Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

$$J(\vec{w}, b) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

But it makes unacceptably large errors in predictions. What do you try next?

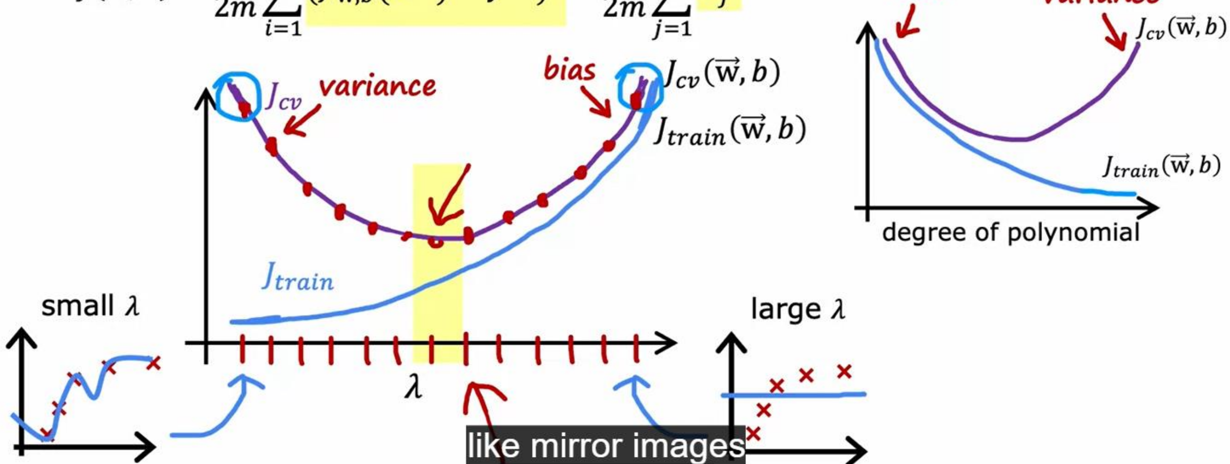
- Get more training examples
- Try smaller sets of features
- Try getting additional features
- Try adding polynomial features $(x_1^2, x_2^2, x_1x_2, etc)$
- Try decreasing λ
- Try increasing λ

and some of these
things not fruitful.

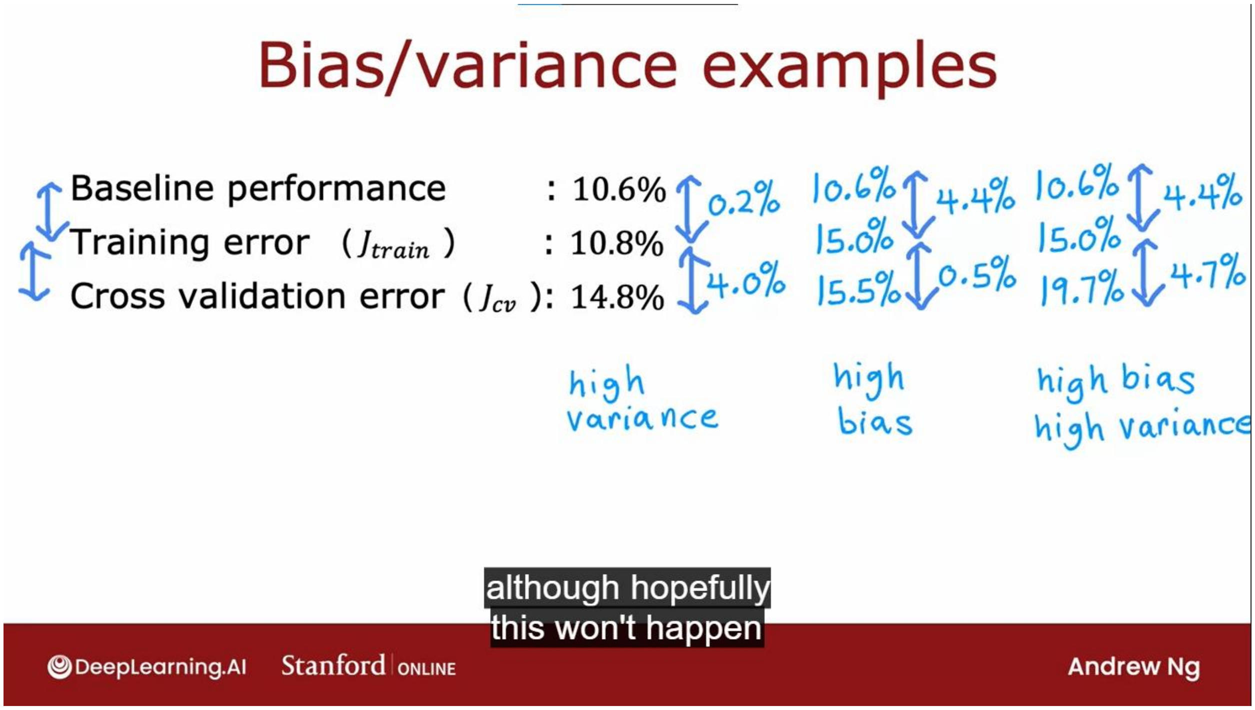
Bias and variance as a function of regularization parameter λ

