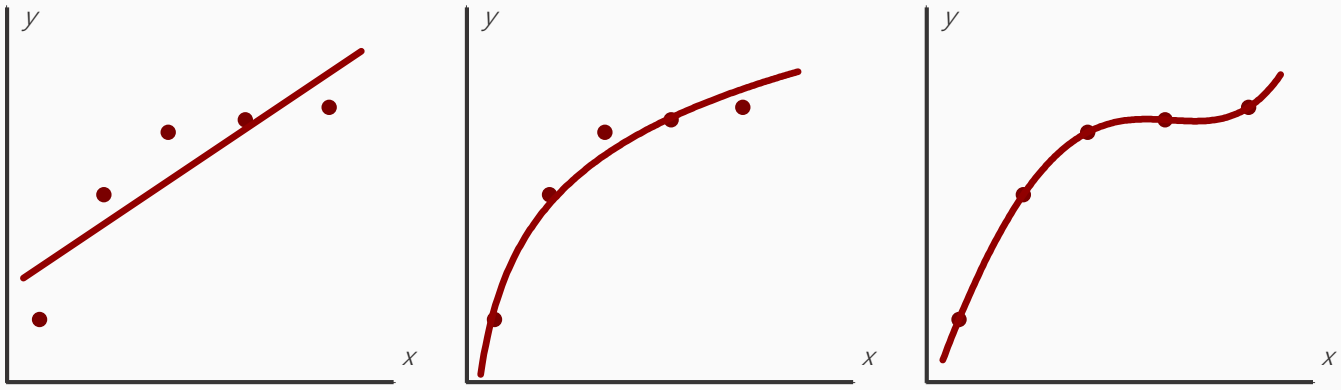
diagnosing bias vs variance

diagnosing bias vs variance

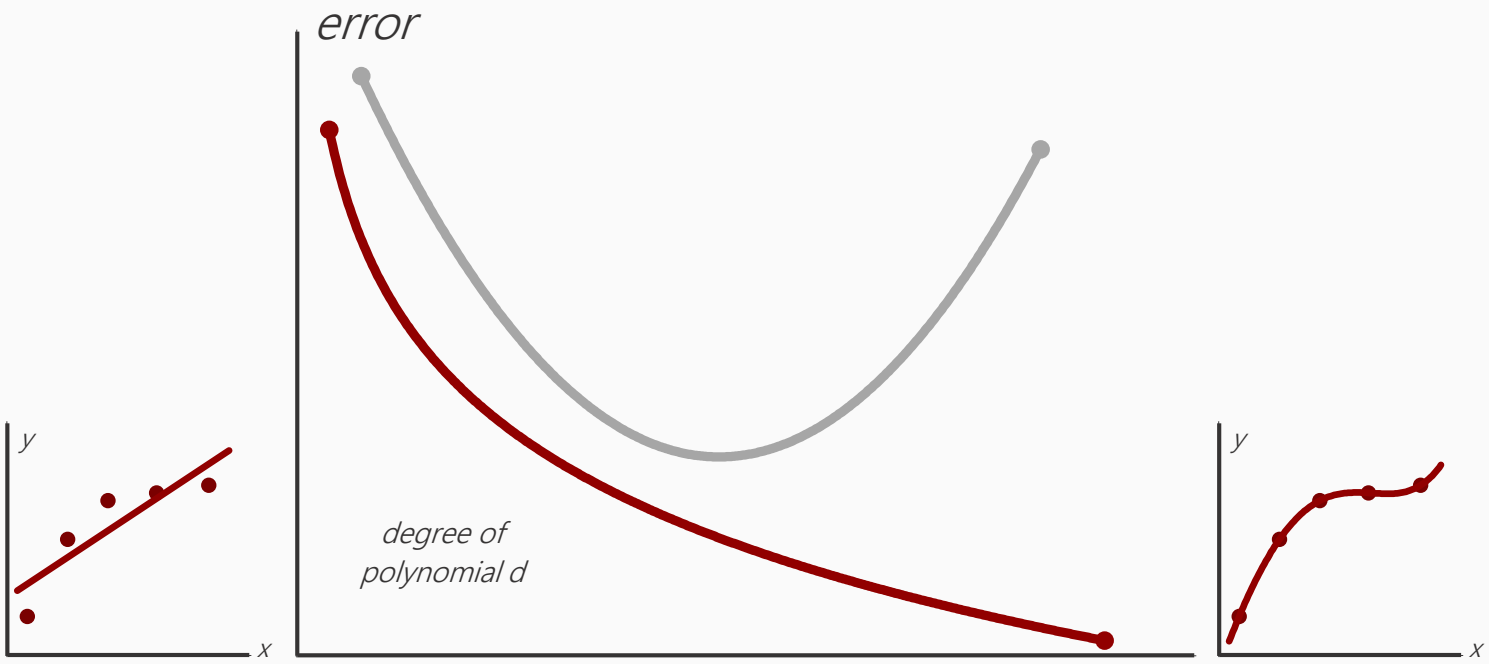


$\theta_0 + \theta_1 x$
high bias
(underfit) d=1

$\theta_0 + \theta_1 x + \theta_2 x^2$
"just right"
(fitted) d=2

$\theta_0, \theta_1 x, \ldots, \theta_4 x^4$
high variance
(overfit) d=4

the relationship between error and the degree of polynomial used on the training and cross validation and test datasets illustrates where the errors depart correlation



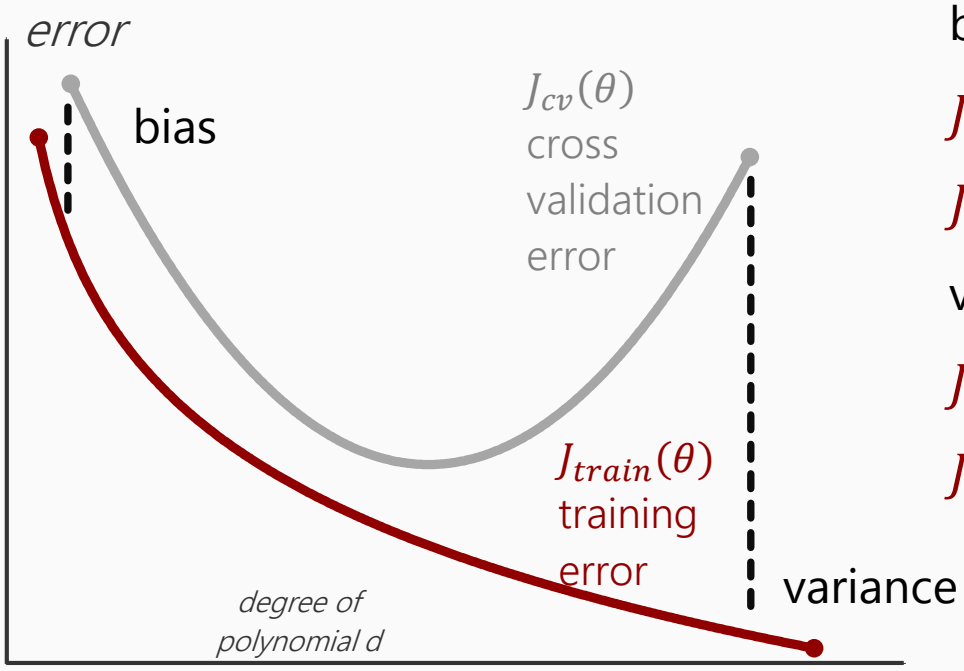
$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right)^2$$

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

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

general expectations where bias and variance are experienced

when a learning algorithm is underperforming ($J_{cv}(\theta)$ or $J_{test}(\theta)$ is high): the diagnosis could be either a bias or variance problem.



bias (underfitted):

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

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

variance (overfitted):

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

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

example

in a classification problem: the (misclassification) error is defined as $\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \text{err}\left(h_\theta\left(x_{test}^{(i)}\right), y_{test}^{(i)}\right)$, and the cross validation (misclassification) error is similarly defined, using cross validation examples $\left(x_{cv}^{(1)}, y_{cv}^{(1)}\right), \ldots, \left(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})}\right)$. the training error is 0.10 and the cross validation error is 0.30. The problem the algorithm is likely suffering from:

high variance (overfitting)

