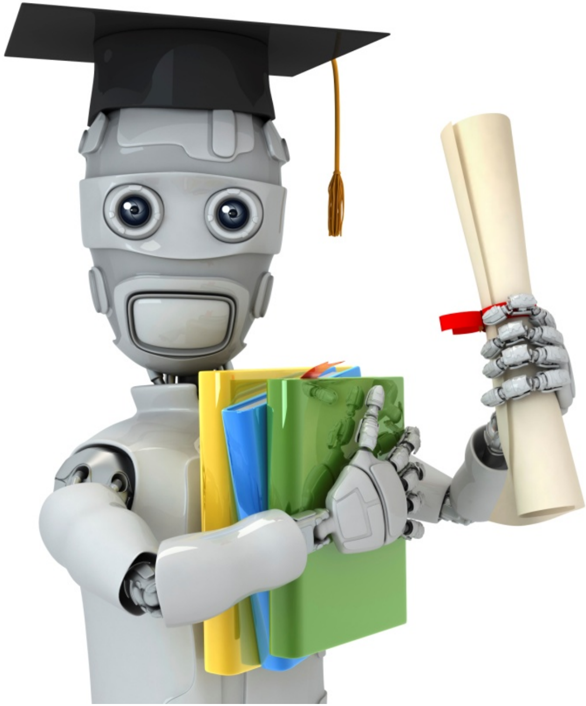


Machine Learning

Advice for applying machine learning

Deciding what to try next

Debugging a learning algorithm:

Suppose you have implemented regularized linear regression to predict housing prices.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{m} \theta_j^2\right]$$

However, when you test your hypothesis on a new set of houses, you find that it makes unacceptably large errors in its predictions. What should you try next?

- Get more training examples get more data, but sometimes more data does not help!
- Try smaller sets of features $x_1, x_2, x_3, \ldots, x_{100}$
- Try getting additional features add more features!
- Try adding polynomial features $(x_1^2, x_2^2, x_1 x_2, \text{etc.})$
- Try decreasing λ
- Try increasing λ

what many people will do is just randomly pick one of these options!
there is a simple technique can easily rule out half of the options!

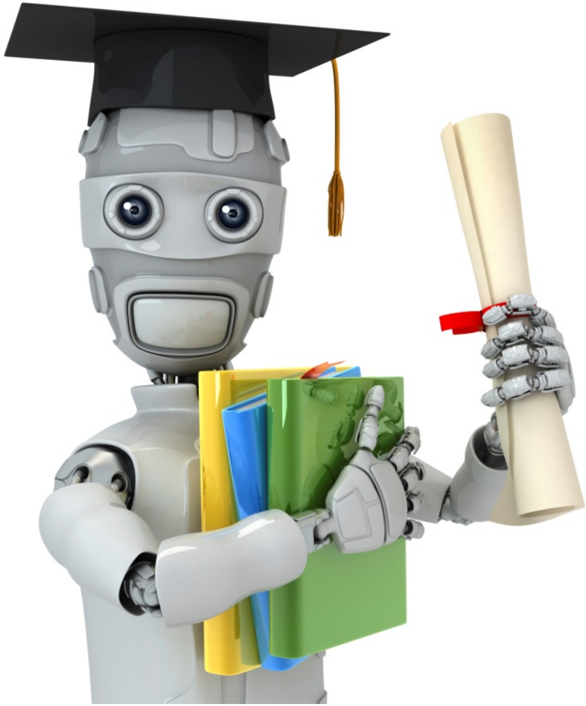
Machine learning diagnostic:

how to evaluate a machine learning algorithm!

Diagnostic: A test that you can run to gain insight what is/isn't working with a learning algorithm, and gain guidance as to how best to improve its performance.

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

can take some time but can be very helpful by doing this!

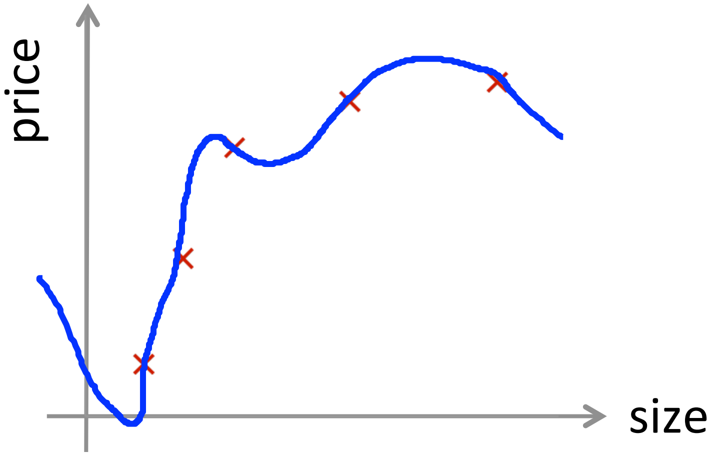


Machine Learning

Advice for applying machine learning

Evaluating a hypothesis

Evaluating your hypothesis



$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

can overfit!

Fails to generalize to new examples not in training set.

but for problems with a lot of features, it is impossible to visualize the overfitting issue! so we need other ways to evaluate hypothesis!

$x_1 =$ size of house

$x_2 =$ no. of bedrooms

$x_3 =$ no. of floors

$x_4 =$ age of house

$x_5 =$ average income in neighborhood

$x_6 =$ kitchen size

$\vdots$

x_{100}

Evaluating your hypothesis

standard way to evaluate a hypothesis!

