EVALUATING A HYPOTHESIS:

Once we have done some trouble shooting for errors in our predictions by:

1.Getting more training examples

2.Trying smaller sets of features

3.Trying additional features

4.Trying polynomial features

5.Increasing or decreasing λ

We can move on to evaluate our new hypothesis.

A hypothesis may have a low error for the training examples but still be inaccurate (because of overfitting). Thus, to evaluate a hypothesis, given a dataset of training examples, we can split up the data into two sets: a training set and a test set. Typically, the training set consists of 70 % of your data and the test set is the remaining 30 %.

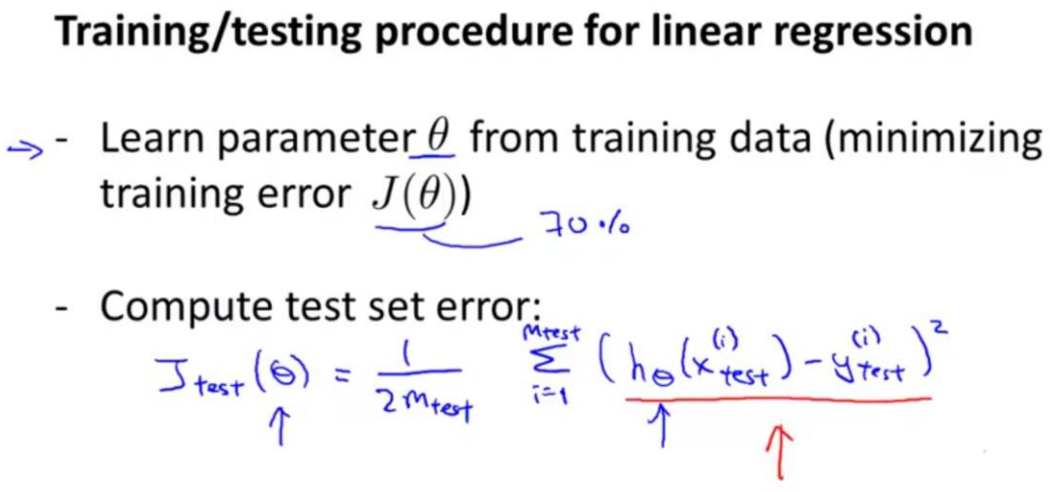
The new procedure using these two sets is then:

Learn Θ and minimize $J_{train}(\Theta)$ using the training set

Compute the test set error $J_{test}(\Theta)$

The test set error:

For linear regression: $J_{test}(\Theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} (h_\Theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$



2. For classification ~ Misclassification error (aka 0/1 misclassification error):

$$err(h_\Theta(x), y) = \begin{matrix} 1 & \text{if } h_\Theta(x) \geq 0.5 \text{ and } y = 0 \text{ or } h_\Theta(x) < 0.5 \text{ and } y = 1 \\ 0 & \text{otherwise} \end{matrix}$$

This gives us a binary 0 or 1 error result based on a misclassification. The average test error for the test set is:

$$\text{Test Error} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} err(h_\Theta(x_{test}^{(i)}), y_{test}^{(i)})$$

This gives us the proportion of the test data that was misclassified.

Misclassification error means that our hypothesis classified a value as 0 whereas it was actually 1 and vice versa.

If we have an error, we denote it by 1 and if we don't , we denote it by 0.

Training/testing procedure for logistic regression

- Learn parameter θ from training data
- Compute test set error:

$$J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)}) + (1 - y_{test}^{(i)}) \log h_\theta(x_{test}^{(i)})$$

- Misclassification error (0/1 misclassification error):

$$err(h_\Theta(x), y) = \begin{cases} 1 & \text{if } h_\Theta(x) \geq 0.5, \ y = 0 \\ & \text{or if } h_\Theta(x) < 0.5, \ y = 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{error}$$

$$\text{Test error} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} err(h_\Theta(x_{test}^{(i)}), y_{test}^{(i)}).$$

Andrew Ng

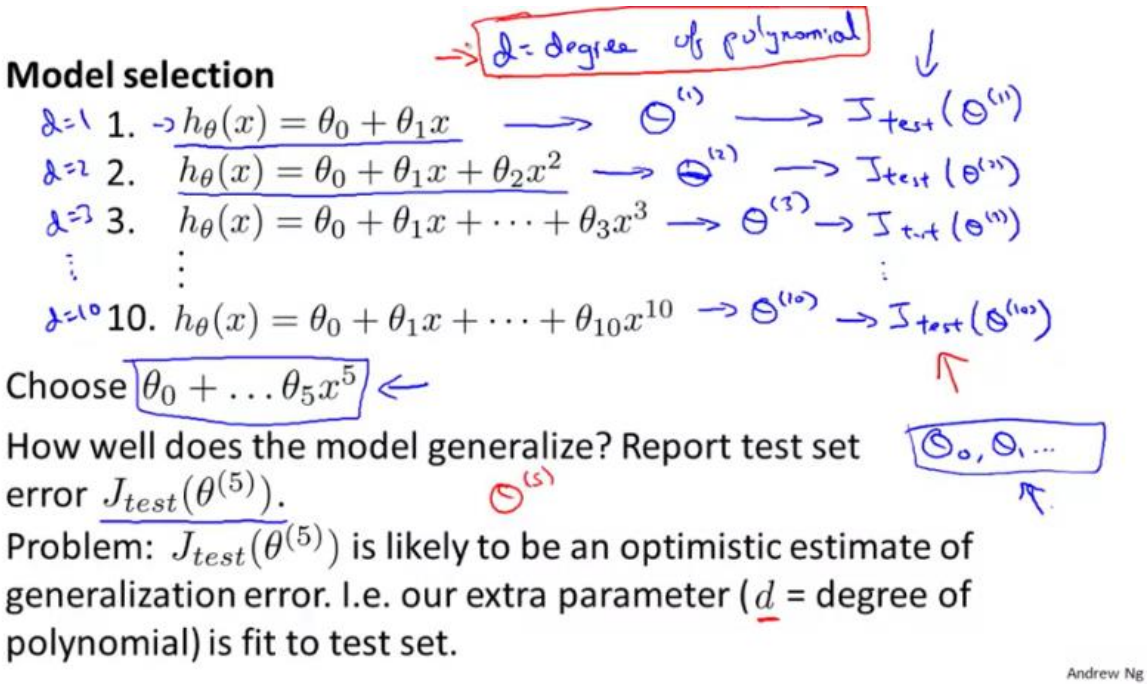
Misclassification error means that our hypothesis classified a value as 0 whereas it was actually 1 and vice versa.

Model selection and train/test/validation sets:

One way to build a model is that take 70% of the datapoints as training examples and 30% of them as test sets.

Now what we do with our training set is that we determine the value of parameters $theta_0$,$theta_1$,…..$theta_n$.

We can use the test set to determine the degree of the equation:



Let us assume that we get the least error in the test set when we take a degree 5 equation.

During this process , unknowingly we do overfitting and fit the model according to the test set. Hence , this model may not work on new data sets so accurately.

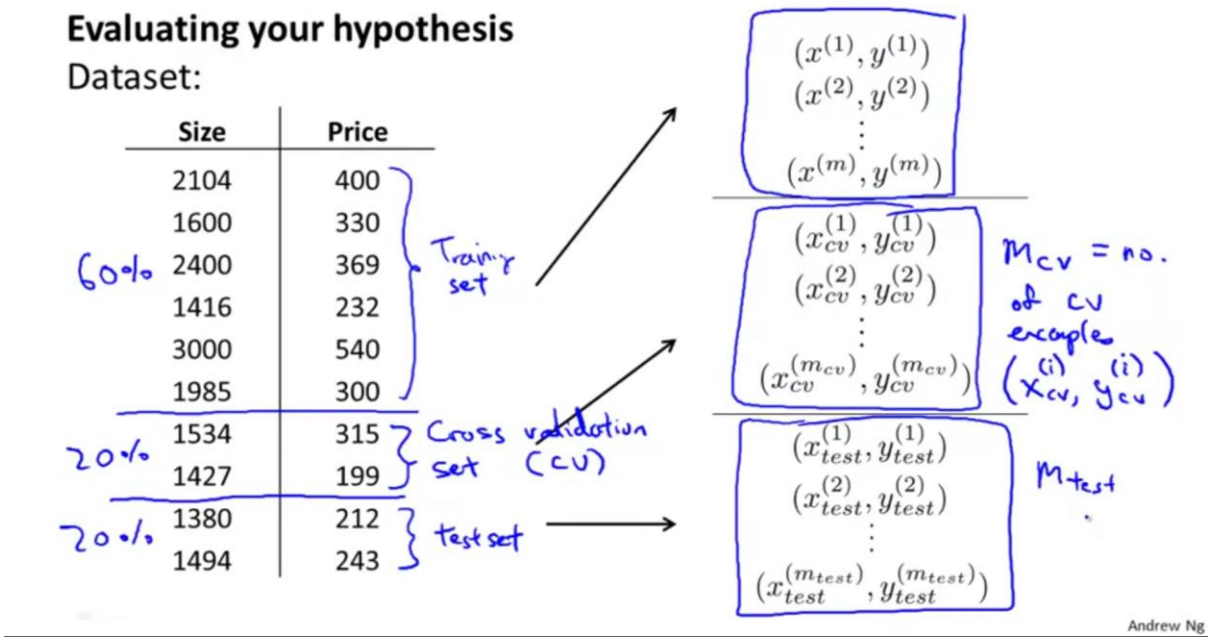
Solution of this problem:

We can divide our data into training set, validation set and testing set:

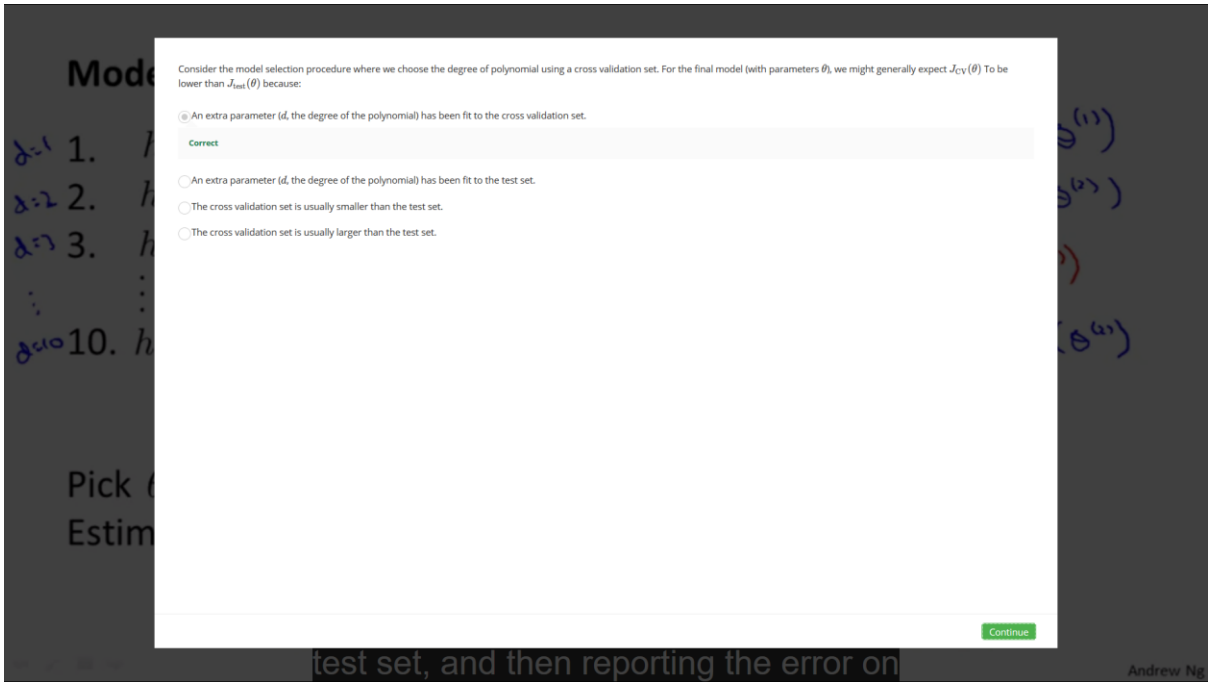
1.We determine theta values using the training set.

