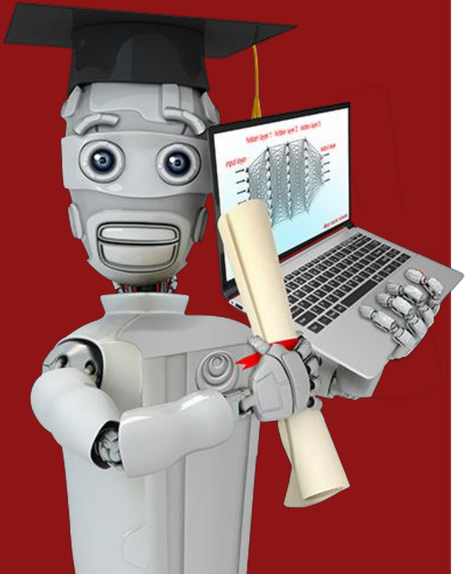
Copyright Notice

DeepLearning.AI

Stanford
ONLINE



Advice for applying machine learning

Deciding what to try next

Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

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

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

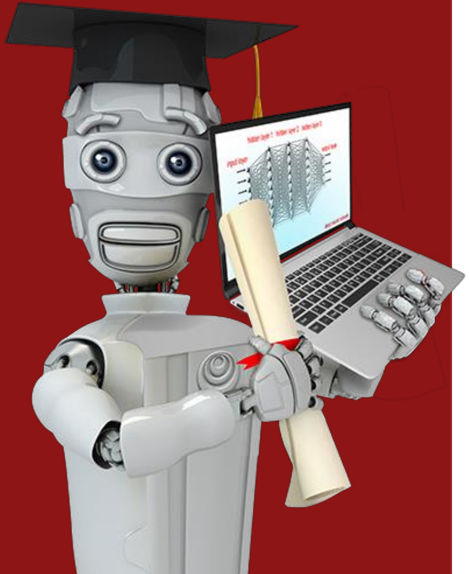
Machine learning diagnostic

Diagnostic: A test that you run to gain insight into what is/isn’t working with a learning algorithm, to gain guidance into improving its performance.

Diagnostics can take time to implement but doing so can be a very good use of your time.

DeepLearning.AI

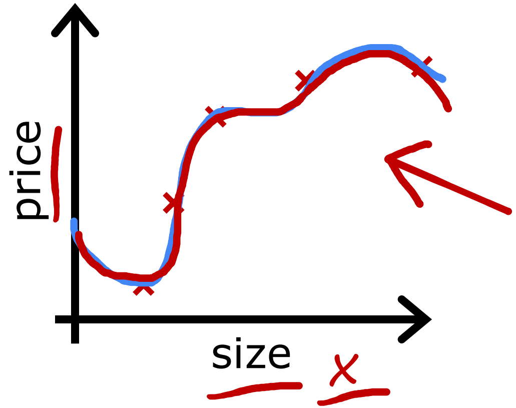
Stanford ONLINE



Evaluating and choosing models

Evaluating a model

Evaluating your model



→ Model fits the training data well but will fail to generalize to new examples not in the training set.

$W_n x^4$

$x = x_1$

$$f_{\vec{w},b}(\vec{x}) = w_1 x + w_2 x^2 + \cdots + w_n x^n + b$$

just x

→ x_1 = size in feet²

→ x_2 = no. of bedrooms

→ x_3 = no. of floors

→ x_4 = age of home in years

$f(\vec{x})$

Evaluating your model

Dataset:

size	price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

70%

training set

$$\begin{gathered}(x^{(1)}, y^{(1)}) \\ (x^{(2)}, y^{(2)}) \\ \vdots \\ (x^{(m_{train})}, y^{m_{train}})\end{gathered}$$

m_{train} = no. training examples = 7

30%

test set

$$\begin{gathered}(x_{test}^{(1)}, y_{test}^{(1)}) \\ \vdots \\ (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})\end{gathered}$$

m_{test} = no. test examples = 3

Train/test procedure for linear regression (with squared error cost)

Fit parameters by minimizing cost function $J(\vec{w}, b)$

$$\rightarrow J(\vec{w}, b) = \min_{\vec{w}, b} \left[\frac{1}{2m_{train}} \sum_{i=1}^{m_{train}} \underline{\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)} \right)^2} + \underbrace{\frac{\lambda}{2m_{train}} \sum_{j=1}^{n} w_j^2} \right] \leftarrow$$

Compute test error:

$$\underline{J_{test}\left(\boxed{\vec{w}, b}\right)} = \frac{1}{2m_{test}} \left[\sum_{i=1}^{m_{test}} \underline{\left(f_{\vec{w},b}\left(\vec{x}^{(i)}_{test}\right) - y^{(i)}_{test} \right)^2} \right] \qquad + \cancel{\sum w^2}$$

Compute training error:

$$\underline{J_{train}(\vec{w}, b)} = \frac{1}{\underline{2m_{train}}} \left[\underline{\sum_{i=1}^{m_{train}} \left(f_{\vec{w},b}\left(\vec{x}^{(i)}_{train}\right) - y^{(i)}_{train} \right)^2} \right]$$

Train/test procedure for linear regression (with squared error cost)

Fit parameters … $, b)$

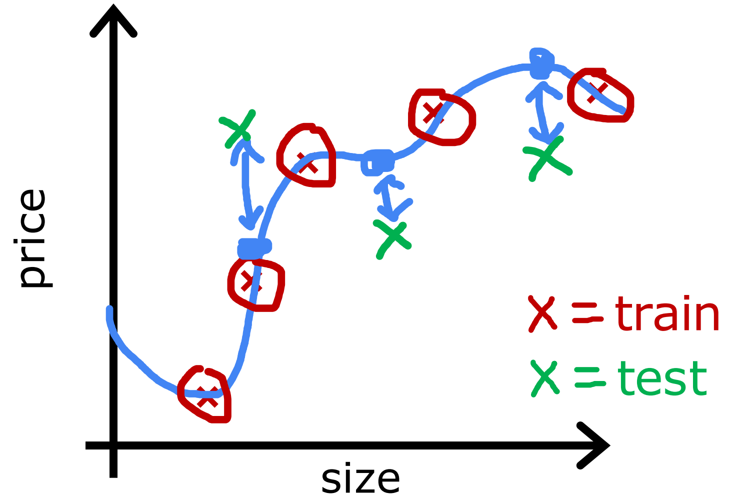
$\rightarrow J(\vec{w}, b) = \min_{\vec{w}, b}$ … $ain \sum_{j=1}^{n} w_j^2 \Big] \leftarrow$

Compute test e…

J_t … $^{(i)}_{est} \Big)^2 \Big]$ $+ \cancel{\sum w^2}$

Compute traini…

J_t … $- y_{train}^{(i)} \Big)^2 \Big]$



$J_{train}(\vec{w}, b)$ will be low

$J_{test}(\vec{w}, b)$ will be high $\leftarrow$

Train/test procedure for classification problem

0/1

Fit parameters by minimizing $\underline{J(\vec{\mathrm{w}}, b)}$ to find $\underline{\vec{\mathrm{w}}, b}$

E.g.,

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

Compute test error:

$$J_{test}(\vec{\mathrm{w}}, b) = -\frac{1}{m_{test}}\sum_{i=1}^{m_{test}}\left[\underline{y_{test}^{(i)}\log\left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{test}^{(i)}\right)\right) + (1-y_{test}^{(i)})\log\left(1-f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{test}^{(i)}\right)\right)}\right]$$

Compute train error:

$$J_{train}(\vec{\mathrm{w}}, b) = -\frac{1}{m_{train}}\sum_{i=1}^{m_{train}}\left[y_{train}^{(i)}\log\left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{train}^{(i)}\right)\right) + \left(1-y_{train}^{(i)}\right)\log\left(1-f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{train}^{(i)}\right)\right)\right]$$

Train/test procedure for classification problem

0/1

Fit parameters by minimizing $J(\vec{w},b)$ to find $\vec{w},b$

E.g.,

$$J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m}\Big[y^{(i)}\log\Big(f_{\vec{w},b}($$

Compute test error:

$$J_{test}(\vec{w},b) = -\frac{1}{m_{test}}\sum_{i=1}^{m_{test}}\Big[y$$

Compute train error:

$$J_{train}(\vec{w},b) = -\frac{1}{m_{train}}\sum_{i=1}^{m_{train}}$$

fraction of the test set and the fraction of the train set that the algorithm has misclassified.

$$\hat{y} = \begin{cases} 1 \text{ if } f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \geq 0.5 \\ 0 \text{ if } f_{\vec{w},b}\left(\vec{x}^{(i)}\right) < 0.5 \end{cases}$$

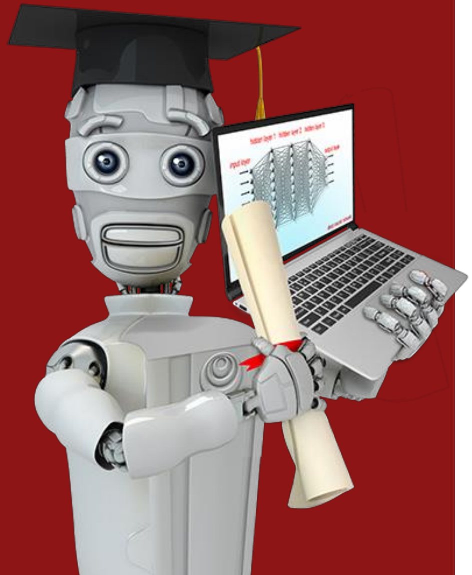
count $\hat{y} \neq y$

$J_{test}(\vec{w},b)$ is the fraction of the test set that has been misclassified.

$J_{train}(\vec{w},b)$ is the fraction of the train set that has been misclassified.

DeepLearning.AI

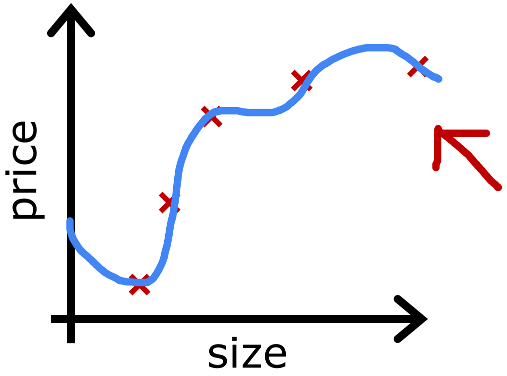
Stanford ONLINE



Evaluating and choosing models

Model selection and training/cross validation/test sets

Model selection (choosing a model)



$$f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x^2 + w_3x^3 + w_4x^4 + b$$

Once parameters $\vec{w}, b$ are fit to the training set, the training error $\underline{J_{train}(\vec{w}, b)}$ is likely lower than the actual generalization error.

$\underline{J_{test}(\vec{w}, b)}$ is better estimate of how well the model will generalize to new data than $J_{train}(\vec{w}, b)$.

Model selection (choosing a model)

d=1 1. $f_{\vec{w},b}(\vec{x}) = w_1 x_1 + b \rightarrow w^{<1>}, b^{<1>} \rightarrow J_{test}(w^{<1>}, b^{<1>})$

d=2 2. $f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x^2 + b \rightarrow w^{<2>}, b^{<2>} \rightarrow J_{test}(w^{<2>}, b^{<2>})$

d=3 3. $f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x^2 + w_3 x^3 + b \rightarrow w^{<3>}, b^{<3>} \rightarrow J_{test}(w^{<3>}, b^{<3>})$

$\vdots$

d=10 10. $f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x^2 + \cdots + w_{10} x^{10} + b \rightarrow J_{test}(w^{<10>}, b^{<10>})$

Choose $\boxed{w_1 x_1 + \cdots + w_5 x^5 + b}$ d=5 $J_{test}(w^{<5>}, b^{<5>})$

How well does the model perform? Report test set error $\underline{J_{test}(w^{<5>}, b^{<5>})}$?

