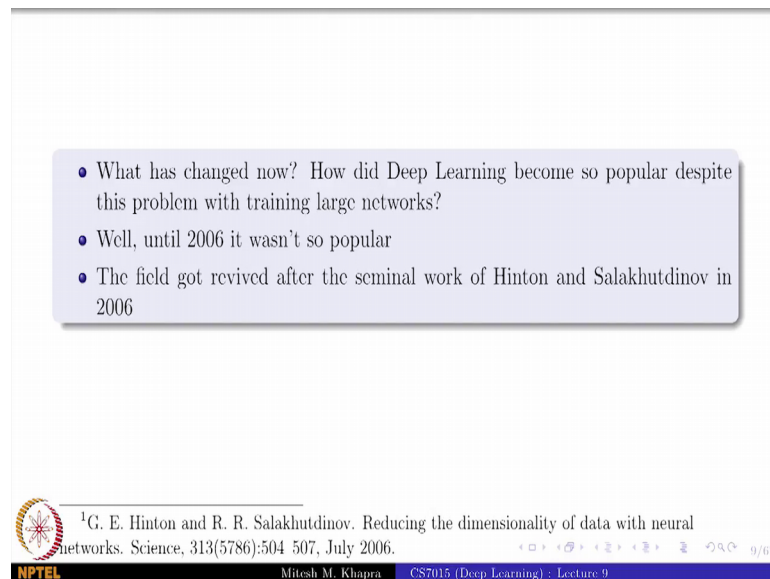


Deep Learning
Prof. Mithesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 9.2
Lecture – 09
Unsupervised pre-training

So, with that we go on to the next module, in which we will talk about Unsupervised pre training.


(Refer Slide Time: 00:20)



• What has changed now? How did Deep Learning become so popular despite this problem with training large networks?

• Well, until 2006 it wasn't so popular

• The field got revived after the seminal work of Hinton and Salakhutdinov in 2006

 ¹G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006. 9/67




NPTEL Mithesh M. Khapra CS7015 (Deep Learning) : Lecture 9

So, this work which I am going to talk about they trying to understand what has changed since the late 90s or the early 2000. How did deep learning become so popular despite this problem with training them right this problem was there?

So, what happened to them solve it right. And this field actually got revived by this seminal work by Hinton and others in 2006 and.

(Refer Slide Time: 00:46)

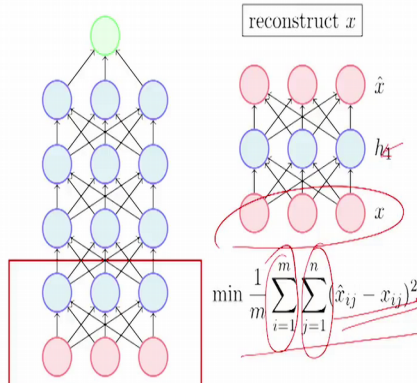
Let's look at the idea of unsupervised pre-training introduced in this paper ...
(note that in this paper they introduced the idea in the context of RBMs but we will discuss it in the context of Autoencoders)



NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9



So, let us see what that idea was. So, this is the idea of unsupervised pre training. In the original paper they introduces idea in the context of something known as R B M's, which we will do in the last 33 percent of the course. But we could do the same with auto encoders which we have already done. So, in this lecture I am going to talk about this idea in the context of auto encoders.

(Refer Slide Time: 01:08)



reconstruct x

- Consider the deep neural network shown in this figure
- Let us focus on the first two layers of the network (x and h_1)
- We will first train the weights between these two layers using an **unsupervised objective**
- Note that we are trying to reconstruct the input (x) from the hidden representation (h_1)

$$\min \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^n (\hat{x}_{ij} - x_{ij})^2$$


Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9

So consider the deep neural network shown in this figure. So, the a module name and the idea was unsupervised pre training. So, that itself is a giveaway of what is going to

happen. So, suppose this is the deep neural network that I have designed for a particular classification task. So, what it is doing is this taking an input which is the red colored neurons that you see at the input it has 4 hidden layers; that means, it is 4 layer deep. And then you have the output layer, which tells you whether positive or negative right. That is the network that I have and I know that this is hard to train such a network the loss will not converge and I will not get anything meaningful.

