

Week 3

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.

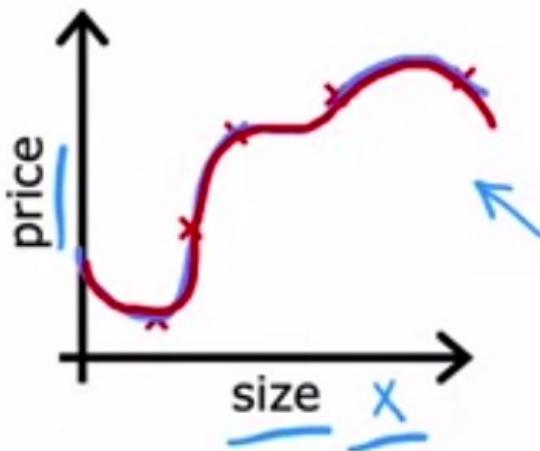
Diagnostics can take time to implement
but doing so can be a very good use of your time.



Evaluating and choosing models

Evaluating a model

Evaluating your model



$$f_{\vec{w}, b}(\vec{x}) = w_1 x_1 + w_2 x_2^2 + \dots + w_n x_n^n + b$$

x = x_1
f(\vec{x}) = $w_1 x_1 + w_2 x_2^2 + \dots + w_n x_n^n + b$
just x

$w_4 x^4$

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

- 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	
70%	2104	400
	1600	330
	2400	369
	1416	232
	3000	540
	1985	300
	1534	315
30%	1427	199
	1380	212
	1494	243

training set → $(x^{(1)}, y^{(1)})$
 $(x^{(2)}, y^{(2)})$
⋮
 $(x^{(m_{train})}, y^{(m_{train})})$

m_{train} = no. training examples
= 7

test set → $(x_{test}^{(1)}, y_{test}^{(1)})$
⋮
 $(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

m_{test} = no. test examples
= 3

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

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

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

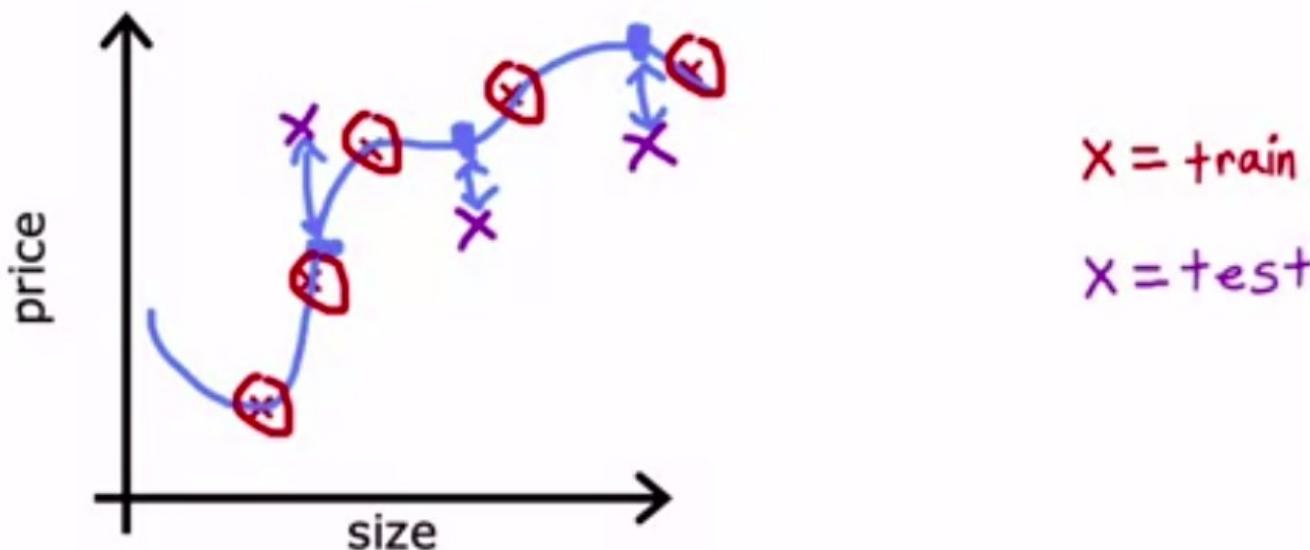
Compute test error:

$$J_{test}(\vec{w}, b) = \frac{1}{2m_{test}} \left[\sum_{i=1}^{m_{test}} \underbrace{\left(f_{\vec{w}, b}(\vec{x}_{test}^{(i)}) - y_{test}^{(i)} \right)^2}_{\text{minimized term}} \right] \quad \cancel{\sum_{j=1}^n w_j^2}$$

Compute training error:

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

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



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

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

Train/test procedure for classification problem

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

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

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.



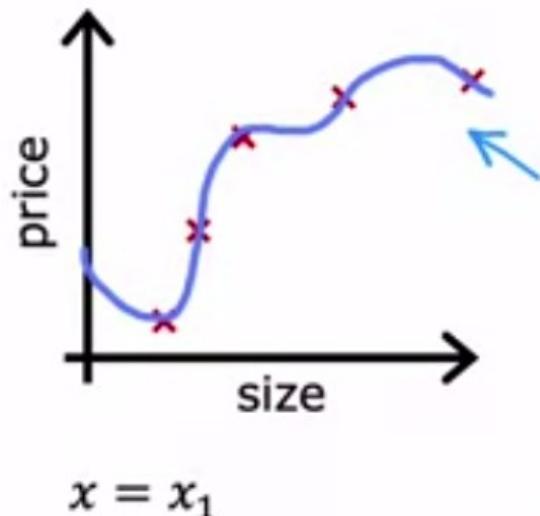
Evaluating and choosing models

Model selection and training/cross validation/test sets

Model selection (choosing a model)

1. $f_{\vec{w}, b}(\vec{x}) = w_1 x + b$
2. $f_{\vec{w}, b}(\vec{x}) = w_1 x + w_2 x^2 + b$
3. $f_{\vec{w}, b}(\vec{x}) = w_1 x + w_2 x^2 + w_3 x^3 + b$
⋮
10. $f_{\vec{w}, b}(\vec{x}) = w_1 x + w_2 x^2 + \dots + w_{10} x^{10} + b$

Model selection (choosing a model)



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

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

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

Model selection (choosing a model)

$d=1$

$$1. f_{\vec{w}, b}(\vec{x}) = w_1 x + b$$

$\rightarrow w^{<1>} , b^{<1>} \rightarrow J_{test}(w^{<1>} , b^{<1>})$

$d=2$

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

$\rightarrow w^{<2>} , b^{<2>} \rightarrow J_{test}(w^{<2>} , b^{<2>})$

$d=3$

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

\vdots

$d=10$

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

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

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

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

w, b are overly optimistic estimate of generalization error on training data.

Training/cross validation/test set

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

Diagram illustrating the splitting of a dataset into training, cross-validation, and test sets:

- training set:** 60% of the data. Contains 6 samples ($m_{train} = 6$). Represented by the first 6 rows.
- cross validation:** 20% of the data. Contains 2 samples ($m_{cv} = 2$). Represented by the next 2 rows.
- test set:** 20% of the data. Contains 2 samples ($m_{test} = 2$). Represented by the last 2 rows.

Annotations:
- Braces group the first 6 rows for training, the next 2 for cross-validation, and the last 2 for testing.
- Blue arrows point from the row numbers to their respective set names.
- Handwritten text indicates percentages: "60%", "20%", and "20%" below the corresponding brace groups.
- Handwritten labels "training set", "cross validation", and "test set" are placed near their respective groups.
- Handwritten labels " $(x^{(1)}, y^{(1)})$ ", " \vdots ", and " $(x^{(m_{train})}, y^{(m_{train})})$ " are placed next to the first 6 rows.
- Handwritten labels " $(x_{cv}^{(1)}, y_{cv}^{(1)})$ ", " \vdots ", and " $(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$ " are placed next to the next 2 rows.
- Handwritten labels " $(x_{test}^{(1)}, y_{test}^{(1)})$ ", " \vdots ", and " $(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$ " are placed next to the last 2 rows.

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%

$$(x^{(1)}, y^{(1)})$$

⋮

$$(x^{(m_{train})}, y^{(m_{train})})$$

$$m_{train} = 6$$

cross validation

20%

$$(x_{cv}^{(1)}, y_{cv}^{(1)})$$

⋮

$$(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$$

$$m_{cv} = 2$$

test set

20%

$$(x_{test}^{(1)}, y_{test}^{(1)})$$

⋮

$$(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$$

$$m_{test} = 2$$

Training/cross validation/test set

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

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

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

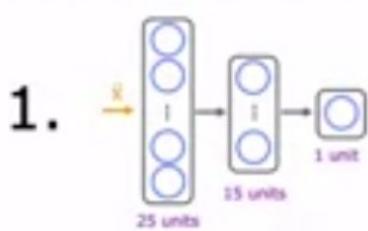
Model selection

- $d=1$ 1. $f_{\vec{w}, b}(\vec{x}) = w_1 x + b \quad w^{(1)}, b^{(1)} \rightarrow J_{cv}(w^{(1)}, b^{(1)})$
- $d=2$ 2. $f_{\vec{w}, b}(\vec{x}) = w_1 x + w_2 x^2 + b \rightarrow J_{cv}(w^{(2)}, b^{(2)})$
- $d=3$ 3. $f_{\vec{w}, b}(\vec{x}) = w_1 x + w_2 x^2 + w_3 x^3 + b \quad \vdots$
- $\vdots \quad \vdots$
- $d=10$ 10. $f_{\vec{w}, b}(\vec{x}) = w_1 x + w_2 x^2 + \dots + w_{10} x^{10} + b \quad J_{cv}(w^{(10)}, b^{(10)})$

→ Pick $w_1 x + \dots + w_4 x^4 + b \quad (J_{cv}(w^{(4)}, b^{(4)}))$

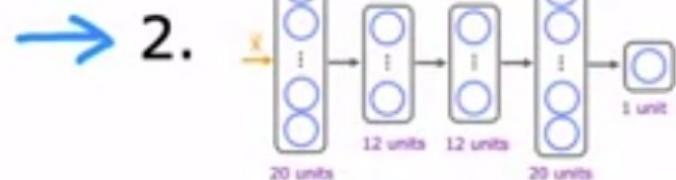
Estimate generalization error using test the set: $J_{test}(w^{(4)}, b^{(4)})$

Model selection – choosing a neural network architecture



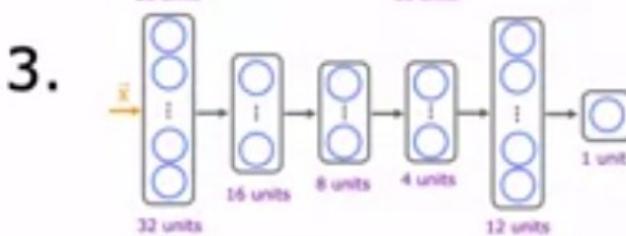
$$w^{(1)}, b^{(1)}$$

$$J_{cv}(w^{(1)}, b^{(1)})$$



$$w^{(2)}, b^{(2)}$$

$$J_{cv}(w^{(2)}, b^{(2)})$$



$$w^{(3)}, b^{(3)}$$

$$J_{cv}(w^{(3)}, b^{(3)})$$

Pick $w^{(2)}, b^{(2)}$

+train, CV

Estimate generalization error using the test set: $J_{test}(w^{(2)}, b^{(2)})$

[Back](#)

Practice quiz: Advice for applying machine learning

Graded Quiz • 30 min

Due Apr 2, 11:59 PM IST

1.

1 / 1 point

In the context of machine learning, what is a diagnostic?

- This refers to the process of measuring how well a learning algorithm does on a test set (data that the algorithm was not trained on).
- A process by which we quickly try as many different ways to improve an algorithm as possible, so as to see what works.
- A test that you run to gain insight into what is/isn't working with a learning algorithm.
- An application of machine learning to medical applications, with the goal of diagnosing patients' conditions.

 **Correct**

Yes! A diagnostic is 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.

2.

1 / 1 point

True/False? It is always true that the better an algorithm does on the training set, the better it will do on generalizing to new data.

- True
- False

 **Correct**

[Back](#)

Practice quiz: Advice for applying machine learning

Graded Quiz • 30 min

Due Apr 2, 11:59 PM IST

1.

1 / 1 point

In the context of machine learning, what is a diagnostic?

- This refers to the process of measuring how well a learning algorithm does on a test set (data that the algorithm was not trained on).
- A process by which we quickly try as many different ways to improve an algorithm as possible, so as to see what works.
- A test that you run to gain insight into what is/isn't working with a learning algorithm.
- An application of machine learning to medical applications, with the goal of diagnosing patients' conditions.



Correct

Yes! A diagnostic is 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.

2.

1 / 1 point

True/False? It is always true that the better an algorithm does on the training set, the better it will do on generalizing to new data.

- True
- False



Actually, if a model overfits the training set, it may not generalize well to new data.

[Back](#)

Practice quiz: Advice for applying machine learning

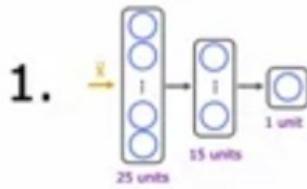
Due Apr 2, 11:59 PM IST

Graded Quiz • 30 min

3.

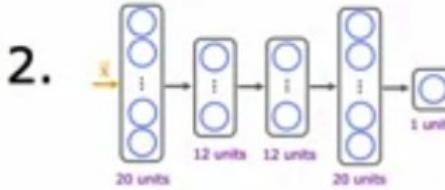
Model selection – choosing a neural network architecture

1 / 1 point



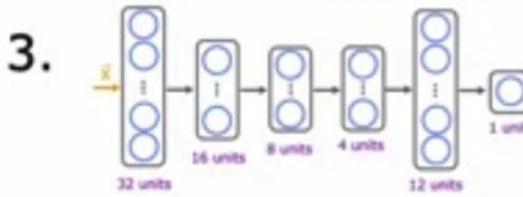
$w^{(1)}, b^{(1)}$

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



$w^{(2)}, b^{(2)}$

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



$w^{(3)}, b^{(3)}$

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

Train, CV

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

[Back](#)

Practice quiz: Advice for applying machine learning

Graded Quiz • 30 min

Due Apr 2, 11:59 PM IST

**Pick $\mathbf{W}^{(2)}, \mathbf{B}^{(2)}$** *Train, CV***Estimate generalization error using the test set: $J_{test}(\mathbf{W}^{(2)}, \mathbf{B}^{(2)})$**

For a classification task; suppose you train three different models using three different neural network architectures. Which data do you use to evaluate the three models in order to choose the best one?

