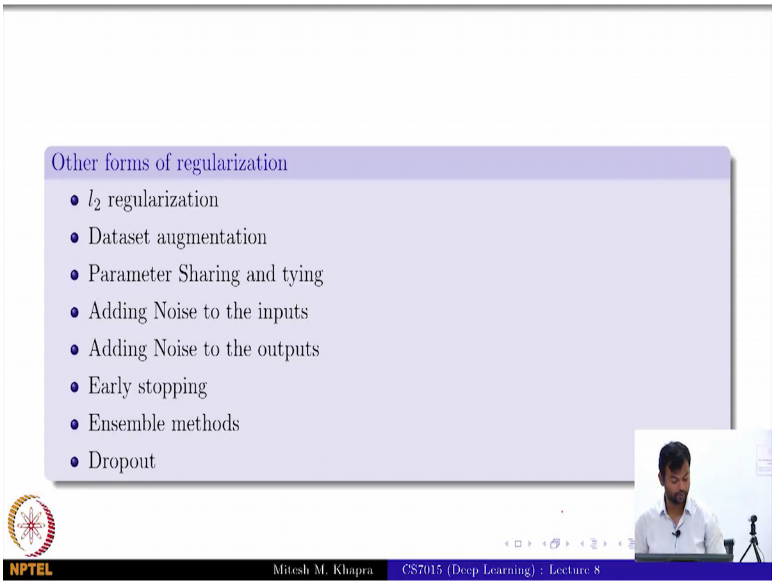
Deep Learning
Prof. Mithesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 8.10
Lecture – 08
Ensemble methods

So next we look at on Ensemble methods, and this is just to build the intuitions for something known as dropout which is very popular technique in deep neural networks and convolution neural networks and even recurrent neural networks.

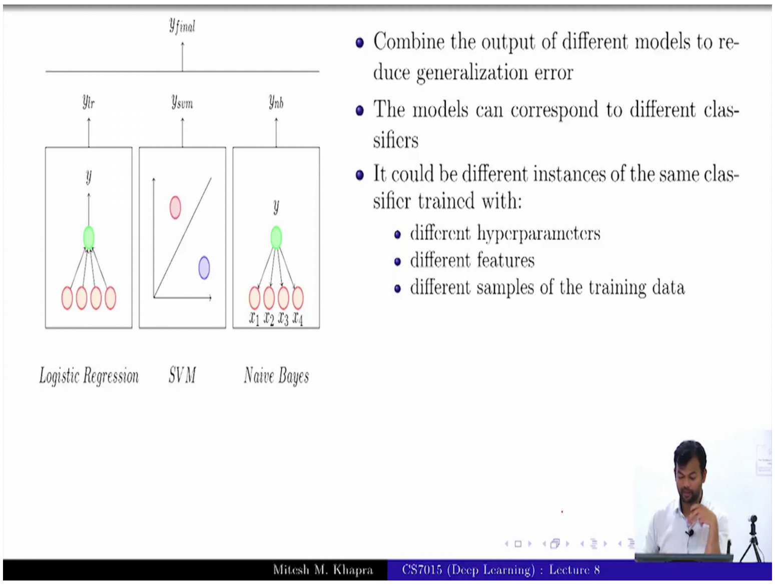
(Refer Slide Time: 00:23)



Other 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

.(Refer Slide Time: 00:27)



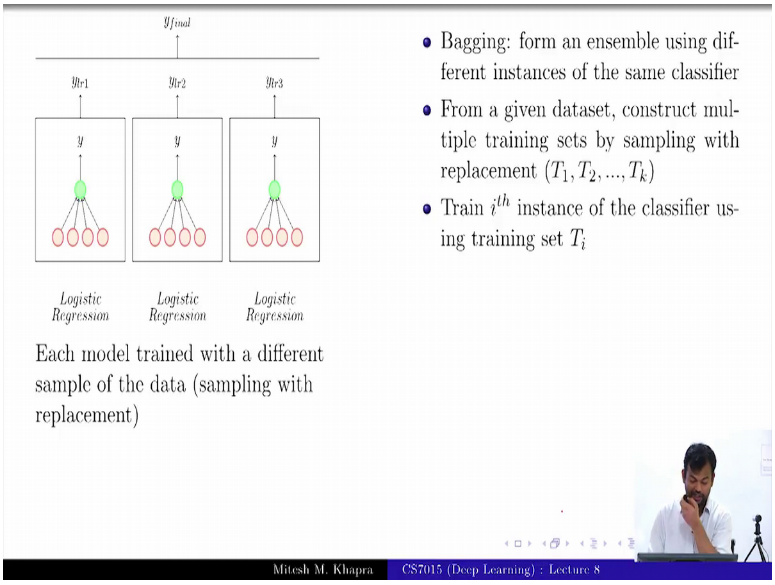
- Combine the output of different models to reduce generalization error
- The models can correspond to different classifiers
- It could be different instances of the same classifier trained with:
  - different hyperparameters
  - different features
  - different samples of the training data