So, what these guys suggested is that forget about the supervised criteria that you have; that means, you are trying to minimize a classification loss just forget about that just take the first 2 layers of this network which is x and h_1 right. So, you take the original input x . You feed it to some transformations and you get the hidden representation h_1 and now try to reconstruct x from h_1 , what is this?

Student: Auto encoder.

Auto encoder ok, what is the objective of the auto encoder?

Student: (Refer Time: 02:19).

It is exactly this, for each of the m training examples look at each of the dimensions of your input and minimize the square difference between the actual input and the predicted input right, is that fine. That is what an auto encoder does. So, this is what they suggested ok. So, right now I am not telling you why this makes sense and all that that is what we will do later right. Now I am just telling you the trick then we will analyze by that trick works. And why is this objective unsupervised?

Student: (Refer Time: 02:52).

Because we are not using any labels we just giving an input and we just reconstructing the input we only have x 's we do not have y 's of course, eventually we will use the y , but at this stage when I am calling it unsupervised pre training I am not using the y .

(Refer Slide Time: 03:10)

reconstruct x

\hat{x}

h_1

x

$$\min \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^n (\hat{x}_{ij} - x_{ij})^2$$

- Consider the deep neural network shown in this figure
- Let us focus on the first two layers of the network (x and h_1)
- We will first train the weights between these two layers using an **unsupervised objective**
- Note that we are trying to reconstruct the input (x) from the hidden representation (h_1)
- We refer to this as an unsupervised objective because it does not involve the output label (y) and only uses the input data (x)

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9 11/07

Now, at the end of this, what would happen, yeah what would h_1 learn?

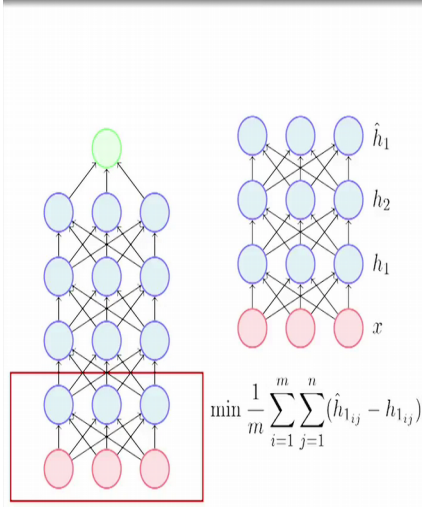
Student: (Refer Time: 03:19).

It will learn an abstract representation of x , was that our original task? What were we interested in.

Student: (Refer Time: 03:29).

In the classification task, but we are doing something very different, why we will see ok. Now guess what would the next step be does this make sense.

(Refer Slide Time: 03:36)



- At the end of this step, the weights in layer 1 are trained such that h_1 captures an abstract representation of the input x
- We now fix the weights in layer 1 and repeat the same process with layer 2
- At the end of this step, the weights in layer 2 are trained such that h_2 captures an abstract representation of h_1
- We continue this process till the last hidden layer (*i.e.*, the layer before the output layer) so that each successive layer captures an abstract representation of the previous layer

$$\min \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^n (\hat{h}_{1ij} - h_{1ij})^2$$

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9 11/07

Now, at the end of the first unsupervised pre training, I have ensured that h_1 which is this layer has learned some abstract representation of the input right, and that I know from the auto encoder I mean the auto encoder which we have learned earlier right that learns an abstract representation of the input.

Now, I have this so; that means, given an input, I know how to compute an extract representation and I am also sure that it captures the important characteristics of the data. I will just repeat this process I know that I have 4 layers in my original network. So, I will now take h_1 try to compute h_2 and then reconstruct h_1 from it. So, the in effect what am I doing in plain English learning and even more.

Student: (Refer Time: 04:19).

Abstract representation of the input, h_1 was already one abstract representation now from this I am learning an even more abstract representation and does the objective function makes sense right. All I have done is replaced x by h_1 right. In both these spaces the rest of it is the same for all the training examples for all the dimensions. And throughout I am assuming that we are n layers, I mean sorry, n neurons and every layer including the input layer is that fine.

Now, what would the next step be fix the weights in h_1 layer fix the weights in it is 2 layer and now try to reconstruct h_2 from this h_2 right. And in this way we will continue

and learn all the hidden representations. Does that look right. So, at least this much we believe it because we know that auto encoder works and you are just using an auto encoder and we are using it incrementally, from every abstract representation learn an even more abstract representation ok.

