WEEK 6 : Machine Learning

Advice for Applying Machine Learning, Machine Learning System Design

Joohyung Kang

Contents

I. Advice for Applying Machine Learning

II. Machine Learning System Design

Advice for Applying Machine Learning

❖ Evaluating a Learning Algorithm

▪ Deciding What to Try Next

- **Once we have done some trouble shooting for errors in our prediction by:**
  - ✓ Get more training examples
  - ✓ Try smaller sets of features
  - ✓ Try getting additional features
  - ✓ Try adding polynomial features
  - ✓ Try decreasing λ
  - ✓ Try increasing λ

- **Machine Learning Diagnostic** ➔ 학습된 모델의 성능을 진단/파악
  - ✓ Evaluating a Hypothesis
  - ✓ Model Selection
  - ✓ Train/Validation/Test Sets

Evaluating a Hypothesis

Evaluating a Learning Algorithm

Advice for Applying Machine Learning

❖ Evaluating a Learning Algorithm

▪ Evaluating a Hypothesis

- 학습된 모델이 올바른지 평가하는 방법
- 주어진 Training Set을 Train/Test Sets으로 분리
  - ✓ 70% → Training Set
  - ✓ 30% → Testing Set

※ Training Set과 Test Set은 Random하게 추출

Dataset:

	Size	Price	
70%	2104	400	Training Set
	1600	330	
	2400	369	
	1416	232	
	3000	540	
	1985	300	
	1534	315	
30%	1427	199	Testing Set
	1380	212	
	1494	243	

Advice for Applying Machine Learning

❖ **Evaluating a Learning Algorithm**

- **Training/Testing procedure for Linear Regression**
  ① Lean parameter from training data (70%)
  ② Compute test set error

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

Advice for Applying Machine Learning

❖ Evaluating a Learning Algorithm

- **Training/Testing procedure for Logistic Regression**
  ① Lean parameter from training data (70%)
  ② Compute test set error

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

- Misclassification error:

$$err\left(h_\theta(x), y\right) = \begin{cases} 1 & \text{if } h_\theta(x) \geq 0.5 \text{ and } y = 0 \\ & \text{if } h_\theta(x) < 0.5 \text{ and } y = 1 \\ 0 & \text{Otherwise} \end{cases}$$

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

Model Selection & Train/Validation/Test Sets

Evaluating a Learning Algorithm

Advice for Applying Machine Learning

❖ Evaluating a Learning Algorithm

▪ Model Selection

- Hypothesis ➔ Polynomial Model
- d = degree of polynomial

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

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

$$d = 3 \qquad h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 \qquad \theta^{(3)} \longrightarrow J_{test}\ (\theta^{(3)})$$

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

$$d = 5 \qquad h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 \qquad \theta^{(5)} \longrightarrow J_{test}\ (\theta^{(5)})$$

$$\vdots$$

$$d = 10 \qquad h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \quad \cdots \quad + \theta_{10} x^{10} \qquad \theta^{(10)} \longrightarrow J_{test}\ (\theta^{(10)})$$

$$\therefore \min J_{test}\ (\theta^{(d)})$$

Advice for Applying Machine Learning

❖ Evaluating a Learning Algorithm

▪ Model Selection

- $\min J_{test}\ (\theta^{(d)})$

※ **Problem:** ***<u>Optimistic estimate of generalization error</u>***

✓ Test Set → Low error, New Data Set → ??

Advice for Applying Machine Learning

❖ Evaluating a Learning Algorithm

▪ Train/Validation/Test Sets

- Training Set을 3그룹으로 나누어 학습, 검증, 최종 테스트 과정을 거침

※ Data Sets

- ✓ 60% ➔ Training Set
- ✓ 20% ➔ Cross Validation Set
- ✓ 20% ➔ Testing Set

Training Error:

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

Cross Validation Error:

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

Testing Error:

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

Advice for Applying Machine Learning

❖ Evaluating a Learning Algorithm

▪ Train/Validation/Test Sets

① Optimize the parameter in θ using the training set for each polynomial degree
② Find the polynomial degree d with the least error using the ***cross validation set***
③ Estimate the generation error using the ***test set*** with $J_{test}\ (\theta^{(d)})$

$$d = 1 \qquad h_\theta(x) = \theta_0 + \theta_1 x \qquad \theta^{(1)} \longrightarrow J_{cv}\,(\theta^{(1)})$$

$$d = 2 \qquad h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \qquad \theta^{(2)} \longrightarrow J_{cv}\,(\theta^{(2)})$$

$$d = 3 \qquad h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 \qquad \theta^{(3)} \longrightarrow J_{cv}\,(\theta^{(3)})$$

$$d = 4 \qquad h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 \qquad \theta^{(4)} \longrightarrow J_{cv}\,(\theta^{(4)})$$

$$d = 5 \qquad h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 \qquad \theta^{(5)} \longrightarrow J_{cv}\,(\theta^{(5)})$$

$$\vdots$$

$$d = 10 \qquad h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \cdots + \theta_{10} x^{10} \qquad \theta^{(10)} \longrightarrow J_{cv}\,(\theta^{(10)})$$

$$\min J_{cv}\,(\theta^{(d)}) \longrightarrow J_{test}\ (\theta^{(d)})$$

Diagnosing Bias vs. Variance

Bias vs. Variance

Advice for Applying Machine Learning

❖ Bias vs. Variance

▪ Diagnosing Bias vs. Variance

- **Bias**
  - ✓ 가설 $h_\theta(x)$가 실제 현상 $y(x)$ 와 얼마나 **적합한가**에 대한 척도
  - ✓ *<u>즉, 예측된 결과가 실제 True와 얼마나 떨어져 있는가?</u>*
  - ✓ 모델이 가진 한계점에서 오는 Error
  - ✓ 모델의 특성에 따라 생겨나는 Error

