DEEP LEARNING BOOTCAMP

Introduction to ML strategy

Kunwoo Park

Oct 10 2018

0. Table of Contents

1. Intoduction to ML strategy

Why ML strategy

Motivating example

90%

Ideas:

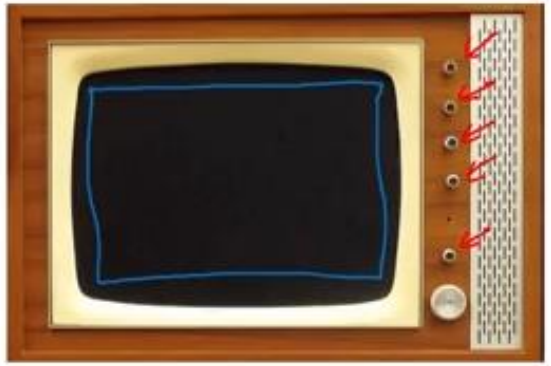
- Collect more data
- Collect more diverse training set
- Train algorithm longer with gradient descent
- Try Adam instead of gradient descent
- Try bigger network
- Try smaller network
- Try dropout
- Add L_2 regularization
- Network architecture
  - Activation functions
  - # hidden units
  - …

Andrew Ng

- How to improve your algorithm effectively? Use ML strategy!!!

Orthogonalization

TV tuning example



Orthogonalization

0.1 x

+ 0.5 x

- 1.7 x

+ 0.8 x

+ ...

Car

→ Steering

→ {Acceleration, Braking}

→ 0.3 x angle − 0.8 speed

→ 2 x angle + 0.9 speed.

Andrew Ng

Orthogonalization

Chain of assumptions in ML

→ Fit training set well on cost function

(≈ human-level performance)

bigger network
Adam
...

early stopping

→ Fit dev set well on cost function

Regularization
Bigger training set

→ Fit test set well on cost function

Bigger dev set

→ Performs well in real world

(Happy customers / app users.)

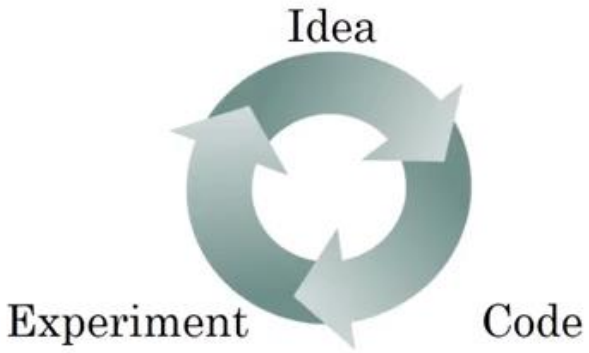
Change dev set or cost function

Andrew Ng

2. Setting up your goal

Single number evaluation metric

Using a single number evaluation metric



Classifier	Precision	Recall	F1 Score
A	95%	90%	92.4%
B	98%	85%	91.0%

$$\frac{2}{\frac{1}{P}+\frac{1}{R}}$$

Precision: of the examples that your classifier recognizes as cats, What percentage actually are cats?
TP/(TP+FP)

Recall (sensitivity): of all the images that really are cats, what percentage were correctly recognized by your classifier? TP/(TP+FN)

Satisficing and optimizing metric

Another cat classification example

optimizing ↓ ↓ satisficing

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

Cost = accuracy − 0.5 × running Time

Maximize accuracy

subject to running Time ≤ 100 ms.

N metrics: 1 optimizing

N−1 satisficing

Train/dev/test distributions

Cat classification dev/test sets

development set, hold out cross validation set

Regions:

- US
- UK
- Other Europe
- South America
- India
- China
- Other Asia
- Australia

Dev ←

Test ←

Randomly shuffle into dev/test

dev set + metric

Andrew Ng

Train/dev/test distributions

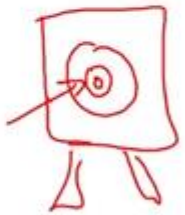
True story (details changed)

Optimizing on dev set on loan approvals for medium income zip codes

↑ x → y (repay loan?)