Now, at the end of this what will I do, what was my original task?

Student: classification.

Classification. So, what will I do?

Student: (Refer Time: 05:23).

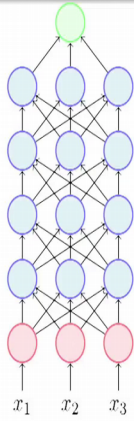
What is a network that I have, when I finish this unsupervised pre training?

Student: (Refer Time: 05:29).


No tell me of the diagrams that you see on the slide, how much of the network would I have right? Everything except the green output layer right because the last step would be take h 4 or sorry, take h 3 and reconstruct h 3 from it and in the process learn h 4 right is that clear.

So, I would have learnt till that point. And now what I am going to do is something very simple.

(Refer Slide Time: 05:59)



- After this layerwise pre-training, we add the output layer and train the whole network using the task specific objective
- Note that, in effect we have initialized the weights of the network using the greedy unsupervised objective and are now fine tuning these weights using the supervised objective

$$\min_{\theta} \frac{1}{m} \sum_{i=1}^m (y_i - f(x_i))^2$$


Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9

I will after this layerwise pre training is done, I will add my output layer now all the weights in my network for every layer have been initialized.

And they have been initialized in a way that, that layer learns a good abstract representation of the input right. That is the thing that we have achieved at the end of unsupervised pre training that every layer has learned and more more and more abstract representation of the input right. Now I will keep all these weights initialized to whatever I learned in the pre training setup does that make sense.

So; that means, instead of taking this big network with the output layer and initializing the way it is randomly, I am just going to use whatever weights I learned using the unsupervised pre training ok. So, can you tell me what has happened in terms of the error surface and so on or my movement in the w b plane or in this case, this very high dimensional w plane.

I have reached some configuration for the w 's, where I know that each of these layers is a good meaningful representation of my original input right. Is that a fair statement in English, how many of you agree with this ah? Anyone has any questions at this point, one layer weights that is what you do in answer because if you train all the problem then you are again entering the same problem which you had earlier right.

You cannot back you cannot back propagate through all the 4 layers because now it is a deep network and we know that does not work. So, at every layer you fix whatever you have learned so far. And at a time you are training only one layer. So, that is one interesting way of looking at it right you know that the deep neural network with 4 layers was not trainable.

So now we have reduced it to one layer at a time, I knew that one layer at a time works right is that fine. Now I will add the output layer and what will I do train the weights of the

Student: Output layer (Refer Time: 07:53).

I will not just do that I will fine tune the entire network; that means, I will train the weights of the output layer. And I will also fine tune the entire network, but now I am

contradicting myself I just gave an answer to him, that again I am doing this deep training and I know that deep training does not work.

But this actually works do you get the difference right. One is that when I start from I take this big network I start from random weight initialization and try to train it that is the story from 1986 to 2006 that in most cases these networks did not converge right.

So now in 2006, we came up or someone came up with this idea of unsupervised pre training where you train the layer network one layer at a time, you do up till the last layer now you add the output layer and then fine tune the entire network; that means, back propagate over the entire network is a set up clear to everyone how many you understand the setup.

Now, again when I am doing the last step, which is known as fine tuning I have to back propagate over the entire network because I am saying I will adjust all the weights, but suddenly this works, as compared to starting from scratch. Right you see the problem and you see why this is important then because this has now given you a way of training deep neural network I still not told you, why it works.

We will delve into it, but not really give any concrete answers because concrete answers do not exist, but we will at least try to get some intuitions behind why it works. So, you get the setup now that this is what was happening till 1986 to 2006. And now with this idea suddenly deep neural networks were being able to train well.

So, in effect what we have done is, we have initialized the weights of the network using the unsupervised objective right. So now, initial starting with random weights, we have some weights which cater to the unsupervised objective that we had and the unsupervised objective was us layer wise reconstruction. So, that is what has happened in plain English is that fine everyone gets that.

(Refer Slide Time: 09:55)

Why does this work better?

- Is it because of better optimization?
- Is it because of better regularization?

