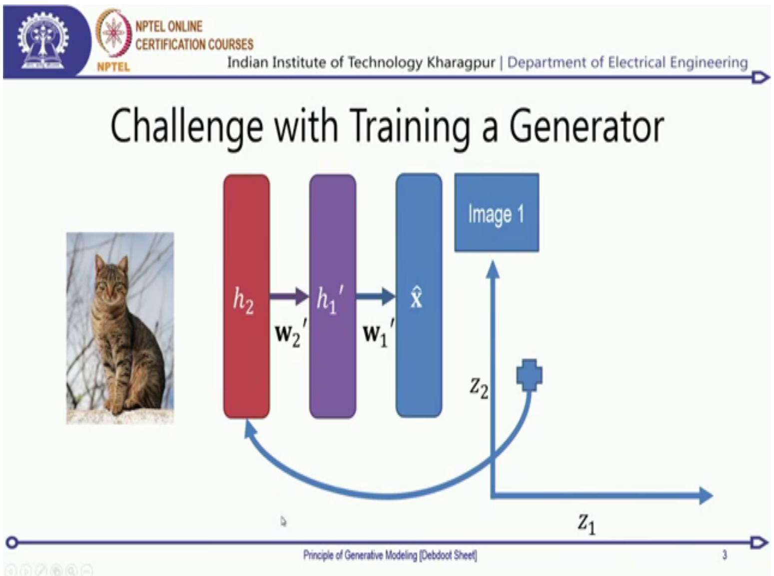
Then now I know essentially I can keep on generating images now what all problems this is going to solve is in a big way. So, one is a trivial problem to say which is we can really create synthetic data sets that is now that is that is a trivial problem if you are looking down into a graphics problem say you want to simulate a crowd and you want unique number of a faces coming down.

Now, it becomes hard to actually get that many number of unique faces to simulate and model out a crowd. Now, using this kind of a generative principle you can definitely get that one well that also imposes one condition that whatever faces are generated they should be looking like human faces.

Now, if one of these faces starts looking alien or some of these face see a majority of the faces which get generated by this random process. If they start appearing to be very alien in nature then that is not going to solve my generative principle in any way the that is more of a brain than a boon in any way.

So, what we are over here trying to do is that we would like to get down images which makes some sense in some way and these do not end up getting noisy or corrupted and nonsensical looking images over there.

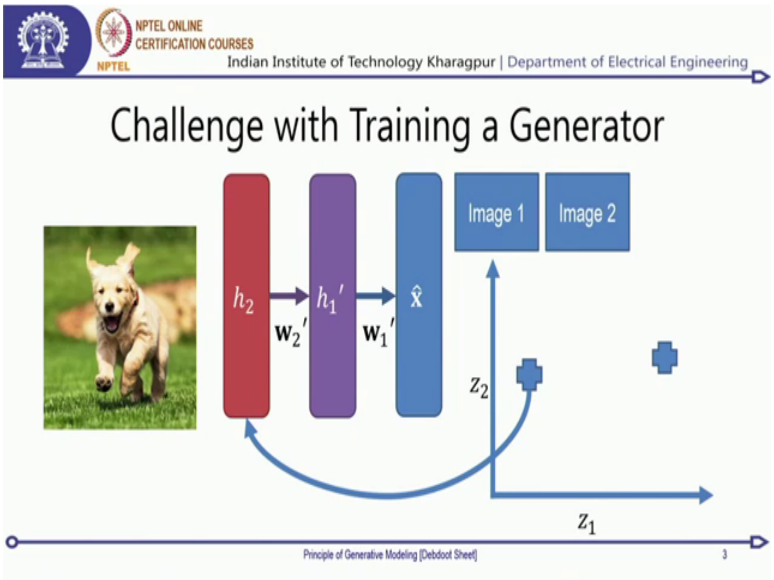
(Refer Slide Time: 04:18)



Now, so that is where comes down this challenge with training a generator. So, I can have this part of my decoder which is created. Now, that is well and good now, if I take some output some of this random value and put it over here now say I get down this image of a cat now I am typically happy that I get done this image of a cat coming down over there.