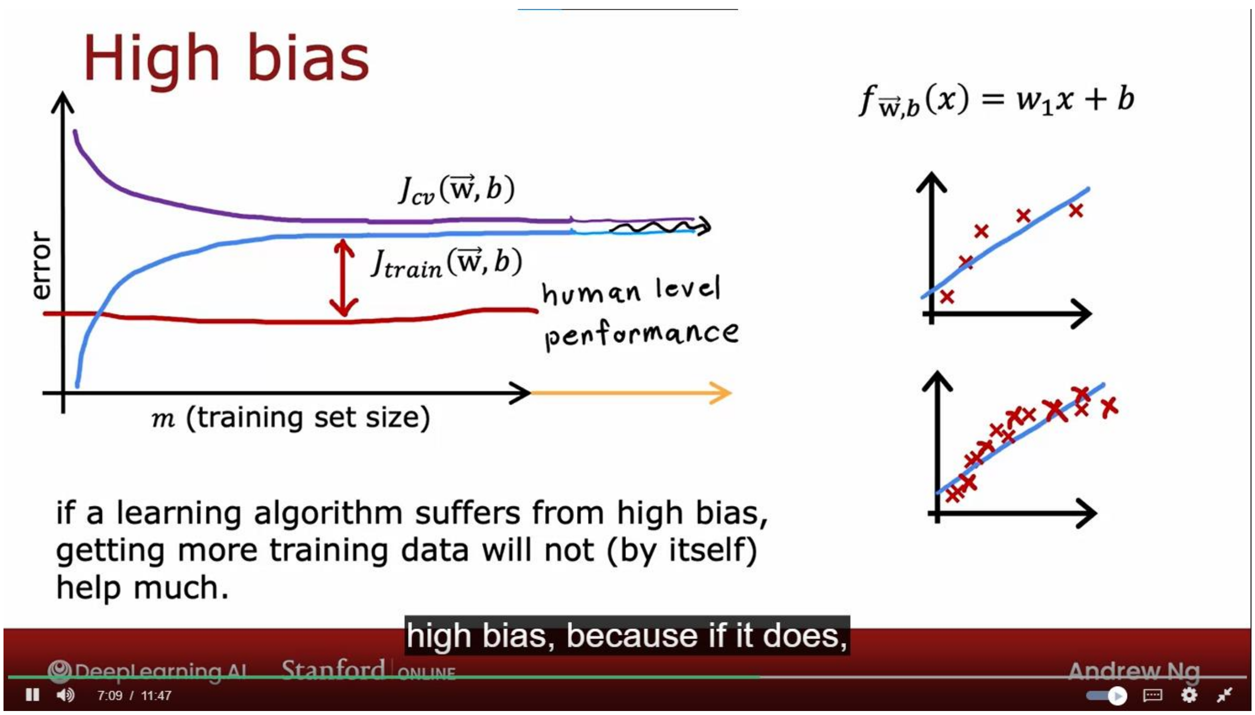
To exit full screen, press Esc

$$J(\vec{w}, b) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$



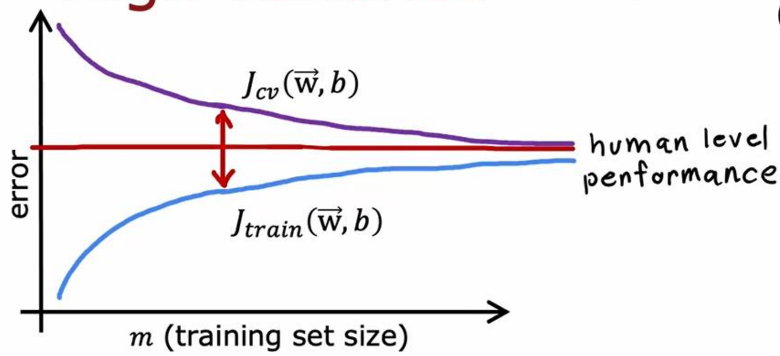
like mirror images
of each other.

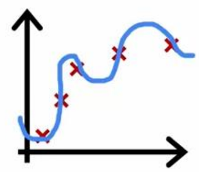

Bias/variance examples
Baseline performance : 10.6%
Training error (J_{train}) : 10.8%
Cross validation error (J_{cv}): 14.8%
0.2%
4.0%
10.6%
15.0%
15.5%
4.4%
0.5%
10.6%
15.0%
19.7%
4.4%
4.7%
high variance
high bias
high bias high variance
although hopefully this won't happen
DeepLearning.AI Stanford ONLINE
Andrew Ng

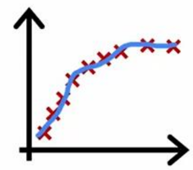

High bias
$f_{\vec{w},b}(x) = w_1x + b$
$J_{cv}(\vec{w}, b)$
$J_{train}(\vec{w}, b)$
error
human level performance
m (training set size)
if a learning algorithm suffers from high bias, getting more training data will not (by itself) help much.
high bias, because if it does,
DeepLearning.AI Stanford ONLINE
Andrew Ng
7:09 / 11:47

High variance

$$f_{\vec{w},b}(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$
(with small λ)







If a learning algorithm suffers from high variance, getting more training data is likely to help.

then you can just get
a better fourth order

Debugging a learning algorithm

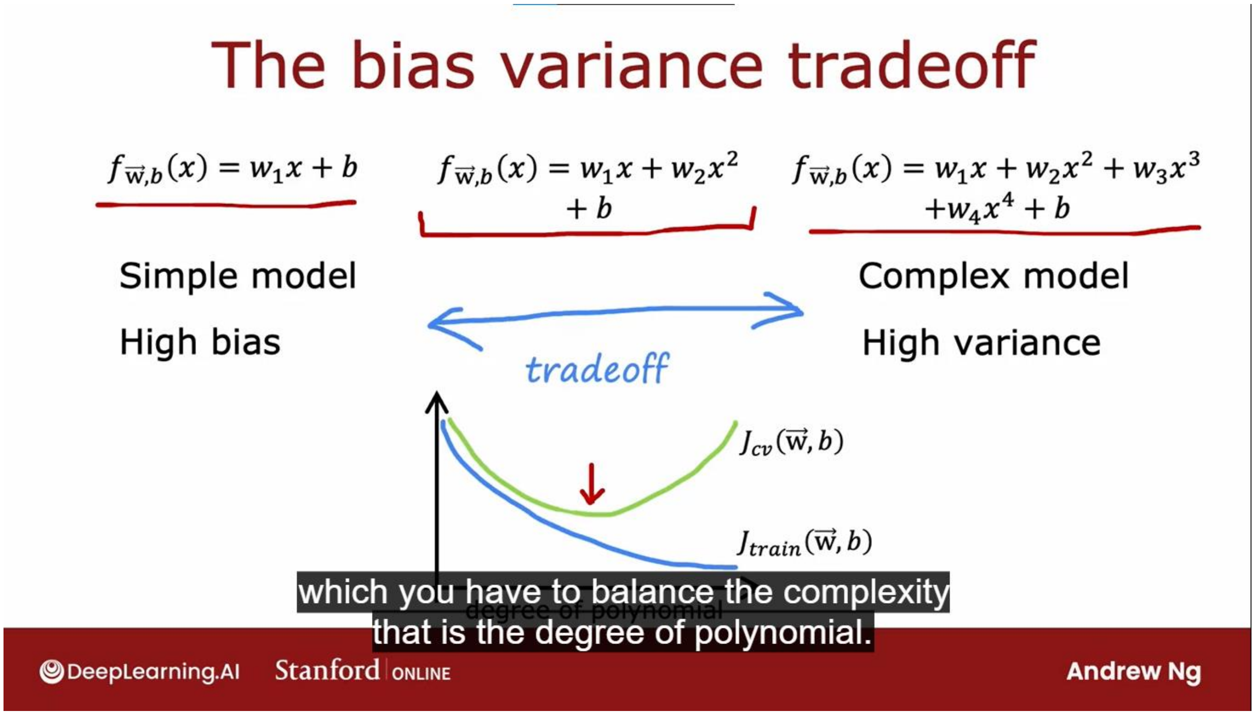
You've implemented regularized linear regression on housing prices

