deeplearning.ai

Introduction to ML strategy

Why ML Strategy?

- what's the best strategy to improve a certain DL system?

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

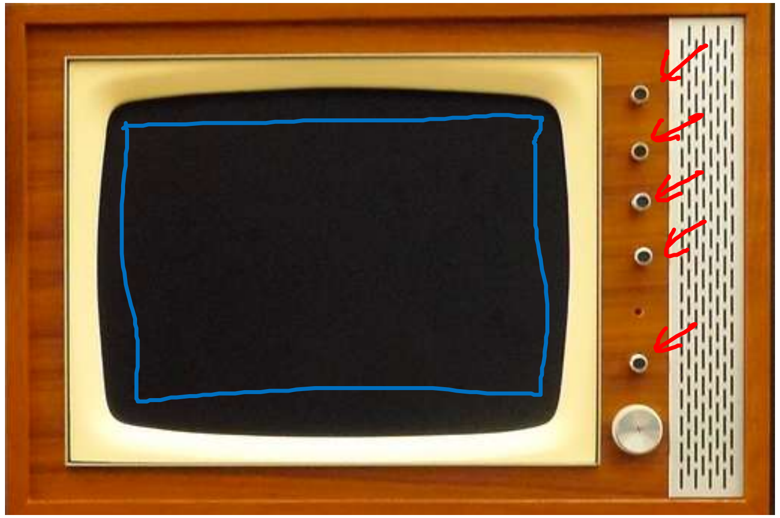
deeplearning.ai

Introduction to ML strategy

Orthogonalization

- It's easier to change/tune the hyperparameters of the DL algorithm if their effects are independent from each other.

TV tuning example



0.1 ×

+ 0.3 ×

− 1.7 ×

+ 0.8 ×

+ ...

Orthogonalization

Car

→ Steering

→ Acceleration

Braking

→ 0.3 × angle − 0.8 speed

→ 2 × angle + 0.9 speed.

speed

angle

Chain of assumptions in ML

For ML applications we hope that

⑤ for ML we have orthogonal hyperparameters to try meeting each of the assumptions.

① basic fitting

→ Fit training set well on cost function

(≈ human-level performance)

bigger network
Adam
...

width

② generalization

DO NOT USE
early stopping

→ Fit dev set well on cost function

Regularization
Bigger train set

because it has effects on network size AND regularization and is therefore not orthogonal.

height

③ performance

→ Fit test set well on cost function

Bigger dev set
(existing dev set is not representative enough of the training set).

④ real-world performance

→ Performs well in real world

(Happy cat pic app users.)

Change dev set or cost function

?

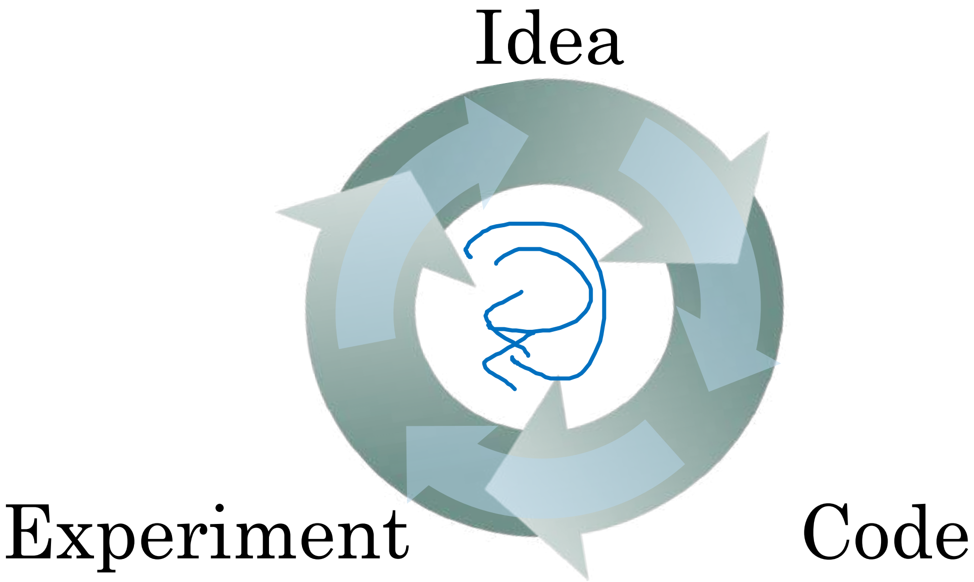
- AT THE BEGINNING OF THE PROJECT ONE MUST IMMEDIATELY SET UP A DEV SET THAT REFLECTS THE REAL APPLICATION AND A PROPER EVALUATION METRICS.
- These should be reassessed periodically to compensate unforeseen side-effects.