Tested on low income zip codes

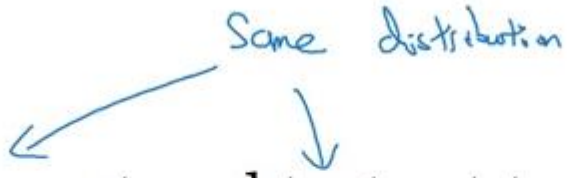
~3 month



Andrew Ng

Train/dev/test distributions

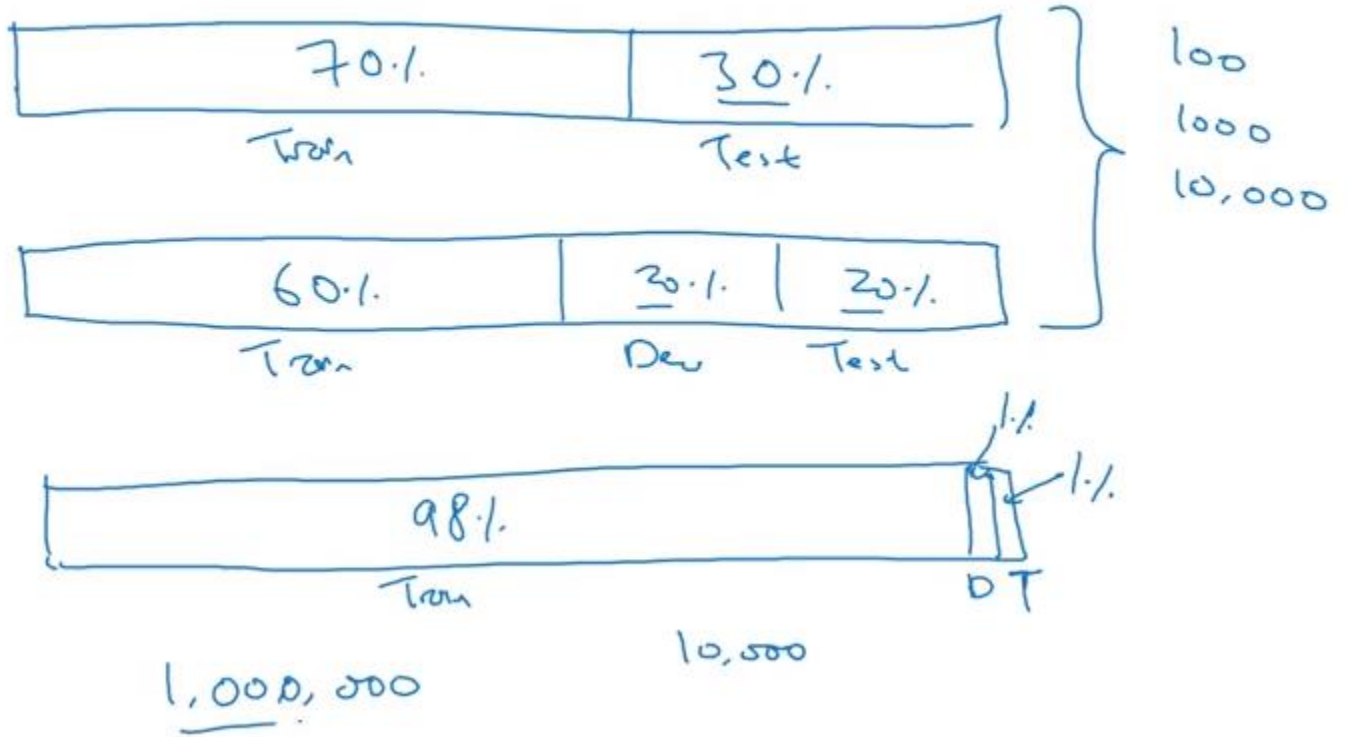
Guideline



Choose a dev set and test set to reflect data you expect to get in the future and consider important to do well on.

Size of dev and test sets

Old way of splitting data



Size of dev and test sets

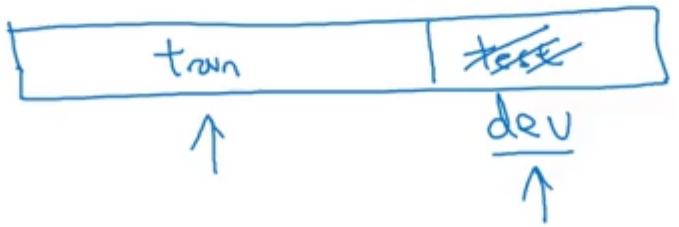
Size of test set

→ Set your test set to be big enough to give high confidence in the overall performance of your system.

10,000
100,000

Train + Dev.

No test set.



When to change dev/test sets and metrics

Cat dataset examples

Metric + Dev : Prefer A
You/users : Prefer B.