Ex) Linear Function

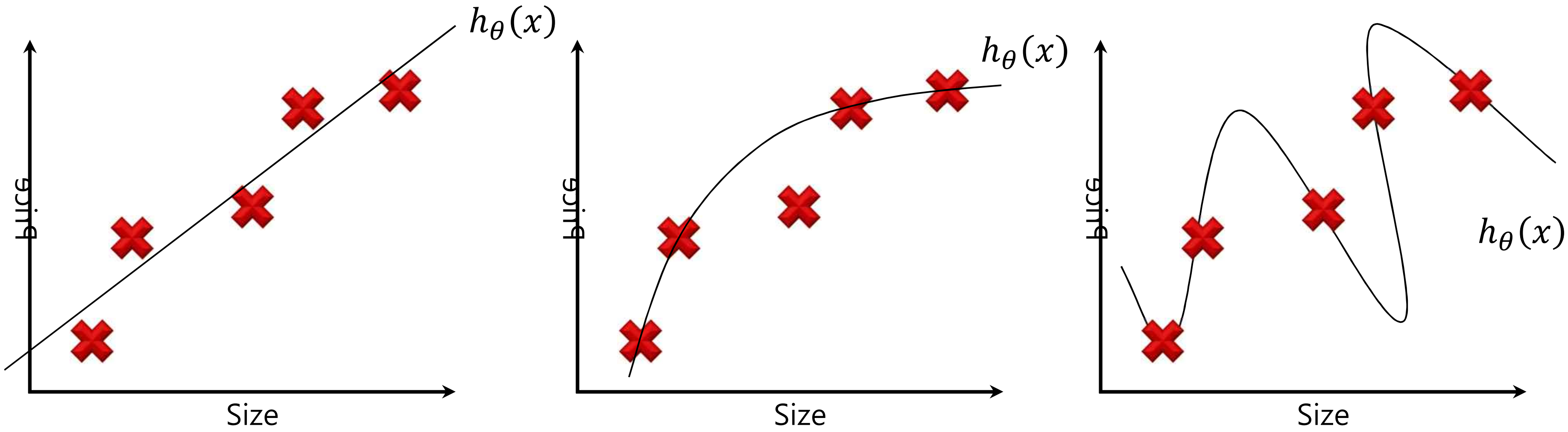
- **Variance**
  - ✓ 가설 $h_\theta(x)$의 입력 데이터에 대한 민감도
  - ✓ 특정 Data Set에만 **특화된 가설(모델)**로 인해 생겨나는 Error
    - Ex) "자동차" 인식의 문제에서 "승용차"의 Data Set으로 학습하게 되면,
    - "승합차"나 "트럭"에 대한 인식 Error가 발생함.

    ※ 즉, 새로운 입력 데이터에 대해 민감한 결과를 가져오는 현상

Advice for Applying Machine Learning

❖ The Problem of Overfitting

- **Train a hypothesis $h_\theta(x)$ for "Regression" problem**
  - Linear Regression and Polynomial Regression Problems
    - ✓ Ex. Housing prices



$\theta_0 + \theta_1 x$	$\theta_0 + \theta_1 x + \theta_2 x^2$	$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$
"Under fit" ➔ "High Bias"	**"Just Right"**	**"Over fit" ➔ "High Variance"**

Advice for Applying Machine Learning

❖ Bias vs. Variance

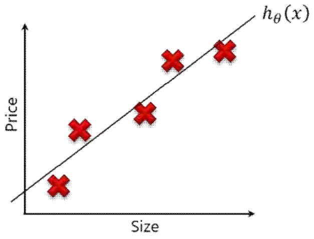
▪ Diagnosing Bias vs. Variance

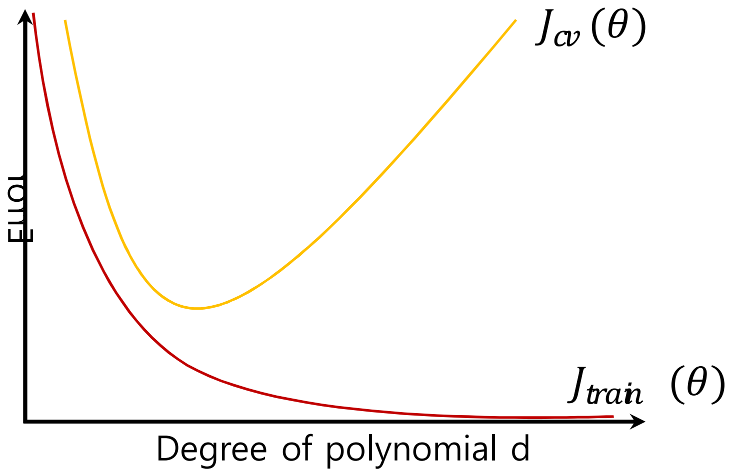
Training Error:

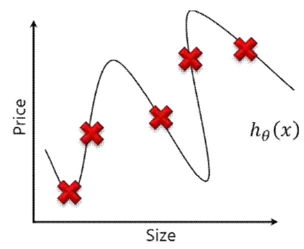
$$J_{train}(\theta) = \frac{1}{2m_{train}} \sum_{i=1}^{m_{train}} (h_\theta(x_{train}^{(i)}) - y_{train}^{(i)})^2$$

Cross Validation Error:

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

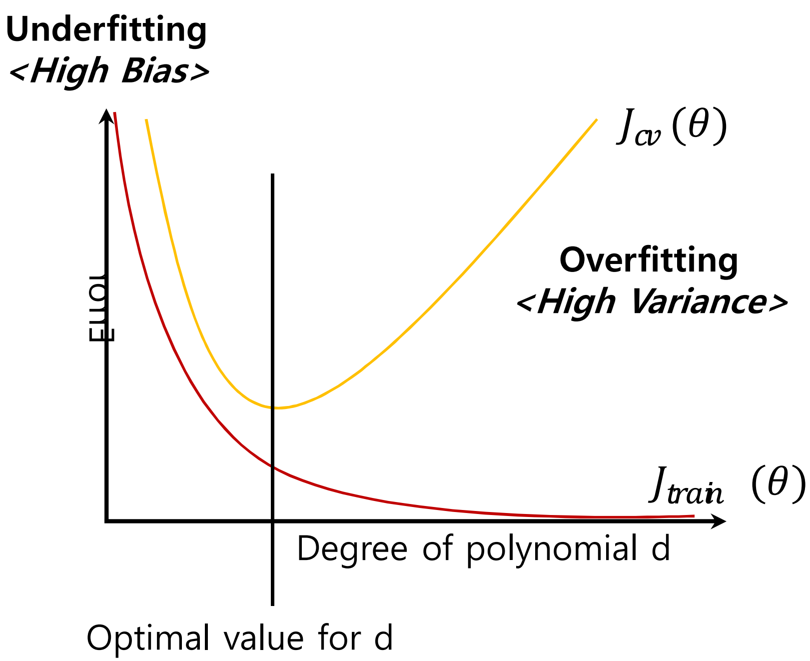






Advice for Applying Machine Learning

- **Bias vs. Variance**
  - **Diagnosing Bias vs. Variance**



Bias (Underfit):

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

Variance (Overfit):

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

Regularization and Bias/Variance

Bias vs. Variance

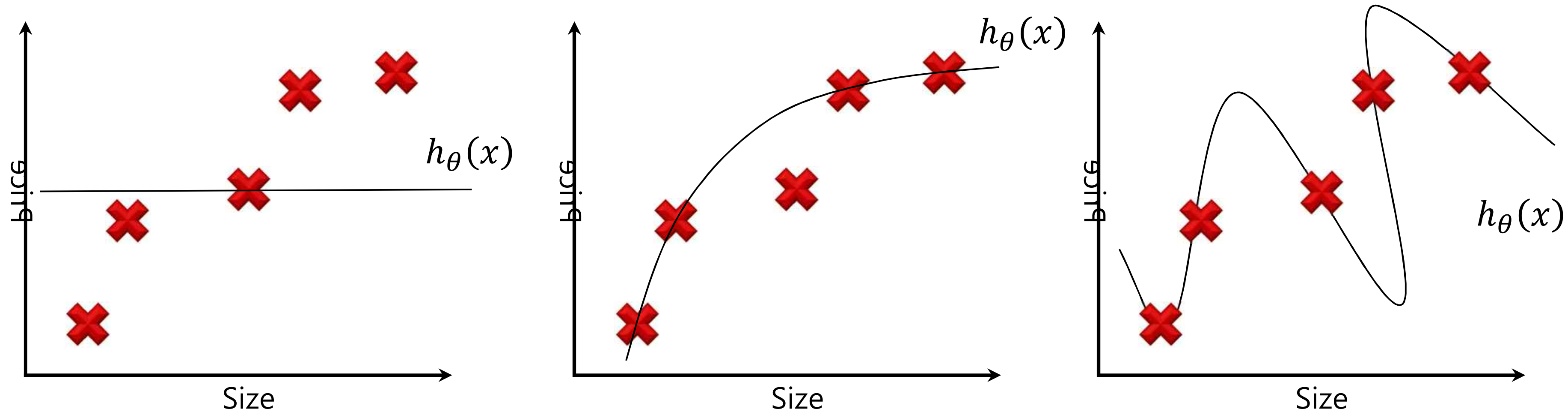
Advice for Applying Machine Learning

❖ Bias vs. Variance

▪ Linear Regression with Regularization

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

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



Large λ | Intermediate λ | Small λ

"Under fit" ➔ "High Bias" | **"Just Right"** | **"Over fit" ➔ "High Variance"**

Advice for Applying Machine Learning

❖ Bias vs. Variance

▪ Linear Regression with Regularization

- **Choosing the regularization parameter λ**
- Model ➔ $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