2. We determine the degree of the equation using validation set.

3.We test the accuracy and check for overfitting using the test set.



QUESTION:



Above question means that we fit our model on the basis of the cross-validation dataset and hence we have less error for our cross-validation dataset, this can also be attributed to some degree of overfitting. Hence, we have more error in our test set compared to the cross-validation set.

Coursera notes:

Model Selection and Train/Validation/Test Sets

Just because a learning algorithm fits a training set well, that does not mean it is a good hypothesis. It could over fit and as a result your predictions on the test set would be poor. The error of your hypothesis as measured on the data set with which you trained the parameters will be lower than the error on any other data set.

Given many models with different polynomial degrees, we can use a systematic approach to identify the 'best' function. In order to choose the model of your hypothesis, you can test each degree of polynomial and look at the error result.

One way to break down our dataset into the three sets is:

- Training set: 60%
- Cross validation set: 20%
- Test set: 20%

We can now calculate three separate error values for the three different sets using the following method:

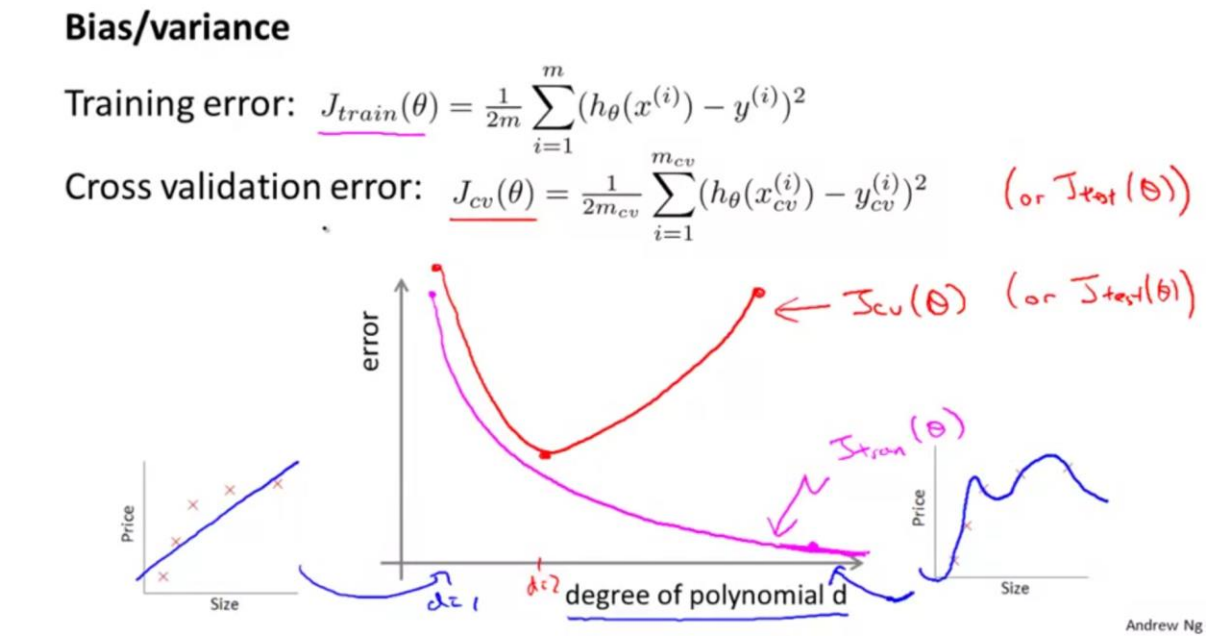
1. Optimize the parameters in Θ using the training set for each polynomial degree.
2. Find the polynomial degree d with the least error using the cross validation set.
3. Estimate the generalization error using the test set with $J_{test}(\Theta^{(d)})$, (d = theta from polynomial with lower error);

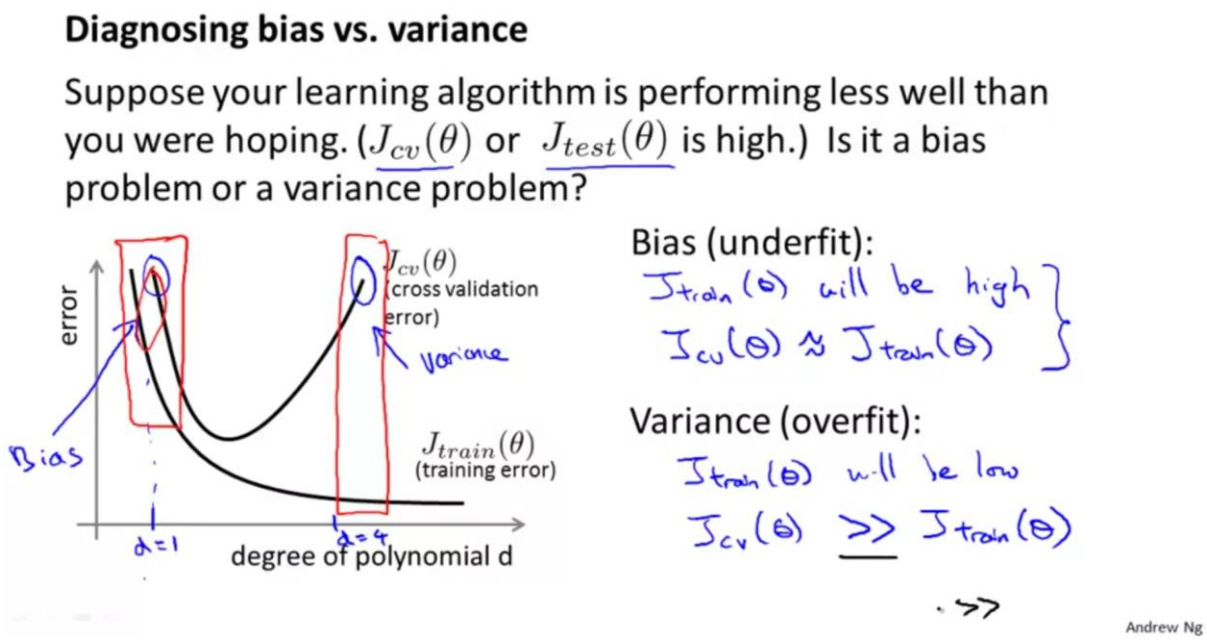
This way, the degree of the polynomial d has not been trained using the test set.

Diagnosing Bias vs. Variance

Now, our model may have high bias(underfitting) or high variance(overfitting).

To solve this problem we make a plot:





1.When we have bias due to underfitting, it is seen that the error for a given degree is almost the same for training and cross validation dataset.

2. When we have bias due to overfitting, it is seen that the error for a given degree is quite large for cross validation dataset as compared to the training dataset.

Coursera Notes:

Diagnosing Bias vs. Variance

In this section we examine the relationship between the degree of the polynomial d and the underfitting or overfitting of our hypothesis.

- We need to distinguish whether **bias** or **variance** is the problem contributing to bad predictions.
- High bias is underfitting and high variance is overfitting. Ideally, we need to find a golden mean between these two.

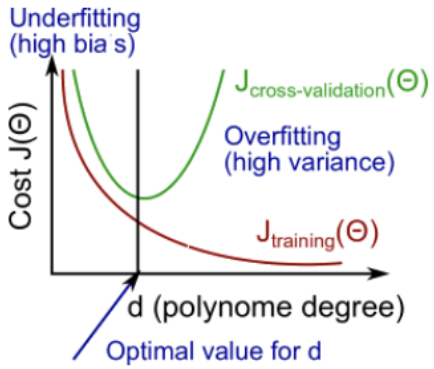
The training error will tend to **decrease** as we increase the degree d of the polynomial.

At the same time, the cross validation error will tend to **decrease** as we increase d up to a point, and then it will **increase** as d is increased, forming a convex curve.

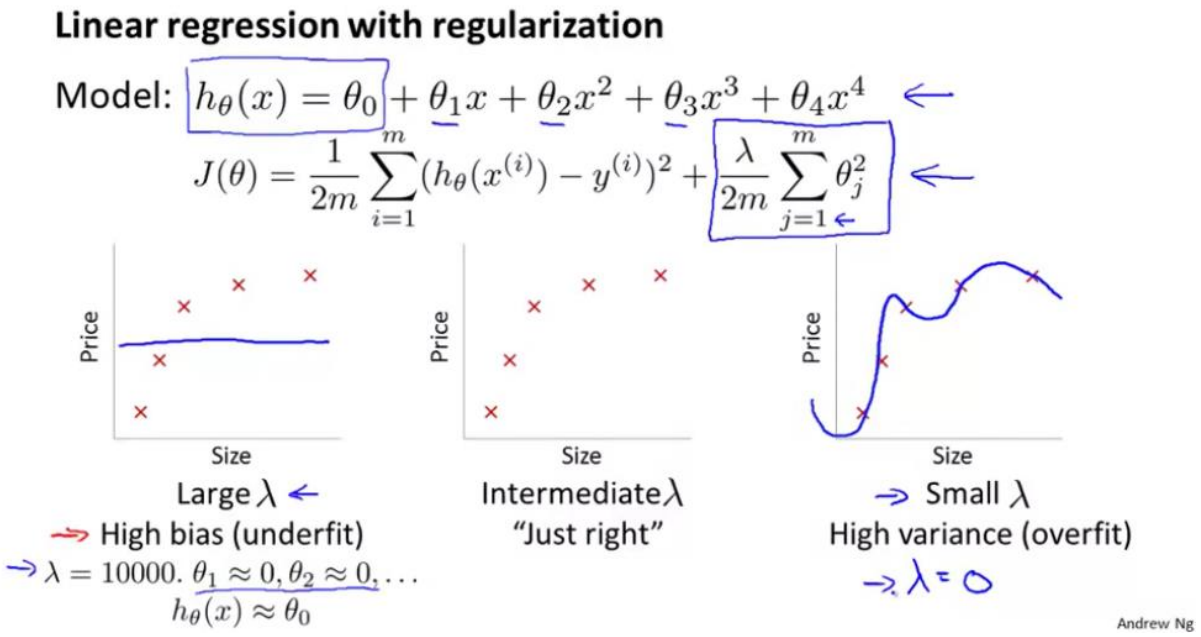
High bias (underfitting): both $J_{train}(\Theta)$ and $J_{CV}(\Theta)$ will be high. Also, $J_{CV}(\Theta) \approx J_{train}(\Theta)$.

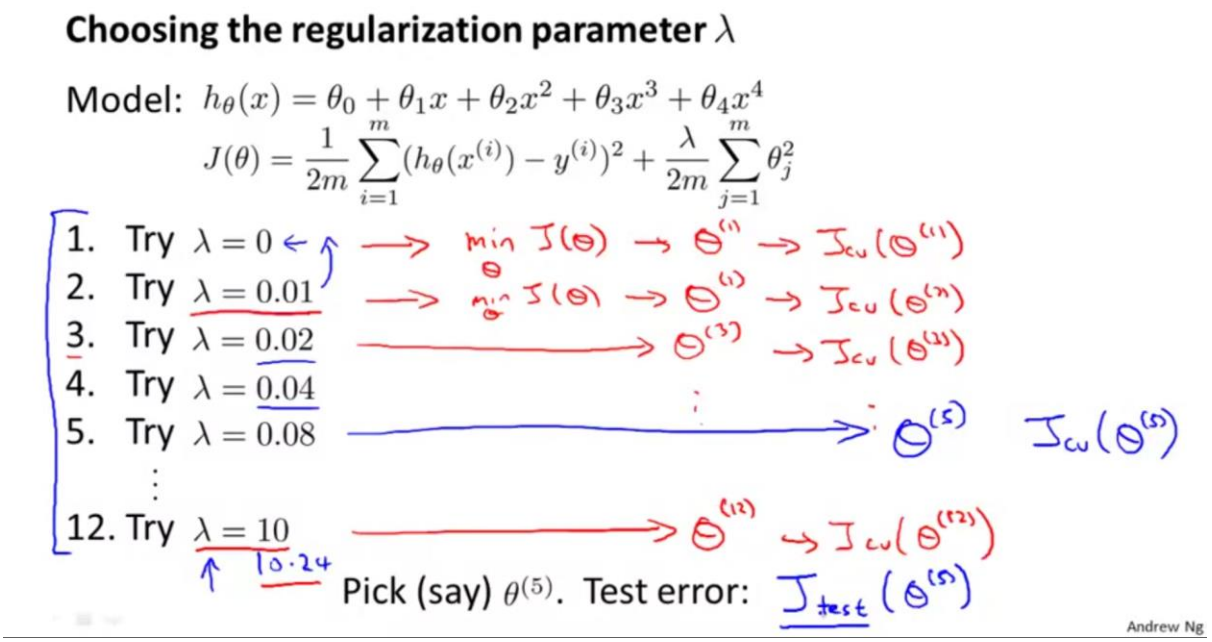
High variance (overfitting): $J_{train}(\Theta)$ will be low and $J_{CV}(\Theta)$ will be much greater than $J_{train}(\Theta)$.

