

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

**Module - 8.3**  
**Lecture - 08**  
**True error and Model complexity**

So, now we will try to see that how does this true error that, we see depend on the model complexity, ok.

(Refer Slide Time: 00:19)

Using Stein's Lemma (and some trickery) we can show that

$$\frac{1}{n} \sum_{i=1}^n \varepsilon_i(\hat{f}(x_i) - f(x_i)) = \frac{\sigma^2}{n} \sum_{i=1}^n \frac{\partial \hat{f}(x_i)}{\partial y_i}$$

- When will  $\frac{\partial \hat{f}(x_i)}{\partial y_i}$  be high? When a small change in the observation causes a large change in the estimation( $\hat{f}$ )
- Can you link this to model complexity?
- Yes, indeed a complex model will be more sensitive to changes in observations whereas a simple model will be less sensitive to changes in observations
- Hence, we can say that  
true error = empirical train error + small constant +  $\Omega(\text{model complexity})$

24/84

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

So, using Steins Lemma and some trickery, we can show the following. What is Steins Lemma? So, I had this deal with my students last year. You do not ask me what Steins Lemma is, I will not ask you what Steins Lemma is, ok. So, it is some lemma which tells us that, this quantity, what was this quantity the last term which was troublesome rate that covariance term which was troublesome. That is this quantity. This quantity is actually equal to this quantity.

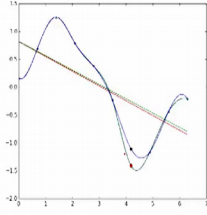
So, let us buy that. Let us all of us agree that Steins Lemma is correct and it tells us that this is the case, and you saw the quiz 1 paper, fine from last year I mean, ok. So, now we will work with this premise and we will see what it actually tells us. Now, when will this quantity be high? So, what this is telling us? I mean jokes apart. Let us try to focus again that this quantity is actually equal to the summation of this quantity, ok.

Now, let us take one term in this summation when would  $\hat{f}(x_i)$  by  $y_i$  be large. What does it actually tell you? If I change one of these  $y_i$  is a bit when the prediction for it is going to change by a lot. Do you get that? How many of you get this? Some of you do not get this, ok. Just think about it. When would this be high? What does the derivative capture? If the derivative is high, that means a small change in the denominator is going to lead to a large change in the numerator.

What is the denominator? Actually the true  $y$  that, we have observed. What is the numerator? That is the predicted  $y$ . So, what you are saying is that if there is a small change in  $y_i$ , then there is going to be a large change in the prediction, ok. When would this happen? Would this happen for simple models or complex models? Complex models; how many of you say complex models? So, this is the link to model complexity rate and I will make a more intuitive case for this, but at least some of you get this that if your model is very complex, that means it is even one of your data points changes and the prediction of the model is going to change largely.

So, now, relate this back to that sinusoidal model that we had and we had this complex model, every model that I was training which was trained on a different set of 25 examples, the model was vastly different and that is exactly what was happening. When you were changing even one data point, your predictions were changing largely. That means, your model was changing largely. Do you get that intuition? So, indeed a complex model will be more sensitive to the changes in the observation whereas, a simple model will be less sensitive to it, and hence, we can say that the true error is actually equal to the empirical train error plus something which relates to the model complexity. Is that fine?

(Refer Slide Time: 03:17)



- Let us verify that indeed a complex model is more sensitive to minor changes in the data
- We have fitted a simple and complex model for some given data
- We now change one of these data points
- The simple model does not change much as compared to the complex model

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8 25/84

Now, let us first verify that indeed a complex model is more sensitive to minor changes in the data. So, this is some data that I had sampled from the same distribution and I trained one simple model which is the green line which you see that was a linear model and I trained one complex model which was a 25 degree polynomial which you see, ok. Now, what I am going to do is, I am going to take one of these points and change it a bit and I retrain the model.

What happens to the simple model, it does not change much, but what happens to the complex model, it is more sensitive to these observations that I have and that is exactly the quantity that we were interested in. That means, a complex for a complex model which is more sensitive that summation that we care about is going to be high. That means, that difference between the true error and the estimated error is going to be high.

(Refer Slide Time: 04:10)

- Hence while training, instead of minimizing the training error  $\mathcal{L}_{train}(\theta)$  we should minimize

$$\min_{w.r.t \theta} \mathcal{L}_{train}(\theta) + \Omega(\theta) = \mathcal{L}(\theta)$$

- Where  $\Omega(\theta)$  would be high for complex models and small for simple models

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

So, that is why instead of minimizing the train error, we should always minimize the train error plus some quantity which is linked to the model complexity. This is the basis for all Dash methods Regularization method.

So, now you see where this comes from. So, where omega theta would be high for complex models and simple for simple models, you get the intuition for this and the rest of the lecture, we will spend in taking various cases where we will actually show that theta would be high and we are trying to control for omega theta, ok. This quantity for the rest of this lecture and for the rest of this course I will assume that we all know how to deal with.

We have done enough of this, we have done a lot of back propagation, we have done enough derivations of the laws with respect to the output layer and so on everything, right. So, all of us understand how to deal with L train theta where L treat theta is this L equal to 1 to m squared error loss or your log likelihood or any of these losses, right. So, we all know how to deal with this today. We are going to focus on this other term which brings in the regularization, ok.