The problem is $\underline{J_{test}(w^{<5>}, b^{<5>})}$ is likely to be an optimistic estimate of generalization error. Ie: An extra parameter d (degree of polynomial) was chosen using the test set.

w,b

Training/cross validation/test set

validation set

development set

→ dev set

size	price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

training set
60%

$$\left(x^{(1)}, y^{(1)}\right)$$
$$\vdots$$
$$\left(x^{(m_{train})}, y^{(m_{train})}\right)$$

$m_{train} = 6$

cross validation
20%

$$\left(x_{cv}^{(1)}, y_{cv}^{(1)}\right)$$
$$\vdots$$
$$\left(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})}\right)$$

$m_{cv} = 2$

test set
20%

$$\left(x_{test}^{(1)}, y_{test}^{(1)}\right)$$
$$\vdots$$
$$\left(x_{test}^{(m_{test})}, y_{test}^{(m_{test})}\right)$$

$m_{test} = 2$

Training/cross validation/test set

Training error: $$J_{train}(\vec{\mathrm{w}},b) = \frac{1}{2m_{train}}\left[\sum_{i=1}^{m_{train}}\left(f_{\vec{\mathrm{w}},b}(\vec{\mathrm{x}}^{(i)}) - y^{(i)}\right)^2\right]$$

Cross validation error: $$J_{cv}(\vec{\mathrm{w}},b) = \frac{1}{\underline{2m_{cv}}}\left[\underline{\sum_{i=1}^{m_{cv}}\left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{cv}^{(i)}\right) - y_{cv}^{(i)}\right)^2}\right]$$ (validation error, dev error)

Test error: $$J_{test}(\vec{\mathrm{w}},b) = \frac{1}{2m_{test}}\left[\sum_{i=1}^{m_{test}}\left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{test}^{(i)}\right) - y_{test}^{(i)}\right)^2\right]$$

Model selection

d=1 1. $f_{\vec{w},b}(\vec{x}) = w_1x_1 + b$ → $w^{<1>},b^{<1>}$ → $J_{cv}(w^{<1>},b^{<1>})$

d=2 2. $f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x^2 + b$ → → $J_{cv}(w^{<2>},b^{<2>})$

d=3 3. $f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x^2 + w_3x^3 + b$

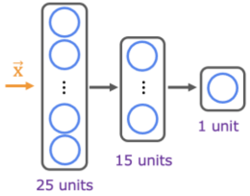
⋮ ⋮ ⋮

d=10 10. $f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x^2 + \cdots + w_{10}x^{10} + b$ $J_{cv}(w^{<10>},b^{<10>})$

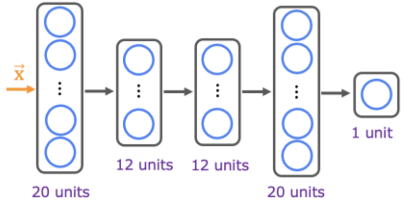
→ Pick $w_1x_1 + \cdots + w_4x^4 + b$ $(J_{cv}(w^{<4>},b^{<4>}))$

Estimate generalization error using test the set: $\underline{J_{test}(w^{<4>},b^{<4>})}$

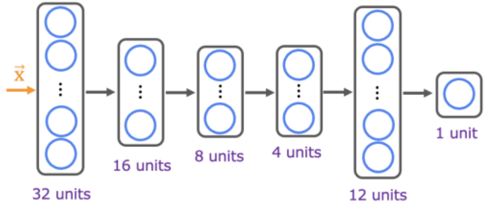
Model selection – choosing a neural network architecture

1. 

$w^{(1)}, b^{(1)}$ $\underline{J_{cv}(\mathbf{W}^{(1)}, \mathbf{B}^{(1)})}$

→ 2. 

$w^{(2)}, b^{(2)}$ $\boxed{J_{cv}(\mathbf{W}^{(2)}, \mathbf{B}^{(2)})}$

3. 

$w^{(3)}, b^{(3)}$ $J_{cv}(\mathbf{W}^{(3)}, \mathbf{B}^{(3)})$

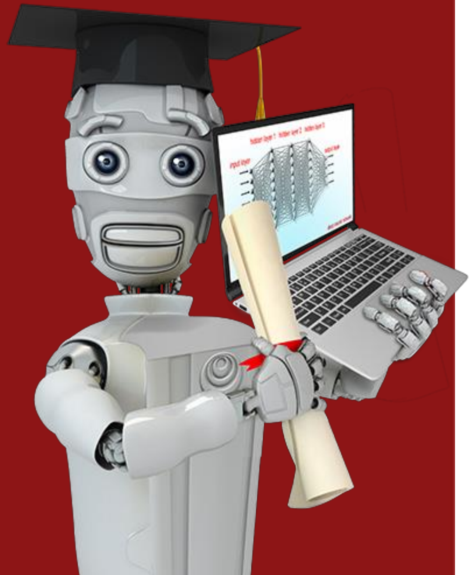
Train, CV

Pick $\mathbf{W}^{(2)}, \mathbf{B}^{(2)}$

Estimate generalization error using the test set: $\boxed{\underline{J_{test}}(\mathbf{W}^{(2)}, \mathbf{B}^{(2)})}$

DeepLearning.AI

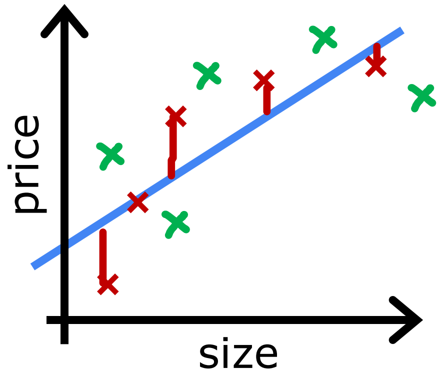
Stanford
ONLINE



Bias and variance

Diagnosing bias and variance

Bias/variance



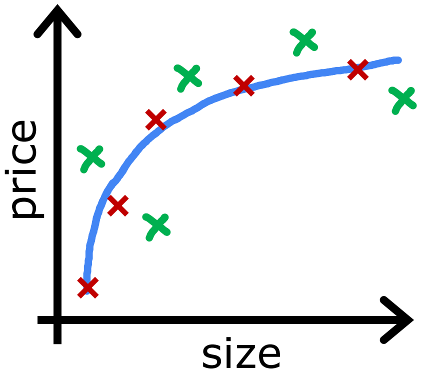
$$f_{\vec{w},b}(x) = w_1x + b$$

→ High bias (underfit)

$d = 1$

J_{train} is high

J_{cv} is high



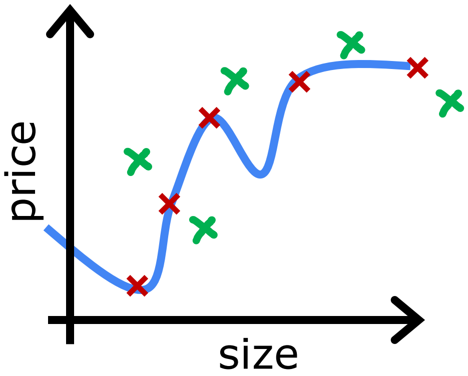
$$f_{\vec{w},b}(x) = w_1x + w_2x^2 + b$$

"Just right"

$d = 2$

J_{train} is low

J_{cv} is low



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

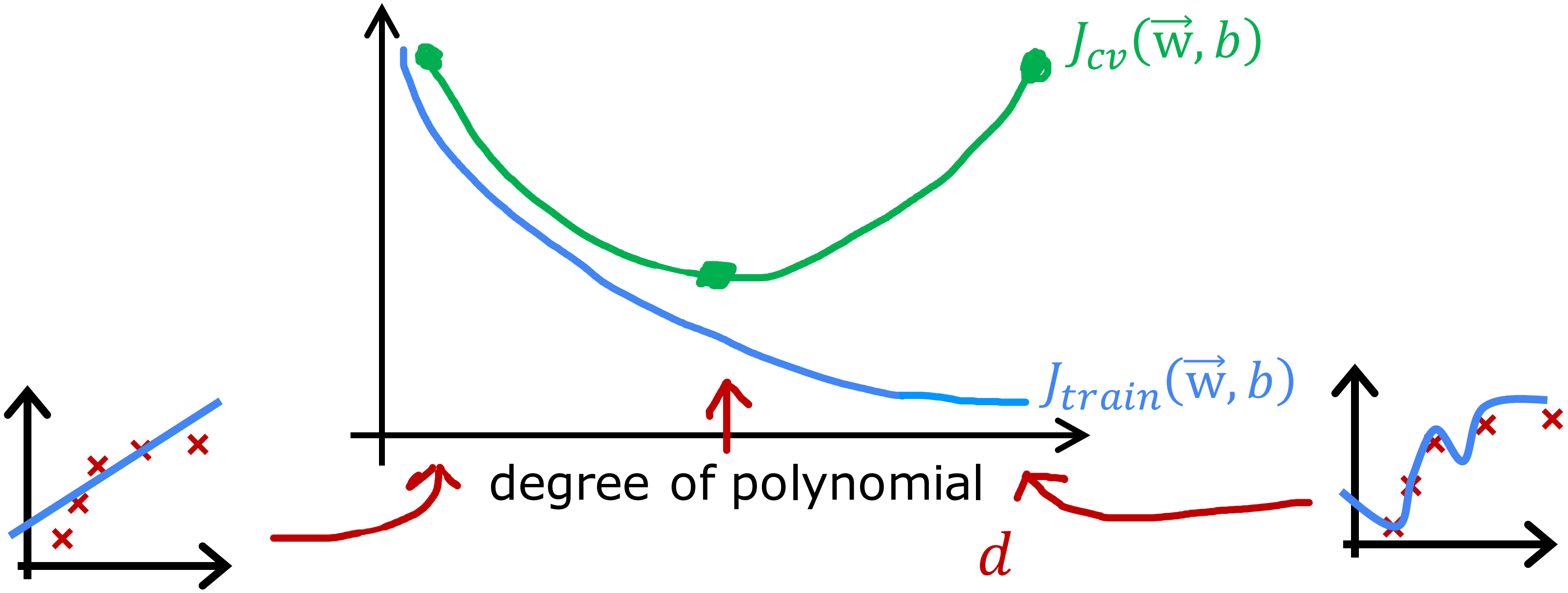
High variance (overfit)

$d = 4$

J_{train} is low

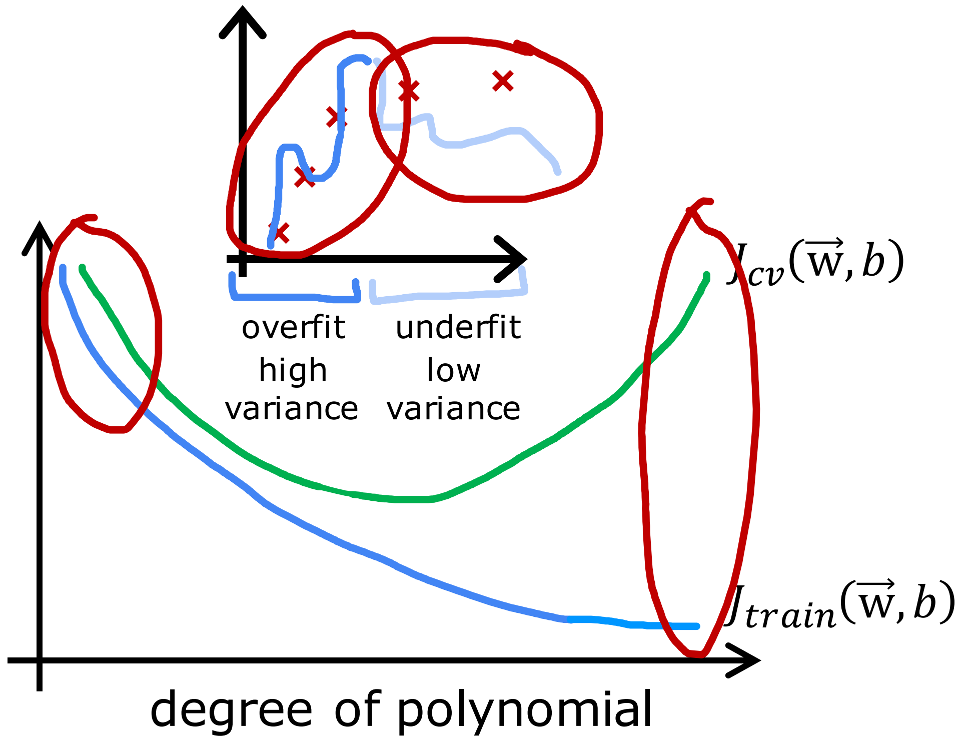
J_{cv} is high

Understanding bias and variance



Diagnosing bias and variance

How do you tell if your algorithm has a bias or variance problem?