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

$$\begin{array}{ll}
\text{Try } \lambda = 0 & \theta^{(1)} \longrightarrow J_{cv}(\theta^{(1)}) \\
\text{Try } \lambda = 0.01 & \theta^{(2)} \longrightarrow J_{cv}(\theta^{(2)}) \\
\text{Try } \lambda = 0.02 & \theta^{(3)} \longrightarrow J_{cv}(\theta^{(3)}) \\
\text{Try } \lambda = 0.04 & \theta^{(4)} \longrightarrow J_{cv}(\theta^{(4)}) \\
\text{Try } \lambda = 0.08 & \theta^{(5)} \longrightarrow J_{cv}(\theta^{(5)}) \\
\quad\quad\vdots & \\
\text{Try } \lambda = 10 & \theta^{(10)} \longrightarrow J_{cv}(\theta^{(10)})
\end{array}$$

$$\min J_{cv}(\theta^{(d)}) \longrightarrow J_{test}(\theta^{(d)})$$

Advice for Applying Machine Learning

❖ Bias vs. Variance

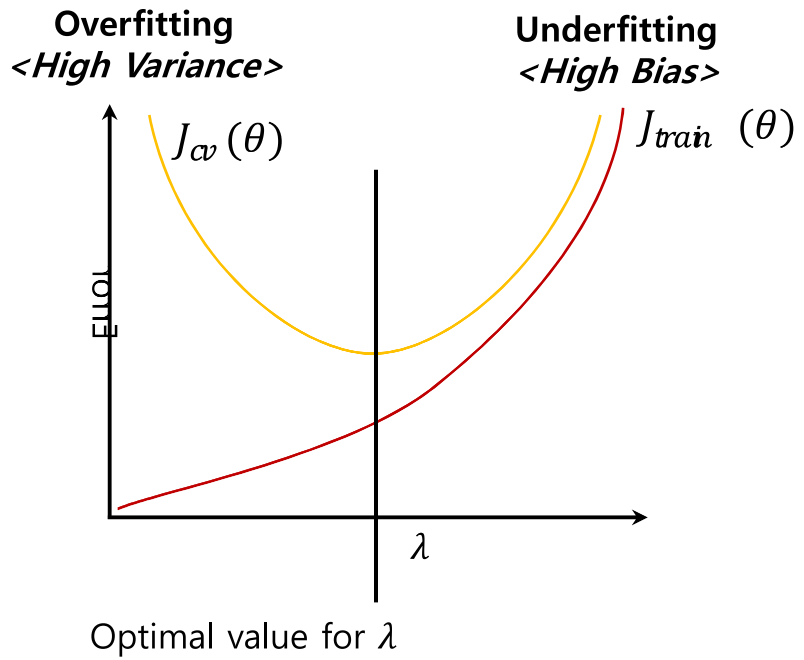
▪ Linear Regression with Regularization

- **Bias/Variance as a function of the regularization parameter λ**

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

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