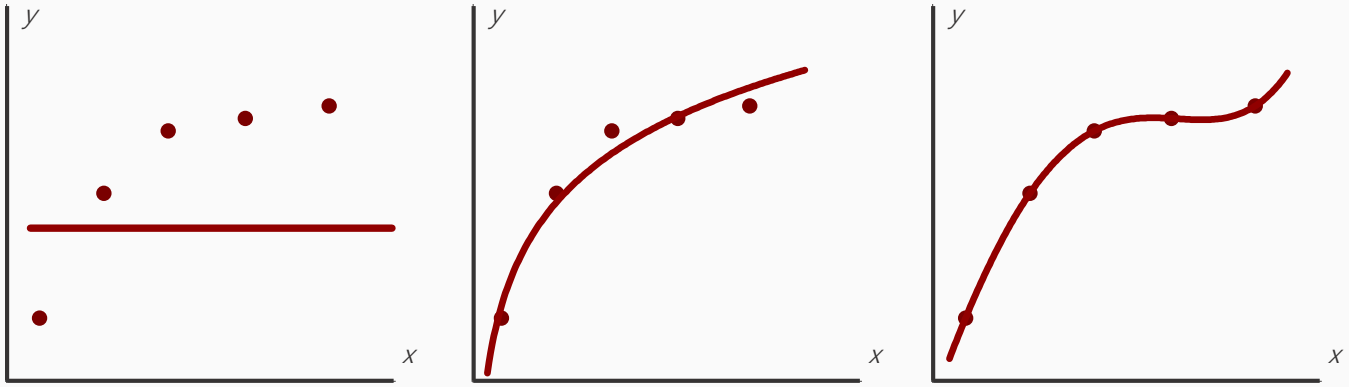
regularization and bias/variance

regularization can prevent overfitting, however, it also can affect the bias and variances of a learning algorithm.

linear regression

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

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



large λ	intermediate λ	small λ
high bias (underfit)	"just right"	high variance (overfit)
λ = 10000:	λ = 0	
$\theta_1 x \approx 0, \theta_2 \approx 0, \ldots$		
$h_\theta(x) \approx \theta_0$		

choosing the regularization parameter

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

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

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

the definition of the parameter $J(\theta)$ in the context *train*, *cv*, *test*:

half the average squared error of without the regularization term

$$
\begin{aligned}
&\text{attempt } \lambda = 0 \quad \rightarrow \min_{\theta} J(\theta) \rightarrow \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)}) \\
&\text{attempt } \lambda = 0.01 \rightarrow \min_{\theta} J(\theta) \rightarrow \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)}) \\
&\text{attempt } \lambda = 0.02 \rightarrow \min_{\theta} J(\theta) \rightarrow \theta^{(3)} \rightarrow J_{cv}(\theta^{(3)}) \\
&\text{attempt } \lambda = 0.04 \quad . \quad . \quad . \\
&\text{attempt } \lambda = 0.08 \quad . \quad . \quad . \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \\
&\text{attempt } \lambda = 10 \quad \rightarrow \min_{\theta} J(\theta) \rightarrow \theta^{(10)} \rightarrow J_{cv}(\theta^{(10)})
\end{aligned}
$$

the chosen $\theta^{(m)}$ value will then be applied to the test error: $J_{test}(\theta)$

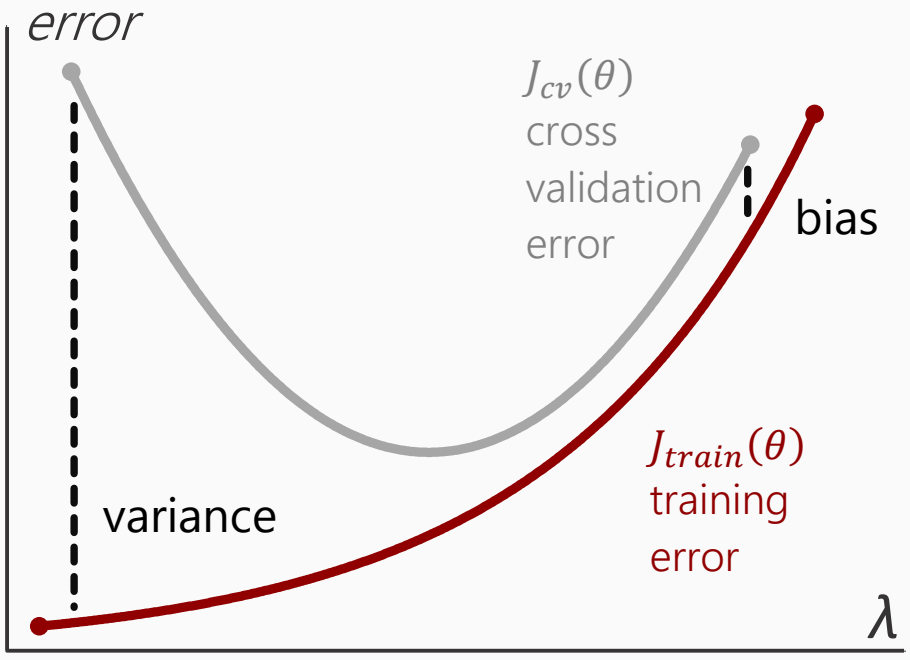
bias and variance as a function of the regularization parameter λ