High bias (underfit)

J_{train} will be high

($J_{train} \approx J_{cv}$)

$>$

High variance (overfit)

$\gg$

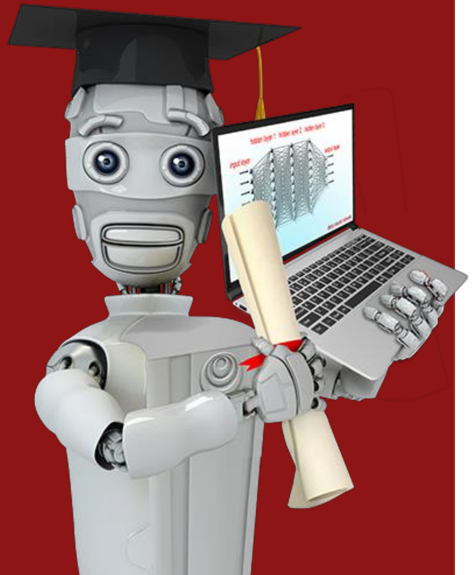
$J_{cv} \gg J_{train}$

(J_{train} may be be low)

High bias and high variance

J_{train} will be high

and $J_{cv} \gg J_{train}$



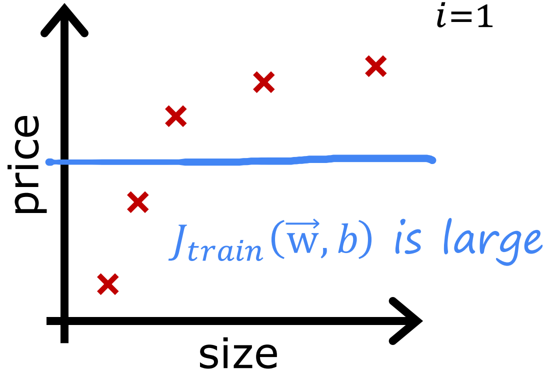
Bias and variance

Regularization and bias/variance

Linear regression with regularization

Model: $f_{\vec{w},b}(x) = \boxed{\underline{w_1}x + \underline{w_2}x^2 + \underline{w_3}x^3 + \underline{w_4}x^4 + b}$

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



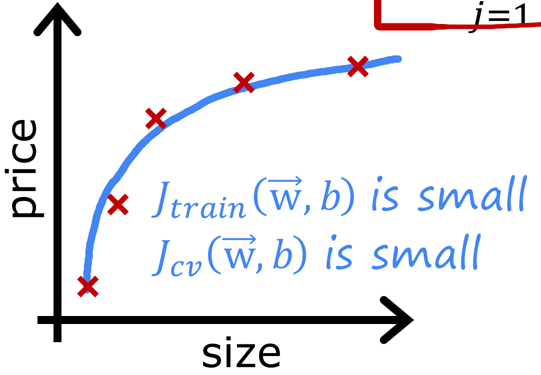
$J_{train}(\vec{w},b)$ is large

Large λ

High bias (underfit)

$\underline{\lambda = 10{,}000}$ $w_1 \approx 0, w_2 \approx 0$

$f_{\vec{w},b}(\vec{x}) \approx b$

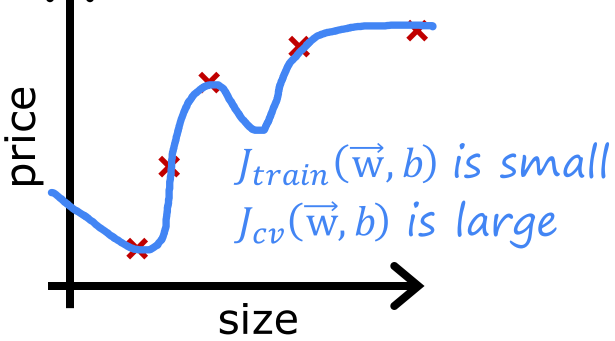


$J_{train}(\vec{w},b)$ is small

$J_{cv}(\vec{w},b)$ is small

Intermediate λ

λ



$J_{train}(\vec{w},b)$ is small

$J_{cv}(\vec{w},b)$ is large

Small λ

High variance (overfit)

$\lambda = 0$

Choosing the regularization parameter λ

Model: $f_{\vec{w},b}(x) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$

$\rightarrow$ 1. Try $\lambda = 0$ $\rightarrow$ $\min\limits_{\vec{w},b} J(\vec{w},b)$ $\rightarrow$ $w^{<1>}, b^{<1>}$ $\rightarrow$ $J_{cv}(w^{<1>}, b^{<1>})$

$\rightarrow$ 2. Try $\lambda = 0.01$ $\rightarrow$ $\rightarrow$ $w^{<2>}, b^{<2>}$ $\rightarrow$ $J_{cv}(w^{<2>}, b^{<2>})$

$\rightarrow$ 3. Try $\lambda = 0.02$ $\rightarrow$ $J_{cv}(w^{<3>}, b^{<3>})$

$\rightarrow$ 4. Try $\lambda = 0.04$

$\rightarrow$ 5. $\boxed{\text{Try } \lambda = 0.08}$ $J_{cv}(w^{<5>}, b^{<5>})$

$\vdots$

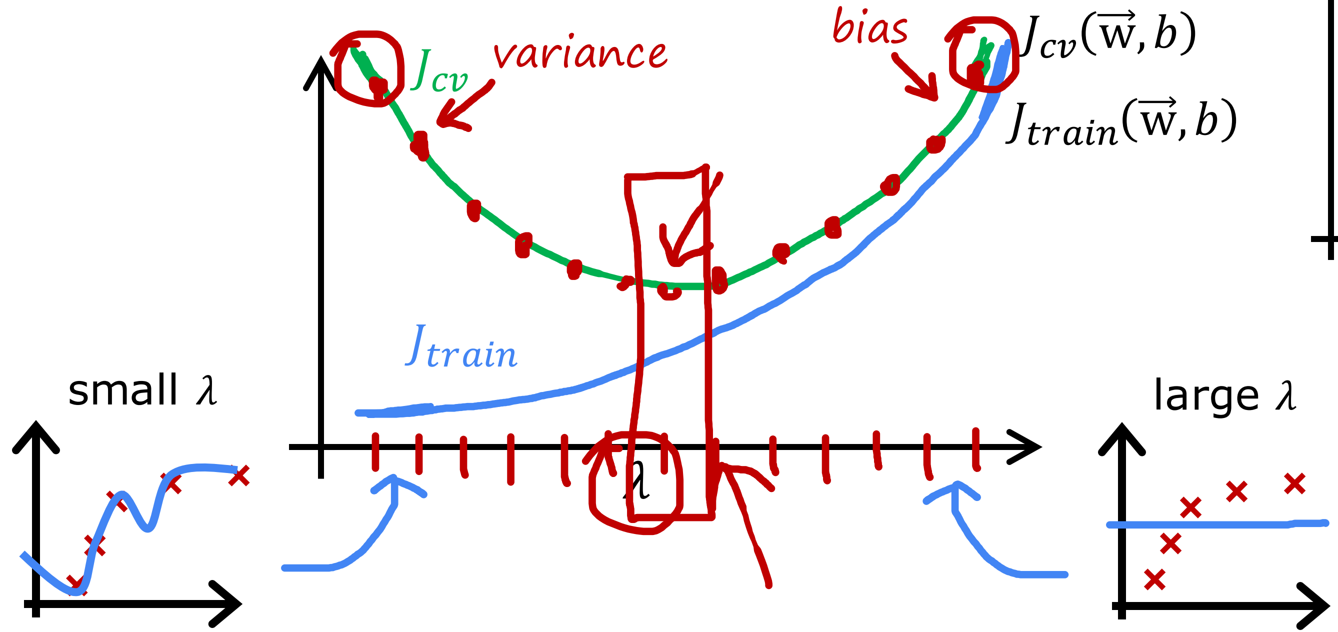
$\rightarrow$ 12. Try $\lambda \approx \underline{10}$ $\rightarrow$ $w^{<12>}, b^{<12>}$ $\rightarrow$ $J_{cv}(w^{<12>}, b^{<12>})$

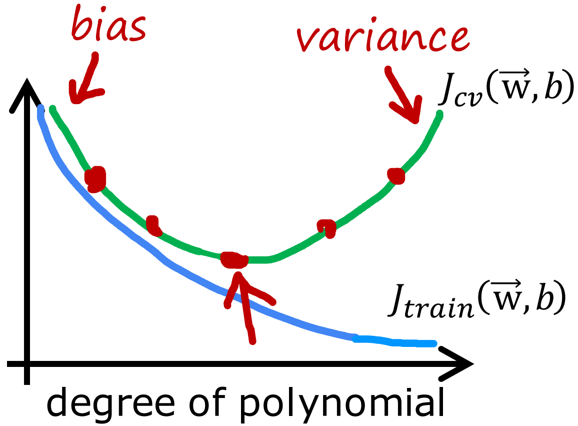
Pick $\underline{w^{<5>}, b^{<5>}}$

Report test error: $\underline{J_{test}(w^{<5>}, b^{<5>})}$

Bias and variance as a function of regularization parameter λ

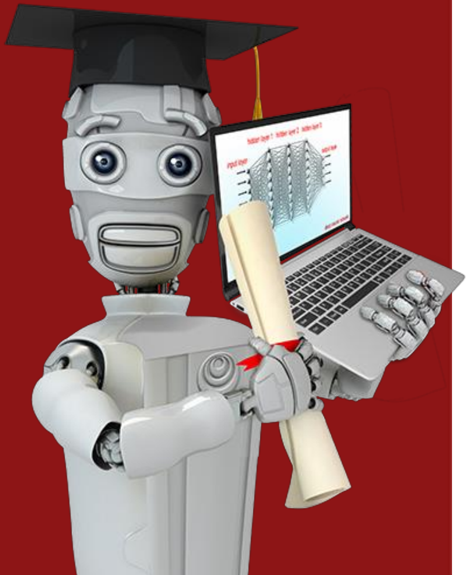
$$J(\vec{w},b) = \boxed{\frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2} + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$





DeepLearning.AI

Stanford
ONLINE



Bias and variance

Establishing a baseline level of performance

Speech recognition example

Human level performance : 10.6%

Training error J_{train} : 10.8%

Cross validation error J_{cv} : 14.8%

0.2%

4.0%

Establishing a baseline level of performance

What is the level of error you can reasonably hope to get to?

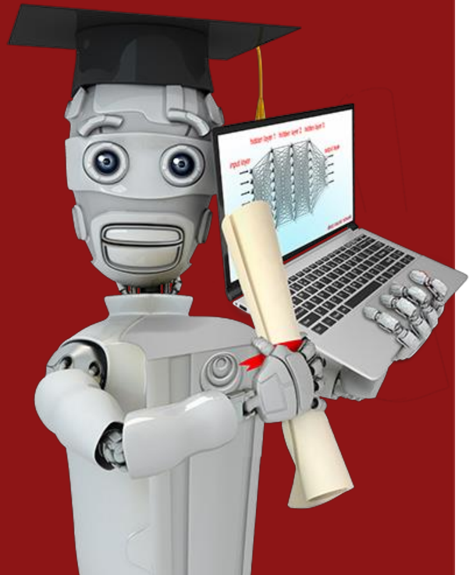
- Human level performance
- Competing algorithms performance
- Guess based on experience

Bias/variance examples

Baseline performance	: 10.6%	10.6%	10.6%
	0.2%	4.4%	4.4%
Training error (J_{train})	: 10.8%	15.0%	15.0%
	4.0%	0.5%	4.7%
Cross validation error (J_{cv})	: 14.8%	15.5%	19.7%
	high variance	high bias	high bias high variance