- All the data -- training, cross validation and test sets put together.
- The training set
- The cross validation set
- The test set

Correct

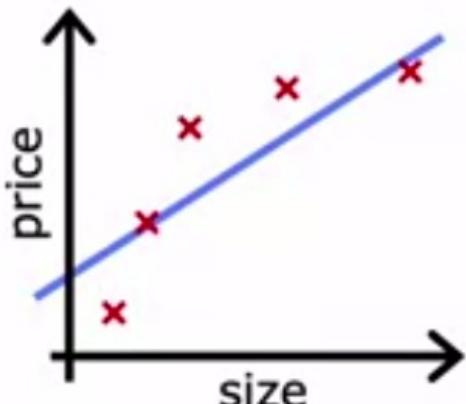
Correct. Use the cross validation set to calculate the cross validation error on all three models in order to compare which of the three models is best.



Bias and variance

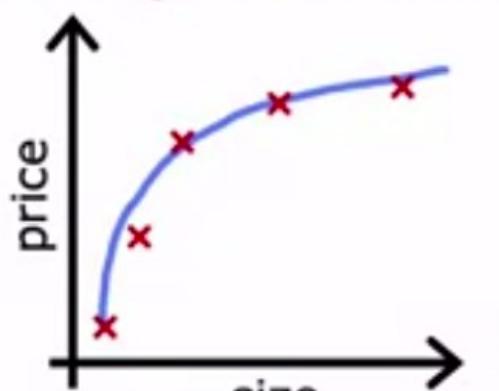
Diagnosing bias and variance

Bias/variance



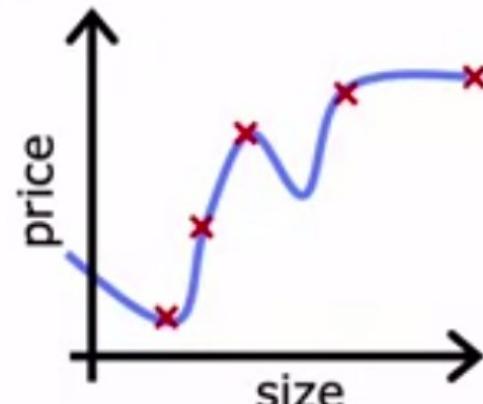
$$f_{\vec{w},b}(x) = w_1x + b$$

High bias
(underfit)



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

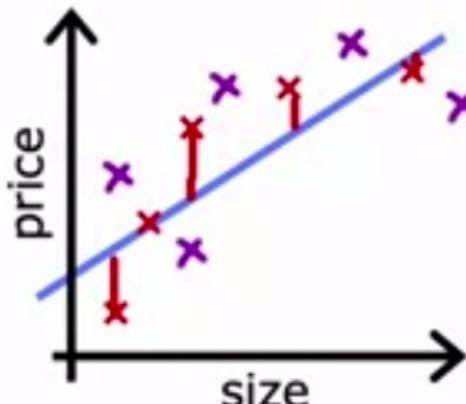
"Just right"



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

High variance
(overfit)

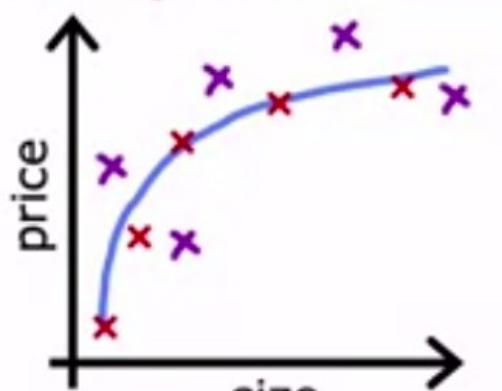
Bias/variance



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

→ High bias
(underfit)

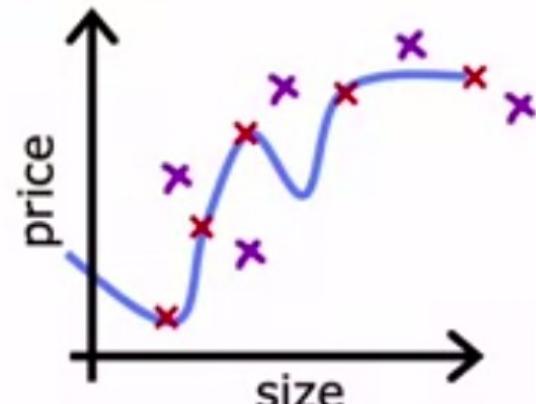
J_{train} is high
J_{CV} is high



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

"Just right"

J_{train} is low
J_{CV} is low

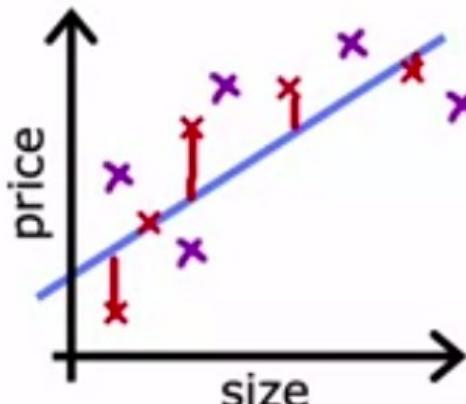


$$f_{\bar{w},b}(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$

High variance
(overfit)

J_{train} is low
J_{CV} is high

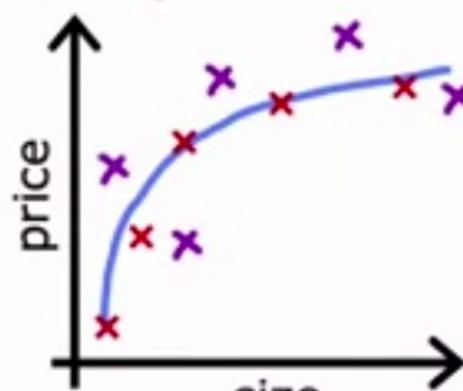
Bias/variance



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

→ High bias
(underfit)

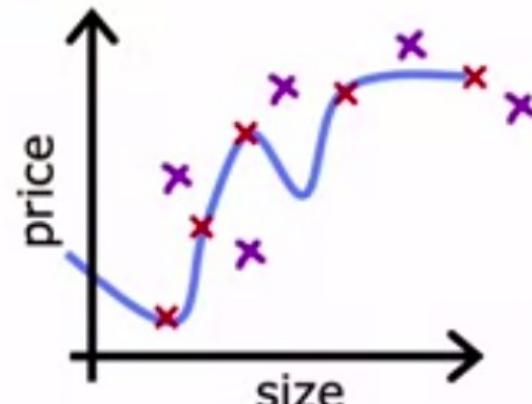
$d=1$ J_{train} is high
J_{CV} is high



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

"Just right"

$d=2$ J_{train} is low
J_{CV} is low

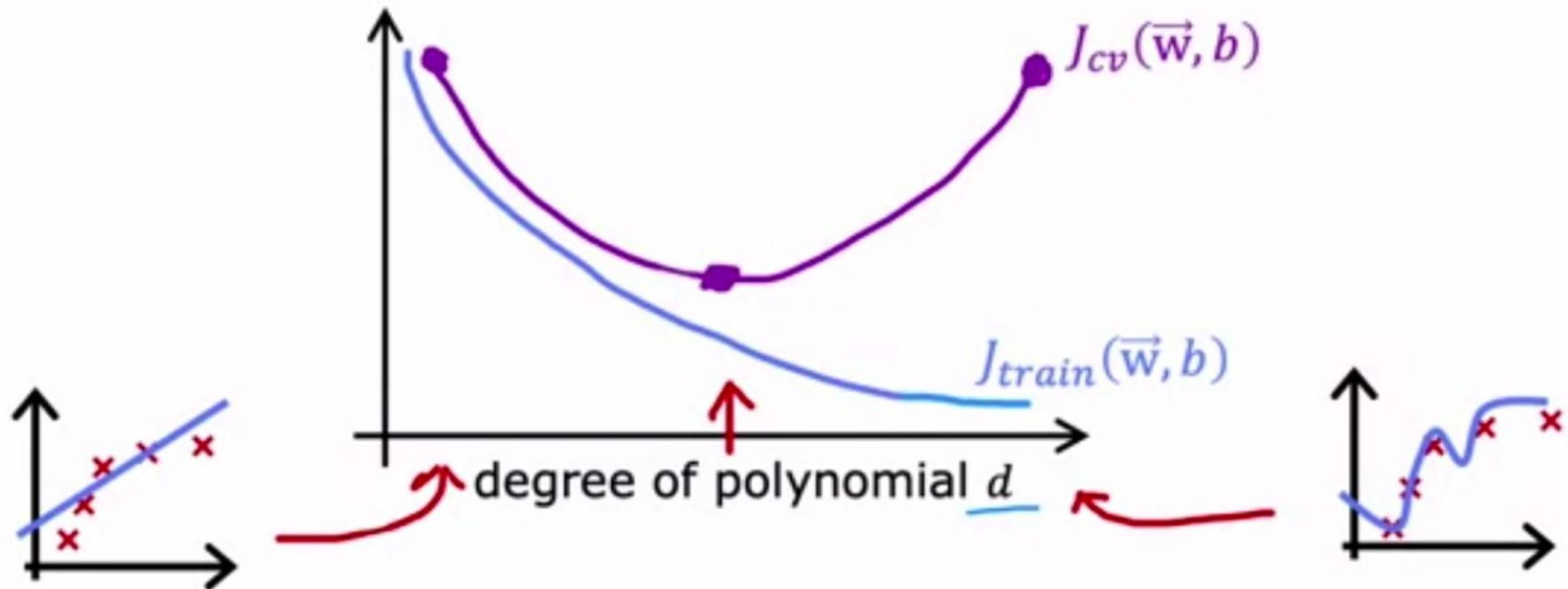


$$f_{\bar{w},b}(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$

High variance
(overfit)

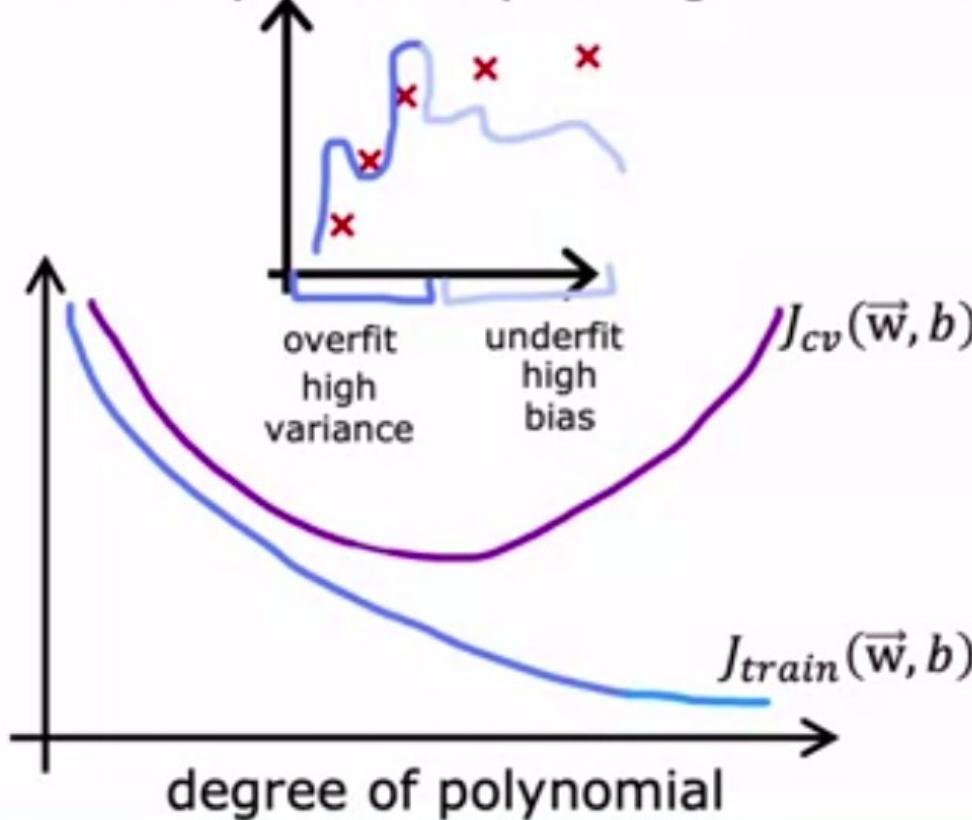
$d=4$ J_{train} is low
J_{CV} is high

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)

$J_{cv} \gg J_{train}$
(J_{train} may be low)

High bias and high variance

J_{train} will be high
and $J_{cv} \gg J_{train}$



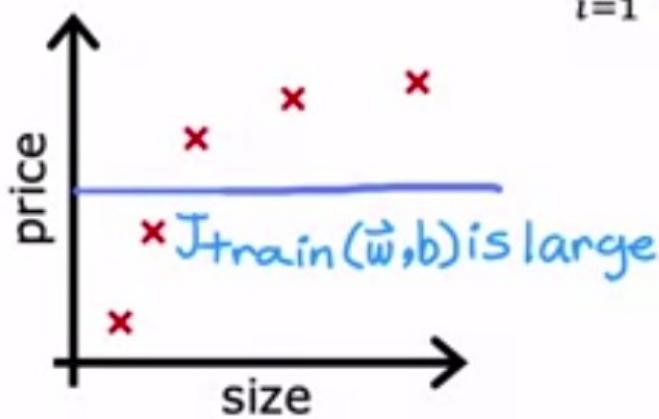
Bias and variance

Regularization and
bias/variance

Linear regression with regularization

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

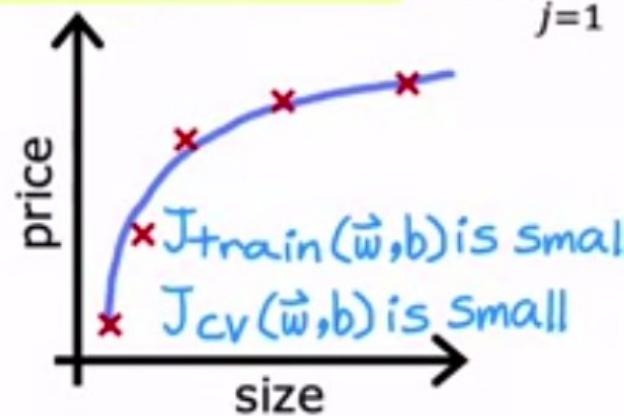
$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$



