

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

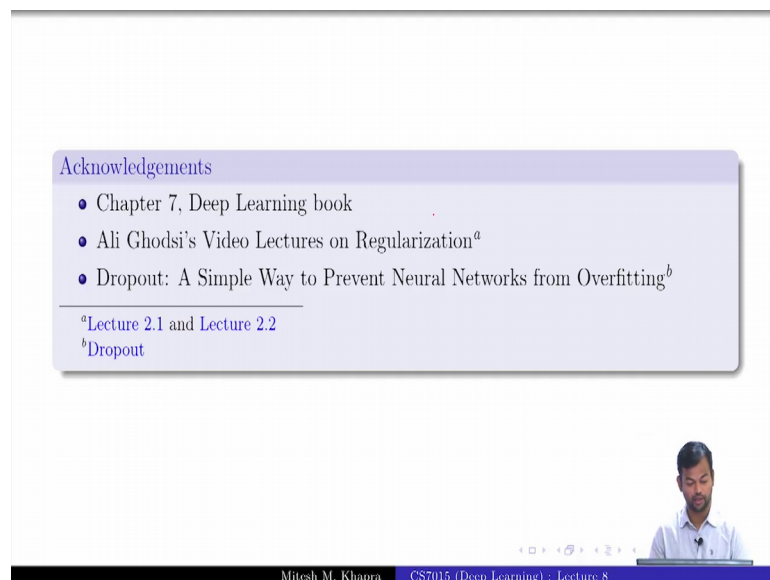
Lecture - 08

Regularization: Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout

So, in this lecture we are going to talk about a bunch of regularization techniques for Deep Neural Networks. You might find some very familiar terms here. For example, L2 regularization, perhaps something else also, but I promise you that we will see a very different interpretation of this from what you have done in your earlier courses, right.

So, again as is the trend in this course, I will start with some basic concepts, I will take today's lecture to finish off the basic part which is the bias variance tradeoff and I will try to make it more informative. Then, what you have done in your earlier courses and in the rest of the lecture which will happen on Friday, we will build upon these basics and then, try to look at these as the regularization forms.

(Refer Slide Time: 00:59)



Acknowledgements

- Chapter 7, Deep Learning book
- Ali Ghodsi's Video Lectures on Regularization^a
- Dropout: A Simple Way to Prevent Neural Networks from Overfitting^b

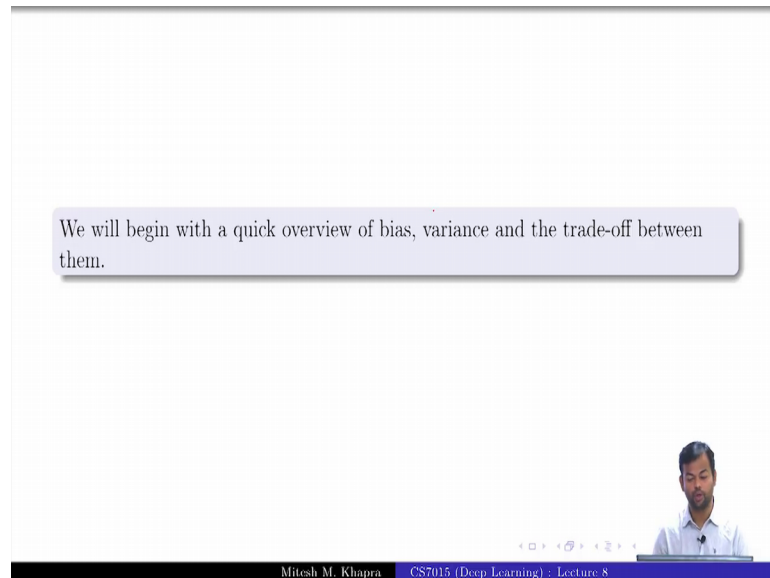
^aLecture 2.1 and Lecture 2.2
^bDropout

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8

So, let us start. So, these are the sources which I have looked at. So, one of them is the chapter 7 from deep learning book. Other is this very good lecture by Ali Ghodsis on Regularization and of course, this paper on drop out, ok. So, let us start with Bias and

Variance. Again some 5-10 minutes would be similar to what you have seen in the middle class, but then I will go on to something different.

