

# orthogonalization

- each control does a specific task and doesn't affect other controls.

# single number evaluation metric

- confusion matrix: regression/recall
- F1 scores:  $F1 = 2 / ((1/P) + (1/R))$

# satisfying and optimizing metric

- facing many metrics, hard to judge which model is better.

Classifier	F1	Running time
A	90%	80 ms
B	92%	95 ms
C	92%	1,500 ms

- e.g.
  - choosing a single **optimizing** metric and decide that other metrics are **satisfying**.
    - Maximize F1 # optimizing metric
    - subject to running time < 100ms # satisficing metric

# train/dev/test sets distribution

- come from same distribution
- Choose dev set and test set to reflect data you expect to get in the future and consider important to do well on

# when to change dev/test sets and metrics

- analogy

1.25

## Orthogonalization for cat pictures: anti-porn

→ 1. So far we've only discussed how to define a metric to evaluate classifiers. ← Place target ⚡

→ 2. Worry separately about how to do well on this metric. ⚡

$$J = \frac{1}{\sum w^{(i)}} \sum_{i=1}^m w^{(i)} \ell(\hat{y}^{(i)}, y^{(i)})$$

↑ Aim (shoot at target)



o

- Figure out how to define a metric that captures what you want to do - place the target.
- Worry about how to actually do well on this metric - **how to aim/shoot accurately at the target.**

- so

## Another example

Algorithm A: 3% error

✓ Algorithm B: 5% error ←

→ Dev/test



→ User images



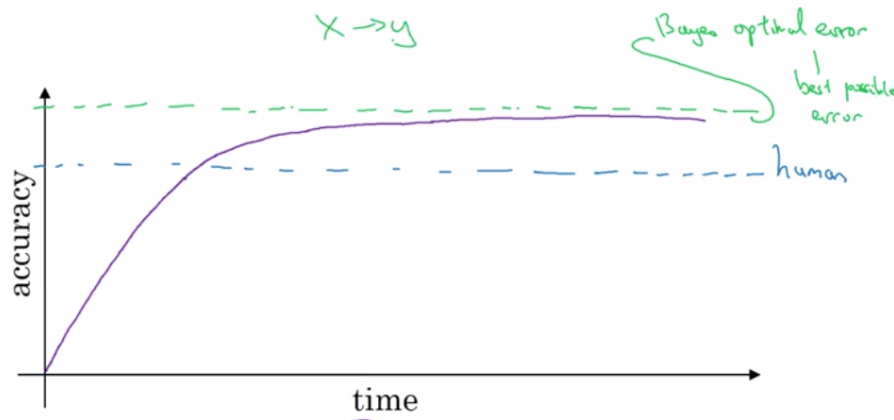
If doing well on your metric + dev/test set does not correspond to doing well on your application, change your metric and/or dev/test set.

Andrew Ng

## human-level performance

- human-level error can be used to estimate **bayes error**

## Comparing to human-level performance



- avoidable error
  - define as **Avoidable bias** = Training error - Human (Bayes) error
  - should be and can be reduce

## Why compare to human-level performance

Humans are quite good at a lot of tasks. So long as ML is worse than humans, you can:

- Get labeled data from humans.  $(x, y)$
- Gain insight from manual error analysis:  
Why did a person get this right?
- Better analysis of bias/variance.

## Improve model performance

- fundamental assumptions of supervised learning:
  - i. fit the training set pretty well (achieve low **avoidable bias**.)
  - ii. The training set performance generalizes pretty well to the dev/test set. ( **variance** is not too bad.)
- To improve your deep learning supervised system
  - i. Look at the difference between human level error and the training error - **avoidable bias**.
  - ii. Look at the difference between the dev/test set and training set error - **Variance**.
  - iii. If **avoidable bias** is large you have these options:
    - Train bigger model.
    - Train longer/better optimization algorithm (like Momentum, RMSprop, Adam).
    - Find better NN architecture/hyperparameters search.
  - iv. If **variance** is large you have these options:
    - Get more training data.
    - Regularization (L2, Dropout, data augmentation).
    - Find better NN architecture/hyperparameters search.