Evaluating a Learning Algorithm

Debugging a learning algorithm

When you test your hypothesis on a new set of data, you find that it makes unacceptably large error in its predictions. What should we try next?

– Get more training examples(some time this is very time consuming and actually no help for us) **(fixing high variance)**

– Try smaller sets of features **(fixes high variance)**

– Try getting additional features **(fixes high bias)**

– Try adding polynomial features (x12, x22 , x1x2,etc) **(fixes high bias)**

– Try decreasing lambda **(fixes high bias)**

– Try increasing lambda **(fixes high variance)**

Evaluating a Hypothesis

The method is that we could split the data into two separated part which the 70% of the data is our usual training set and the rest 30% of data is test set.

(Note: we should split the data randomly when we split the data. If the data is sorted randomly, we can split it frankly, but if it is not sorted randomly, we should shuffle it or resorted randomly first!)

The typical procedure of Training/testing for linear or logistic regression:

1. Learn parameters Theta from training data.

2. Compute test set error:

Jtest(Theta) using different Cost Function.

Another alternative method is **Misclassification error**

for logistic regression:

err(h(x), y) = 1 if h(x) >= 0.5, y =0 or if h(x) < 0.5 , y = 1(this means error)

0 otherwise(this means correct!)

then:

Test error = 1/mtest \* sum(err(h(x\_i), y\_i)(i from 1 up to mtest));

Train/validation/test error:

the data will be split into three parts training data, cross validation data, test data.

Training error:

Jtrain(Theta) = 1/2m \* sum((h(x\_i) – y\_i)^2)(i is from 1 up to Mtrain)

Cross validation error:

Jcv(Theta) = 1/2m \* sum((h(x\_i) – y\_i)^2)(i is from 1 up to Mcv)

Test error:

Jtest(Theta) = 1/2m \* sum((h(x\_i) – y\_i)^2)(i is from 1 up to Mtest)

When facing to select model, we also use cross validation set to select model instead test set.

After choose the model by validation set, we can evaluate it by the test set.

Diagnosing Bias vs. variance

If your hypothesis is suffering a bias problem(underfitting problem), the training error will be high and the validation error approximately equal to training error. In other words, the two type error are both too high.

If your hypothesis is suffering a variance problem(overfitting problem), the training error will be low and in contrast, the validation error will be more larger than training error.

Learning curves

Using this curve to diagnose if the learning algorithm suffering the bias problem or different variance problem or a little both.

If a learning algorithm is suffering from high bias, getting more training examples will not help much.

If a learning algorithm is suffering from high variance, getting more training data is likely to help.

In Neural Network, it turns out if you're applying neural network very often using a large neural network often it's actually the larger, the better, but if it's overfitting, you can then use regularization to address overfitting, usually using a larger neural network by using regularization to address the overfitting that's often more effective than using smaller neural network. And the main disadvantage is the more computationally expensive.

When we decide how many hidden layers in our neural network, we could use the cross-validation set. To compute the different Jcv(Theta) and select the best hidden layer.