Setting up your goal

deeplearning.ai

Single number evaluation metric

Using a single number evaluation metric



→ Of examples recognized as ~~cat~~ CATS, what % actually are cats? $\frac{TP}{TP+FP}$

→ what % of actual cats are correctly recognized $\frac{TP}{TP+FN}$

Classifier	Precision	Recall
A	95%	90%
B	98%	85%

F_1 score = "Average" of P and R.

($\frac{2}{\frac{1}{P}+\frac{1}{R}}$. "Harmonic mean")

Dev set + Single [real] number evaluation metric

Speed up iterating

Andrew Ng

Another example

Algorithm	US	China	India	Other
A	3%	7%	5%	9%
B	5%	6%	5%	10%
C	2%	3%	4%	5%
D	5%	8%	7%	2%
E	4%	5%	2%	4%
F	7%	11%	8%	12%

AVERAGE

deeplearning.ai

Setting up your goal

Satisficing and optimizing metrics

Idea: one can have and combine different types of metrics. Some metrics need to be optimized (max/minimized), eg accuracy, others need only to be satisfied in a binary way (below/above a threshold), e.g. running time ≤ 100 ms.

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.

"Satisficing"

N metrics : 1 optimizing

N−1 satisficing

Wakewords / Trigger words

Alexa, OK Google,

Hey Siri, nihaobaidu

你好百度

accuracy.

#false positive

maximize accuracy.

s.t. ≤ 1 false positive

every 24 hours.

e.g. for Alexa we might want to maximize the recall, while ensuring the #FP ≤ 1 every 24 hours.

Andrew Ng

deeplearning.ai

Setting up your goal

Train/dev/test distributions

1- make sure that dev and test set come from the same distribution and represent well the real data the system will be used on.

Cat classification dev/test sets

development set, hold out cross validation set

Regions:

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

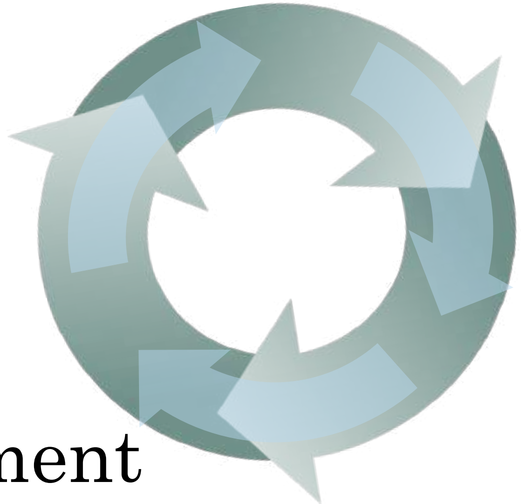
Dev

Test

Randomly shuffle into dev/test

dev set + metric

Idea



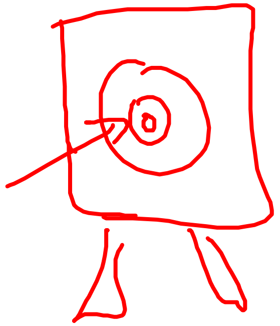
Experiment

Code

True story (details changed)

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

$x \longrightarrow y$ (repay loan?)



