Deep Learning for Visual Computing

Prof. Debdoot Sheet
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 18
Sparse and Denoising Autoencoder

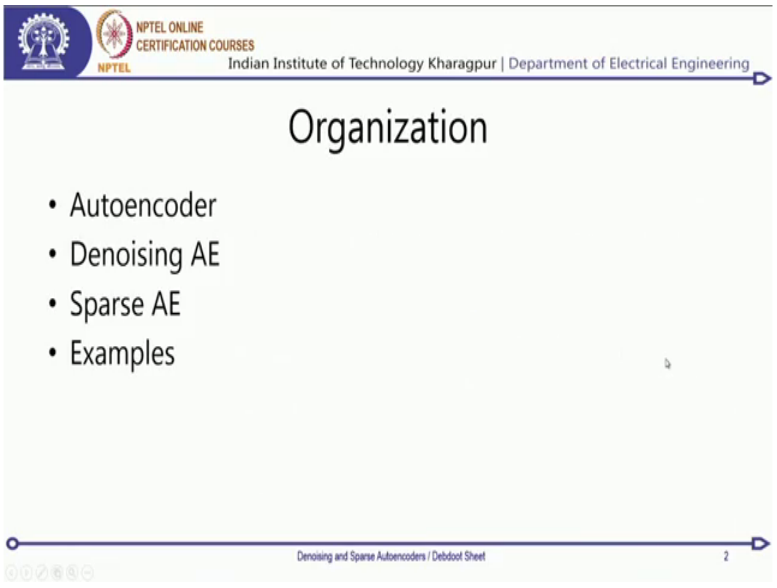
So, welcome and so, in the last class and the lab you had seen how to use down stacked autoencoders and the idea was that if we have a large network and in order to get around with the complexities with training down the weights and then the total complexity associated with the dynamics of the free space of weights, then can we have a simple way of doing it out. So, following that is something which is called as denoising and sparse autoencoders and where this comes down to play is that denoising auto encoder as the name suggests is that it can have a major use in terms of denoising or removing out noise from matches or examples over there and smart autoencoders are where you would be making use of sparsity theory.

Now, these two are very important in two different aspects; one is, in order to prevent something which is called as memorizing of examples by a network or an over fitting scenario in which what typically happens is the network would tend to learn examples and memorize them instead of trying to learn down patterns it would be memorizing the examples itself, so within an autoencoder since you your basic job is basically whatever is the input you given you are supposed to predict down the same as the output over there.

Now, you would try to make the network not memorize these things, but rather to actually represent it in terms of features and that is the basic idea of learning even for human learning I mean one way is that a lot of people just memorize and then they go and just write down in an exam and then they forget it out, but that is not what is the essence of even human education or human learning, societal learning is where you would try to grab an essence and then develop your intellectual potential which is your own neurons in your brain they are supposed to fire it down appropriately for anything unseen, it should not be that whatever you have seen in the book anything which comes down in your across your life which is very different from that that you will not be able to solve it out.

So, that is that is exactly the same way which we try to do with these kind of deep neural networks as well and that is where Denoising and sparsity within Auto encoders does come to play. So, without reading much let us get into what I would be doing.

(Refer Slide Time: 02:32)


NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Organization

- Autoencoder
- Denoising AE
- Sparse AE
- Examples

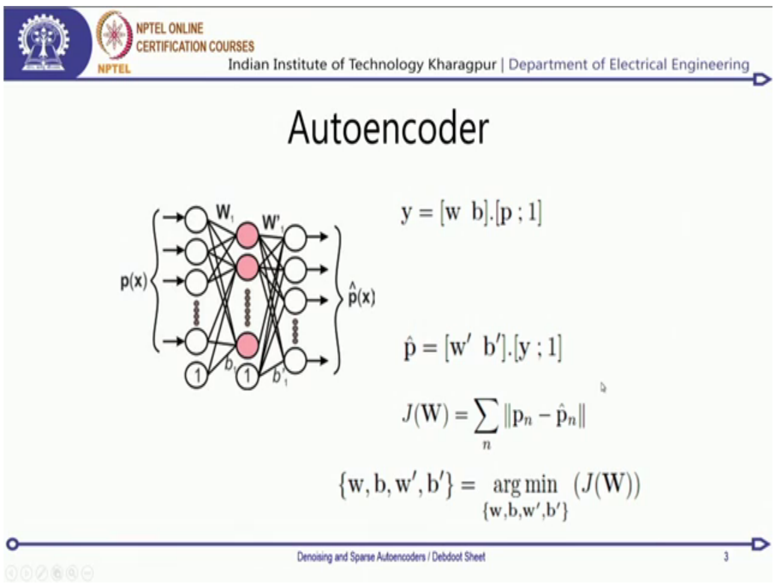
Denoising and Sparse Autoencoders / Debdoot Sheet 2

So, typically what it is arranged as I would be revising that first basic math of Autoencoder once again and so the reason that I am going through them sort of a lot of you will be getting down that I am just showing on the same set of equations and a lot of those slides which look like repeaters, but the main reasoning is; so that you can actually learn to relate this salient changes between what was earlier and what we are doing now. So, this what happens from an Autoencoder when we go into a denoising Autoencoder and what is the change which happens from going from a denoising Autoencoder to a sparse Autoencoder and are there some ways of combining these denoising and sparse factors together and also you need to keep in mind that actually all of these examples can be extended down to your stacked Autoencoder.

So, it is not something like if you are stacking it down then you cannot use these kinds of different strategies because each is a different way of doing it, so stacking is a way of geometrically building up the network where by using lesser and lesser number of resources in order to create a much deeper and complicated network and denoising Autoencoder is a way of basically training down the network and getting a way in which you stop it from memorizing examples sparsity in autoencoders is trying to make down redundancies in weights and imposed on some sort of redundancy and whatever it is learning. So, each has a different objective and all of them can be combined very much together.

So, at the end we will be looking into examples of stacks pass denoising autoencoders basically taken down all of them together and what it goes down. So, in the subsequent tutorials using coding which we will be doing we will be also touching upon all of these examples. So, you will be seeing a repetition of what you had solved with autoencoders and then brought down the stacking concept over there then also going over denoising and sparsity criteria to find out whether that does work out in terms of your examples as well and an incrementally we will also be seeing down what we gain by bringing all of these concepts as well.

