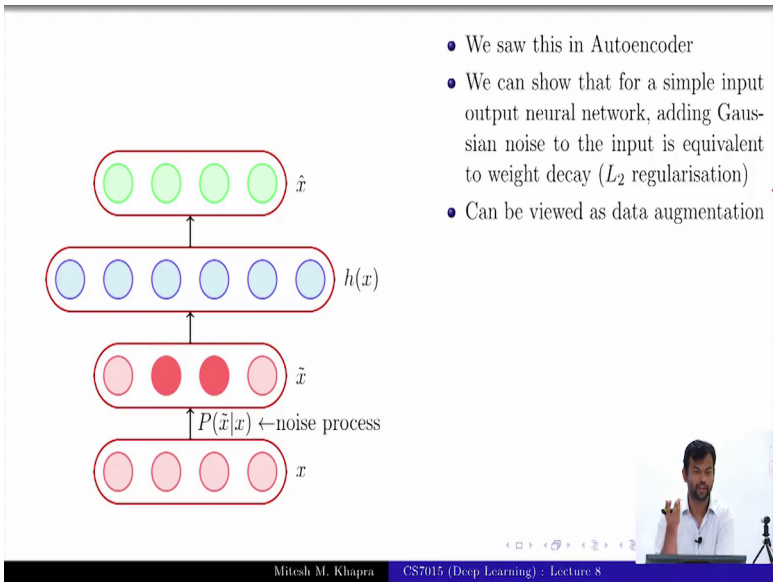
Deep Learning
Prof. Mithesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 8.7
Lecture – 08
Adding Noise to the Inputs

We go down the next module, which is adding noise to the inputs right.

(Refer Slide Time: 00:17)



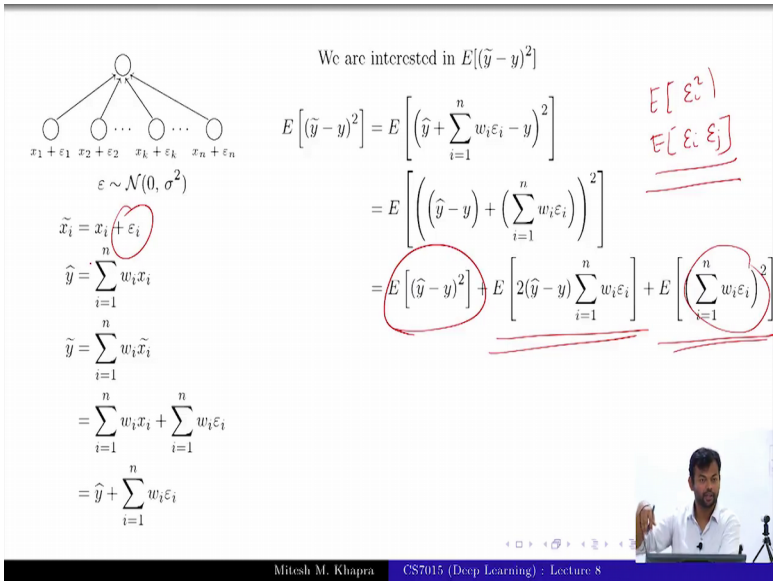
So, we have some kind of a noise process and now can you relate that how that was related to regularization, that was exactly the motivation in that case that we could have an over complete auto encoder which is a very complex model, because it has a large number of parameters.

And to avoid that we were adding this noise to the inputs so that even if it tries to minimize the training error, it is not actually minimizing the true training error right, because you have fed some noise to it everyone gets this? right. Now actually we can show that for a simple input output neural network right; that means, you do not have any hidden layer you just have a set of inputs and you have the output layer. Then adding noise to the input or rather adding Gaussian noise to the input, it is equivalent to weight decay.

So this can also be viewed; so, we will do this part right. So, we will just quickly do a small derivation, where we show that adding Gaussian noise to the inputs is the same as doing a l 2 regularization, that is a very neat idea. So, this can also be viewed as data augmentation right. Exactly what I shown on the previous slide you added 2 you just corrupted some inputs of it that is the same as adding noise to the data.

So, the essentially augmenting the data right, you have some training data and just augmenting it. So, to get more training data is that fine ok.

(Refer Slide Time: 01:41)



Now, about this smallest derivation this is again just a set of steps, I will go over it reasonably fast. I will give you the set up and then it is quickly work through the derivation right.

So, what I was trying to say is that if you have a simple input output neural network; that means, you just have inputs and the output you do not have a hidden layer, right? Then adding a Gaussian noise to the input units where the noise comes from this distribution; it is a Gaussian distribution 0 mean. I want to show that doing this is effectiveness the same as doing l 2 regularization ok.