Large λ

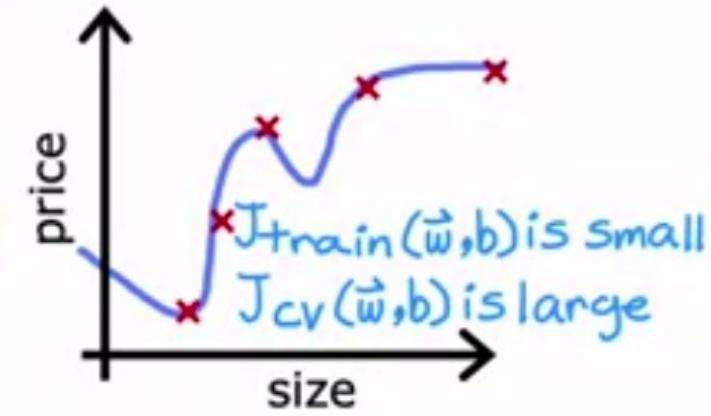
High bias (underfit)

$$\lambda = 10,000 \quad w_1 \approx 0, w_2 \approx 0 \\ f_{\vec{w}, b}(\vec{x}) \approx b$$



Intermediate λ

$$\lambda$$



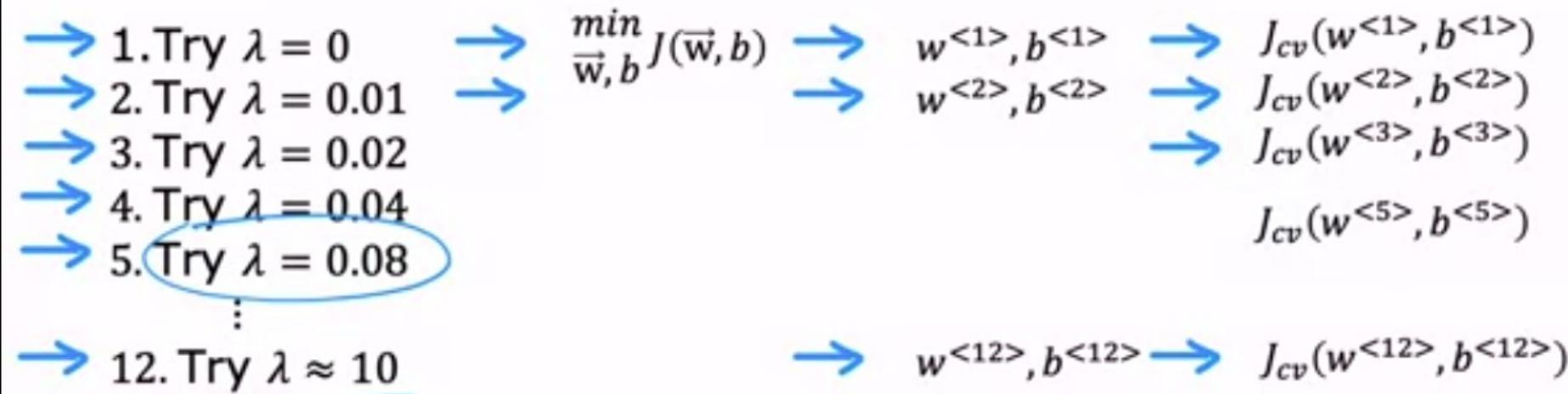
Small λ

High variance (overfit)

$$\lambda = 0$$

Choosing the regularization parameter λ

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

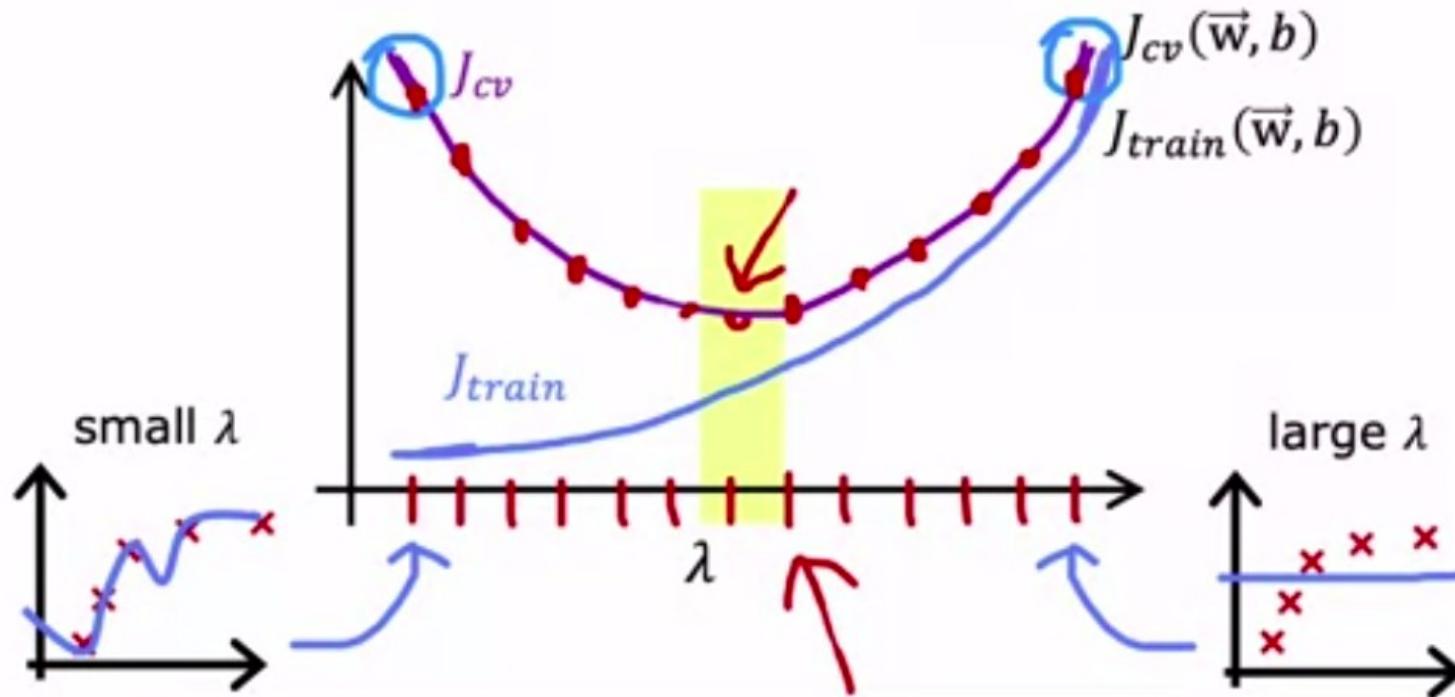


Pick $w^{<5>}, b^{<5>}$

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

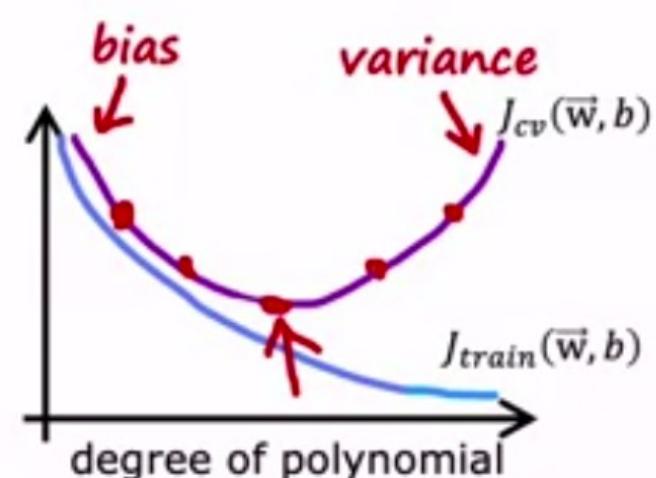
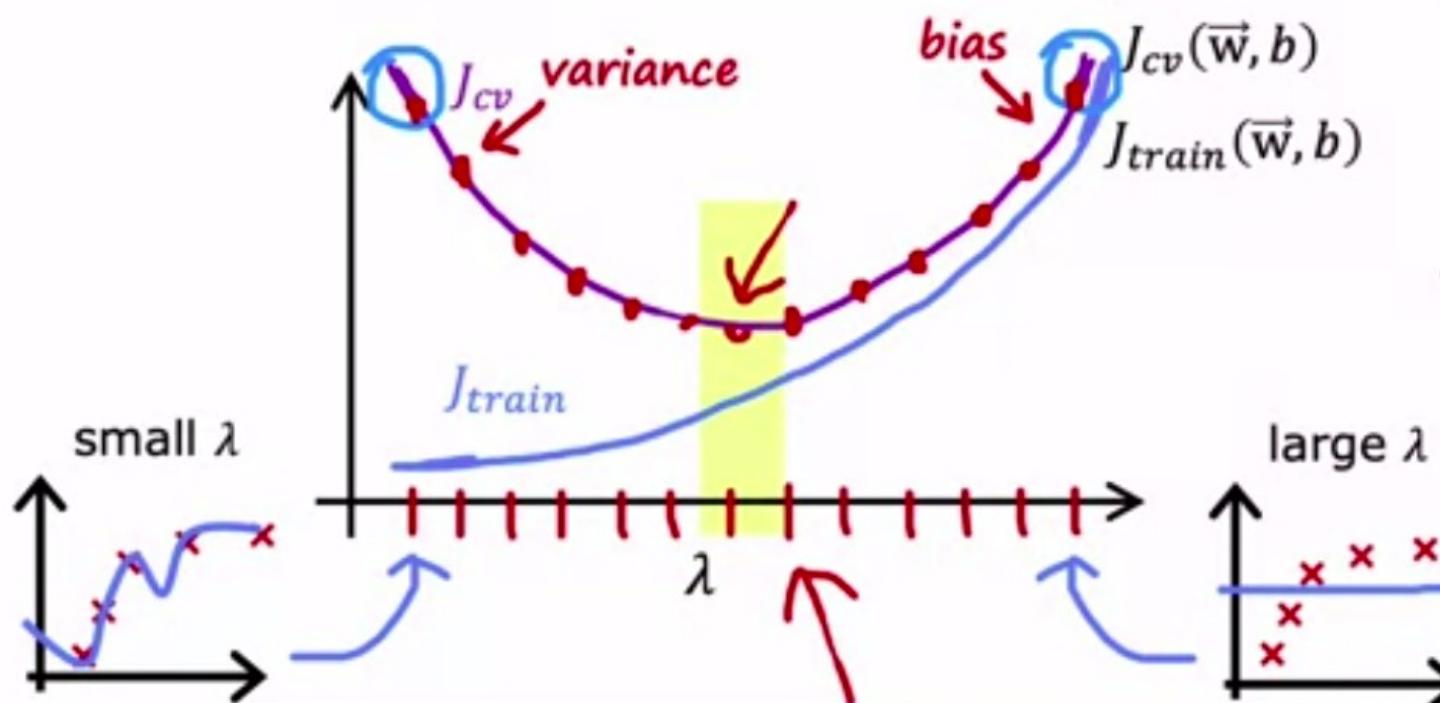
Bias and variance as a function of regularization parameter λ