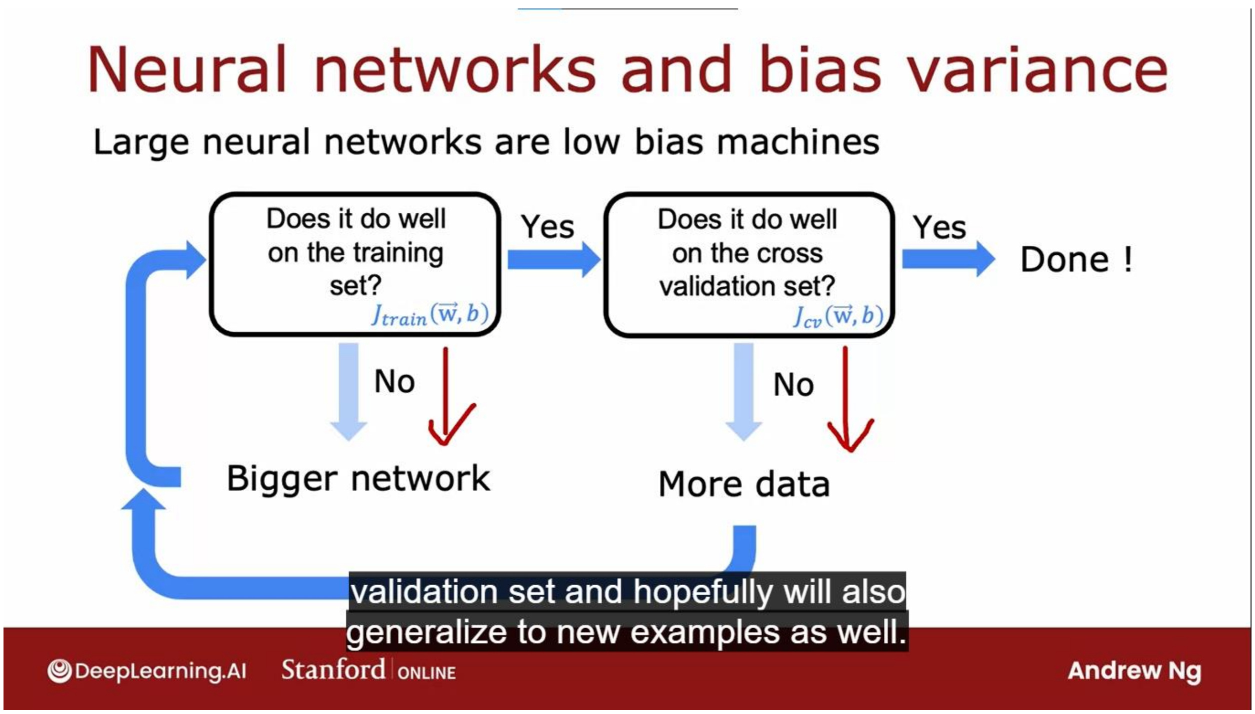
$$J(\vec{w}, b) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

But it makes unacceptably large errors in predictions. What do you try next?

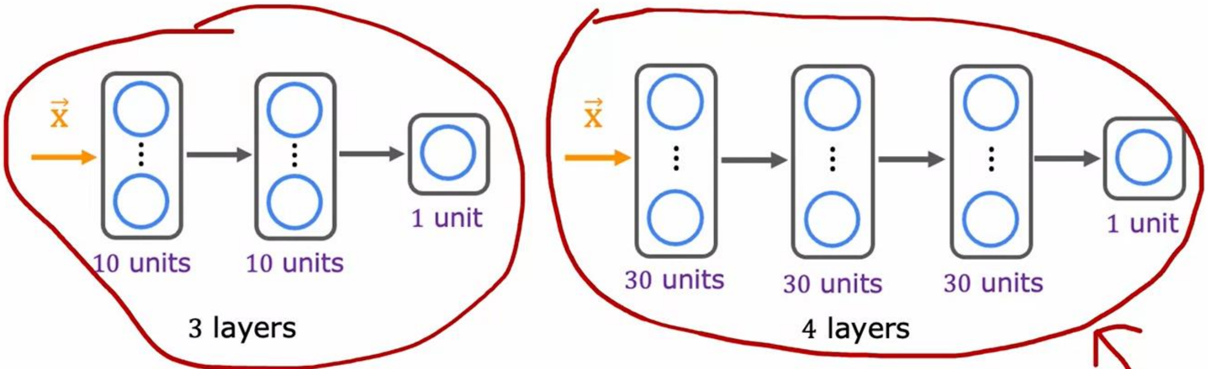
- Get more training examples — fixes high variance
- Try smaller sets of features $x, x^2, \cancel{x^3}, \cancel{x^4}, \cancel{x^5} \ldots$ — fixes high variance
- Try getting additional features — fixes high bias
- Try adding polynomial features $(x_1^2, x_2^2, x_1x_2, etc)$ — fixes high bias
- Try decreasing λ — fixes high bias
- Try increasing λ — fixes high variance

If that's the case,
the main fixes


The bias variance tradeoff
$f_{\vec{w},b}(x) = w_1x + b$
$f_{\vec{w},b}(x) = w_1x + w_2x^2 + b$
$f_{\vec{w},b}(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$
Simple model
High bias
Complex model
High variance
tradeoff
$J_{cv}(\vec{w}, b)$
$J_{train}(\vec{w}, b)$
degree of polynomial
which you have to balance the complexity
that is the degree of polynomial.
DeepLearning.AI
Stanford ONLINE
Andrew Ng


Neural networks and bias variance
Large neural networks are low bias machines
Does it do well on the training set?
$J_{train}(\vec{w}, b)$
Yes
Does it do well on the cross validation set?
$J_{cv}(\vec{w}, b)$
Yes
Done !
No
Bigger network
No
More data
validation set and hopefully will also
generalize to new examples as well.
DeepLearning.AI
Stanford ONLINE
Andrew Ng

Neural networks and regularization



A large neural network will usually do as well or better than a smaller one so long as regularization is chosen appropriately.

