

# Structuring ML Project

## Error Analysis

### What to examine?

#### Types of Errors



Multiple aspects simultaneously

Random

Systematic

require correction on the system

#### When to correct?

If offer significant improvement

Dev Set

Test Set

#### What to correct?

Same Distribution

Training Set

NN handles well

Too large for manual correction

Correct and Wrong predictions

Dev & Test Set

Metrics

Build 1st system quickly

Bias & Variance

Error

Define next step

### Strategy



## Dataset mismatch

### Mitigation

Training set vs Dev/Test set

Synthesise Data

Sub Topic

Collect more data : training set == Dev/Test set

Data distribution

Manual Error Analysis

Danger of small subset of whole problem space

Transfer learning

When to use?

Low level feature from Task A is helpful for Task B

Multi-tasks Learning

End-to-End Learning

Pros

Cons

Let data speaks

less hand crafting

large amount of data

exclude potentially useful hand crafted components

## Orthogonalisation

### Chain of Assumption

Early Stopping

Fit training set well

bigger network

optimizer

Fit dev set well

regularisation

Fit test set well

bigger training set

bigger dev set

Perform well in real world

change dev set

change metrics

## Setting Goal

### Metrics

Single Value Metric

Types of Metrics

Satisficing

Optimizing

When to change?

Rank of error

Weighted error metrics

Dataset Train/Dev/Test

Distribution

Dev == Test == RealWorld

Dev Set

Size

Just enough Dev/Test

Determine which classifier is better

## Performance



Bayesian Optimal error

Proxy to Bayesian Optimal Error

Human Level Performance

Human Error	0.01	0.07
Training Error	0.08	0.08
Dev Error	0.10	0.10

Avoidable Bias

Variance

Analysis

Avoidable Bias

Bigger Model

Better optimizer / train longer

Hyper-parameter

NN Architecture

Variance

More data

Regularization

Hyper-parameter

NN Architecture