Dataset:

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

random 70 %

70%

Training set

kind like identification and validation in system identification!

$$(x^{(1)}, y^{(1)})$$
$$(x^{(2)}, y^{(2)})$$
$$\vdots$$
$$(x^{(m)}, y^{(m)})$$

split the data we have into two portion!

random 30 %

30%

Test set

$$(x^{(1)}_{test}, y^{(1)}_{test})$$
$$(x^{(2)}_{test}, y^{(2)}_{test})$$
$$\vdots$$
$$(x^{(m_{test})}_{test}, y^{(m_{test})}_{test})$$

# of test examples!

m_{test} = no. of test example

$(x^{(i)}_{test}, y^{(i)}_{test})$

Andrew Ng

Training/testing procedure for linear regression

standard procedure!

- Learn parameter θ from training data (minimizing training error $J(\theta)$)

70%

- Compute test set error:

$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \left(h_\theta(x^{(i)}_{test}) - y^{(i)}_{test} \right)^2$$

the avg square error of your test examples!

Training/testing procedure for logistic regression

similar to the cases in linear regression!

- Learn parameter θ from training data
- Compute test set error:

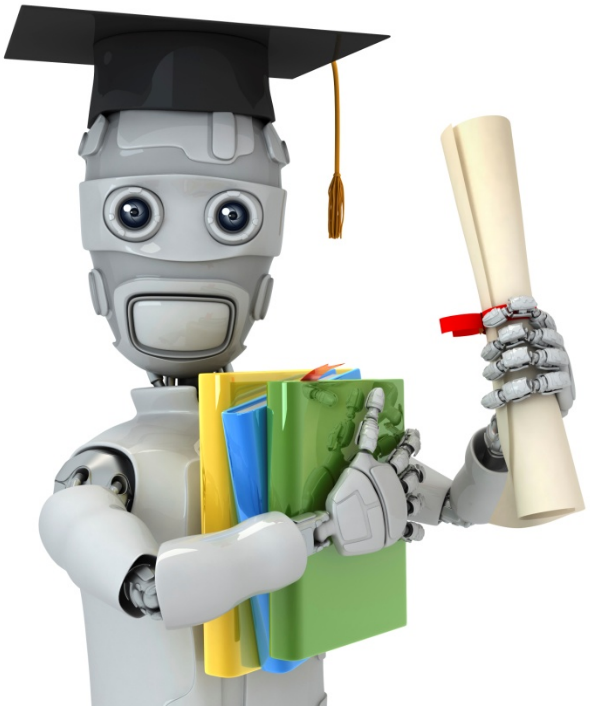
$$\boxed{J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)}) + (1 - y_{test}^{(i)}) \log h_\theta(x_{test}^{(i)})}$$

- Misclassification error (0/1 misclassification error):

other test error measure!

$$err(h_\theta(x), y) = \begin{cases} 1 & \text{if } h \geq 0.5,\ y=0 \text{ or if } h<0.5,\ y=1 \\ 0 & \text{o.w} \end{cases}$$

$$\text{Test error} = \sum err(h_\theta, y)$$

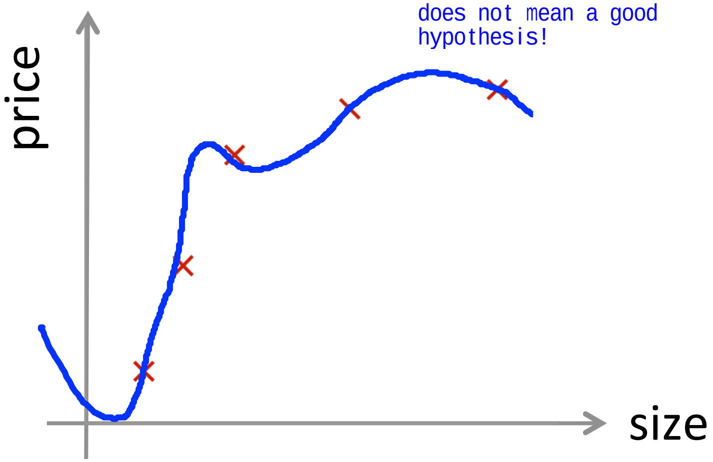


Machine Learning

Advice for applying machine learning

Model selection and training/validation/test sets

Overfitting example



$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Once parameters $\theta_0, \theta_1, \ldots, \theta_4$ were fit to some set of data (training set), the error of the parameters as measured on that data (the training error $J(\theta)$) is likely to be lower than the actual generalization error.

which model to select!

degree of polynomial you want to pick!

d = degree of polynomial

take each model and compute the test error!

Model selection

$d=1$ 1. $h_\theta(x) = \theta_0 + \theta_1 x \longrightarrow \Theta^{(1)} \longrightarrow J_{test}(\Theta^{(1)})$

$d=2$ 2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \longrightarrow \Theta^{(2)} \longrightarrow J_{test}(\Theta^{(2)})$

$d=3$ 3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3 \longrightarrow \Theta^{(3)} \longrightarrow J_{test}(\Theta^{(3)})$

$\vdots$

$d=10$ 10. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10} \longrightarrow \Theta^{(10)} \longrightarrow J_{test}(\Theta^{(10)})$

say if d = 5 is the best!

Choose $\boxed{\theta_0 + \ldots \theta_5 x^5}$

$\Theta_0, \Theta_1, \ldots$

How well does the model generalize? Report test set error $J_{test}(\theta^{(5)})$.

$\Theta^{(5)}$

Problem: $J_{test}(\theta^{(5)})$ is likely to be an optimistic estimate of generalization error. I.e. our extra parameter (d = degree of polynomial) is fit to test set.

d is picked using the test set, so it likely to do well with the test set!

Andrew Ng

this is what we do to evaluate
a hypothesis!

Evaluating your hypothesis

Dataset: we break the data into 3 pieces!

these should be randomly chosen!