The is summarized in the figure below:



Regularization and bias/variance:

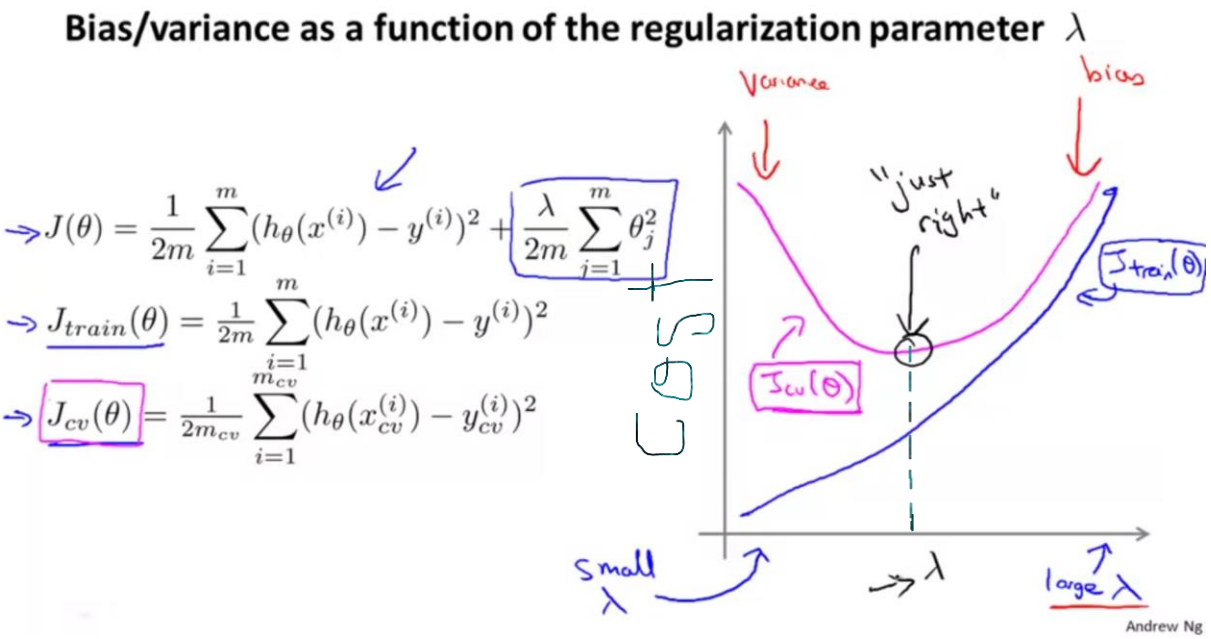




1.We can choose a certain value of lambda and then increase it in multiples of 2 and calculate value of cost function for the cross validation dataset each time.

2.Then we choose the best value of lambda(for which value of cost function is the least).

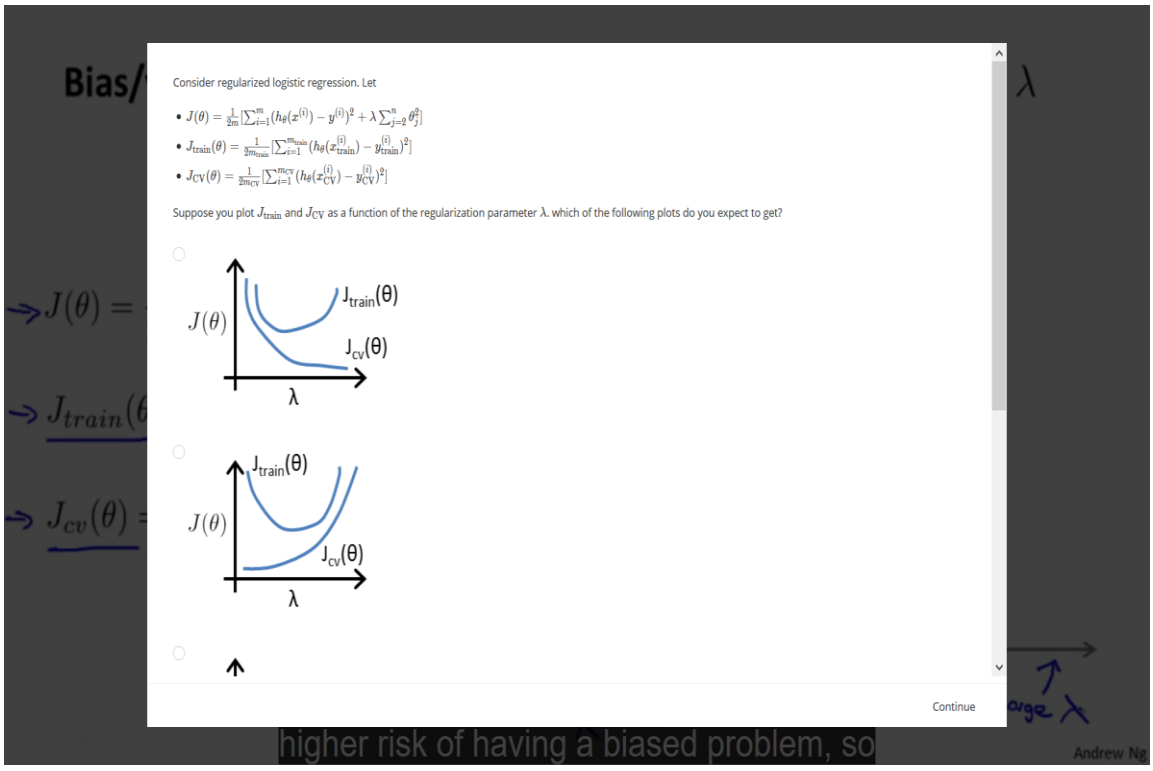
3. Then we calculate the cost for the test dataset using this chosen value of lambda.

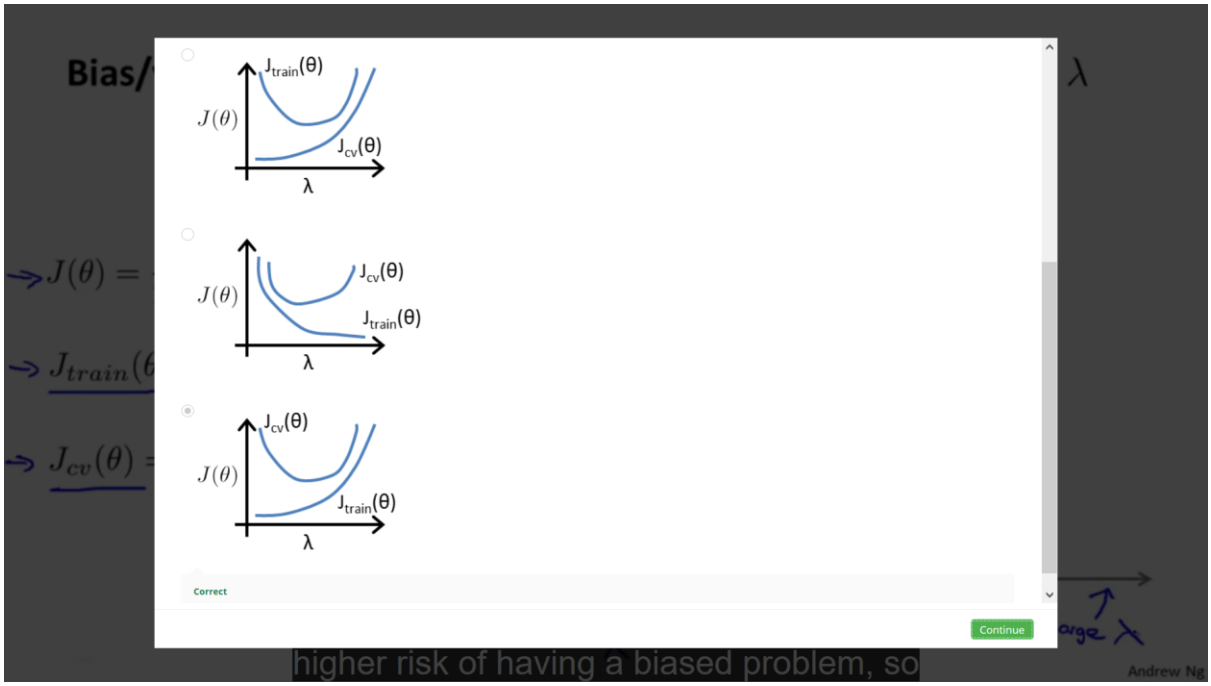


1.Larger values of lambda mean a greater tendency for underfitting(since when lambda is large, theta values get small.)

2. Smaller values of lambda mean a greater tendency for overfitting(since when lambda is small, theta values get significantly large.)

QUESTION:

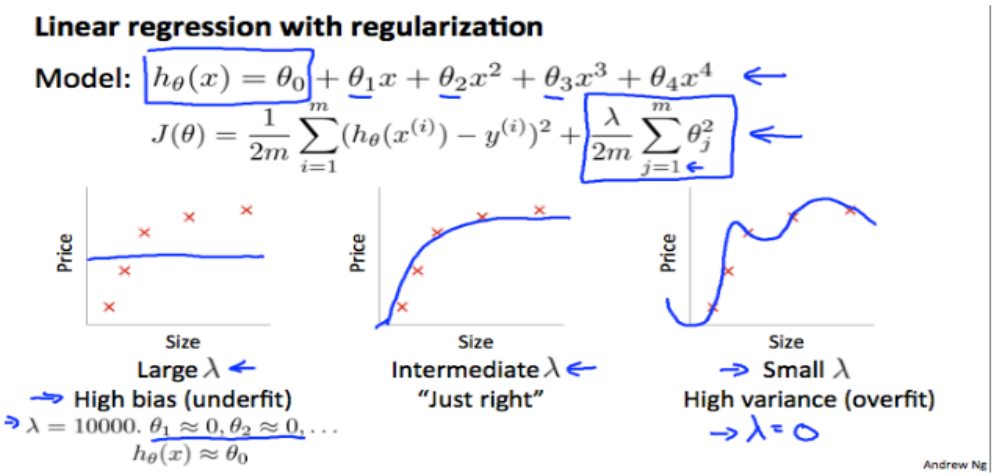




Coursera notes:

Regularization and Bias/Variance

Note: [The regularization term below and through out the video should be $\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$ and **NOT** $\frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$]

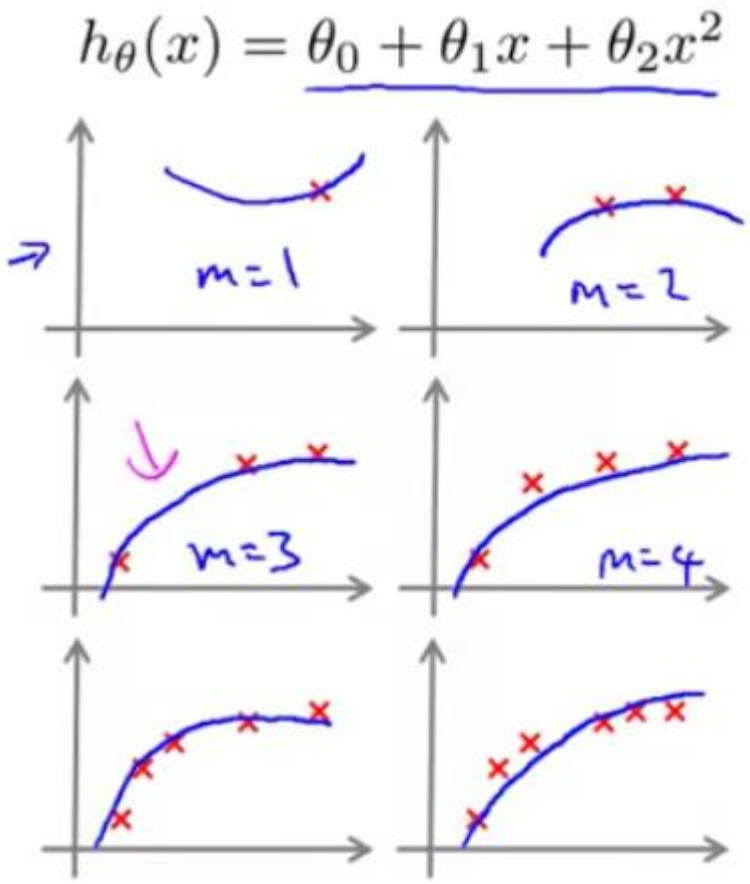


In the figure above, we see that as λ increases, our fit becomes more rigid. On the other hand, as λ approaches 0, we tend to over overfit the data. So how do we choose our parameter λ to get it 'just right' ? In order to choose the model and the regularization term λ, we need to:

1. Create a list of lambdas (i.e. $\lambda \in \{0, 0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.28, 2.56, 5.12, 10.24\}$);
2. Create a set of models with different degrees or any other variants.
3. Iterate through the λs and for each λ go through all the models to learn some Θ.
4. Compute the cross validation error using the learned Θ (computed with λ) on the $J_{cv}(\Theta)$ **without** regularization or $\lambda = 0$.
5. Select the best combo that produces the lowest error on the cross validation set.
6. Using the best combo Θ and λ, apply it on $J_{test}(\Theta)$ to see if it has a good generalization of the problem.

