

Standalone Deep-Learning

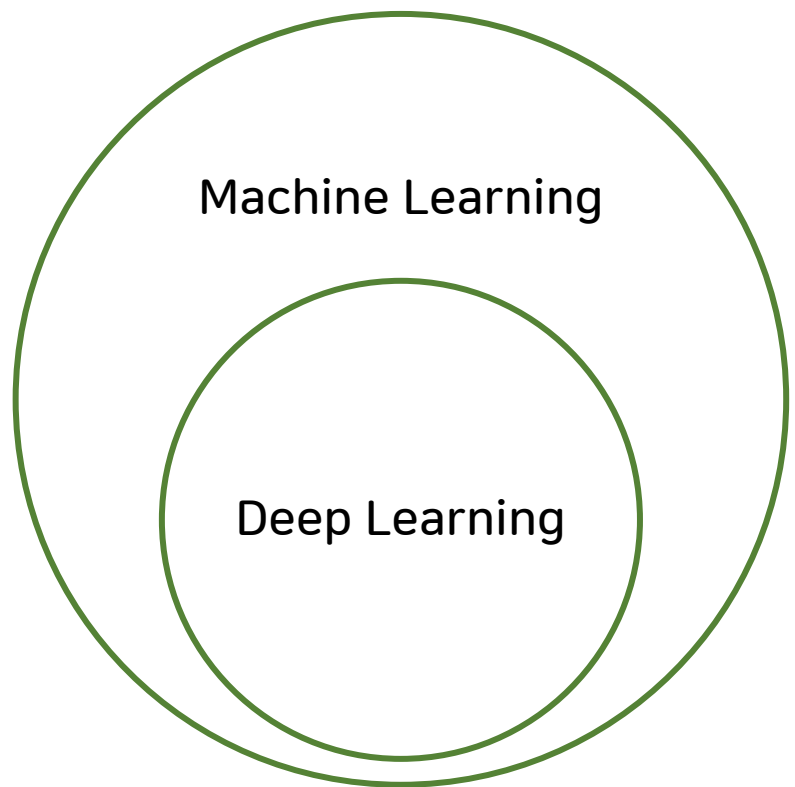
#2 ML Basic

#3 Linear Regression

#4 Linear Regression Practice

Contents

1	ML Basic	ML vs. DL Categories of ML Problems
2	Linear Regression	regression cost function gradient descent
3	Linear Regression Practice	code review



- 머신러닝(Machine Learning)

: AI의 한 종류.

학습을 통해 특정 업무를 실행할 수 있는 AI.
주로 사람이 특징을 정의.

- 딥러닝(Deep Learning)

: 머신러닝의 한 종류.

인간 뇌의 신경세포(뉴런)를 따라한 학습법에서 발전.
주로 기계가 특징을 자동으로 정의.

- 머신러닝(Machine Learning)



- 딥러닝(Deep Learning)



출처 : <https://modern-manual.tistory.com/3>

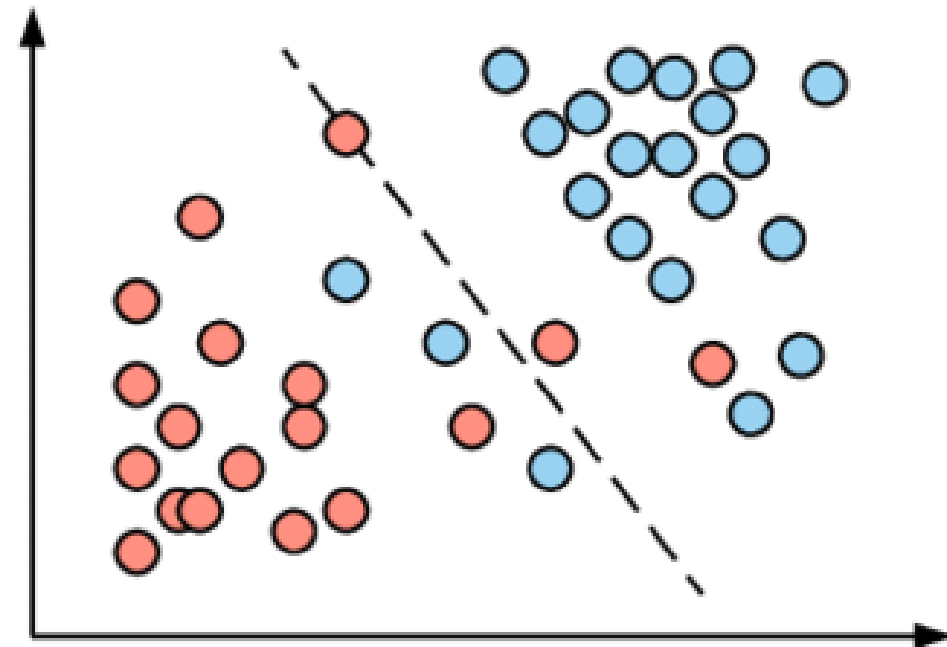
	Supervised	Unsupervised	Reinforcement
Discrete	Classification	Clustering	Discrete Action Space Agent
Regression	Regression	Dimensionality Reduction	Continuous Action Space Agent