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

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

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

as lambda increases in regularization, the training error will increase while the validation error decreases. After lambda has increased to an extent, the validation error regress upward



learning curves

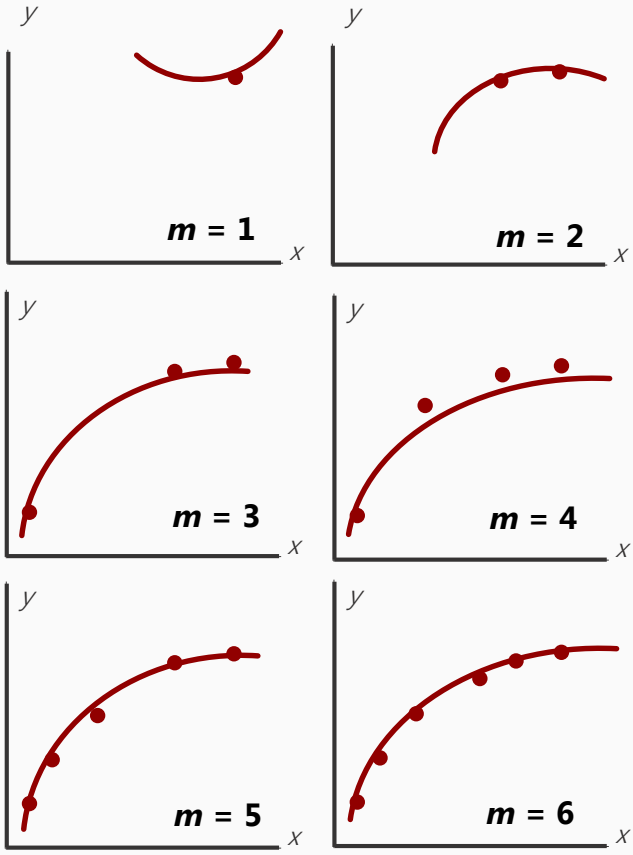
a tool to sanity check if learning algorithms are performing as expected, correctly, or identify areas of improvement from suffering from bias/variance

initially plot:

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

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

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



as a function of *m*

a single parameter will fit perfectly without regularization

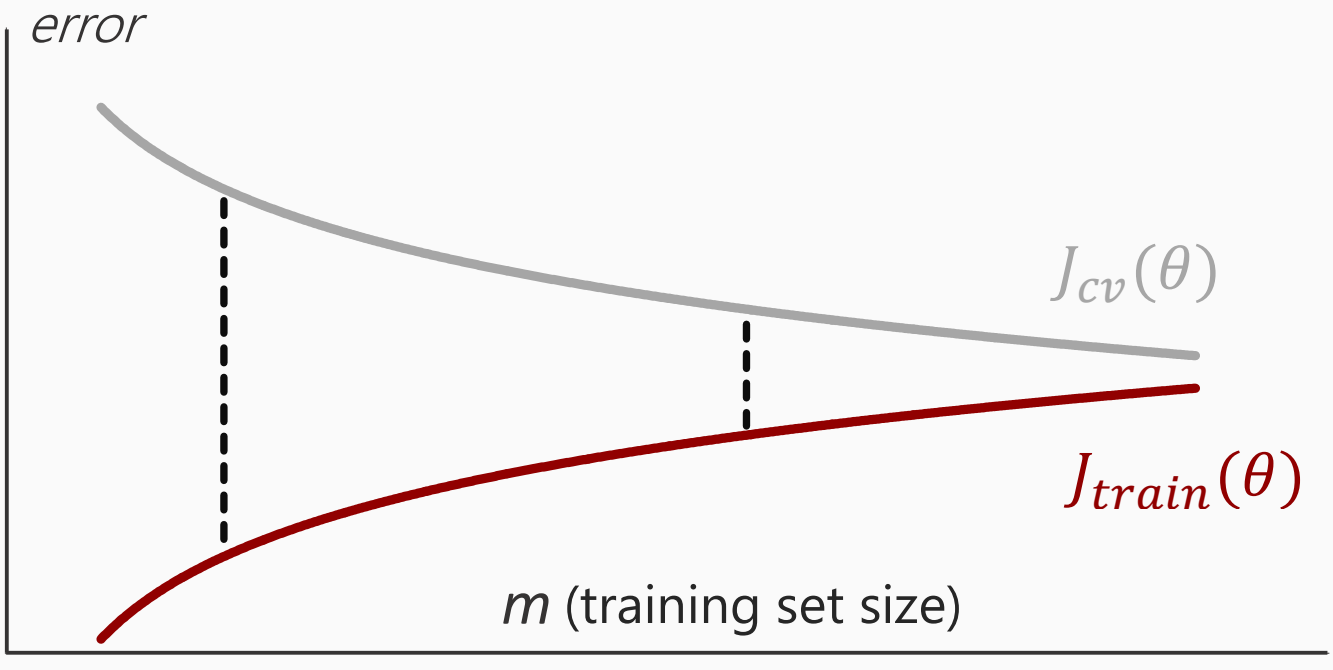
two parameters will fit almost perfectly without regularization and perfectly with regularization

