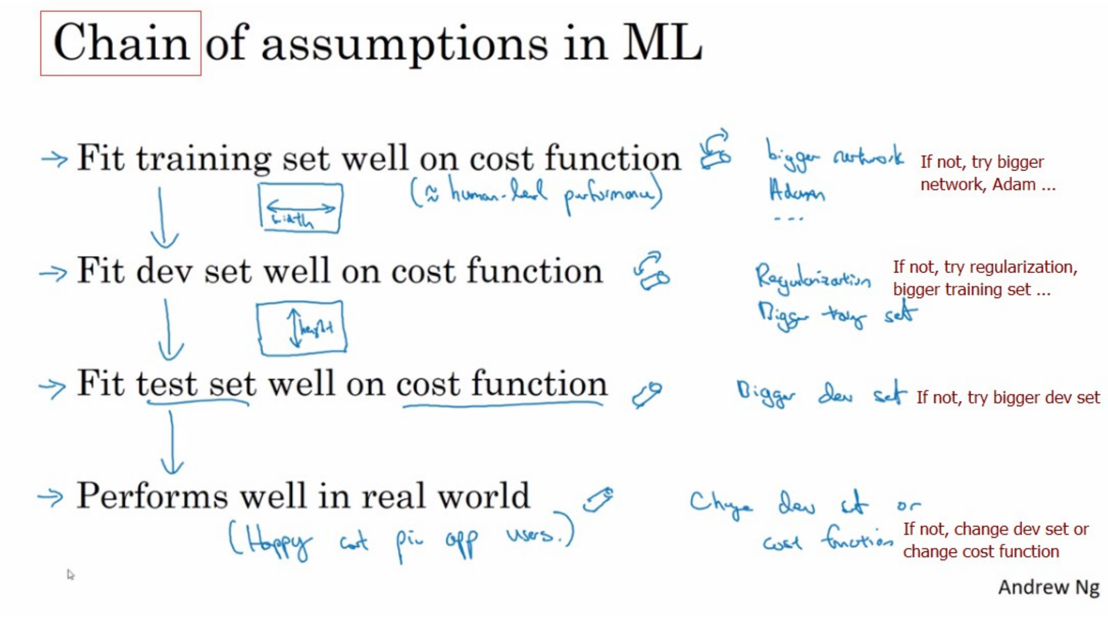


Chain of assumptions in ML

→ Fit training set well on cost function

(≈ human-level performance)

bigger network
Adam
...

If not, try bigger network, Adam ...

→ Fit dev set well on cost function

Regularization
Bigger train set

If not, try regularization, bigger training set ...

→ Fit test set well on cost function

Bigger dev set

If not, try bigger dev set

→ Performs well in real world

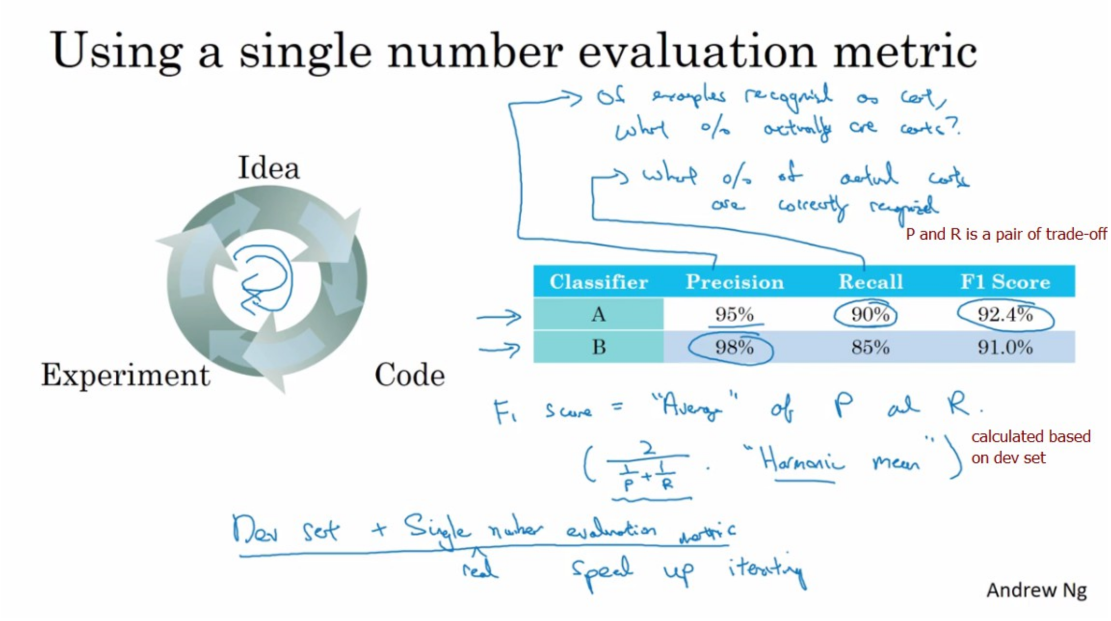
(Happy cat pic app users.)