Learning Curves:

Learning curves give us the relationship between training set size and the error given by the cost function.

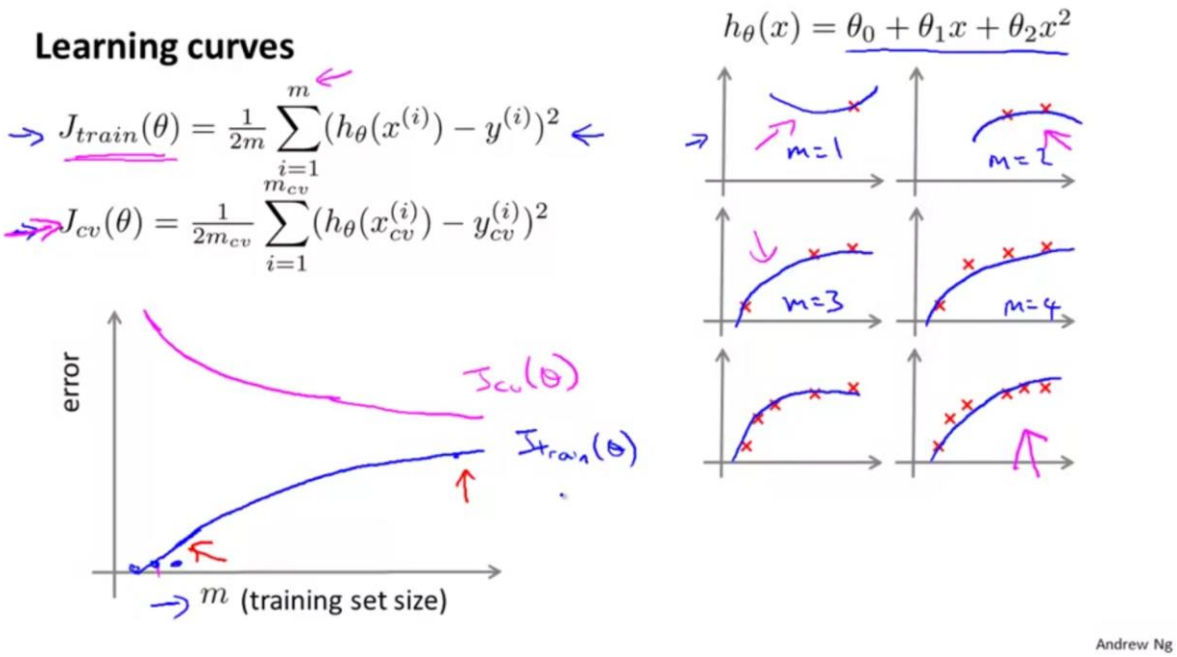


Above image tells us that:

1. When we have only one training example, we can fit a perfectly well quadratic curve.

2.When we have 2, 3 or 4 training examples, even then we can fit a quite well quadratic example.

3.Although as we keep on increasing the number of training examples, the data points don't fit the curve of quadratic equation very well.



In the above image, we observe:

1.As we keep on increasing the set size, the cross-validation error keeps on decreasing till it a certain point. Although, the opposite is true for training set error.

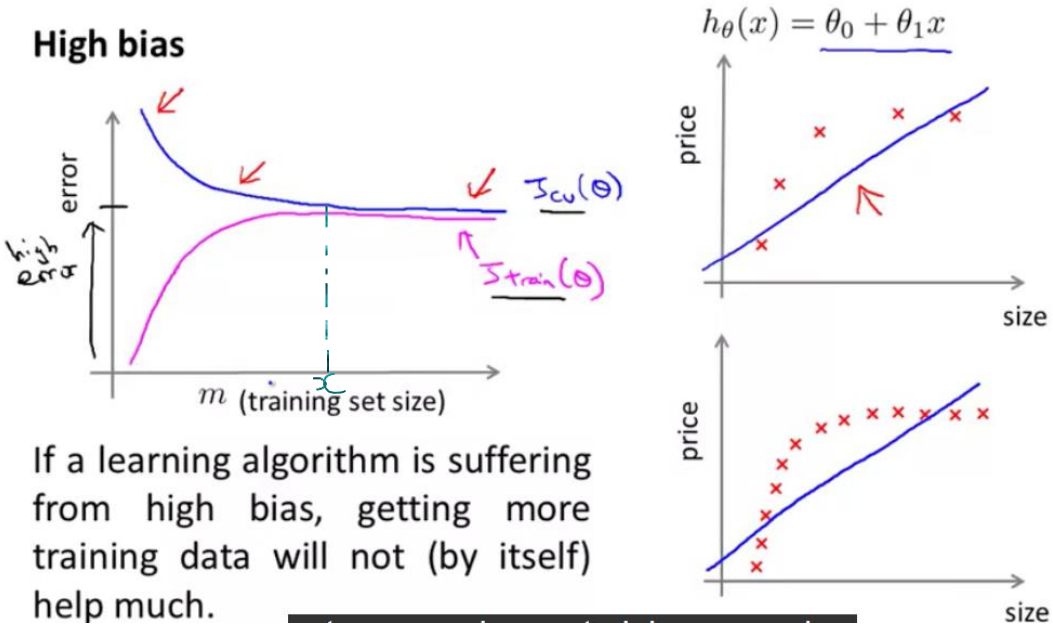
2.We know that the cross validation is error on the cross validation set that we haven't seen and so, when we have a very small training set, we do not generalize well, just not going to do well on that. So, right, this hypothesis here doesn't look like a good one, and it's only when we get a larger training set that, we get a hypothesis that may fit the data somewhat better. So the cross validation error and the test set error will tend to decrease as the training set size increases because the more data we have, the better we do at generalizing to new examples. So, just the more data we have, the better the hypothesis we can fit. So, if we plot J_{Train}, and J_{CV} the above graph is the sort of thing that we would get.

3. Hence we conclude cross validation error and test set error will tend to decrease as our training set size increases because the more data we have, the better we do at generalizing to new examples. So, just the more data we have, the better the hypothesis we can fit.

What the learning curves look like if we have either High Bias or High Variance problems :

CASE 1: Suppose the hypothesis has high bias.

We use an example, of fitting a straight line to data that, you know, can't really be fit well by a straight line.

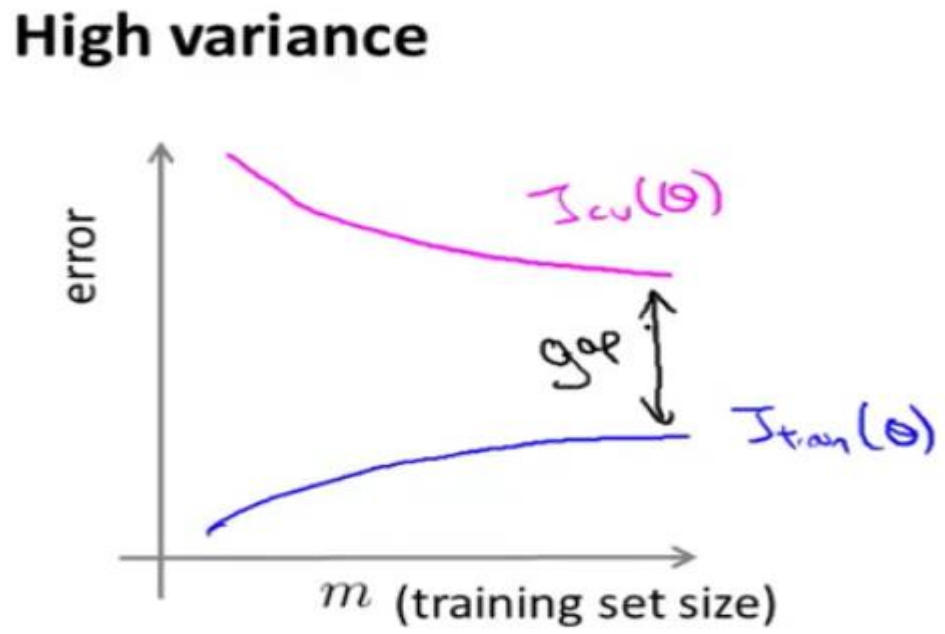


What we infer from the above figure:
1. On the right side of above figure we see that even as we increase the training set size, our hypothesis more or less remains the same. Hence , the error increases as we increase the training set size because now we are not able to fit more data points properly as compared to when we had less data points.

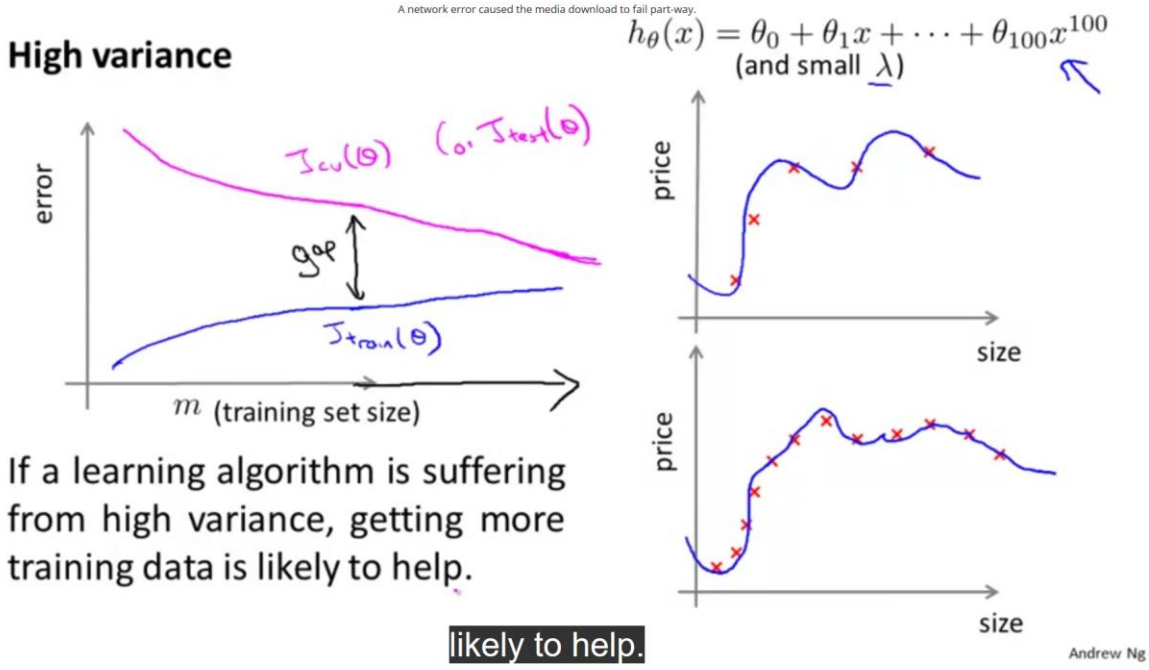
2.This shows that if we have high bias, increasing our training set size or cross validation set size would not help in decreasing the cross-validation/test set error after a certain amount of set sizes say x (shown in above figure).

