

Debugging Strategies

Sargur N. Srihari
srihari@cedar.buffalo.edu

Topics

- Overview
 1. Performance Metrics
 2. Default Baseline Models
 3. Determining whether to gather more data
 4. Selecting hyperparameters
 5. Debugging strategies
 6. Example: multi-digit number recognition

Difficulty of Debugging

- When a ML system performs poorly, difficult to tell whether poor performance is intrinsic to algorithm itself or whether there is an implementation bug
- Cannot tell a priori the behavior of algorithm
- Entire point of ML is that it will discover useful behavior we were not able to specify ourselves
- If classification test error rate is 5% we cannot tell whether this is expected behavior or suboptimal behavior

Multiple Adaptation Levels

- A difficulty is that ML models have multiple parts that are each adaptive
- If one part is broken, other parts can adapt and get acceptable performance

Debugging strategies

- Need to get around both difficulties:
 1. Whether performance is intrinsically poor or has a bug
 2. Whether parts are compensating for each other
- Design a test case that is so simple that the test behavior can be predicted, or
- Design a test that exercises one part of the neural net implementation in isolation

Important Debugging Tests

1. Visualize the model in action
2. Visualize the worst mistakes
3. Reasoning about software using train and test error
4. Fit a tiny dataset
5. Compare back-propagated derivatives to numerical derivatives
6. Monitor histograms of activations and gradient