Change dev set or cost function

If not, change dev set or change cost function

Andrew Ng

Single real number evaluation metric:



Using a single number evaluation metric

→ Of examples recognized as cat, what % actually are cats?

→ what % of actual cats are correctly recognized

P and R is a pair of trade-off

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

F₁ score = "Average" of P and R.

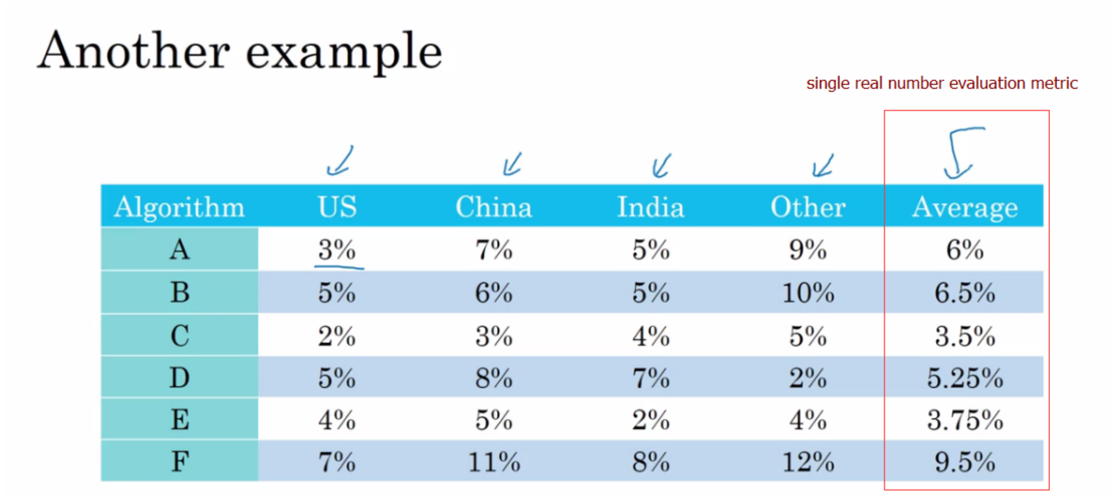
$\left(\frac{2}{\frac{1}{P}+\frac{1}{R}} \right.$. "Harmonic mean" $\left. \right)$

calculated based on dev set

Dev set + Single real number evaluation metric

Speed up iterating

Andrew Ng

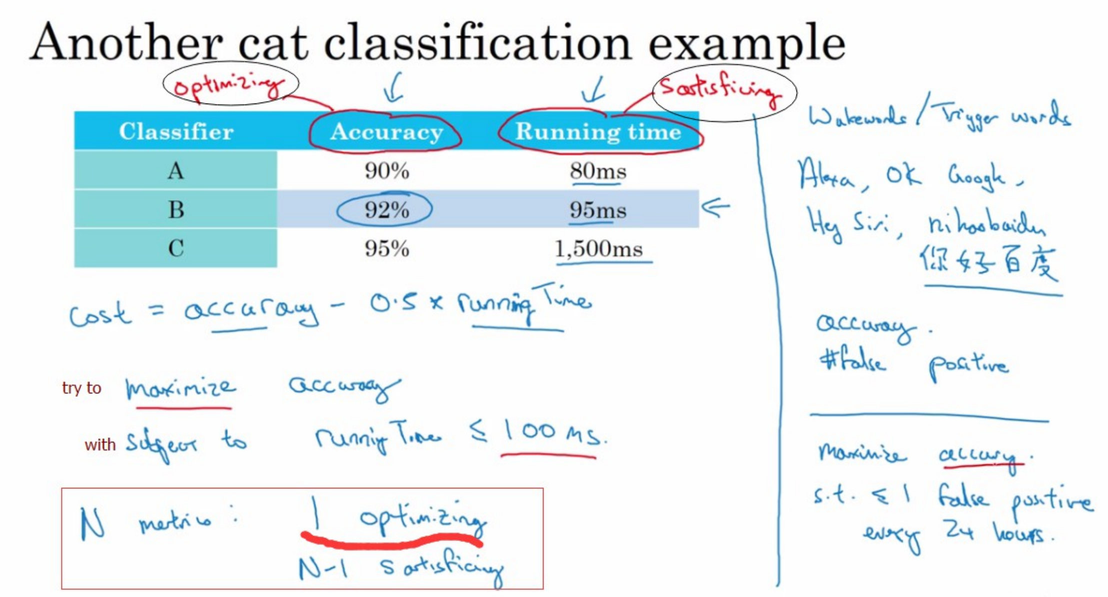


Another example

single real number evaluation metric

Algorithm	US	China	India	Other	Average
A	3%	7%	5%	9%	6%
B	5%	6%	5%	10%	6.5%
C	2%	3%	4%	5%	3.5%
D	5%	8%	7%	2%	5.25%
E	4%	5%	2%	4%	3.75%
F	7%	11%	8%	12%	9.5%

Optimizing and satisfying 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

try to maximize accuracy

with subject to running Time ≤ 100ms.