(Refer Slide Time: 04:33)



NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Autoencoder

$$y = [w\ \ b].[p\ ;1]$$

$$\hat{p} = [w'\ \ b'].[y\ ;1]$$

$$J(W) = \sum_n \|p_n - \hat{p}_n\|$$

$$\{w, b, w', b'\} = \underset{\{w,b,w',b'\}}{\arg\min}\ (J(W))$$

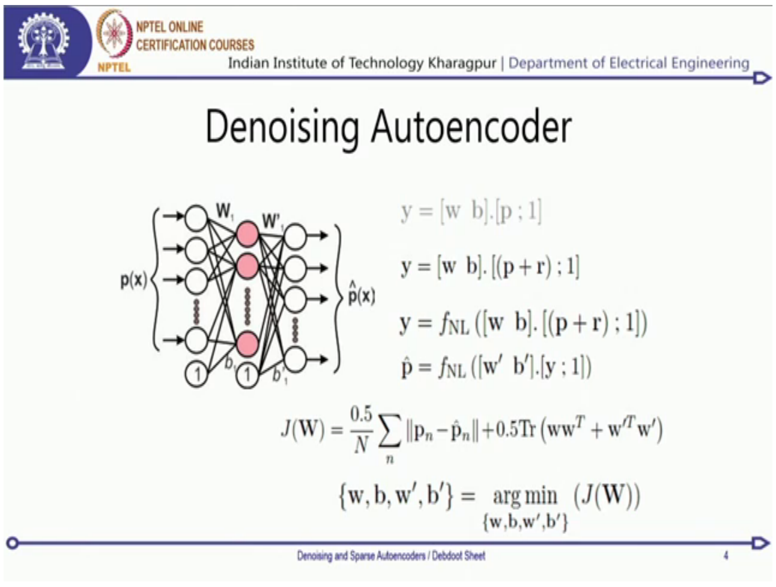
Denoising and Sparse Autoencoders / Debdoot Sheet 3

So, I will be getting started with a very simple auto encoder. So, what you had was that say at a particular location x which is x comma y location within an image and you take a small patch around that location and that is what is called as p of x now p of x is going to be a set of pixels; which get arranged linearized and arranged over here, so for our 5 cross5 patch from gray scale image you have 25 pixels, for a 5 cross 5 patch from a colour image and RGB colour image you would be having 75 pixels over here, you have your bias connected over here and then so in some of my examples you might not be seeing this node as one and then this bias connected, but always keep in mind that this is an intrinsic factor, so if I have just a bus kind of one single arrow which connects some input to my hidden layer always remember that it is nothing that it is a 0 bias network it is a perfectly valid bias network. So, you have a bias connecting the these like that there is always a bias in the connection to the input of the hidden layer as well and similarly it goes from the hidden layer to the output over here as in this case. So, this is just one single zone of an Autoencoder or as you had looked down in your stack autoencoder you would remember that it is just with using one hidden layer, so you had W1 and W1 dash which you were training them and then you could do something else. So, we will stick down to only this hidden layer we are not looking into classification problem as such currently.

So, a simple auto encoder is what can be defined mathematically of this form that the output of say this hidden layer all the neurons over here is what is the tensor y and that is basically, so this is not output of exactly this1 that there should be a nonlinearity as well, but if we consider that nonlinearities 1thenif unit transfer nonlinearity then you can pretty much have it represented in this one. Now, the result of this next layer is what will be coming down into this 1 and in terms of w dashed and b dashed? Now, and your basic learning rule is where you would like to have this particular cost function which is the difference or the Euclidean norm between your p and p hat for n^{th} number of samples and summed over all the n number of sample these needs to be minimized and that is your basic learning rule which are going to use within simple auto encoder.

Now, standing on top of that if I would like to go down with Denoising Autoencoder then let us see what happens over there? So with a denoising Autoencoder the structure as such remains the same you do not make any change within the structure, so that is what I was saying; that, if you would like to do a denoising Autoencoder on top of a stacked auto encoder so stacking is just a way of geometrically building up the network denoising has nothing to do and a 0 relationship with any of those aspects of stacking down, now for a stack or for Denoising Autoencoder what happens is, for a simple Autoencoder while you will have this particular kind of a form, for a denoising Autoencoder the difference is that you bring in this extra tensor over here which is called as r and this is a nosier or a noisy parameter.

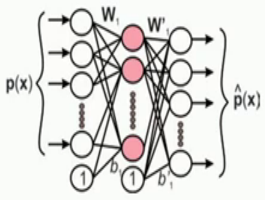
(Refer Slide Time: 07:06)



NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Denoising Autoencoder



$$y = [w\ \ b].[p\,;1]$$

$$y = [w\ \ b].[(p+r)\,;1]$$

$$y = f_{NL}([w\ \ b].[(p+r)\,;1])$$

$$\hat{p} = f_{NL}([w'\ \ b'].[y\,;1])$$

$$J(W) = \frac{0.5}{N}\sum_n \|p_n - \hat{p}_n\| + 0.5\,\mathrm{Tr}\left(ww^T + w'^T w'\right)$$

$$\{w, b, w', b'\} = \underset{\{w,b,w',b'\}}{\arg\min}\ (J(W))$$

Denoising and Sparse Autoencoders / Debdoot Sheet 4

So, what we do typically is? That I would try to add down a noisy parameter along with my input patch over there and this noise is what is randomly generated it is a stochastic generated noise so within your learning framework within what you do is? Typically you have a set of learning examples and then on within every book you iterate over all of these set of learning examples over there. Now, whatever is the value of p for a given sample number over there it is going to be the same across all the epochs, so there is no going to be no change over there, but, now if I bring in a random nosier over here which means this r is going to be a random number which is generated every time you are trying to access some patch over there.

So, for the first epoch whatever is the value of r for a given sample; in the second epoch the value of r is going to be different and according to that this p plus r this combination is also going to be a different value which comes and that what it would mean is that around the same sample you are going to draw a certain amount of slight perturbations and these perturbations are what will be to the network what it stops the network from doing is it will stop it from memorizing it because you are not going to see the same written way of example every time, but you are going to get done some different way in which it keeps on coming down.

Now, if you have a nonlinearity after that then this nonlinearity adds and this is the typical form and on the output side what you would try to do is you will get down a p hat, but here the output of this y or the first hidden neuron that does not get any noise added to it, the noise is just added on the input side of it and then when I am trying to look into the output over there my cost function will include just trying to find out the l 2 norm between p and p hat, so here I have no more going to put down p plus r. So, what that necessarily means is; that I cannot keep my training data already pre corrupted on to my data set.