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



Bias and variance as a function of regularization parameter λ

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



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}

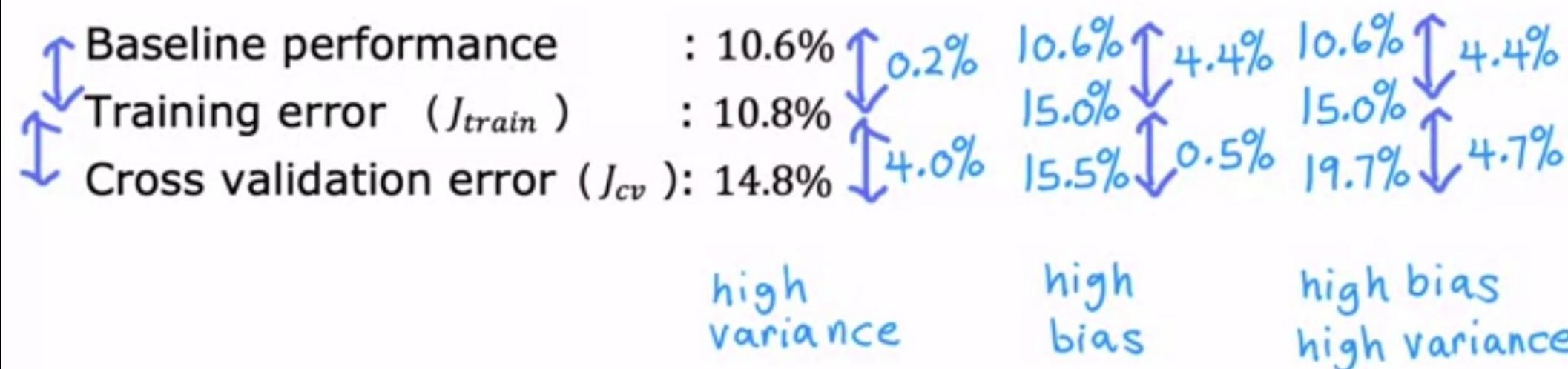
: 14.8%

0.2%

4.0%



Bias/variance examples





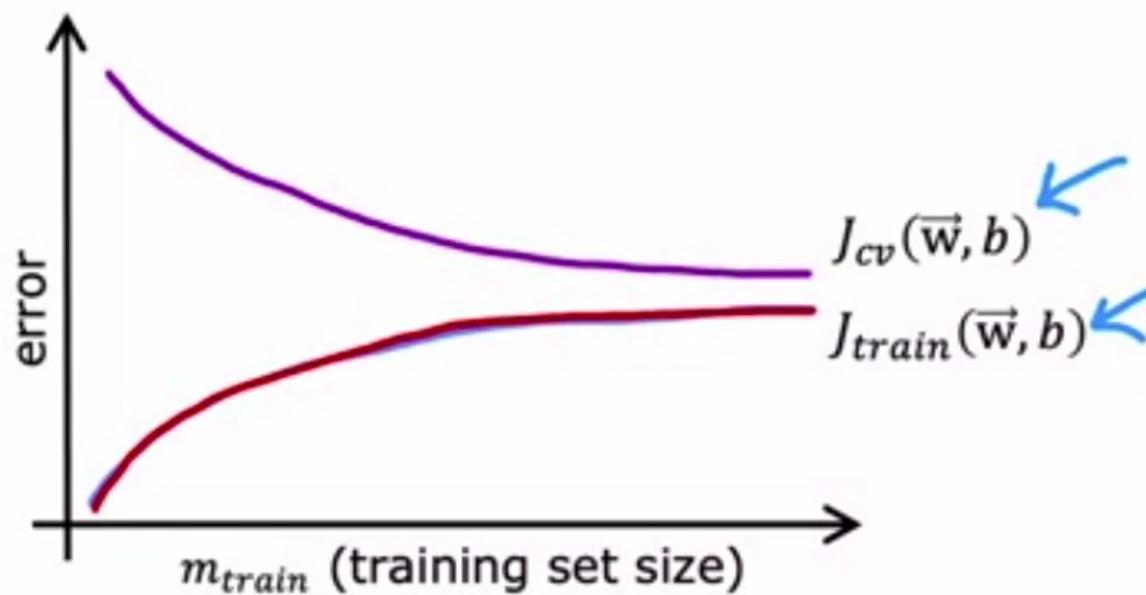
Bias and variance

Learning curves

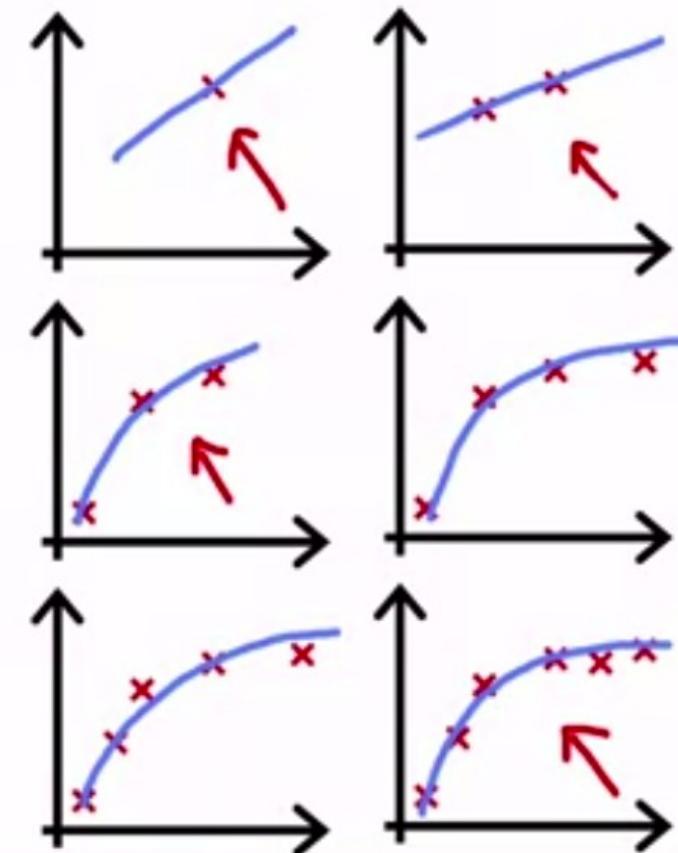
Learning curves

J_{train} = training error

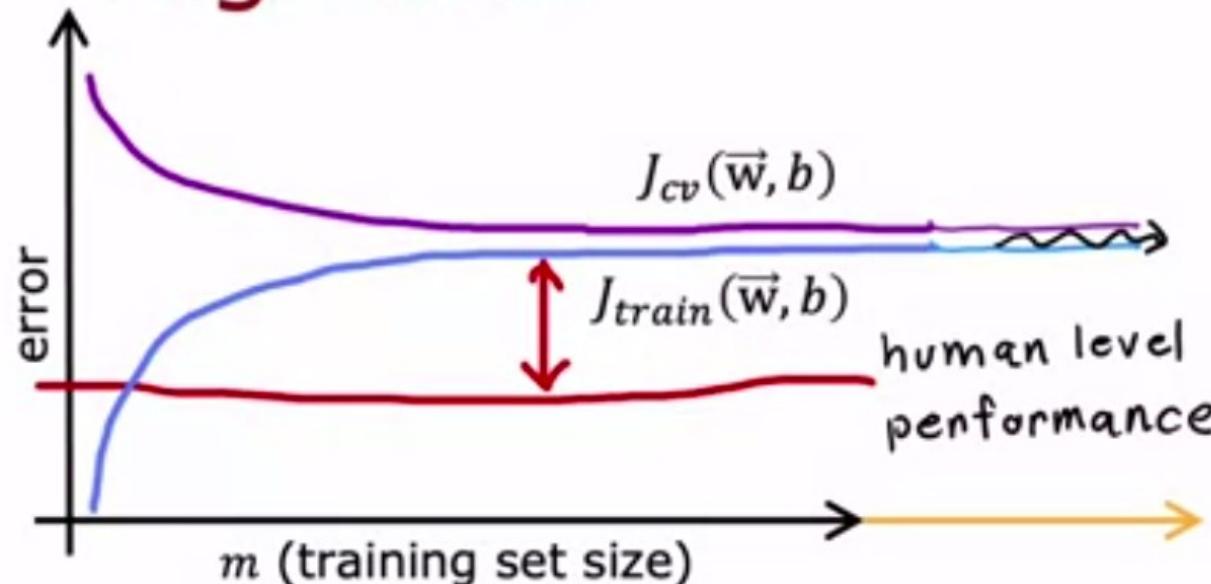
J_{cv} = cross validation error



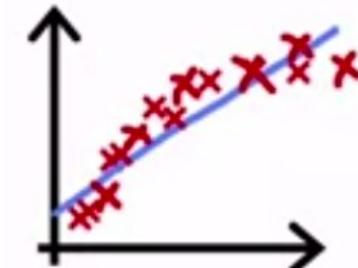
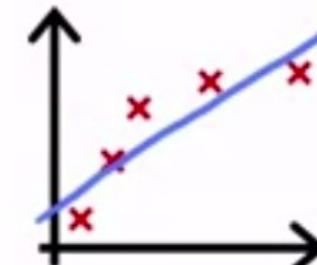
$$f_{\vec{w},b}(x) = w_1x + w_2x^2 + b$$



High bias



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

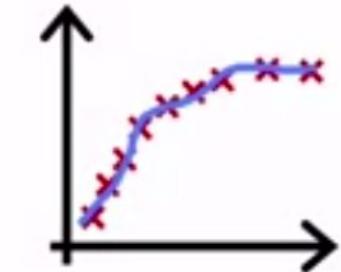
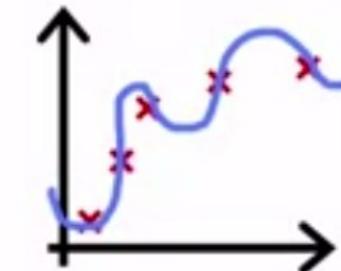
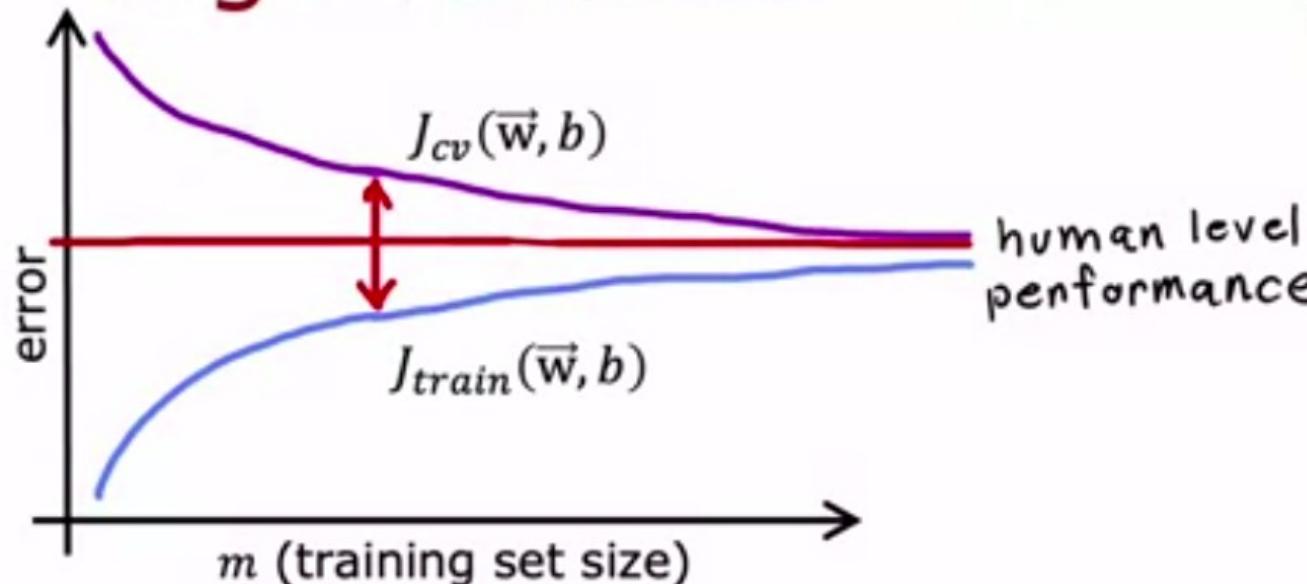


if a learning algorithm suffers from high bias,
getting more training data will not (by itself)
help much.

High variance

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

(with small λ)



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



Bias and variance

Deciding what to try next
revisted

Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

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

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 $x, x^2, \cancel{x}, \cancel{x^3}, \cancel{y}$... fixes high variance
- Try getting additional features fixes high bias
- Try adding polynomial features $(x_1^2, x_2^2, x_1 x_2, \text{etc})$ fixes high bias
- Try decreasing λ fixes high bias
- Try increasing λ fixes high variance



Bias and variance

Bias/variance and
neural networks

The bias variance tradeoff

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

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

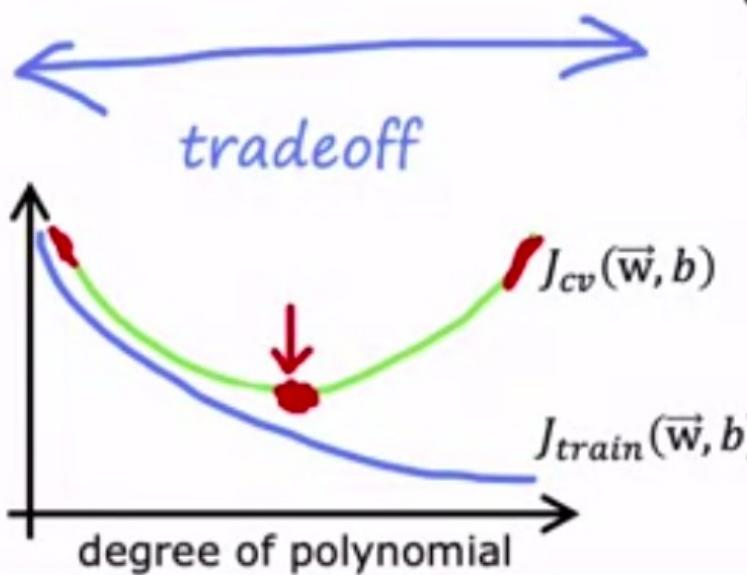
$$f_{\vec{w}, b}(x) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$$