Semi-Supervised Learning

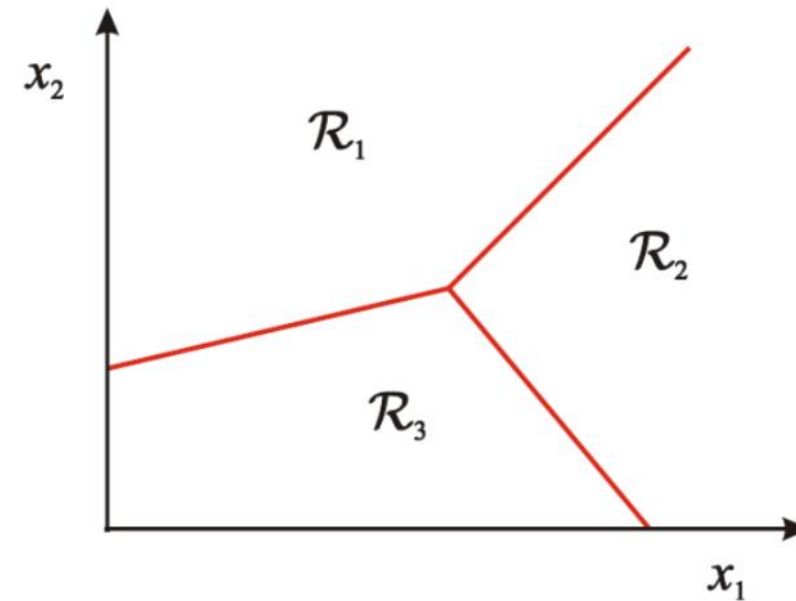
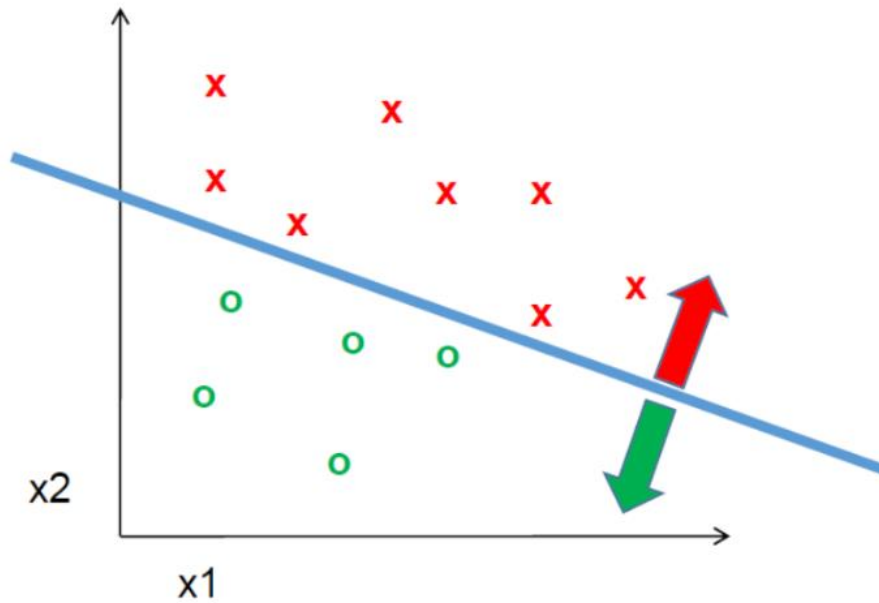
- Supervised Learning

	Supervised
Discrete	Classification
Regression	Regression

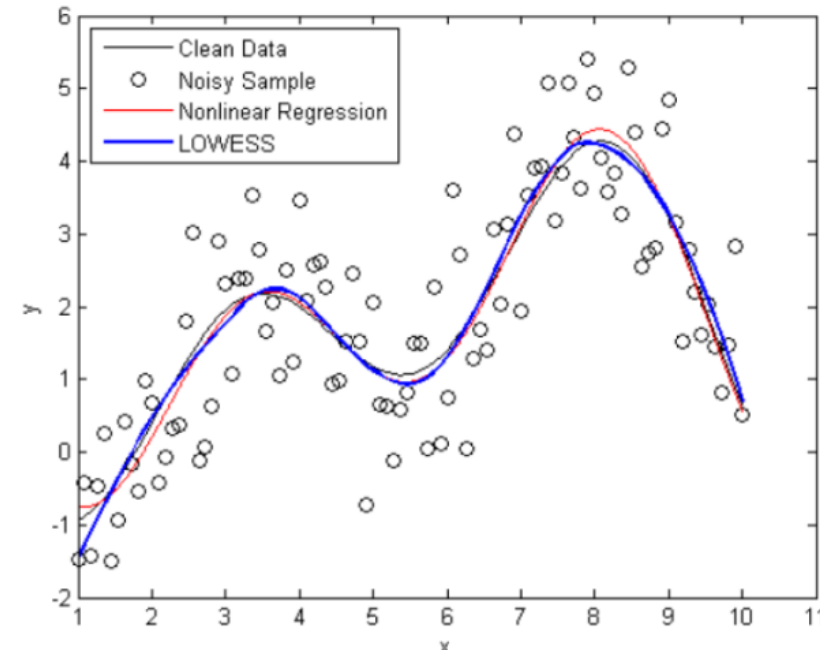
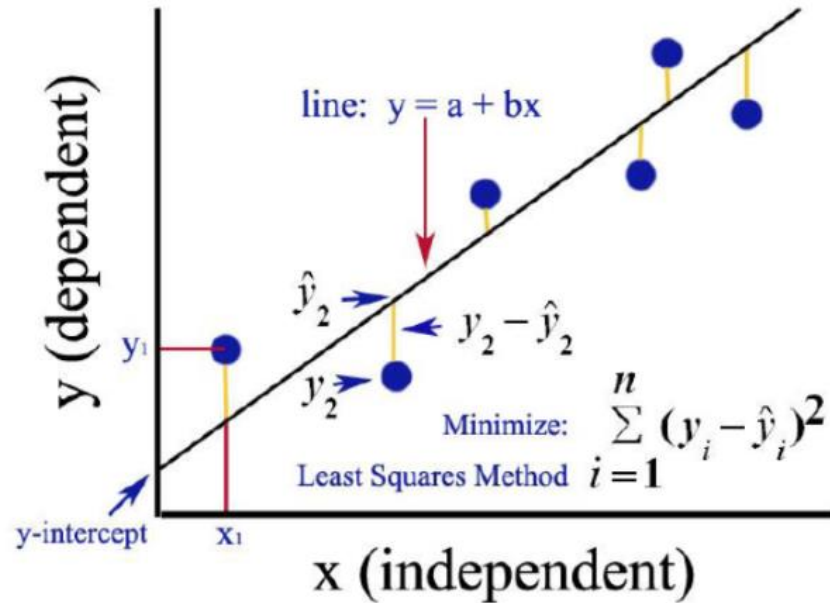