(Refer Slide Time: 05:15)

- Hence while training, instead of minimizing the training error  $\mathcal{L}_{train}(\theta)$  we should minimize

$$\min_{w.r.t \theta} \mathcal{L}_{train}(\theta) + \Omega(\theta) = \mathcal{L}(\theta)$$

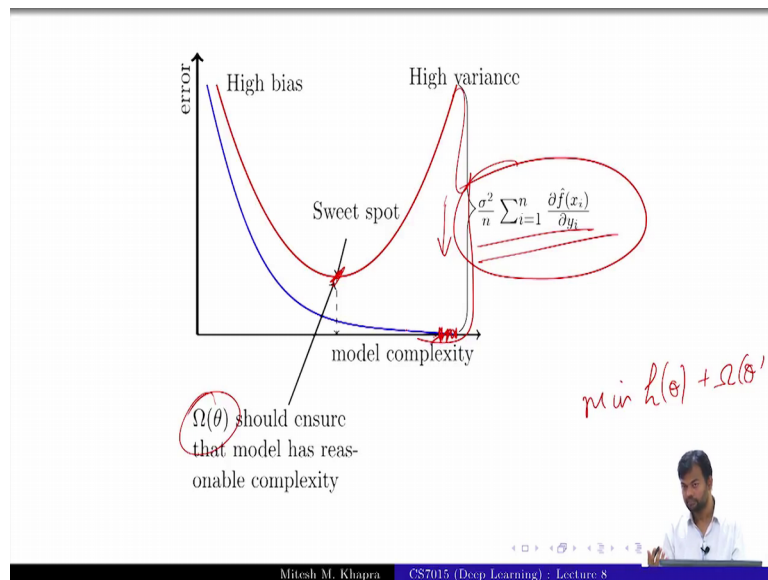
- Where  $\Omega(\theta)$  would be high for complex models and small for simple models
- $\Omega(\theta)$  acts as an approximate for  $\frac{\sigma^2}{n} \sum_{i=1}^n \frac{\partial^2 f(x_i)}{\partial y_i^2}$
- This is the basis for all regularization methods
- We can show that  $l_1$  regularization,  $l_2$  regularization, early stopping and injecting noise in input are all instances of this form of regularization.

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8 26/84

So, what omega theta does is actually acts as an approximation for this. So, what I should have actually tried to minimize is not just  $L_{train}(\theta)$ , but  $L_{train}(\theta)$  plus this other quantity which was there in my equation. You get this my true equation was that my loss is equal to  $L_{train}(\theta)$  plus this term, right which we approximated using Steins Lemma. So, I should have tried to minimize this quantity, but I do not know how to really compute this quantity.

So, I am going to just substitute it by omega theta and ensure that omega theta is such that it is high for complex models and low for simple models. Do you get the recipe? Everyone gets this? How many of you understand this? So, we can show that  $L_1$  regularization  $L_2$  regularization early stopping all of these are actually special cases of this particular formulation that we have, ok.

(Refer Slide Time: 06:09)



Remember that this is the sweet spot that we were aiming for, and this gap is actually, this quantity because we are making a very optimistic estimation of the error whereas, there is actually this quantity which we have been ignoring and that is why we see that the validation error is high, ok. So, is the full picture in terms of the diagram and all the equations that we have seen; Clear fine?

So, we should ensure using omega theta that this gap is also minimized. Therefore, our function should be minimized L theta plus omega theta. So, essentially what we are trying to do is minimize this gap and hence, the model would generalize better on the test is this intuition. Clear to everyone?

(Refer Slide Time: 06:57)

- Why do we care about this bias variance tradeoff and model complexity?
- Deep Neural networks are highly complex models.
- Many parameters, many non-linearities.
- It is easy for them to overfit and drive training error to 0.
- Hence we need some form of regularization.

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

Why do we care about this bias variance tradeoff model complexity? This is not a course on machine learning. They are highly complex models, they have many parameters, many non-linearities. In fact, now can you relate this back to the universal approximation theorem? What is the universal approximation theorem say give me any data, I will give you a deep neural network which will exactly over fit the data, right and that is exactly what we want to avoid. That is why regularization is important in the context of deep neural networks, fine. It is very easy for them to over fit the data and driver training area to 0 and that is why we need some regularization, ok.

(Refer Slide Time: 07:35)

Different forms of regularization

- $l_2$  regularization
- Dataset augmentation
- Parameter Sharing and tying
- Adding Noise to the inputs
- Adding Noise to the outputs
- Early stopping
- Ensemble methods
- Dropout

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

So, today we are going to look at different forms of regularization starting with two regularization; some simple tricks. So, some of these are going to be mathematically motivated, some of these are just going to be heuristics or empirical stuff. So, data set augmentation is one such empirical stuff. How many of you tried data set augmentation for the immunized assignment or the back propagation as parameter sharing and tying is something that? No, I am not.

Please do not give me that look. Yeah I am not suggesting that adding noise the inputs, adding noise to the outputs, early stopping, ensemble methods and drop off, right. So, these are the things that we are going to talk about this and all of this is in the context of regularization where you want to avoid some kind of model complexity.