(Refer Slide Time: 01:22)

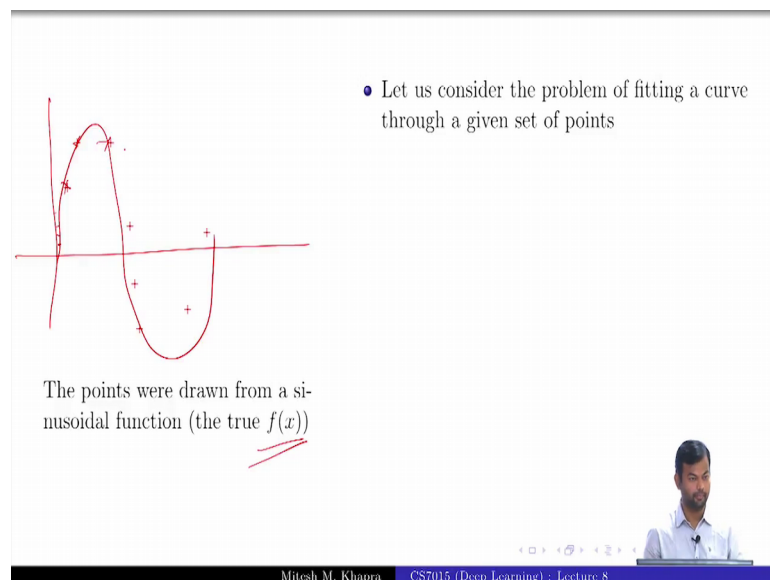


We will begin with a quick overview of bias, variance and the trade-off between them.

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8

So, we will begin with a quick overview of Bias Variance and the tradeoff between them.

(Refer Slide Time: 01:26)



- Let us consider the problem of fitting a curve through a given set of points

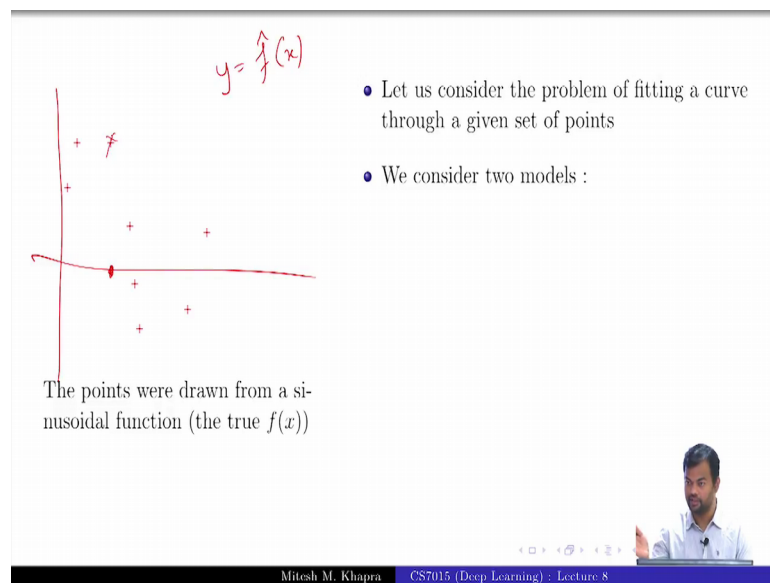
The points were drawn from a sinusoidal function (the true $f(x)$)

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8

So, let us consider the problem of fitting a curve through a given set of points, ok. Now, remember I have always been telling you that there is always this true relation between x and y which is f of x , and which we never know.

So, we do not know what this is. In the movie example, we do not know what this is in the credit card for detection or in the oil mining, example. In this particular example I know it. So, what I have done is, I know that the true relation between x and y is the sinusoidal curve. I know this, but instead of giving you every point on this sinusoidal curve what I have done is, I have such sampled some points from it. I have taken some points and given to you.