Supervised learning

- Classification

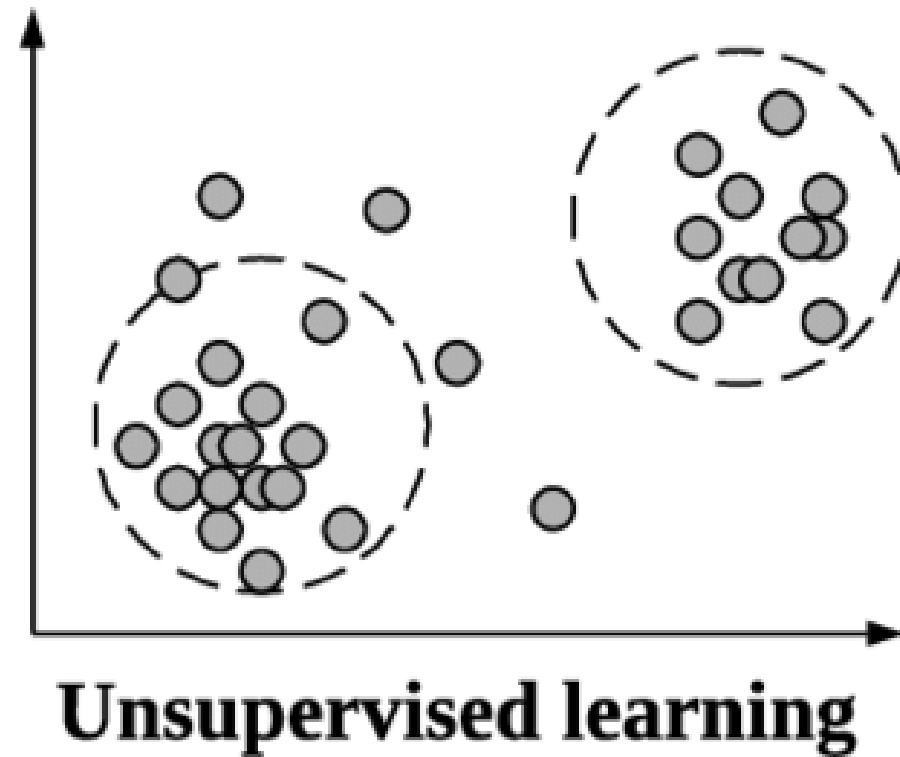


- Regression

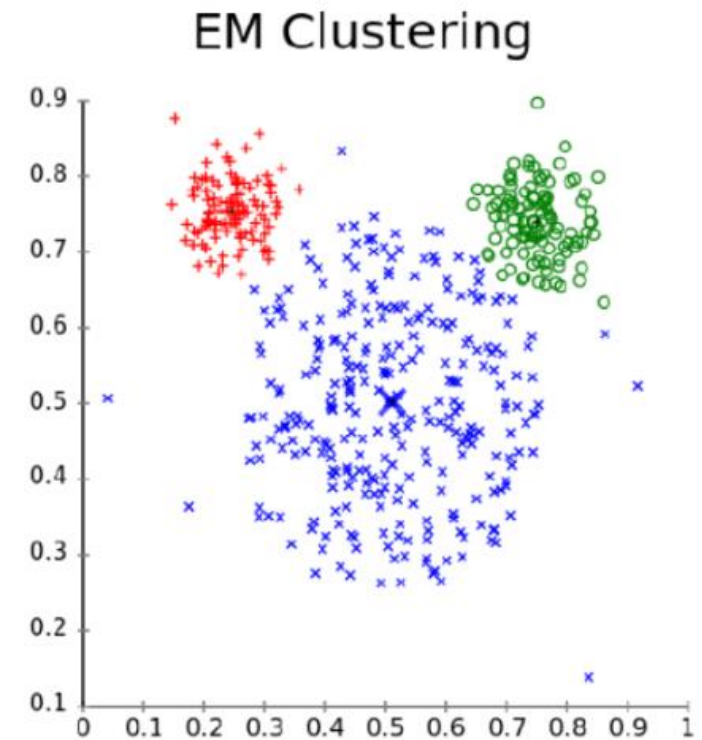
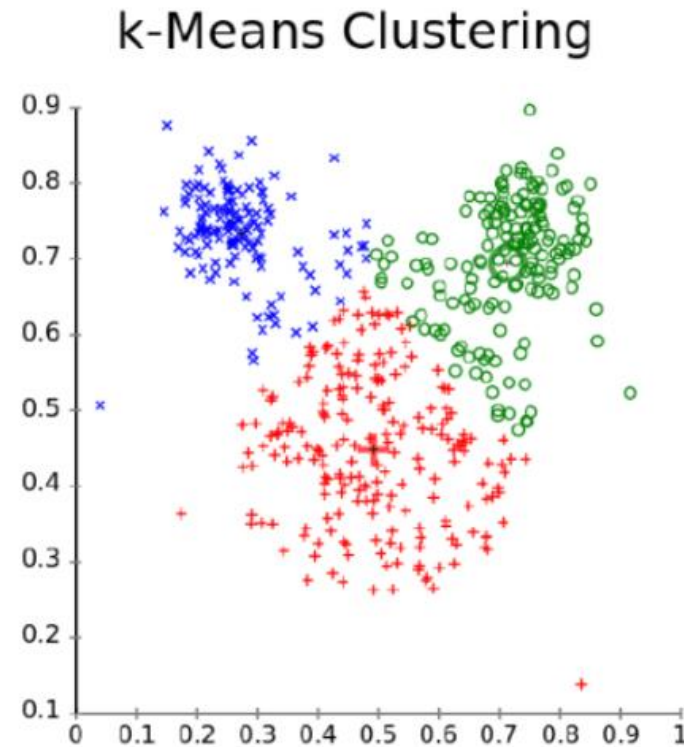
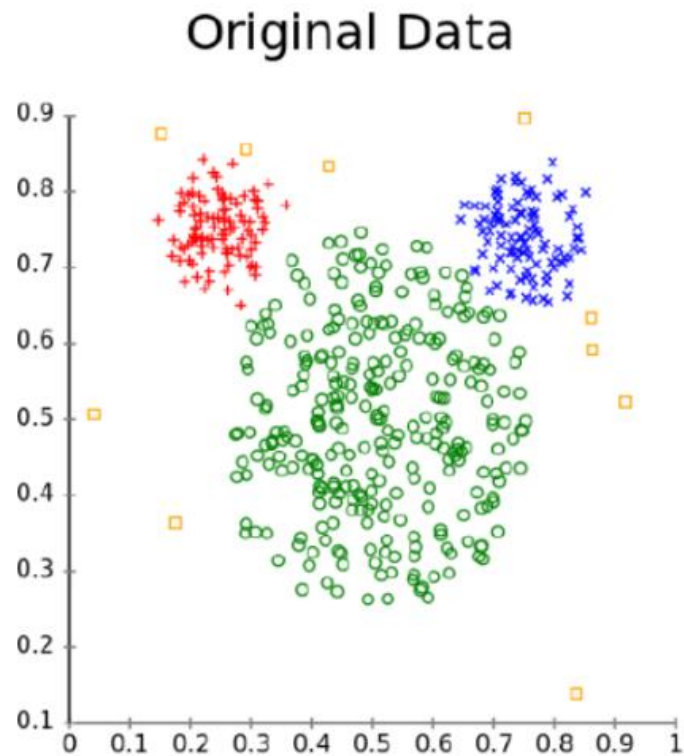