So the main way it hurts,
it will slow down your training and

DeepLearning.AI Stanford ONLINE Andrew Ng

Neural network regularization

$$J(\mathbf{W},\mathbf{B}) = \frac{1}{m}\sum_{i=1}^{m} L\left(f\left(\vec{\mathrm{x}}^{(i)}\right), y^{(i)}\right) + \frac{\lambda}{2m}\sum_{\mathit{all\ weights}\ \mathbf{W}} \left(w^2\right)$$

b

Unregularized MNIST model

```
layer_1 = Dense(units=25, activation="relu")
layer_2 = Dense(units=15, activation="relu")
layer_3 = Dense(units=1, activation="sigmoid")
model = Sequential([layer_1, layer_2, layer_3])
```

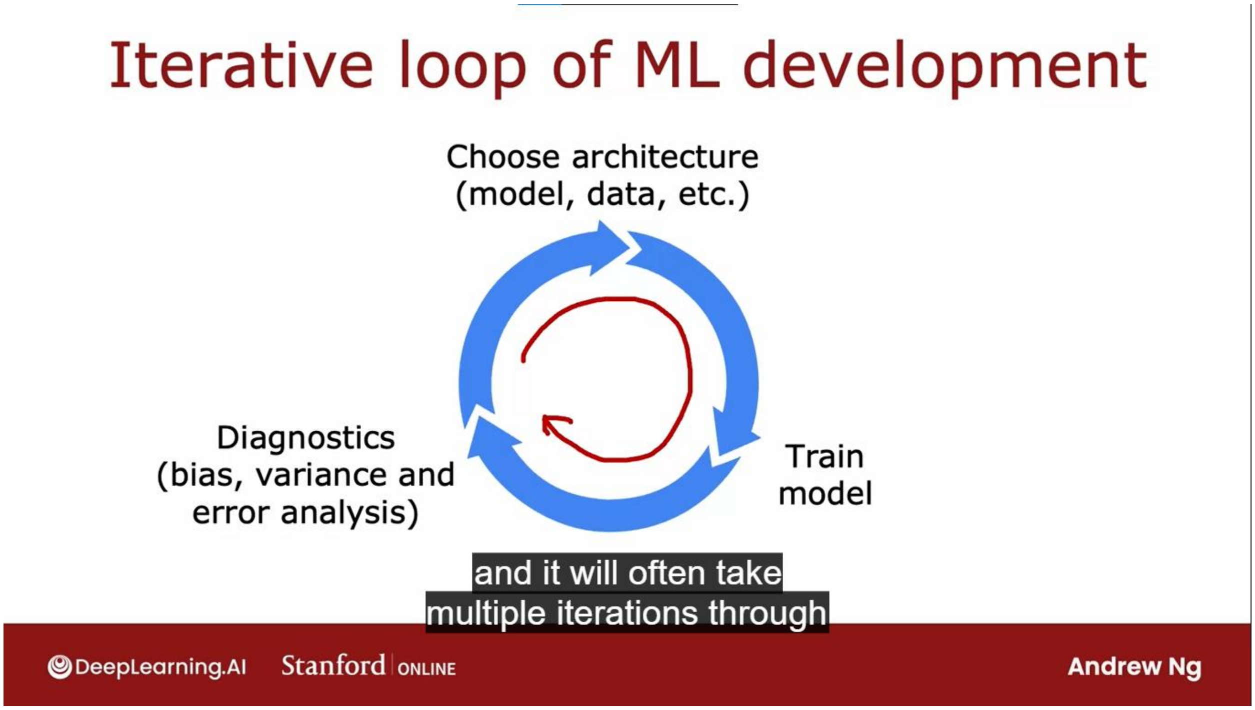
Regularized MNIST model

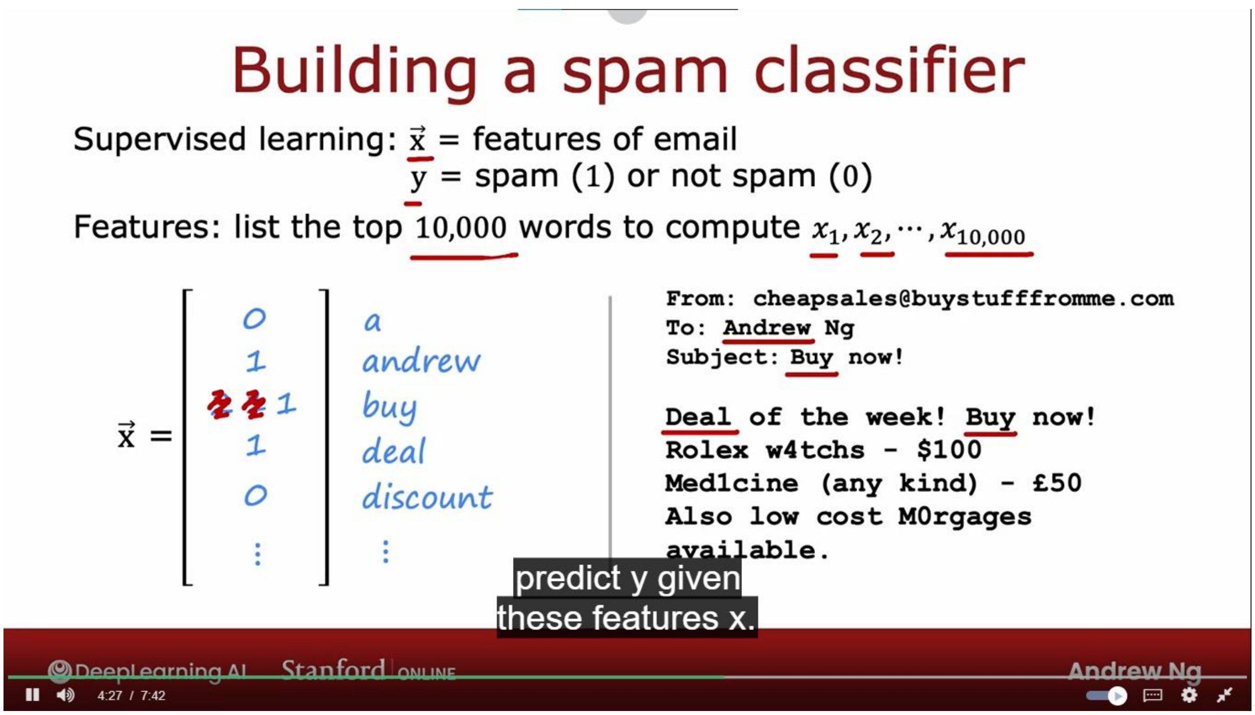
λ

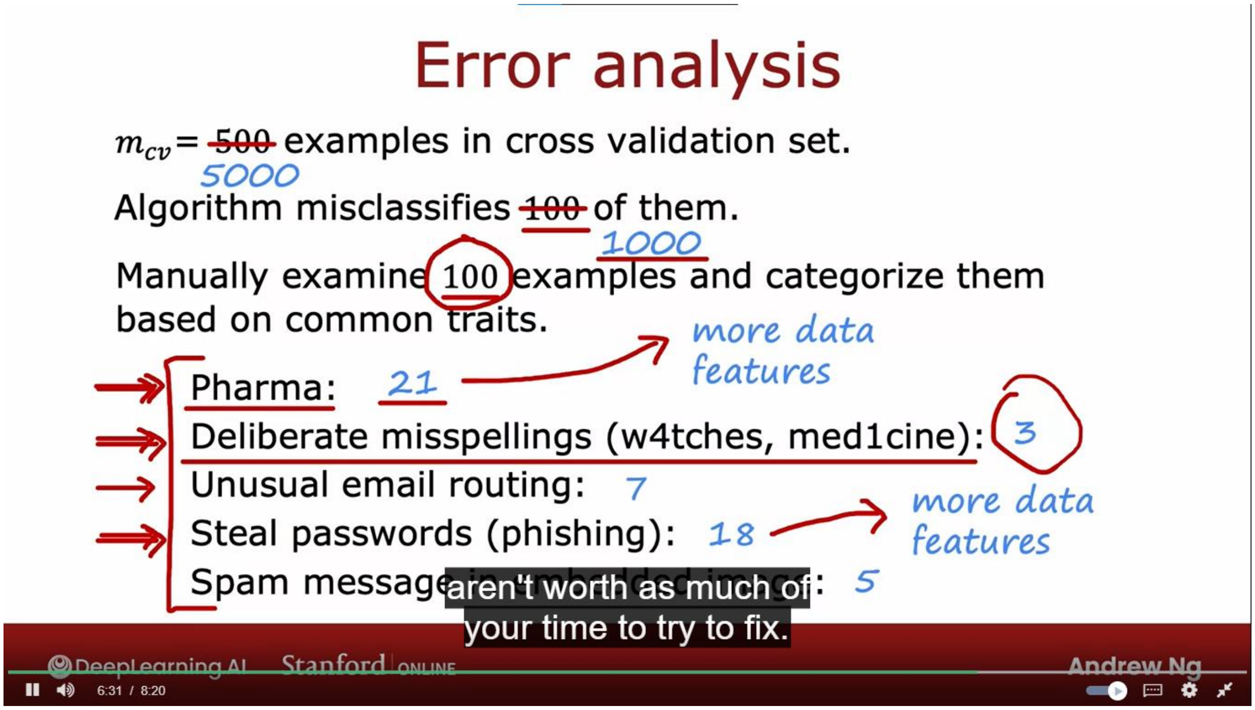
```
layer_1 = Dense(units=25, activation="relu", kernel_regularizer=L2(0.01))
layer_2 = Dense(units=15, activation="relu", kernel_regularizer=L2(0.01))
layer_3 = Dense(units=1, activation="sigmoid", kernel_regularizer=L2(0.01))
model = Sequential
```

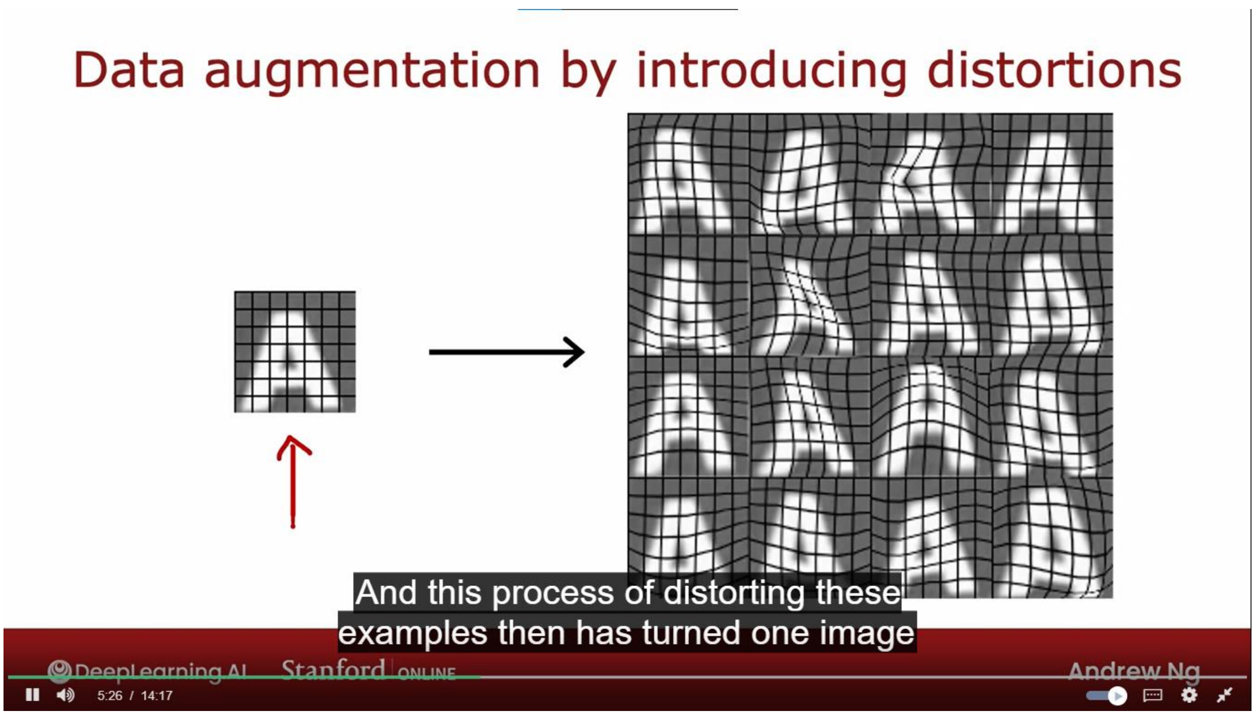
lets you choose different values of
lambda for different layers although for