Tested on low income zip codes

~3 month

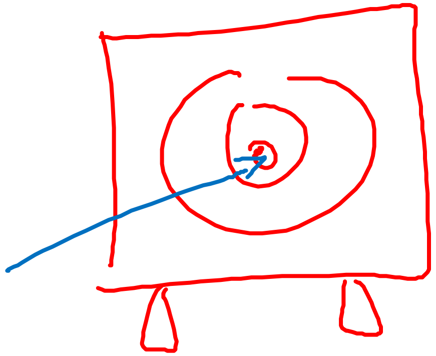
Andrew Ng

Guideline

Same distribution

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

training



dev

metric

test

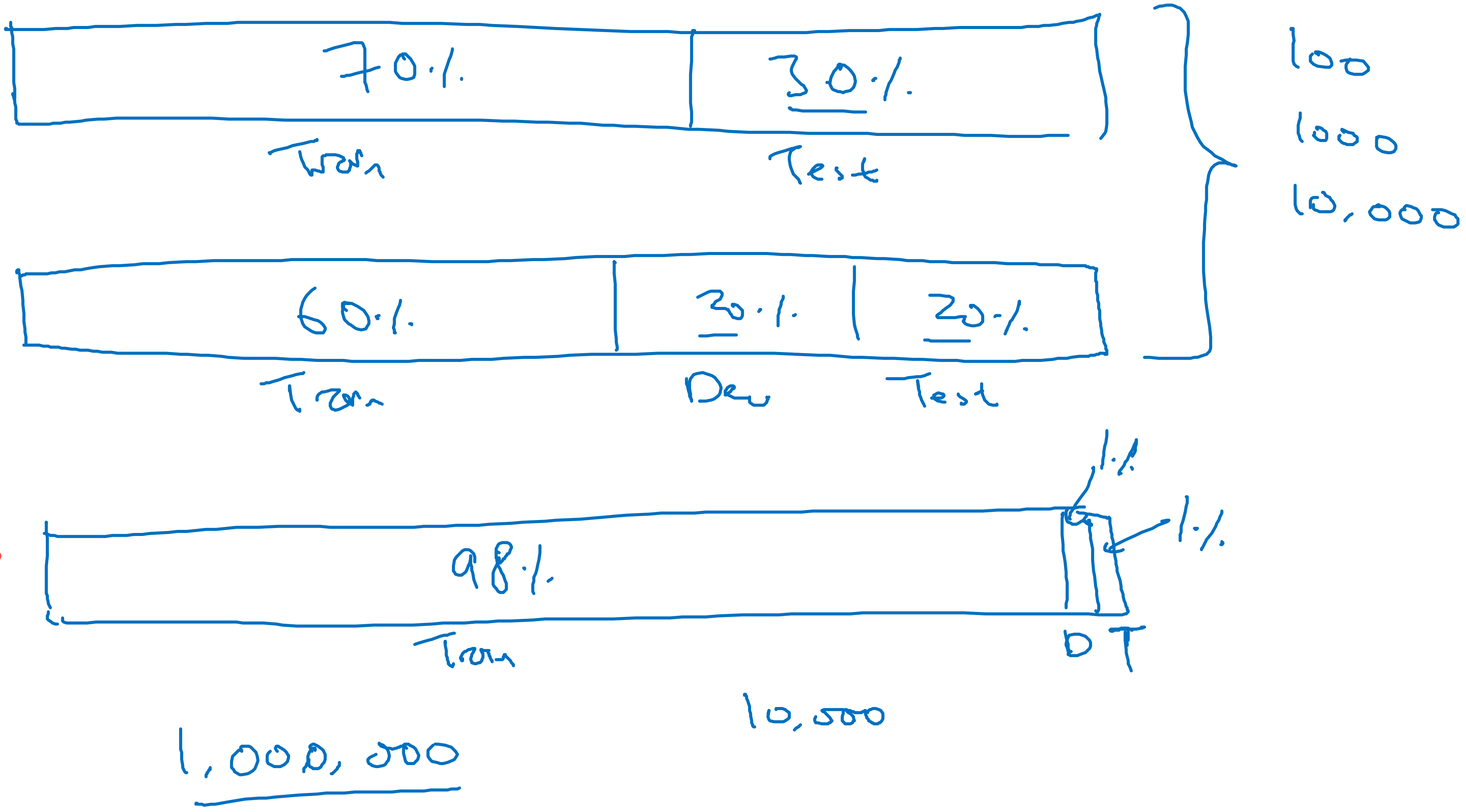
Andrew Ng

deeplearning.ai

Setting up your goal

Size of dev and test sets

Old way of splitting data



In huge datasets (>1M samples) used for DL, one can split TR/DEV/TE 98% 1% 1%

Size of dev set

A B

Set your dev set to be big enough to detect differences in algorithm/models you're trying out.