Simple model

High bias

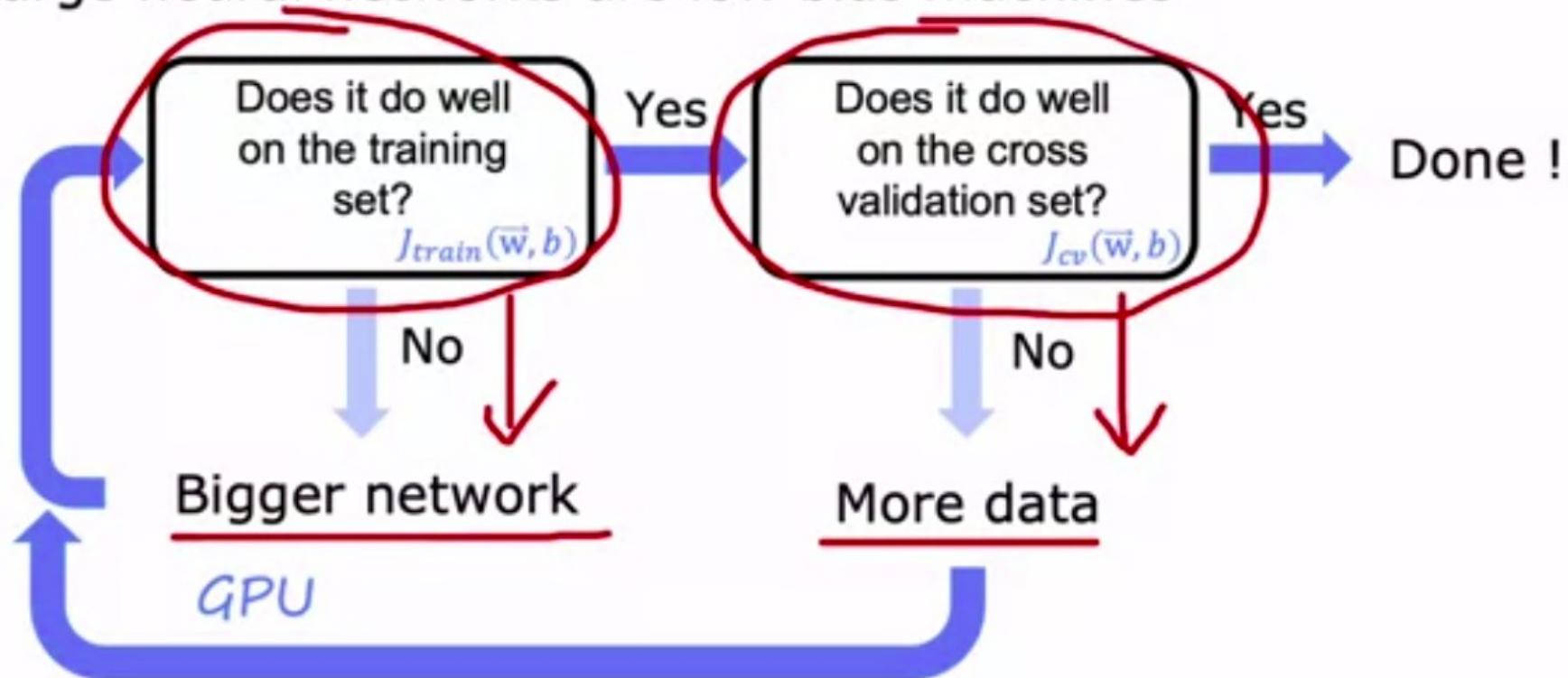
Complex model

High variance

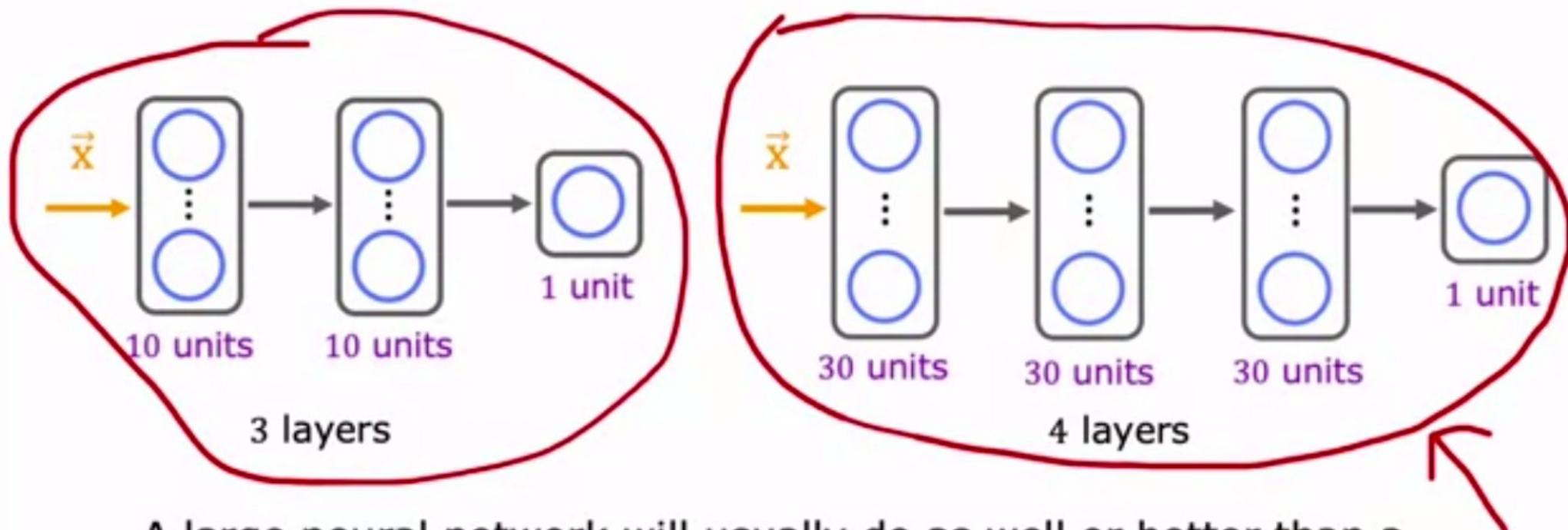


Neural networks and bias variance

Large neural networks are low bias machines



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

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

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

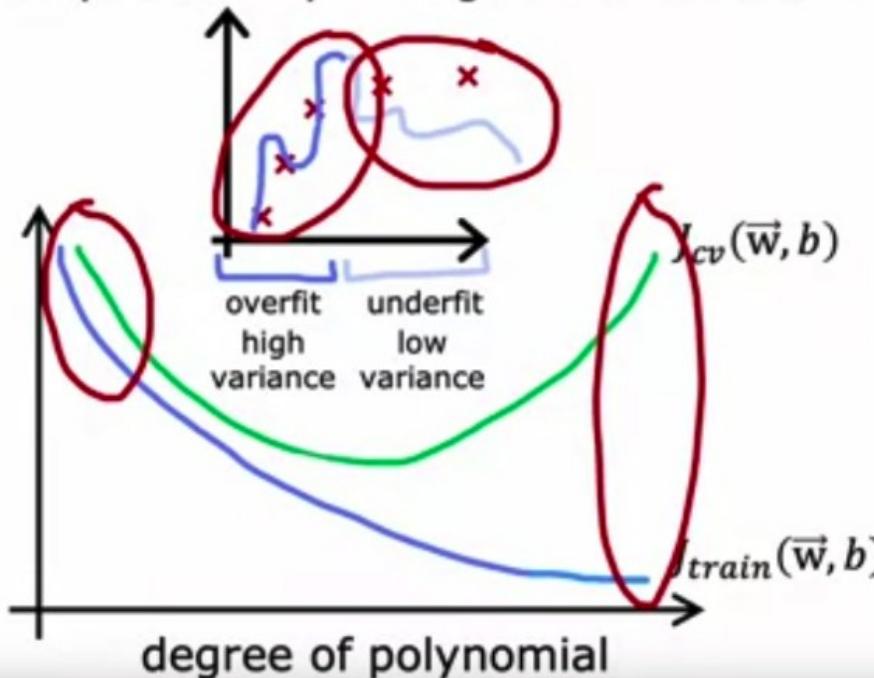
```
layer_1 = Dense(units=25, activation="relu", kernel_regularizer=L2(0.01))
layer_2 = Dense(units=15, activation="relu", kernel_regularizer=L2(0.01))
layer_3 = Dense(units=1, activation="sigmoid", kernel_regularizer=L2(0.01))
model = Sequential([layer_1, layer_2, layer_3])
```

1.

1 / 1 point

Diagnosing bias and variance

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



High bias (underfit)

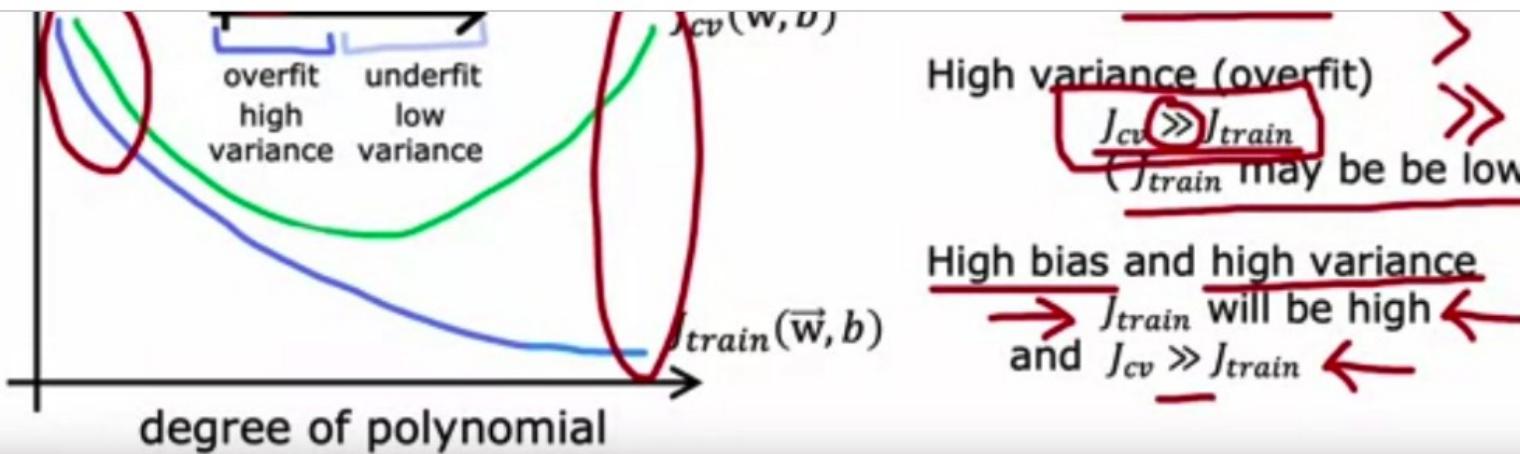
J_{train} will be high
 $(J_{train} \approx J_{cv})$

High variance (overfit)

$J_{cv} \gg J_{train}$
 $(J_{train} \text{ may be low})$

High bias and high variance

$\rightarrow J_{train}$ will be high
and $J_{cv} \gg J_{train}$



If the model's cross validation error J_{cv} is much higher than the training error J_{train} , this is an indication that the model has...

- high variance
- high bias
- Low variance
- Low bias

✓ Correct

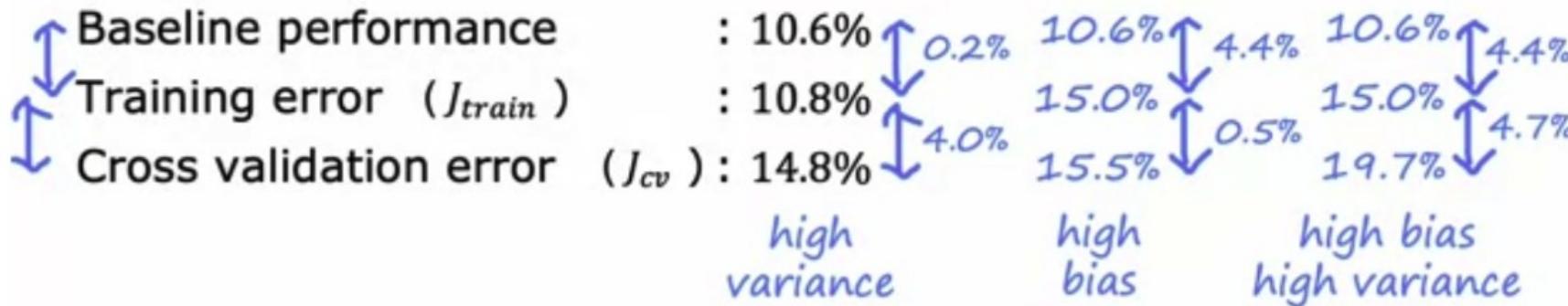
When $J_{cv} \gg J_{train}$ (whether J_{train} is also high or not, this is a sign that the model is overfitting to the training data and performing much worse on new examples.

examples.

2.

1 / 1 point

Bias/variance examples



Which of these is the best way to determine whether your model has high bias (has underfit the training data)?

See if the cross validation error is high compared to the baseline level of performance

high
variance

high
bias

high bias
high variance

Which of these is the best way to determine whether your model has high bias (has underfit the training data)?

- See if the cross validation error is high compared to the baseline level of performance
- See if the training error is high (above 15% or so)
- Compare the training error to the baseline level of performance
- Compare the training error to the cross validation error.



Correct

Correct. If comparing your model's training error to a baseline level of performance (such as human level performance, or performance of other well-established models), if your model's training error is much higher, then this is a sign that the model has high bias (has underfit).

3.

1 / 1 point

Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

$$1 \sum_{i=1}^m \text{...}^2 + \lambda \sum_{j=1}^n \text{...}^2$$

3.

1 / 1 point

Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

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

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 $x, x^2, \cancel{x}, \cancel{x^2}, y \dots$ fixes high variance
- Try getting additional features ← fixes high bias
- Try adding polynomial features $(x_1^2, x_2^2, x_1 x_2, \text{etc})$ ← fixes high bias
- Try decreasing λ ← fixes high bias
- Try increasing λ ← fixes high variance

You find that your algorithm has high bias. Which of these seem like good options for improving the algorithm's performance? Hint: two of these are correct.

- Get more training examples
- Try smaller sets of features $x, x^2, \cancel{x}, \cancel{x^2}, y \dots$
- Try getting additional features ←
- Try adding polynomial features $(x_1^2, x_2^2, x_1 x_2, \text{etc})$
- Try decreasing λ ←
- Try increasing λ ←

fixes high variance
fixes high variance
fixes high bias
fixes high bias
fixes high bias
fixes high variance

You find that your algorithm has high bias. Which of these seem like good options for improving the algorithm's performance? Hint: two of these are correct.

- Collect additional features or add polynomial features



Correct. More features could potentially help the model better fit the training examples.

- Collect more training examples

- Remove examples from the training set

- Decrease the regularization parameter λ (lambda)



Correct. Decreasing regularization can help the model better fit the training data.

Correct

Correct. Decreasing regularization can help the model better fit the training data.

4.

1 / 1 point

You find that your algorithm has a training error of 2%, and a cross validation error of 20% (much higher than the training error). Based on the conclusion you would draw about whether the algorithm has a high bias or high variance problem, which of these seem like good options for improving the algorithm's performance? Hint: two of these are correct.

- Reduce the training set size
- Collect more training data

Correct

Yes, the model appears to have high variance (overfit), and collecting more training examples would help reduce high variance.

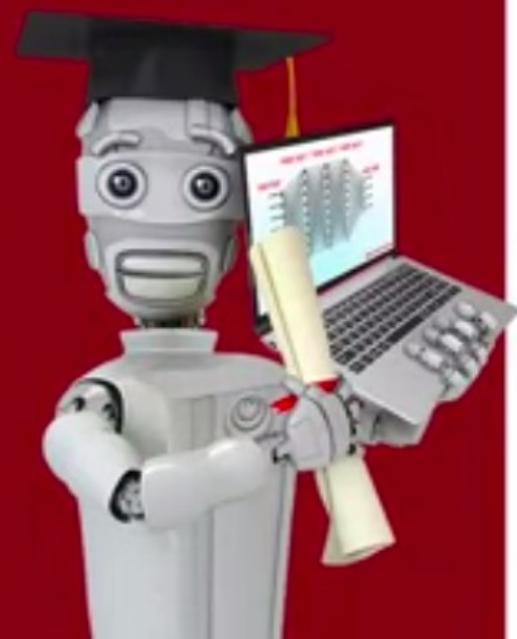
- Increase the regularization parameter λ

Correct

Yes, the model appears to have high variance (overfit), and increasing regularization would help reduce high variance.