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



Learning Curves

Bias vs. Variance

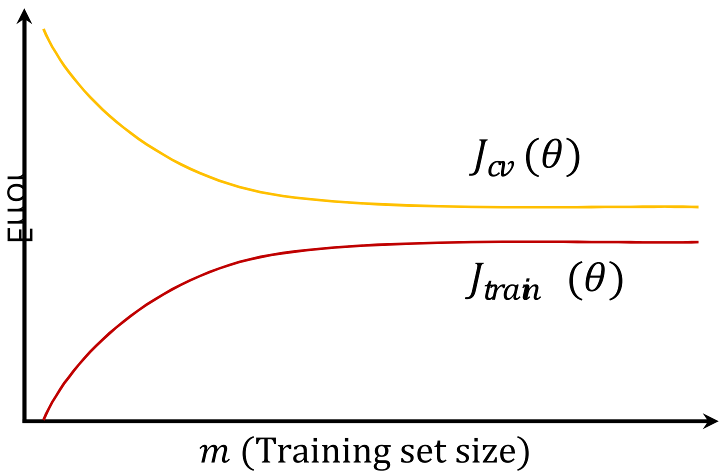
Advice for Applying Machine Learning

❖ Bias vs. Variance

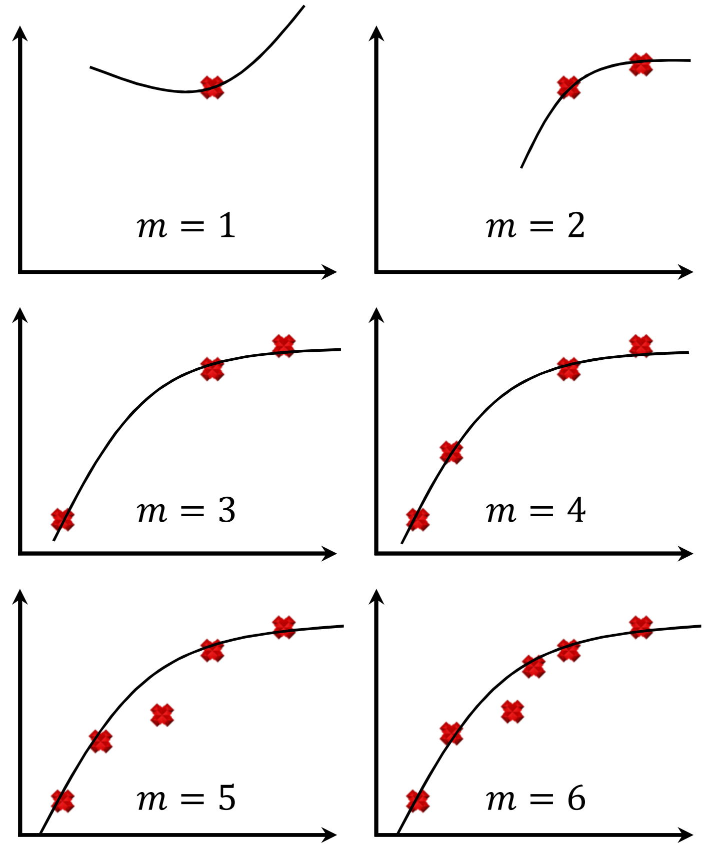
- **Learning Curves**

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

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



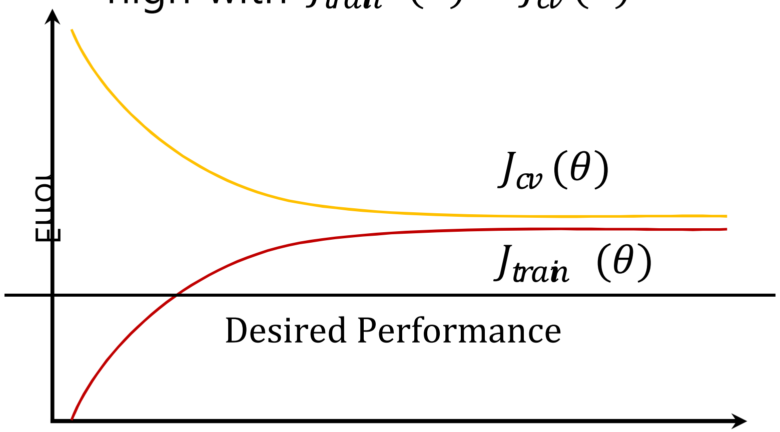
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$



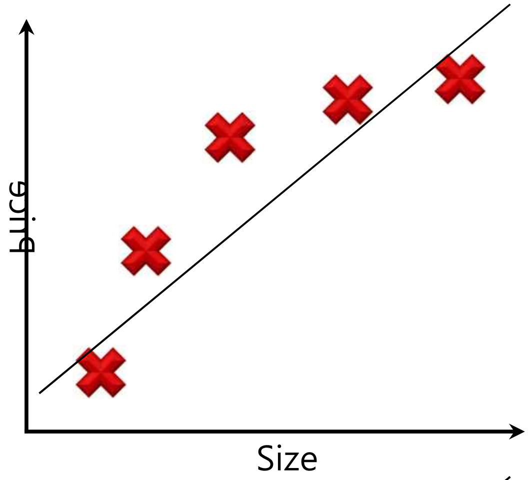
Advice for Applying Machine Learning

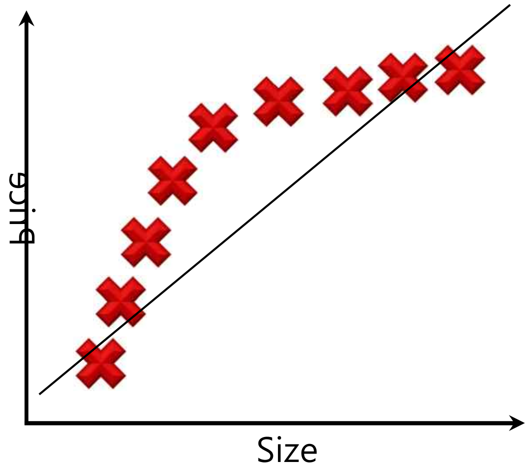
❖ **Bias vs. Variance**

- **Learning Curves**
  - **Typical learning curve for "*High Bias*"**
    - ✓ Low training set size
      - ➢ $J_{train}(\theta)$ to be low and $J_{cv}(\theta)$ to be high
    - ✓ Large training set size
      - ➢ both $J_{train}(\theta)$ and $J_{cv}(\theta)$ to be high with $J_{train}(\theta) \approx J_{cv}(\theta)$