100 : small

1%

1,000

10,000

100,000

A 97% → B 97.1%

0.1%

0.01%

0.001%

Online advertising

Andrew Ng

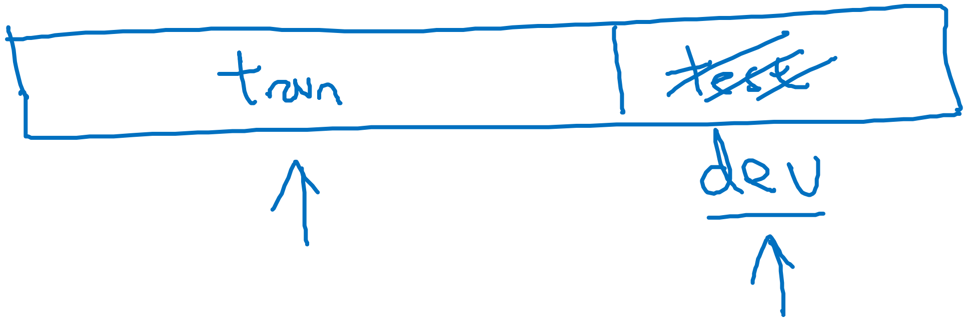
Size of test set

- 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.



People sometimes do this but Ng does not recommend this.

deeplearning.ai

Setting up your goal

When to change dev/test sets and metrics

The performance metric and dev set must be corrected to catch unforeseen side effects.

Cat dataset examples

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

→ Metric: classification error

Algorithm A: 3% error → pornographic

✓ Algorithm B: 5% error

$$\text{Error}: \frac{1}{\sum_i w^{(i)}} \cancel{\frac{1}{m_{dev}}} \sum_{i=1}^{m_{dev}} w^{(i)} \, \mathcal{I}\{ y_{pred}^{(i)} \neq y^{(i)} \}$$

predicted value (0/1)

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

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

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

In case you wanna keep the metric (accuracy) you can decide to (orthogonally) modify the algorithm (loss in this case) so to counteract side effects.

↳ ? I don't fully get why one should change the loss but keep the old metric...

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.

deeplearning.ai

Comparing to human-level performance

Why human-level performance?