DeepLearning.AI

Stanford
ONLINE



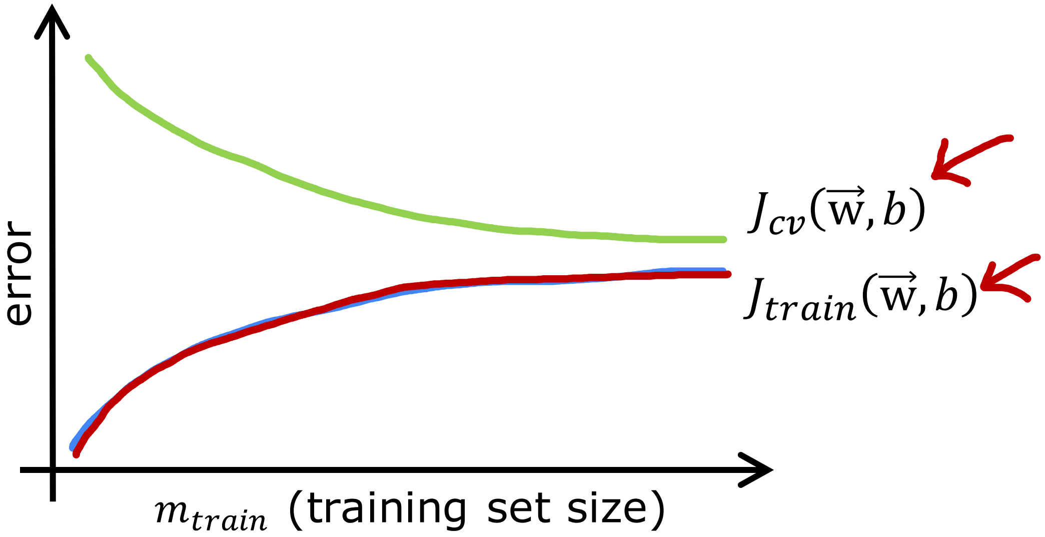
Bias and variance

Learning curves

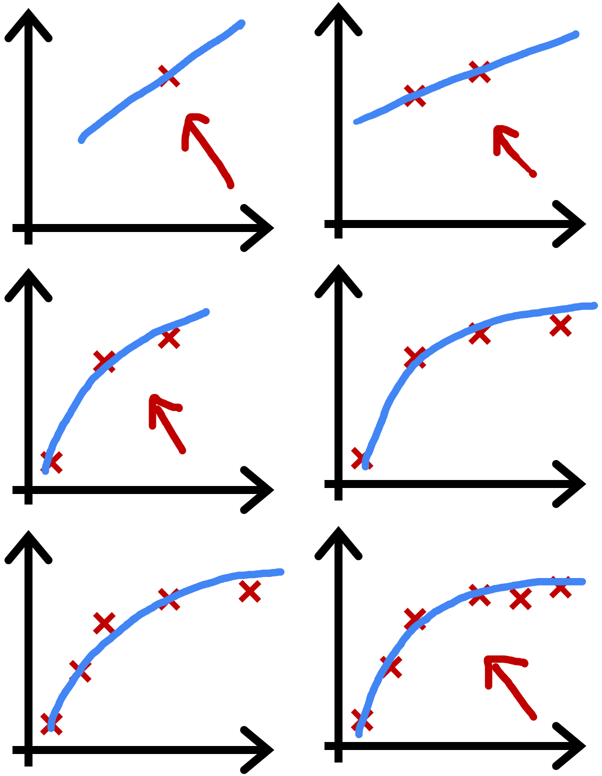
Learning curves

J_{train} = training error

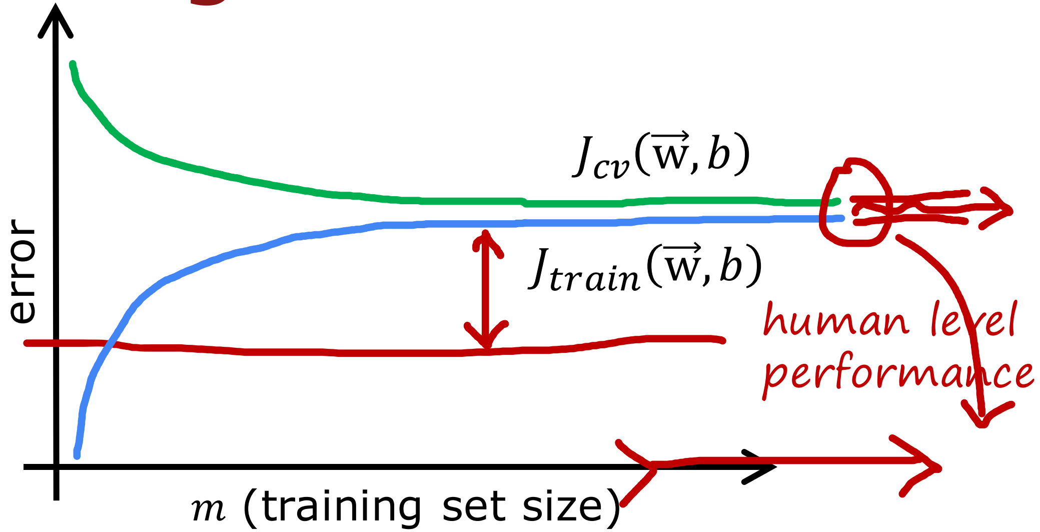
J_{cv} = cross validation error



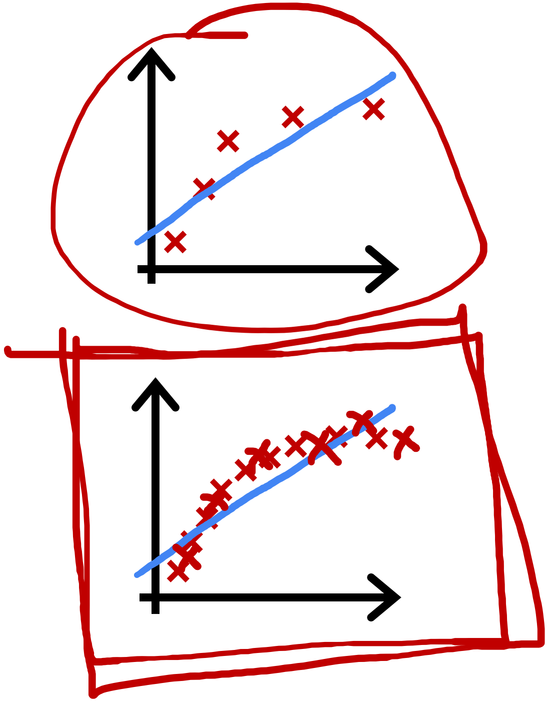
$$f_{\vec{w},b}(x) = w_1 x + w_2 x^2 + b$$



High bias

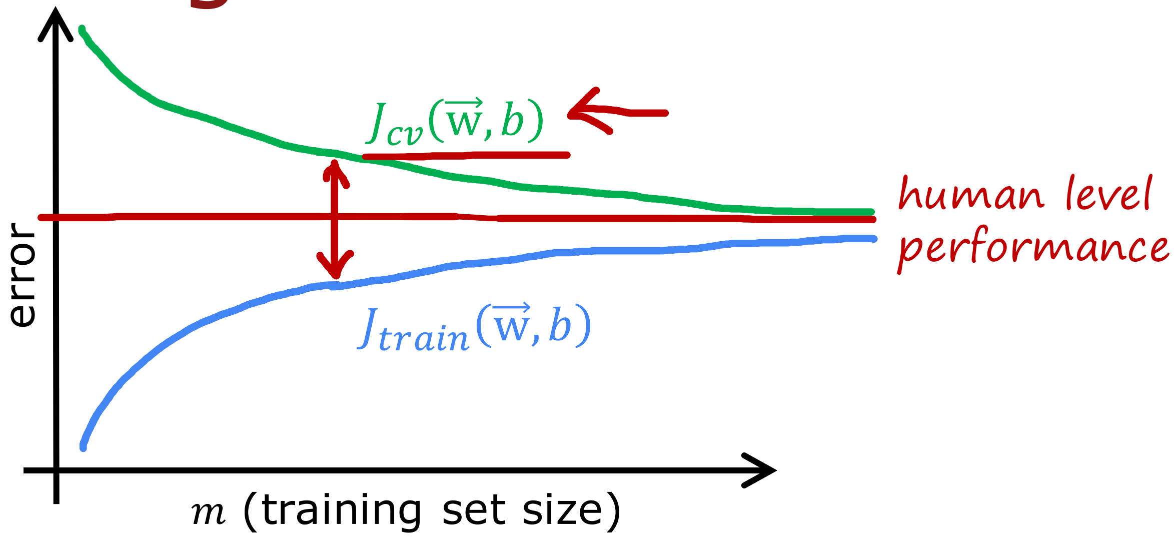


$f_{\vec{w},b}(x) = w_1 x + b$



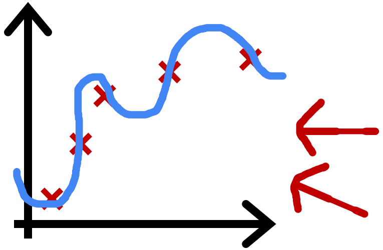
if a learning algorithm suffers from high bias, getting more training data will not (by itself) help much.

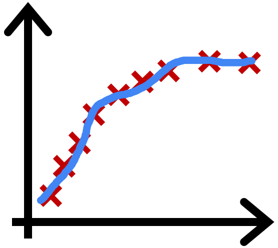
High variance



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

(with small λ)

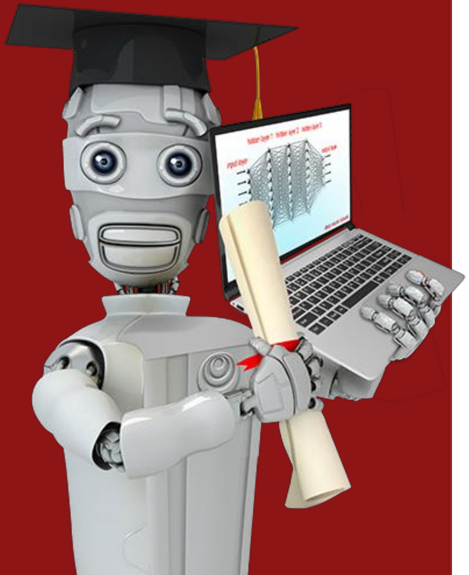




if a learning algorithm suffers from high variance, getting more training data is likely to help.

DeepLearning.AI

Stanford
ONLINE



Bias and variance

Deciding what to try next revisited

Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

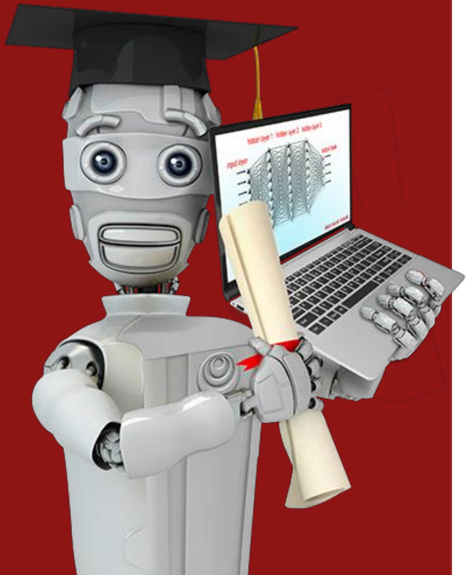
$$J(\vec{w},b) = \underbrace{\frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)^2}_{} + \underbrace{\frac{\boxed{\lambda}}{2m}\sum_{j=1}^{n} w_j^2}_{}$$