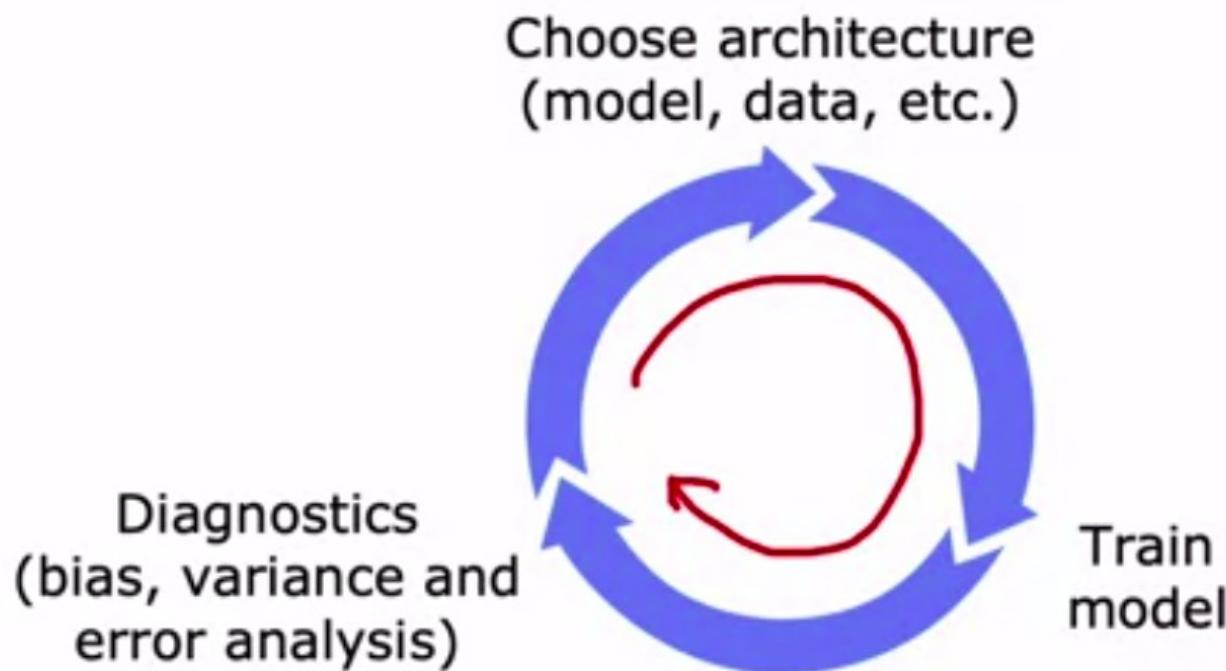
- Decrease the regularization parameter λ

Machine learning development process



Iterative loop of
ML development

Iterative loop of ML development



Spam classification example

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

Deal of the week! Buy now!
Rolex w4tchs - \$100
Medicine (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 $x_1, x_2, \dots, x_{10,000}$

$$\vec{x} = \begin{bmatrix} 0 \\ 1 \\ \textcolor{red}{\# \# 1} \\ 1 \\ 0 \\ \vdots \end{bmatrix} \quad \begin{array}{l} a \\ \text{andrew} \\ \text{buy} \\ \text{deal} \\ \text{discount} \\ \vdots \end{array}$$

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

Deal of the week! Buy now!
Rolex w4tchs - \$100
Medicine (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.

Machine learning development process

Error analysis



Error analysis

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

Algorithm misclassifies 100 of them.

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

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

Error analysis

$m_{cv} = \frac{500}{5000}$ examples in cross validation set.

Algorithm misclassifies 100 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

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.

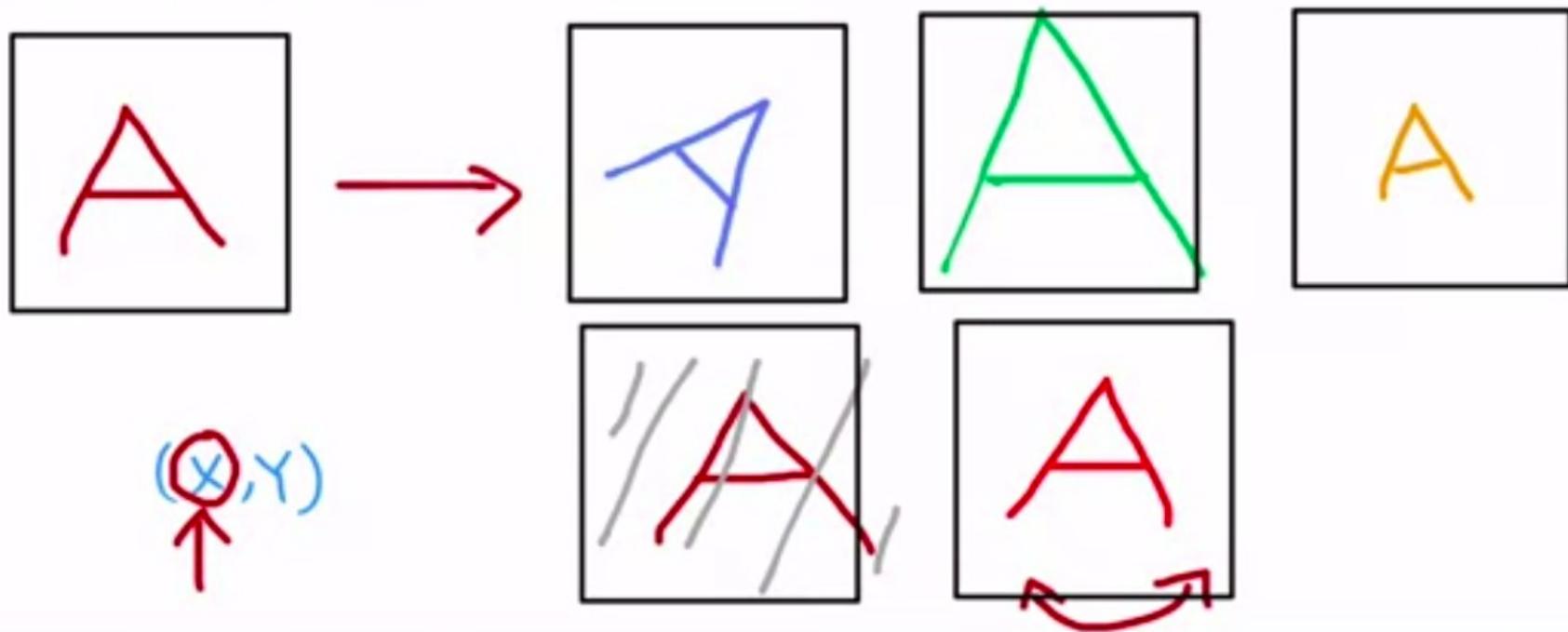
Pharma spam

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

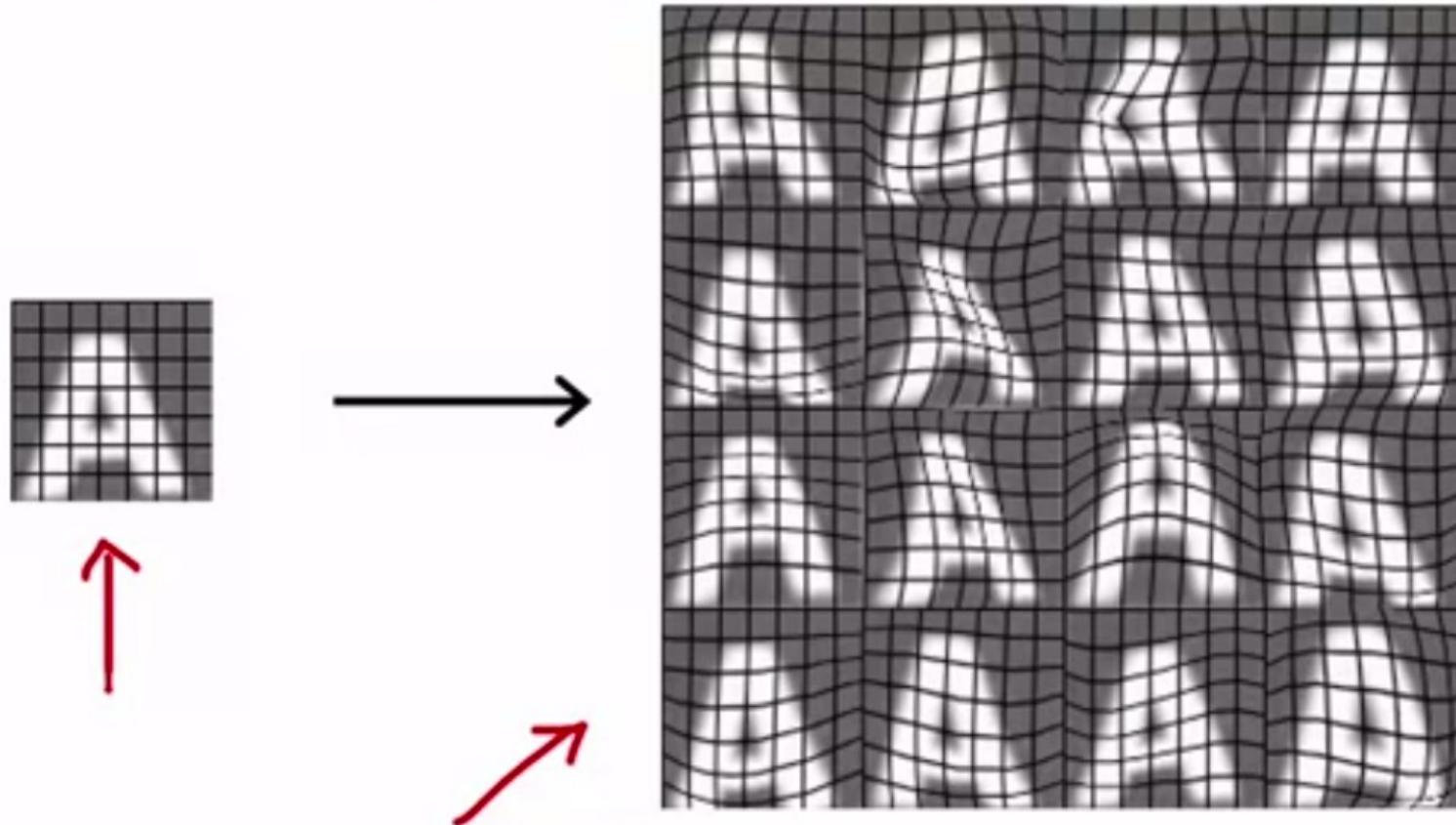
Beyond getting brand new training examples (x, y) , another technique: Data augmentation

Data augmentation

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



Data augmentation by introducing distortions



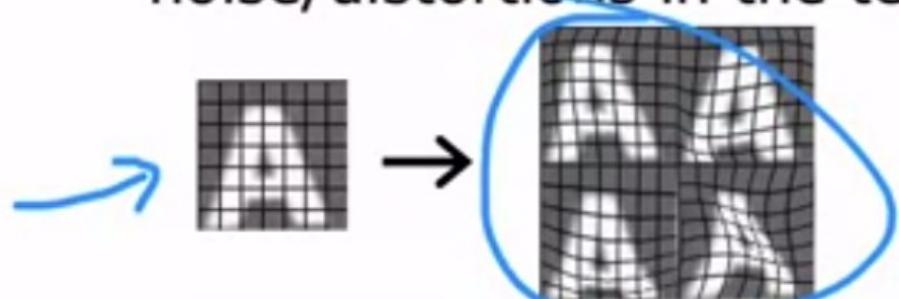
Data augmentation for speech

Speech recognition example

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

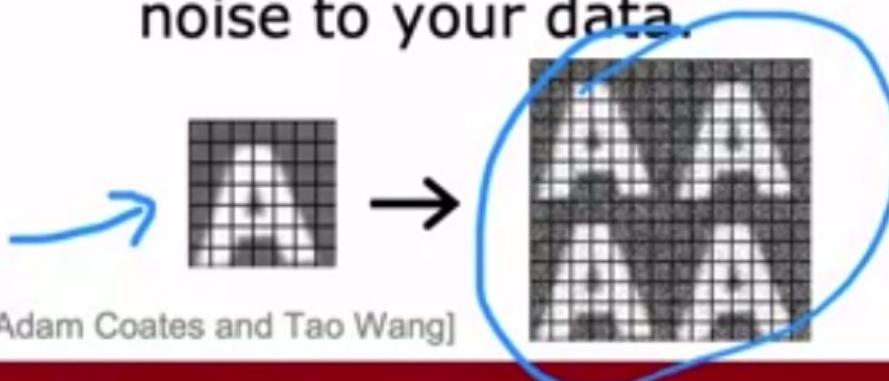
Data augmentation by introducing distortions

Distortion introduced should be representative 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
 $x_i \leftarrow x_i + \text{random noise}$

[Adam Coates and Tao Wang]

Data synthesis

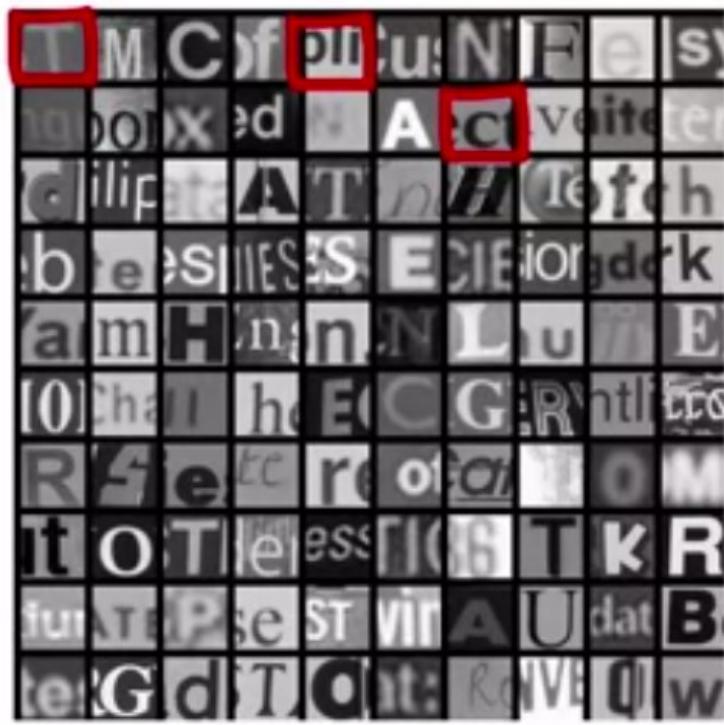
Synthesis: using artificial data inputs to create a new training example.

