

10. ADVICE FOR APPLYING MACHINE LEARNING

Debugging learning algorithms

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

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

However you find out that your hypothesis on a new set of houses makes unacceptable large errors on predictions.

What do you try next?

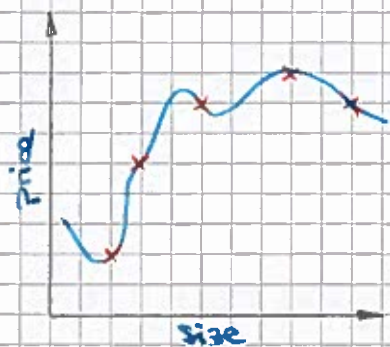
- get more training examples
- try smaller sets of features
- try getting additional features
- try add polynomial features
- try increasing / decreasing λ

} might lead to long Projects if you take the wrong road

Machine learning diagnostics:

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

Diagnostic can take time to implement, but doing so can be very good use of time.



Fails to generalize to new examples not in training set.

Features

x_1 = size of house

\vdots = \vdots

x_{100} = kitchen size

is hard or impossible to plot the hypothesis

$$h(\mathbf{x}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Data set:

	size	Price	
70%	2104	400	Training set
	\vdots	\vdots	
	1745	340	
	1375	212	Test set
30%	\vdots	\vdots	
	2473	365	

$$= \begin{pmatrix} x^{(1)} & y^{(1)} \\ x^{(2)} & y^{(2)} \\ \vdots & \vdots \end{pmatrix}$$

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

$$x_{\text{test}}^{(n)}, y_{\text{test}}^{(n)}$$

m_{test} = # of test examples

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

Training/testing procedure for linear regression

- learn parameter θ from training data (minimizing training error $J(\theta)$)
- Compute test set error:

$$J_{\text{test}}(\theta) = \frac{1}{2m_{\text{test}}} \cdot \sum_{i=1}^{m_{\text{test}}} (\underbrace{h(\mathbf{x}_{\text{test}}^{(i)}) - y_{\text{test}}^{(i)}}_{\text{measure test set error}})^2$$

=> Randomly choose the training set (70%) and test set (30%)

Training / testing procedure for logistic regression

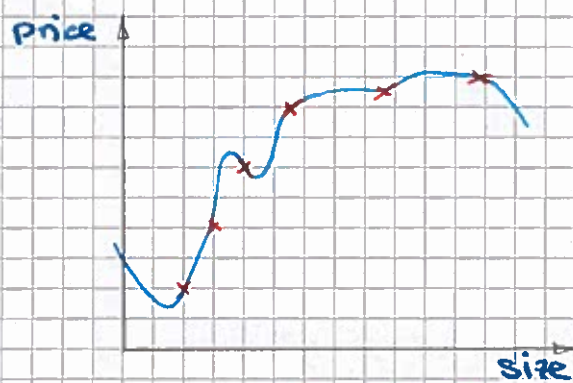
- learn parameter θ from training data
- Compute test set error

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

- misclassification error (0/1 misclassification error)

$$\text{err}(h_{\theta}(x), y) = \begin{cases} 1 & \text{if } h_{\theta}(x) \geq 0.5, y = 0 \\ & \text{or if } h_{\theta}(x) < 0.5, y = 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{error}$$