CASE 2: Suppose the hypothesis has high variance.

We have a case of high variance when , we have a large gap between cross validation set error and training set error:



When we extend the J_{CV}(theta) and J_{Train}(theta) curves further, we observe that as we add more and more data points, the error of cross-validation data points decreases. Hence adding more training/cross-validation sets is beneficial in reducing cross validation error.s



1.On the right side we see that as we keep on adding more and more data points/training sets, we unable to fit the data perfectly well as compared to when we had less training sets.

2.In case of overfitting, we see that as we add more and more data points, the error of cross-validation data points decreases. Hence adding more training/cross-validation sets is beneficial in reducing cross validation error.

Question:

In which of the following circumstances is getting more training data likely to significantly help a learning algorithm's performance?

- [ ] Algorithm is suffering from high bias.

Un-selected is correct

- [x] Algorithm is suffering from high variance.

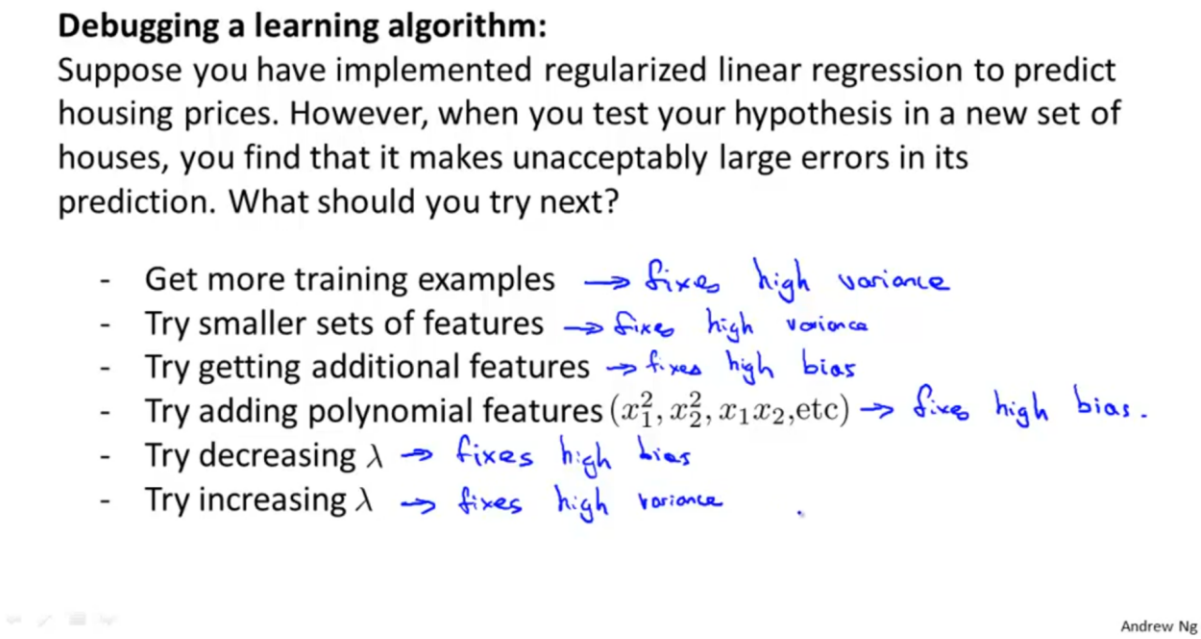
Correct

- [x] $J_{\text{CV}}(\theta)$ (cross validation error) is much larger than $J_{\text{train}}(\theta)$ (training error).

Correct

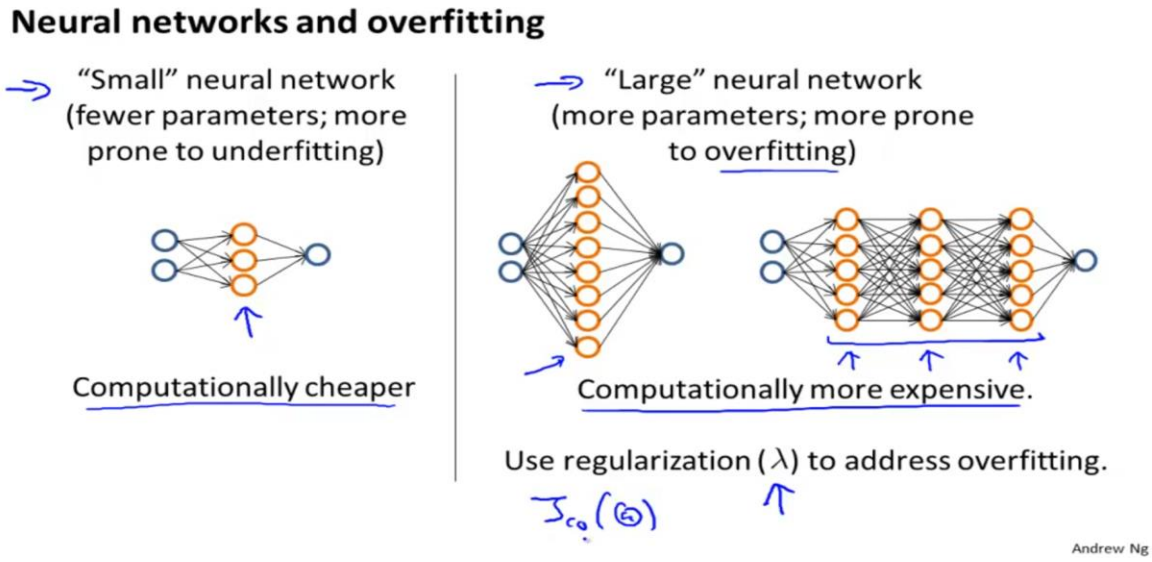
- [ ] $J_{\text{CV}}(\theta)$ (cross validation error) is about the same as $J_{\text{train}}(\theta)$ (training error).

Un-selected is correct



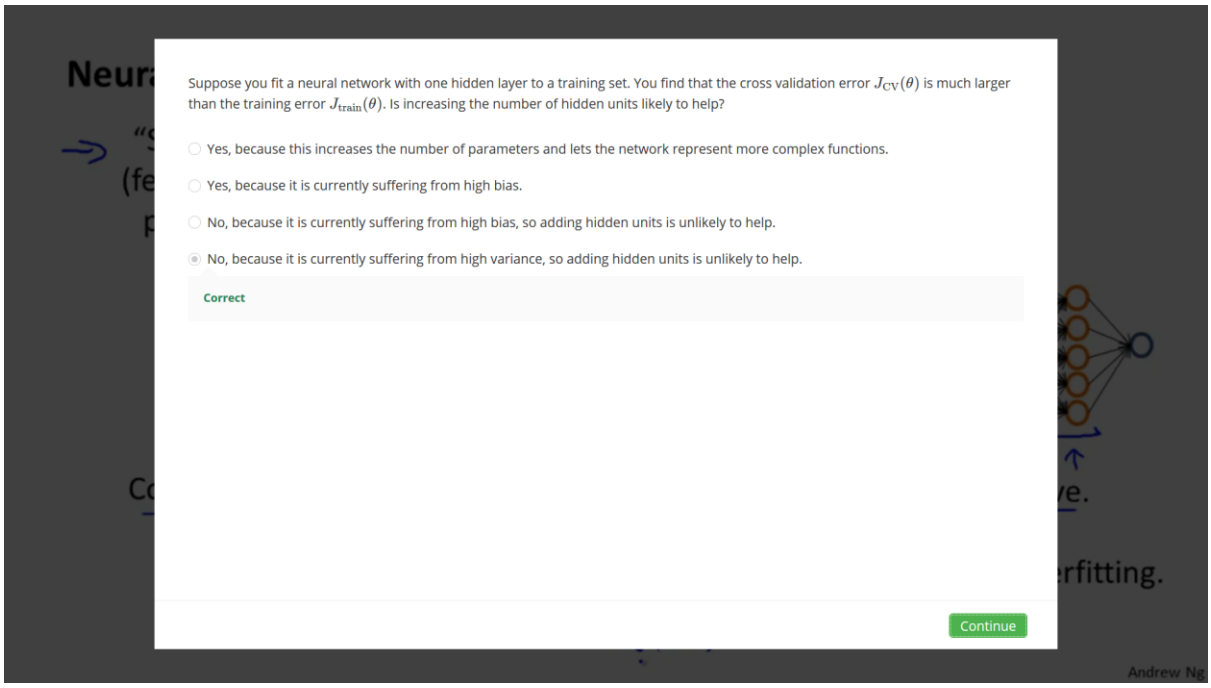
Explaining last two points:
1. Smaller values of lambda lead to overfitting and to solve this problem of overfitting or high variance, we increase the value of lambda.
2. Larger values of lambda lead to underfitting and to solve this problem of underfitting or high bias, we decrease the value of lambda.

Neural networks and overfitting:



Note: Using a large network with regularisation is often more beneficial than using a small network.

Question:

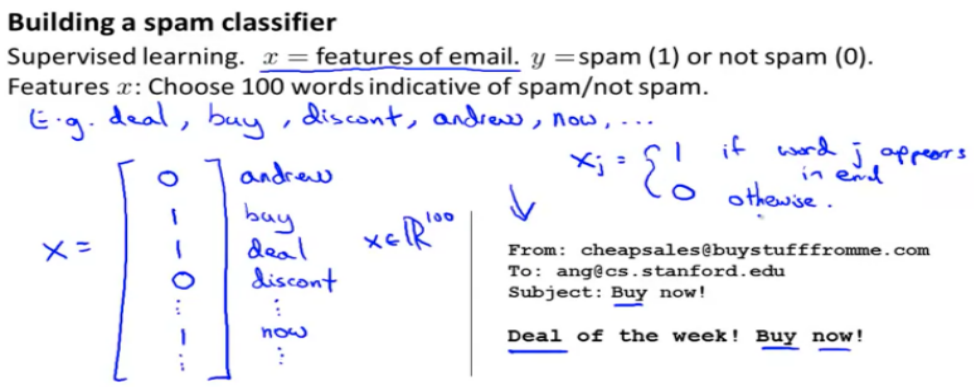


Note: Adding hidden units is not equivalent to adding more training examples. Adding more training examples means adding more rows in the data.

Prioritizing what to work on:

System Design Example:

Given a data set of emails, we could construct a vector for each email. Each entry in this vector represents a word. The vector normally contains 10,000 to 50,000 entries gathered by finding the most frequently used words in our data set. If a word is to be found in the email, we would assign its respective entry a 1, else if it is not found, that entry would be a 0. Once we have all our x vectors ready, we train our algorithm and finally, we could use it to classify if an email is a spam or not.