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

60% Training set

$(x^{(1)}, y^{(1)})$
$(x^{(2)}, y^{(2)})$
$\vdots$
$(x^{(m)}, y^{(m)})$

20% Cross validation set (CV)

we just the validation set

$(x_{cv}^{(1)}, y_{cv}^{(1)})$
$(x_{cv}^{(2)}, y_{cv}^{(2)})$
$\vdots$
$(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$

m_{cv} = no. of cv example

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

20% test set

$(x_{test}^{(1)}, y_{test}^{(1)})$
$(x_{test}^{(2)}, y_{test}^{(2)})$
$\vdots$
$(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

m_{test}

Train/validation/test error

Training error:

$$\boxed{J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}$$

$J(\theta)$

Cross Validation error:

$$\boxed{J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} (h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2}$$

Test error:

$$\boxed{J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} (h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2}$$

Model selection

test these hypothesis on cross validation set and pick the model with the lowest validation error!

d=1 1. $h_\theta(x) = \theta_0 + \theta_1 x \rightarrow \min_\theta J(\theta) \rightarrow \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)})$

d=2 2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \rightarrow \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)})$

d=3 3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3 \rightarrow \theta^{(3)}$

$J_{cv}(\theta^{(4)})$

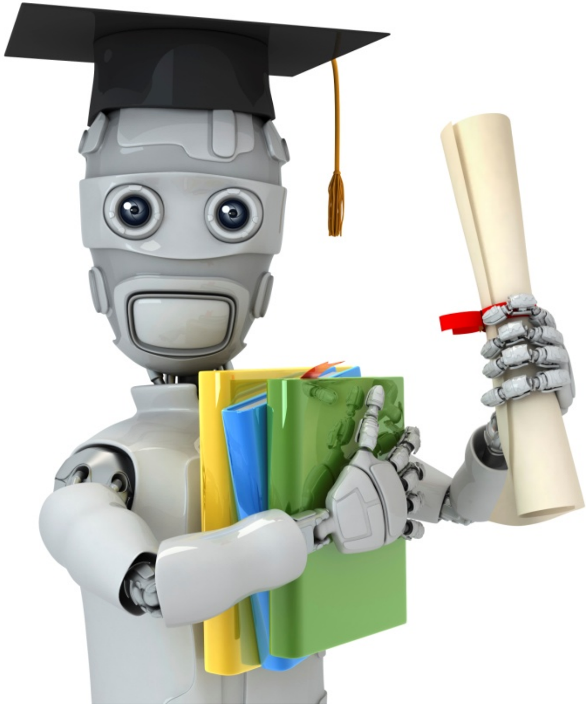
$\vdots$

d=10 10. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10} \rightarrow \theta^{(10)} \rightarrow J_{cv}(\theta^{(10)})$

$d = 4$

Pick $\theta_0 + \theta_1 x_1 + \cdots + \theta_4 x^4$ ← say if the 4th order is the best!

Estimate generalization error for test set $\boxed{J_{test}(\theta^{(4)})}$ ←



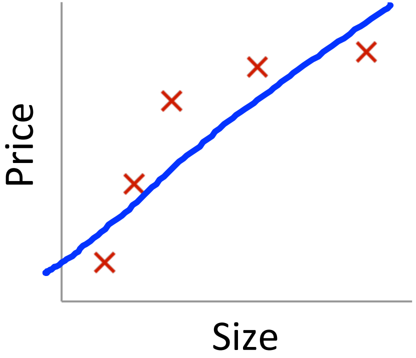
Machine Learning

Advice for applying machine learning

Diagnosing bias vs. variance

Bias/variance

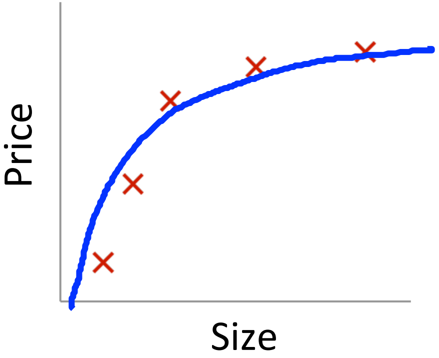
high variance: the error for a set of new prediction can be large since overfit hypothesis often fails to predict general data set.



$\theta_0 + \theta_1 x$

High bias (underfit)

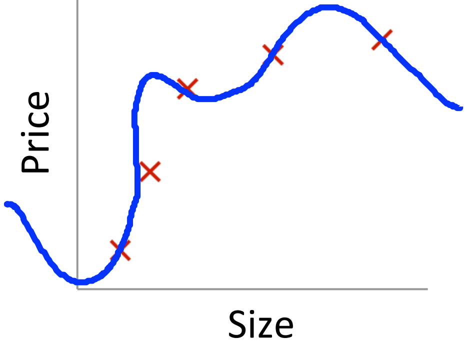
d=1



$\theta_0 + \theta_1 x + \theta_2 x^2$

"Just right"