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

Abcdefg

*A*bcdefg

A
bcdefg

A
bcdefg

A
bcdefg

[Adam Coates and Tao Wang]

Artificial data synthesis for photo OCR



Real data



Synthetic data

[Adam Coates and Tao Wang]

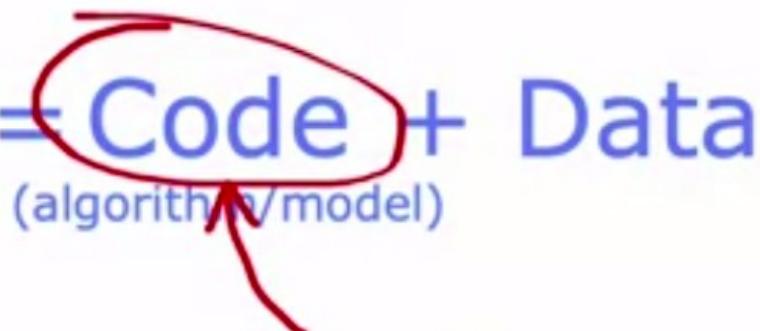
Engineering the data used by your system

Conventional
model-centric
approach:

$$\text{AI} = \text{Code} + \text{Data}$$

(algorithm/model)

work on this



Data-centric
approach:

$$\text{AI} = \text{Code} + \text{Data}$$

(algorithm/model)

work on this

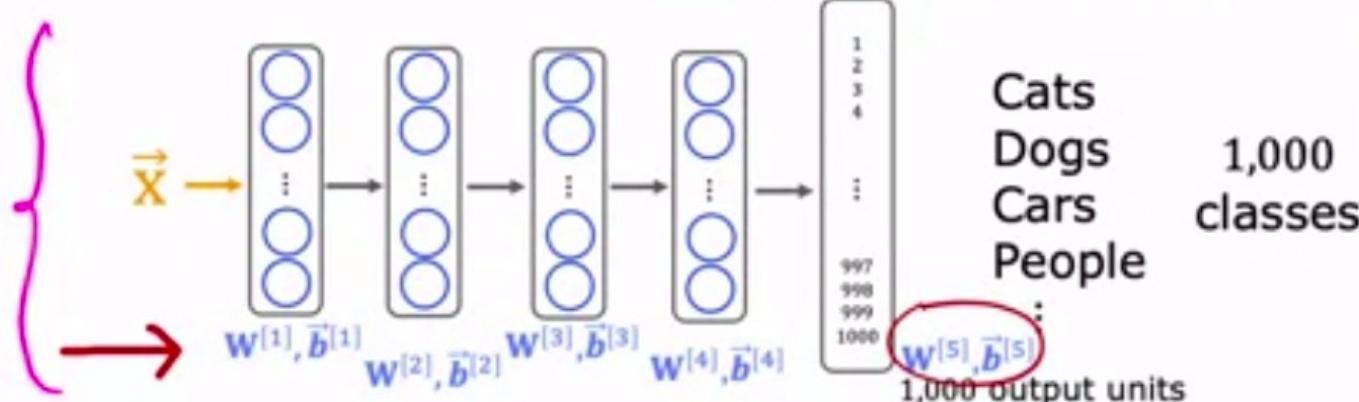


Machine learning development process



Transfer learning: using data
from a different task

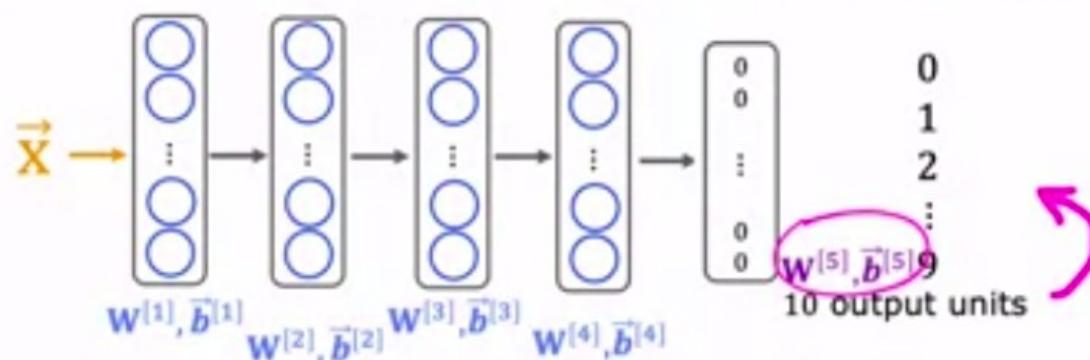
Transfer learning



1,2,3, ..., 9
1 million images

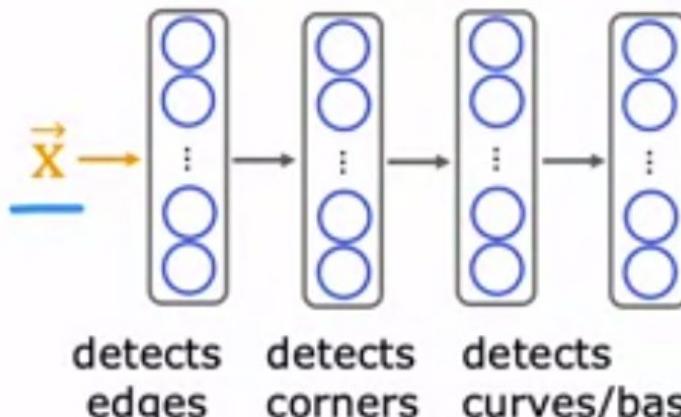
Supervised pretraining

fine tuning

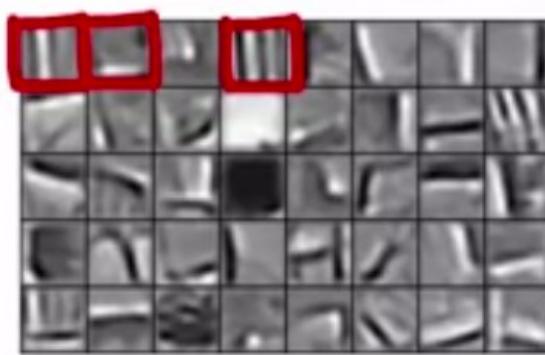


- Option 1: only train output layers parameters.
Option 2: train all parameters.

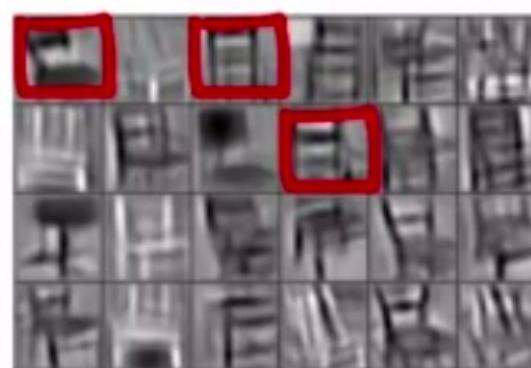
Why does transfer learning work?



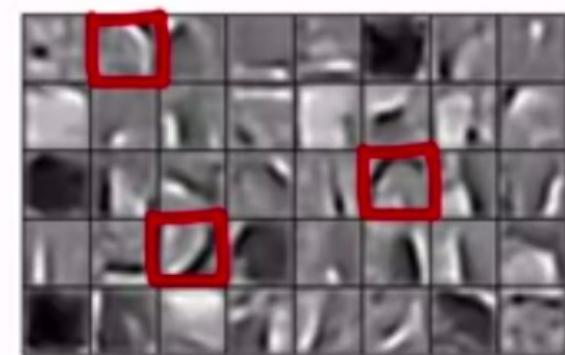
use the same input type



Edges



Corners



Curves / basic shapes

Transfer learning summary

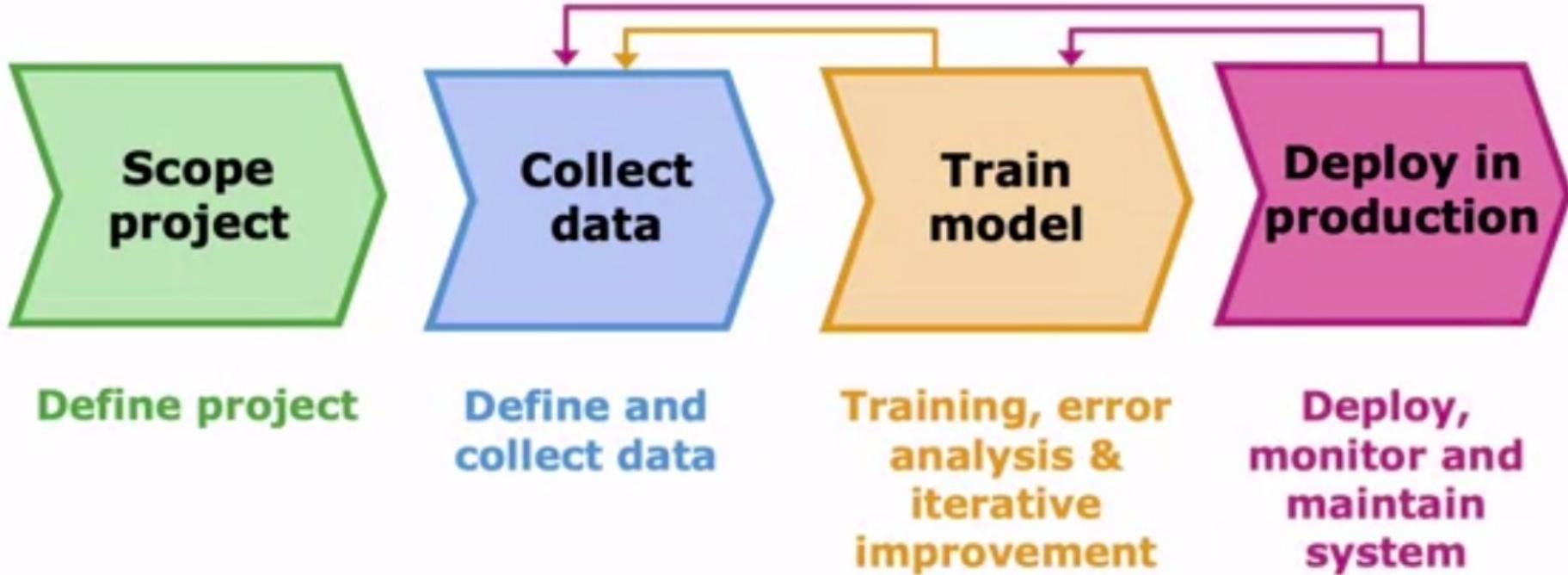
1. Download neural network parameters pretrained on a large dataset with same input type (e.g., images, audio, text) as your application (or train your own).
1 million images
2. Further train (fine tune) the network on your own data.
10 000 images
50 images

Machine learning development process

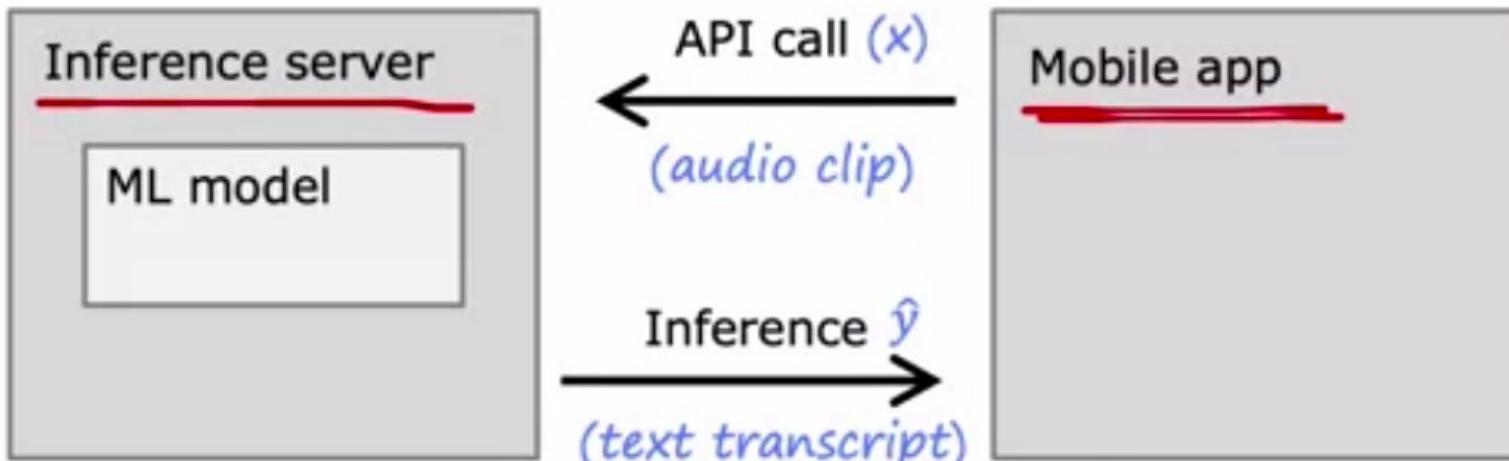


Full cycle of a
machine learning project

Full cycle of a machine learning project



Deployment



→ Software engineering may be needed for:

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

MLOps
machine learning
operations

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



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.

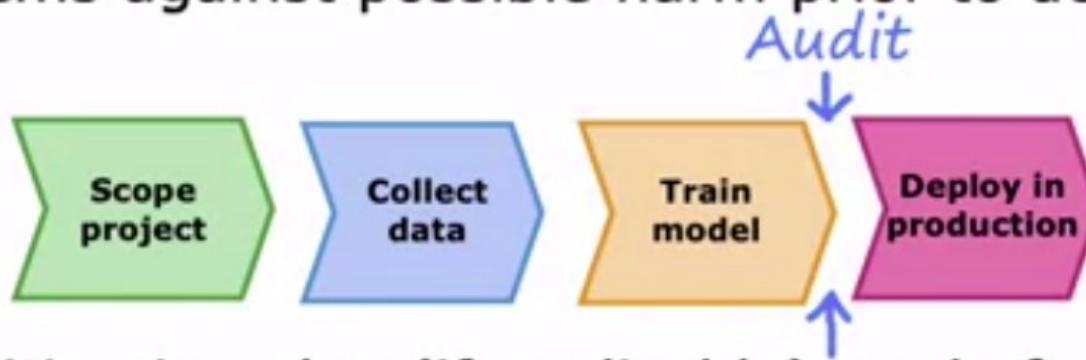
Spam vs anti-spam : fraud vs anti-fraud.