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

DeepLearning.AI

Stanford
ONLINE



Bias and variance

Bias/variance and neural networks

The bias variance tradeoff

$f_{\vec{w},b}(x) = w_1x + b$

$f_{\vec{w},b}(x) = w_1x + w_2x^2 + b$

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

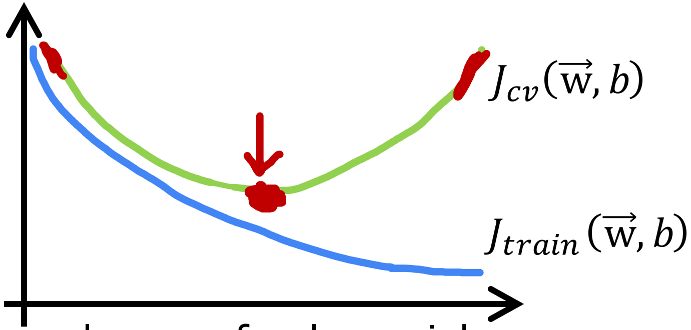
Simple model

High bias

Complex model

High variance

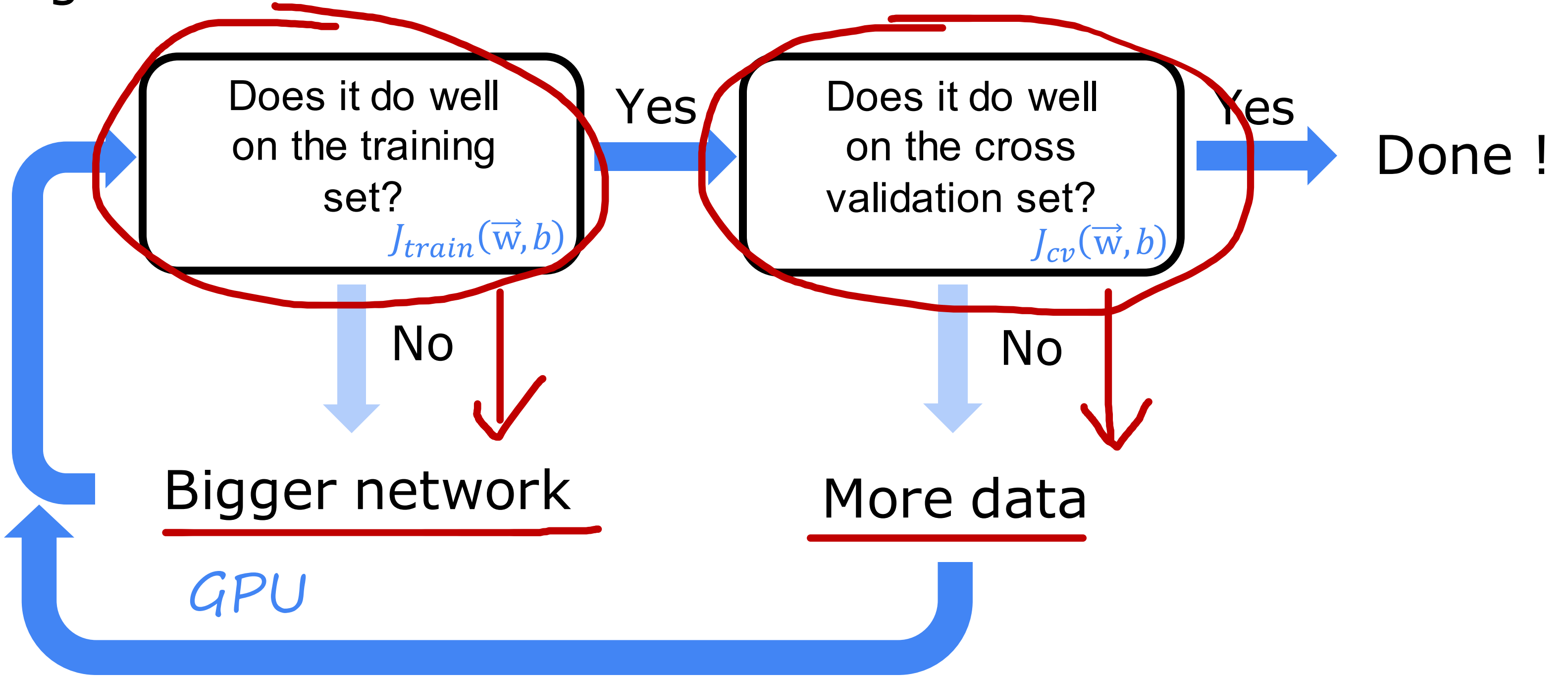
tradeoff



degree of polynomial

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

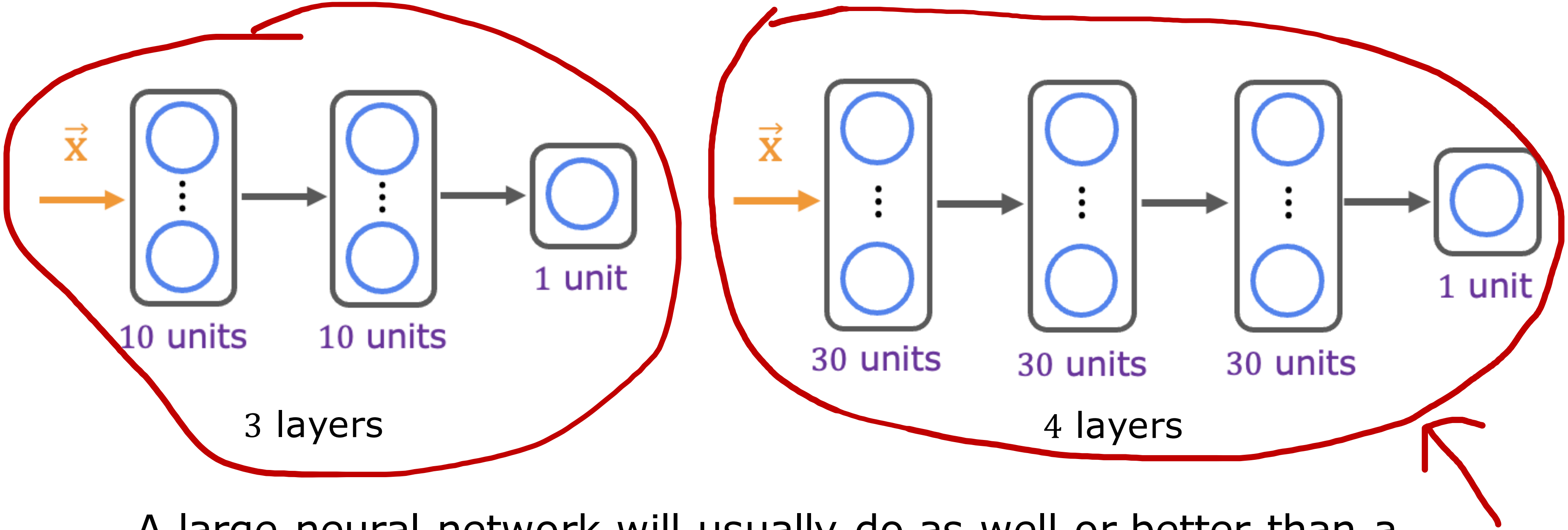
Bigger network

No

More data

GPU

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.

Neural network regularization

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

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])
```

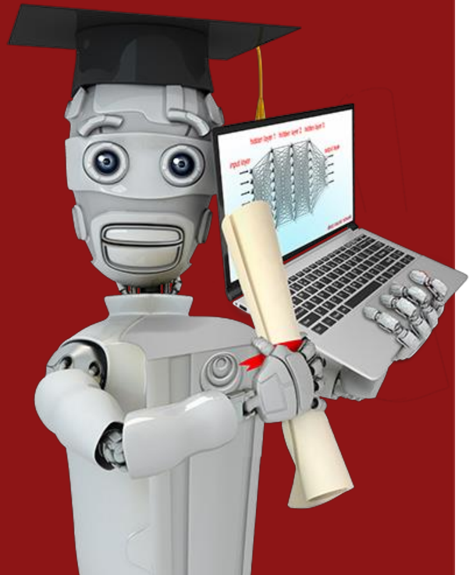
λ

Regularized MNIST model

```
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([layer_1, layer_2, layer_3])
```

DeepLearning.AI

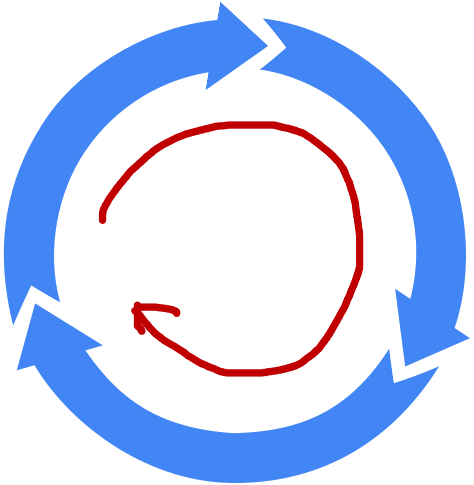
Stanford
ONLINE



Machine learning development process

Iterative loop of ML development

Iterative loop of ML development



Choose architecture
(model, data, etc.)

Train
model

Diagnostics
(bias, variance and
error analysis)

Spam classification example

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.

From: Alfred Ng
To: Andrew Ng
Subject: Christmas dates?

Hey Andrew,
Was talking to Mom about plans
for Xmas. When do you get off
work. Meet Dec 22?
Alf

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 $x_1, x_2, \cdots, x_{10{,}000}$

$$\vec{x} = \begin{array}{cl} \begin{bmatrix} 0 \\ 1 \\ \cancel{2}\;\cancel{2}\;1 \\ 1 \\ 0 \\ \vdots \end{bmatrix} & \begin{array}{l} a \\ andrew \\ buy \\ deal \\ discount \\ \vdots \end{array} \end{array}$$

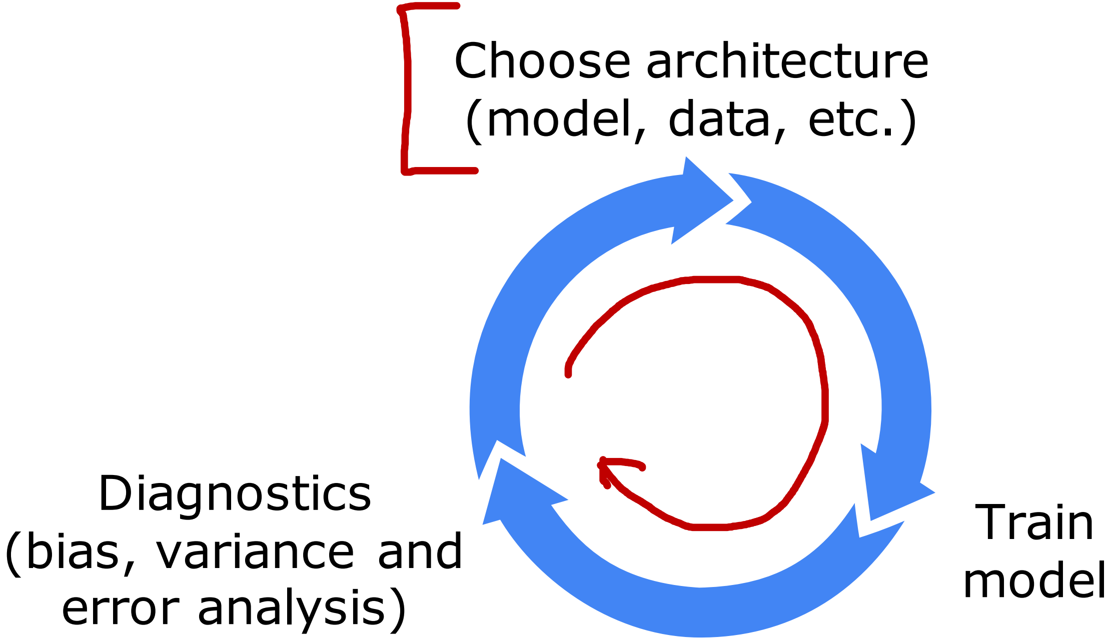
```
From: cheapsales@buystufffromme.com
To: Andrew Ng
Subject: Buy now!