- Unsupervised Learning

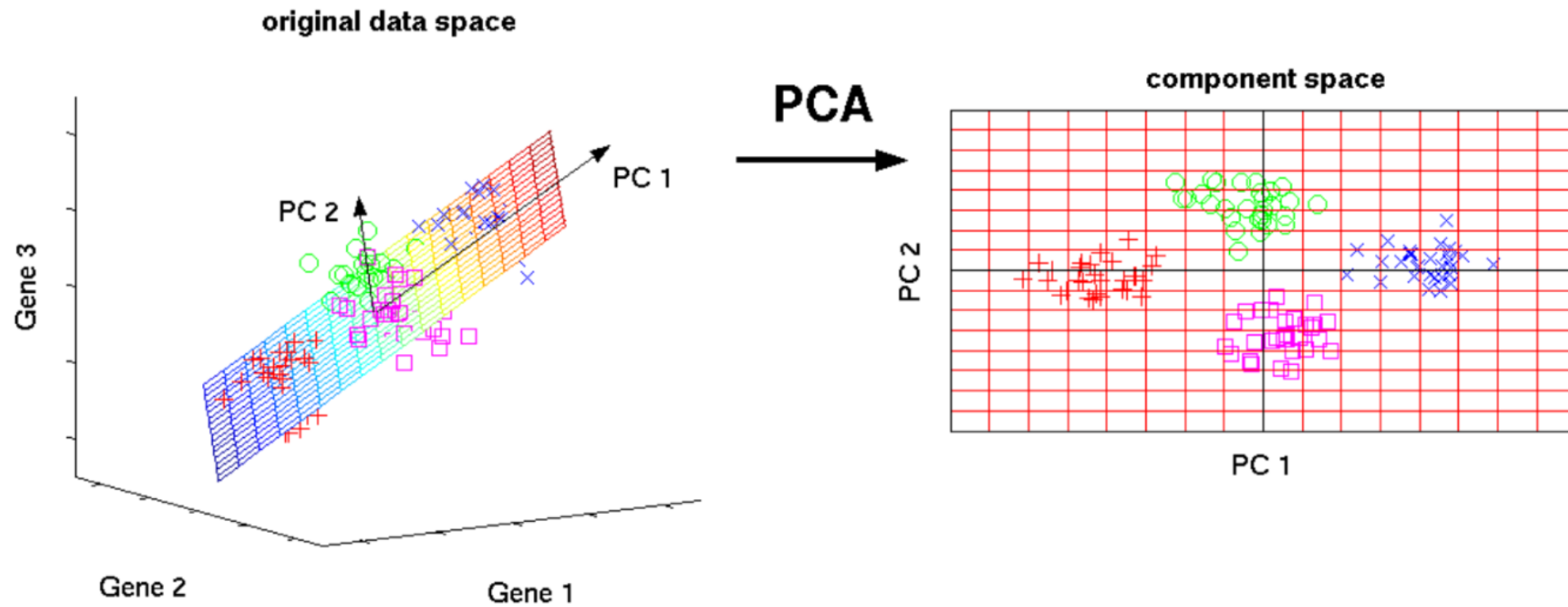
	Unsupervised
Discrete	Clustering
Regression	Dimensionality Reduction



- Clustering



- Dimensionality Reduction



- Dimensionality Reduction

Feature selection	Feature extraction
Feature 중에서 필요한 것만 선택	원본 feature의 조합으로 새로운 feature 생성
Genetic Algorithms	PCA, LDA