DeepLearning.AI Stanford ONLINE Andrew Ng


Iterative loop of ML development
Choose architecture
(model, data, etc.)
Train
model
Diagnostics
(bias, variance and
error analysis)
and it will often take
multiple iterations through
DeepLearning.AI
Stanford ONLINE
Andrew Ng


Building a spam classifier
Supervised learning: $\vec{x}$ = features of email
y = spam (1) or not spam (0)
Features: list the top 10,000 words to compute $x_1, x_2, \cdots, x_{10,000}$
$\vec{x} =$
0 a
1 andrew
~~2~~ ~~2~~ 1 buy
1 deal
0 discount
From: cheapsales@buystufffromme.com
To: Andrew Ng
Subject: Buy now!
Deal of the week! Buy now!
Rolex w4tchs - $100
Med1cine (any kind) - £50
Also low cost M0rgages
available.
predict y given
these features x.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
4:27 / 7:42


Error analysis
m_{cv} = ~~500~~ examples in cross validation set.
5000
Algorithm misclassifies ~~100~~ of them.
1000
Manually examine 100 examples and categorize them based on common traits.
more data features
Pharma: 21
Deliberate misspellings (w4tches, med1cine): 3
Unusual email routing: 7
Steal passwords (phishing): 18
more data features
Spam message
: 5
aren't worth as much of
your time to try to fix.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
6:31 / 8:20


Data augmentation by introducing distortions
And this process of distorting these
examples then has turned one image
DeepLearning.AI
Stanford ONLINE
Andrew Ng
5:26 / 14:17

Data augmentation for speech

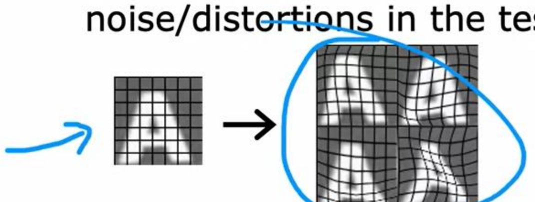
Speech recognition example

- Original audio (voice search: "What is today's weather?")
- \+ Noisy background: Crowd
- \+ Noisy background: Car
- \+ Audio on bad cellphone connection

And it turns out that if you
take these two audio clips,

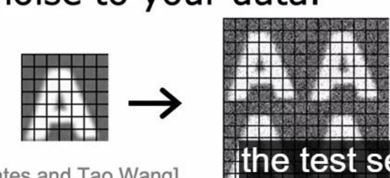
Data augmentation by introducing distortions

Distortion introduced should be representation of the type of noise/distortions in the test set.



Audio:
Background noise,
bad cellphone connection

Usually does not help to add purely random/meaningless noise to your data.



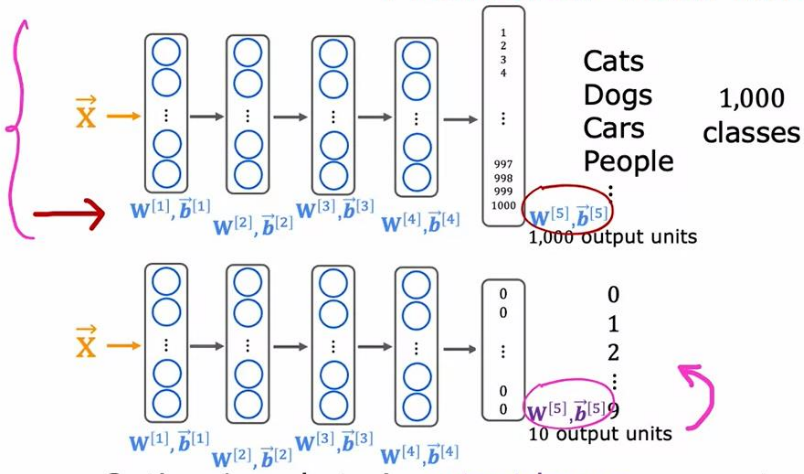
x_i = intensity (brightness) of pixel i

$x_i \leftarrow x_i$ + random noise

[Adam Coates and Tao Wang]

the test set because you don't often
get images like this in the test

Transfer learning



1, 2, 3, ..., 9

1 million images

supervised pretraining

fine tuning

Option 1: only train output layers parameters.

Option 2: train all parameters.

and will have posted

Transfer learning summary

1. Download neural network parameters pretrained on a large dataset with same input type (e.g., images, audio, text) as your application (or train your own).

1 million images

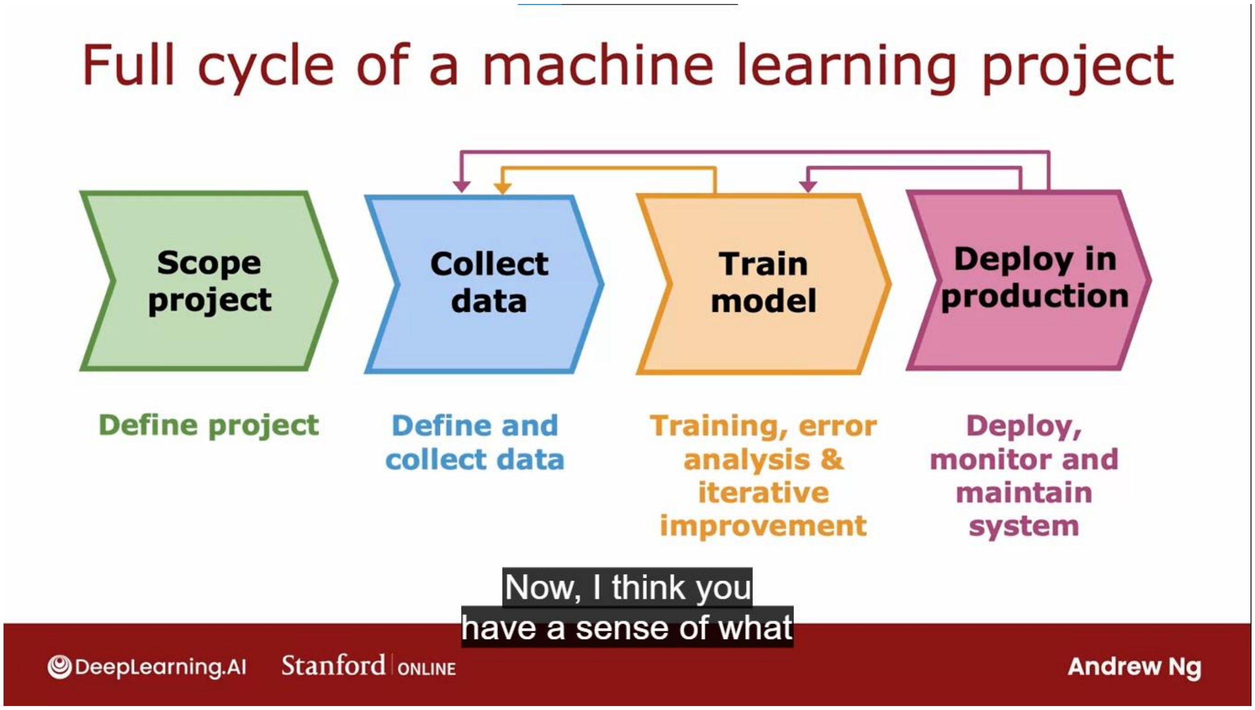
2. Further train (fine tune) the network on your own data.