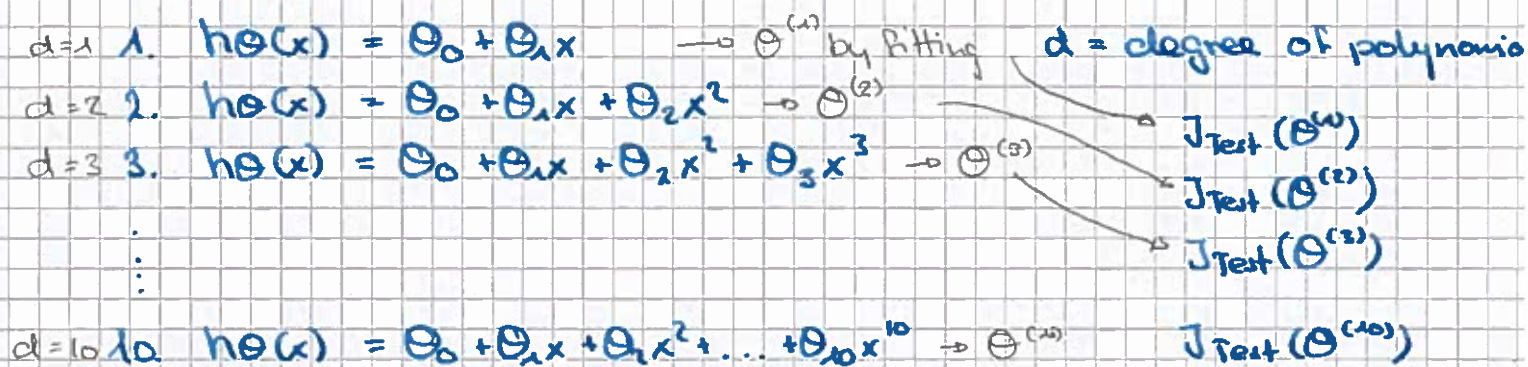
$$\text{Test error} = \frac{1}{m_{\text{test}}} \sum_{i=1}^{m_{\text{test}}} \text{err}(h_{\theta}(x_{\text{test}}^{(i)}), y_{\text{test}}^{(i)})$$



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

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

Model selection:



Choose (e.g.) $\theta_0 + \dots + \theta_5 x^5$

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

Problem:

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

Fitting $\theta_0, \theta_1, \theta_2, \dots$ does not give a decent answer how good a prediction is. \rightarrow How good it is generalizing

Evaluating your test set

Data set:

Size	price	
2104	400	} Training set
1600	330	
60% 2400	365	
1416	232	
3000	540	
1985	300	} Cross validation set (cv)
20% 1539	315	
1427	199	
20% 1380	212	} test set
1494	243	

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

$$(x^{(2)}, y^{(2)})$$

$$\vdots$$

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

$$(x_{cv}^{(1)}, y_{cv}^{(1)}) \quad m_{cv} = \# \text{ of}$$

$$(x_{cv}^{(2)}, y_{cv}^{(2)}) \quad \text{cross val. example}$$

$$\vdots$$

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

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

$$(x_{test}^{(2)}, y_{test}^{(2)}) \quad m_{test} = \# \text{ of test-example.}$$

$$\vdots$$

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

Train / Validation / test error:

Training error:

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

Cross validation error:

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

Test error:

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

Model selection:

1. $h_{\theta}(x) = \theta_0 + \theta_1 x \rightarrow \min_{\theta} J(\theta) \rightarrow \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)})$
2. $h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \rightarrow \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)})$
3. $h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 \rightarrow \theta^{(3)} \rightarrow J_{cv}(\theta^{(3)})$
- \vdots
- $\theta^{(4)} \rightarrow J_{cv}(\theta^{(4)})$
best fit
10. $h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_{10} x^{10} \rightarrow \theta^{(10)} \rightarrow J_{cv}(\theta^{(10)})$

Pick $\theta_0 + \theta_1 x + \dots + \theta_4 x^4$

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

Diagnosing bias vs. variance

4

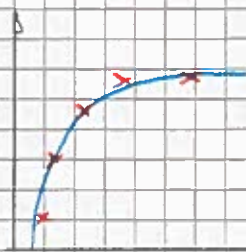
Bias / Variance



$$\theta_0 + \theta_1 x$$

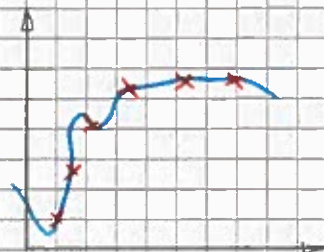
high bias
(underfit)