Metric: classification error

Algorithm A: 3% error → pornographic

✓ Algorithm B: 5% error

$$\text{Error}: \frac{1}{m_{dev}} \sum_{i=1}^{m_{dev}} 1\{y_{pred}^{(i)} \neq y^{(i)}\}$$

$$\text{Error}: \frac{1}{\sum w^{(i)}} \sum_{i=1}^{m_{dev}} w^{(i)} 1\{y_{pred}^{(i)} \neq y^{(i)}\}$$

predicted value (0/1)

$$\rightarrow w^{(i)} = \begin{cases} 1 & \text{if } x^{(i)} \text{ is non-porn} \\ 10 & \text{if } x^{(i)} \text{ is porn} \end{cases}$$

When to change dev/test sets and metrics

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. ↑ Aim/shoot at target.

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

When to change dev/test sets and metrics

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

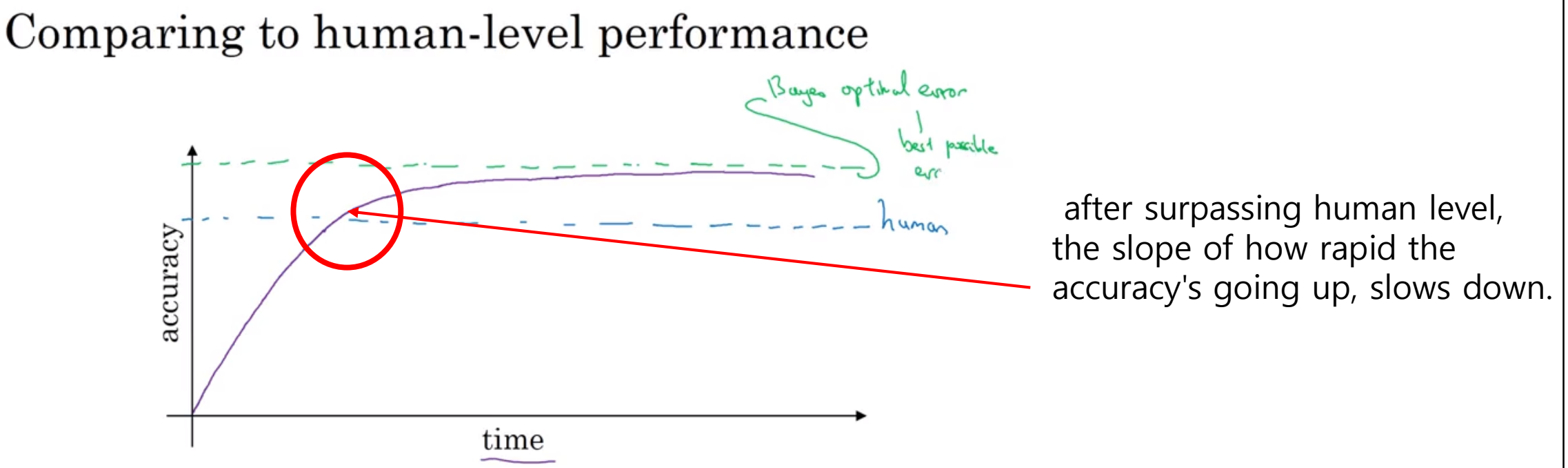
3. Comparing to human level performance

Why human level performance?

1. because of advances in deep learning, machine learning algorithms become competitive with human-level performance.

2. workflow is much more efficient when you're trying to do something that humans can also do.

Why human level performance?



1. Human-level performance is close to Bayesian optimal error so there is not much room to improve.
2. When performance is less than Human-level performance, you can try some tools to improve performance that are harder to use once you've surpassed human level performance.

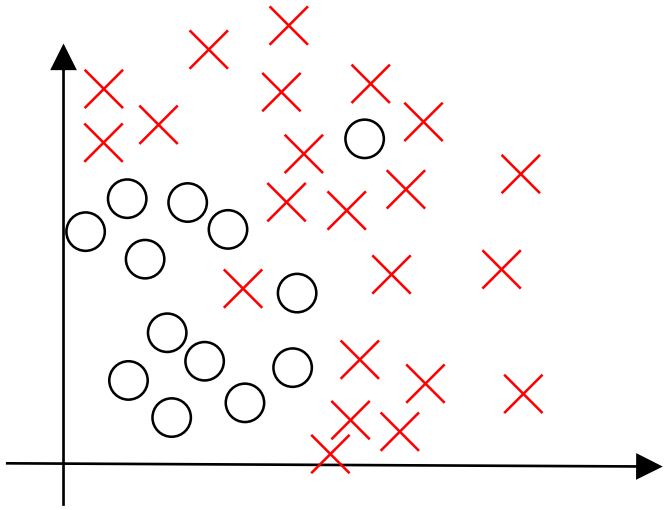
Why human level performance?

