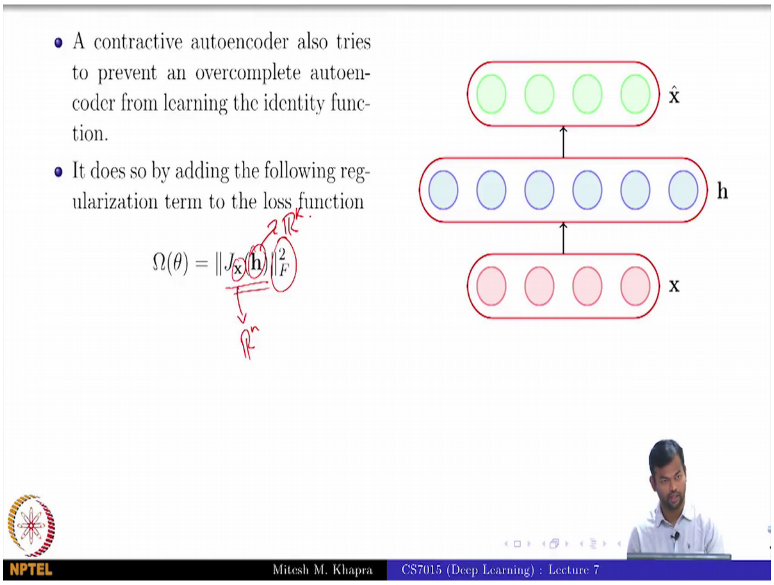
Deep Learning
Prof. Mitesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 7.6
Lecture – 07
Contractive Autoencoders

So, with that we will move on to something known as Contractive Autoencoder. So, this is yet another type of auto encoders again with the same aim that you want to do some kind of a regularization ok.

(Refer Slide Time: 00:25)



So, it again tries to prevent and over complete auto encoder or even an under complete auto encoder for that point, from learning the identity function right.

So, it does not allow you to simply copy the inputs to the outputs ok, that is what it is trying to learn. And it does so by adding the following the regularization term to last function and the way it does this is by defining the following regularization term ok. What is this term? Ok, let us see some things which we already know, what is this? Frobenius norm of some matrix, what is this matrix?

Student: Jacobean.

Jacobean, what is the Jacobean?

Student: (Refer Time: 01:00).

What are the two variables here that you see?

Student: H.

H and?

Student: X.

H is a scalar matrix vector.

Student: Vector.

Vector; x?

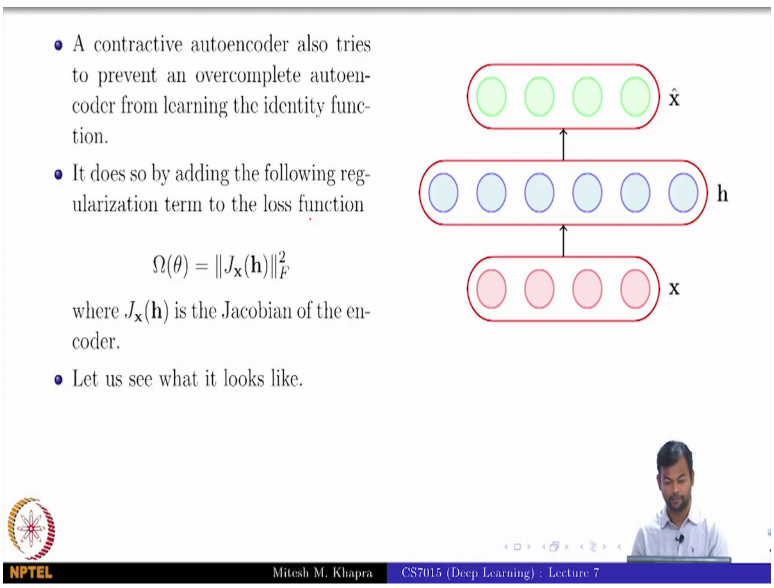
Student: Vector.

Vector right. So, it is some function between two vectors and it is a matrix. So, take a guess how many entries would not you have, if x is R n and h is R k.

Student: N cross k.