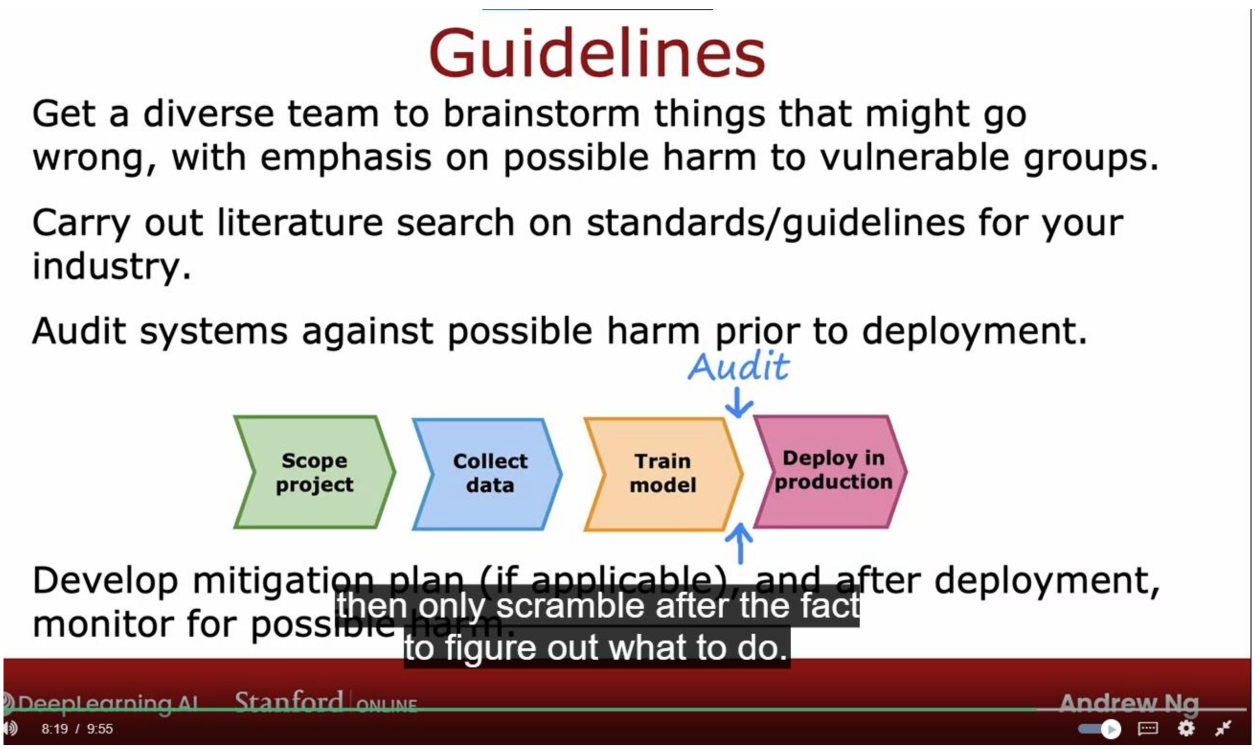
1000 images

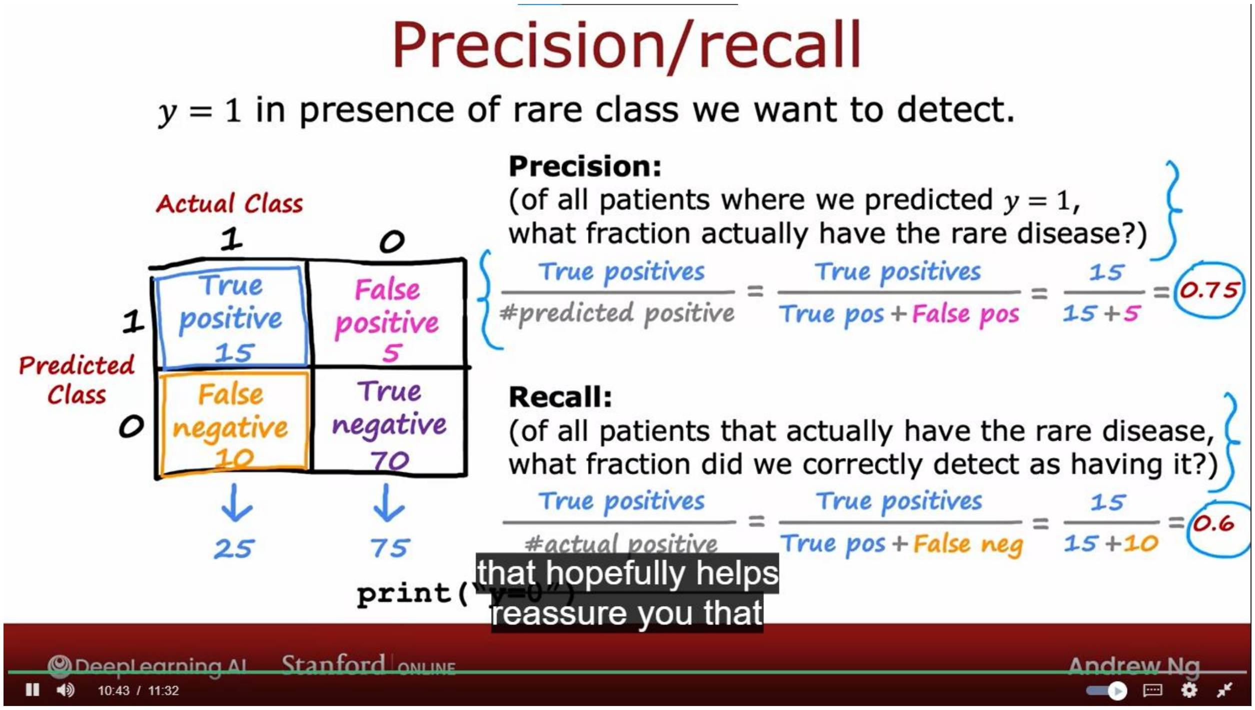
50 images

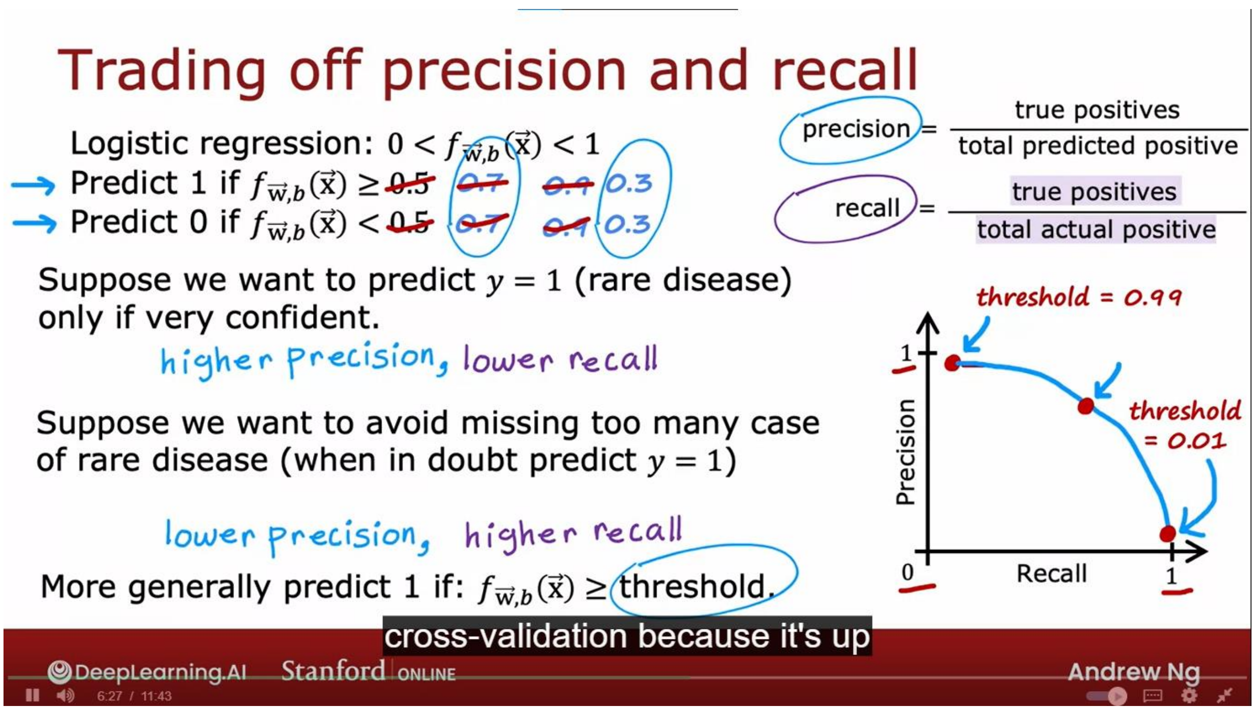
any single person
by themselves can.

DeepLearning.AI Stanford ONLINE Andrew Ng


Full cycle of a machine learning project
Scope project
Collect data
Train model
Deploy in production
Define project
Define and collect data
Training, error analysis & iterative improvement
Deploy, monitor and maintain system
Now, I think you
have a sense of what
DeepLearning.AI
Stanford ONLINE
Andrew Ng


Guidelines
Get a diverse team to brainstorm things that might go wrong, with emphasis on possible harm to vulnerable groups.
Carry out literature search on standards/guidelines for your industry.
Audit systems against possible harm prior to deployment.
Audit
Scope project
Collect data
Train model
Deploy in production
Develop mitigation plan (if applicable), and after deployment, monitor for possible harm.
then only scramble after the fact
to figure out what to do.
DeepLearning.AI
Stanford ONLINE
Andrew Ng
8:19 / 9:55


Precision/recall
y = 1 in presence of rare class we want to detect.
Actual Class
1
0
True positive 15
False positive 5
False negative 10
True negative 70
Predicted Class
1
0
25
75
print("y=0")
Precision:
(of all patients where we predicted y = 1, what fraction actually have the rare disease?)
True positives / #predicted positive = True positives / True pos + False pos = 15 / 15+5 = 0.75