So, this is what I was exactly explaining.

(Refer Slide Time: 04:44)



Now, I take down another random point over there and then I am I get down a image of a dog ok. Now, what I would essentially expect is that if this point is going to give me an image of a cat. So, I put down just this random number of 2 cross 1 size tensor into this network and I get an output over there and say that output whatever is coming is an image of a cat so this was this part ok.

Now, I am it should be pretty much happy now any point, but the other question is that will it be so that any point I take in this Z once Z 2 space is going to always give me an image of a cat that might not be. So, maybe that some points which are located in the close proximity of this exact point might be giving me an image of a cat. But any point which is located far off may not necessarily give me an image of a cat coming down over there ok.

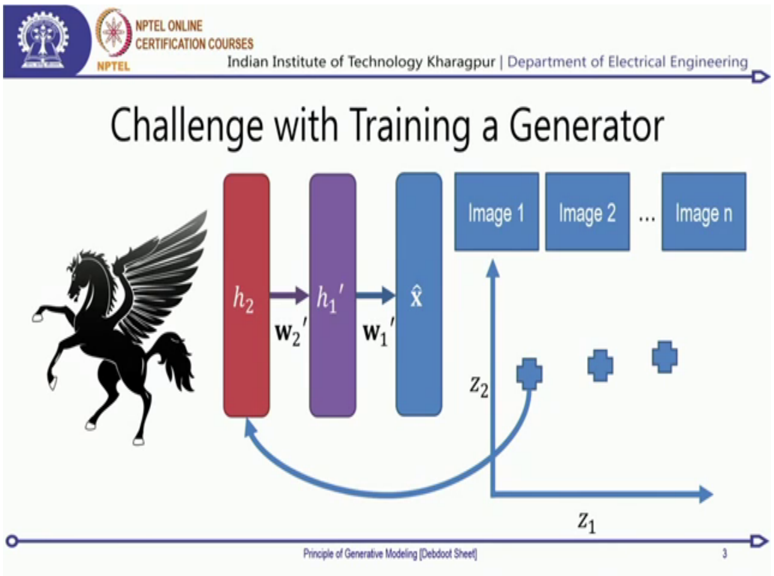
Now, if I take a point over here and try to feed it to this network I get my second image which comes out as an image of a dog ok. So, it means that somewhere over here I will be getting an image of a dog, if I take a point somewhere over here I am going to get an image of a cat. Now, the question is what if I take a point over here somewhere which is not closer to these ones, but very far off or I can take a point which is exactly in between these two.

Now, technically if I take a point in between these two I I can in some way I mean from a very logical reasoning perspective what you can see is that if this point is what is going to give me an image of a cat. Then this point what is going to give me an image of a dog then somewhere in between is going to be say a hybrid between a cat and a dog.

Now, biologically that does not exist in nature you do not have an hybrid between a cat and a dog, but maybe on an image I end up getting something which looks like a cat and a dog, that that can be a pretty much hypothesis you are synthesizing what a new species will look like even if nature has not made those species or even if biology does not permit you to do that.

Now, the question is will that be possible that is that is definitely a major asking which we have over there. Let us see what may happen over there.

(Refer Slide Time: 06:46)