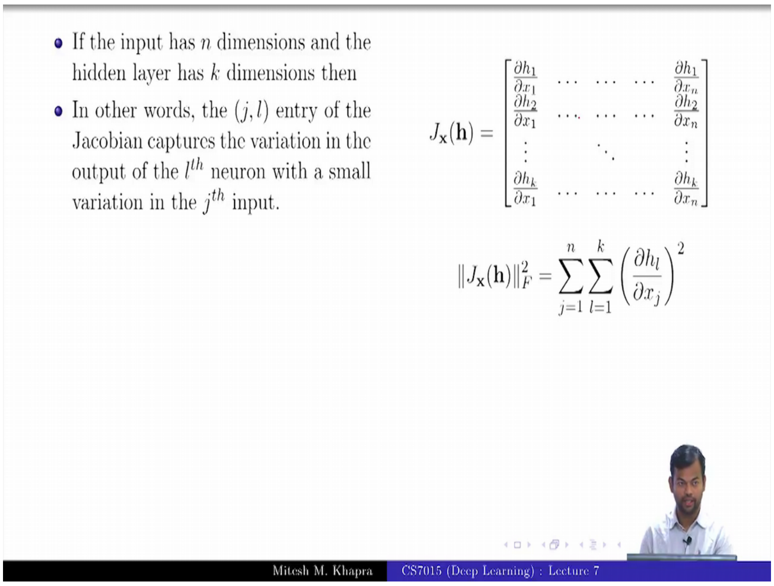
N cross k, even if you do not know what the entries are you are able to guess that it is going to be a n cross k matrix right.

(Refer Slide Time: 01:34)



Now, let us see what this n cross k matrix looks like ok.

(Refer Slide Time: 01:37)



So, it has, the input has n dimensions and the hidden layer has k dimensions. So, this is what the Jacobean looks like.

What is the first column? If the partial derivative of every neuron in the first hidden layer with respect to the first input right and now you can see what the other columns would be. This is what the Jacobean is, this basically the derivative of h with respect to the vector x, answer is just you are taking a derivative of a vector with respect to another vector you will get a matrix as the output ok. Now, what does the j lth entry here capture actually?

Student: (Refer Time: 02:12).

What does a derivative capture?

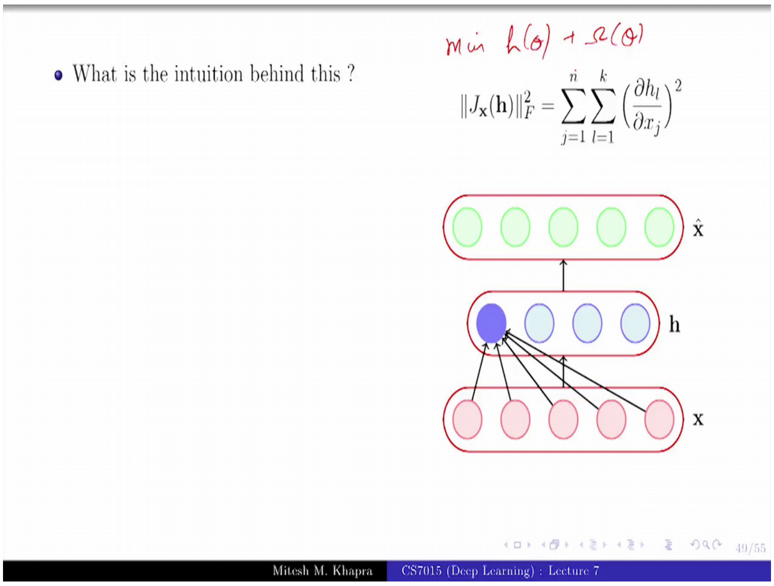
Student: (Refer Time: 02:14).

How much does h l change with a small change in.

Student: x k.

X k right, that is what a derivative captures is that fine and then what does the frobenius norm capture, it is just the square of sum of the square of all the elements of the matrix right. So, it is basically how much each of these elements vary with respect to the input and we are just taking the square of that. So, you see what is the term that we have added, ok.

(Refer Slide Time: 02:44)



Now, tell me what is intuition behind this ok. So, when would this term; so, remember this term is added to the loss function and you are trying to minimize the loss function. So, that means, you want this term to go to.