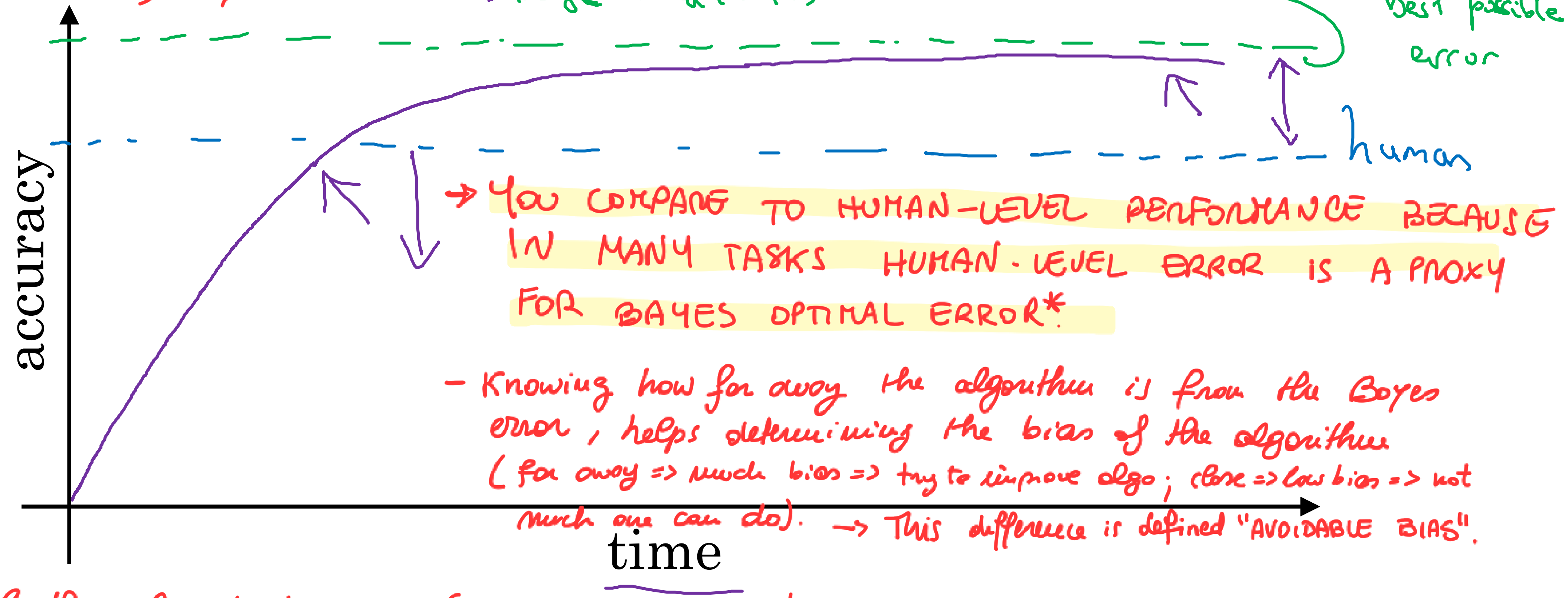
Comparing to human-level performance

- It is useful to know the human level performance on tasks where humans are not great, such as speech recognition with very noisy audio.

$X \rightarrow y$

→ audio → transcript

→ image → cat (0/1)



→ YOU COMPARE TO HUMAN-LEVEL PERFORMANCE BECAUSE IN MANY TASKS HUMAN-LEVEL ERROR IS A PROXY FOR BAYES OPTIMAL ERROR*.

- Knowing how far away the algorithm is from the Bayes error, helps determining the bias of the algorithm (far away => much bias => try to improve algo; close => low bias => not much one can do). → This difference is defined "AVOIDABLE BIAS".

- Moreover, until the algorithm's error (training error) is below human's error, the human can try to help the algorithm further (better labeling, identifying sources of noise); when the algorithm performs better than the human, this is not possible anymore.

*BAYES OPTIMAL ERROR: prediction error of the best function theoretically possible to solve the problem

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. → explained later

- Until the algorithm's performance are worse than humans, the human can do a lot to improve the algorithm's performance (labeling, identifying sources of imprecision in the algorithm, deciding if the bias of the algorithm is too high).