N metrics: 1 optimizing
N-1 satisficing

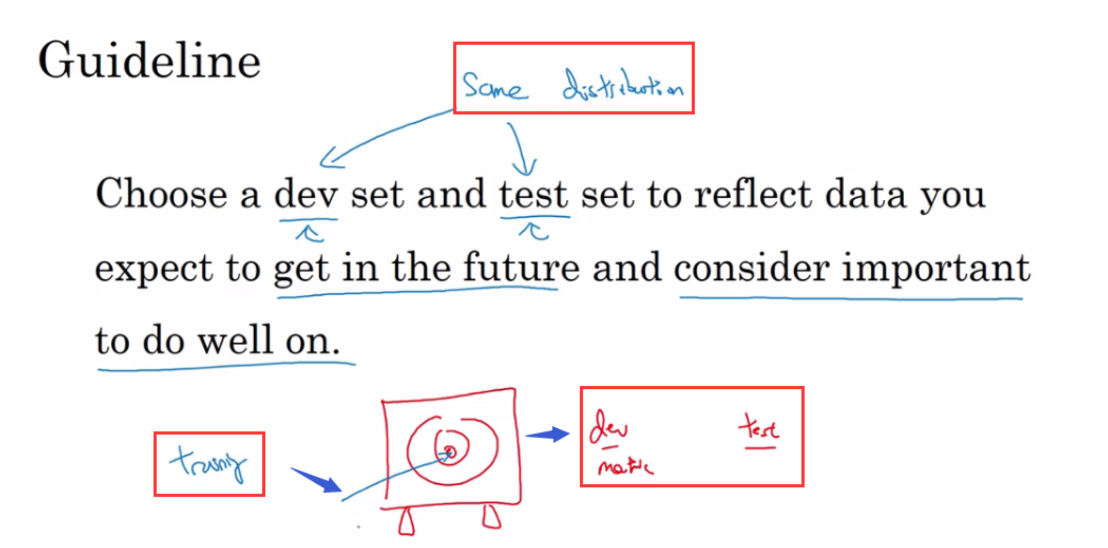
Wakewords / Trigger words

Alexa, OK Google,
Hey Siri, nihaobaidu
你好百度

accuracy.
#false positive

maximize accuracy.
s.t. ≤ 1 false positive
every 24 hours.

Train, Dev and Test set splitting:



Guideline

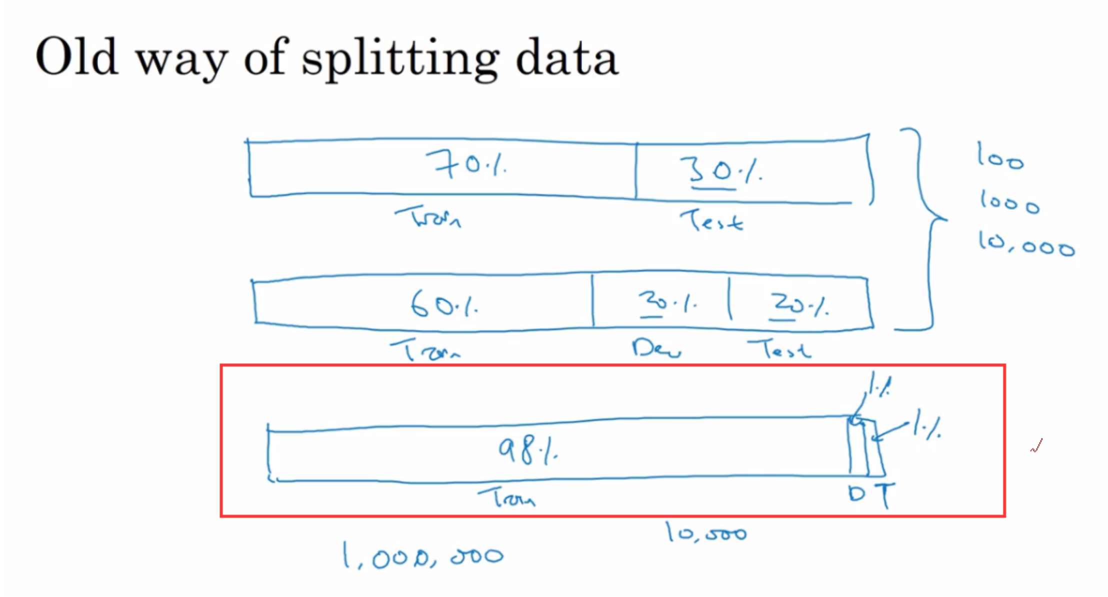
Same distribution

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

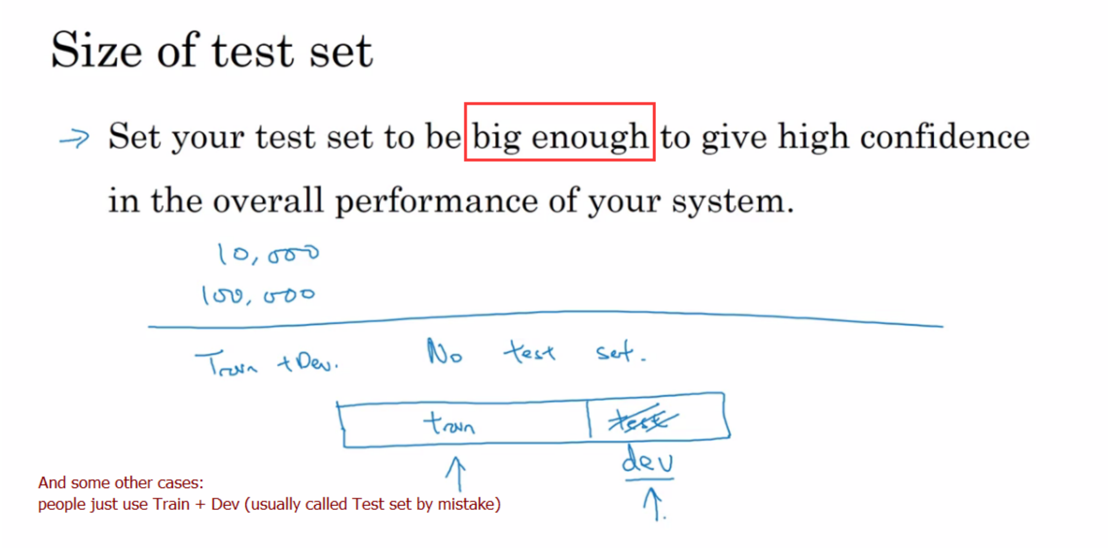
training

dev
metric

test



Old way of splitting data

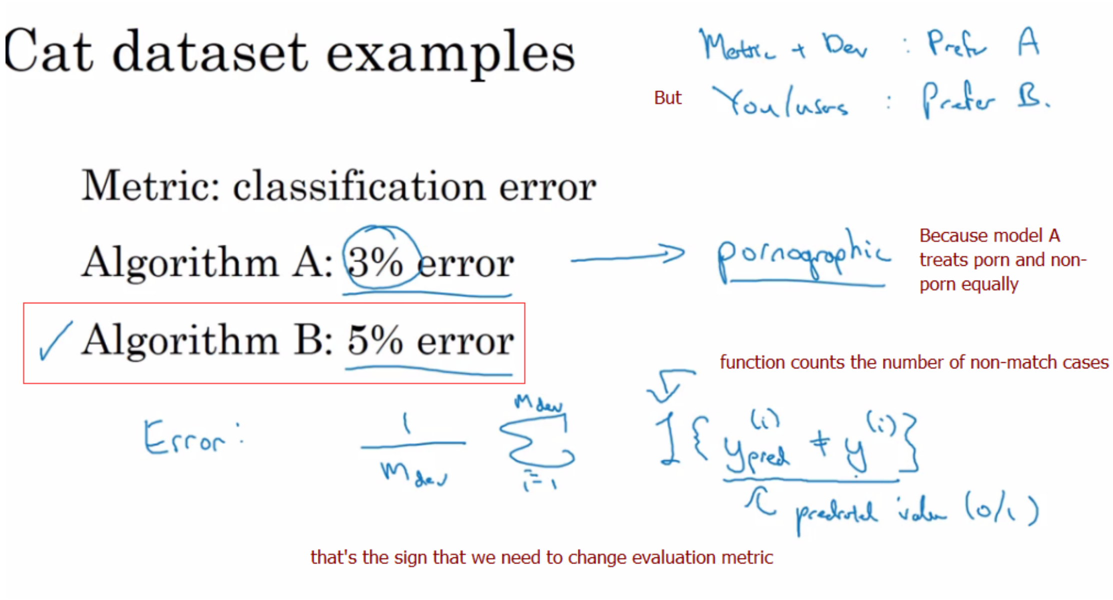


Size of test set

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

And some other cases:
people just use Train + Dev (usually called Test set by mistake)

When to change Dev/Test set and metric:



Cat dataset examples

Metric + Dev : Prefer A

But You/users : Prefer B.

