Evaluating the performance of a model and hypertuning:
Methods to improve the performance of a model on a new set of data input.

- ➔ Get more training examples
  - o However this may not help very much in many cases and leads to a lot of time wasted
- ➔ Try a smaller set of features
- ➔ Try getting additional features
- ➔ Try adding polynomial features
- ➔ Increase λ
- ➔ Decrease λ

Hence how should we decide which of these diagnoses we should consider pursuing first?
This is where Machine Learning Diagnostics come into play, as tests to gain insight into what is/is not working with a learnt model and gain direction on how to improve it's performance.

Evaluating the Hypothesis:

Overfitting the dataset?
Soln: Split the raw dataset into a train-test split
70% train, 30%

Train-Test procedure for Linear Regression/ any other models:

1) Learn Params from Train split of the data.
2) Compute test set error on cost function.

$$Eg: for\ linear\ regression: J_{test}(\theta) = \frac{1}{2m}\sum_{i=1}^{m_{test}} (h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

For Classification Problems:
Another option is misclassification error:

$$err\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right) = \begin{cases} 1\ if\ h_\theta\left(x^{(i)}\right) \geq threshold\ ; y = 0\ or\ h_\theta\left(x^{(i)}\right) < threshold; y = 1 \\ 0\ otherwise \end{cases}$$

$$Test\ error = \frac{1}{m_{test}}\sum_{i=1}^{m_{test}} err\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right)$$

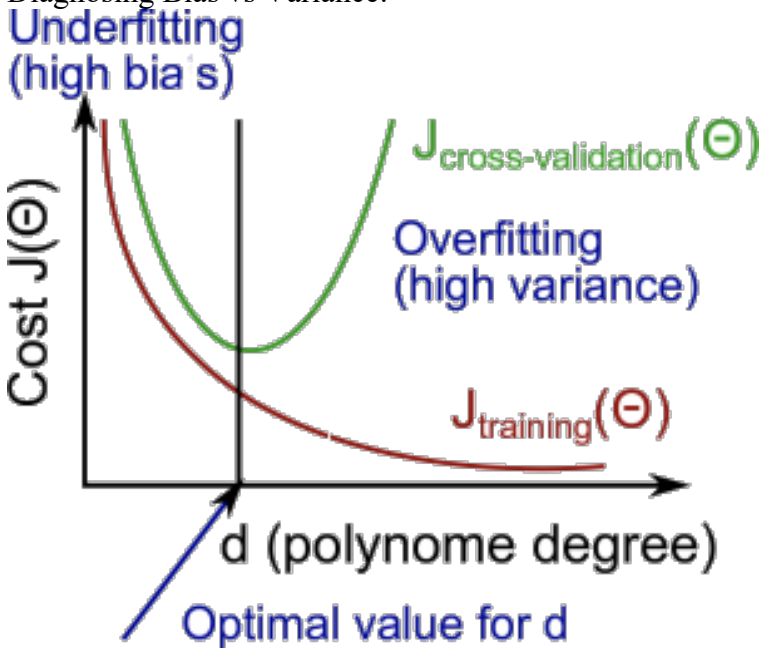
A percentage of misclassified images

Model Selection:
Once params θ_0, ..., θ_n are fit to a training set the error of params as measured on that set of data is likely to be lower than the actual generalization error, as the params are optimized to that particular set of data.

Hence we further split the data into:
Train, Test, Cross Validation(CV) sets at a proportion of 60%,20%,20% after random permutation of the dataset.

Then we iterate through degrees of polynomials on the CV set in order to select a best fit polynomial for the parameters on the CV set. The data would then have a bias on the minimization of the CV set. However, not on the test set which measures the ability of the model to predict on new data.
This helps resolve the performance bias we may face on the test dataset, and hence obtain a more generalizable metric of error.