- Feature selection

→ Feature 중에서 필요한 것만 선택

Full Feature Set



Identify Useful Features



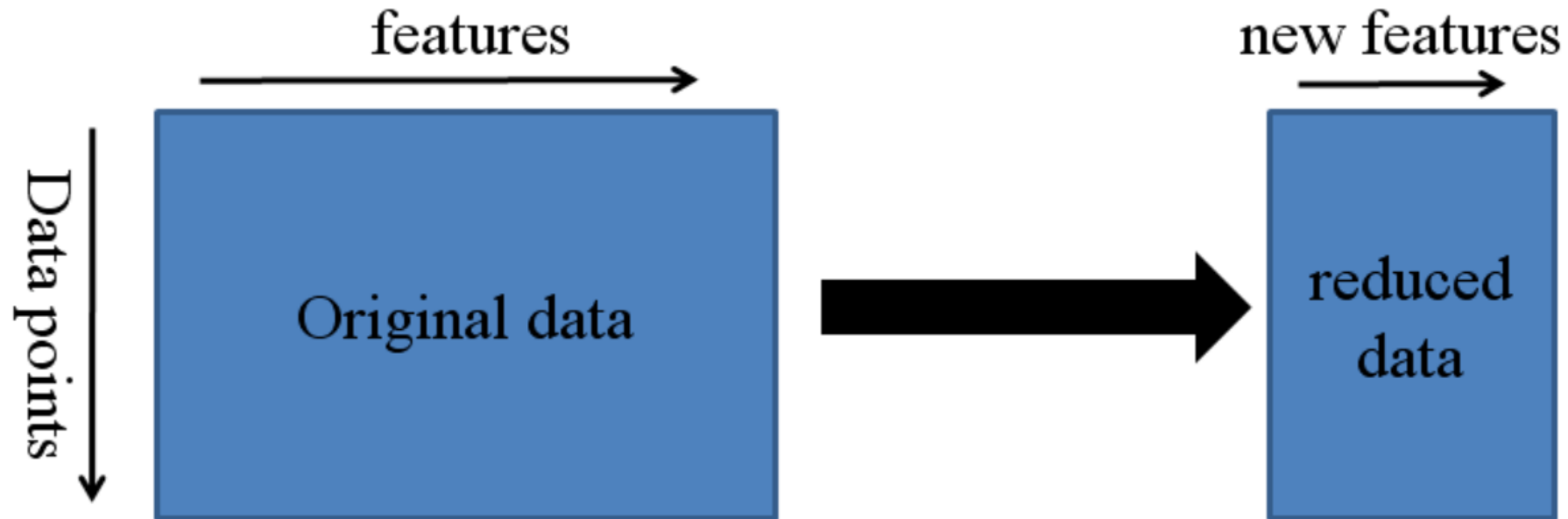
Selected Feature Set



출처 : <https://bioinformaticsandme.tistory.com/188>

- Feature extraction

→ 원본 feature의 조합으로 새로운 feature 생성



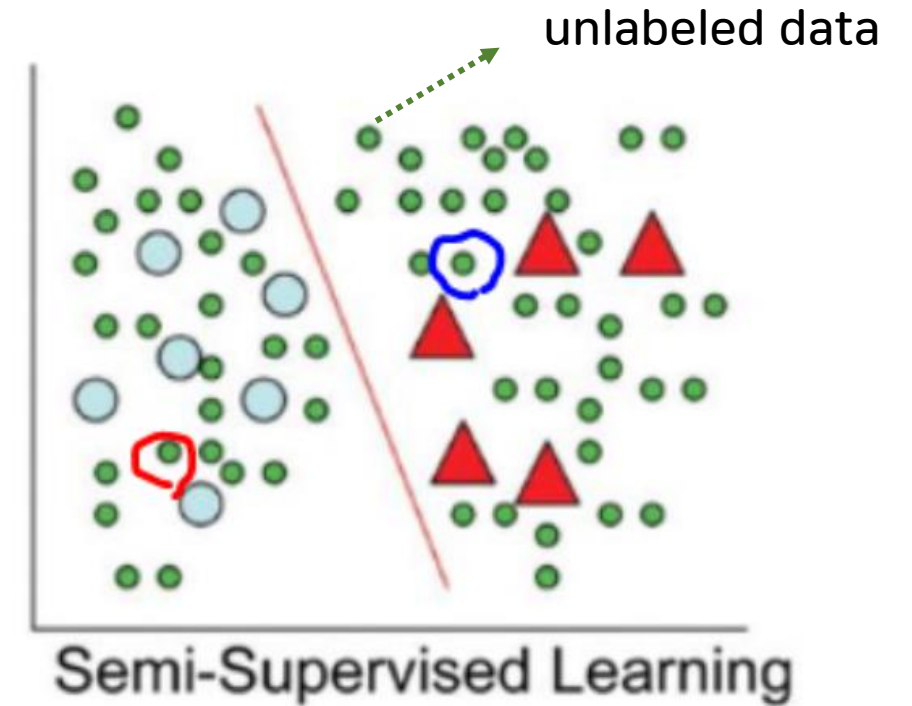
출처 : <https://bioinformaticsandme.tistory.com/188>

- Semi-Supervised Learning

	Supervised	Unsupervised
Discrete	Classification	Clustering
Regression	Regression	Dimensionality Reduction



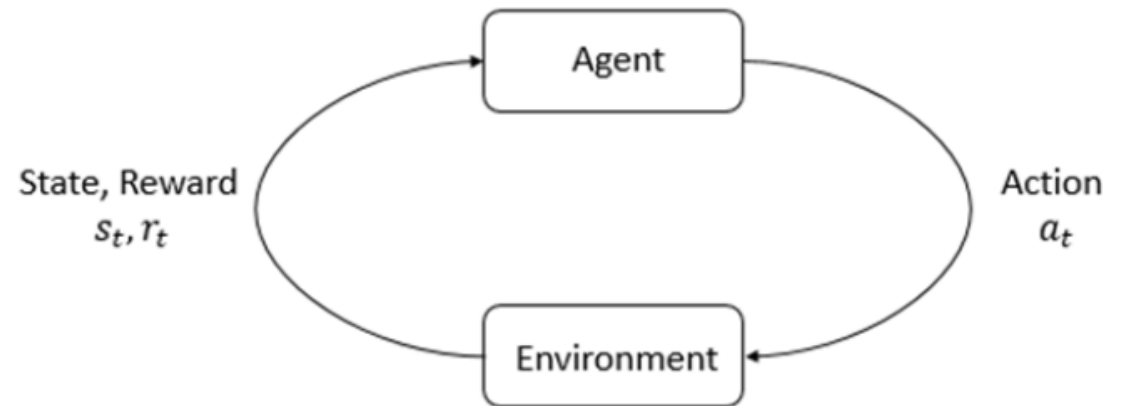
Semi-Supervised Learning



출처 : <https://techblog-history-younghunjo1.tistory.com/82>

- Reinforcement Learning

	Reinforcement
Discrete	Discrete Action Space Agent
Regression	Continuous Action Space Agent