d=2



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

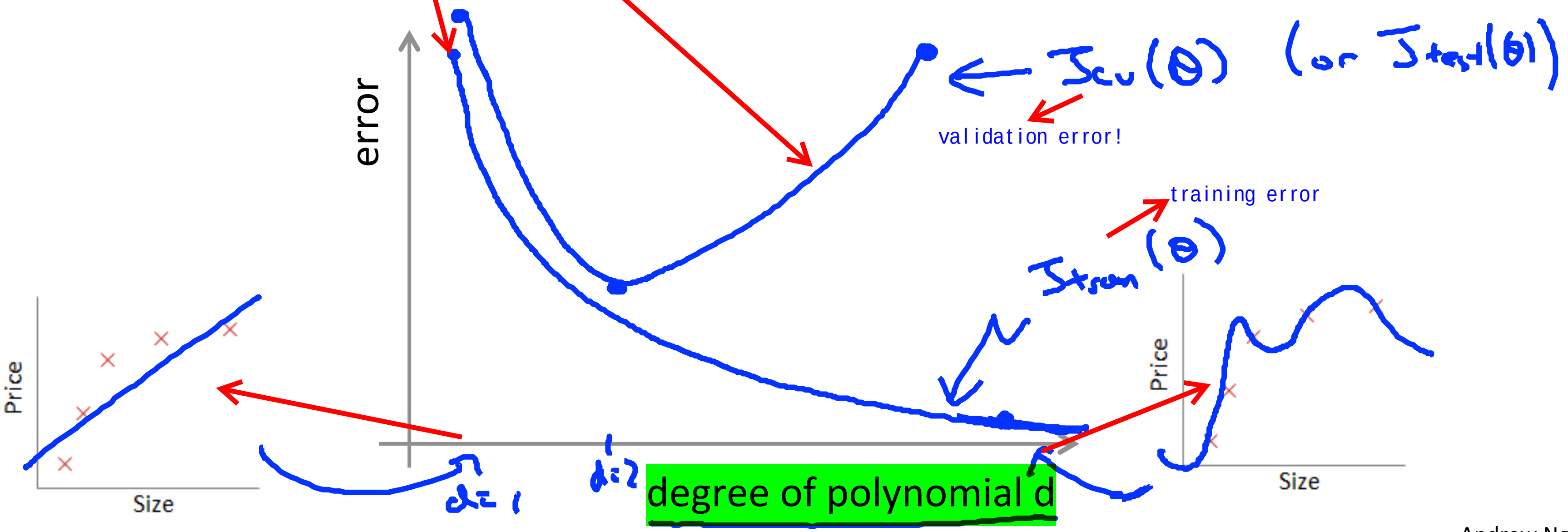
High variance (overfit)

d=4

Bias/variance

Training error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

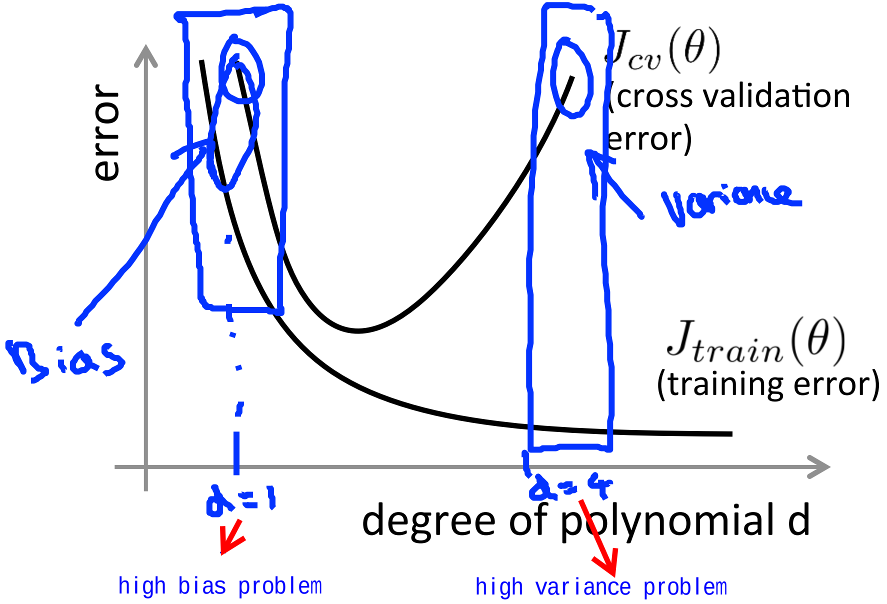
Cross validation error: $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x^{(i)}_{cv}) - y^{(i)}_{cv})^2$ (or $J_{test}(\theta)$)



Andrew Ng

Diagnosing bias vs. variance

Suppose your learning algorithm is performing less well than you were hoping. ($J_{cv}(\theta)$ or $J_{test}(\theta)$ is high.) Is it a bias problem or a variance problem?



Bias (underfit):

→ $J_{train}(\theta)$ will be high

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

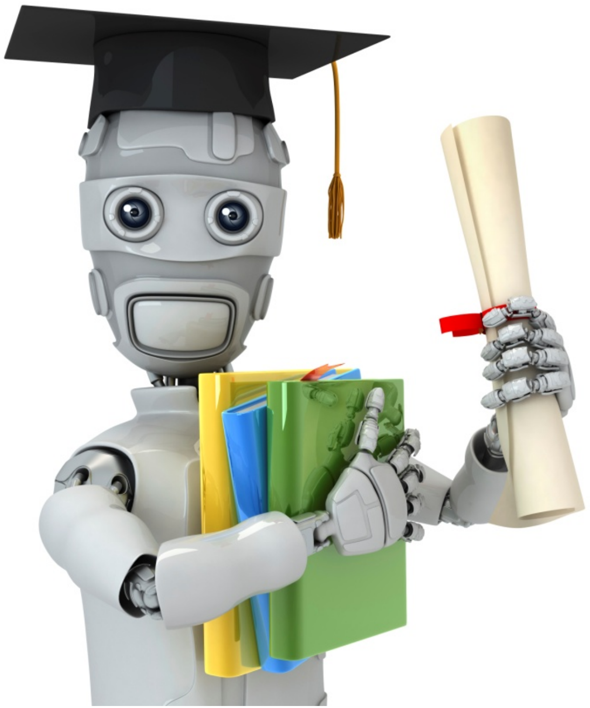
training error

Variance (overfit):

→ $J_{train}(\theta)$ will be low

$J_{cv}(\theta) \gg J_{train}(\theta)$

$\gg$



Machine Learning

Advice for applying machine learning

Regularization and bias/variance

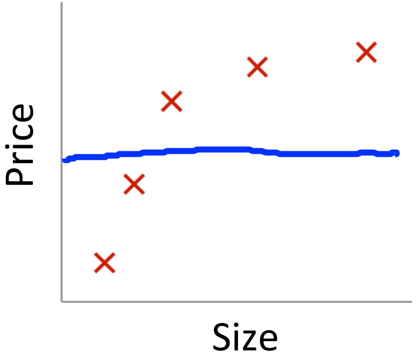
how regularization can affect bias and variance