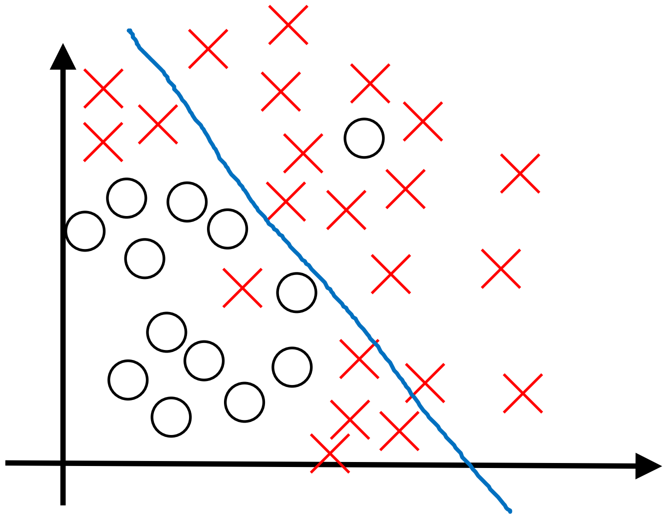
Andrew Ng

deeplearning.ai

Comparing to human-level performance

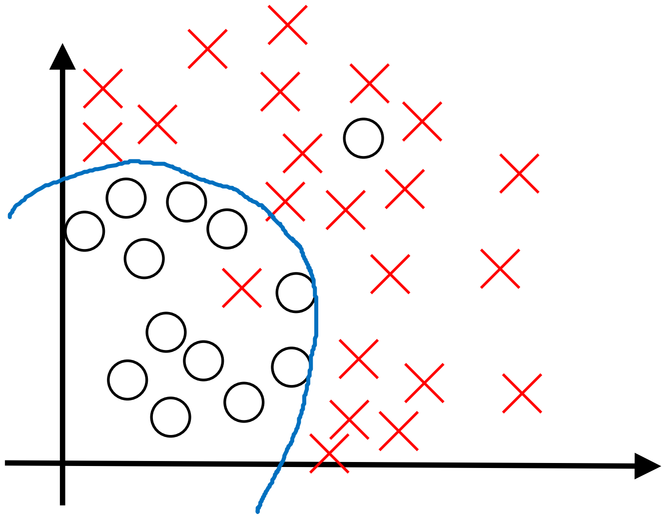
Avoidable bias

Bias and Variance

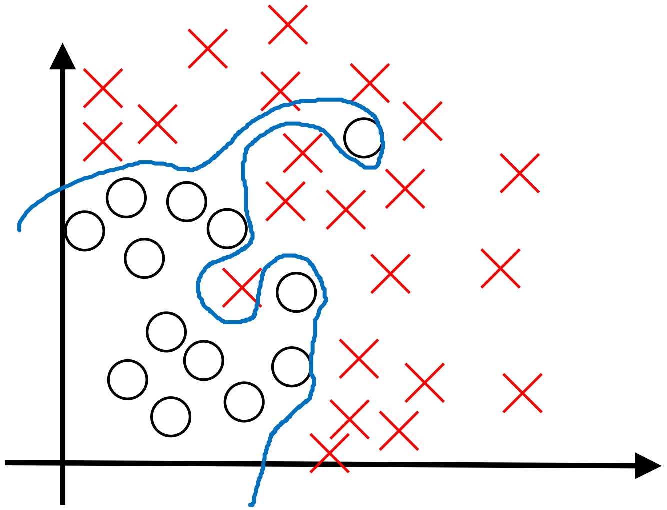


high bias

underfitting



"just right"



high variance

overfitting.

Bias and Variance

Cat classification

Human-level ≈ 0% ....

Training set error:

Dev set error:

high variance   high bias   high bias   low bias
                                    high variance   low variance

Andrew Ng

Cat classification example

AVOIDABLE BIAS: difference between training error and (approximation of) Bayes error. This bias can likely be modified acting on the algorithm.

Humans (≈ Bayes)	1%	7.5%
Training error	8%	8 %
Dev error	10%	10 %

7% 2% 0.5% 2% 2%

Variance

Avoidable bias

Variance

Focus on bias

Focus on variance

Human-level error as a proxy for Bayes error.

deeplearning.ai

Comparing to human-level performance

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

Baye error $\leq$ 0.5%

What is "human-level" error?

Error analysis example

Human (proxy for Bayes error)	1% 0.7% 0.5%	1% 0.7% 0.5%	0.7% 0.5% 1%
Avoidable bias	4% 4.5%	0% 0.5%	0.2% 0.0%
Training error	5%	1%	$\boxed{0.7\%}$
Variance	1%	4%	0.1%
Dev error	6%	5%	0.8%
	Bias	Variance	

Summary of bias/variance with human-level performance

0%

"Bias"

Human-level error

(proxy for Bayes error)

"Avoidable bias"

Training error

"Variance"

Dev error

Andrew Ng

deeplearning.ai

Comparing to human-level performance

Surpassing human-level performance

Surpassing human-level performance

Team of humans	0.5%	0.5%
One human	~~1%~~	1%
Training error	0.6%	0.3%
Dev error	0.8%	0.4%

0.1

0.2

0.1%

0.2%

0.3%

What is avoidable bias?

Andrew Ng

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, ...

deeplearning.ai

Comparing to human-level performance

Improving your model performance

The two fundamental assumptions of supervised learning

1. You can fit the training set pretty well.

   Reduce

   ~ Avoidable bias

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

   Reduce

   ~ Variance

Andrew Ng

Reducing (avoidable) bias and variance

Human-level = Proxy for Bayes error

Avoidable bias

Training error

Variance

Dev error

Train bigger model

Train longer/better optimization algorithms

- momentum, RMSprop, Adam
- step size, #epochs

NN architecture/hyperparameters search

RNN
CNN

- activation functions, #layers, # hidden units

More data

Regularization

- L_2, dropout, data augmentation

NN architecture/hyperparameters search