출처 : <https://www.secmem.org/blog/2019/12/15/RL-key-concepts/>

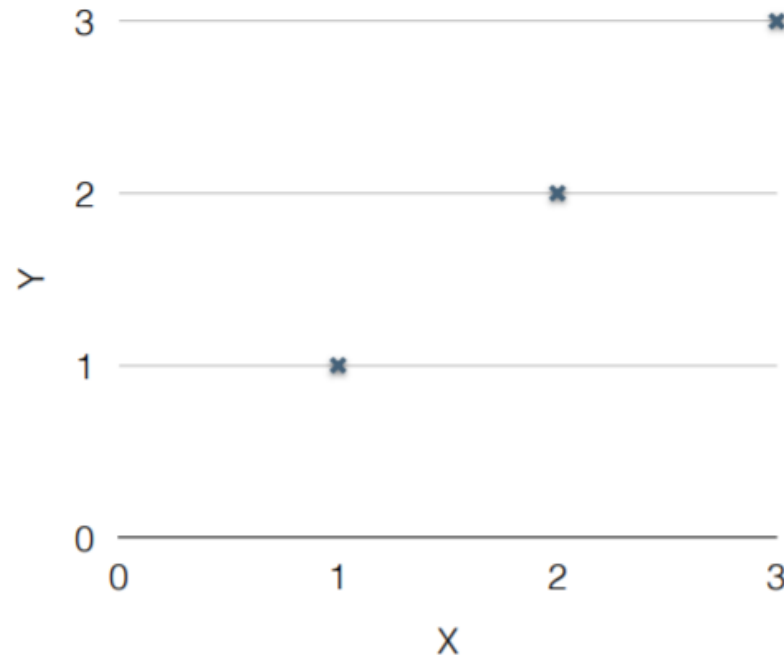
Linear Regression

regression

- Regression

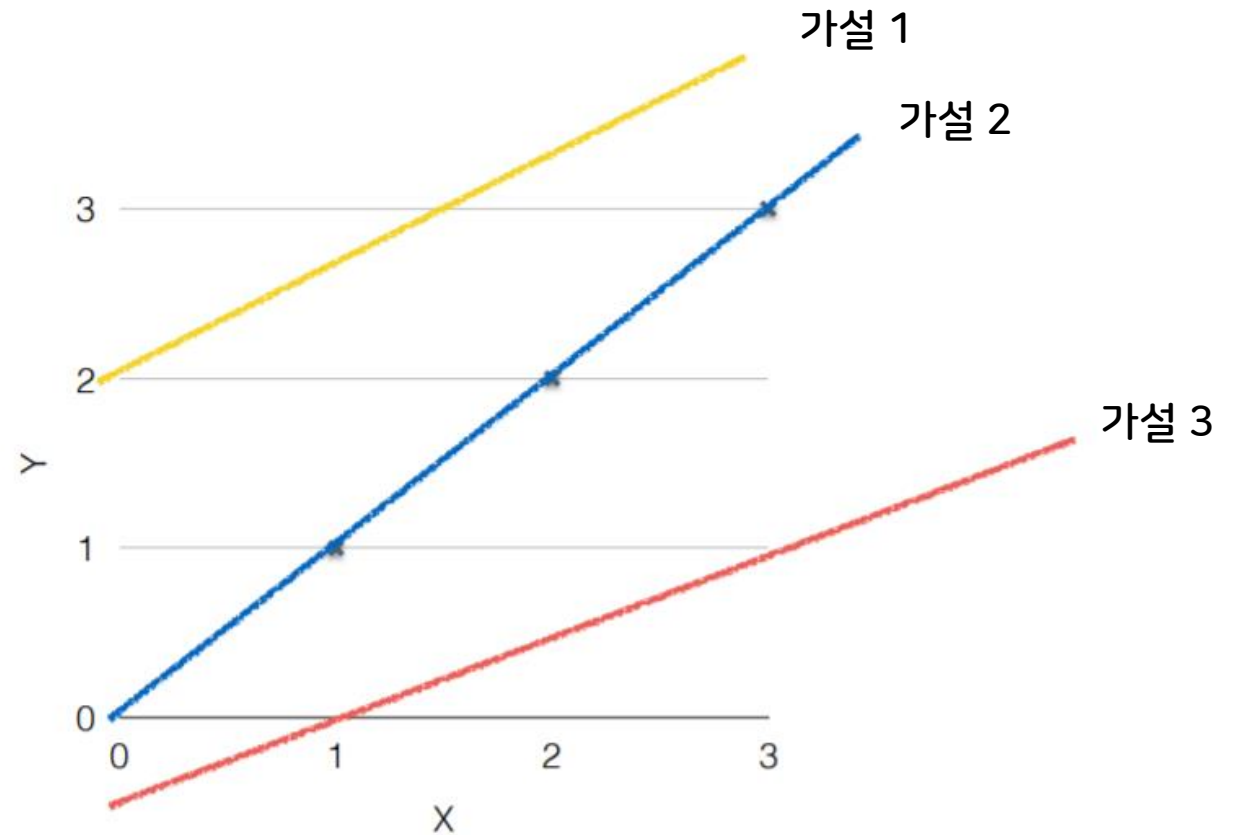
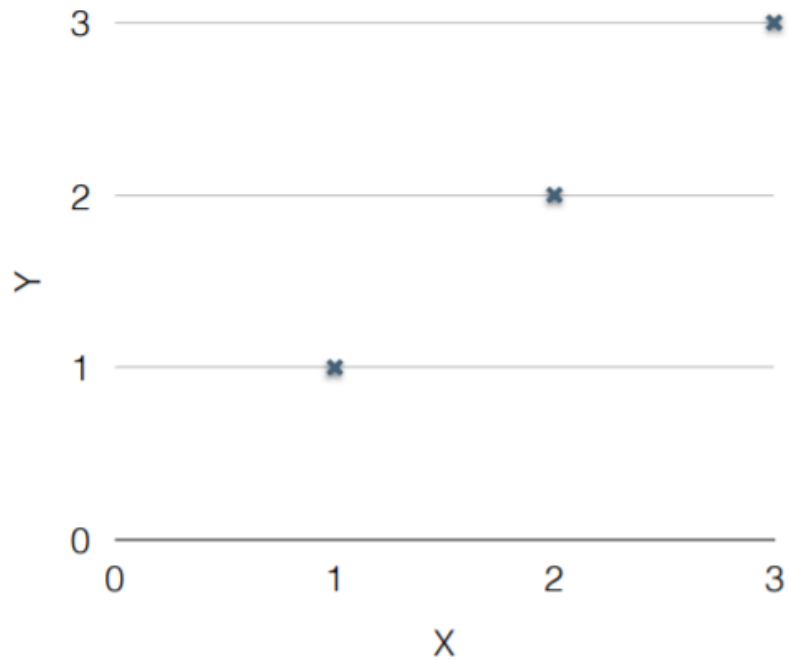
: 독립변수 x 로 종속변수 y 를 예측