three parameter behave similarly to the latter two. Thus, training error will effectively be 0 without regularization and almost 0 with regularization

four parameters will not plot perfectly and will increase training error

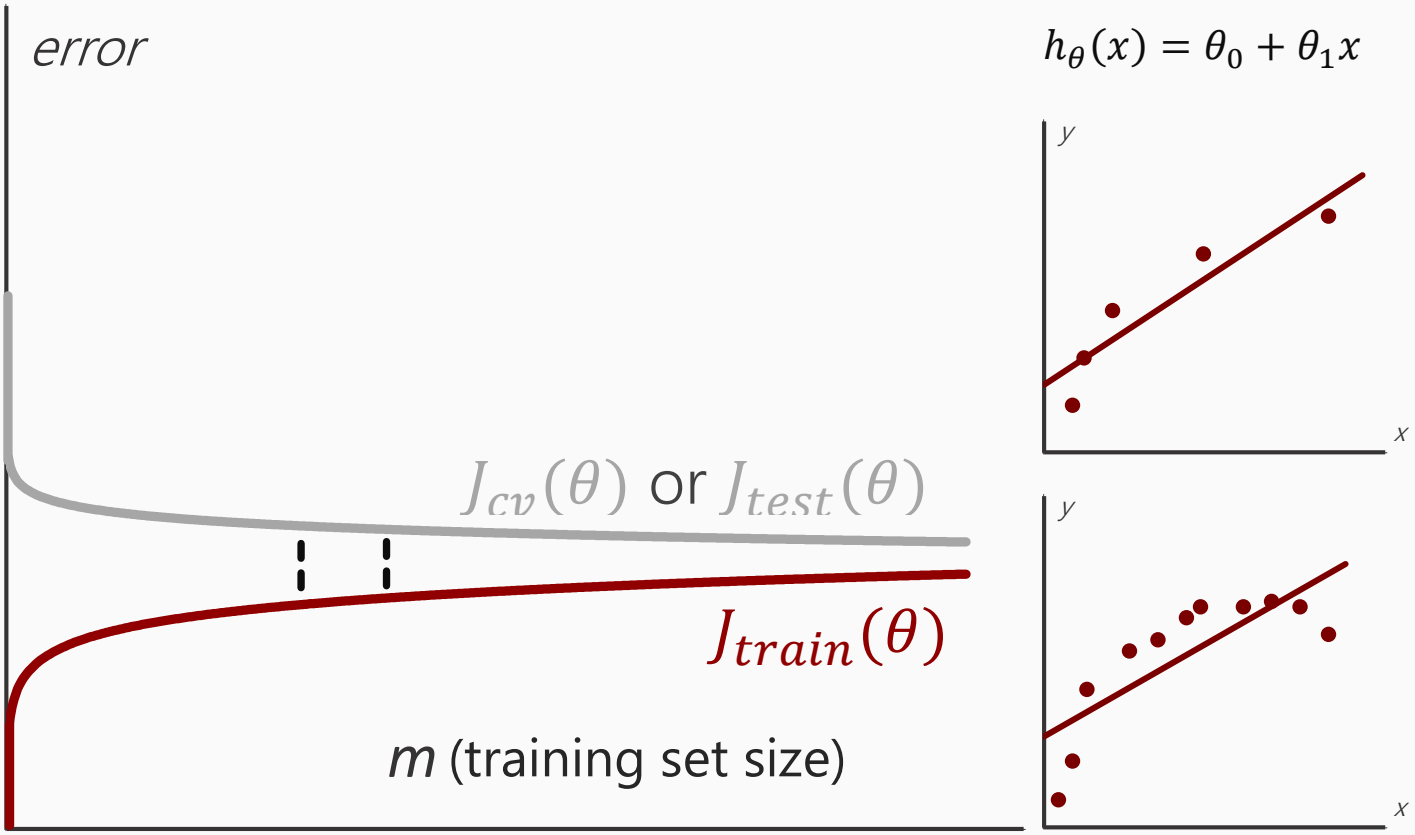
five parameters will continue to increase training

the more data, the better a model will generalize to new examples:



a problem that experiences high bias

the errors of the training and validation data will converge at a faster rate and remain level as the sample continues to grow. The error rate will reach an optimal goodness of fit after so many examples before increases in samples have little effect on the model:

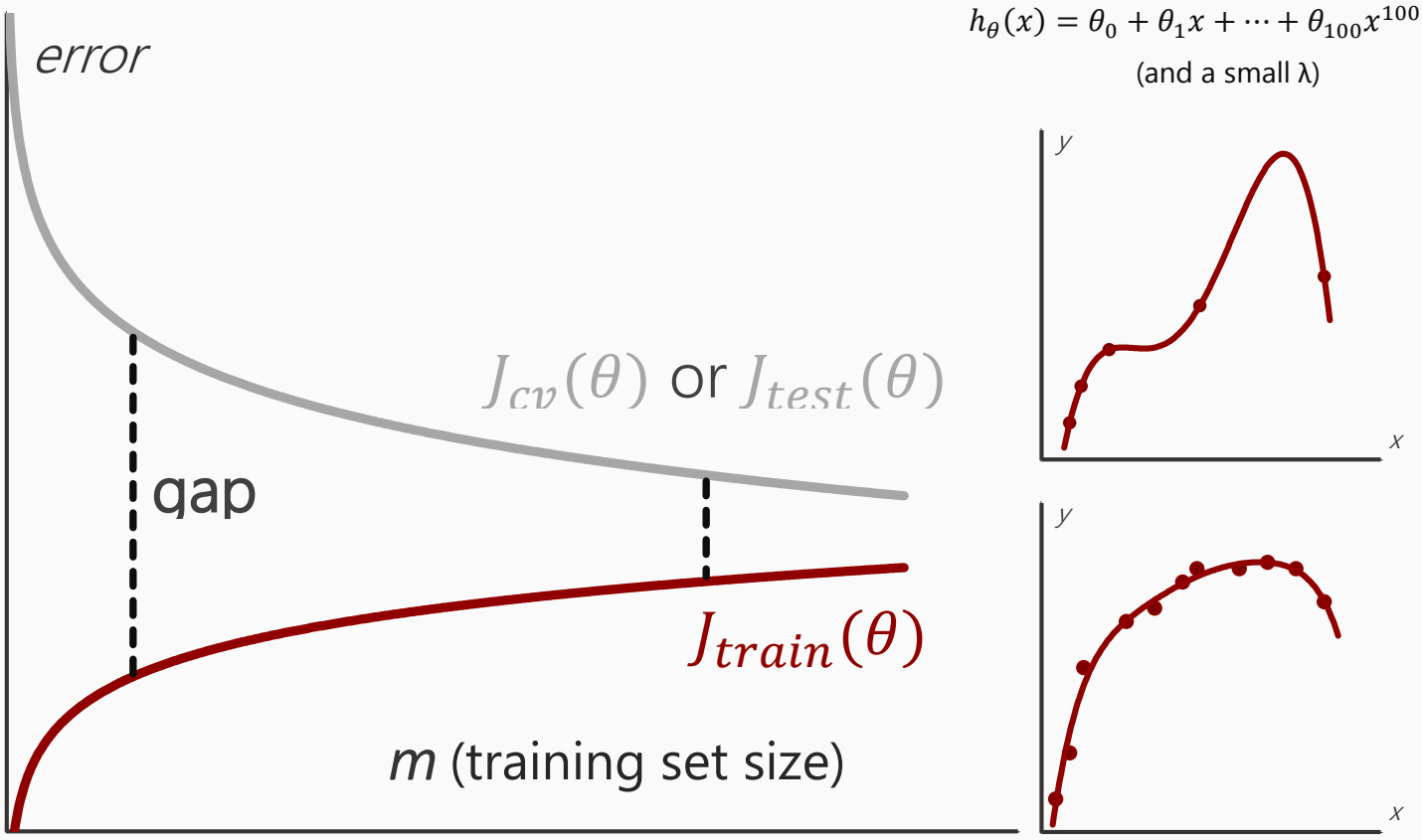


the problem with high bias reflects in the fact that both validation and training error are high, ending up with a relatively high value for both $J_{train}(\theta)$ and $J_{cv}(\theta)$. essentially the error will not increase by a large amount when the sample continues to increase thus the model will remain overfitted to the data if it was overfit in the initial expression

a learning algorithm suffering from high bias **will not (alone)** be substantially effective in reaching a more generalized fit through obtaining more training data

a problem that experiences high variance

with high variance problems, $J_{train}(\theta)$ error will initially be low and remain relatively low. reciprocally, the $J_{cv}(\theta)$ error will initially be high and remain as such. The **gap** between the two errors is indicative of high variance. extrapolating the sample to the right indicates the conditions of obtaining more samples is likely to help with training



a learning algorithm suffering from high variance **will likely** be substantially effective in reaching a more generalized fit through obtaining more training data