Let's see what these two questions mean and try to answer them based on some (among many) existing studies^{1,2}

¹The difficulty of training deep architectures and effect of unsupervised pre-training - Erhan et al, 2009

²Exploring Strategies for Training Deep Neural Networks, Larocelle et al, 2009

MITESH M. KHAPRA CS7015 (Deep Learning) : Lecture 9

Now, the question is, why does this work better. And I give you 2 options and I want to think about both these options ok. Is it because of better optimization or is it because of better generalization no that is not an option, but I of course, we will relate it to that, but given these 2 I want you to think whether there is any difference between these 2 statements or not, that is the first thing I want you to see. How many if you get the difference between these 2 statements, not many why is it so? What is optimization deal with dash data or dash data.

Student: (Refer Time: 10:35).

The answer you can give dash right, dash 1 data or dash 2 data what is optimization deal with?

Student: Training data.

Training data optimization remains on training data what does generalization depend on?

Student: It as 0.

It as 0 so, you get the difference between these 2 questions fine. So, let us try to answer this again here right this is 2006 to 2009 period that I am going to talk about. There are some answers and just bear with me I will give you those answers some of them will not look very convincing, but what happened after that or as a result of these investigations,

that is more important right whether these answers make sense or not, they will make sense to an extent I am not saying that we will just be bluffing.

But it will not be very convincing because there is no theory behind it right. So, what is convincing if I give you a proof that this less this is equal to that right then if we give you a proof and everything you do not have any other questions, that is not what I am going to give you. I am going to give you some intuitions, because that is all these existing works from 2006 to 2009, had and then I will make a commentary on that which will lead us to some other things ok. So, just bear with me for a few minutes right.

Student: (Refer Time: 11:46).

That is the optimization problem if that was the case the, I will just come to that that is what I want to talk about ok. So, it is. So, these are the 2 questions that we are dealing with right, and the answer is depends. So, we will see what it is.

(Refer Slide Time: 11:58)

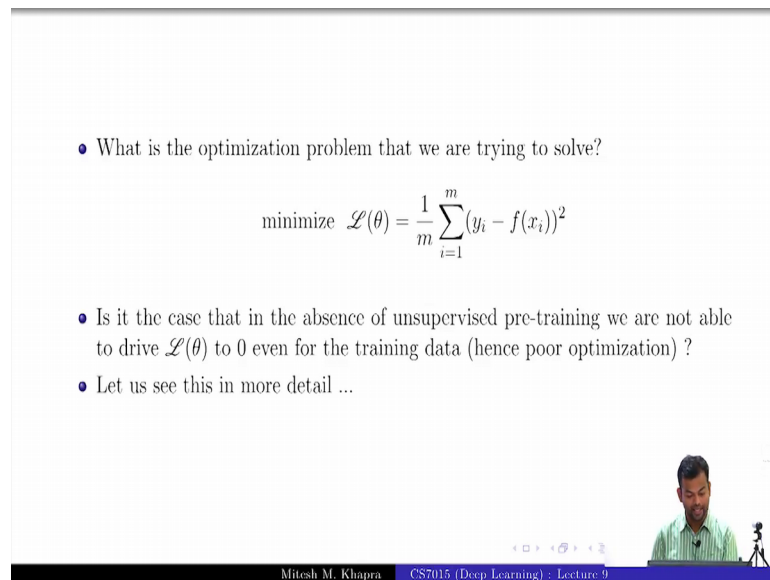
Why does this work better?

- Is it because of better optimization?
- Is it because of better regularization?

Mitesh M. Khapra CS7015 (Deep Learning) - Lecture 9

So, let us first examine the case when it is because of better optimization.

(Refer Slide Time: 12:03)



• What is the optimization problem that we are trying to solve?

$$\text{minimize } \mathcal{L}(\theta) = \frac{1}{m} \sum_{i=1}^m (y_i - f(x_i))^2$$

• Is it the case that in the absence of unsupervised pre-training we are not able to drive $\mathcal{L}(\theta)$ to 0 even for the training data (hence poor optimization) ?

• Let us see this in more detail ...

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9

So, let us first understand what is the meaning of this question, when I ask is it because of better optimization then the question that I am asking you is that, the first set up where I was trying to train everything from scratch, compared to the second set up where I had this unsupervised pre training, is it that the optimization problem becomes easier in the second set up. Now if the optimization problem becomes easier what do I actually mean by that that I was able to drive the dash to dash.

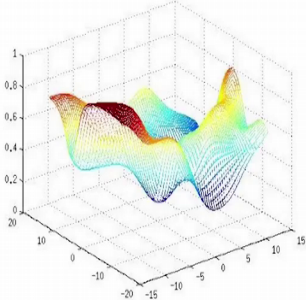
Student: Loss to 0.