Metric: classification error

Algorithm A: 3% error → pornographic

Because model A treats porn and non-porn equally

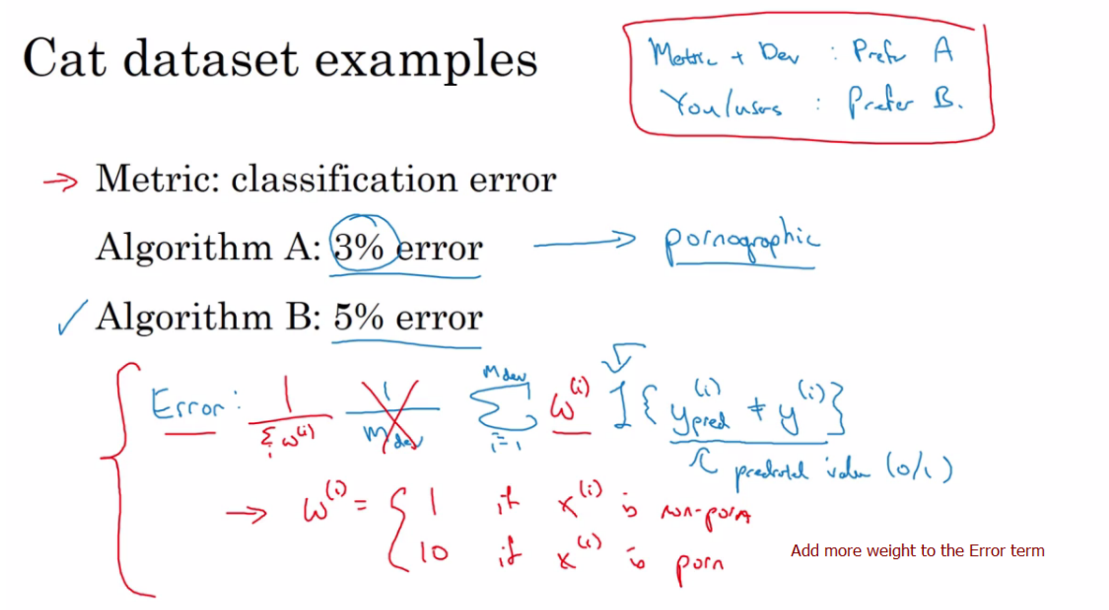
✓ Algorithm B: 5% error

function counts the number of non-match cases

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

predicted value (0/1)

that's the sign that we need to change evaluation metric



Cat dataset examples

Metric + Dev : Prefer A

You/users : Prefer B.

→ Metric: classification error

Algorithm A: 3% error → pornographic

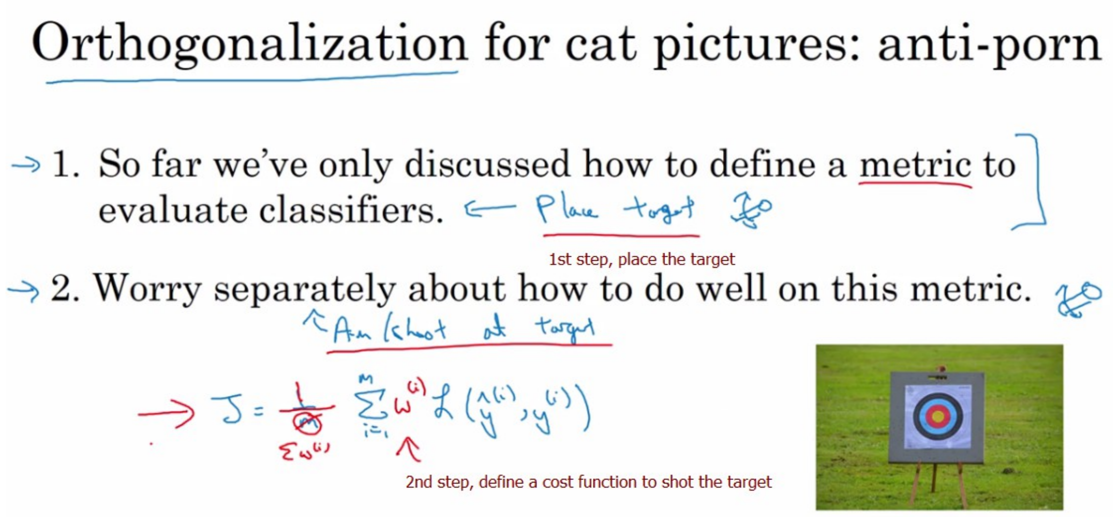
✓ Algorithm B: 5% error

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

predicted value (0/1)