Why compare to human-level performance

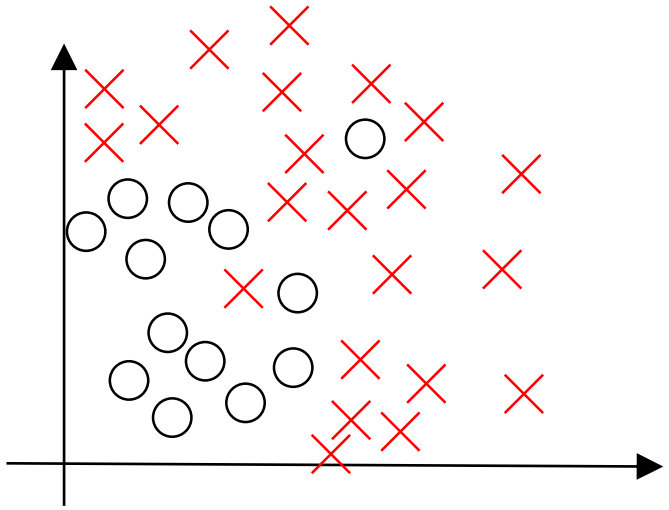
Humans are quite good at a lot of tasks. So long as ML is worse than humans, you can:

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

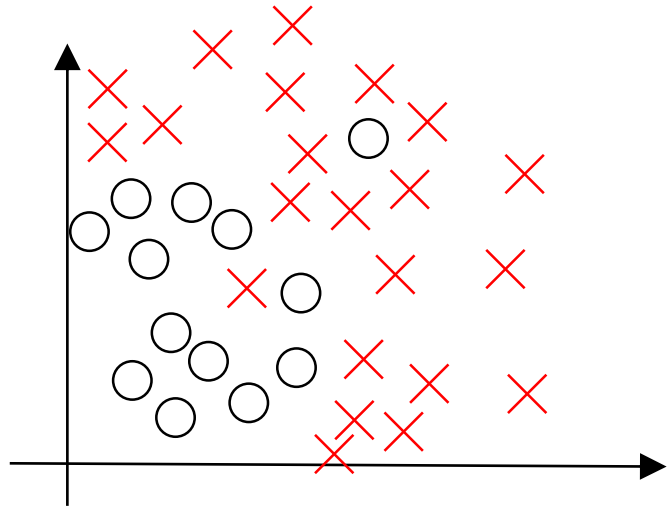
Bias and Variance



high bias



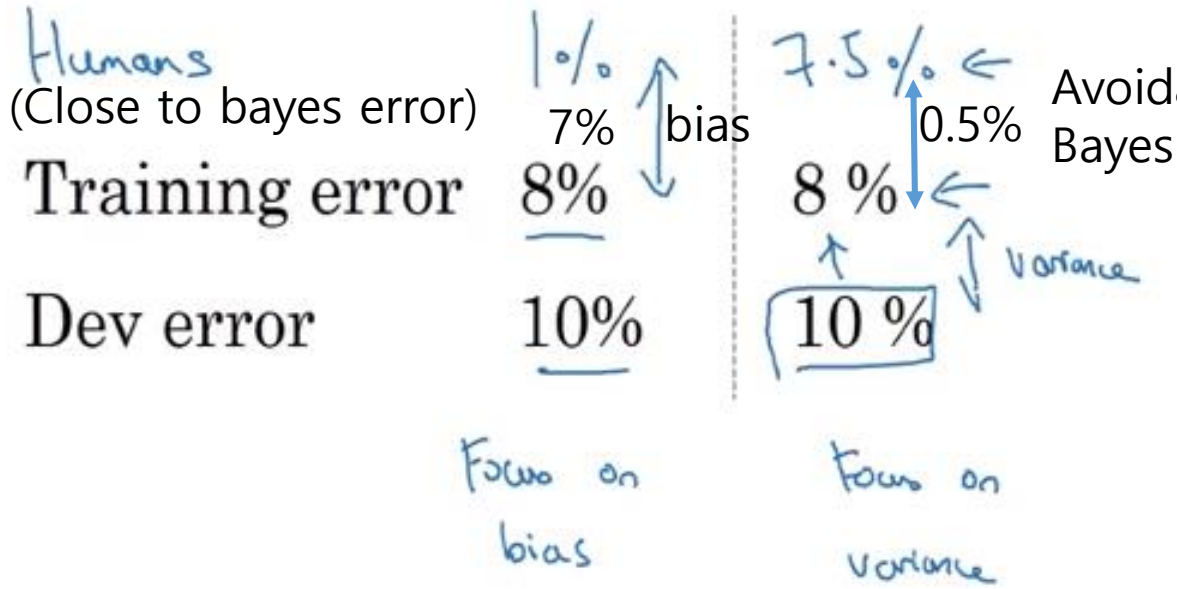
“just right”