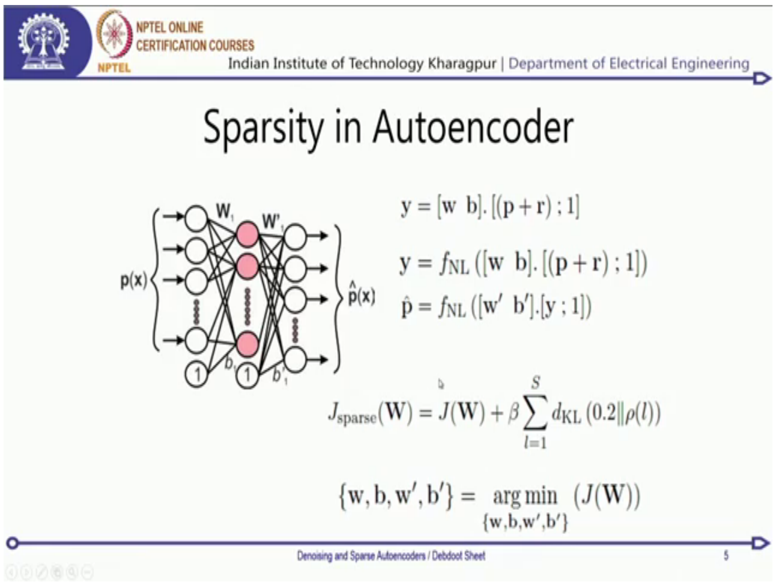
So, every time I am trying to access a particular sample from the training data I will have to introduce my corruption of the noise this random value exactly at that point of time I cannot keep it pre corrupted at any point of time, now this is one part of my cost function which is p and which is made out of p and p hat. Along with that, I also have another part of my cost function so if you look into it all of these are equally weighted so this is 0.5 weight and this is 0.5 weight which means that there are two parts of a cost function and i am weighing two parts of a cost function equal hopefully.

Now, this other part of it which is just an orthogonalization factor over here and which is typically used for the factor so that I have maximum amount of dissimilarity between neighbouring between the kernels which I am learning, so what w transpose plus and w dashed transpose w would do is; that this would return me a square matrix over here which is the same size as that of my number of neurons over there in my hidden layer and then I try to try to take out a trace. Now, my objective is that I would try to minimize this whole cost function so that would mean that I need to minimize this part of my cost function as well as this part of my cost function.

Now, if I want to minimize this cost part of my cost function so that would mean that my trace or the diagonal sum over there has to also be minimized and that means, that very strong correlations at any point of time between kernels from one side to the other side that has to be reduced or what typically would also mean is; that there has to be minimum correlation between w and w dashed say one cannot be the inverse of the other, so this will be maximum only when w and w dashed are inverse of each other that says what comes down from the mathematics over there.

So, this together as a cost function does restrain me from memorizing samples and then my total learning rule is what goes down according to this kind of a form that I would try to minimize my cost function over all of these values of w and b’s together taken. Now, this is what qualifies as a typical case of straightforward denoising Autoencoder.

(Refer Slide Time: 12:00)


p(x)
W_1
W'_1
b_1
b'_1
$\hat{p}(x)$

NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Sparsity in Autoencoder

$$y = [w\ \ b].[(p+r)\,;1]$$

$$y = f_{NL}([w\ \ b].[(p+r)\,;1])$$

$$\hat{p} = f_{NL}([w'\ \ b'].[y\,;1])$$

$$J_{\text{sparse}}(W) = J(W) + \beta \sum_{l=1}^{S} d_{KL}(0.2 \| \rho(l))$$

$$\{w, b, w', b'\} = \underset{\{w,b,w',b'\}}{\arg\min}\ (J(W))$$

Denoising and Sparse Autoencoders / Debdoot Sheet 5

Now, we did stay say that we would be touching upon another which is also called sparsity, now for spicy tea what you do is? Let us say I have a simple Autoencoder which can be modelled in this form and if I put down a nonlinearity then I have this extra nonlinearities coming down, now in sparsity i do not so here I am assuming that this is denoiser which is already added on to it so this is a noisy vector which is added and then I am trying to predict out p hat says that this network can learn to be Denoising, but then I would also want that; so this JW over here is what comes down from my previous case of J W so in my earlier case on my denoising autoencoders whatever is my cost function that is known as J W so I would be using this part of J W and I would bring it over here in case of my sparsity as well, so this J W part is written plus I put down on another factor beta times of this scale divergence of rho l, now this rho l is basically my sparsity coefficient or this is the probability of getting anon 0 value in this weights over weight matrices over there.

So, if I have W1and W1dashed over there. So, what typically means is? That all of these are some sort of tensors or matrices over there a lot of those companies 0 and some of them can be non0 so my probability of getting a non-sparse matrix over that is basically the number of elements of this matrix which are non 0 divided by the total number of elements on this matrix.

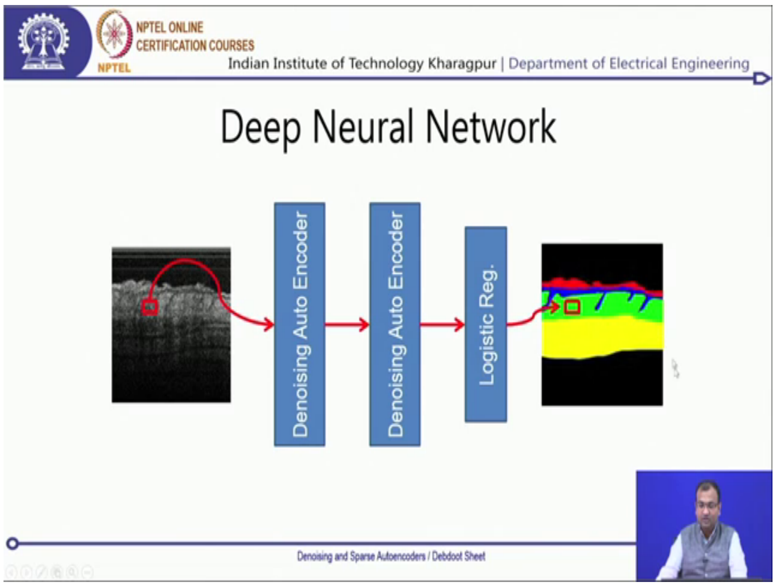
Now if I have a target that I want to have 20 percent of it as my target of non 0 values and everything else has a 0 value then this scale divergence will be 1 when these two probabilities are almost the same, now if this is likes way ahead of it. So, it might be a very non sparse or a highly sparse, but in both the cases the distance will be large and it will try to impose it to coming down to a particular sparsity level itself and as in over here it is a 20 percent sparsity. Now what I do is? That I add this 1 on to my earlier cost function and then create a new cost function which is called as J sparse and then my learning algorithm is basically to minimize this J sparse over here through this learning rule and that is what will impose a sparsity.