Linear regression with regularization

Model: $\boxed{h_\theta(x) = \theta_0} + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \boxed{\frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2}$$

regularization term to prevent overfitting

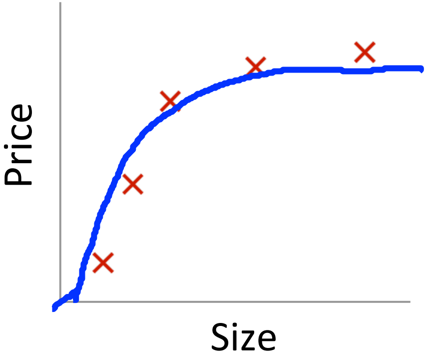


Large λ

High bias (underfit)

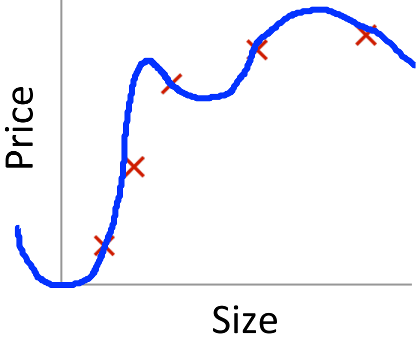
$\lambda = 10000.\ \theta_1 \approx 0, \theta_2 \approx 0, \ldots$

$h_\theta(x) \approx \theta_0$



Intermediate λ

"Just right"



Small λ

High variance (overfit)

$\lambda = 0$

Choosing the regularization parameter λ

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

model

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \underline{\frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2}$$

cost function

$$\boxed{J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2}$$

$$\boxed{J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x^{(i)}_{cv}) - y^{(i)}_{cv})^2}$$

$$\boxed{J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x^{(i)}_{test}) - y^{(i)}_{test})^2}$$

$J(\theta)$

J_{train}

J_{cv}

J_{test}

without regularization term

Andrew Ng

Choosing the regularization parameter λ

Model: $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

to get thetas from diff lambdas

then use cross validation set to validate them!

1. Try $\lambda = 0$ → $\min_\theta J(\theta)$ → $\theta^{(1)}$ → $J_{cv}(\theta^{(1)})$
2. Try $\lambda = 0.01$ → $\min_\theta J(\theta)$ → $\theta^{(2)}$ → $J_{cv}(\theta^{(2)})$
3. Try $\lambda = 0.02$ → $\theta^{(3)}$ → $J_{cv}(\theta^{(3)})$
4. Try $\lambda = 0.04$
5. Try $\lambda = 0.08$ → $\theta^{(5)}$ → $J_{cv}(\theta^{(5)})$

$\vdots$

12. Try $\lambda = 10$ → $\theta^{(12)}$ → $J_{cv}(\theta^{(12)})$

10.24

the we pick the one with the lowest cross validation error!