Now, again see this is the same thing squared again a way vacuum because this is not the kind of networks that we deal with, but it is good to see what happens at least in these neat conditions, because we will never have a simple input output network, at least not in

this course. We will have a deep neural network always. So, but at least she what happens in the simple case right. So, what we are doing is from the x i's we are creating a noisy x i by just adding some epsilon noise to that. And what is our model going to be? It is just an aggregation of all the inputs. So, this is what our original model would have been without the noise fine.

I would have just aggregated all the inputs, I am assuming there is no non-linearity at the output and I am just taking y i is equal to summation of all my inputs everyone fine with this side. Or this is too simple for you guys to understand because we have been doing a lot of deep neural networks. So, suddenly one-layer network I do not know what it is again gets it right.

And instead of y hat, now I have y tilde because instead of x i I have x i tilde ok, but what is x i tilde x i plus epsilon i right. So, I can write it as this just fine. So, actually y tilde is nothing but y hat plus some quantity. What are we interested in? Always this quantity the expected mean square error ok? I mean expected squared error and why not y hat?

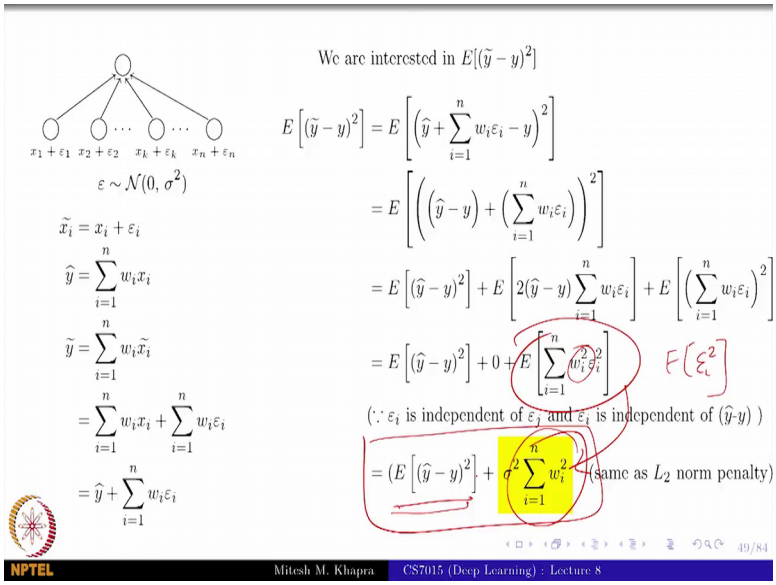
So, we have added noise to the input. So now, y tilde are the outputs that we are going to tilde. So, let us see what that quantity is; and again just going to be some simple stuff. So, I replaced y tilde by this that we just derived on the right hand side, on the left hand side ok. So, I am going to take these 2 terms together; so, I can write it as this plus this the whole square fine. And I am going to keep this as it is. What is this quantity? The original squared error expected squared error right, when I was not adding noise to the inputs ok. and you see how we got these 2 quantities, this is just a plus b the whole square is equal to whatever it is equal to right? Now let us look at the last term this is a square of a sum right.

So, what kind of terms would you have inside? You will have some terms which are epsilon i squares and you would have some terms which were epsilon i epsilon j right ok. So, we will have some expectations which are going to be something into epsilon i square, and some expectations which are going to be epsilon i, epsilon j. Everyone gets this? Some terms there; now which of these terms would disappear?

Student: (Refer Time: 05:09).

These terms right, why? Because the noises are independent ok. I am notl if I have drawn a noise for one instance; it does not have any influence on the noise that I am going to add to the next instance. If I have taken one x I, corrupted it with some noise there is no bearing on the noise that I am going to use for the next epsilon i right? All these features are the noise added to the features are independent, right is it ok? Fine.

(Refer Slide Time: 05:38)



So now from these terms only the square terms are going to remain, is that fine? And similarly this quantity, what can you say about this? We just did something similar, why I am a saying that this is going to 0? Again I can show that this is the covariance between this random variable and this random variable. And now are these 2 random variables dependent? What is epsilon i? The noise that I am adding to the input, does it have any effect on y hat no, right? Because y hat does not depend on the noise, what is y? True output does it have anything to do with the noise? No right.

So, that is why these 2 random variables are independent. So, I can again write there the expectation of their product as a product of expectations, and then the expectation of this is going to be 0, because epsilon i was drawn from a 0 mean distribution is that fine everyone gets that the same trickery that we did earlier. So, this is the quantity that we are left with, you see how I got from here to here. This is an expectation of a sum, which is equal to a sum of expectations, w i has nothing to do with it is not a random variable.

So, it is just the expectation of sigma i square which is nothing but the variance right. So, I get this what does this look like? I already told you the answer before starting right, this looks like l 2 regularization this is the true error, I mean this is the empirical estimate from the training error. And this is the weight decay term everyone get this? How you see that this is an equivalent thing? So, at least in this neat set up you get the intuition that adding noise to the inputs is a same as adding a l 2 regularization term, everyone is fine with this?