Error Analysis

There is an old saying that you learn more from your mistakes than from your successes.

This is true in training models as well.

Hyper-focus on the Performance Metric, as is often the case, may mask subtle problems in the model.

By examining where the model succeeds and where it fails

- we may be able to improve the Performance Metric by adjusting the model to make correct predictions on previousy incorrect examples
- uncover undesirable properties of the model: bad predictions that are associated with a particular subset of examples
    - so overall Performance Metric may be good, but *always* wrong on important examples

The main message is that, to improve models, we must go deeper than summary statistics, such as the Performance Metric or Average Loss.

- We must examine errors on validation examples
    - determine if the errors are systematic
    - find ways to correct

Perhaps having answers to the following questions is *the most important* topic from this course

- How can you *improve* your models ?
- How can you tell if your model's predictions "make sense"
- How can I be most productive ?

Introduction

We glossed over Error Analysis in our first pass at the Classification task.

We motivate the need for Error Analysis using the MNIST digits classification task.

Let's visit the notebook Introduction (Error_Analysis.ipynb#Classification:-Beyond-accuracy)

Classification: Error Analysis

As we did with the Classification task

- We will start with Error Analysis for Binary Classification
- Move onto Error Analysis for Multinomial Classification

In addition to the all-encompassing Accuracy measure, we will examine several measures of Conditional Accuracy.

Let's visit the notebook section Error Analysis: Classification (Error_Analysis.ipynb#Classification:-Beyond-accuracy)

Balancing Precision and Recall

It turns out that there are competing goals for Classification that are hard to jointly satisfy:

- Correctly identify all Positive examples in a set (Recall)
- Don't mis-identify an example as Positive unless it really is Positive (Precision)

To see how they compete, consider

- Suppose we predict *all* examples to be Positive
    - We get perfect Recall, potentially by reducing Precision.
- Suppose we only predict Positive on the *single* example of which we are most certain
    - If we're right, we get perfect Precision, but with poor Recall.

Let's return to the notebook section Precision versus Recall (Error_Analysis.ipynb#Precison/Recall-Tradeoff) to explore the tradeoff.

Multinomial Classification: Error Analysis

Conditional Accuracy can be generalized to the Multionomial Classification task.

Let's return to the notebook section Multinomial Classification (Error_Analysis.ipynb#Multinomial-classification:-Confusion-matrix)

Regression: Error Analysis

We introduced Error Analysis for the Regression task in our module on the Recipe for ML

Let's return to the notebook section Regression (Error_Analysis.ipynb#Regression:-beyond-RMSE/R^2) for a quick review.

Hands-on: Beneath the covers of MNIST

Time to roll up your sleeves and get your hands dirty !

We will dig into the results of an MNIST classification model.

- We don't settle on a summary statistic
- We don't expect any "magical" conditional statistic to give us an immediate answer
- We show how to use a combination of intuition, tools, coding and hard work to understand why our model behaves as it does

In a Deeper Dive, we will visit the notebook (Error_Analysis_MNIST.ipynb)

In [1]:
```
print("Done")
```

```
Done
```