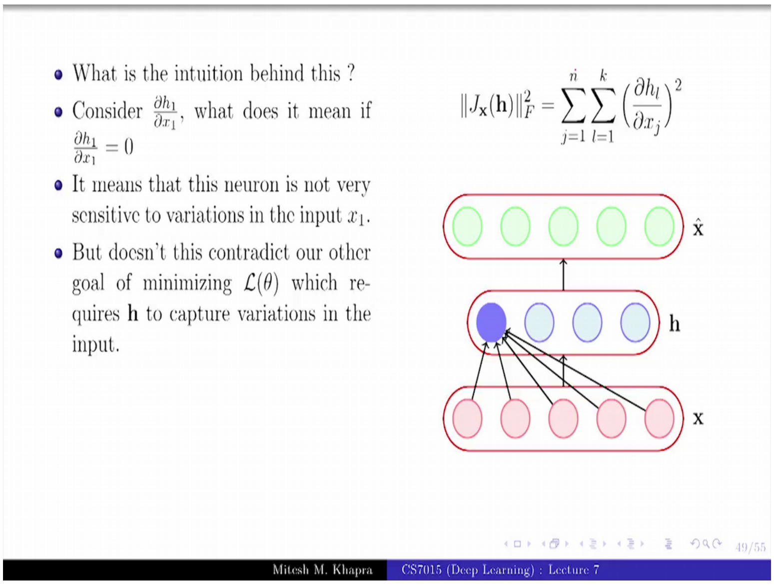
Student: (Refer Time: 03:01).

You want the frobenius norm to be.

Student: 0 (Refer Time: 03:03).

0 right ideally of course, that will not happen because there is always a tradeoff between l theta and omega theta, if you make it 0 then l theta would be very high right.

(Refer Slide Time: 03:17)



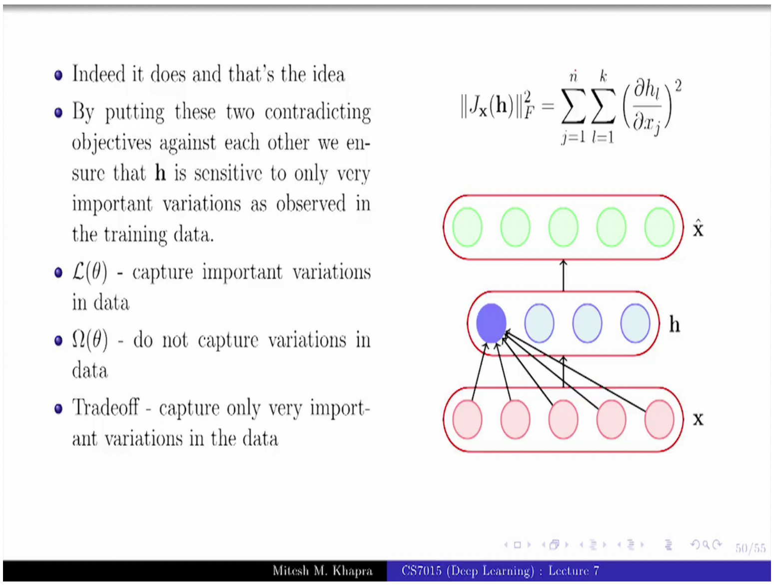
So, now what would happen, if one of these guys say dou h 1 by dou x 1 actually goes to 0. What does that mean? h 1 is not sensitive to variations in x 1 right fine. But was our original mandate, what did we want these neurons to capture? We wanted the neurons to capture these important characteristics right.

So, if x 1 changes we want h 1 to change, do you get that? How many of you get that? We wanted the neurons to capture the important characteristics of the data right. But now, we have added a contradictory condition which says that we do not want the neuron to capture a variations in the data, do you see this? So, what is happening here? L theta says that I should be able to capture these variations right, otherwise I will not be able to reconstruct.

If all my h i's are not sensitive to variances x 1; that means, I give it any x 1 it will produces the same h i, is that clear is that with everyone right. That means, so see this is this. So, I have these training examples occurs all these training examples my bold x, which is vector x is going to change. That means, x is which are the elements of this vectors are going to change.

Now, what this condition is saying is that if I change x I, I do not want the h l's to change, I do not want the values of the hidden representations to change. So;, that means, it is changing the respective of what is the input fed to it try to produce the same output, do you get this argument? Ok. That means, it is not capturing any important characteristics of the data, is that fine is that valid argument, but that is not what we wanted, we wanted it to capture the important characteristic of the data. So, what are we trying to do now? Ok.