x	Y
1	1
2	2
3	3



Linear Regression

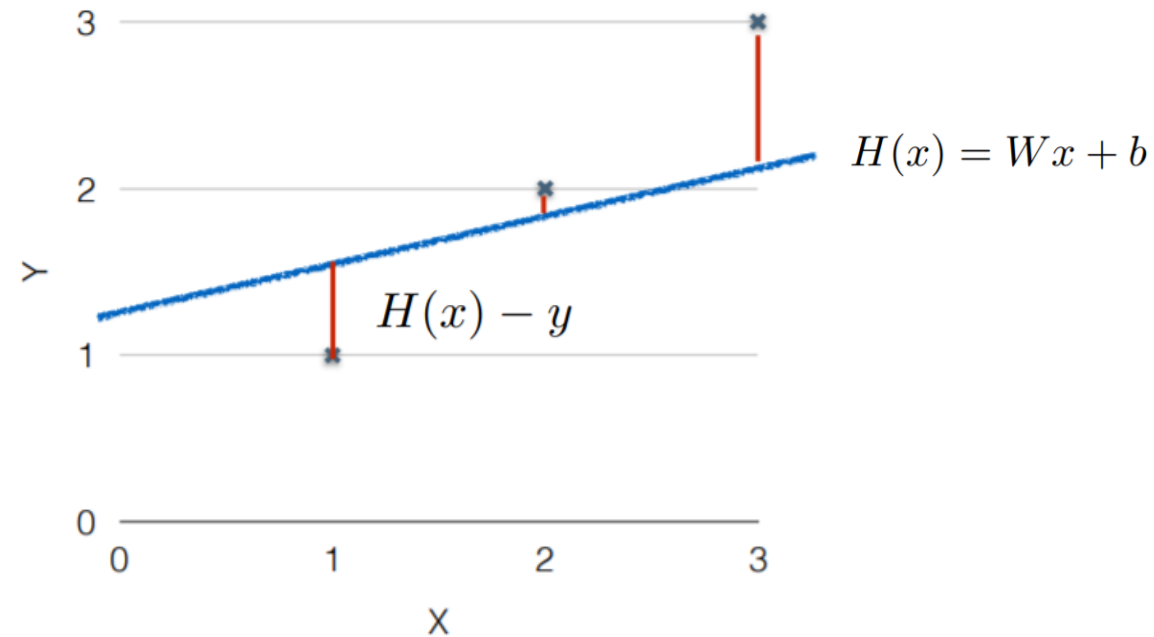
regression



$$H(x) = Wx + b$$

Linear Regression

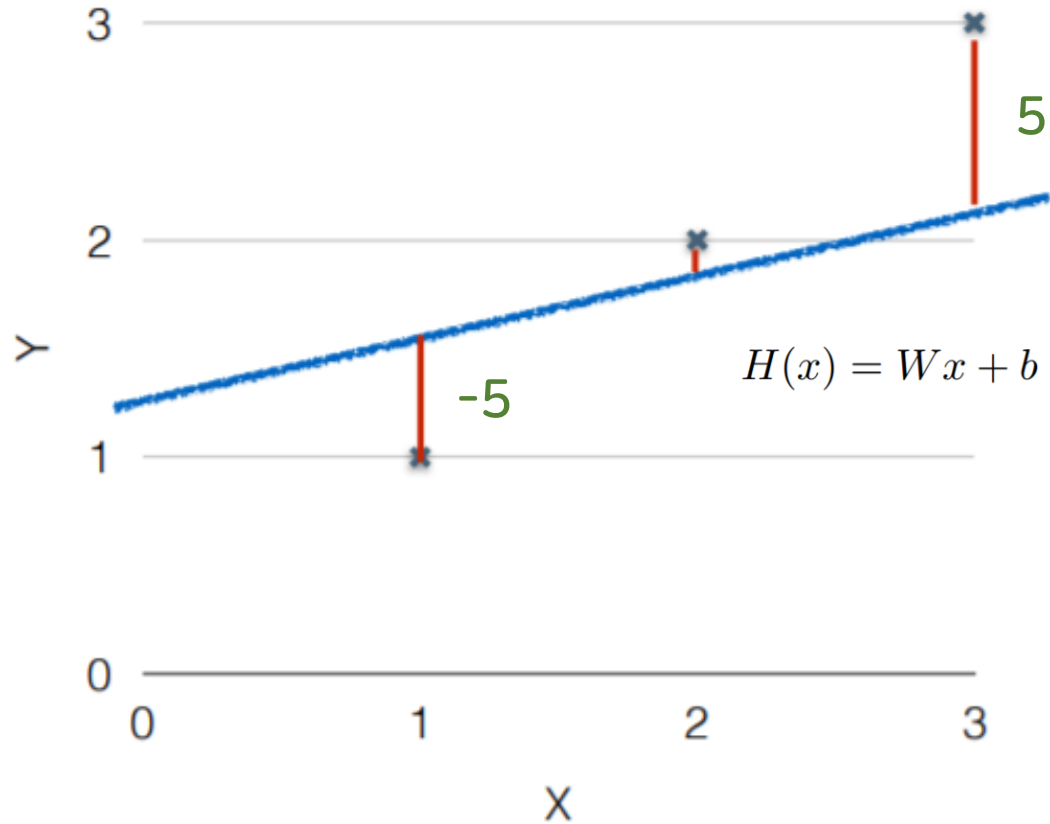
cost function



$$\frac{(H(x^{(1)}) - y^{(1)})^2 + (H(x^{(2)}) - y^{(2)})^2 + (H(x^{(3)}) - y^{(3)})^2}{3}$$

Linear Regression

cost function



$$H(x) - y \quad \times$$

$$(H(x^{(i)}) - y^{(i)})^2 \quad \circ$$

- loss function

: 하나의 input data에 대해서 오차를 계산하는 함수
→ single dataset

$$(H(x^{(i)}) - y^{(i)})^2$$

- cost function

: 모든 data에 대한 오차를 계산하는 함수
→ entire dataset

$$cost = \frac{1}{m} \sum_{i=1}^m (H(x^{(i)}) - y^{(i)})^2$$

Linear Regression

cost function

- Minimize Cost

X	Y
1	1
2	2
3	3

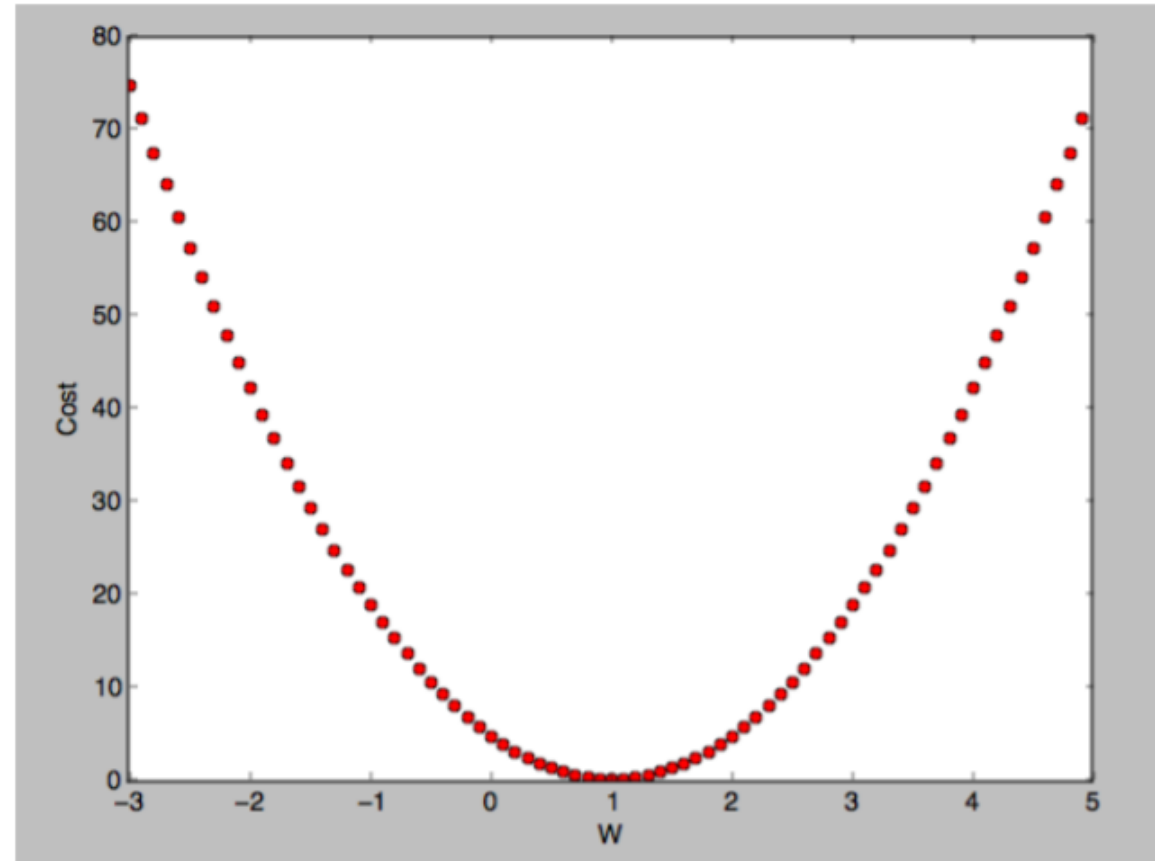
$$H(x) = Wx$$

$$\text{cost}(W) = \frac{1}{m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})^2$$

- $W=1, \text{cost}(W)=0$
- $W=0, \text{cost}(W)=4.67$
- $W=2, \text{cost}(W)=4.67$