So, how many you have seen ensembles before? Seen it in machine learning ensemble was not done in machinery done with ok. Ravi did it. So, as a combine so the ensemble is essentially just the combining the output of different models, to reduce the generalization error right. Why does that make sense? Have these different models all of these would have different biases and variances right.

So now you are combining them. So, I will end up with a better thing on the test error right. So, that is the idea behind ensemble, now the models could correspond to different classifiers right for example, here I have a Logistic Regression and SVM and a Naive Bayes. I have trained them independently using the same data or different subsets of the data. And a test time I am taking a prediction from all of them and then, taking an ensemble of those predictions that is the basic idea.

Now it could be different instances of the same classifier, trained with different hyper parameters. I could have the same neural network a 3-layer neural network, but trained with different hyper parameters. So, the hyper parameters could be learning rate, it could be bad size, it could be the number of neurons in each layer and so on right. So, it could be same classifier, but different hyper parameters, different features right. So, instead of looking at all the 100 features that i have given, I could train these classifiers with different subsets of the features or different samples of the training data.

(Refer Slide Time: 01:46)



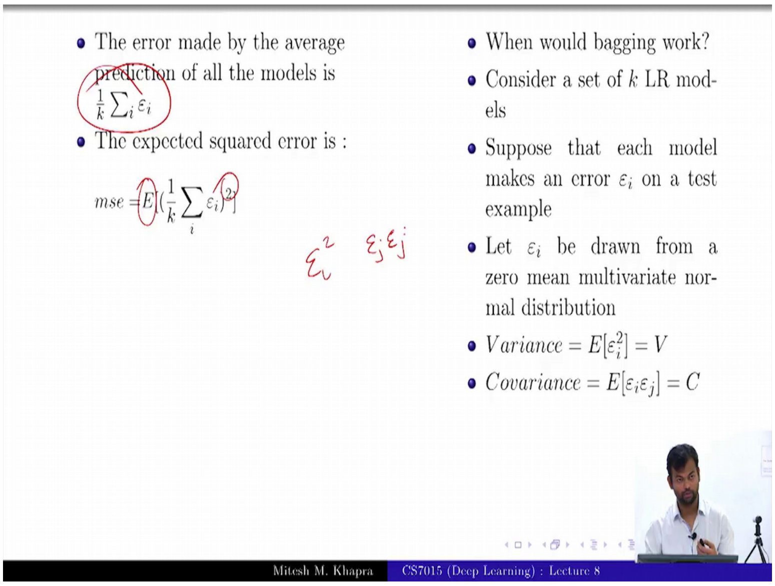
Each model trained with a different sample of the data (sampling with replacement)

- Bagging: form an ensemble using different instances of the same classifier
- From a given dataset, construct multiple training sets by sampling with replacement $(T_1, T_2, ..., T_k)$
- Train i^{th} instance of the classifier using training set T_i

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

So, bagging is one such ensemble method where you have different instances of the same classifier, which are trained on different samples of the training data ok. So, I have one classifiers trained on a subset T 1 of the training data another classifier trained, on a subset T 2 of the training data and so on right and so, each of these model is trained with a different sample of the data.

(Refer Slide Time: 02:12)



- The error made by the average prediction of all the models is $\frac{1}{k}\sum_i \varepsilon_i$
- The expected squared error is :

$$mse = E[(\frac{1}{k}\sum_i \varepsilon_i)^2]$$

$\varepsilon_i^2 \quad \varepsilon_j \varepsilon_j$

- When would bagging work?
- Consider a set of k LR models
- Suppose that each model makes an error ε_i on a test example
- Let ε_i be drawn from a zero mean multivariate normal distribution
- $Variance = E[\varepsilon_i^2] = V$
- $Covariance = E[\varepsilon_i \varepsilon_j] = C$

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

Now, when would bagging actually work, what would you want these classifiers to be? So, each classifier is going to make certain errors ok.