$h_\theta(x) = \theta_0 + \theta_1 x$





※ **High Bias: Getting more training data will not help much.**

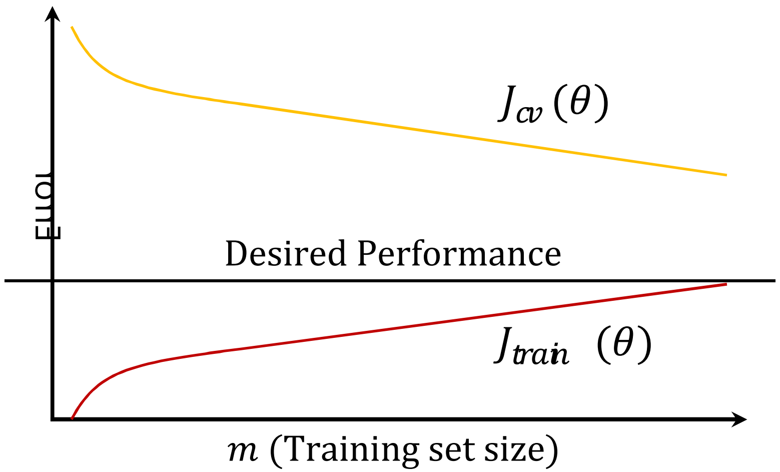
Advice for Applying Machine Learning

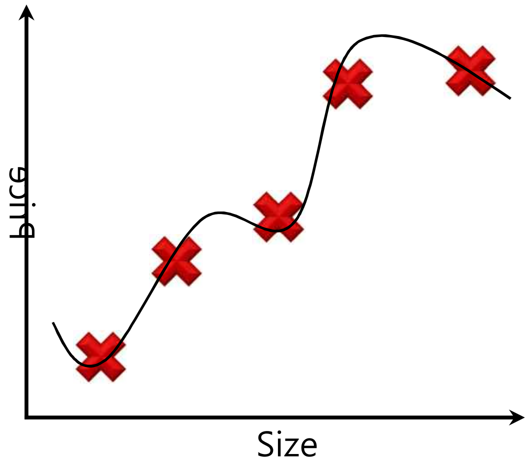
❖ Bias vs. Variance

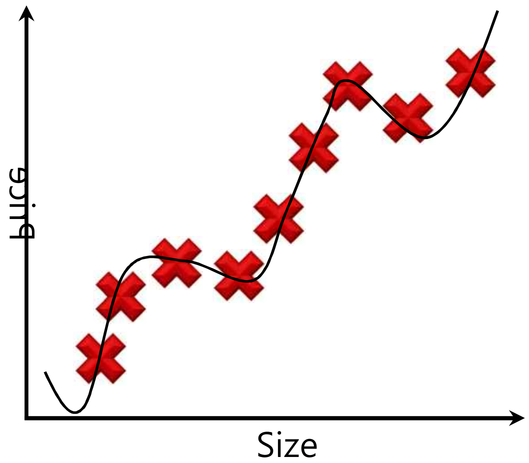
$$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$$

▪ Learning Curves

- **Typical learning curve for "*<u>High Variance</u>*"**
  - ✓ Low training set size
    - ➢ $J_{train}(\theta)$ to be low and $J_{cv}(\theta)$ to be high
  - ✓ Large training set size
    - ➢ $J_{train}(\theta) < J_{cv}(\theta)$







※ **High Variance: Getting more training data is likely to help.**

Deciding What to do Next(Revisited)