$$d=1$$



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

Just right
 $d=2$



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

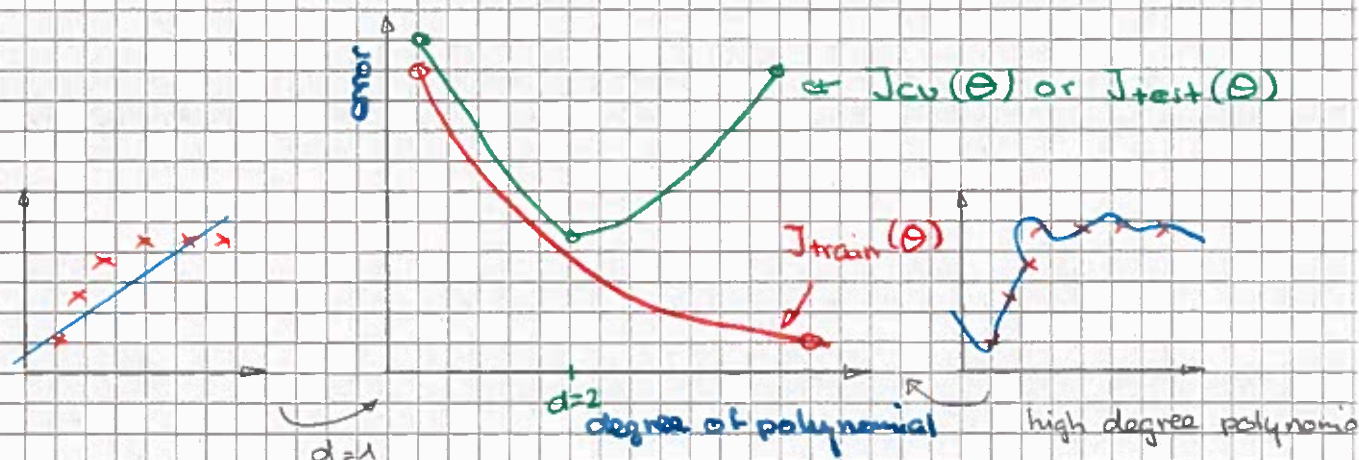
High variance
(overfit)

$$d=4$$

Bias / Variance

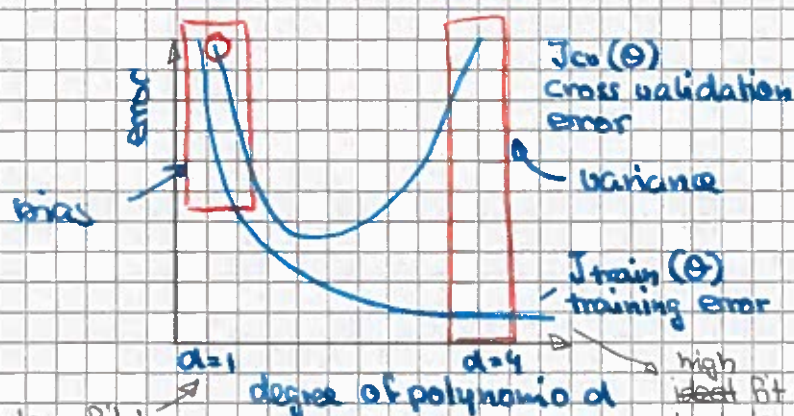
Training error: $J_{\text{train}}(\theta) = \frac{1}{2m} \cdot \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$

Cross validation error: $J_{\text{cv}}(\theta) = \frac{1}{2m_{\text{cv}}} \cdot \sum_{i=1}^{m_{\text{cv}}} (h_{\theta}(x_{\text{cv}}^{(i)}) - y_{\text{cv}}^{(i)})^2$ or $J_{\text{test}}(\theta)$



Diagnosing bias vs. variance

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



Bias (underfit)

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

Variance (overfit)

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

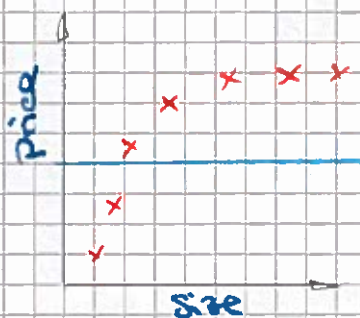
low fit to training data

high fit to training data

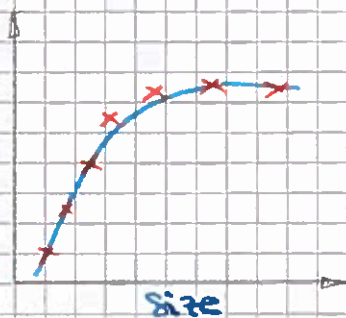
Linear Regression with regularization:

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

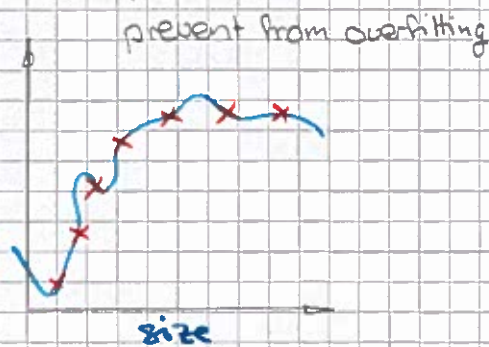
$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \underbrace{\frac{\lambda}{2m} \sum_{j=1}^m \theta_j^2}_{\text{prevent from overfitting}}$$



Large λ
High bias (underfit)



intermediate λ
"Just right"



Small λ
High variance (overfit)

$\lambda = 10,000$ $\theta_1 \approx 0, \theta_2 \approx 0$

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

penalized

$\lambda = 0 \Rightarrow$ no regularization

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

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

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

J_{train} average
 J_{cv} squared
 J_{test} error

without regularization term

1. try $\lambda = 0 \rightarrow \min_{\theta} J(\theta) \rightarrow \theta^{(1)} \rightarrow J_{\text{cv}}(\theta^{(1)})$

2. $\lambda = 0.01 \rightarrow \theta^{(2)} \rightarrow J_{\text{cv}}(\theta^{(2)})$

3. $\lambda = 0.02 \rightarrow \theta^{(3)} \vdots$

4. $\lambda = 0.04$

5. $\lambda = 0.07$

\vdots

10. $\lambda = 10.24$

pick the one with lowest error e.g. $\theta^{(5)} \rightarrow J_{\text{cv}}(\theta^{(5)})$

$\rightarrow \theta^{(10)} \rightarrow J_{\text{cv}}(\theta^{(10)})$

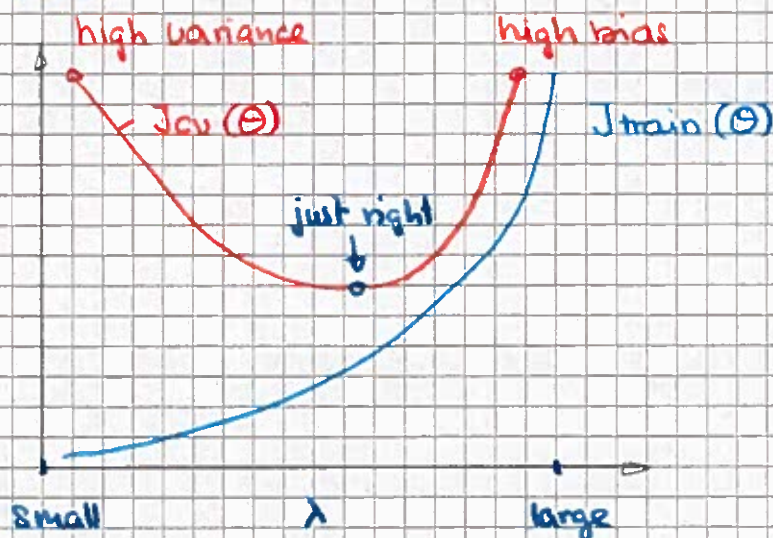
Pick left say $\theta^{(5)}$ Test error: $J_{\text{test}}(\theta^{(5)})$

use it on Test set

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

$$J_{\text{train}}(\theta) = \frac{1}{2m_{\text{train}}} \sum_{i=1}^{m_{\text{train}}} (h(x_{\text{train}}^{(i)}) - y_{\text{train}}^{(i)})^2$$