What do you want these errors across classifiers to be dependent, independent?

Student: Independent.

Independent right; so, if one classifier makes the errors on certain test instances, other classifier makes errors on a different set of test instances and the third classifier makes errors on a very different set of instances, that is the condition that you are looking for right. There is errors if all of them make error on the same instance then all of them are collectively going to make an error on the final prediction also right.

Because it is like I asked 3 guys all of them gave me the wrong answer. So, my final answer is going to be wrong, but at least 2 of these 3 guys gave me the correct answer then my final answer is going to be correct right. So; that means, the errors that these models make, I want these errors to be independent if I treat error as a random variable I want these errors to be independent ok.

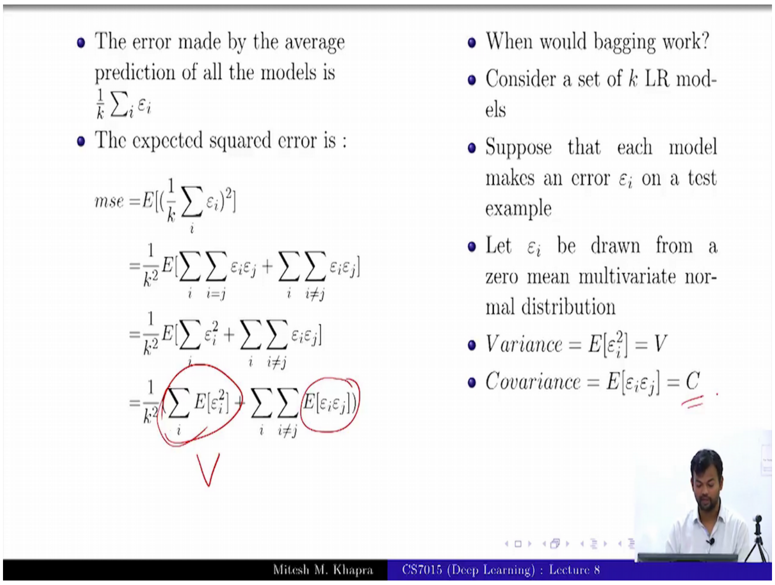
So, so consider a set of k such logistic regression models, suppose that each model makes an error epsilon i on the test example; now let epsilon i be drawn from a 0 mean multivariate normal distribution. So, the variance is equal to V; and how many such epsilons do i have? How many such distributions I am considering?

Student: k.

K right because for each classifier there is a distribution. So, then I can compute the covariance between these random variables ok. I will add that let that covariance be C. Is that fine? Now the true the error made by the average prediction of all the models is going to be given by this model one made an error of epsilon 1, model 2 made an error of epsilon 2.

So, the average error is going to be given by this ok. Now what is this expected squared error? This is the error; this is the expectation this is the square. That is the expected squared error is that fine? Again this is a square of a sum; so, it will lead to a lot of terms of the form epsilon i squares and what will happen now which terms will go to 0 ok.

(Refer Slide Time: 04:23)



- The error made by the average prediction of all the models is $\frac{1}{k}\sum_i \varepsilon_i$
- The expected squared error is :

$$
\begin{aligned}
mse =& E[(\frac{1}{k}\sum_i \varepsilon_i)^2] \\
=& \frac{1}{k^2}E[\sum_i \sum_{i=j} \varepsilon_i\varepsilon_j + \sum_i \sum_{i\neq j} \varepsilon_i\varepsilon_j] \\
=& \frac{1}{k^2}E[\sum_i \varepsilon_i^2 + \sum_i \sum_{i\neq j} \varepsilon_i\varepsilon_j] \\
=& \frac{1}{k^2}(\underbrace{\sum_i E[\varepsilon_i^2]}_{V} + \sum_i \sum_{i\neq j} E[\varepsilon_i\varepsilon_j])
\end{aligned}
$$

- When would bagging work?
- Consider a set of k LR models
- Suppose that each model makes an error ε_i on a test example
- Let ε_i be drawn from a zero mean multivariate normal distribution
- $Variance = E[\varepsilon_i^2] = V$
- $Covariance = E[\varepsilon_i\varepsilon_j] = C$