So, while in the earlier case we did look into that so that the inversion problem does not, so that the memorization does not happen within the network we would try to keep w and w dashed not as invert of each other and that can be achieved by your trace summation over the trace part of it and then trying to minimize that a total matrix sum, now the other part of it is which is where I would try to have most of these weights as 0'sand the advantage which you get down by putting most of these weights as 0's is that the dot product or multiplication product is going to be 0 and as the number of values which you get through it are going to be very low valued not very high values which will just saturate out your complete network.

The good thing on top of it is what we had done in the first class on autoencoders where I was selling showing you a few of the classical example. So, in one case where the image was looking very diverse and you had a lot of noisy and randomness features present over there we did learn out and that was an OCT image we did learn out that it is not necessary that a very noisy and randomly appearing image will have a lot many number of unique kernels. So, you did find out less number of unique kernels and then in the second layer there was a huge amount of sparsity.

So, we will be sure seeing revisiting that example once again over here as well and now on the other side of it you did see that when you were looking at retinal images then over there although the image appeared thin, but the amount of diversity was very high and the system was not at all sparse, so this sort of a learning algorithm sparsity actually imposes that and there are many more theoretical implications which we will seeing down in the later on lectures as well.

(Refer Slide Time: 15:52)


NPTEL ONLINE CERTIFICATION COURSES

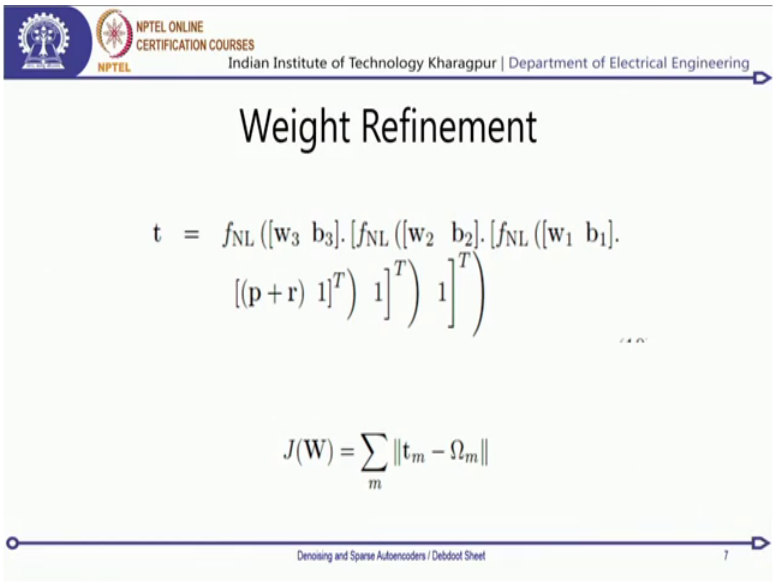
Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Deep Neural Network

Denoising and Sparse Autoencoders / Debdoot Sheet

So, let us start like just get a revision concept of that one so the earlier thing which we had done with a standard MLP problem of a pixel classification was where I take down a patch over here then train down to water encoders and here it is a denoising Autoencoder and now that if I train it down by stacking one behind the other so you can very much call this as a stacked denoising Autoencoder. So, your denoising criteria is applied and now if I apply down the sparsity criteria as well then I can call this as a stacked Denoising sparse Autoencoder and accordingly I take down a patch over here and then I keep on moving it pixel by pixel and that will give me a raster movement and based on that raster movement I would be getting done each pixel classification which is collated to form this matrix of my classified tissue map coming down over here.

(Refer Slide Time: 16:43)


NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Weight Refinement

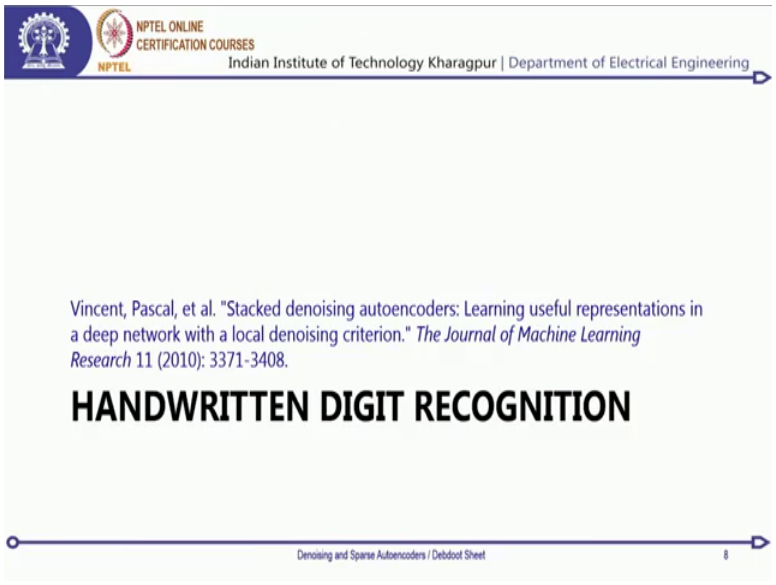
$$t = f_{NL}\left([w_3 \;\; b_3].\left[f_{NL}\left([w_2 \;\; b_2].\left[f_{NL}\left([w_1 \;\; b_1].\; [(p+r) \;\; 1]^T\right) \;\; 1\right]^T\right) \;\; 1\right]^T\right)$$

$$J(W) = \sum_m \|t_m - \Omega_m\|$$

Denoising and Sparse Autoencoders / Debdoot Sheet 7

Now, where it goes down is that, my noisy factor is what is added down to my input over here and even in my final training also I would be having that noisy part added and then the final cost function after the auto encoder training is over is something which will no more make use of other factors, but it will just be making use of this factor of l 2 norm between my predicted class which is my tm and my actual class in my ground truth which is omega m. So, that was a pretty straightforward thing which we had done in the earlier one, but here we are just introducing the concept of sparsity Denoising as well as the stacking way of learning it done with lesser number of weights.