Deal of the week! Buy now!
Rolex w4tchs - $100
Med1cine (any kind) - £50
Also low cost M0rgages
available.
```

Building a spam classifier

How to try to reduce your spam classifier’s error?

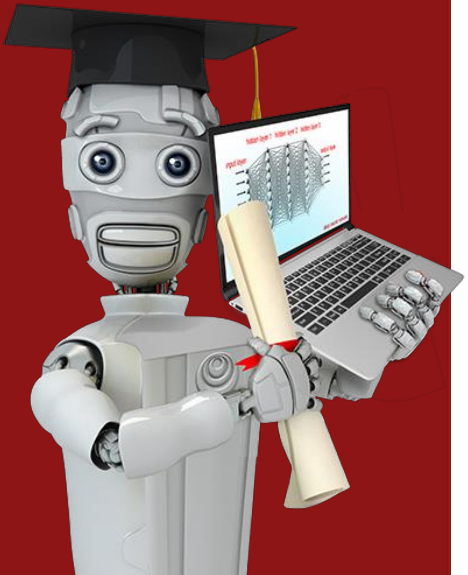
- Collect more data. E.g., “Honeypot” project.
- Develop sophisticated features based on email routing (from email header).
- Define sophisticated features from email body. E.g., should “discounting” and ”discount” be treated as the same word.
- Design algorithms to detect misspellings. E.g., w4tches, med1cine, m0rtgage.

Iterative loop of ML development



DeepLearning.AI

Stanford
ONLINE



Machine learning development process

Error analysis

Error analysis

m_{cv}= ~~500~~ 5000 examples in cross validation set.

Algorithm misclassifies ~~100~~ 1000 of them.

Manually examine 100 examples and categorize them based on common traits.

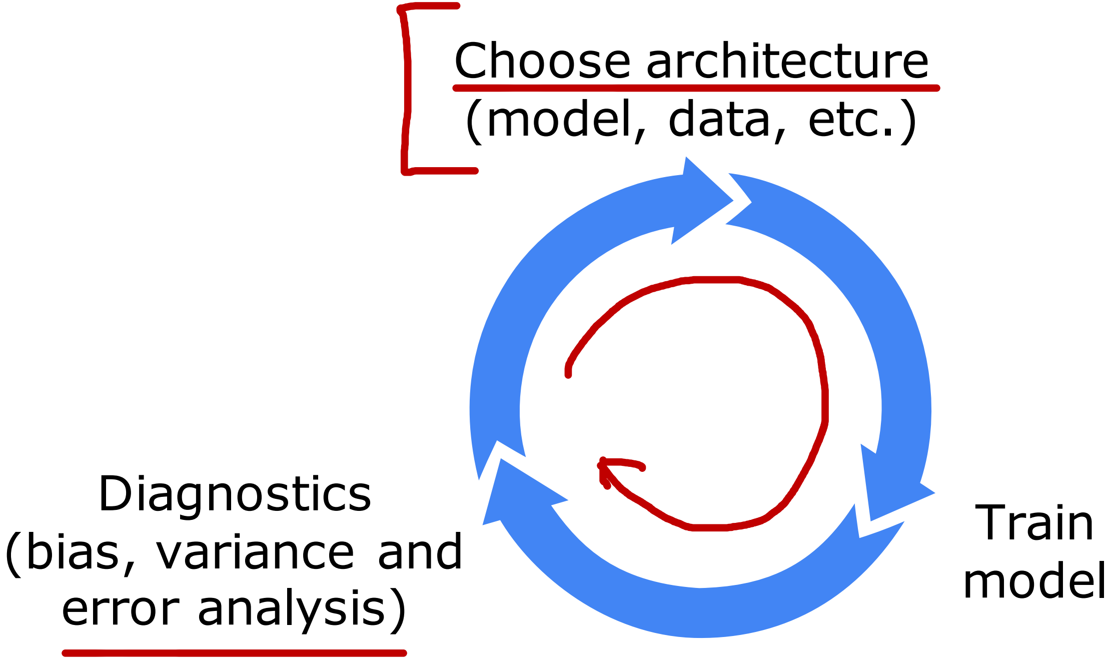
Pharma:	21	more data features
Deliberate misspellings (w4tches, med1cine):	3	
Unusual email routing:	7	
Steal passwords (phishing):	18	more data features
Spam message in embedded image:	5	

Building a spam classifier

How to try to reduce your spam classifier’s error?

- Collect more data. E.g., “Honeypot” project.
- Develop sophisticated features based on email routing (from email header).
- Define sophisticated features from email body. E.g., should “discounting” and ”discount” be treated as the same word.
- Design algorithms to detect misspellings. E.g., w4tches, med1cine, m0rtgage.

Iterative loop of ML development



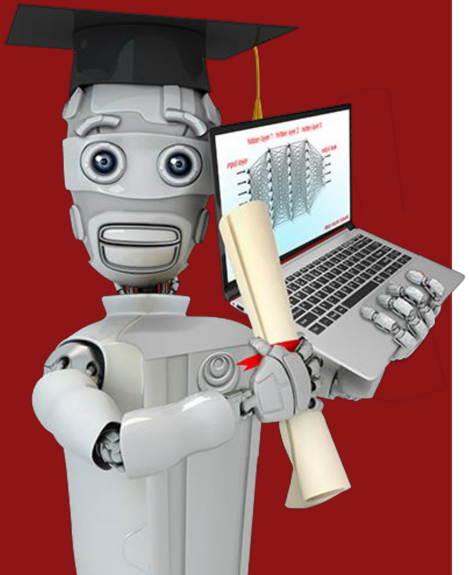
Choose architecture
(model, data, etc.)

Train
model

Diagnostics
(bias, variance and
error analysis)

DeepLearning.AI

Stanford ONLINE



Machine learning development process

Adding data

Adding data

→ Add more data of everything. E.g., “Honeypot” project.

→ Add more data of the types where error analysis has indicated it might help.

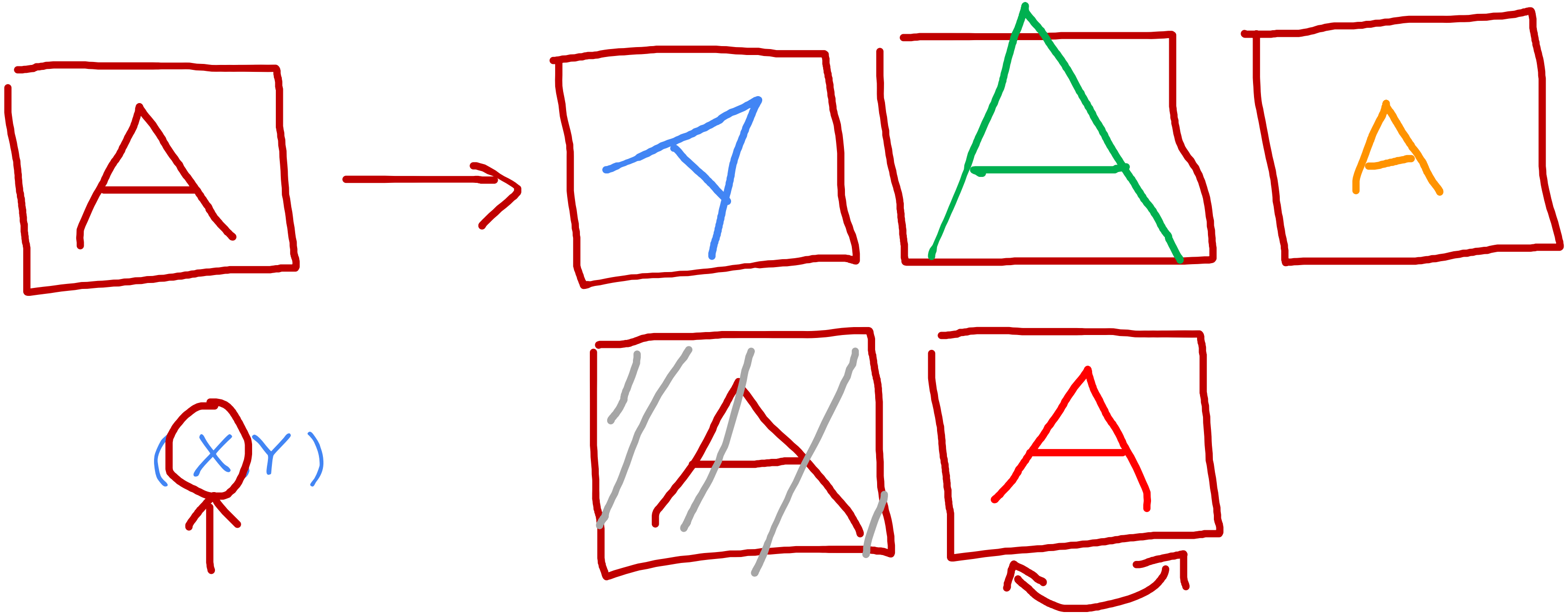
Pharma spam

E.g., Go to unlabeled data and find more examples of Pharma related spam.

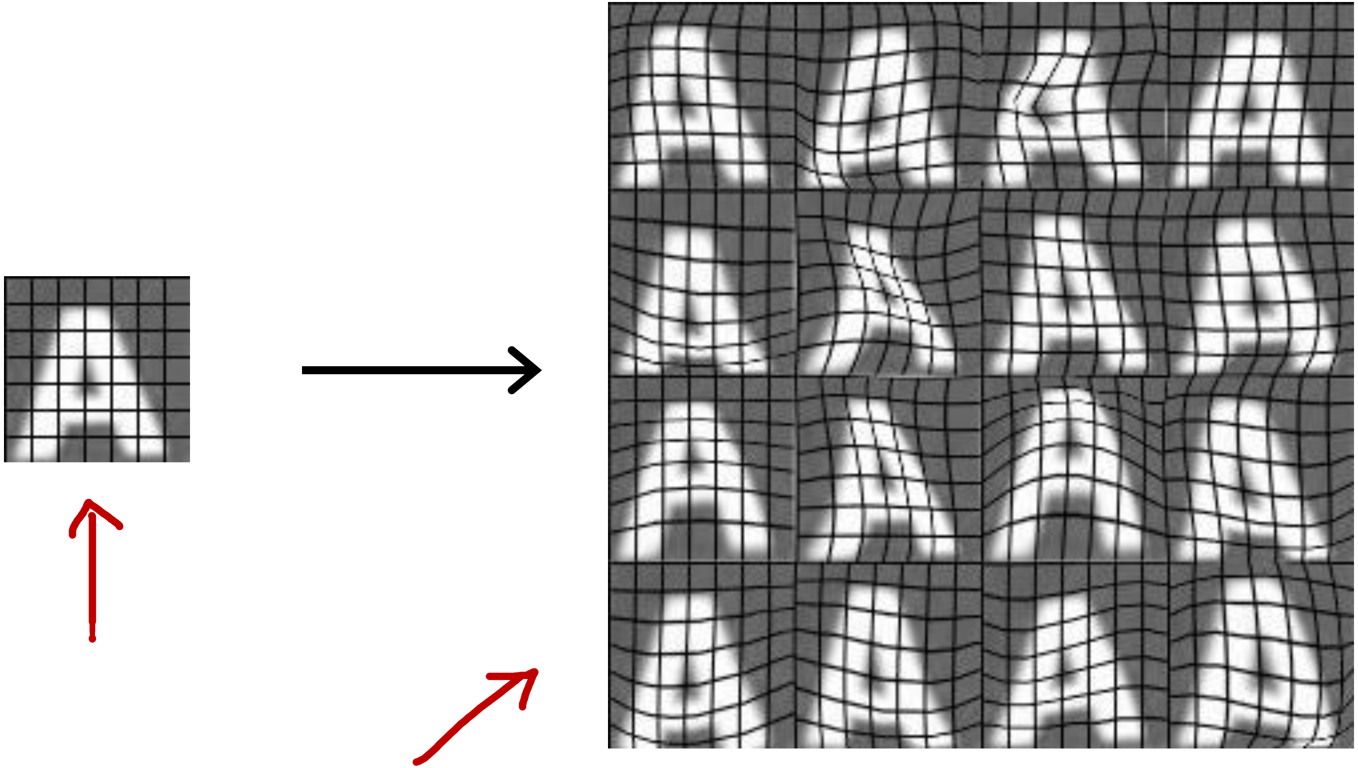
(X,Y)

Data augmentation

Augmentation: modifying an existing training example to create a new training example.



Data augmentation by introducing distortions



Data augmentation for speech

Speech recognition example

🔊 Original audio (voice search: ”What is today’s weather?”)

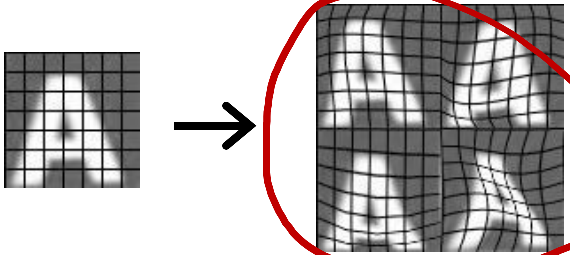
🔊 + Noisy background: Crowd

🔊 + Noisy background: Car

🔊 + Audio on bad cellphone connection

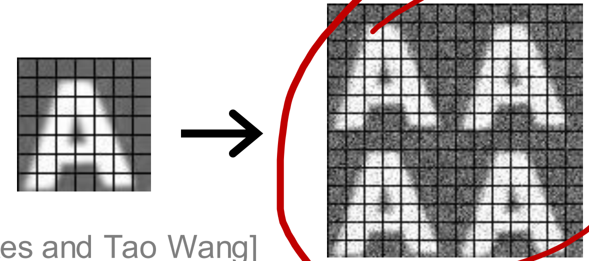
Data augmentation by introducing distortions

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

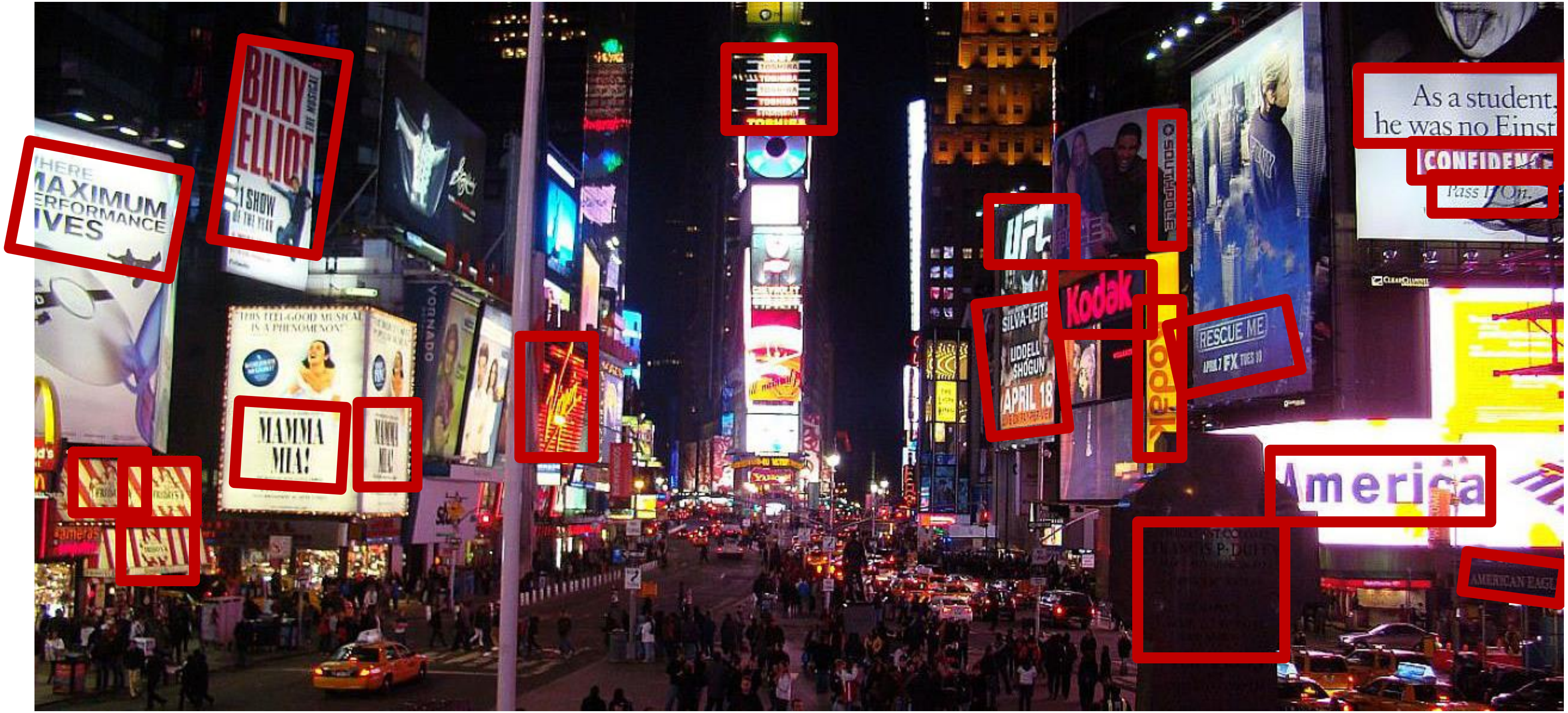
$x_i \leftarrow x_i +$ random noise

[Adam Coates and Tao Wang]

Data synthesis

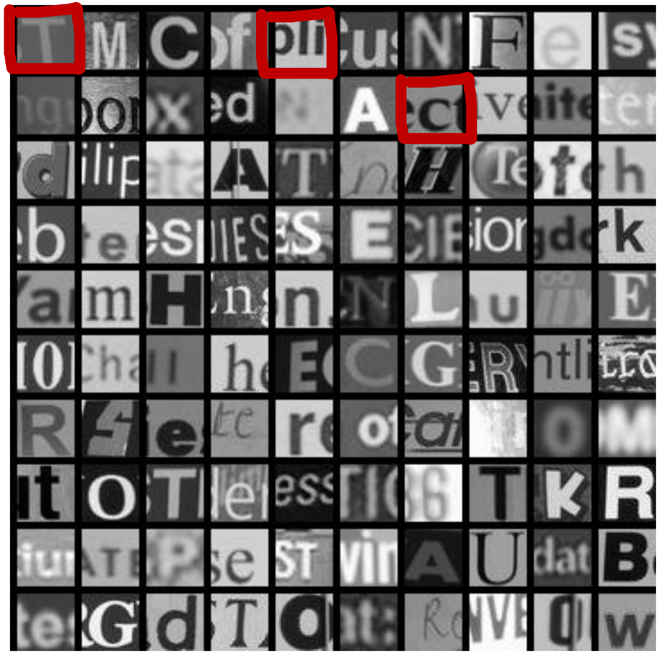
Synthesis: using artificial data inputs to create a new training example.

Artificial data synthesis for photo OCR



[http://www.publicdomainpictures.net/view-image.php?image=5745&picture=times-square]

Artificial data synthesis for photo OCR



Real data

[Adam Coates and Tao Wang]

Abcdefg

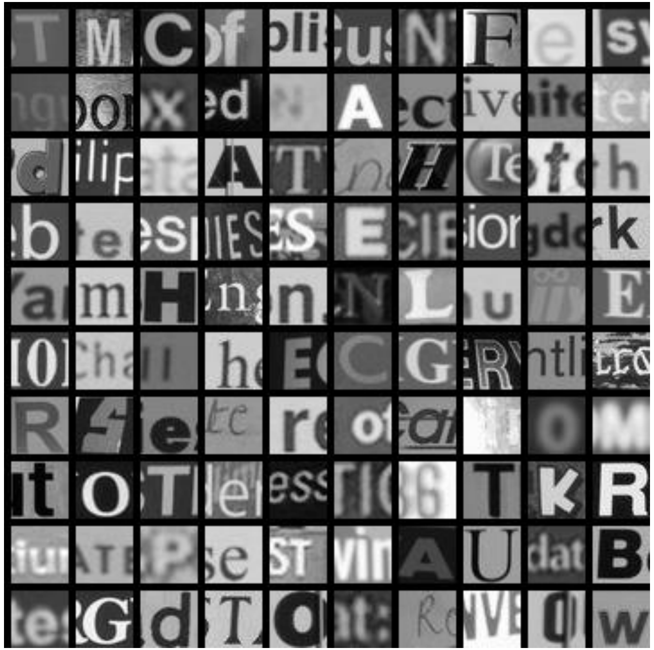
Abcdefg

Abcdefg

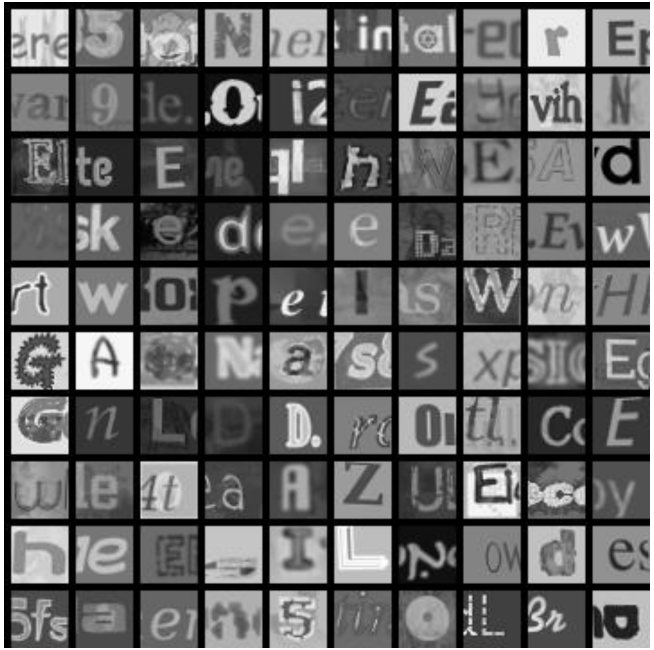
Abcdefg

Abcdefg

Artificial data synthesis for photo OCR



Real data



Synthetic data

[Adam Coates and Tao Wang]

Engineering the data used by your system

Conventional model-centric approach:

AI = Code + Data

(algorithm/model)

Work on this

Data-centric approach:

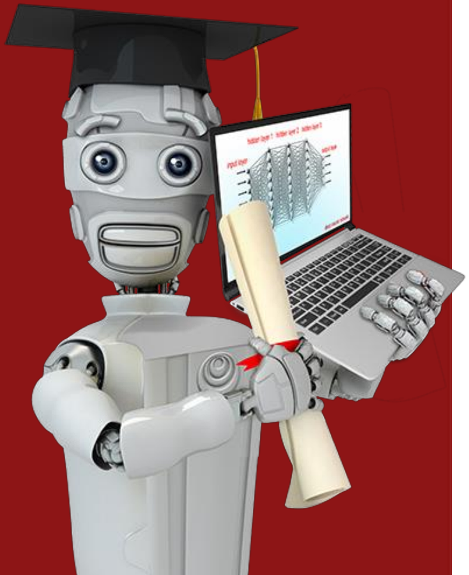
AI = Code + Data

(algorithm/model)

Work on this

DeepLearning.AI

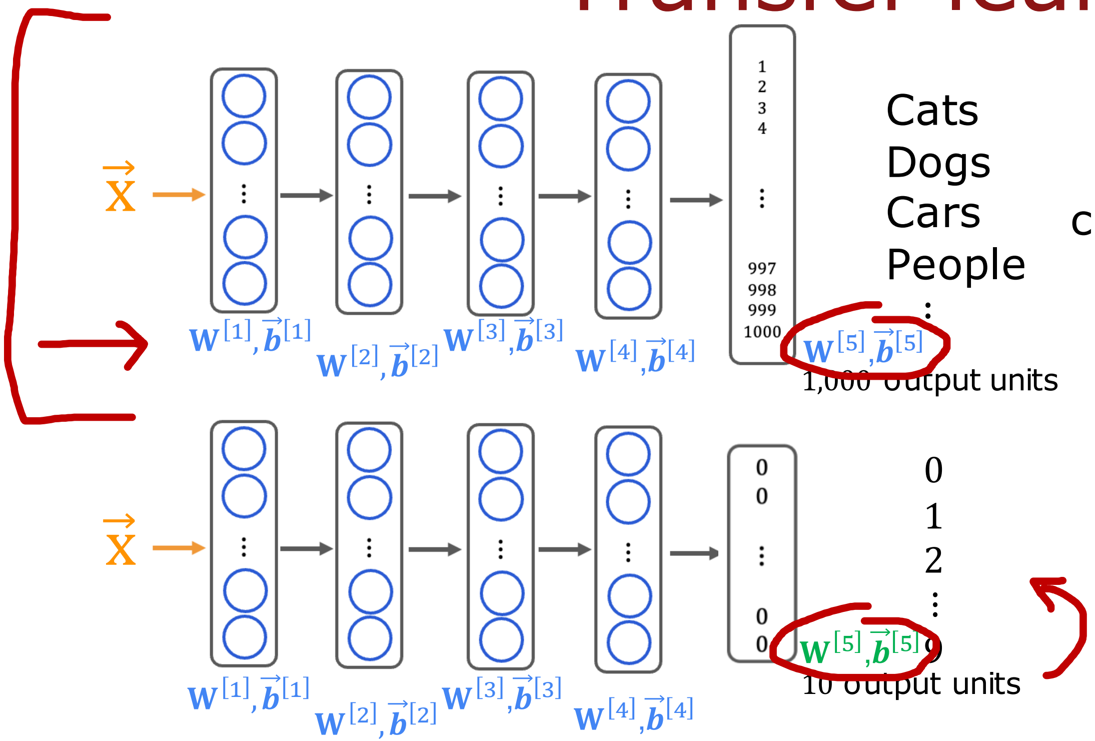
Stanford
ONLINE



Machine learning development process

Transfer learning: using data from a different task

Transfer learning



0, 1, 2, ... ,9

1 million images

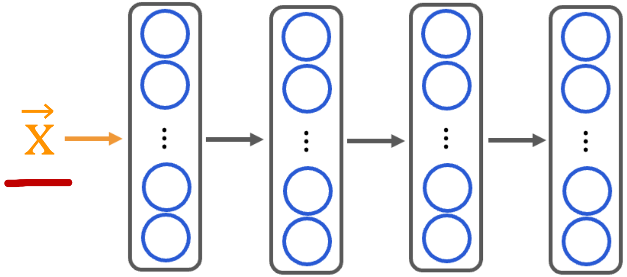
Supervised pretraining

Fine tuning

Option 1: only train output layers parameters.

Option 2: train all parameters.

Why does transfer learning work?

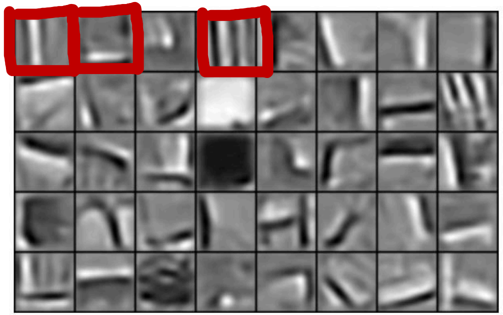


detects edges

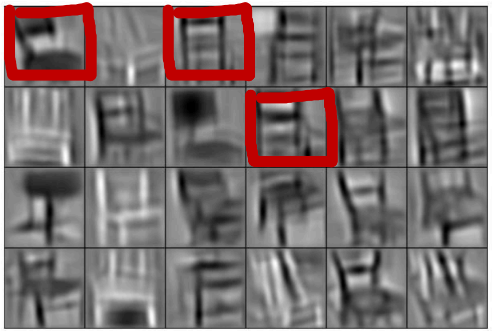
detects corners

detects curves/basic shapes

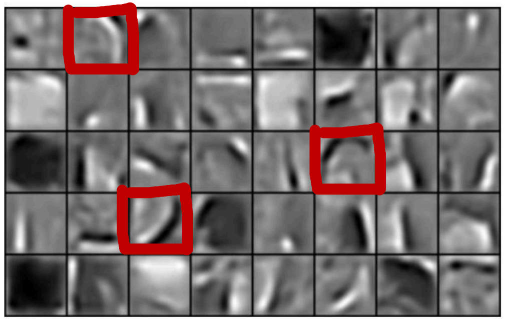
use the same input type



Edges



Corners



Curves / basic shapes

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).

1M

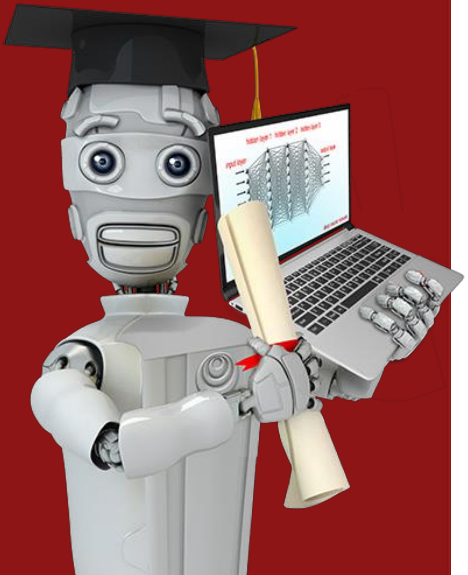
→ 2. Further train (fine tune) the network on your own data.

1000

50

DeepLearning.AI

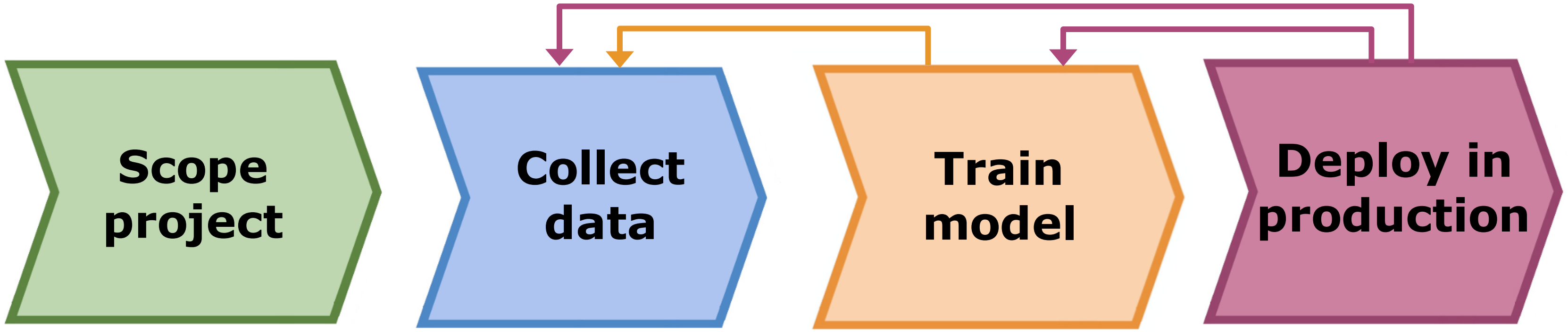
Stanford ONLINE



Machine learning development process

Full cycle of a machine learning project

Full cycle of a machine learning project



Scope project

Collect data

Train model

Deploy in production

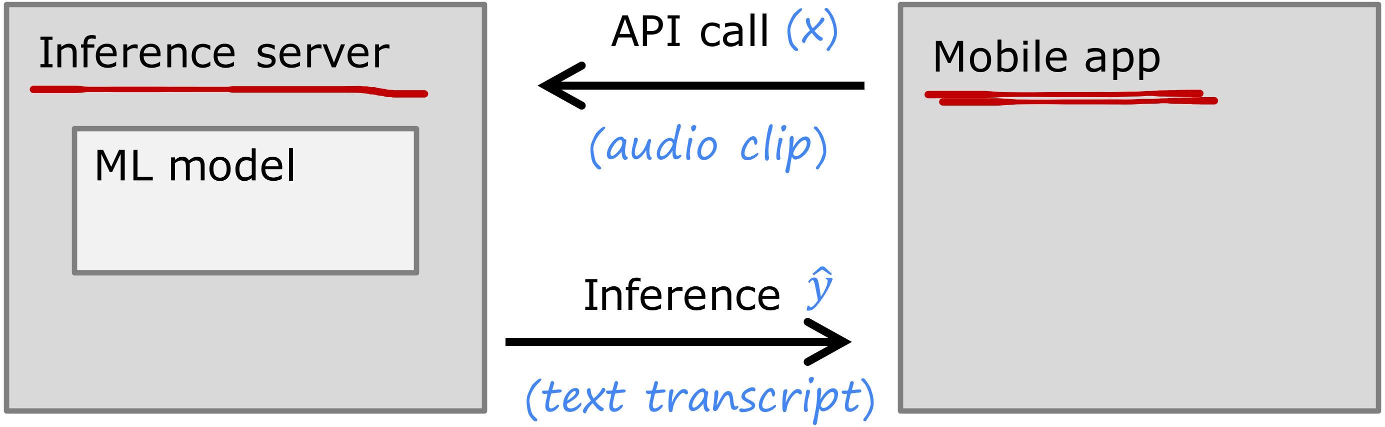
Define project

Define and collect data

Training, error analysis & iterative improvement

Deploy, monitor and maintain system

Deployment



Inference server

ML model

API call (x)

(audio clip)

Inference $\hat{y}$

(text transcript)

Mobile app

→ Software engineering may be needed for:

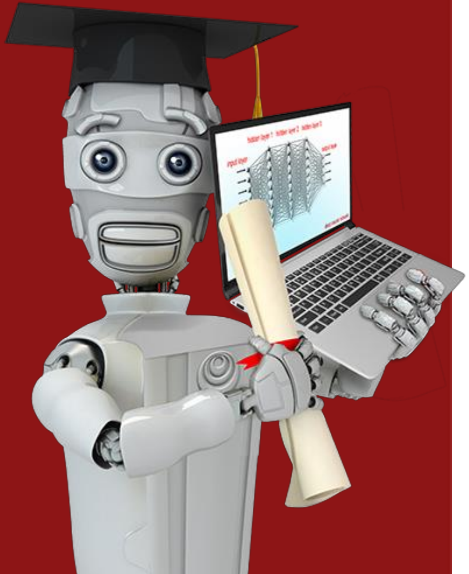
- Ensure reliable and efficient predictions
- Scaling
- Logging
- System monitoring
- Model updates

MLOps

machine learning operations

DeepLearning.AI

Stanford ONLINE



Machine learning development process

Fairness, bias, and ethics

Bias

Hiring tool that discriminates against women.

Facial recognition system matching dark skinned individuals to criminal mugshots.

Biased bank loan approvals.

Toxic effect of reinforcing negative stereotypes.

Adverse use cases

Deepfakes

Spreading toxic/incendiary speech through optimizing for engagement.

Generating fake content for commercial or political purposes.

Using ML to build harmful products, commit fraud etc.

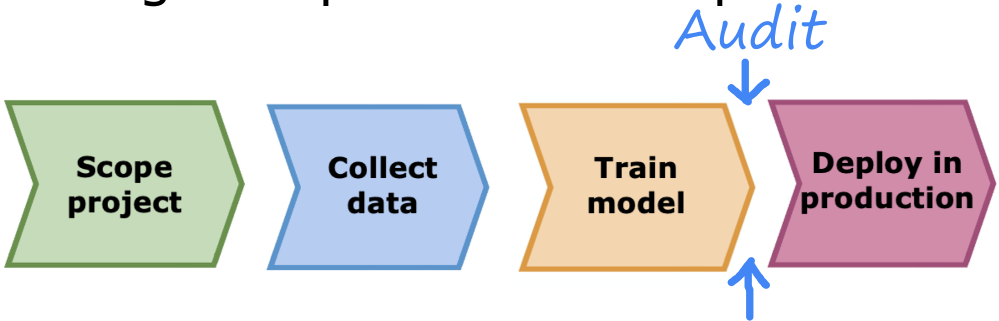
Spam vs anti-spam : fraud vs anti-fraud.

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.

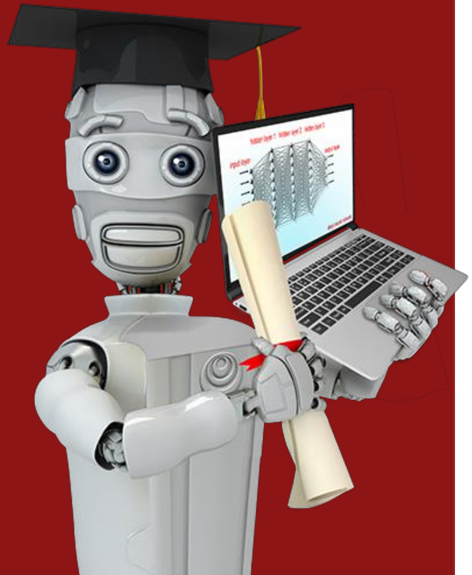
Audit systems against possible harm prior to deployment.



Develop mitigation plan (if applicable), and after deployment, monitor for possible harm.

DeepLearning.AI

Stanford
ONLINE



Skewed datasets (optional)

Error metrics for skewed datasets

Rare disease classification example

Train classifier $f_{\vec{w},b}(\vec{x})$ ($y = 1$ if disease present, $y = 0$ otherwise)

Find that you've got $\boxed{1\%}$ error on test set (99% correct diagnoses)

Only 0.5% of patients have the disease

