# Machine Learning V2

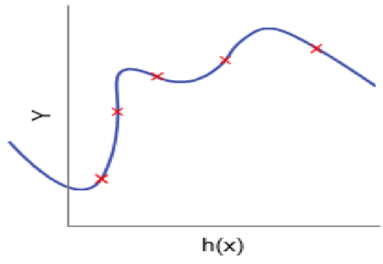
## Machine learning model evaluation

### Overfitting

After you have successfully trained your model, you need a methodology to follow to evaluate your machine learning model performance. A classic mistake is to use the same sample data that is used in training to test a model, which produces a false perfect score. This is called "overfitting" (also referred as "high variance"). The problem with overfitting is that your model fails at predicting future unseen data.

Another case that can cause overfitting is where you have unbalanced data. For example, assume that you are working on a data set for churn analysis. The customers who churned are actually 2% of your data set. Using this data set "as is" causes overfitting.

The objective of a good machine learning model is to generalize for any future data points. Overfitting also can occur if you are using too many features. Relatively, if the number of features is the same as or greater than the number of training samples, that can cause overfitting. One of the solutions to overcome overfitting is to increase the number of data set samples that is used for training compared to features. Another solution is to manually decrease the number of features, but that might result in removing useful information. Another solution is to perform model selection by using cross-validation.




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

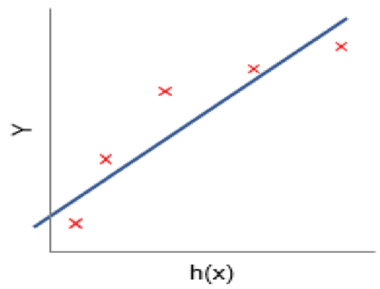
**High variance (overfit)**

### Underfitting

Underfitting (also referred to as "high bias") occurs when a machine learning model cannot fit the training data or generalize to new data.

A possible reason might be that the model is using a simple estimator. For example, you might be using a linear estimator, but what you actually need is a quadratic or higher degree polynomial estimator to develop your model. Another reason might be that you are not using enough features, so your estimator fails to capture the structure of the data. A possible solution would be to add more features and try a different estimator.

There are other methods that are used to help resolve the overfitting and underfitting of your model like regularization, but these methods are beyond the scope of this course.




$$\theta_0 + \theta_1 x$$

**High bias (underfit)**

It is common practice when applying a (supervised) machine learning task is to hold out part of the available data as a test set. There are different methods to achieve that task:

- **Cross-validation (CV)** is a process to evaluate a machine learning model by splitting a data set once or several times to train and test the model. The data set can be split into a training set to train the model and a validation set to pre-test the model. Select the model that has least error. Finally, there is a test set to evaluate the model. Thus, the data set can be split as 60% - 20% - 20% for training, validation, and testing sets.

One criticism of this process is that splitting the data set into three parts reduces the number of samples that can be used for training the model.

- **The hold-out method** partitions the data set into a majority set for training and minority set for testing. The split of the training set to test set is 80% - 20% or 70% - 30%, with no fixed rule.
- **K-fold cross validation** randomly partitions data into K equal sized subsamples. For each iteration, one subsample is kept as validation set and the rest of the subsamples (K-1) are the training set. The iterations are repeated K times, where each subsample has one chance to be the validation set. The K results can then be averaged to produce a single model. The biggest advantage of K-fold is that all data is changed to be used for both training and validation. There is no strict rule for the number K, but it is commonly K=5 or K=10, which are 5-fold cross-validation or 10-fold cross-validation. For each subsample, you maintain approximately the same percentage of data of each target class as in the complete set, which is known as the Stratified K-fold method.
- **Leave one out CV (LOO-CV)** is similar to K-fold, but in this case each one sample data point is held out as a validation set, and the rest of data set is the training set. Comparing LOO-CV and K-fold, K-fold is faster and requires less computation, but in terms of accuracy, LOO-CV often has a high variance as an estimator.