(Refer Slide Time: 05:01)



So, just I, it is hard for me to do evaluate what you have said, but just pay attention and see if that is correct you can judge it on your own, right. So, that is the actually the idea right we have put these two contradictory conditions with each other right, l theta says capture the important variances of the data. Omega theta says do not capture variations in the data, watch the tradeoff capture only very important variations in the data do not capture the variations which are not important. Can you relate this to something that you have seen before?

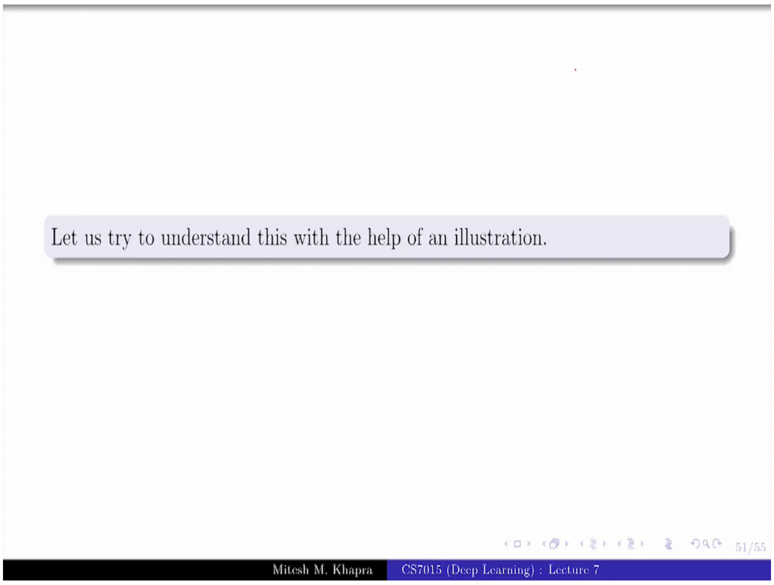
Student: Bias variance.

No, the other answer there are only two answers bias variance and PCA when I say the other answer.

Student: Pca.

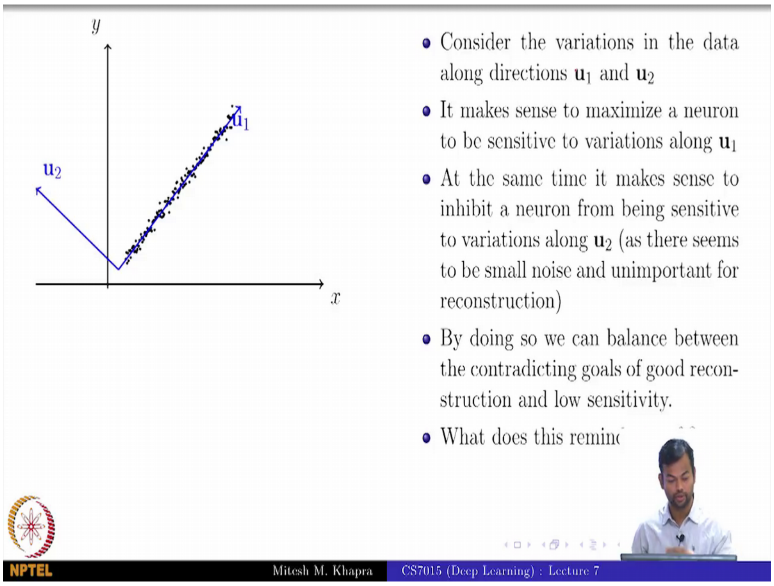
What am I trying to force it to do capture only the important variation, it is if it is not clear right now, we will come back to this ok.

(Refer Slide Time: 05:52)



So, let us try to understand with this with the help of an illustration right, how many of you get the argument which I made on this slide ok, most of all.

(Refer Slide Time: 06:03)



Now, this is the situation, I have u 1 and u 2 as my dimensions fine, which of this is important? u 1, the variations in the data across u 1 is something that I should care about. Because I can see that brings in some difference what about the variations in u 2.

Student: Not important.