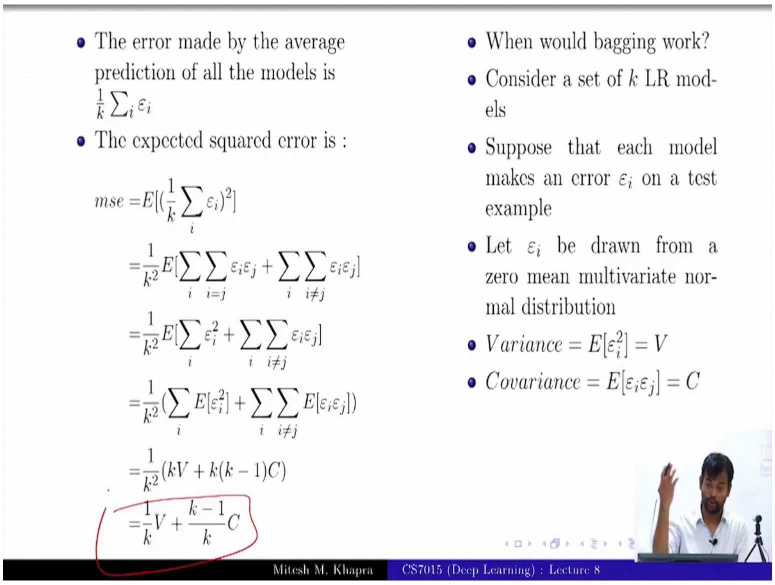
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8

The terms having epsilon i epsilon j again the same thing they are independent. So, I can write the expectation of a product as the product of expectations and those expectations are 0. So, this is what it is going to look like, what is this? Oh sorry actually we had not assumed that the covalence sorry, sorry yeah.

What is this right? And what is this? Covariance I am sorry I have not we had assumed that there is some covariance said wed not assume they are independent right. We would want it to be independent, but in the general case we will assume some covariance and then I will show you the special case where they are independent.

So, then how many Vs do i have here? K right, and how many Cs do i have here?

(Refer Slide Time: 05:08)



- The error made by the average prediction of all the models is $\frac{1}{k}\sum_i \varepsilon_i$
- The expected squared error is :

$$
\begin{aligned}
mse &= E[(\frac{1}{k}\sum_i \varepsilon_i)^2] \\
&= \frac{1}{k^2}E[\sum_i\sum_{i=j}\varepsilon_i\varepsilon_j + \sum_i\sum_{i\neq j}\varepsilon_i\varepsilon_j] \\
&= \frac{1}{k^2}E[\sum_i\varepsilon_i^2 + \sum_i\sum_{i\neq j}\varepsilon_i\varepsilon_j] \\
&= \frac{1}{k^2}(\sum_i E[\varepsilon_i^2] + \sum_i\sum_{i\neq j}E[\varepsilon_i\varepsilon_j]) \\
&= \frac{1}{k^2}(kV + k(k-1)C) \\
&= \frac{1}{k}V + \frac{k-1}{k}C
\end{aligned}
$$

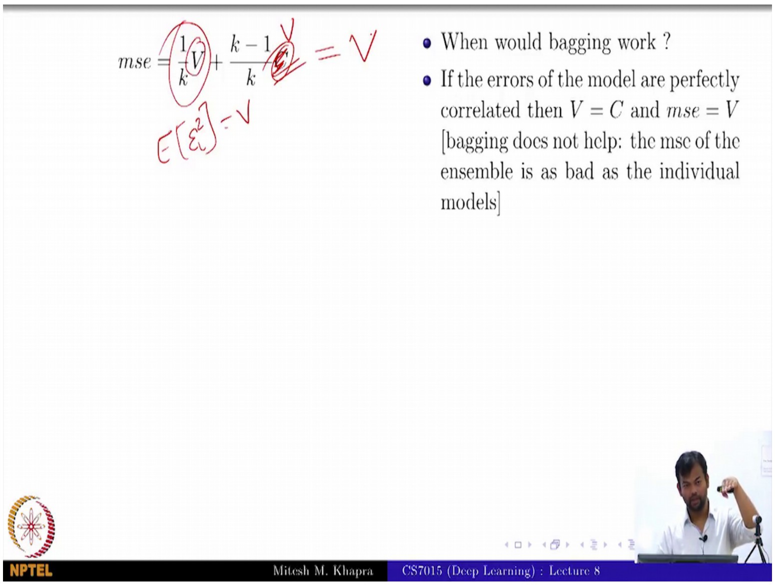
- When would bagging work?
- Consider a set of k LR models
- Suppose that each model makes an error ε_i on a test example
- Let ε_i be drawn from a zero mean multivariate normal distribution
- $Variance = E[\varepsilon_i^2] = V$
- $Covariance = E[\varepsilon_i\varepsilon_j] = C$