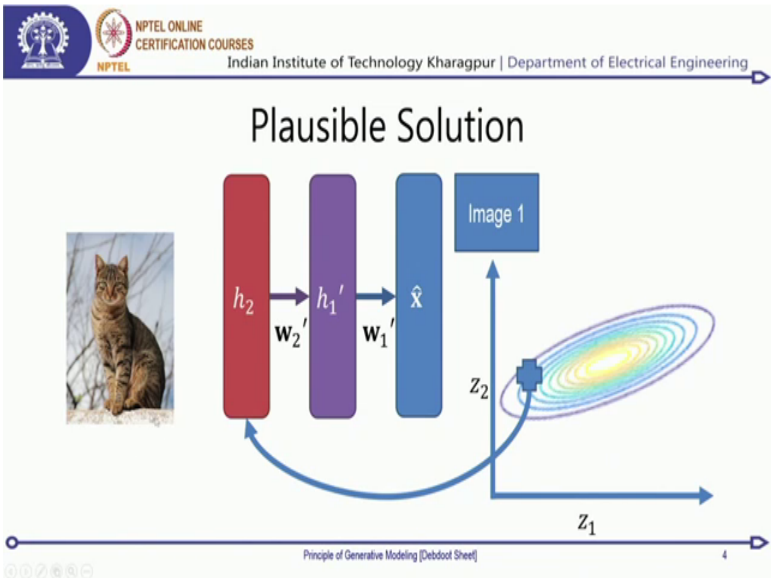
Now, you pull out a sample over here and feed it through this network and then you are going to end up getting something which looks like this. Now, that is a mythical character it is a Pegasus so that is a one of those horses winged horse on which the Gods travel.

So, if you look into Indian mythology you have similar kind of animals as well if you look into Greek mythology you also get done similar kind of animals and Pegasus is not an unknown in case of either the Indian mythology or the Greek mythology any of them coming over there.

Now, nonetheless the main question is that Pegasus is possibly an animal which does not naturally exist and it is not even biologically known to exist over there. So, this definitely creates a challenge over there. So, suddenly in between my this point and this point where I was expecting to find an animal, which might have looked somewhat in between a cat and a dog starts looking like a winged horse or Pegasus.

So, this raises our concern this does definitely raise a major concern among us and we have to find out some way of solving it out. Now, that is where this whole learning with the generative model comes into play like how can we ensure that any point which we pick up over there on this space between two different points or under a certain kind of a given guided distribution actually leads to a very meaningful image. So, this is what we are going to study over here.

(Refer Slide Time: 08:19)



So, now, one of these plausible solutions which we can think of is something of this one that I have my decoder created over there and on this space I try to impose some sort of a distribution function.

Now, if you are quite well aware of this one now it is its even otherwise I can just explain it to you. So, this is basically your 2 D Gaussian. So, a 2 D Gaussian is where you have known about 1 D Gaussian distributions over there. So, we are always keep on telling that something is not drawn down from a random Gaussian distribution or a normal distribution over there.

Now, in case that there is just one random variable to be drawn out and this is going to give you a probability with which it can be drawn out from that distribution now a 2 D is that if you have a 2 D random variable to be drawn out. So, typically you have 2 cross 1 tensor which is made out of Z 1 comma Z 2 in case of your latent representation which goes into h 2 is this what would be making a 2 D variable.

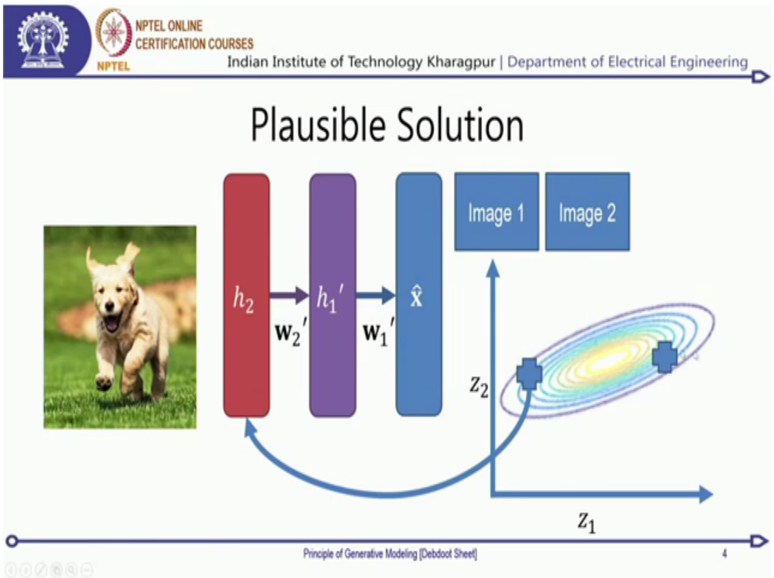
So, now, that you have a 2 D random variable. So, I would try to draw it down from a 2 D Gaussian distribution. So, this is say a possible Gaussian distribution which has a centroid somewhere over here and it has two different variances as said because it is 2 D if you are having isotropic variances along x and y then you would have more of gotten this kind of circular rings.