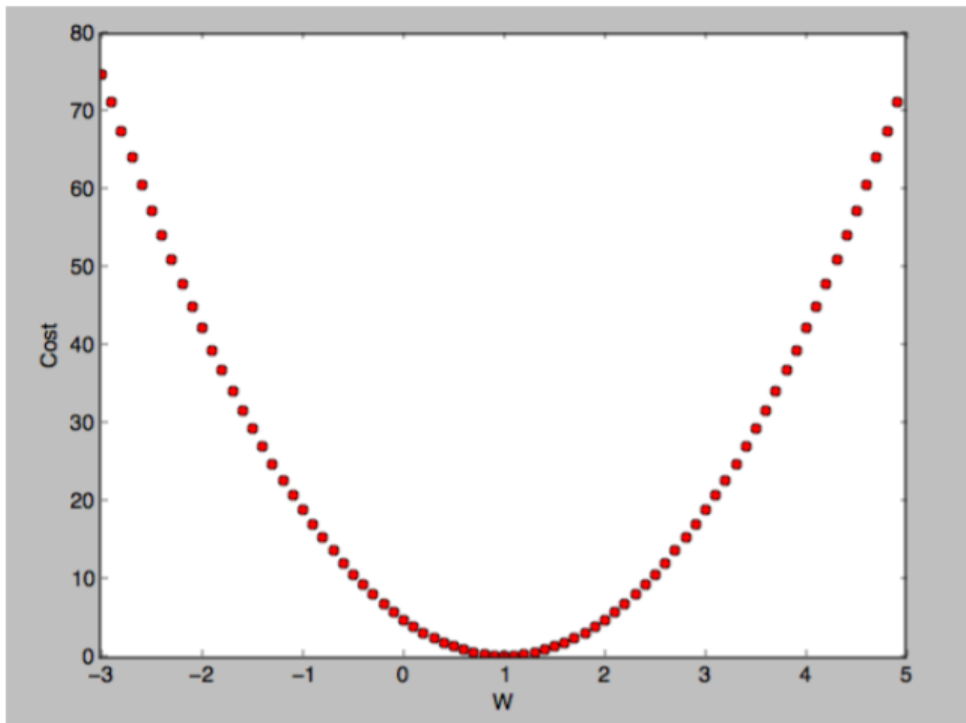
Linear Regression

gradient descent



Linear Regression

gradient descent



$$cost(W) = \frac{1}{2m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{\partial}{\partial W} cost(W)$$

Linear Regression

gradient descent

$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^m 2(Wx^{(i)} - y^{(i)})x^{(i)}$$

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})x^{(i)}$$

- cost function

```
def better_cost(pred_y, true_y):  
    error = 0  
    for i in range(len(x)):  
        error += (pred_y[i] - true_y[i])**2  
    error = error / len(x)  
    return error  
  
pred_y = [h.forward(x[i]) for i in range(len(x))]  
print('cost value with better code structure :', better_cost(pred_y, y))  
  
cost value with better code structure : 222.2
```

- gradient

```
def cal_grad2(w, cost):  
    h = H(w)  
    grad = 0  
    for i in range(len(x)):  
        grad += 2 * (h.forward(x[i]) - y[i]) * x[i]  
    grad = grad / len(x)  
    c = cost(h, x, y)  
    return grad, c
```

$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^m 2(Wx^{(i)} - y^{(i)})x^{(i)}$$

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})x^{(i)}$$

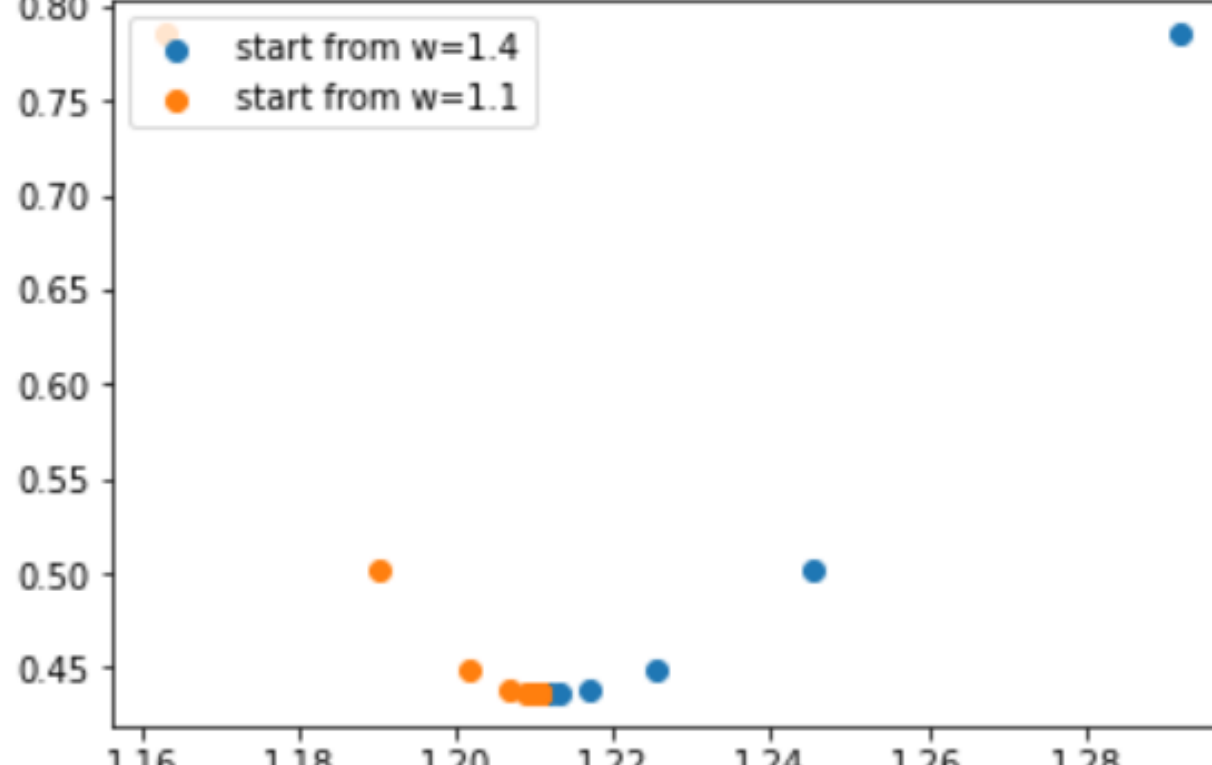
- gradient

```
for i in range(100):  
    grad, mean_cost = cal_grad(w1, cost)  
    grad2, mean_cost2 = cal_grad2(w2, cost)  
  
    w1 -= lr * grad  
    w2 -= lr * grad2  
    list_w1.append(w1)  
    list_w2.append(w2)  
    list_c1.append(mean_cost)  
    list_c2.append(mean_cost2)
```

$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^m 2(Wx^{(i)} - y^{(i)})x^{(i)}$$

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})x^{(i)}$$



Q&A