high variance

Avoidable bias

Cat classification example



Avoidable error:
Bayes error(≈human error) – training error

train a bigger neural network or run training set longer

regularization

Understanding human-level performance

Human-level error as a proxy for Bayes error

Medical image classification example:

Suppose:

(a) Typical human ..... 3 % error

→ (b) Typical doctor ..... 1 % error

(c) Experienced doctor ..... 0.7 % error

→ (d) Team of experienced doctors .. 0.5 % error ←

Bayes error ≤ 0.5%

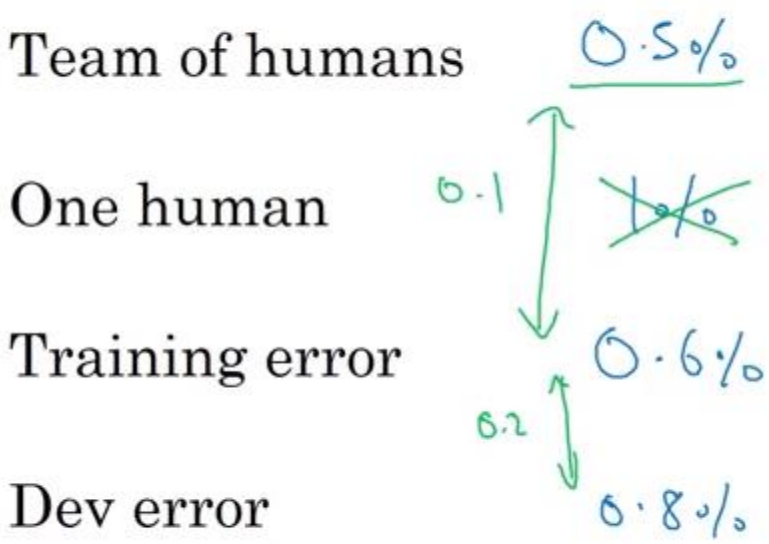
What is "human-level" error?

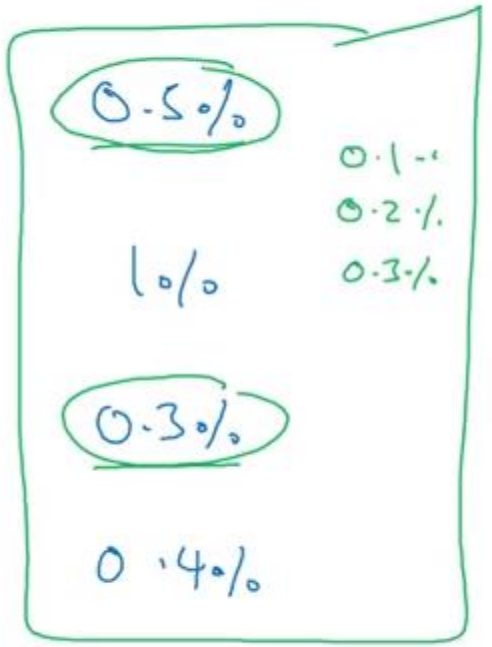
Understanding human-level performance

1%	1%	1%
0.7%	0.7%	0.7%
0.5%	0.5%	0.5%
5%	1%	0.7%
6%	5%	0.8%

Surpassing human level performance

Surpassing human-level performance





What is avoidable bias?

Surpassing human level performance

Problems where ML significantly surpasses human-level performance

- Online advertising
- Product recommendations
- Logistics (predicting transit time)
- Loan approvals

Structured data
Not natural perception
Lots of data

- Speech recognition
- Some image recognition
- Medical
  - ECG, Skin cancer, ...

Improving your model performance

The two fundamental assumptions of supervised learning

1. You can fit the training set pretty well.

~ Avoidable bias

2. The training set performance generalizes pretty well to the dev/test set.

~ Variance

Improving your model performance