So, it looks so it will be raised out in space, but here I do not have this third axis coming out of my screen in any way. So, we are just using colors in order to show this one ok. Now, one of these plausible solutions would say that if there is a mechanism I can say that within a certain confidence limit. So, within say plus minus of sigma or plus minus of 2 sigma within this if I am sampling it out I should be getting under a certain confidence this exact kind of a image being generated or some variants around that image.

So, if I say I sent sample out from the centroid, I get down an Indian cat I sample out from somewhere over here, I get down an Italian version of a cat somewhere over here, I get a American version of a cat so that is pretty much possible. So, this is what may be one of the solutions of imposing.

So, let us look at what might happen over there. So, if I sample at this location maybe I am getting this kind of a cat which it is looks to me as an some sort of a snow cat in an in a mechanism in a way.

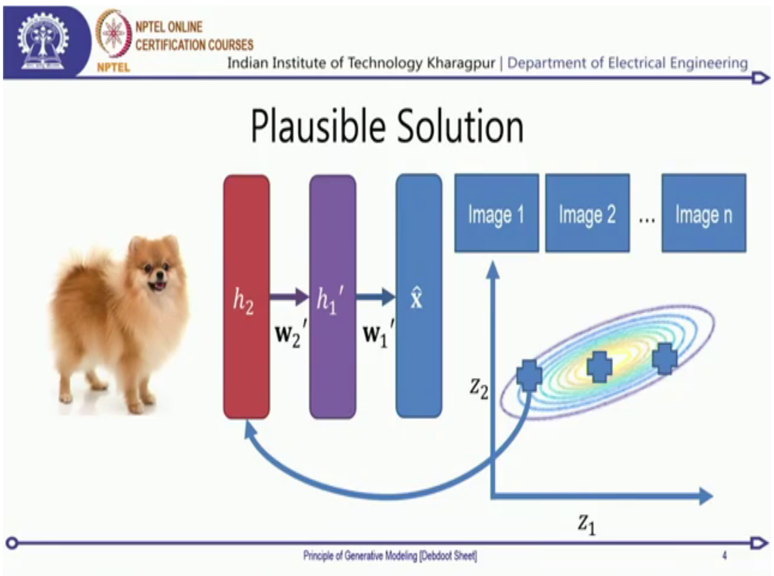
(Refer Slide Time: 10:41)



If I sample out from the other side of it maybe I get down a dog, that is pretty normal. Because these are pretty far off and maybe I was training down a network with a mix of images of cats and dogs.

Now, if I was just training it down only with cats I might be getting only cats if I was training it only with dogs I would be able to synthesize only dogs, but if there are mix of cats and dogs and it might have something over there.

(Refer Slide Time: 11:08)



Now, what this would now imply is that any number which I draw somewhere from the middle with give me an image which will be conformal to either of these animal categories or which will be still looking down as natural and normal.

So, if I am sampling out from this middle point over here in the earlier case you had seen that if you are sampling down from this exact centroid over there you end up getting a mythical animal like Pegasus which does not naturally exist or which will not be over there to be formed out, but when you are now sampling out from this central location over there.