Loss to 0 right. So, is it that this is the optimization problem that we were interested in.

So, is it the case that in the absence of unsupervised pre training, we are not able to drive the loss to 0 for the training data and hence poor optimization right that if; you do not do this unsupervised pre training even for the training data we cannot drive at loss to 0; that means, our optimization problem itself is not working properly right I mean the problem is fine.

But the solution is not good you get that, do you understand what is the subtle meaning of this? How many if you get this? So, let us see this in more detail right.

(Refer Slide Time: 13:03)



- The error surface of the supervised objective of a Deep Neural Network is highly non-convex
- With many hills and plateaus and valleys
- Given that large capacity of DNNs it is still easy to land in one of these 0 error regions
- Indeed Larochelle et.al.¹ show that if the last layer has large capacity then $\mathcal{L}(\theta)$ goes to 0 even without pre-training
- However, if the capacity of the network is small, unsupervised pre-training helps

¹Exploring Strategies for Training Deep Neural Networks, Larochelle et al. 2009

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9 16/07

So, the error surface of the supervised objective of a deep neural network is highly non convex it looks something like this or even nastier than this. And in particular it has many hills and plateaus and valleys we saw this even in the toy examples that, we were dealing with right. And given the large capacity of deep neural networks it is still easy to land in one of these 0 error regions, on what basis am I making the statement which theorem?

Student: (Refer Time: 13:32).

Universal approximation theorem that is what the universal approximation theorem told us. In fact, there is a study the paper which has been cited. It showed that if the last year has a very large capacity then you can drive the loss to 0 even without pre training. Do you get the meaning of this? What does it mean? So, I have the input I have a series of hidden layers what do I mean by the last layer has a lot of capacity, what do I mean by that? It has a lot of dash.

Student: Parameters.

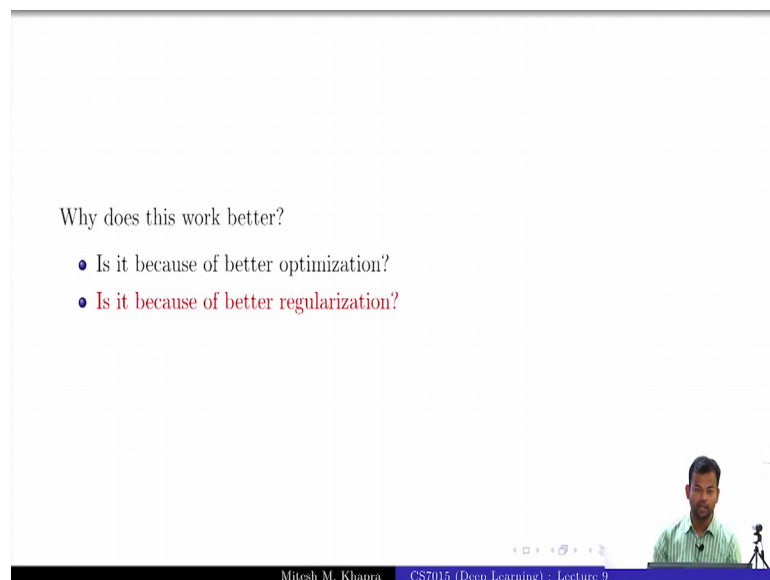
Parameters now how do I create these parameters I will just grow the size of the last hidden layer right. And using that then I will predict this one y , how many of you get this? Please raise your hands good.

So, so that is how I could increase so that is exactly what they did they took a very deep neural network and made sure that the last layer was given a very high capacity, and then they showed that even if you do not do an unsupervised pre training, you can still drive the training loss to 0 right.

So, this was hinting that maybe this is not an optimization problem this is something it is still not very conclusion, but we will just go with these studies we will just all I am saying is that do not shoot the messenger this is what the study says I am just relaying it back to you right. And they will have questions on these which will try to address, but if the capacity of the network is small then the unsupervised pre training helps ok.

So, if you do not have these large capacity networks, but you have very deep networks, in that case unsupervised pre training helps and this is all empirical observation right there is no proof which says that given a capacity k with so much error bound I can guarantee that the loss would be epsilon within the 0 loss and so on. It nothing like that that is what it should have been ideally the case in which case life is much easier for me, but that is not the case this is just an empirical study as are most of the studies done in the period of 2006 to 2009.