(Refer Slide Time: 02:11)



$y = \hat{f}(x)$

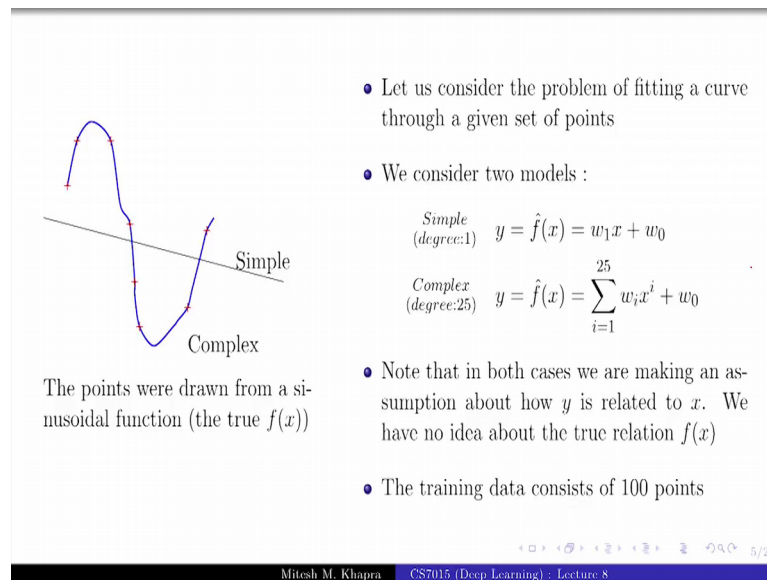
- Let us consider the problem of fitting a curve through a given set of points
- We consider two models :

The points were drawn from a sinusoidal function (the true $f(x)$)

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8

So, from now on we will behave as if we do not know that this is how it came. It is a big secret, and we now want to fit a curve to this. That means, I want to learn the function \hat{f} of x which of course will have some parameters and what will be my goal is that now let us look at this. Again my goal would be if I feed at this point after the model strain, the output should be as close to this point as possible. That is our training criteria. Everyone gets this?

(Refer Slide Time: 02:48)



• Let us consider the problem of fitting a curve through a given set of points

• We consider two models :

$$\begin{array}{l} \text{Simple} \\ \text{(degree:1)} \end{array} \quad y = \hat{f}(x) = w_1 x + w_0$$
$$\begin{array}{l} \text{Complex} \\ \text{(degree:25)} \end{array} \quad y = \hat{f}(x) = \sum_{i=1}^{25} w_i x^i + w_0$$

• Note that in both cases we are making an assumption about how y is related to x . We have no idea about the true relation $f(x)$

• The training data consists of 100 points

The points were drawn from a sinusoidal function (the true $f(x)$)

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8 5/29

So, we consider two models. The first model is a simple model. How many parameters does it have?

Student: 2

Two parameters, right. The other model and this is what happens when I train the simple model. Of course, I will get a line, but do you see something special about this line. Why did I get this as a line or this as a line? So, on average it is trying to minimize the distance from all the points. If I have this as the line, then I will have a very high error for these points, right. So, just something which goes along the average, and hence the sum of the squared errors would be minimized.

So, it is important that when you see these figures, you should make these connections to the math behind it. So, this is the geometry, you have to make connections to the math behind it, right and I hope all of you make that connection. Now, I take a complex model which is a degree 25 polynomial. So, this is $w_1 x + w_2 x^2 + w_3 x^3$ and so on. It is a degree 25 polynomial that I have used and I again learn the parameters of this using how will you learn the parameters. You have a quiz 2 days from now on Gradient Descent.

What else do you know? If you know any other algorithm, of course you know, but getting this end right what else will you use? You can use gradient descent for learning.

These parameters, the same idea, right. You will define a loss, you will compute the gradients with respect to all these parameters. How many of them are there? Here 26 and just update those parameters till a fixed number of iterations or any convergence criteria, and this is the curve which I get for the complex model. Note this in both these cases, we are making an assumption about how y is related to x , right. In this case, I made a simple assumption. In this case, I made a slightly complex assumption, but in both the cases, we do not know what is true.

Relation is the two, relation is actually the sine curve, but we do not know that we are just making an assumption. So, you remember the five things in machine learning, you have a data, you make an assumption about how the input is related to the output. So, these are my two assumptions. Then, I have some parameters. You know the number of parameters in these cases? I use a learning algorithm which happens to be gradient descent and then, I minimize an objective function which would be squared error loss in this case, fine.