Diagnosing Bias vs Variance:



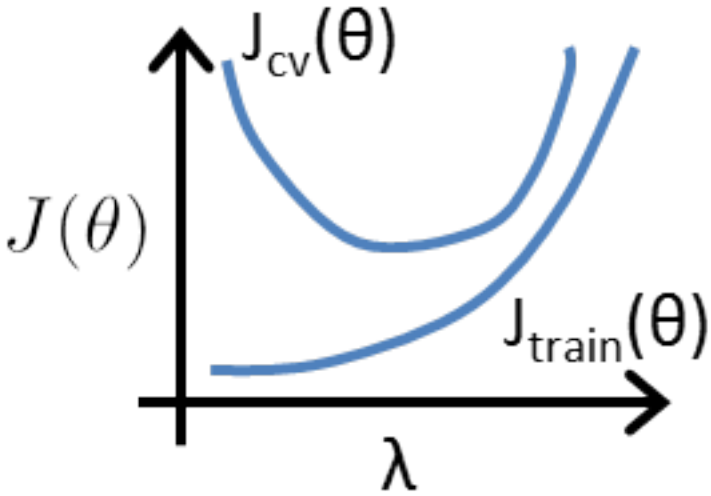
Bias (Underfit):

$$J_{train}(\theta)\ is\ high$$
$$J_{CV}(\theta) \approx J_{train}(\theta)$$

Variance(Overfit):

$$J_{train}(\theta)\ is\ low$$
$$J_{CV}(\theta) \gg J_{train}(\theta)$$

Regularization:



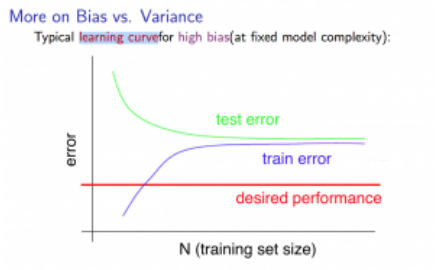
If $\lambda\ too\ small\ J_{CV}(\theta)\ will\ be\ large\ and\ J_{train}(\theta)\ is\ small\ due\ to\ overfitting$

If $\lambda\ too\ large\ J_{CV}(\theta)\ will\ be\ large\ and\ J_{train}(\theta)\ is\ large\ due\ to\ underfitting$

Hence how do we solve this?
Workflow:

1. Create a list of lambdas (i.e. $\lambda \in \{0,0.01,0.02,0.04,0.08,0.16,0.32,0.64,1.28,2.56,5.12,10.24\}$);
2. Create a set of models with different degrees or any other variants.
3. Iterate through the \lambdaλs and for each \lambdaλ go through all the models to learn some Θ.
4. Compute the cross validation error using the learned Θ (computed with λ) on the $J_{CV}(\Theta)$ **without** regularization or $\lambda = 0$.
5. Select the best combo that produces the lowest error on the cross validation set.
6. Using the best combo Θ and λ, apply it on $J_{test}(\Theta)$ to see if it has a good generalization of the problem.

Learning Curves:
High Bias:



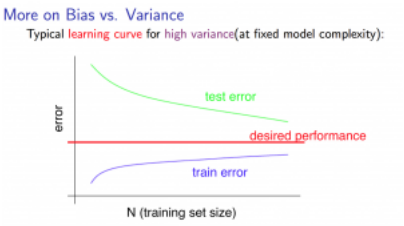
Low training set size: causes $J_{train}(\Theta)$ to be low and $J_{CV}(\Theta)$ to be high.

Large training set size: causes both $J_{train}(\Theta)$ and $J_{CV}(\Theta)$ to be high with $J_{train}(\Theta) \approx J_{CV}(\Theta)$.

Getting more training data **will not** (by itself) **help much**.

Possibly requires feature analysis, or polynomial regressions

Experiencing high variance:



Low training set size: $J_{train}(\Theta)$ will be low and $J_{CV}(\Theta)$ will be high.

Large training set size: $J_{train}(\Theta)$ increases with training set size and $J_{CV}(\Theta)$ continues to decrease without leveling off. Also, $J_{train}(\Theta) < J_{CV}(\Theta)$ but the difference between them remains significant.

More training data is likely to help.

Applied to Neural Networks:
“Small” Neural Networks (few parameter, more prone to underfitting)

“Large” Neural Networks (more prone to overfitting)

- Use λ to address overfitting

CrossValidation is a good method to select the number of hidden layers in a network.