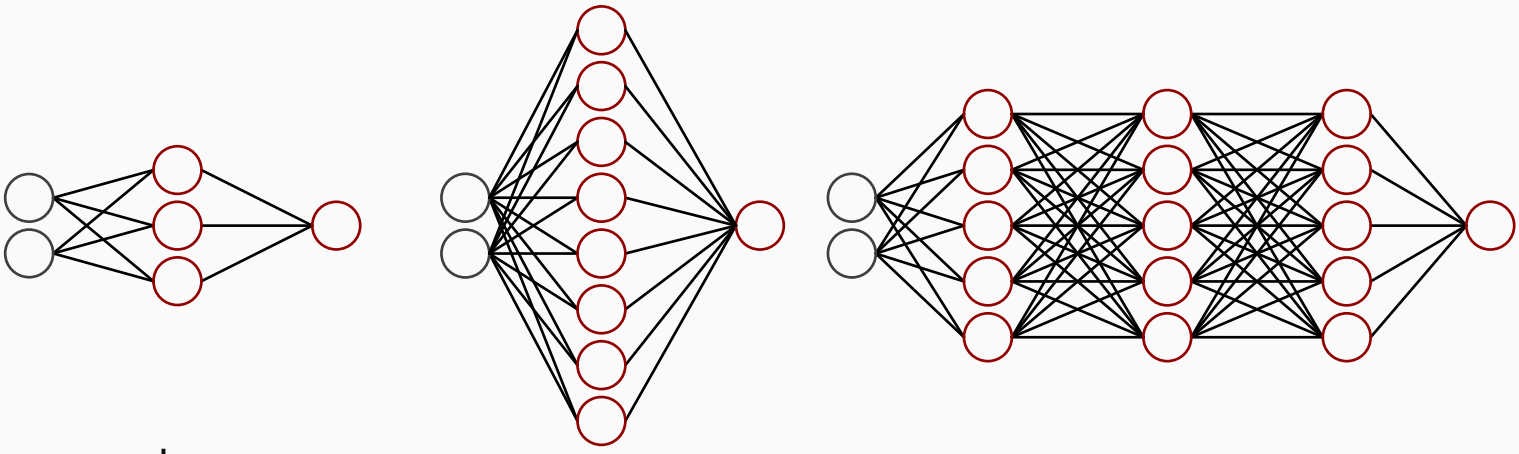
an algorithm suffering from high variance will be synonymous to the $J_{cv}(\theta)$ (cross validation error) being **>>** (much larger) than the $J_{train}(\theta)$ (training error)

debugging a learning algorithm

after implementing a regularized linear regression algorithm to predict some target variable a unacceptably large amount of errors occur in a prediction of *new* data:

obtain more training examples	→ addresses high variance (not bias)
use smaller sets of features	→ addresses high variance (not bias)
obtain additional features	→ addresses high bias
attempt adding polynomial features	→ addresses high bias
attempt decreasing or increasing λ	→ decrease in bias/increase in variance