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

Add more weight to the Error term



Orthogonalization for cat pictures: anti-porn

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

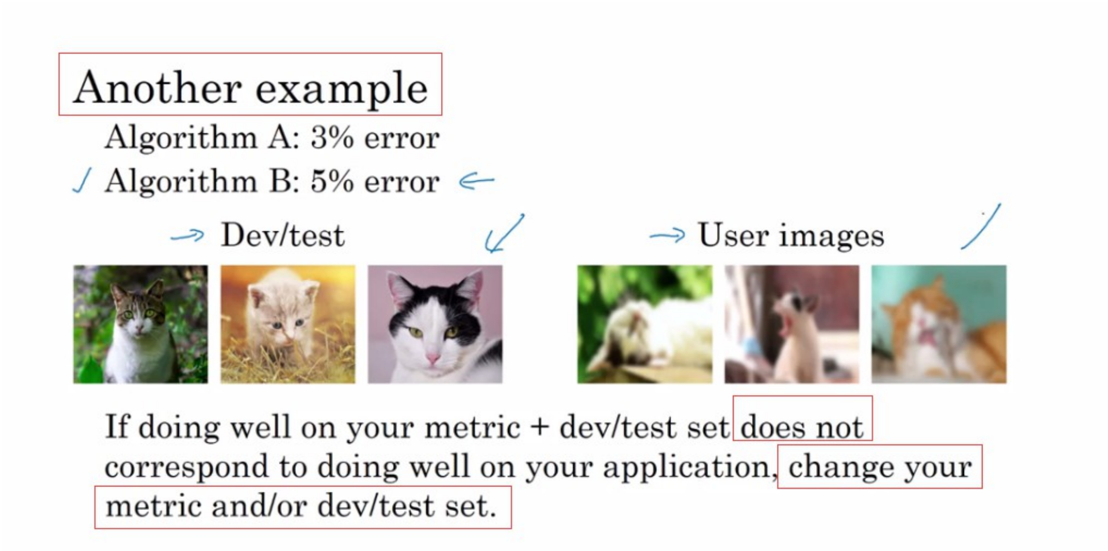
1st step, place the target

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

Aim/shoot at target

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

2nd step, define a cost function to shot the target



Another example

Algorithm A: 3% error

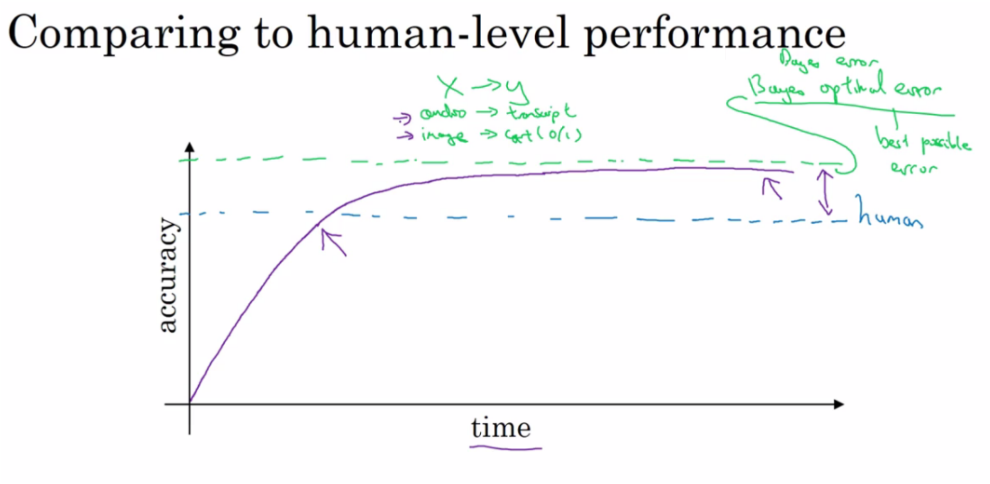