Bias vs. Variance

Advice for Applying Machine Learning

❖ Bias vs. Variance

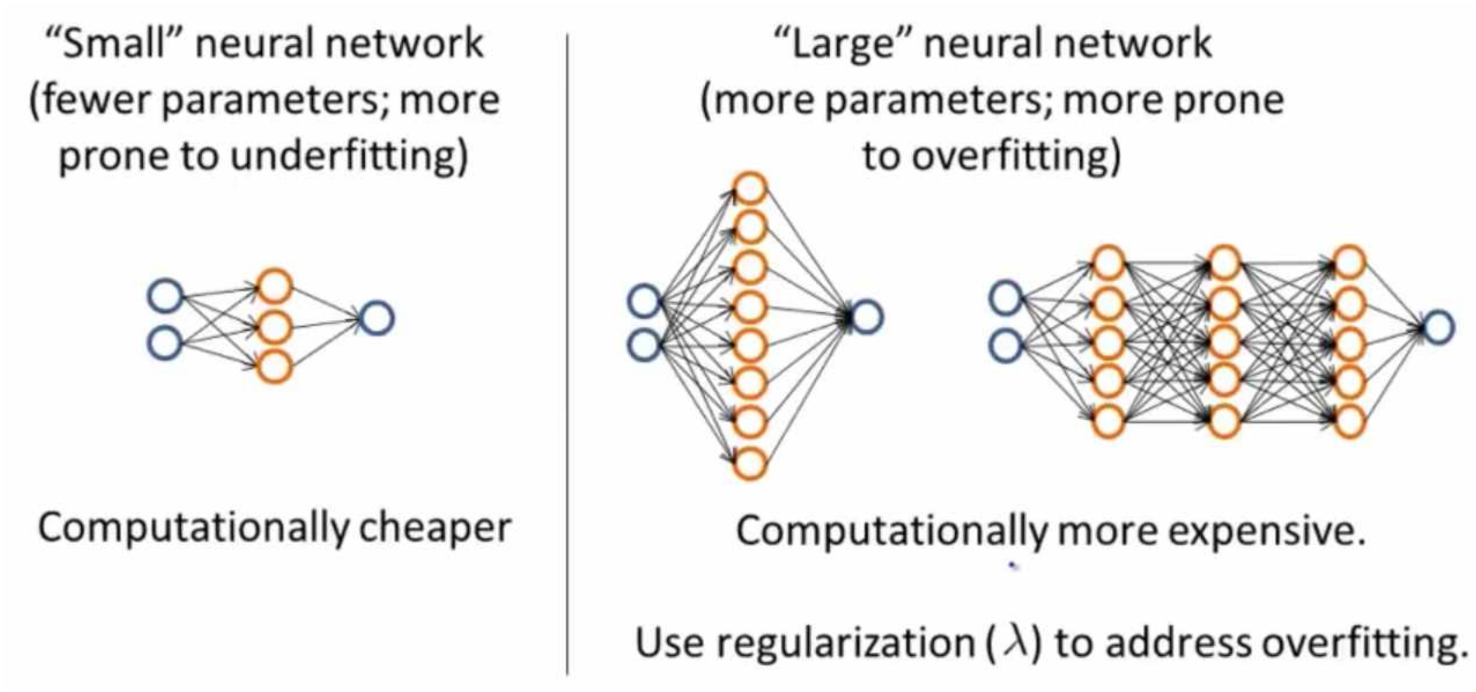
▪ Deciding What to Try Next

- **Once we have done some trouble shooting for errors in our prediction by:**
  - ✓ Get more training examples → **Fixing High Variance**
  - ✓ Try smaller sets of features → **Fixing High Variance**
  - ✓ Try getting additional features → **Fixing High Bias**
  - ✓ Try adding polynomial features → **Fixing High Bias**
  - ✓ Try decreasing λ → **Fixing High Bias**
  - ✓ Try increasing λ → **Fixing High Variance**

Advice for Applying Machine Learning

❖ **Bias vs. Variance**

- **Deciding What to Try Next**
  - **Neural Networks and Overfitting**



High Bias Problem **High Variance Problem**

Precision / Recall

Machine Learning System Design

Machine Learning System Design

- **Precision and Recall**
  - **Performance Evaluation**

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Machine Learning System Design

❖ Precision and Recall

- **Precision** (정확률)
  - True라고 예측한 것 중에서 실제로 True인 비율
    - ✓ $PRE = \frac{TP}{TP + FP}$

- **Recall** (재현률)
  - 실제 True인 것 중에서 True라고 예측한 비율
    - ✓ $REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

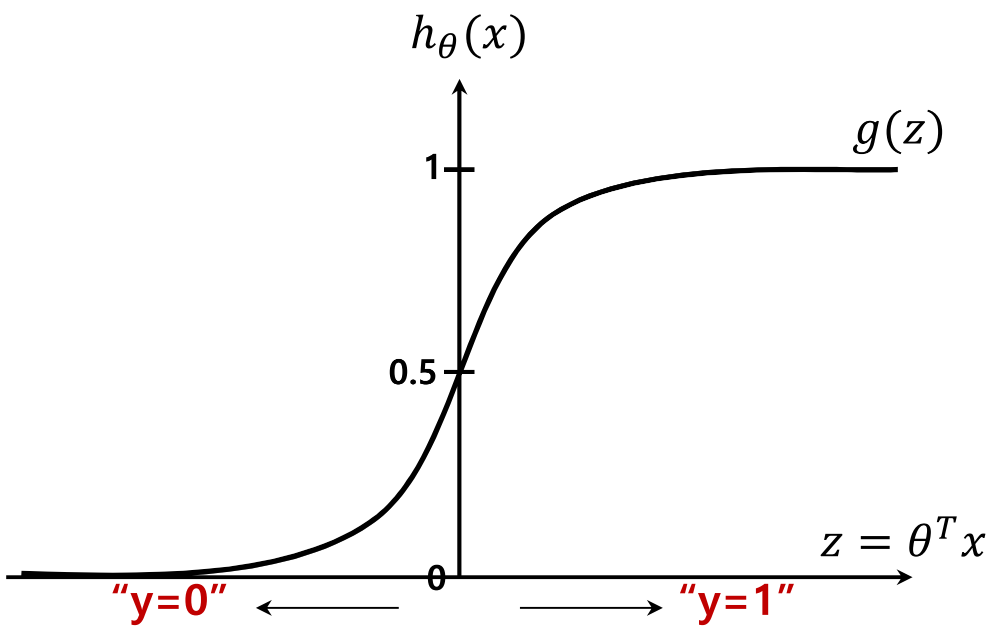
Machine Learning System Design

❖ Precision and Recall

- **Trading off → Threshold**
  - Logistic Regression $0 \leq h_\theta(x) \leq 1$