(Refer Slide Time: 17:24)


NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

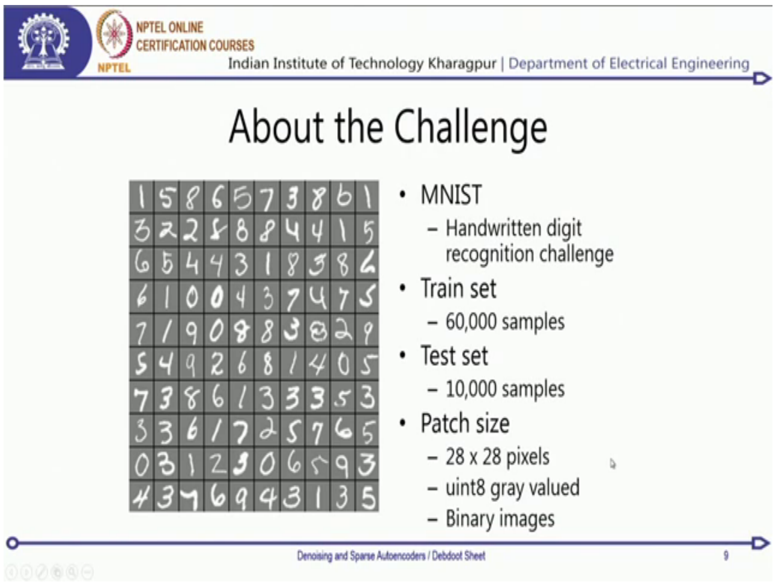
Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *The Journal of Machine Learning Research* 11 (2010): 3371-3408.

HANDWRITTEN DIGIT RECOGNITION

Denoising and Sparse Autoencoders / Debdoot Sheet 8

Now, so let us get into this revision of that written digit recognition so remember that in the last class we did say about this particular problem so this was about trying to recognize hand written digits and that is a 10 class classification so you need to understand which numeral is it from 0 to 9 and so you basically have 28 cross 28 pixel sized patches over here and everything is integer valued 8 bit numeral though if they are binary images which means that the gray area over here the background is 0 and whatever is white is has a value of 255 and or training you have 60000 samples for testing you have 10000 samples a very classical example.

(Refer Slide Time: 17:31).



NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

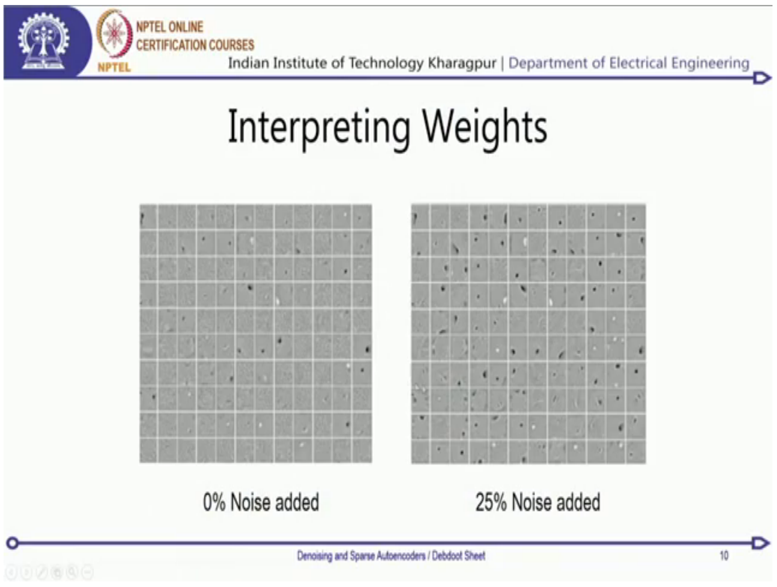
About the Challenge

- MNIST
  - Handwritten digit recognition challenge
- Train set
  - 60,000 samples
- Test set
  - 10,000 samples
- Patch size
  - 28 x 28 pixels
  - uint8 gray valued
  - Binary images

Denoising and Sparse Autoencoders / Debdoot Sheet 9

Now, if i try to train this network without adding any kind of a noise then on my first layer if I try to visualize these colours and this is what it looks like. So, let us count down how many do we have 1,2, 3, 4, 5, 6, 7, 8, 9, 10, 11,12 okay and 1,2, 3, 4, 5, 6, 7, 8, 9, 10 so it means that there are 120 hidden neurons over there and each of these is 28 cross 28 which we have. Now, the these kernels which you see these are basically convolution kernels which has it has done and these are my first level of feature extractors, but now a lot of these are very bland looking and very near to 0 and as if like they do not extract any particular feature now that is a risk over there.

(Refer Slide Time: 18:07)


NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Interpreting Weights

0% Noise added

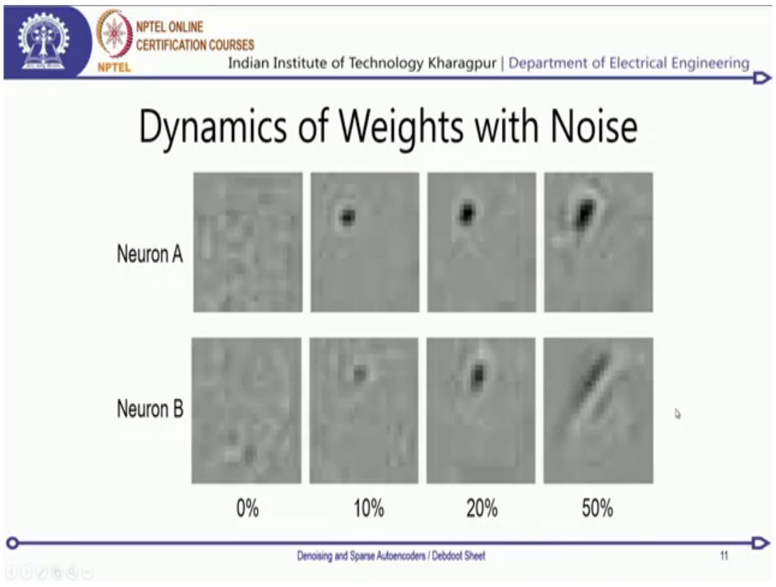
25% Noise added

Denoising and Sparse Autoencoders / Debdoot Sheet 10

Now, what happened is? That if we started adding noise which was at those perturbations to it and say this noise is 25 percent of a noise power which means that the maximum amplitude of the noise or the energy of the noise of 25 percent of the energy of the original signal, so you can control down that factor over there. Now once we have 25 percent noise added over there you see that it has started to learned down much better representation of features and then these bland appearing or no feature extracting once also went out and then you can see down exact features coming down.