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



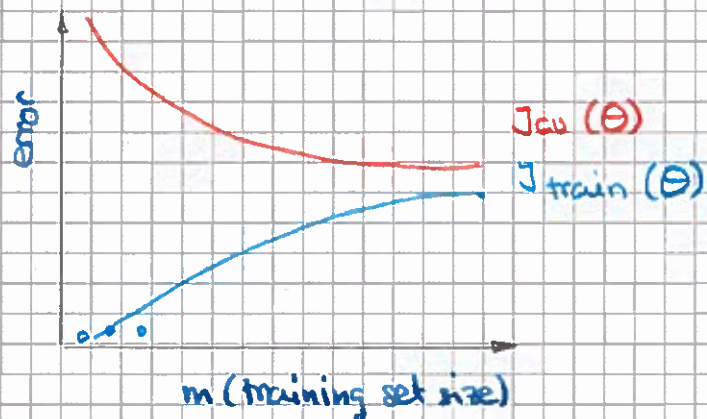
Learning Curves

10

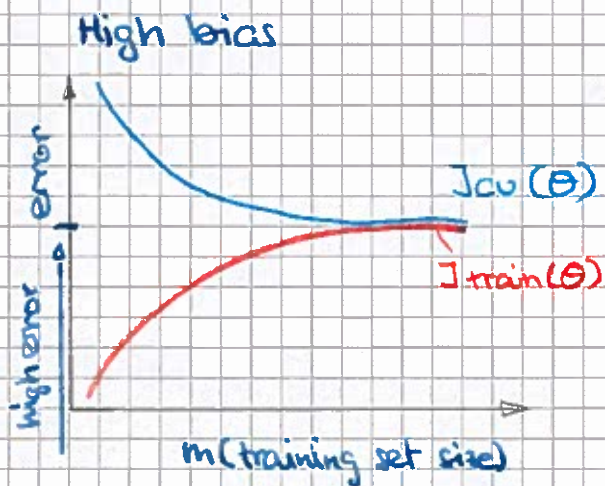
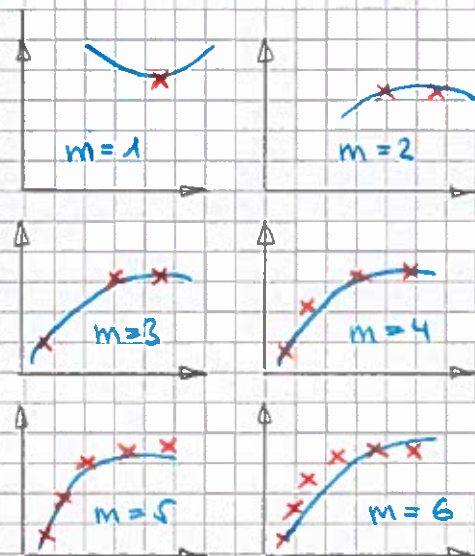
$$J_{\text{train}}(\theta) = \frac{1}{2m} \sum_{i=1}^m (h\theta(x^{(i)}) - y^{(i)})^2$$

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

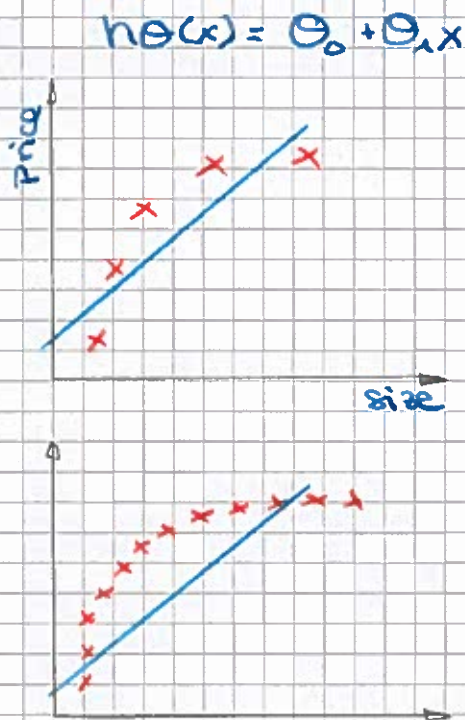
$$h\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$



