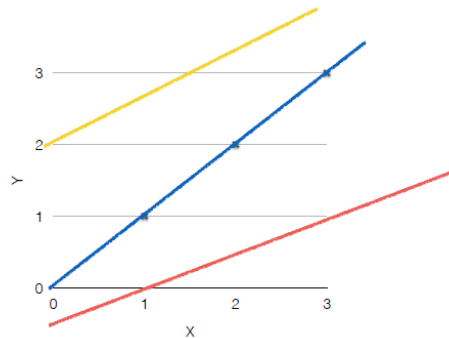
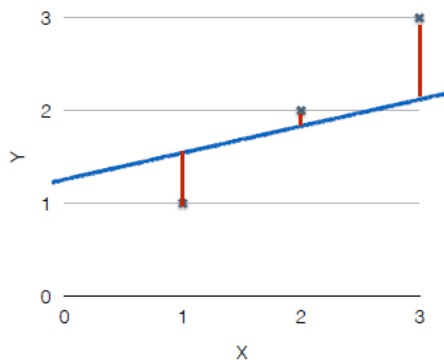


[선형 회귀]

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



Cost function (Loss)



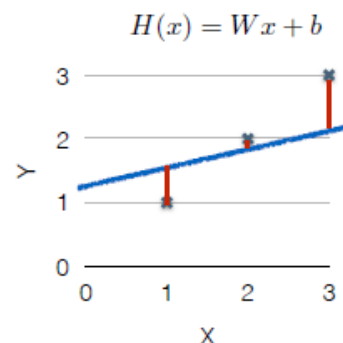
어떤 것이 좋은 가설인가? 실제 값과 거리가 먼 것은 나쁘고, 가까운 것은 좋은 것.

Cost function (Loss) : 우리가 세운 가설과 실제 데이터가 얼마나 가까운가를 구하는 함수.

가설을 세운 $H(x)$ 에서 실제 값 y 의 차를 구한 후 제곱의 합

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

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



머신 러닝의 목표 : minimize cost

회귀분석(regression analysis)은 D차원 벡터 독립 변수 x 와 이에 대응하는 스칼라 종속 변수 y 간의 관계를 정량적으로 찾아내는 작업이다.

회귀분석에는 결정론적 모형(deterministic Model)과 확률적 모형(probabilistic Model)이 있다.

결정론적 회귀분석 모형은 독립 변수 x 에 대해 대응하는 종속 변수 y 와 가장 비슷한 값 y^{\wedge} 를 출력하는 함수 $H(x)$ 를 찾는 과정이다.

만약 독립 변수 x 와 이에 대응하는 종속 변수 y 간의 관계가 다음과 같은 선형 함수 $H(x)$ 이면 선형 회귀분석(linear regression analysis)이라고 한다.

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + \cdots + w_Dx_D = w_0 + w^T x$$

위 식에서 w_0, \dots, w_D 를 함수 $H(x)$ 의 계수(coefficient)이자 이 선형 회귀모형의 parameter 라고 한다.

(핵심 함수)

```
model = LinearRegression()    # 학습 모델 선택
model = model.fit(x, y)       # 학습
result = model.predict([[7]]) # 예측
```

ex07_regression.py (단순 선형 회귀) x: [공부시간]

```
from sklearn.linear_model import LinearRegression

x = [[10], [5], [9], [7]]    #공부시간 10시간, 5시간, 9시간, 7시간
y = [[100], [50], [90], [77]] #시험점수 100점, 50점, 90점, 77점

model = LinearRegression()

model = model.fit(x, y)
result = model.predict([[7]])

print(result)
```

실행 결과:

```
[[72.01694915]]
```

ex08_regression.py (다중 선형 회귀) x: [공부시간, 학년]

```
from sklearn.linear_model import LinearRegression

x = [[10,3],[5,2],[9,3],[7,3]]      #공부시간,학년: 10시간,3   5시간,2, 9시간,3, 7시간,3
y = [[100],[50],[90],[77]]          #시험점수:      100점      50점,      90점      77점

model = LinearRegression()

model = model.fit(x, y)
result =model.predict([[7,2]])      #7시간공부, 2학년

print(result)
```

실행 결과:

[[65.]]