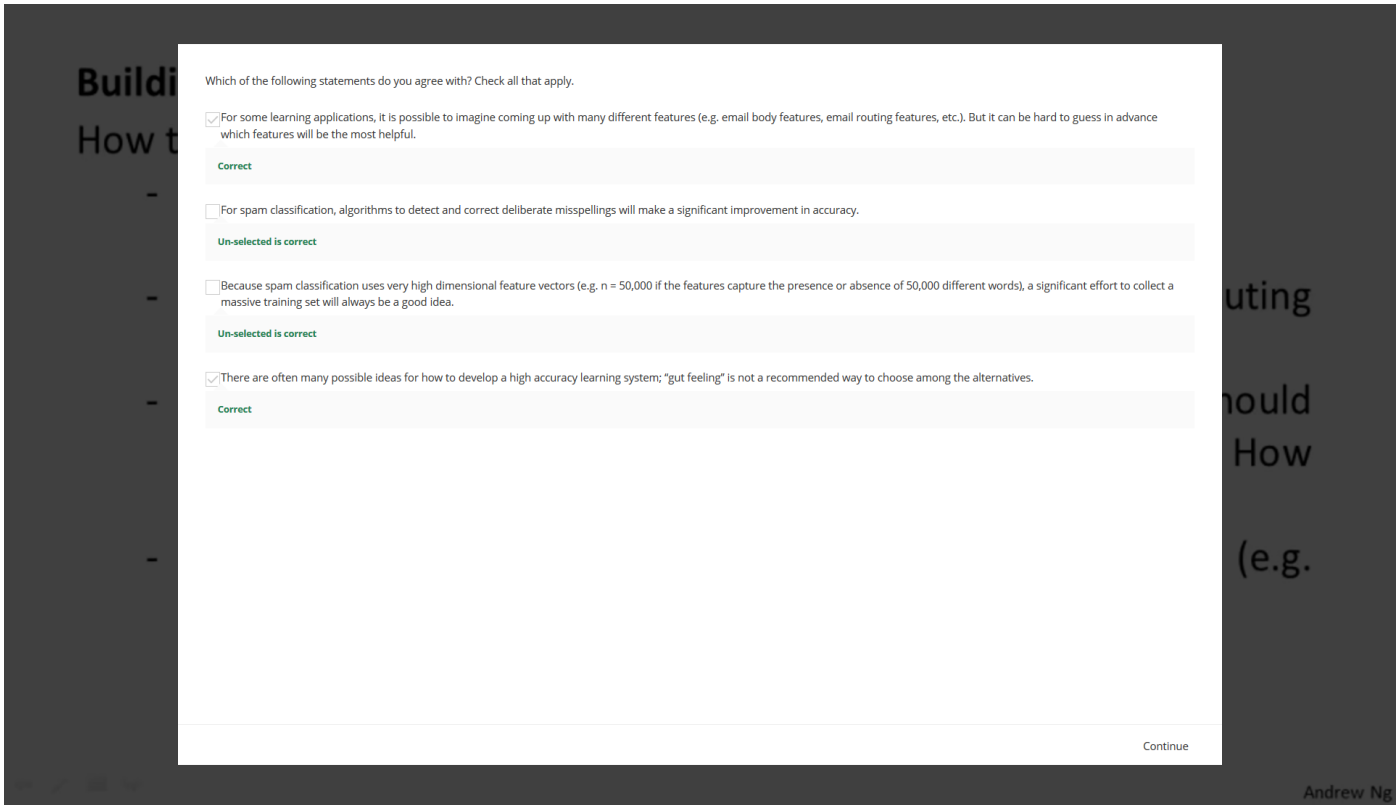


So how could you spend your time to improve the accuracy of this classifier?

- Collect lots of data (for example "honeypot" project but doesn't always work)
- Develop sophisticated features (for example: using email header data in spam emails)
- Develop algorithms to process your input in different ways (recognizing misspellings in spam).

It is difficult to tell which of the options will be most helpful.

Question:



ERROR ANALYSIS:

Error Analysis

The recommended approach to solving machine learning problems is to:

- Start with a simple algorithm, implement it quickly, and test it early on your cross validation data.
- Plot learning curves to decide if more data, more features, etc. are likely to help.
- Manually examine the errors on examples in the cross validation set and try to spot a trend where most of the errors were made.

For example, assume that we have 500 emails and our algorithm misclassifies a 100 of them. We could manually analyze the 100 emails and categorize them based on what type of emails they are. We could then try to come up with new cues and features that would help us classify these 100 emails correctly. Hence, if most of our misclassified emails are those which try to steal passwords, then we could find some features that are particular to those emails and add them to our model. We could also see how classifying each word according to its root changes our error rate:

The importance of numerical evaluation

Should discount/discounts/discounted/discounting be treated as the same word?
Can use "stemming" software (E.g. "Porter stemmer")
universe/university.
Error analysis may not be helpful for deciding if this is likely to improve performance. Only solution is to try it and see if it works.
Need numerical evaluation (e.g., cross validation error) of algorithm's performance with and without stemming.
Without stemming: 5% error With stemming: 3% error
Distinguish upper vs. lower case (Mom/mom): 3.2%

It is very important to get error results as a single, numerical value. Otherwise it is difficult to assess your algorithm's performance. For example if we use stemming, which is the process of treating the same word with different forms (fail/failing /failed) as one word (fail), and get a 3% error rate instead of 5%, then we should definitely add it to our model. However, if we try to distinguish between upper case and lower case letters and end up getting a 3.2% error rate instead of 3%, then we should avoid using this new feature. Hence, we should try new things, get a numerical value for our error rate, and based on our result decide whether we want to keep the new feature or not.

Error metrics for skewed classes:

Skewed class means that we have extreme values in our training set. Say, in a case where we classify cancer vs non cancer patients we only have 0.5% negative cancer cases. This means if we make an algorithm that always predicts negative cancer cases (a value of 0, in case of logistic regression), then we will have 99.5% accuracy.

Cancer classification example

Train logistic regression model $h_\theta(x)$. ($y = 1$ if cancer, $y = 0$ otherwise)

Find that you got 1% error on test set.

(99% correct diagnoses)

Only 0.50% of patients have cancer.

→ skewed classes.

```
function y = predictCancer(x)
  → y = 0; %ignore x!
return
```

→ 0.5% error

→ 99.2% accuracy (0.8% error)

→ 99.5% accuracy (0.5% error)

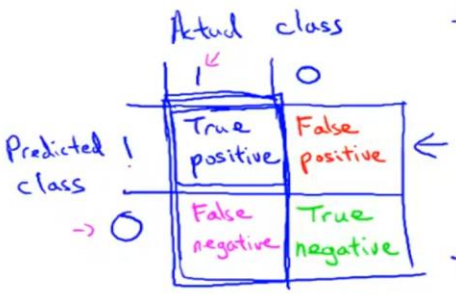
Suppose in another case, we get a training set accuracy of 99.2%. Although when we make an algorithm that always predicts say 0, our accuracy increases to 99.5%. Though the latter algorithm has less error, it is not right to classify everything as 0 just to reduce error rate.

Precision and recall:

To solve above problem, we use the concept of precision and recall:

Precision/Recall

$y = 1$ in presence of rare class that we want to detect



→ **Precision**
(Of all patients where we predicted $y = 1$, what fraction actually has cancer?)

$$\frac{\text{True positives}}{\#\text{predicted positive}} = \frac{\text{True positive}}{\text{True pos + False pos}}$$

→ **Recall**
(Of all patients that actually have cancer, what fraction did we correctly detect as having cancer?)

$$\frac{\text{True positives}}{\#\text{actual positives}} = \frac{\text{True positives}}{\text{True pos + False neg}}$$

$y = 0$

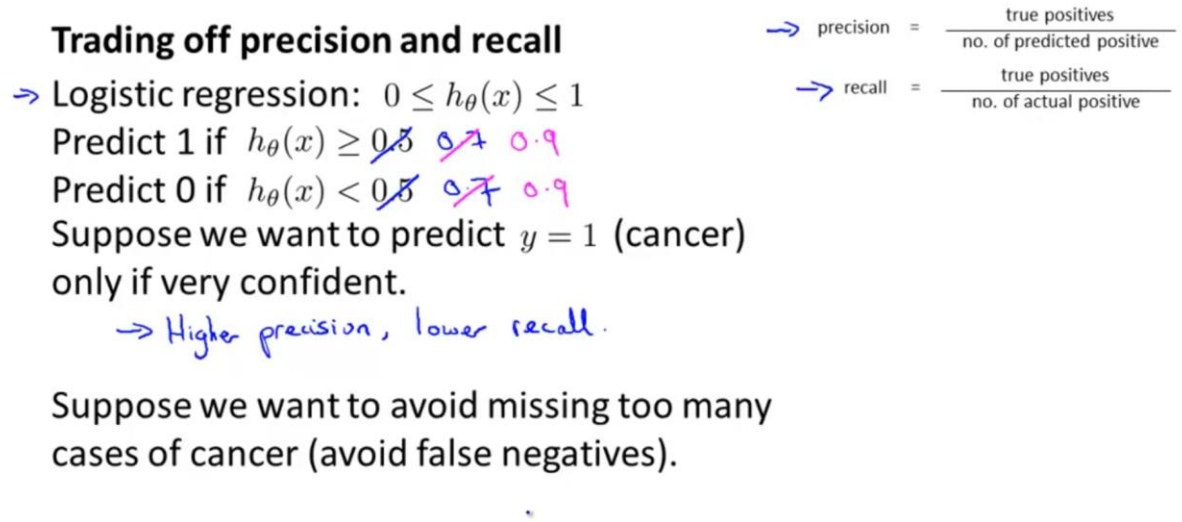
recall = 0

Andrew Ng

We usually prefer a high value of recall. If we set our predictors to always give y=0, then recall=0 which means our model is not good.

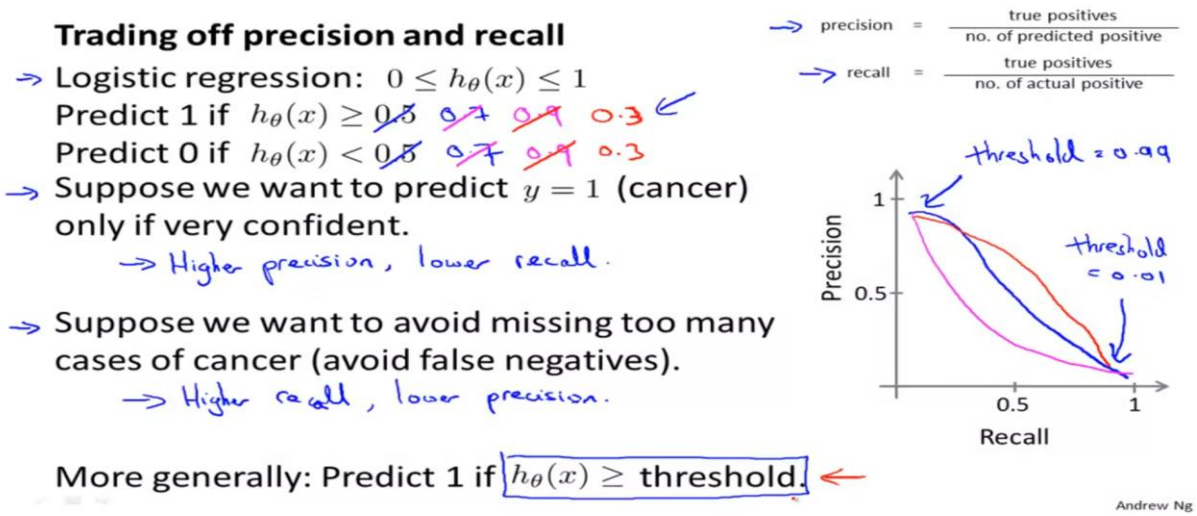
For more information on this topic refer to the topic ‘Key Quantities’ given on page 4 of machine learning basics file.

Trade off between precision and recall:



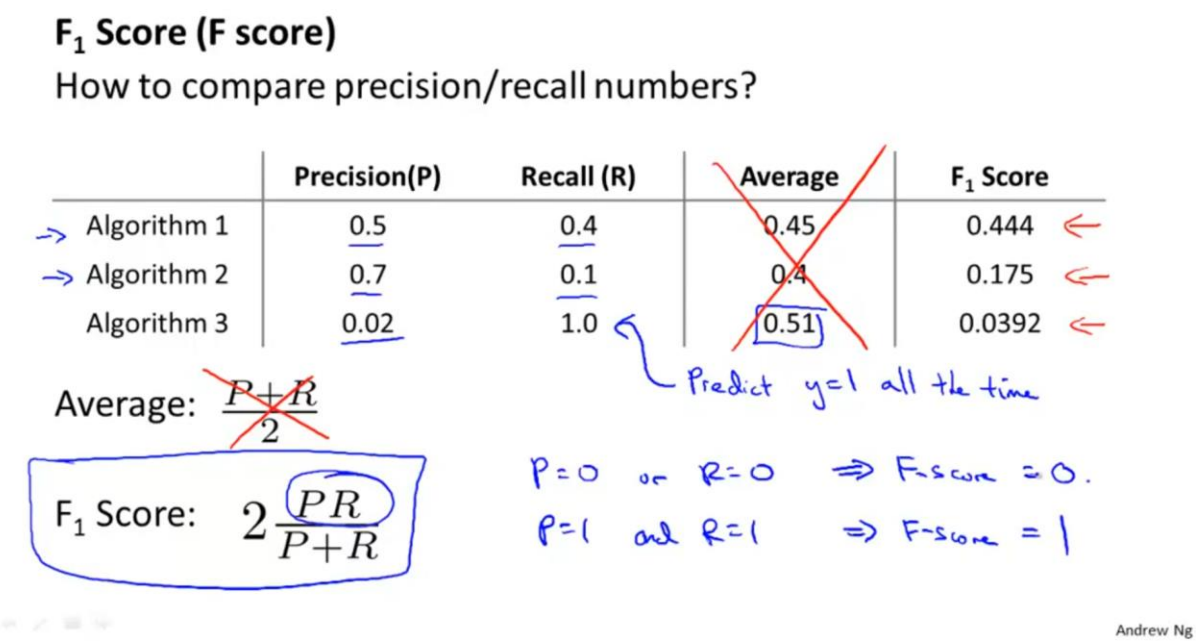
If we keep on increasing the threshold parameter, then only few of the cases that come in would be predicted to be positive for cancer. This would increase our precision, since no. of wrong predictions will go down due to ‘stricter’ threshold value say 0.7 and 0.9 instead of 0.5.