Recall:
(of all patients that actually have the rare disease, what fraction did we correctly detect as having it?)
True positives / #actual positive = True positives / True pos + False neg = 15 / 15+10 = 0.6
that hopefully helps
reassure you that
DeepLearning.AI
Stanford ONLINE
Andrew Ng
10:43 / 11:32


Trading off precision and recall
Logistic regression: 0 < f w,b(x) < 1
Predict 1 if f w,b(x) ≥ ~~0.5~~ ~~0.7~~ ~~0.9~~ 0.3
Predict 0 if f w,b(x) < ~~0.5~~ ~~0.7~~ ~~0.9~~ 0.3
Suppose we want to predict y = 1 (rare disease) only if very confident.
higher precision, lower recall
Suppose we want to avoid missing too many case of rare disease (when in doubt predict y = 1)
lower precision, higher recall
More generally predict 1 if: f w,b(x) ≥ threshold.
precision = true positives / total predicted positive
recall = true positives / total actual positive
threshold = 0.99
threshold = 0.01
Precision
Recall
1
0
1
cross-validation because it's up
DeepLearning.AI
Stanford ONLINE
Andrew Ng
6:27 / 11:43

F1 score

How to compare precision/recall numbers?

	Precision (P)	Recall (R)	~~Average~~	F_1 score
Algorithm 1	0.5	0.4	~~0.45~~	0.444
Algorithm 2	0.7	0.1	~~0.4~~	0.175
Algorithm 3	0.02	1.0	~~0.501~~	0.0392

```
print("y=1")
```

$$\cancel{\text{Average} = \frac{P+R}{2}}$$

Harmonic mean

$$F_1 \text{ score} = \frac{1}{\frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)} = 2\frac{PR}{P+R}$$

But for the purposes of this class,