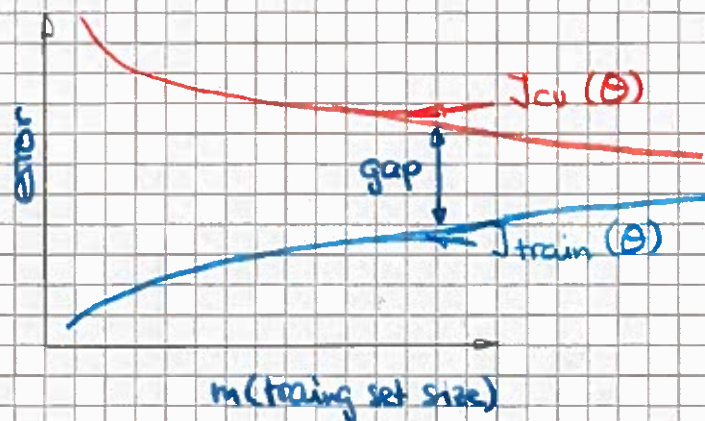
Small amount of training sets
 J_{cv} does not generalize well
 only when training set gets larger



if a learning algorithm is suffering from high bias, getting more data will not help much.

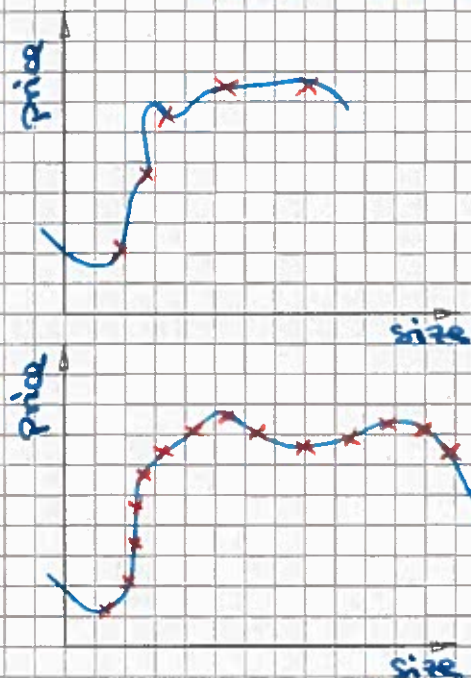


High variance



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

$$h(\theta(x)) = \theta_0 + \theta_1 x + \dots + \theta_{100} x^{100} \quad (\text{and small } \lambda)$$



Debugging a learning algorithm

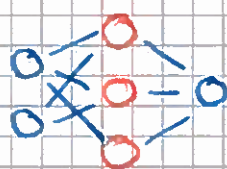
Suppose you have implemented regularized linear regression to predict housing prices. However when you test your hypothesis in a new set of houses, you find that it makes unacceptable large errors in its prediction. What should you do?

- get more training data → fix high variance
- try smaller sets of features → fix high variance
- try getting additional features → fixing high bias problem
- try add polynomial features → fix high bias
- decrease λ → fixes high bias
- increase λ → fixes high variance

Neural networks and overfitting

Small neural networks

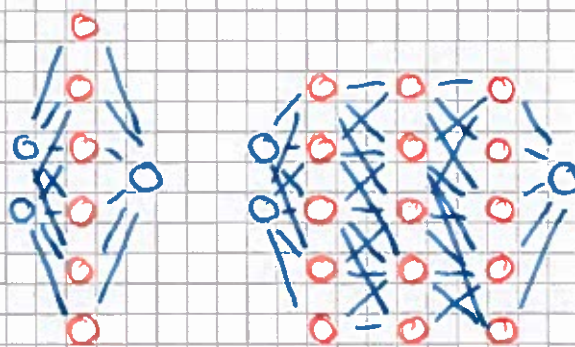
(fewer params, more prone to underfitting)



Computationally cheaper

Large neural network

(more params, more prone to overfitting)



Computationally more expensive

use regularization to address overfitting



$J_{\text{cv}}(\theta)$

using multiple hidden layer is better even if there is the chance to overfit, but then play with lambda to reduce overfitting.