Now, suppose we want to avoid false negatives, then what we do is that we increase the recall value .



The three coloured lines shown in the graph in the above figure depict the three different relationships that can be there between precision and recall.

COMPARING ALGORITHMS HAVING DIFFERENT VALUES OF PRECISION AND RECALL:

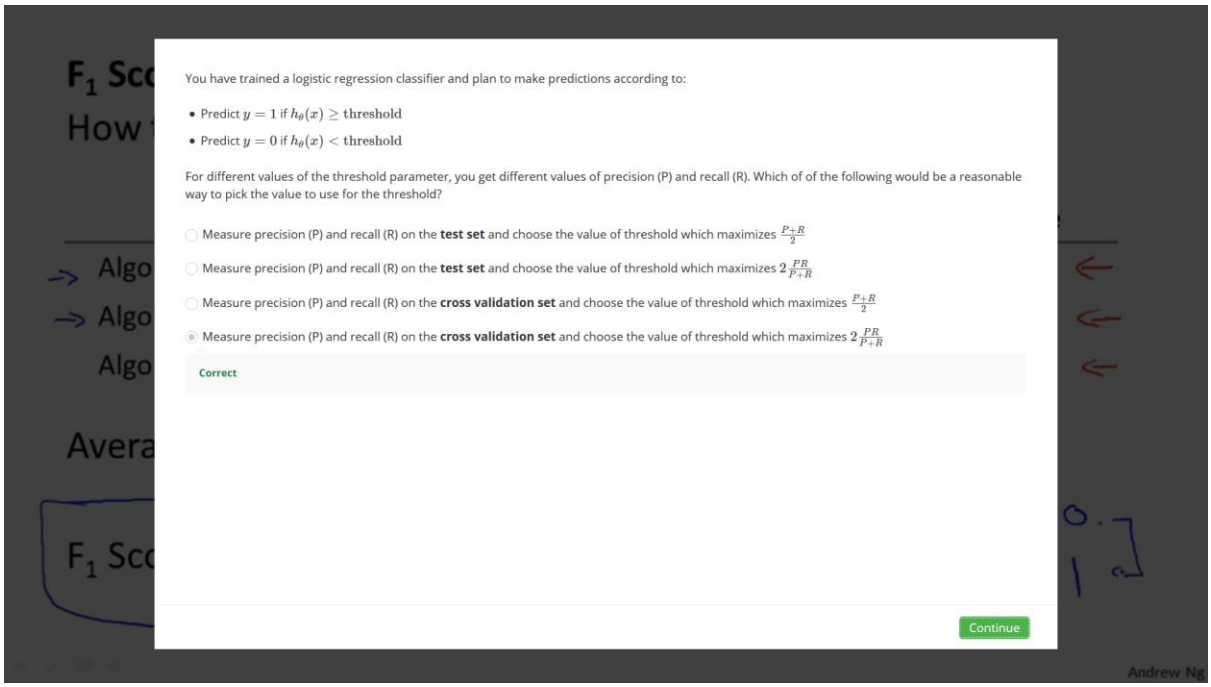


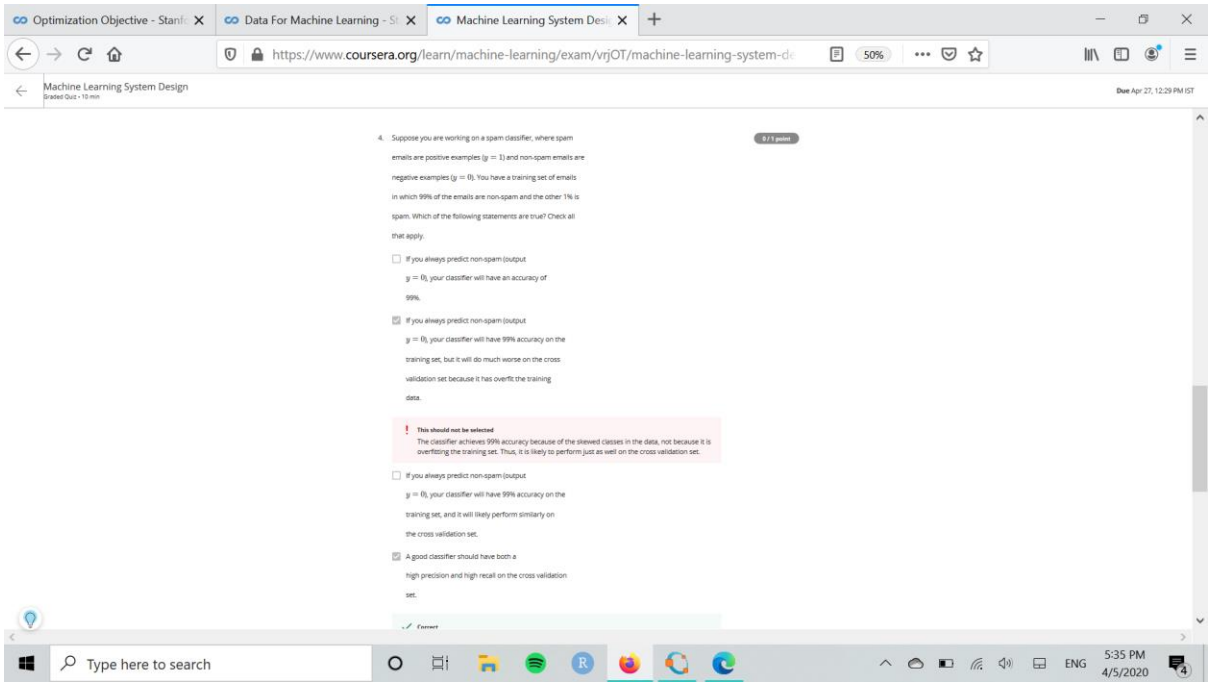
We do not take averages to compare different algorithms because as shown in algorithm 3, if we have skewed classes , then the average can misguide us.

F_1 score or F score solves the above inadequacy shown by averages method. This is because when we are calculating F score we take the product of precision and recall. If either of the values are low, the F_1 score becomes low.

We say if an algorithm has high value of F_1 score, it has a good balance of precision and recall values.

Questions:





Optimization Objective - Stanf | Data For Machine Learning - St | Machine Learning System Desi

https://www.coursera.org/learn/machine-learning/exam/vrjOT/machine-learning-system-design 60%

Machine Learning System Design
Graded Quiz • 10 min

Due Apr 27, 12:29 PM IST

Machine Learning System Design

TOTAL POINTS 5

1. You are working on a spam classification system using regularized logistic regression. "Spam" is a positive class (y = 1) and "not spam" is the negative class (y = 0). You have trained your classifier and there are m = 1000 examples in the cross-validation set. The chart of predicted class vs. actual class is: 1 point

	Actual Class: 1	Actual Class: 0
Predicted Class: 1	85	890
Predicted Class: 0	15	10

For reference:

- Accuracy = (true positives + true negatives) / (total examples)
- Precision = (true positives) / (true positives + false positives)
- Recall = (true positives) / (true positives + false negatives)
- F_1 score = (2 * precision * recall) / (precision + recall)

What is the classifier's recall (as a value from 0 to 1)?

Enter your answer in the box below. If necessary, provide at least two values after the decimal point.

0.85

2. Suppose a massive dataset is available for training a learning algorithm. Training on a lot of data is likely to give good performance when two of the following conditions hold true. 1 point

Which are the two?

Type here to search ENG 5:40 PM 4/5/2020

Optimization Objective - Stanf | Data For Machine Learning - St | Machine Learning System Desi

https://www.coursera.org/learn/machine-learning/exam/vrjOT/machine-learning-system-design 60%

Machine Learning System Design
Graded Quiz • 10 min

Due Apr 27, 12:29 PM IST

0.85

2. Suppose a massive dataset is available for training a learning algorithm. Training on a lot of data is likely to give good performance when two of the following conditions hold true. 1 point

Which are the two?

- [x] We train a learning algorithm with a large number of parameters (that is able to learn/represent fairly complex functions).
- [x] We train a model that does not use regularization.
- [ ] We train a learning algorithm with a small number of parameters (that is thus unlikely to overfit).
- [ ] The features x contain sufficient information to predict y accurately. (For example, one way to verify this is if a human expert on the domain can confidently predict y when given only x).

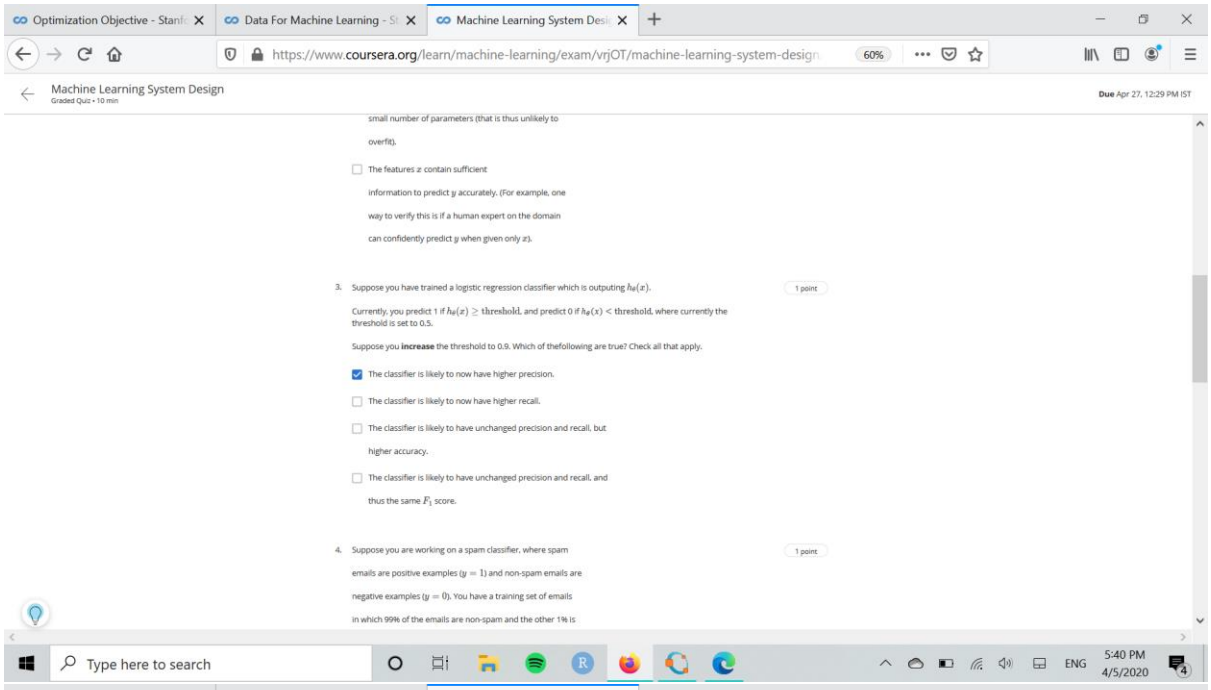
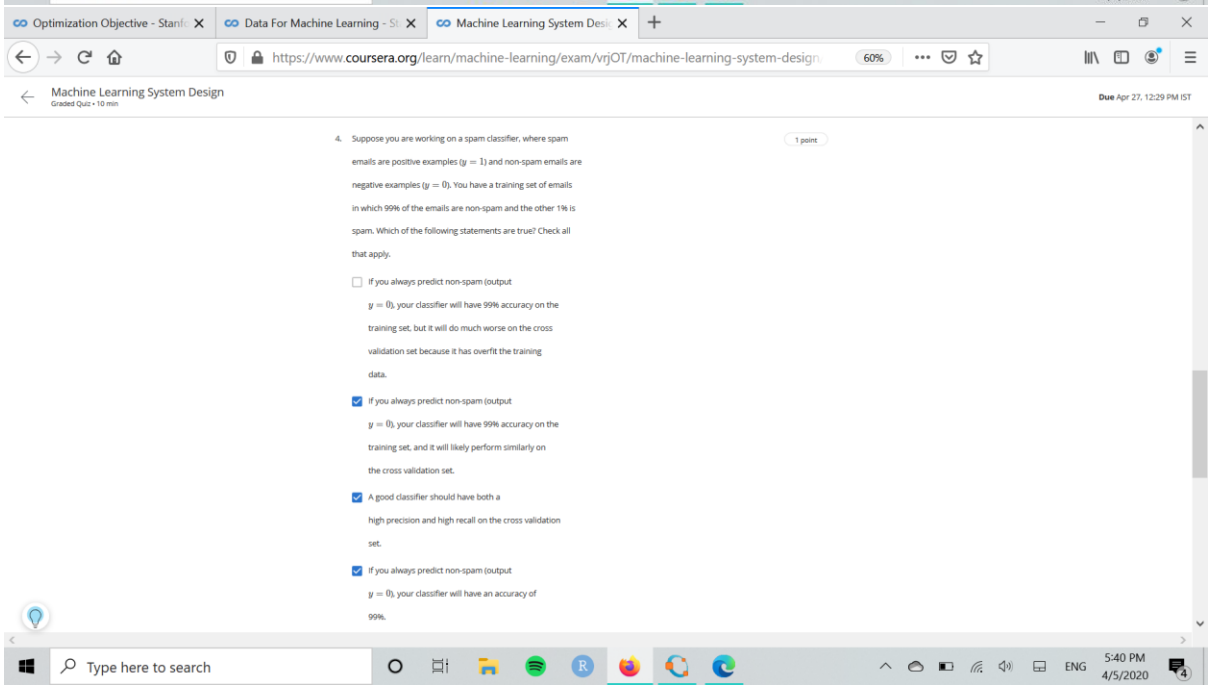
3. Suppose you have trained a logistic regression classifier which is outputing $h_\theta(x)$. 1 point