✓ Algorithm B: 5% error ←

→ Dev/test ✓

→ User images ✓

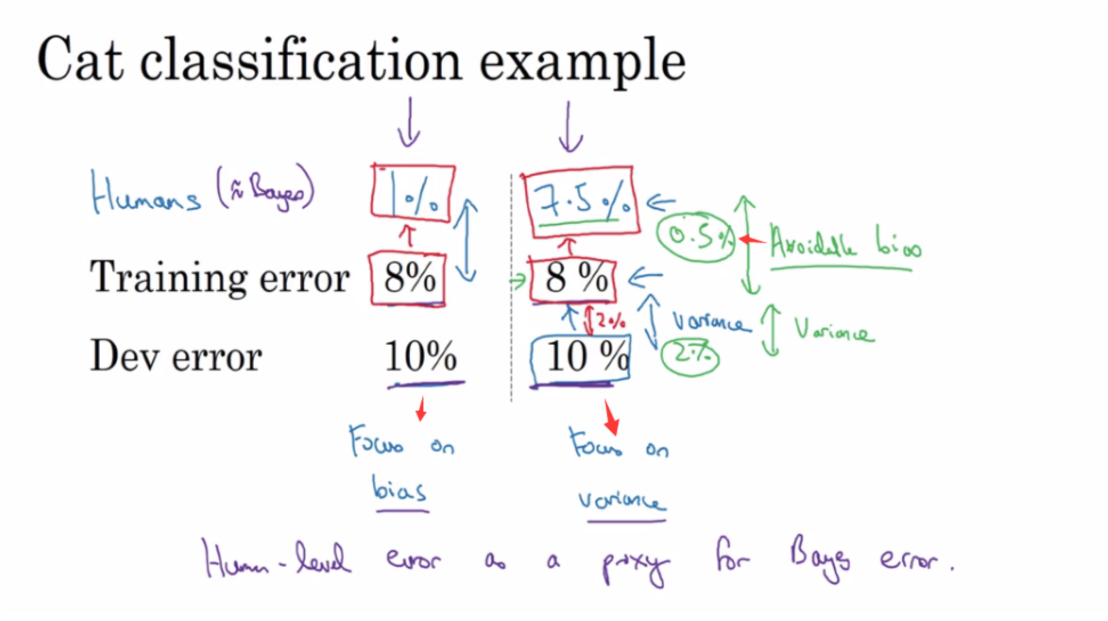
If doing well on your metric + dev/test set does not correspond to doing well on your application, change your metric and/or dev/test set.

Human Level Performance:



Comparing to human-level performance

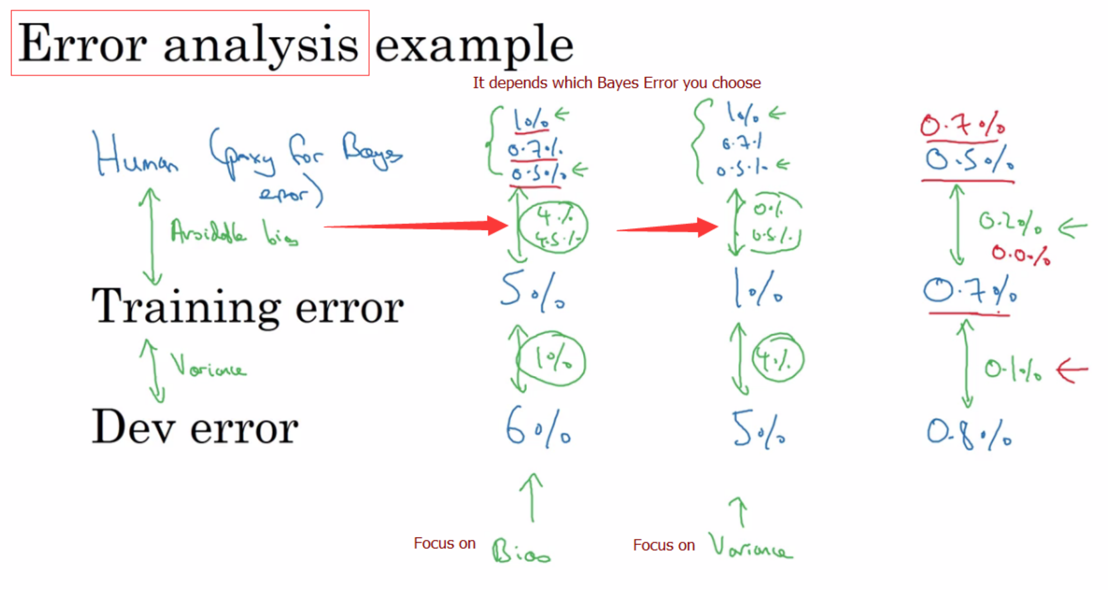
Avoidable Bias:



Cat classification example

Humans (≈ Bayes)	1%	7.5%
Training error	8%	8%
Dev error	10%	10%
	Focus on bias	Focus on variance

Human-level error as a proxy for Bayes error.