Suppose predict "y=1" if $h_\theta(x) \geq 0.5$

predict "y=0" if $h_\theta(x) < 0.5$



Machine Learning System Design

❖ Precision and Recall

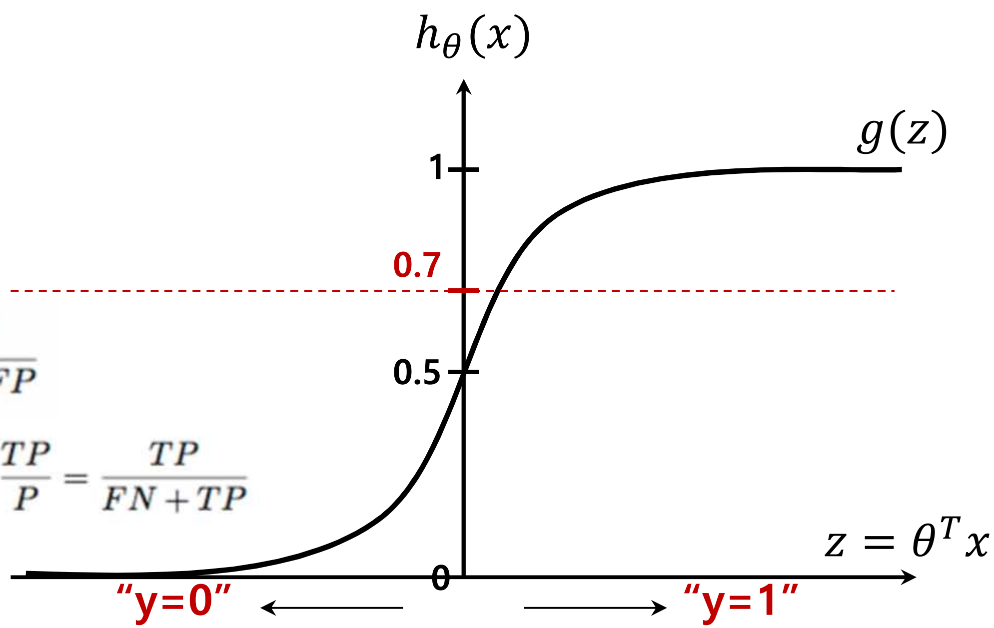
- **Trading off → Threshold**
  - Logistic Regression $0 \le h_\theta(x) \le 1$

Suppose predict "y=1" if $h_\theta(x) \ge 0.7$

predict "y=0" if $h_\theta(x) < 0.7$

$$PRE = \frac{TP}{TP + FP}$$

$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$$



Machine Learning System Design

❖ Precision and Recall

- **Trading off → Threshold**
  - Logistic Regression $0 \le h_\theta(x) \le 1$

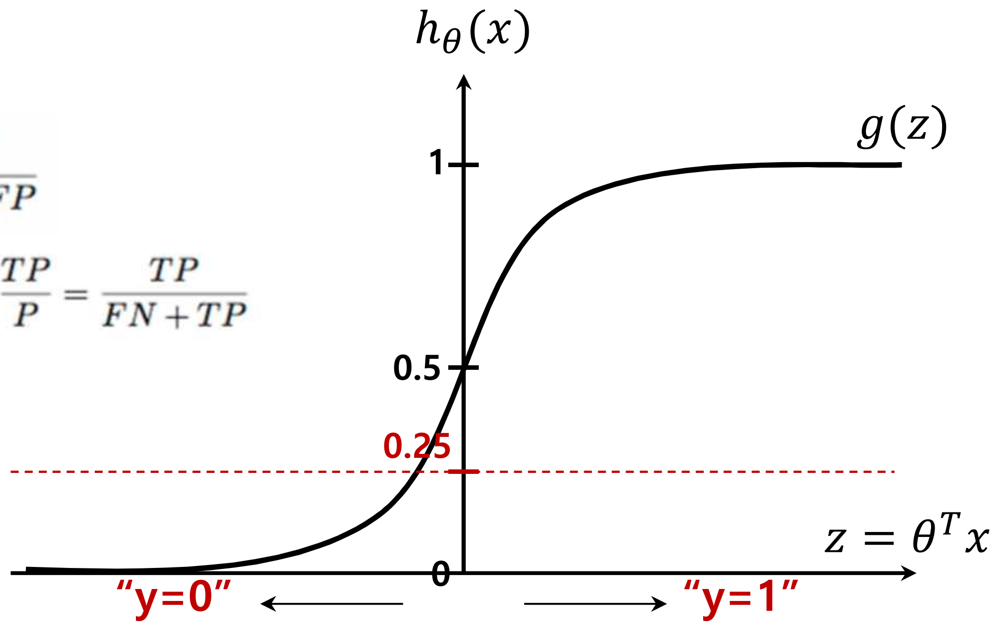
Suppose predict "y=1" if $h_\theta(x) \ge 0.25$

predict "y=0" if $h_\theta(x) < 0.25$

F_1 Score: $2\frac{PR}{P+R}$

$$PRE = \frac{TP}{TP+FP}$$

$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN+TP}$$



Machine Learning System Design

❖ Data for Machine Learning

▪ Large Data Rationale

- **Use a learning algorithm with many parameters**
  - ✓ Logistic Regression/Linear Regression with many features
  - ✓ Neural Network with many hidden units

$$\begin{cases} \text{Low bias algorithm} \\ J_{train}(\theta) \text{ will be small} \end{cases}$$

- **Use a very large training set**
  - ✓ Unlikely to overfit

$$\begin{cases} J_{train}(\theta) \approx J_{test}(\theta);\ \text{Low variance algorithm} \\ J_{test}(\theta) \text{ will be small} \end{cases}$$

※ **Large parameters && Very large training set = Best algorithm!**