```
print("y=0")
```

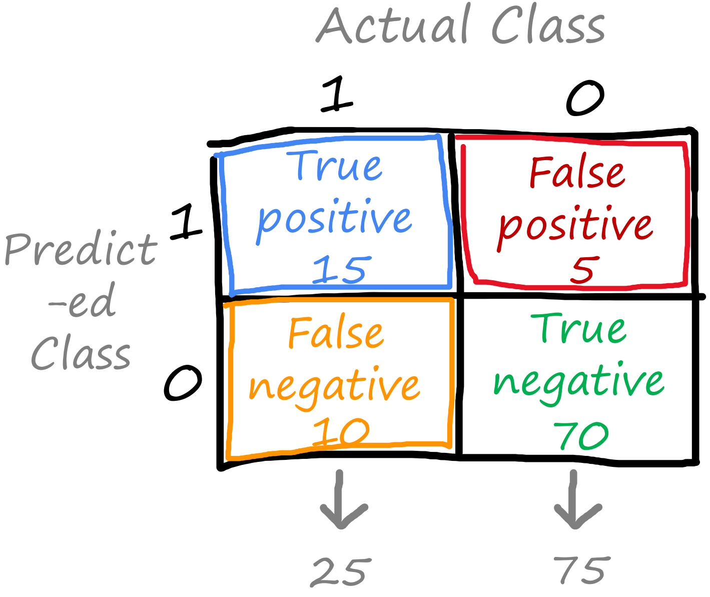
99.5% accuracy, $\boxed{0.5\% \text{ error}}$ ←

1% ←

1.2%

Precision/recall

$y = 1$ in presence of rare class we want to detect.



		Actual Class 1	Actual Class 0
Predicted Class	1	True positive 15	False positive 5
	0	False negative 10	True negative 70
		25	75