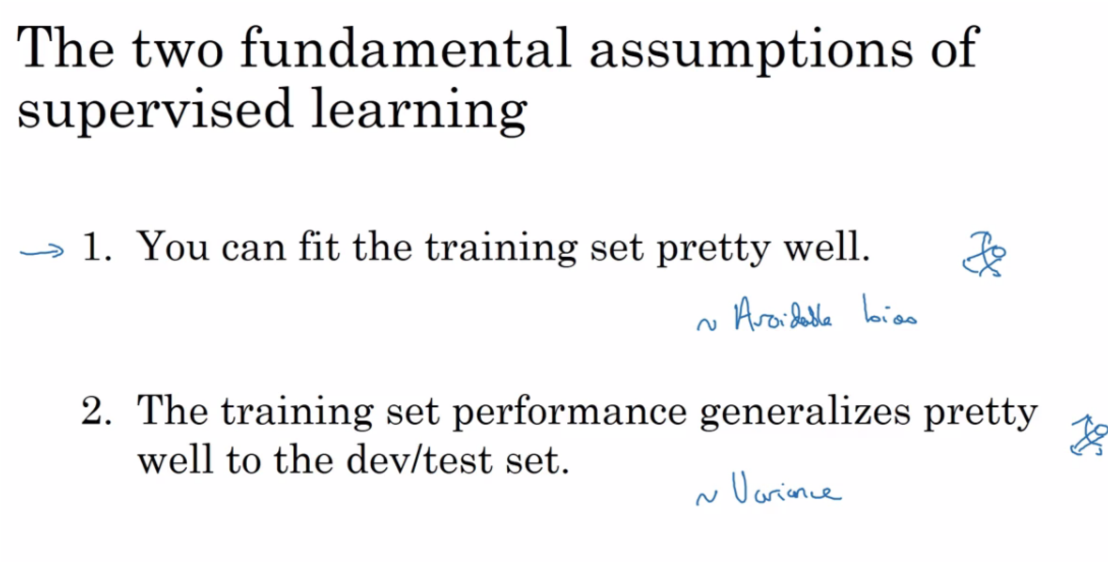


Error analysis example

It depends which Bayes Error you choose

Human (proxy for Bayes error)	1% 0.7% 0.5%	1% 0.7% 0.5%	0.7% 0.5%
Avoidable bias	4% 4.5%	0.% 0.5%	0.2% 0.0%
Training error	5%	1%	0.7%
Variance	1%	4%	0.1%
Dev error	6%	5%	0.8%
	Focus on Bias	Focus on Variance	

Improving Model Performance:



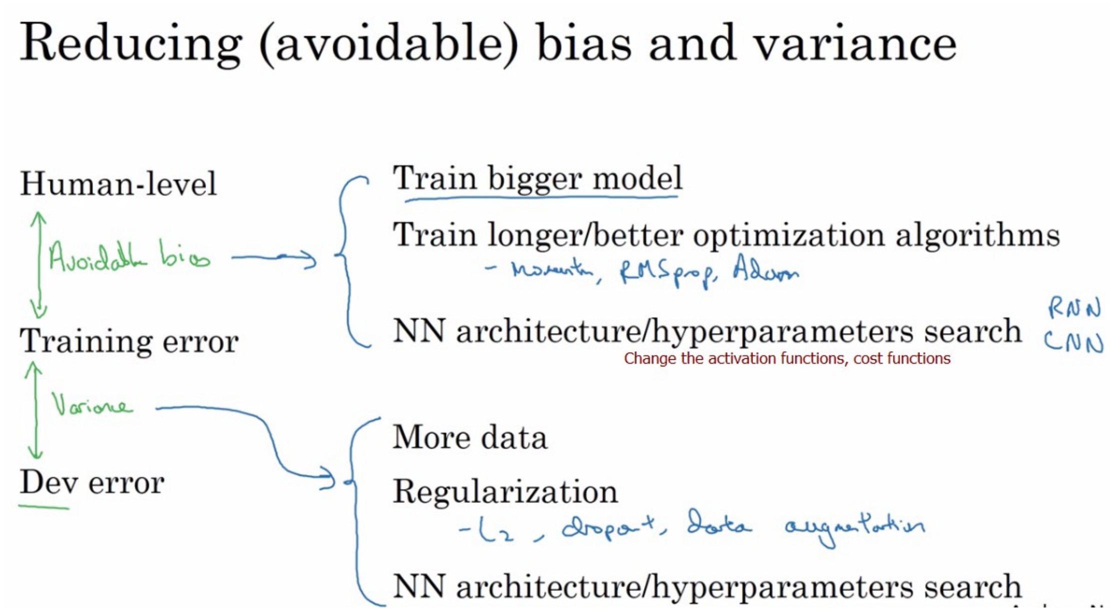
The two fundamental assumptions of supervised learning

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



Reducing (avoidable) bias and variance

Human-level

Avoidable bias

Training error

Variance

Dev error

Train bigger model

Train longer/better optimization algorithms

- momentum, RMSprop, Adam

NN architecture/hyperparameters search

RNN
CNN

Change the activation functions, cost functions

More data

Regularization

- L_2, dropout, data augmentation

NN architecture/hyperparameters search