Pick (say) $\theta^{(5)}$. Test error: $J_{test}(\theta^{(5)})$

report the test error using test set data

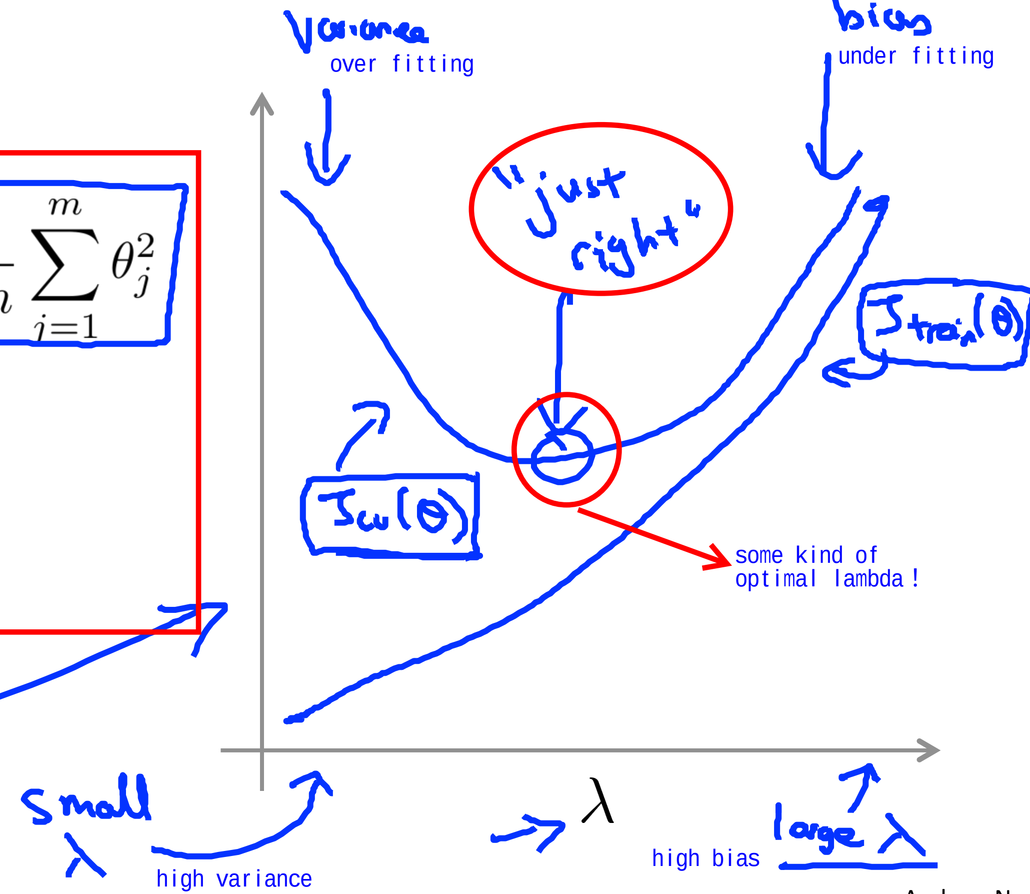
Andrew Ng

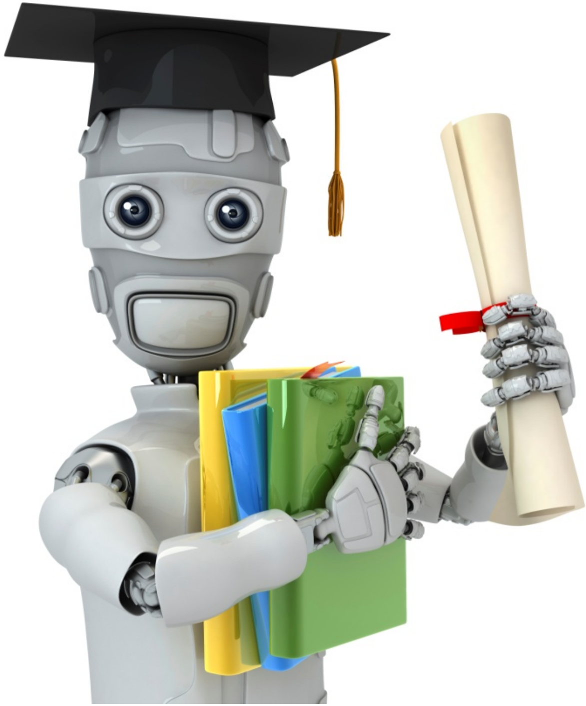
Bias/variance as a function of the regularization parameter λ

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \boxed{\frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2}$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\boxed{J_{cv}(\theta)} = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$





Machine Learning

Advice for applying machine learning

Learning curves

A tool to diagnose a learning algorithm (use very often!!!)

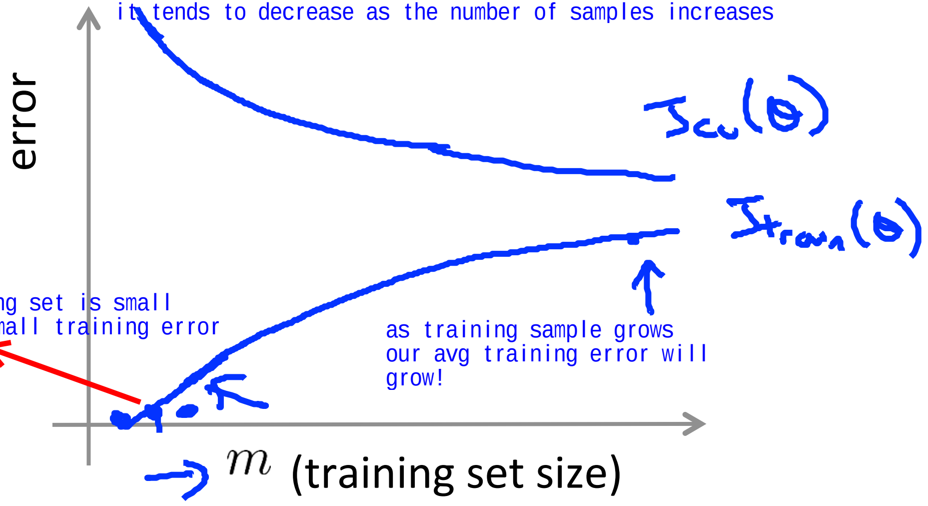
a very useful thing to plot. check your algorithm is working correctly, or you want to improve the performance of the algorithm

Learning curves

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

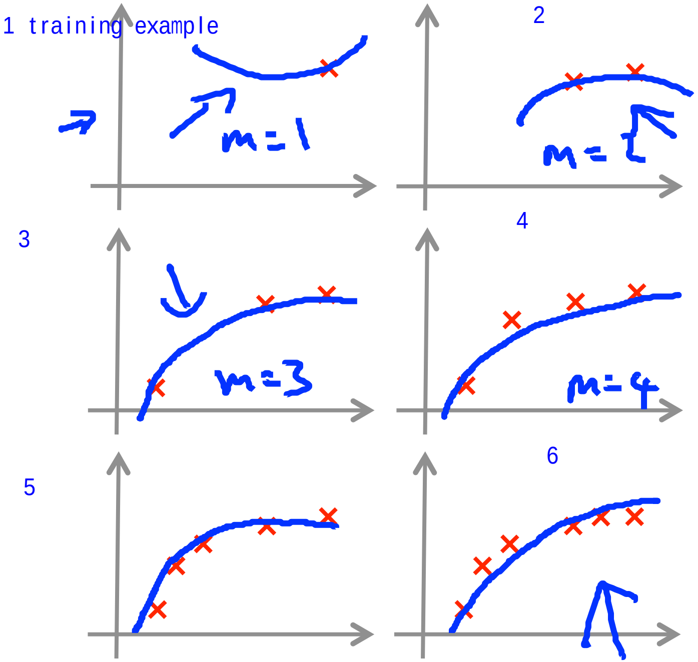
cross validation error
error on data we have't seen before!
it tends to decrease as the number of samples increases



if training set is small
we have small training error
as well!