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

This summation is k into k minus 1 right or i equal to 1 to k and j equal to i plus 1 to k fine. And so, this is what it looks like now can you make some inferences from this equation, this is what the expected mean square error is going to be. Now think in terms of variance, covariance and tell me when would this be beneficial. I have already told you the answer, if the errors are independent what would covariance be? 0 right.

So, then what is the mean square error 1 by k 1 by k into V right so; that means, bagging would work when your classifiers the k classifiers that you are combining.

(Refer Slide Time: 05:33)



$$mse = \frac{1}{k}\boxed{V} + \frac{k-1}{k}\overset{V}{\cancel{C}} = V$$

$$E[\varepsilon_i^2] = V$$

- When would bagging work ?
- If the errors of the model are perfectly correlated then $V = C$ and $mse = V$ [bagging does not help: the mse of the ensemble is as bad as the individual models]

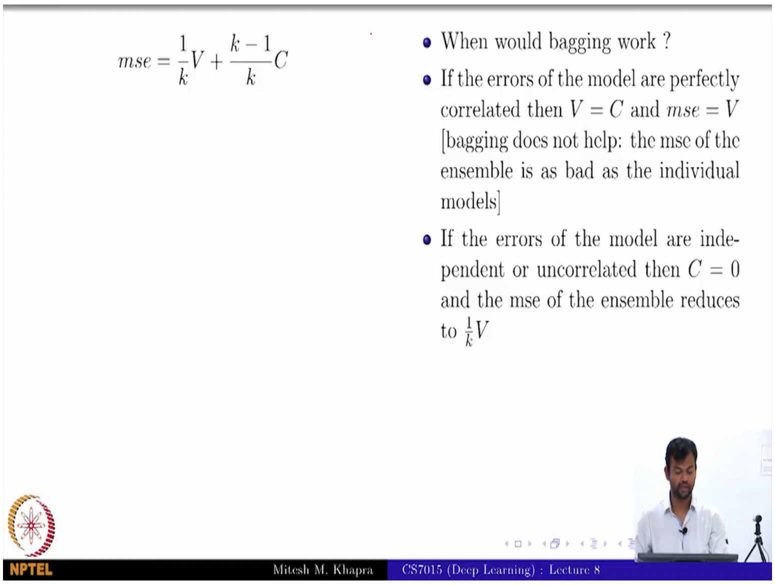
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 8

If the errors are independent, then the mean square error should actually have been V right, for a single classifier it was V right because mean square error is nothing but the expectation of the error expectation of epsilon i square which is nothing but V.

But if you are if you are combining k classifiers and if these classifiers are independent in terms of their errors, then your mean square error is going to be 1 by k into V, because this term is going to disappear ok. Now if your classifiers are perfectly correlated, then what would happen? And basically C is equal to V right, is that fine. So now what would happen? What is the net result if I substitute this as V? Going to be V right.

So, if you are all your classifiers are perfectly correlated, that is the other case we had tried taken. And all of them are making errors on the same test instances and the same errors, right? Then you will not get any benefit of doing bagging, but if you look at the other extreme, where all your errors are independent or all your classifiers are making independent errors, then you will get a benefit your expected mean square error would go down from V to 1 by k into V; everyone gets that?

(Refer Slide Time: 07:10)



$$mse = \frac{1}{k}V + \frac{k-1}{k}C$$

- When would bagging work ?
- If the errors of the model are perfectly correlated then $V = C$ and $mse = V$ [bagging does not help: the mse of the ensemble is as bad as the individual models]
- If the errors of the model are independent or uncorrelated then $C = 0$ and the mse of the ensemble reduces to $\frac{1}{k}V$

So, this was just to develop an intuition that taking an ensemble helps right. And using this intuition now we are going to see at how to do this ensemble in the case of deep neural networks.