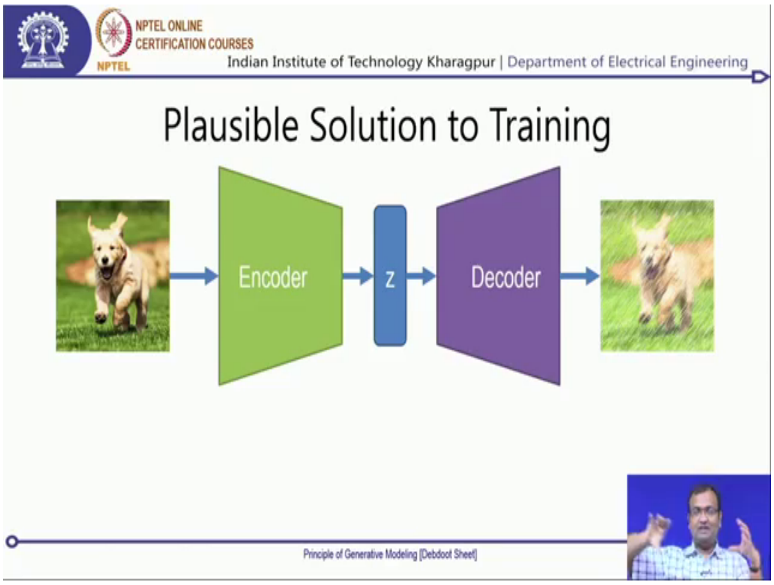
Now, you see that you are getting down this kind of a Pomeranian dog which comes out over there. Now, one plausible explanation is that yes the face is quite fierce and looks almost like a cat maybe for that this is something which got sampled out in between.

So, it has features like the rest of the body looks almost like a dog a fluffy dog over here and the face over here for the dog is a soothing out face, but then this cat had a very fierce looking face over here. And then when you have sampled out from this middle and then you get this kind of a dog the dog has a fierce looking face.

So, certain features which came up from one side of it and certain features from the other side of it and based on where in the manifold you are located well from where you are sampling out it gives you a hybrid looking image coming in over there so that is one of the plausible ways of solving it out.

Now, the challenge is if this is a plausible way of solving it out how can we implement this one. Because I said I am going to draw from a random distribution which from work I am going to draw a random number from a Gaussian distribution. Now, the point is how will you ensure that the number comes from a distribution and then on top of it how will you ensure that it comes from a Gaussian distribution these are challenges which you will face.

(Refer Slide Time: 12:54)



So, one plausible solution which you can have for this training may be something of this sort ok. So, you have an encoder you get a latent space and then you have a decoder this is a very simple formation for a neural network which we are seeing down till now.

So, this latent variable Z is my most critical fact over here and I am my encoder which is a neural network which maps an image to a latent space and I have the decoder which masks my this latent variable onto an image which will be forming on the output over there great.

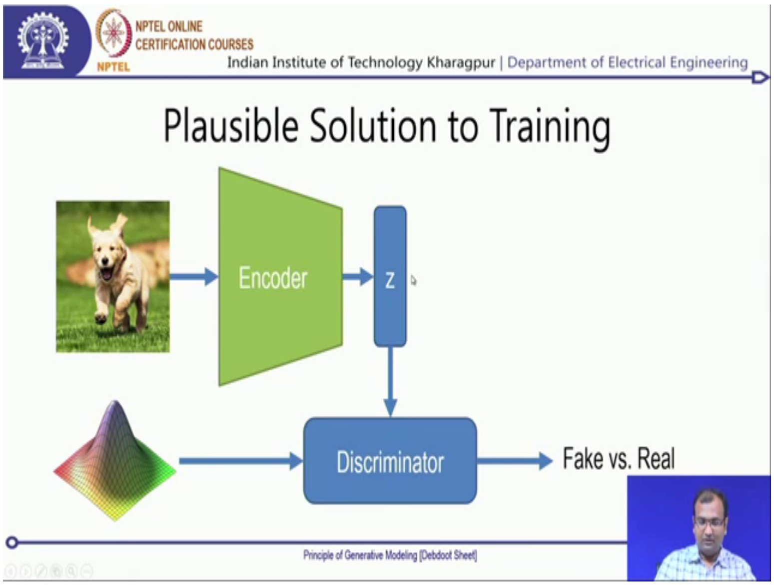
Now, I put an image over here and I am suppose to get down an image with this one. So, this is by training it down in the straightforward form of a standard auto encoder. So, if I have just a plain simple vanilla auto encoder over there then this is how it is going to behave.