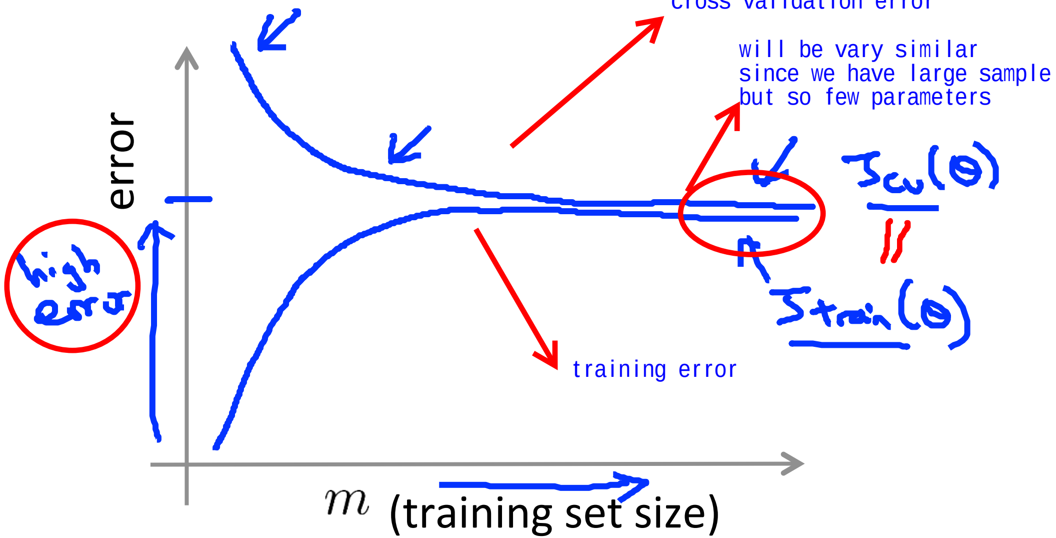
as training sample grows
our avg training error will
grow!

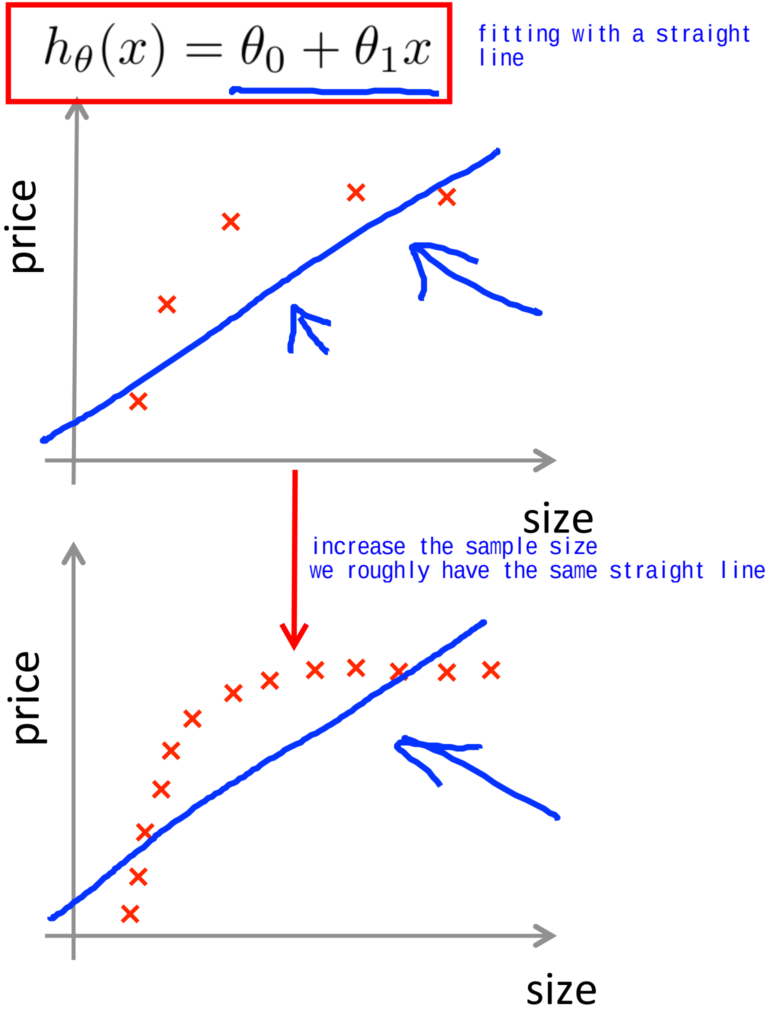
$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2}$$



Andrew Ng

High bias





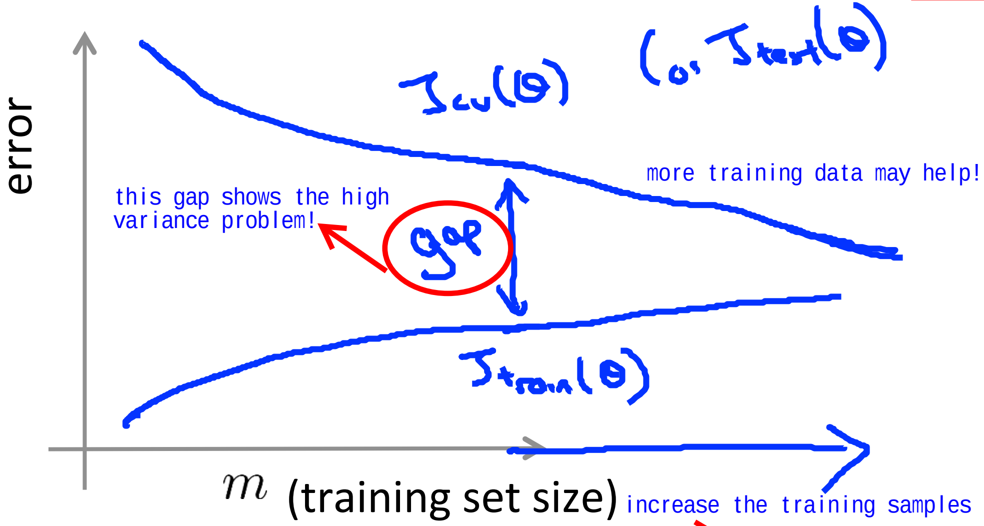
If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.