Now, the training data actually consists of 100 points, but you do not see 100 points here.

(Refer Slide Time: 04:58)

The points were drawn from a sinusoidal function (the true $f(x)$)

- We sample 25 points from the training data and train a simple and a complex model
- We repeat the process ' k ' times to train multiple models (each model sees a different sample of the training data)
- We make a few observations from these plots

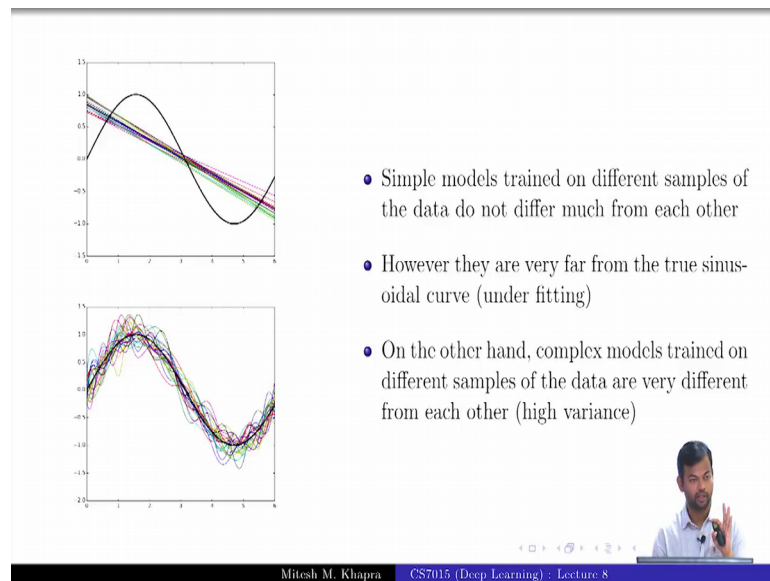
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8

So, what I have done is, I have sampled some 25 points from here and use that as the training data. So, I have learned my parameters w_1 and w_{naught} or w_{25} up to w_{naught} . Using these 25 points, now I will repeat this experiment k times. What I do is every time I will get a different sample of 25 points and I will try to learn the parameters

of the model. Will I get the same curve every time? Will I get the same function every time? No, my parameters would change slightly, right because my training data is different.

So, I am trying to learn it differently to adjust to that training data. So, my function is going to be different. It is the same form. It is either the linear function or the polynomial function, but the parameters, the coefficients are going to be different.

(Refer Slide Time: 05:43)



I will actually draw these different functions and we will make some observations from that. So, this is the black curve that you see is the true sinusoidal curve from which the data has come. The blue line is one of these functions which I have trained from one random sample of the data.

Now, I train different functions from different random samples of the data and see what happens. I get different lines. This obvious can you relate to this every time. I am basically learning a different value of w_1 and w_{naught} . Is that and I have done this 25 times and plotted these lines. What do you observe with respect to each of these if you compare any line to any other line?

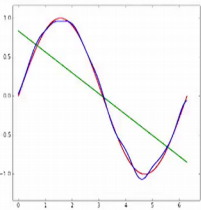
So, if you compare one of these lines to the remaining 24 lines, what do you observe? They are very close to each other. They are not very different from each other, however there is a problem. They are very far from the actual function; that means we are under

fitting. We have very few parameters in fact only 2. That is why we are under fitting. Let us look at the other case, fine. This is the function, the polynomial, the blue curve that you see is the polynomial that I learned from one random sample of the data.

Now, I am going to learn this from a different sample of the data. You see what happens? You see that the green curve is actually very different from the blue curve. You see that here actually this was speaking whereas, this is going down. Similarly this was speaking, but this is going down and so on. So, you see that there are clear differences between the two curves and if I draw the next curve, you see it is even more different. The same function learnt from different data point is turning out to be very different. Why, because it is over fitting on those 25 points that I have given. The simple model did not even have the capacity to do or fit because it is just two parameters.