neural networks and overfitting



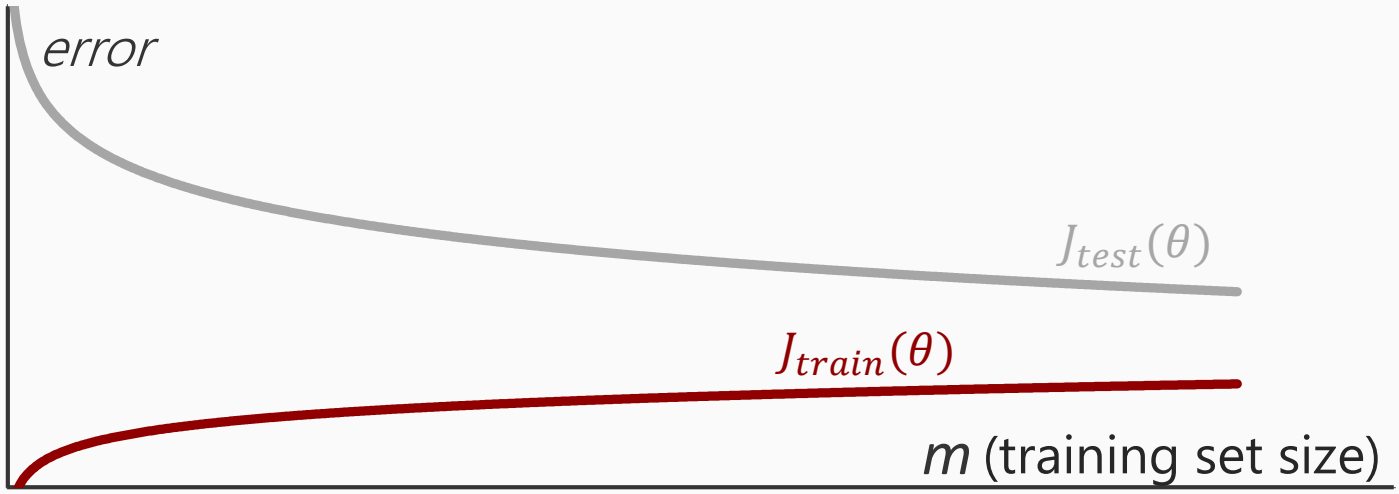
example

when fitting a neural network with one hidden layer to a training set, the cross validation error $J_{cv}(\theta)$ is much larger than the training error $J_{train}(\theta)$. Increasing the number of hidden units is likely to:

not help because the model suffers from high variance. adding hidden units to a model with a high variance problem is not effective.

advice for applying machine learning

while training a learning algorithm, it has unacceptably high error on the test set. After plotting the learning curve, and obtaining the figure below, the algorithm is suffering from high variance



given the following hypotheticals:

after implementing regularized logistic regression to classify what object is in an image (i.e., to do object recognition), the tested hypothesis on a new set of images makes unacceptably large errors with its predictions on the new images. However, the hypothesis performs **well** (has low error) on the training set. methods to address:

obtain more training examples	→ the gap in errors between the training and test suggest high variance problem where the algorithm has overfit the training set. addresses high variance. adding more training data increases the complexity of the training set and help with variance
use smaller sets of features	→ reducing the feature set will ameliorate the overfitting and help with the variance problem
attempt increasing λ	→ increasing the regularization parameter will reduce overfitting in a high variance problem

after implementing regularized logistic regression to predict what items customers will purchase on a web shopping site, the tested hypothesis on a new set of customers makes unacceptably large errors in predictions. Furthermore, the hypothesis performs **poorly** on the training set. the following addresses the issue:

obtain additional features	→ poor performance on both training and test sets indicate a high bias problem. additional features increase complexity of the hypothesis, improving the fit to both the training and test data under high bias
attempt adding polynomial features	→ adding more complex features increases the complexity of the hypothesis and improves the fit to both training and test data under high bias
attempt decreasing λ	→ decreasing the regularization parameter will reduce overfitting in a high bias problem

given the following hypotheticals:

suppose you are training a regularized linear regression model. the recommended way to choose what value of regularization parameter λ to use is to choose the value of λ which gives the lowest **training set** error.

the training error should not be used to choose the regularization parameter. the training error can always improve through using less regularization (smaller value of λ). however, too small of a value will not generalize well on the test dataset.

suppose you are training a regularized linear regression model. the recommended way to choose what value of regularization parameter λ to use is to choose the value of λ which gives the lowest **cross validation** error.

the cross validation allows for finding the “just right” setting of the regularization parameter given the fixed model.

the performance of a learning algorithm on the training set will typically be better than its performance on the test set.

suppose you are training a regularized linear regression model. the recommended way to choose what value of regularization parameter λ to use is to choose the value of λ which gives the lowest **test set** error.

the test set should not be used to choose the regularization parameter, considering the model will have an artificially low value for test error and it will not provide good estimate of generalization error.

when debugging learning algorithms, it is useful to plot a learning curve to understand if there is a high bias or high variance problem. the shape of the learning curve is indicative of bias or variance within the learning algorithm.

a model with more parameters is more prone to overfitting and typically has higher variance. more model parameters increases the model’s complexity, tightly fitting it to the training data and increasing the chances of overfitting.

if a learning algorithm is suffering from high bias, only adding more training examples may **not** improve the test error significantly.

if a neural network has much lower training error than test error, then adding more layers will further increase the variance problem through an increased complexity in the model. the variance problem will become worse than before.