So, while this one is retained this particular patch is what corresponds to this patch and it has retained on these three patches, while it was not learning anything it did end up learning new kind of a representation over there and that is something which is interesting to see that newer kind of representations get learned when you add down noise over there and to go down to make even a statement that noise is actually good noise, now which is good in terms of making it more robust to learn down newer kind of features which it would not have learnt otherwise you cannot guarantee that you will be putting on all sorts of samples which are ever existing in the world you are training it as ait is always limited, but then in order to generalize it will have to try to figure out water features which will describe an unseen unforeseen object of the same class or category as well.

(Refer Slide Time: 20:17)


NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Dynamics of Weights with Noise

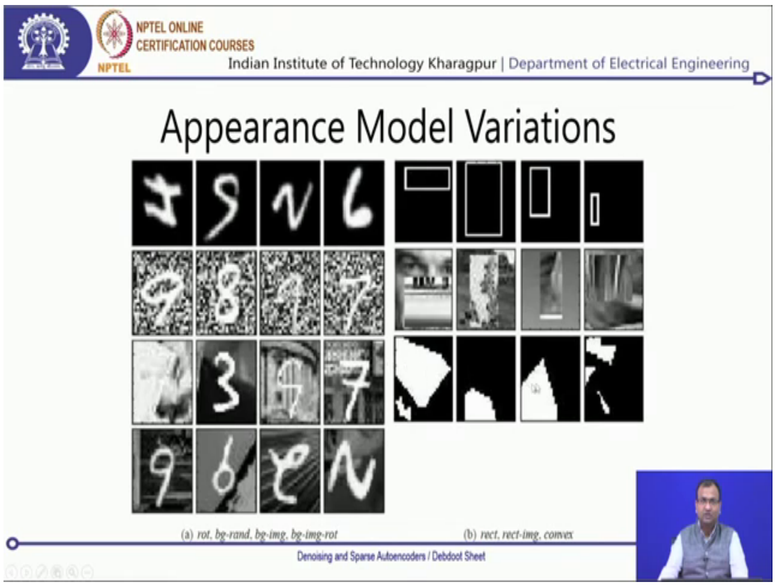
Denoising and Sparse Autoencoders / Debdoot Sheet 11

So, you see that adding a noise over here does worked out that magic and then we also try to look into dynamics of weights with noise which is if more and more amount of noise is added then does it go on good . So, over here we consider two different neurons; Neuron A and B which appear as if almost similarly noisy not basically capturing anything, so these values are very close to 0 when you add down a 0 percent noise.

Now, when you add a 10 percent noise they start to have some sort of an appearance over there so you would see this dark blobs coming down, but at two different regions and they appear still to be like as if learning something similar sort of a nature, at 20 percent noise they also appear to be learning similar sort of a nature and this dark lump has grown bigger, so this is where it more of mimics and inverted Gaussian kind of or more of like there is a 0 value there is some sort of a positive value and then there is a very high negative value is what it comes from.

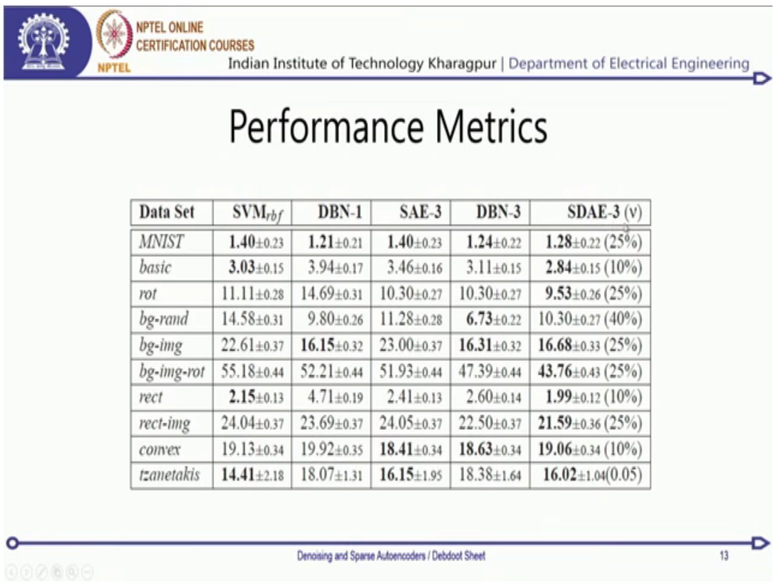
So, it is more of like a Laplacian of Gaussian which is created over there so it is a LOG equivalent kind of thing which it is learning, but then the moment you see that you add 50 percent noise while the top neuron retained it is own Laplacian of Gaussian like behaviour the bottom neuron started to mimic something like a Gabor wavelet and this is quite interesting, because you see that by adding them noise while it is trying to generalize over a larger number of unseen examples in order to generalize it did end up learning newer kind of features and that is a really interesting fact that it will not memorize, but actually learn to be even better from now on.

(Refer Slide Time: 21:53)


NPTEL ONLINE CERTIFICATION COURSES
Indian Institute of Technology Kharagpur | Department of Electrical Engineering
Appearance Model Variations
(a) rot, bg-rand, bg-img, bg-img-rot
(b) rect, rect-img, convex
Denoising and Sparse Autoencoders / Debdoot Sheet

Now, from there let us get into the other kind of an example where they try to like really perturb the data in a different form of way. So, these examples are basically ways in which you can change down the nature of the data and then make your training more and more robust.