(Refer Slide Time: 15:30)



Why does this work better?

- Is it because of better optimization?
- Is it because of better regularization?

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9

So, that tells us something about what optimization means and whether this was an optimization problem or not.

(Refer Slide Time: 15:38)

- What does regularization do? It constrains the weights to certain regions of the parameter space
- L-1 regularization: constrains most weights to be 0
- L-2 regularization: prevents most weights from taking large values

¹Image Source: The Elements of Statistical Learning - T. Hastie, R. Tibshirani
Pg 71

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9

So, let us look at the other question is it because of better regularization. So, what does regularization do or you gave the exact answer it constrains the weights to lie between lie in some regions. So, it does not allow the weights a lot of freedom right. And so, you know what 1 one regulation does it constrains the weights to this box and 1 2 regularization constrains us to this circle why no why this I know this, but why?

Student: (Refer Time: 16:01).

In why the circle I am pretty sure most of you do not know what you are saying, but you are saying the right answers, but anyways I will test this in the quiz ok. So, I have given you another quiz question on camera. So, yeah so a prevents a loss from taking large values.

(Refer Slide Time: 16:18)

• Unsupervised objective:

$$\Omega(\theta) = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^n (x_{ij} - \hat{x}_{ij})^2$$

• We can think of this unsupervised objective as an additional constraint on the optimization problem

• Supervised objective:

$$\mathcal{L}(\theta) = \frac{1}{m} \sum_{i=1}^m (y_i - f(x_i))^2$$

• Indeed, pre-training constrains the weights to lie in only certain regions of the parameter space

• Specifically, it constrains the weights to lie in regions where the characteristics of the data are captured well (as governed by the unsupervised objective)

• This unsupervised objective ensures that the learning is not greedy w.r.t. the supervised objective (and also satisfies the unsupervised objective)

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9

So, indeed pre training also constrains to the way to lie in certain regions of the parameter space, why am I making this statement? What is the meaning of the statement?.

So, I told you that what regularization does and from there I am making this jump and saying that even with pre training the same thing happens that your weights are actually constrained to certain regions of the parameter space, why am I making this statement ? And what are these regions that the weight is constrained to? Think $\mathcal{L}(\theta)$ think $\Omega(\theta)$. Any regulation is of the form $\mathcal{L}(\theta) + \Omega(\theta)$, how many of you get that? Very few ok.

Let us see so it constrains the way to lie in regions where the characteristic of the data are captured well, that is what unsupervised pre training does it is trying to train the network in a way that each layer actually captures the important characteristics of the data, and this is based on our understanding and belief in auto encoders. So, you could actually think of this that the unsupervised objective that you had for all these layers that was actually $\Omega(\theta)$, you are first trying to optimize $\Omega(\theta)$.

So, in a normal regulation problem you put $\mathcal{L}(\theta)$ and $\Omega(\theta)$ together and then you try to balance them, but here you have done it slightly differently you first gave it $\Omega(\theta)$, which is the loss of reconstruction and you asked it to minimize this loss across for every layer, do you get that? How many of you get this yeah?

Student: (Refer Time: 17:48).

No is this fine tuning. So now, what; that means, is that see remember that this is a very high dimensional region. Where you initialize makes a lot of difference. So, with this unsupervised pre training you are at least ending up in reason. So, you could think of it as a constraint that ok, move wherever you want to, but start from here which automatically means that I have I mean I have how to it is some other regions in that parameter space you get that.

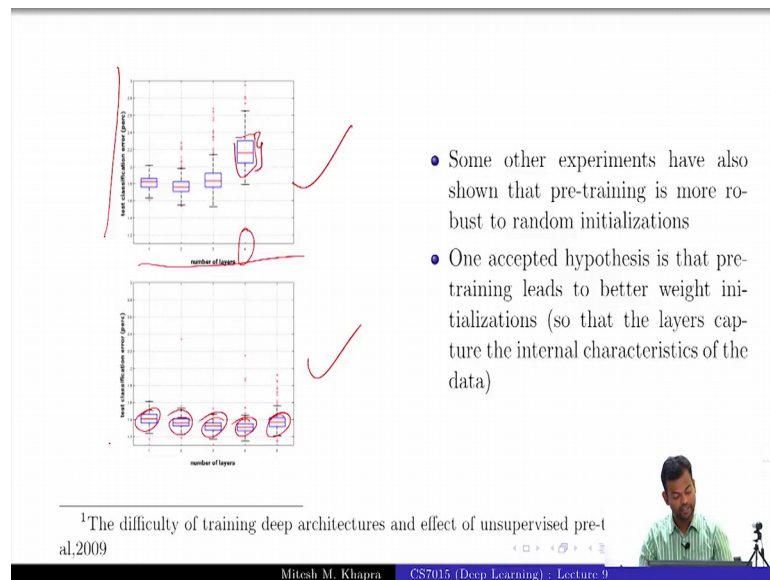
Student: (Refer time: 18:16).