How much can I over fit? I will just end up drawing the average line right, but here it is really able to over fit and you see that these 25 curves or I do not know how many curves that I will draw, all of these are going to be very different from each other. You see that, and everyone agrees that this would happen if you actually try to do this. So, complex models train on different samples of the data are very different from each other. What is happening there is over fitting.

(Refer Slide Time: 07:54)



Green Line: Average value of $\hat{f}(x)$ for the simple model
Blue Curve: Average value of $\hat{f}(x)$ for the complex model
Red Curve: True model ($f(x)$)

- Let $f(x)$ be the true model (sinusoidal in this case) and $\hat{f}(x)$ be our estimate of the model (simple or complex, in this case) then,
$$\text{Bias}(\hat{f}(x)) = E[\hat{f}(x)] - f(x)$$
- $E[\hat{f}(x)]$ is the average (or expected) value of the model
- We can see that for the simple model the average value (green line) is very far from the true value $f(x)$ (sinusoidal function)
- Mathematically, this means that the simple model has a high bias
- On the other hand, the complex model has a low bias

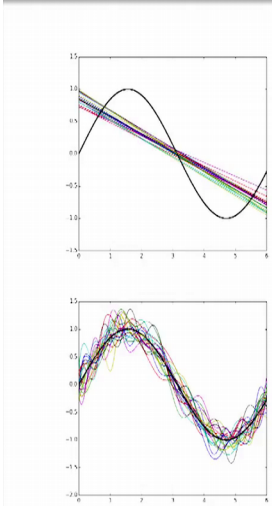
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8 8/29

Now, let me define two concepts. From statistics 1 is bias. Bias is very simple. It tells us that this is the true function. If you are trying to learn the approximate function and you

do it many times, then you will get an expected value of the function. So, it tells you how much does this expected value differ from the true function, ok. You get the definition. The definition is straightforward, ok. Now, for the simple line or the simple model, the green line that you see is actually the average of all those 25 lines that you had seen, ok. What can you say about the bias, very high right because this difference is very high.

This green line is very different from the red curve which is my true function, right predicted and true function. Now, what about complex model? The blue curve that you see is actually the average of all those 25 different curves that I had drawn. So, what is the bias? It is very low. Does that make sense? This means that the simple model has a high bias and the complex model has a low bias. Is it clear to everyone?

(Refer Slide Time: 09:06)



- We now define,

$$\text{Variance } (\hat{f}(x)) = E[(\hat{f}(x) - E[\hat{f}(x)])^2]$$
 (Standard definition from statistics)
- Roughly speaking it tells us how much the different $\hat{f}(x)$'s (trained on different samples of the data) differ from each other
- It is clear that the simple model has a low variance whereas the complex model has a high variance

9/29

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8

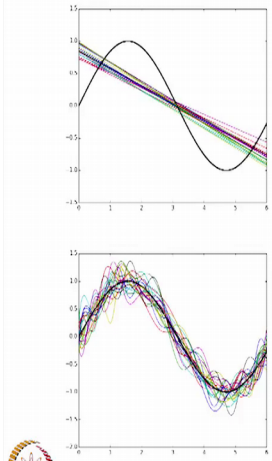
Now, let us define another quantity which is variance. Everyone knows what variance is. So, this is one of the functions that I have learned. This is the average of that function and the variance tells me the spread. Now, based on the figures that you have seen, can you tell me what would happen for the simple model; low variance or high variance?

Student: Low variance.



Low variance, because all these models were very close to each other; there was not much spread in the models. What about the complex model?

High variance: all these models were very far from each other. The spread was very high, ok. So, roughly speaking it tells us how much the different f act $f(x)$ is that you are learning, how different are they from the average f of x .

(Refer Slide Time: 09:50)



- In summary (informally)
- Simple model: high bias, low variance
- Complex model: low bias, high variance
- There is always a trade-off between the bias and variance
- Both bias and variance contribute to the mean square error. Let us see how



NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8

So, informally I can say the following simple model has a high bias low variance, complex model has a low bias high variance and as always going to be a tradeoff between the bias and variance.

So, why is there always a tradeoff between the bias and variance? People have done MI course. Why is there a tradeoff? How many of you know the mathematical answer to that? You have not done this in the above course, no. So, it turns out that both bias and variance contribute to the mean square error and let us see how.