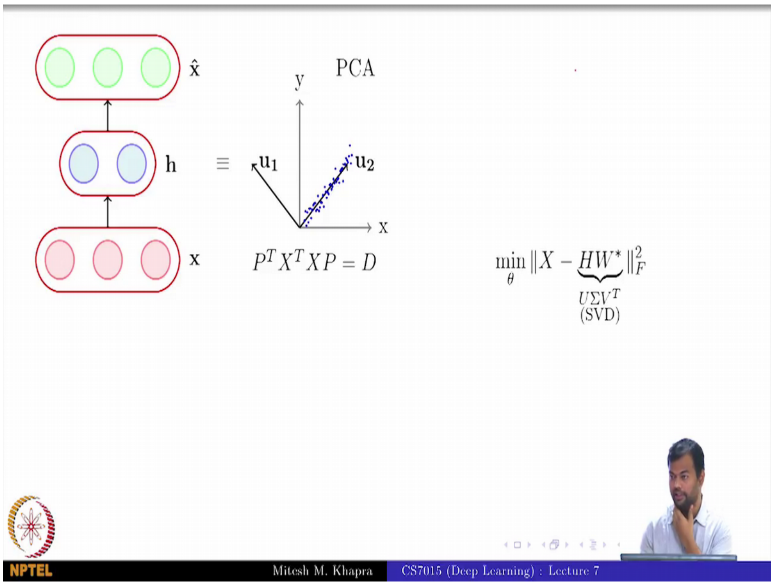
Not important, they seem like noises because these variations are there they are not all lying on the central line, they are slightly away from the line, here are some variations. But should I go out of my way to capture these variations, does it make sense to do that? No right. So, it makes sense to maximize a neuron to be sensitive to variations along u 1.

But it does not make sense to make the neuron sensitive to variations along this other dimension which is u 2 ok, by doing so we can balance the two conditions. So, one condition was trying to capture all the important variations do this, but do it only for the dimensions which really matter. The other conditions says that do not capture important variations do this, but do it only for those dimensions which do not matter. What is this remind you of? At least the diagram should have it away right.

Student: (Refer Time: 07:17)

It is same as principle component analysis right, so that is exactly what you try to do in PCA, you try to capture the variations across the important dimensions, but not across the non important dimensions. How many of you get the concept of contractive order encoders? Ok good. So, I think that is a where we will end lecture 7.

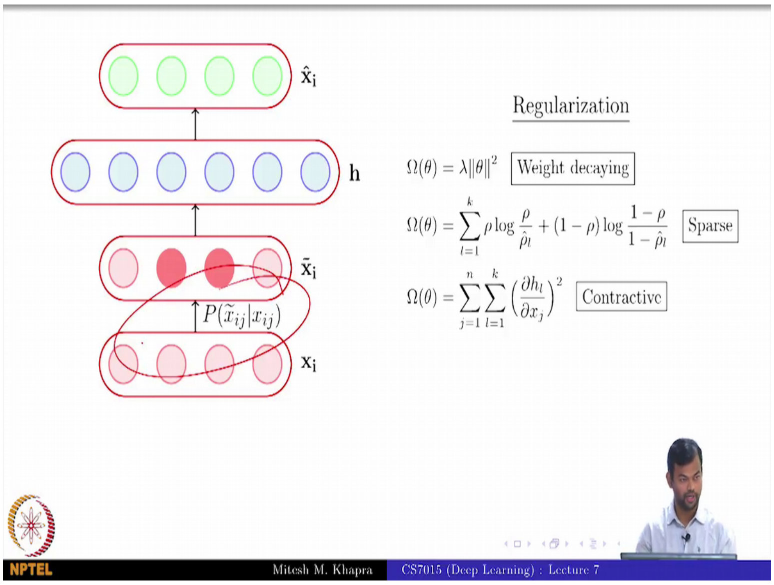
(Refer Slide Time: 07:38)



And just a quick summary; so, we showed that under certain conditions autoencoders are equivalent to PCA.

And we use this result very crucially there, that SVD theorem I will not state it.

(Refer Slide Time: 07:51)



And then we looked at different types of regularizations for auto encoders where we looked at weight decaying. That means, the standard l 2 norm, we looked at the sparse auto encoder, the contractive auto encoder and we also looked at these denoising auto encoders right. So, that is the summary of this lecture.