As you typically that would be one thing. And it would also mean that you are starting from there. So, with this early stopping and other criteria you will not be able to grow much out from here right. So, just if that makes sense geometrically from here, you would not be able to move all the way there you get that everyone gets this question and the answer.

So, you see what the answer per is object was and you also see the difference between a normal regularization and this regularization, in the normal case you had l theta plus omega theta put together and then you are trying to minimize the sum of these 2. It was a joint optimization here you have first done omega theta ensured that the weights that you learn minimize this objective. And now you add in the supervise objective which is L theta right.

So now this makes sure that your network cannot be too greedy with respect to L theta because it has been constrained, that has to first honor the omega theta because that is where you started and now from there on it has to decide how to do L theta, does that make sense ? You see how this is acting as a regularizer is that ok. And that links back to your weight initialization thing right fine.

(Refer Slide Time: 19:20)



So, some other experiments have also shown that pre training is more robust to random initializations.

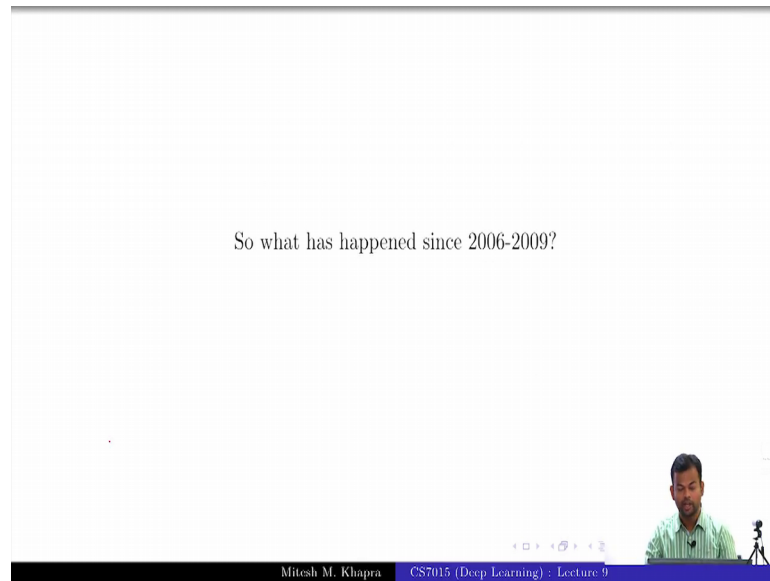
Now, what do I mean that mean by that. So, in these 2 graphs that you see here. So, this on the x axis you have the number of layers that you add to your deep neural network. And on the y axis you have the error that your network gives, when you try different initializations right. So, this box actually tells you the variance in the error.

So; that means, I tried training a network with 4 layers and I tried different initializations and the error varied in this range ok, is that good or bad? What would we want typically something which looks like the plot below right, where all these variances are little; that means, even once you do unsupervised pre training, right it is more robust to random initializations random initializations of what?

Student: (Refer Time: 20:15).

The original random initializations from which point you started the unsupervised pre training ok, because once you have done the unsupervised pre training that is your initialization everyone gets that.

(Refer Slide Time: 20:29)



So, these are some let us see ok. So, these are some empirical studies and let me just make a comment on these right.

So, what happened from 2006 to 2009 is people showed that see this is possible you can actually train a deep neural network, using some of these tricks. We do not have a very clear answer for why this works and you could argue different way. So, this is optimization this is regularization and so on, but I do not have any theory supporting it there is no proof for why unsupervised pre training works all of these are empirical observations

But what it at least established was that it is possible to do this. So now, if it is possible to do this let me see if there are better ways of doing this, do we actually need to do unsupervised pre training, oh I think it is better regularization then why not I try better regularization techniques and see whether that helps. So, that led to the evolution of which thing that you have already seen yeah which regularization technique that you saw in the last class.