(Refer Slide Time: 22:05)



NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

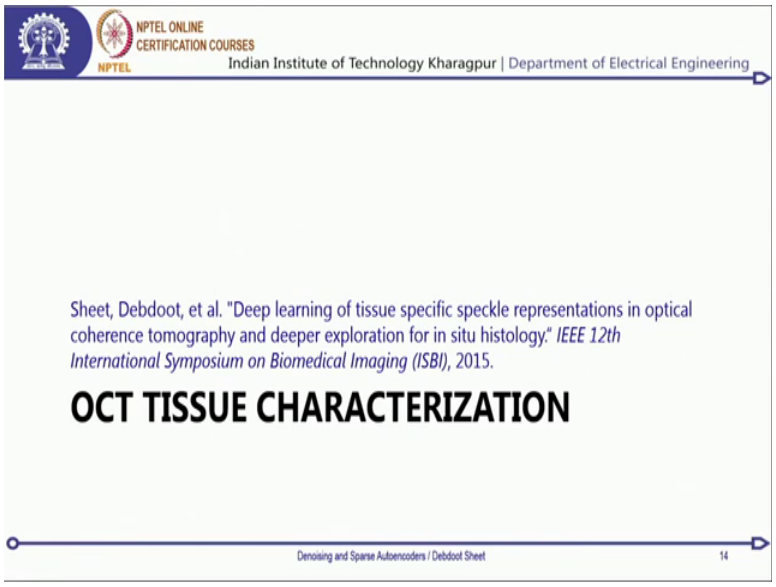
Performance Metrics

Data Set	SVM$_{rbf}$	DBN-1	SAE-3	DBN-3	SDAE-3 (ν)
MNIST	**1.40**±0.23	**1.21**±0.21	**1.40**±0.23	**1.24**±0.22	**1.28**±0.22 (25%)
basic	**3.03**±0.15	3.94±0.17	3.46±0.16	3.11±0.15	**2.84**±0.15 (10%)
rot	11.11±0.28	14.69±0.31	10.30±0.27	10.30±0.27	**9.53**±0.26 (25%)
bg-rand	14.58±0.31	9.80±0.26	11.28±0.28	**6.73**±0.22	10.30±0.27 (40%)
bg-img	22.61±0.37	**16.15**±0.32	23.00±0.37	**16.31**±0.32	**16.68**±0.33 (25%)
bg-img-rot	55.18±0.44	52.21±0.44	51.93±0.44	47.39±0.44	**43.76**±0.43 (25%)
rect	**2.15**±0.13	4.71±0.19	2.41±0.13	2.60±0.14	**1.99**±0.12 (10%)
rect-img	24.04±0.37	23.69±0.37	24.05±0.37	22.50±0.37	**21.59**±0.36 (25%)
convex	19.13±0.34	19.92±0.35	**18.41**±0.34	**18.63**±0.34	**19.06**±0.34 (10%)
tzanetakis	**14.41**±2.18	18.07±1.31	**16.15**±1.95	18.38±1.64	**16.02**±1.04(0.05)

Denoising and Sparse Autoencoders / Debdoot Sheet 13

And so what was found out is that? Stacked Denoising Autoencoder and SDA stacked sparse denoising Autoencoder and SDAE of with 3 hidden layers over there was sort of outperforming on most of these kind of rotated versions or unforeseen versions on which it was training down for digit recognition then any of the other classical ways including the belief networks or even support vector machines.

(Refer Slide Time: 22:37)



NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

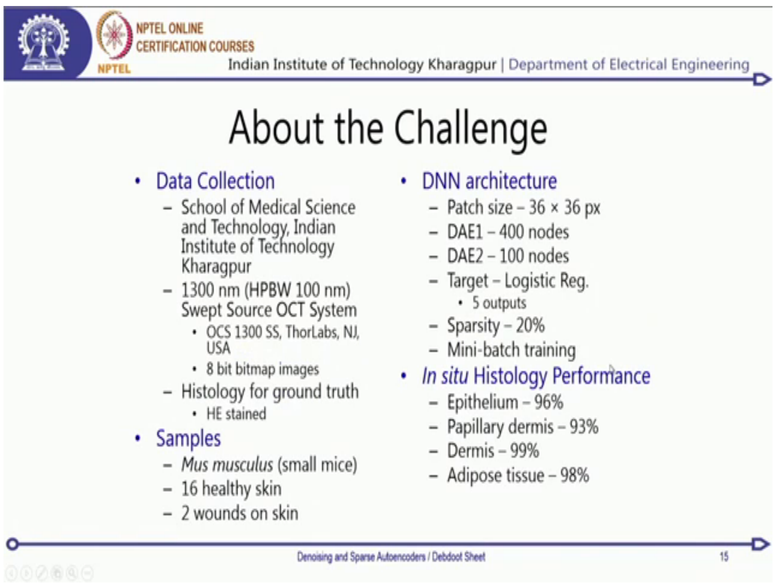
Sheet, Debdoot, et al. "Deep learning of tissue specific speckle representations in optical coherence tomography and deeper exploration for in situ histology." *IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015.

OCT TISSUE CHARACTERIZATION

Denoising and Sparse Autoencoders / Debdoot Sheet 14

So, that is quite an interesting observation to make from these kind of very simple learners.

(Refer Slide Time: 22:43)



NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

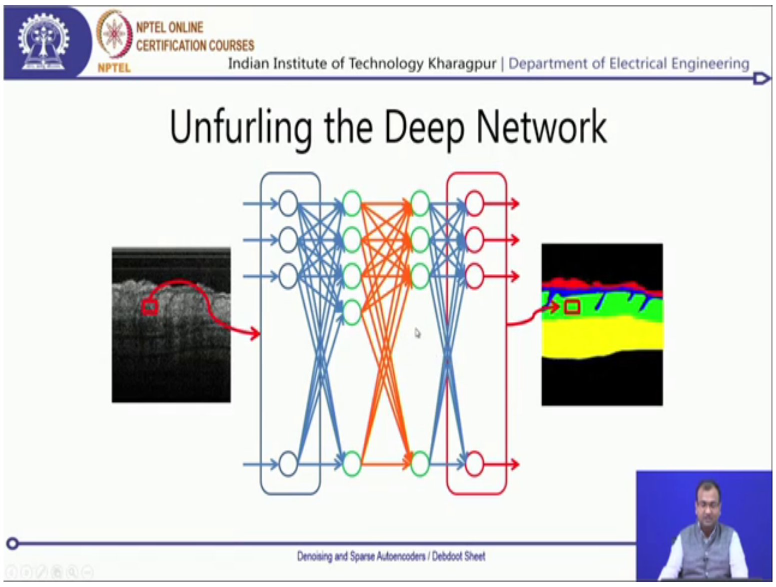
About the Challenge

- Data Collection
  - School of Medical Science and Technology, Indian Institute of Technology Kharagpur
  - 1300 nm (HPBW 100 nm) Swept Source OCT System
    - OCS 1300 SS, ThorLabs, NJ, USA
    - 8 bit bitmap images
  - Histology for ground truth
    - HE stained
- Samples
  - *Mus musculus* (small mice)
  - 16 healthy skin
  - 2 wounds on skin
- DNN architecture
  - Patch size – 36 × 36 px
  - DAE1 – 400 nodes
  - DAE2 – 100 nodes
  - Target – Logistic Reg.
    - 5 outputs
  - Sparsity – 20%
  - Mini-batch training
- *In situ* Histology Performance
  - Epithelium – 96%
  - Papillary dermis – 93%
  - Dermis – 99%
  - Adipose tissue – 98%

Denoising and Sparse Autoencoders / Debdoot Sheet 15

Now, the next one which I would discuss is on OCT tissue characterization from our earlier work and so I would not be going down any further through the details on this dataset because this is what we had done in the earlier week as well.