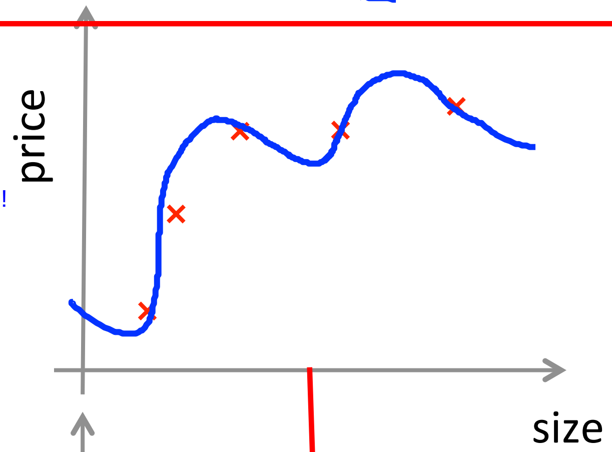
fitting with HOD polynomial

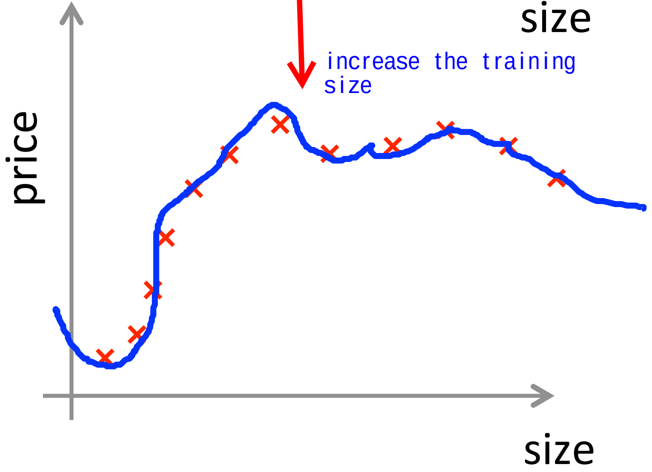
High variance

$$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{100}$$

(and small λ)

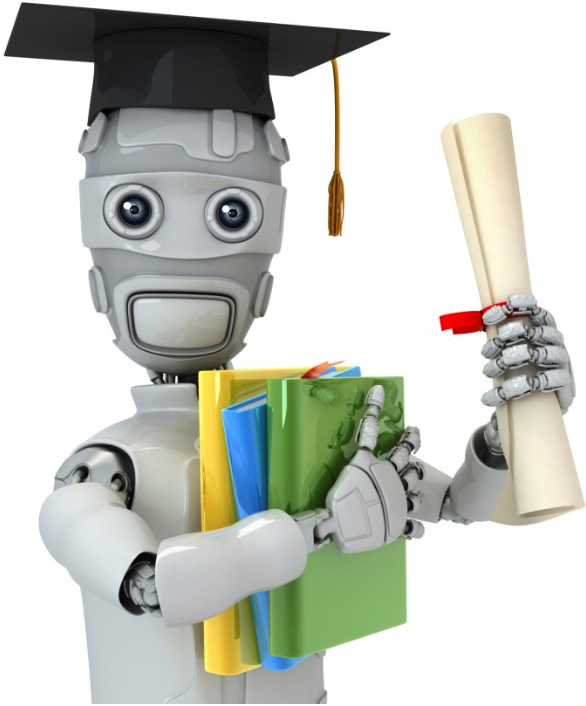






If a learning algorithm is suffering from high variance, getting more training data is likely to help.

Andrew Ng



Machine Learning

Advice for applying machine learning

Deciding what to try next (revisited)

Debugging a learning algorithm:

Suppose you have implemented regularized linear regression to predict housing prices. However, when you test your hypothesis in a new set of houses, you find that it makes unacceptably large errors in its prediction. What should you try next?

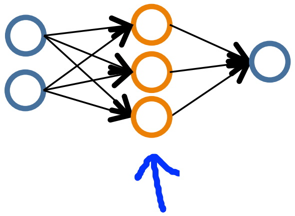
some advices

- Get more training examples → fixes high variance
- Try smaller sets of features → 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

Neural networks and overfitting

advices of How Andrew chooses the architecture of NN

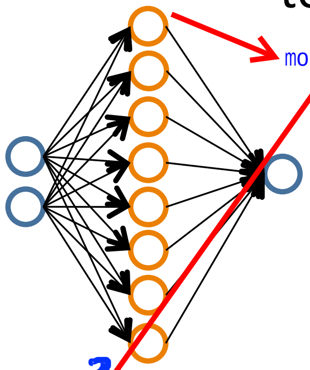
"Small" neural network
(fewer parameters; more prone to underfitting)



Computationally cheaper

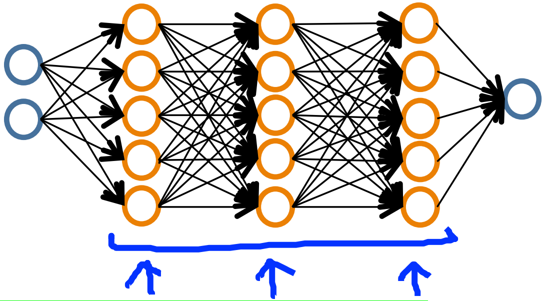
"Large" neural network
(more parameters; more prone to overfitting)

more hidden unit



either

more hidden layers



Computationally more expensive.

Use regularization (λ) to address overfitting.

$J_{cv}(\theta)$

usually use large NN with regularization is mode effective than small NN.