So, I have a one image of a dog coming into this encoder I get my latent variable then my decoder and then I have an image of the dog formed over here and though this is somewhat hazy and rustic in nature, but that is again based on that the network has not yet converged and come down to a perfect reproduction stage.

So, whatever is you have produced over there will be some sort of a losses coming down. So, there is some sort of a loss image, but then this mimics the actual content of the image in a significant way over there. Now, in this forward pass what I have done is first

I will put in this image and I will try to reconstruct and then I do a back propagation over there. So, this is going to train my network whereby I am able to reconstruct the original the image in somewhat close to the original form over there.

(Refer Slide Time: 14:24)



Now, in the next pass what I can do is I can pull out a number from a 2 D Gaussian distribution. So, this will be some random number which gets pulled out over there and then I train another neural network to be called as a discriminator.

Now the job of this discriminator is to distinguish whether this one this random number is coming from this 2 D Gaussian over there or is it coming from an image over there. Now, if it comes from an image you can call that as a real image; if it comes from this Gaussian then you can call this as a something which is from a fake or even the other way round.

So, either you can say that the anything which raises out from this Gaussian as real in a way that this is actually following down the real distribution which you would assume that Z follows and whatever is coming from this route is what you would call down as fake ok.

Now, you are going to train this kind of a network over here as well in order to discriminate whether it is coming through this route or whether it is coming from this route just to identify the route of it whether it is from a random distribution or it is from an image based on just the value of this Z over here or the latent variable ok.

Now, in my first pass what I am going to do is if we get back in my first pass of in 1 epoch. So, first I am going to train this as an encoder decoder network to create out a generated image then after this part of my training. So, I will have multiple batches of images over there I finish off over all the batches. So, I have these weights of my encoder and weights of my decoder which are upgraded now all of these updated weights.

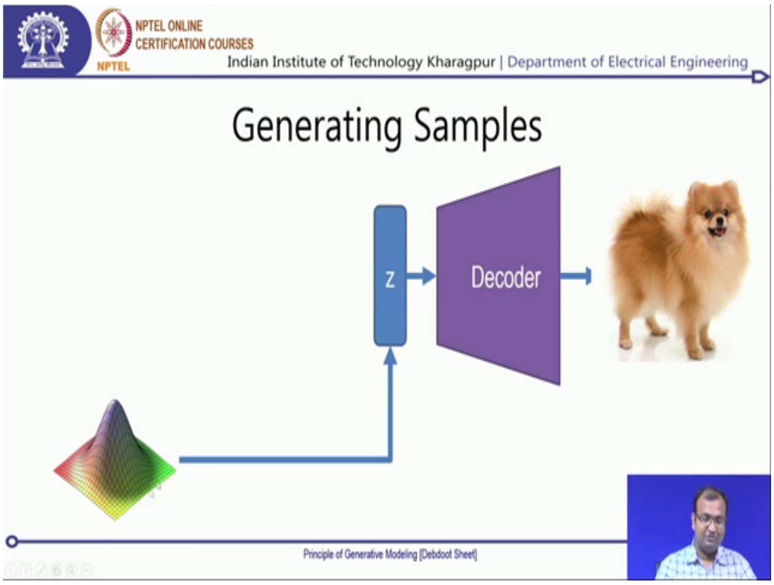
Now, I am going to push all the images which I had in my training set through my encoder and get their corresponding latent variable representation now similar number of random numbers I am going to draw from this random distribution over here and then create this discriminator over here ok. Now, when I am doing a back propagation I would try to do something like this that this error is going to back propagate through this one which is going to update the discriminator as well as this is going to back propagate via this route in order to update the encoder as well ok.