```
print("y=0")
```

Precision:
(of all patients where we predicted $y = 1$, what fraction actually have the rare disease?)

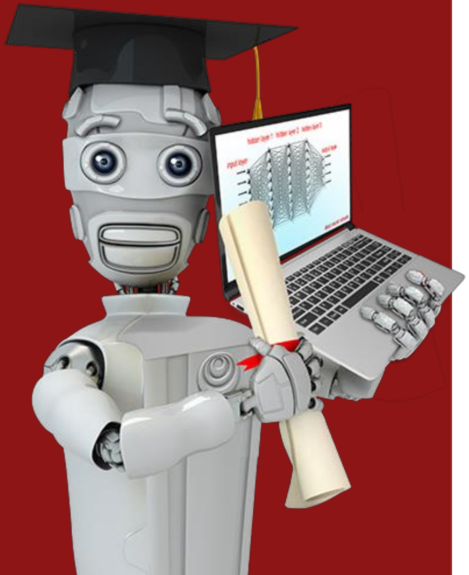
$$\frac{\text{True positives}}{\#\text{predicted positive}} = \frac{\text{True positives}}{\text{True pos} + \text{False pos}} = \frac{15}{15+5} = \underline{0.75}$$

Recall: ←
(of all patients that actually have the rare disease, what fraction did we correctly detect as having it?)

$$\frac{\text{True positives}}{\#\text{actual positive}} = \frac{\text{True positives}}{\text{True pos} + \text{False neg}} = \frac{15}{15+10} = \underline{0.6}$$

DeepLearning.AI

Stanford
ONLINE



Skewed datasets (optional)

Trading off precision and recall

Trading off precision and recall

Logistic regression: $0 < f_{\vec{w},b}(\vec{x}) < 1$

Predict 1 if $f_{\vec{w},b}(\vec{x}) \geq$ ~~0.5~~ ~~0.7~~ ~~0.9~~ 0.3

Predict 0 if $f_{\vec{w},b}(\vec{x}) <$ ~~0.5~~ ~~0.7~~ ~~0.9~~ 0.3

$$\text{precision} = \frac{\text{true positives}}{\text{total predicted positive}}$$

$$\text{recall} = \frac{\text{true positives}}{\text{total actual positive}}$$

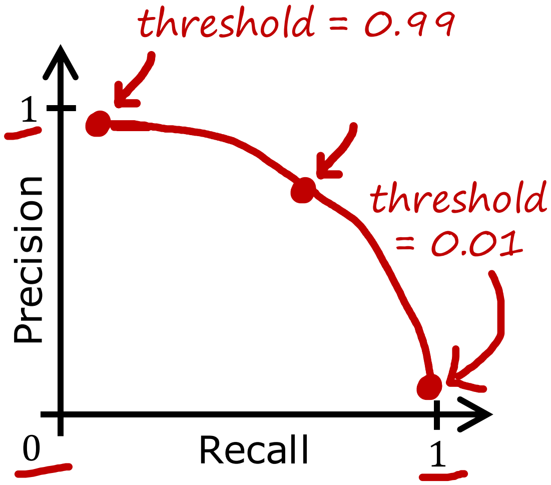
Suppose we want to predict $y = 1$ (rare disease) only if very confident.

→ higher precision, lower recall.

Suppose we want to avoid missing too many cases of rare disease (when in doubt predict $y = 1$)

→ lower precision, higher recall.

More generally predict 1 if: $f_{\vec{w},b}(\vec{x}) \geq$ threshold.



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")
```

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

$$\text{F1 score} = \frac{1}{\frac{1}{2}\left(\frac{1}{P}+\frac{1}{R}\right)} = 2\frac{PR}{P+R}$$