Student: Drop out.

Drop out right. So, drop out was something specific to neural networks which was introduced in the context of neural networks. So, this is because people started believing it is possible. So, let us try even better ways of doing that. So, that is how dropout came

out right. Then people said maybe optimization is the problem maybe these earlier algorithms, which up till that point was which algorithm.

Student: (Refer Time: 21:48).

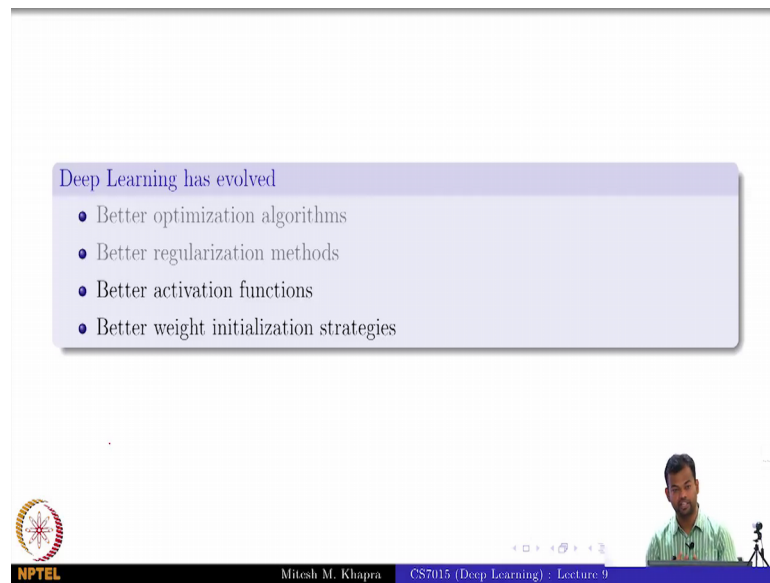
Gradient (Refer Time: 21:49) maybe that was not good. So, let us try to decide and design better optimization methods and that led to the evolution of adam ada gard r m s proper so on right so although these studies were not. So, theoretical in what they were trying to prove, they created this hope which then led to a lot of prolific work in that field right. So, at least you get the context now right the some of these might look oh this is one data set people did experiments on m l s, but I could have taken a different data set and showed that these results do not hold and so on. You could always ask those questions.

But at least what happened is people started believing these and people started questioning that ok, unsupervised pre training is one thing what else can I do. And now what has eventually happened is today no one uses unsupervised pre training right that method which led to the revival of this field, and you would have hoped that that would actually survive for many years that is out.

Now, hardly anyone uses unsupervised pre training it is only used in the context of transfer learning. So, what I mean by that is that if you have a model trained for one classification say classification of images on one data set right.

Now, you have a very small amount of data in some other domain. So, instead of training a network from scratch for this domain you will just initialize it with the weights for whatever you have trained on data set one. So, that is more of transfer learning rather than unsupervised pre training. So, that is still very prevalent, but this reliance on unsupervised training to make sure that the network actually trains that is largely phased out.

(Refer Slide Time: 23:13)



Deep Learning has evolved

- Better optimization algorithms
- Better regularization methods
- Better activation functions
- Better weight initialization strategies

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 9

Because what has happened since 2006 and 2009, is that we have better optimization algorithms which are rms prop ada grad adam even. So, on right many various and even now that research area is active as I was saying just in December there was a paper which pointed out some flaws in adam and how to improve it and so on. We are better regularization methods the most prominent among those being.

Student: Dropout.

Dropout. So, these 2 are things which you have already seen today we are going to talk about better activation functions this is again something which evolved that maybe sigmoid tanh are not good. So, maybe something else is needed and then better weight initialization strategies. So, then people took this inference oh one way of looking at unsupervised pre training is that it actually initializes the weights in a better way from where on it becomes easier for me to reach convergence.

So, why do not i come up with better weight initialization methods itself, instead of relying on this indirect way of initializing the weights. So, you get this. So, get the whole picture, now what we have been doing in the past few lectures and how it connects to the history and these studies which were done from the period 2000 to 2009. How many if you get the whole picture ok? So, that is where we are now. So, today we are going to talk about better activation functions and better weight initialization methods.