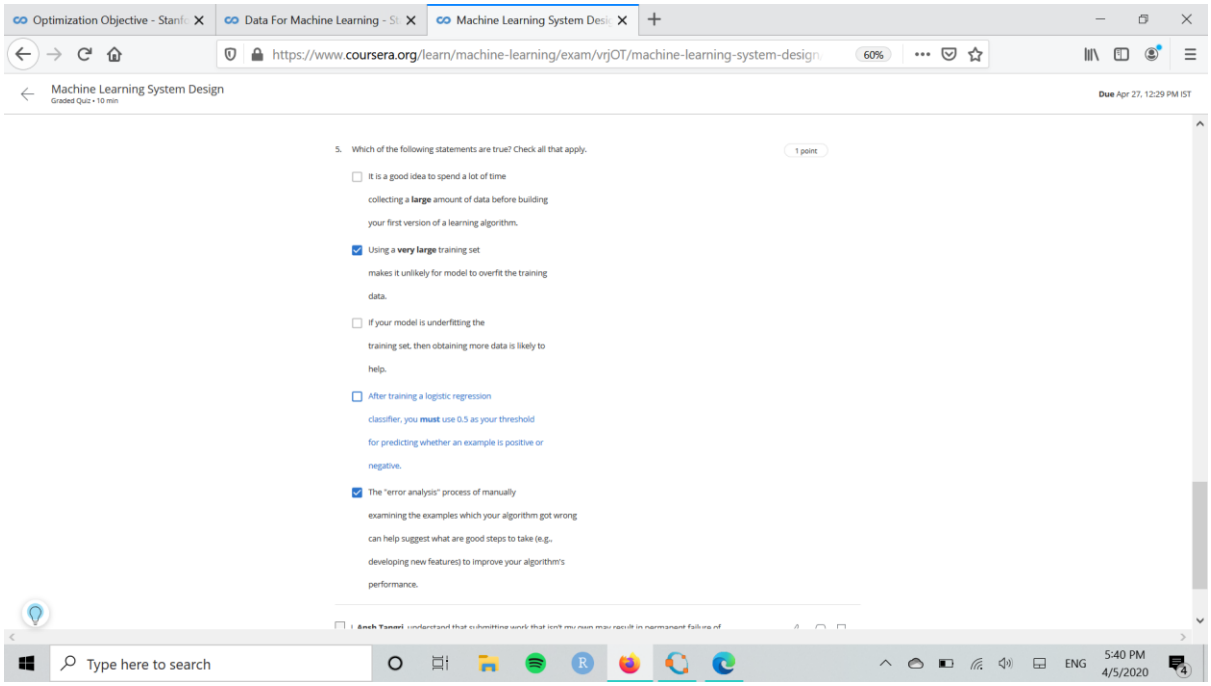
Currently, you predict 1 if $h_\theta(x) \geq$ threshold, and predict 0 if $h_\theta(x) <$ threshold, where currently the threshold is set to 0.5.

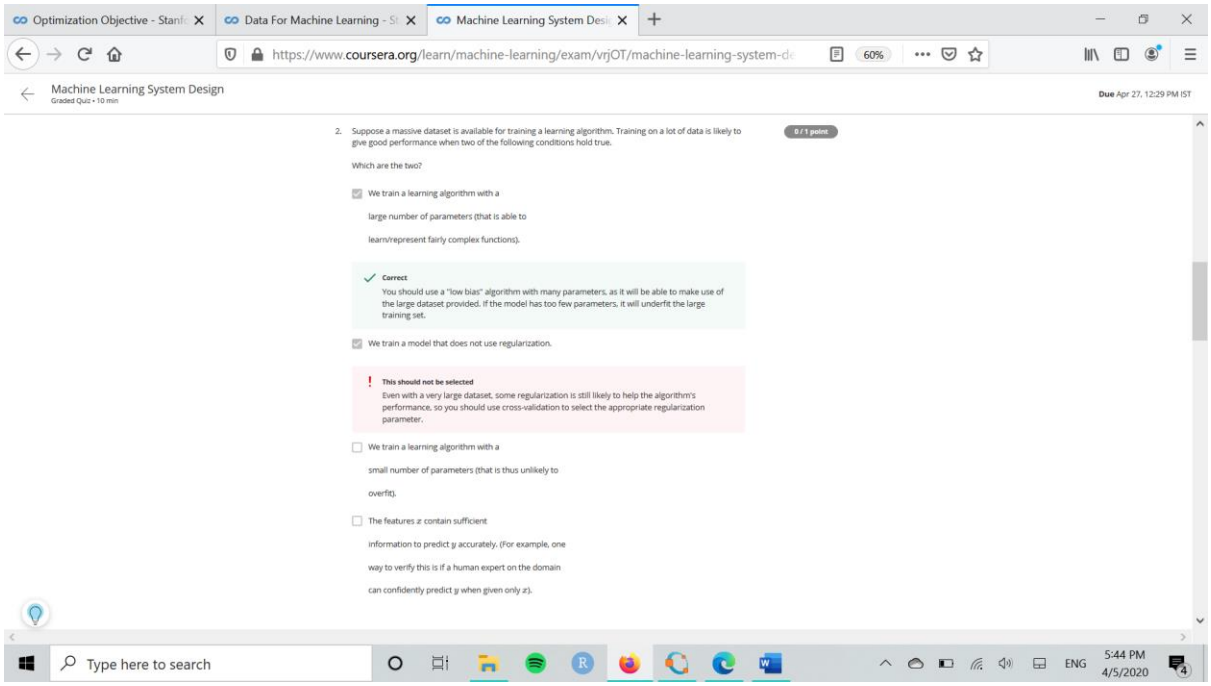
Suppose you **increase** the threshold to 0.9. Which of thefollowing are true? Check all that apply.

- [x] The classifier is likely to now have higher precision.

Type here to search ENG 5:40 PM 4/5/2020


Optimization Objective - Stanf
Data For Machine Learning - St
Machine Learning System Desi
https://www.coursera.org/learn/machine-learning/exam/vrjOT/machine-learning-system-design
60%
Machine Learning System Design
Graded Quiz • 10 min
Due Apr 27, 12:29 PM IST
small number of parameters (that is thus unlikely to overfit).
The features x contain sufficient information to predict y accurately. (For example, one way to verify this is if a human expert on the domain can confidently predict y when given only x).
3. Suppose you have trained a logistic regression classifier which is outputting $h_\theta(x)$. 1 point
Currently, you predict 1 if $h_\theta(x) \geq$ threshold, and predict 0 if $h_\theta(x) <$ threshold, where currently the threshold is set to 0.5.
Suppose you increase the threshold to 0.9. Which of thefollowing are true? Check all that apply.
The classifier is likely to now have higher precision.
The classifier is likely to now have higher recall.
The classifier is likely to have unchanged precision and recall, but higher accuracy.
The classifier is likely to have unchanged precision and recall, and thus the same F_1 score.
4. Suppose you are working on a spam classifier, where spam emails are positive examples ($y = 1$) and non-spam emails are negative examples ($y = 0$). You have a training set of emails in which 99% of the emails are non-spam and the other 1% is
1 point
Type here to search
ENG
5:40 PM
4/5/2020

Optimization Objective - Stanf
Data For Machine Learning - St
Machine Learning System Desi
https://www.coursera.org/learn/machine-learning/exam/vrjOT/machine-learning-system-design
60%
Machine Learning System Design
Graded Quiz • 10 min
Due Apr 27, 12:29 PM IST
4. Suppose you are working on a spam classifier, where spam emails are positive examples ($y = 1$) and non-spam emails are negative examples ($y = 0$). You have a training set of emails in which 99% of the emails are non-spam and the other 1% is spam. Which of the following statements are true? Check all that apply.
1 point
If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, but it will do much worse on the cross validation set because it has overfit the training data.
If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, and it will likely perform similarly on the cross validation set.
A good classifier should have both a high precision and high recall on the cross validation set.
If you always predict non-spam (output $y = 0$), your classifier will have an accuracy of 99%.
Type here to search
ENG
5:40 PM
4/5/2020


Optimization Objective - Stanf
Data For Machine Learning - St
Machine Learning System Desi
https://www.coursera.org/learn/machine-learning/exam/vrjOT/machine-learning-system-design
60%
Machine Learning System Design
Graded Quiz • 10 min
Due Apr 27, 12:29 PM IST
5. Which of the following statements are true? Check all that apply.
1 point
It is a good idea to spend a lot of time collecting a large amount of data before building your first version of a learning algorithm.
Using a very large training set makes it unlikely for model to overfit the training data.
If your model is underfitting the training set, then obtaining more data is likely to help.
After training a logistic regression classifier, you must use 0.5 as your threshold for predicting whether an example is positive or negative.
The "error analysis" process of manually examining the examples which your algorithm got wrong can help suggest what are good steps to take (e.g., developing new features) to improve your algorithm's performance.
Type here to search
ENG
5:40 PM
4/5/2020


Optimization Objective - Stanf
Data For Machine Learning - St
Machine Learning System Desi
https://www.coursera.org/learn/machine-learning/exam/vrjOT/machine-learning-system-de
60%
Machine Learning System Design
Graded Quiz • 10 min
Due Apr 27, 12:29 PM IST
2. Suppose a massive dataset is available for training a learning algorithm. Training on a lot of data is likely to give good performance when two of the following conditions hold true.
0 / 1 point
Which are the two?
We train a learning algorithm with a large number of parameters (that is able to learn/represent fairly complex functions).
Correct
You should use a "low bias" algorithm with many parameters, as it will be able to make use of the large dataset provided. If the model has too few parameters, it will underfit the large training set.
We train a model that does not use regularization.
This should not be selected
Even with a very large dataset, some regularization is still likely to help the algorithm's performance, so you should use cross-validation to select the appropriate regularization parameter.
We train a learning algorithm with a small number of parameters (that is thus unlikely to overfit).
The features x contain sufficient information to predict y accurately. (For example, one way to verify this is if a human expert on the domain can confidently predict y when given only x).
Type here to search
ENG
5:44 PM
4/5/2020