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.

1.

1 / 1 point

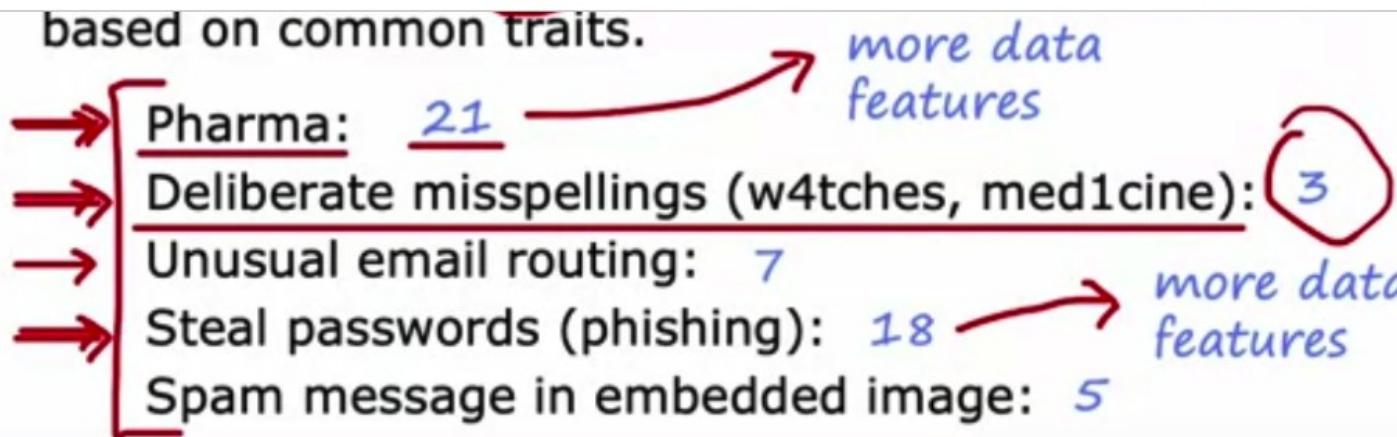
Error analysis

$m_{cv} = \frac{500}{5000}$ examples in cross validation set.

Algorithm misclassifies 100 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



Which of these is a way to do error analysis?

- Collecting additional training data in order to help the algorithm do better.
- Calculating the test error J_{test}
- Calculating the training error J_{train}
- Manually examine a sample of the training examples that the model misclassified in order to identify common traits and trends.

Correct

Correct. By identifying similar types of errors, you can collect more data that are similar to these misclassified examples in order to train the model to improve on these types of examples.

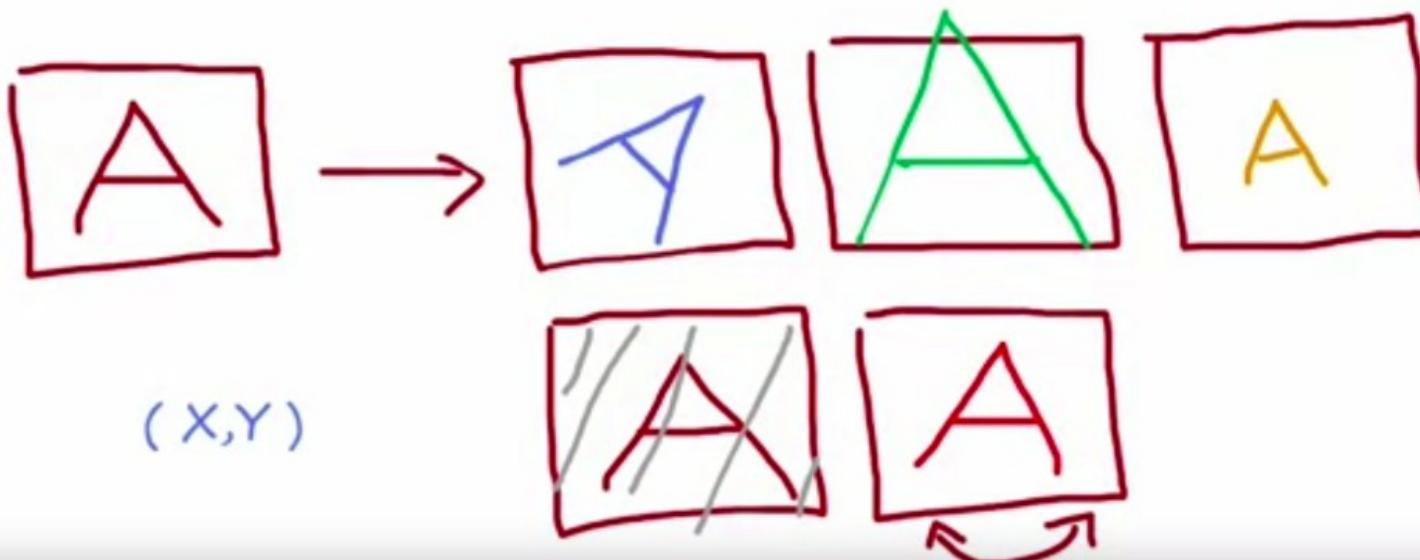
on these types or examples.

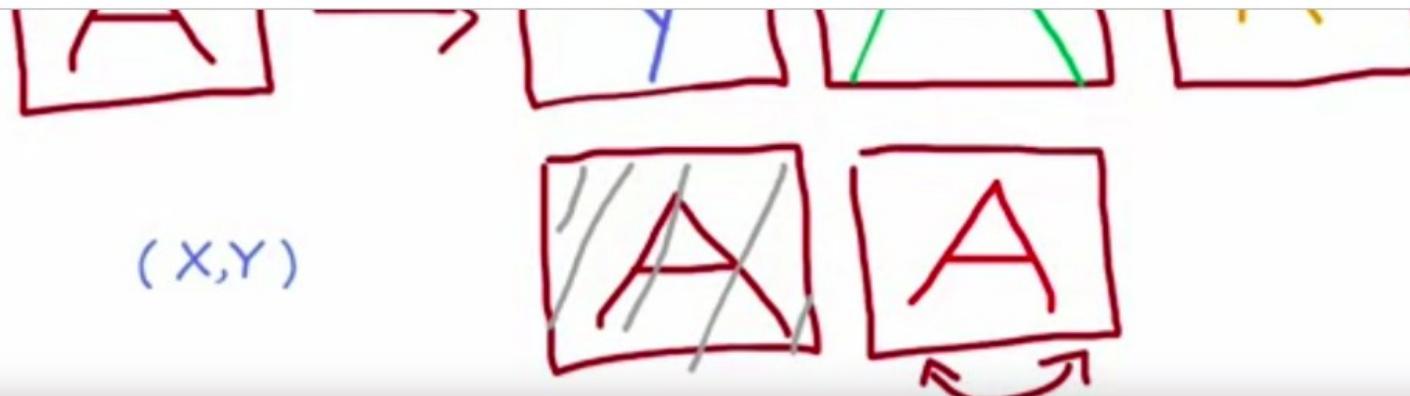
2.

1 / 1 point

Data augmentation

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





We sometimes take an existing training example and modify it (for example, by rotating an image slightly) to create a new example with the same label. What is this process called?

- Error analysis
- Data augmentation
- Machine learning diagnostic
- Bias/variance analysis

✓ Correct

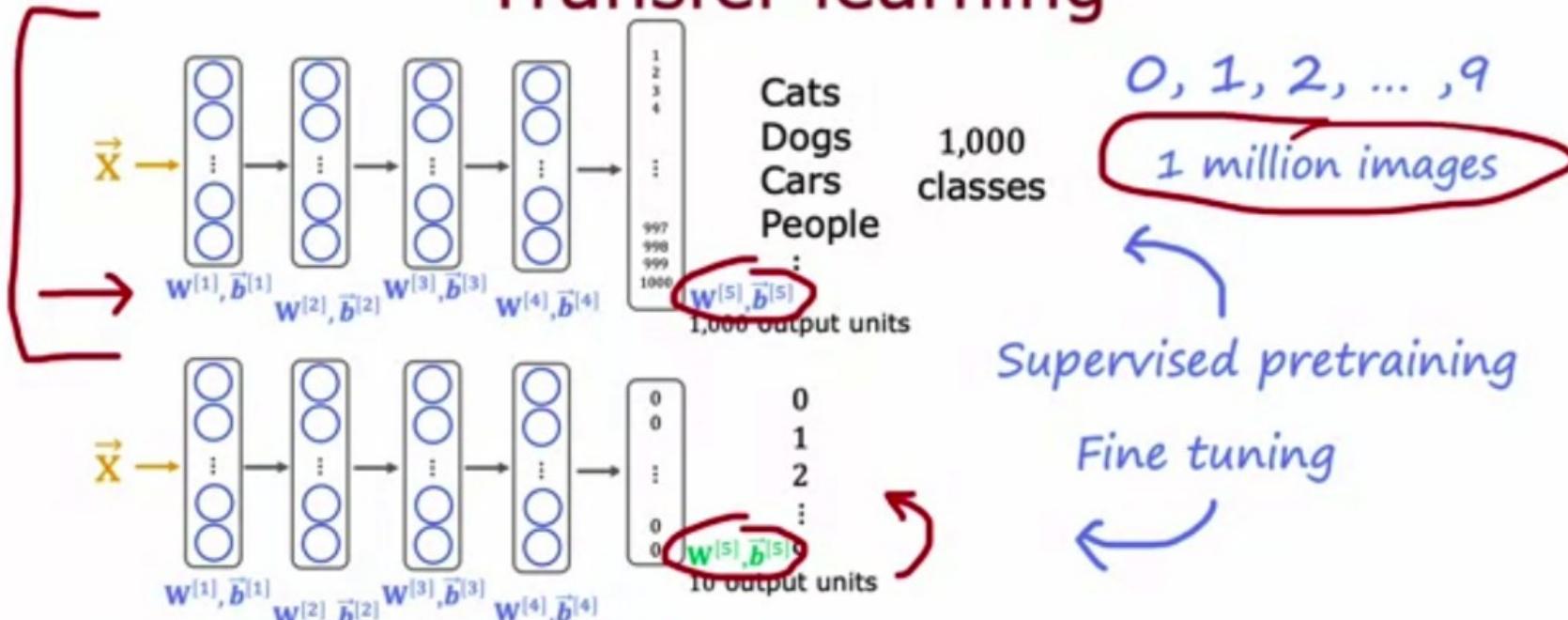
Yes! Modifying existing data (such as images, or audio) is called data augmentation.

Yes! Modifying existing data (such as images, or audio) is called data augmentation.

3.

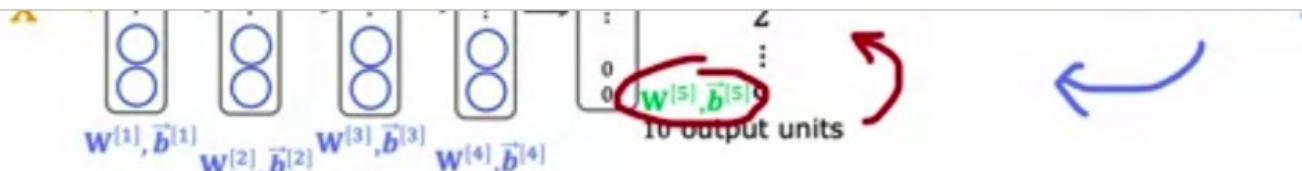
1 / 1 point

Transfer learning



Option 1: only train **output layers** parameters.

Option 2: **train all parameters**.



Option 1: only train **output layers** parameters.

Option 2: **train all parameters**.

What are two possible ways to perform transfer learning? Hint: two of the four choices are correct.

- You can choose to train just the output layers' parameters and leave the other parameters of the model fixed.

Correct

Correct. The earlier layers of the model may be reusable as is, because they are identifying low level features that are relevant to your task.

- Given a dataset, pre-train and then further fine tune a neural network on the same dataset.

- Download a pre-trained model and use it for prediction without modifying or re-training it.

- You can choose to train all parameters of the model, including the output layers, as well as the earlier layers.

Correct

Correct. It may help to train all the layers of the model on your own training set. This may take more time compared to if you just trained the parameters of the output layers.

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 1% error on test set
(99% correct diagnoses)

Only 0.5% of patients have the disease

`print("y=0")`

99.5% accuracy, 0.5% error

1%

1.2%

less

Usefulness

more

Precision/recall

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

Actual Class			
		1 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?)

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

Recall:

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

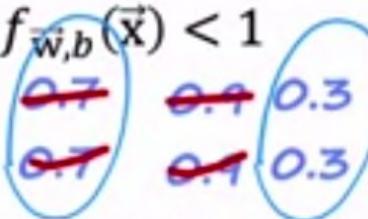
$$\frac{\text{True positives}}{\#\text{actual positive}} = \frac{\text{True positives}}{\text{True pos} + \text{False neg}} = \frac{15}{15+10} = 0.6$$

Trading off precision and recall

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

→ Predict 1 if $f_{\vec{w}, b}(\vec{x}) \geq 0.5$

→ Predict 0 if $f_{\vec{w}, b}(\vec{x}) < 0.5$



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

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

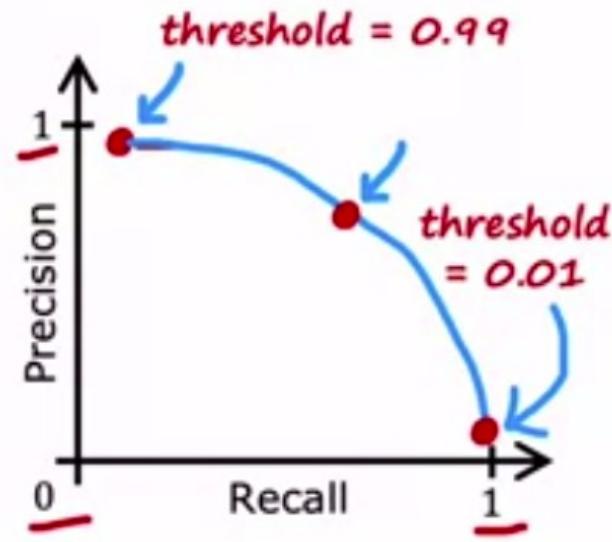
Suppose we want to predict $y = 1$ (rare disease) only if very confident.

higher precision, lower recall

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

lower precision, higher recall

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



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")`

Harmonic
mean

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

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

Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

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

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