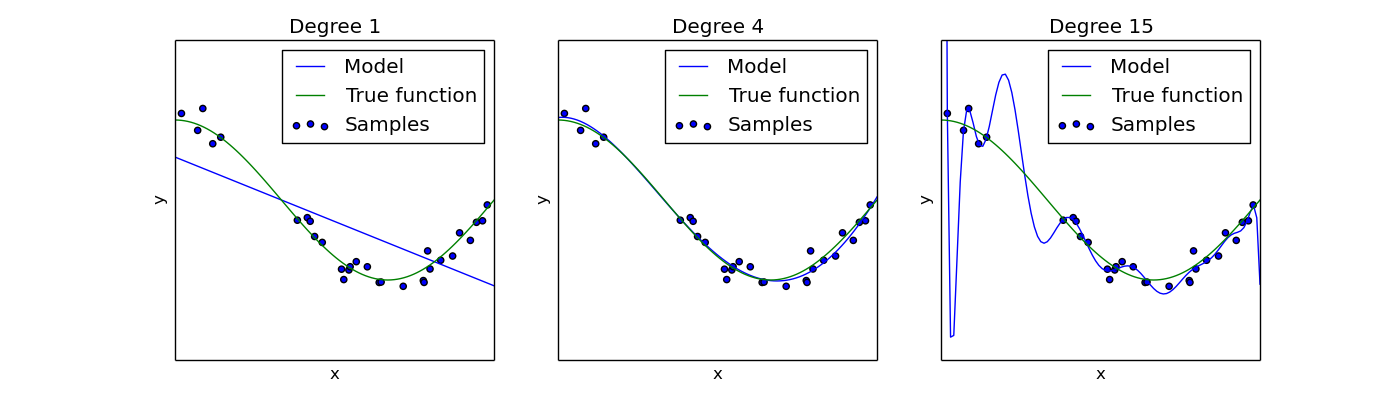
Neural Networks part 2: Overfitting

Machine Learning Mini Course

**Introduction to Underfitting and Overfitting**

Now we have this algorithm of Neural Networks. A Neural Network can be as complicated as you need it to be. Simply keep on adding extra and extra hidden layers, and you’ll get a more complex function. But sometimes this complexity isn’t good. Consider the folowing graph



The blue data points is the data distribution which we’re trying to predict. (Specifically, given x, we want to predict y.)

The first graph shows a linear regression model. Such a model doesn’t have enough degrees of freedom to match the data precisely. We call this **underfitting** or high bias. (High bias is the statistics term, I prefer underfitting, but some people will use “bias.”)

The second graph shows a neural network with a few layers. This model predicts the data well. (Yay!)

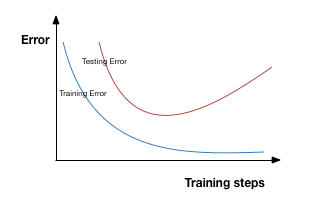
The final graph shows a neural network with many layers. This model predicts the data too well. This model won’t generalize to new data, thus is a bad predictor of future data. We call this **overfitting** or high variance. Again, “variance” is a term from statistics. The model has fit the training data **too well** so that it doesn’t generalize to new data.

**Training vs Test Datasets**

What can we do to ensure we don’t underfit or overfit? After all, the data usually can’t be plotted like it has been above. Most datasets will be at least ten dimensions, so human observation of overfitting vs underfitting is difficult.

Well, what if we checked if it generalized well? The idea is that models can only overfit on the data which we allow it to see. Thus, we simply break up our dataset into two parts: **training** and **testing** (also known as **validation**)[[1]](#footnote-0).

The training dataset is what the model has access to train on. The testing dataset is for checking the model’s performance.



Displayed above is the idea of how training and testing loss should look like. **An Optimal Model has low Testing Loss, not Training Loss.** Testing error determines how well a model generalizes, and that’s what’s useful in the end.

1. Technically, there’s a difference between testing and cross-validation datasets as well. However, I’m not going to go into that here; people are sloppy enough with their own wording that it doesn’t matter too much anyways. [↑](#footnote-ref-0)