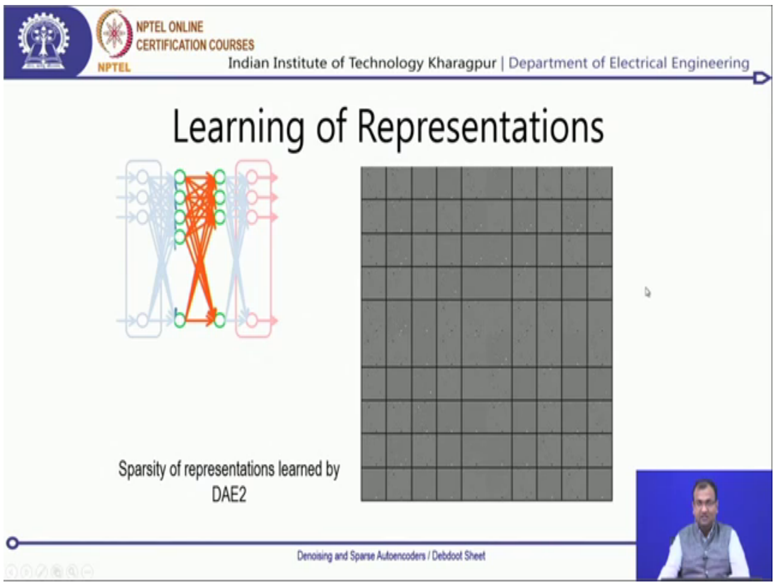
(Refer Slide Time: 22:51)


NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Indian Institute of Technology Kharagpur | Department of Electrical Engineering
Denoising and Sparse Autoencoders / Debdoot Sheet

Unfurling the Deep Network

So, here if we look into so this was the sort of network which I was dealing with so these was just two hidden layers in the network and this is what was predicted out and then my objective was to relook into what it had learned.

(Refer Slide Time: 23:03)

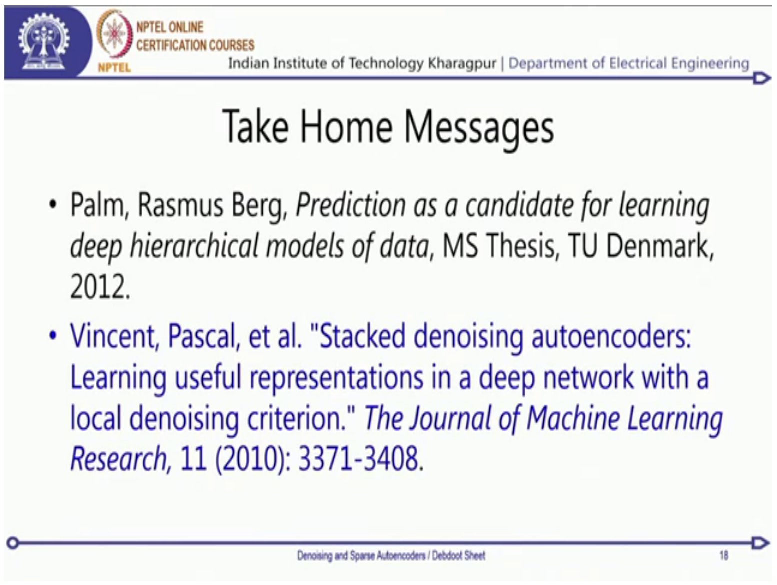

NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Indian Institute of Technology Kharagpur | Department of Electrical Engineering
Denoising and Sparse Autoencoders / Debdoot Sheet

Learning of Representations

Sparsity of representations learned by DAE2

So, you see you remember that this is what I was showing you out as it was learning in the first hidden layer and a lot of these weights were similar and that should have an impact in the second, so the connection between the first hidden layer to the second hidden layer in terms of it creating a lot of zeros and this matrix this sort of a weight matrix so this is one weight matrix which connects known my neurons with my first hidden layer to the neurons in my second hidden layer and this is practically a sparse matrix because very few pixels except for5, 6 pixels over there everything else is perfectly 0 value and that brings it very close to it being an exactly very typical representation of a sparse Autoencoder.

(Refer Slide Time: 23:51)


NPTEL ONLINE CERTIFICATION COURSES
Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Take Home Messages

- Palm, Rasmus Berg, *Prediction as a candidate for learning deep hierarchical models of data*, MS Thesis, TU Denmark, 2012.
- Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *The Journal of Machine Learning Research*, 11 (2010): 3371-3408.

Denoising and Sparse Autoencoders / Debdoot Sheet 18

So, on the take home messages over here what I have is that; I would re suggest you to get back into this thesis from Rasmus book, Palm and which has much more details if you would like to get into understanding what happens on other kinds of cost functions and other types of sparsities or any other more improvisations which you can do and what changes you need to bring over there? As well as Pascal Vincent's the MLR paper is very good treatise on understanding sparse Denoising Autoencoders. So, this is what brings me an end to our understanding with very basic Autoencoders and while we are still taking down the seeking down to one single sort of a learning algorithm at itself.

So, in the next week we will be looking into different kind of cost functions and different kind of learning algorithms and initialization methods and what they, how they impact down your process and the rest of the classes we will be doing a bit more of coding based tutorials in order to get you more closer to understanding practical concepts and how to train down these networks from scratch and create down your own auto encoder networks in stack mode modes in which sparsity and with denoising criteria.

So, with that, stay tuned and we will get back with more details in the subsequent classes.

Thank you.