So, now the whole purpose over here is that the discriminator is trying to make itself the robust to be able to discriminate whether it is from here to here and this encoders error which we are passing based on a different rule. So, the encoder is trying to fool and make this discriminator belief that it was generated from this random distribution and not from the image. So, in a sense that this encoder is going to force this latent representation which was otherwise coming in a different way to start mimicking it is behavior to this kind of a distribution and only in that case this discriminator will get fooled.

Now, in the next so this happens within one epoch now in the second epoch you are again going to get back and train this encoder decoder. Now, remember that after this adversarial part of the training which was when I was making fooling the discriminator so have this encoder weights updated. So, that necessarily means that once this update is over here when I am doing a forward pass I will have some changes coming out over here and that error is no more going to be the least error or the zero error coming down so that means, that this decoder will now again get updated.

So, it is a simultaneous update between this encoder and decoder in this first pass in order to be able to generate a real looking image as closer to the possibility and in the second pass of it I am going to one is strengthened my discriminator in order to find the real versus fake as well as strengthen my encoder to be able to fool my discriminator now this is where my training goes on. Now, when I want to generate samples it becomes a very straightforward job so and my encoder decoder which has been trained.

(Refer Slide Time: 17:58)

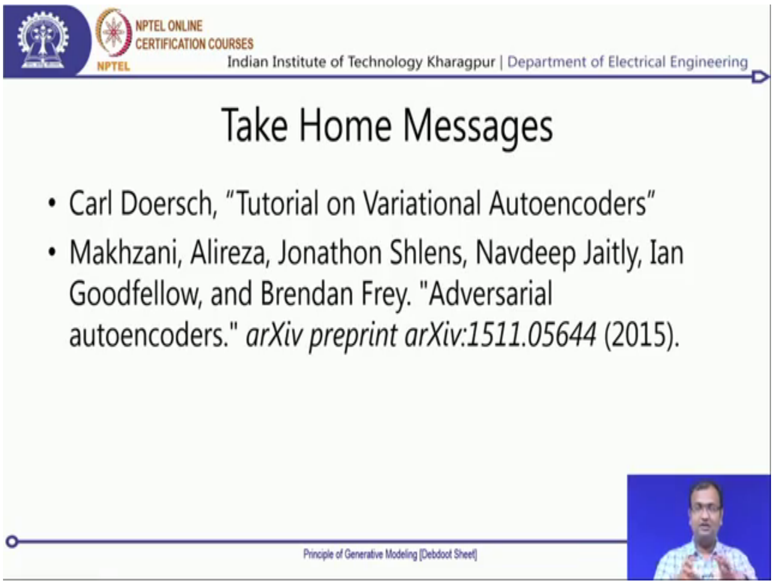


Now, one way is that I can put an image and I get down some version of the image, but now I want to generate synthetic samples and what that would mean is that I am going to take one of these random distributions take a number from here feed it to this decoder and get down an image coming down over here.

Now, this would be easy to get an image given the fact that this decoder is trying to generate an image looking at anything which comes on the Z and this Z on the other side your encoder was always trying to make this Z look very much similar to as if it was generated from this real Gaussian process over here.

And that is how you would be fooling it out and making it conformal to be able to generate a real looking image. So, this is the concept of how you would be able to generate out samples using this kind of a network.

(Refer Slide Time: 18:51)



So, if you want to read more about them then the first tutorial which I would typically suggest is in order to understand more about is on this tutorial on variational auto encoders. And then you definitely have this particular paper on adversarial auto encoders by Ian Goodfellow.

Now, this is a very important piece and a treatise on this particular way of doing and the explanations which I have been using in this particular presentation on today's lecture are based more of a on adversarial auto encoders and is a very simple way so and now this is not just the only way of doing it.

So, you have a generative adversarial networks as well so that is another family of doing it out, but we are sticking down to a very simple one to do the philosophy. So, once you are able to understand an adversarial auto encoder it makes it easier to understand even a generative adversarial network as well.

So, till then we stay tuned and in the next class I am going to get into more details about adversarial auto encoders and some of it is variants and what are the advantages you can get out of it with then.

Thanks.