Linear Regression

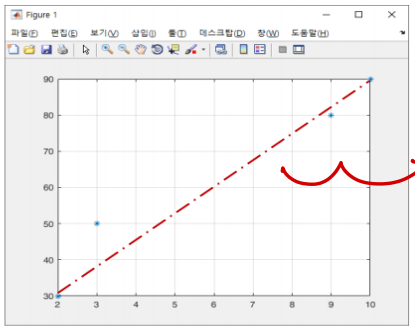
무인이동체공학과 17011882 김 우 혁

- 인공지능: 데이터와 task별(classification, regression 등) 모델을 지정해주면 최적의 솔루션을 기계가 찾아줌

- 기계가 배우는 것? 모델(선형 회귀, 선형 분류) 파라미터를 배움

  - Predicting exam score

x (hours)	y (score)
10	90
9	80
3	50
2	30

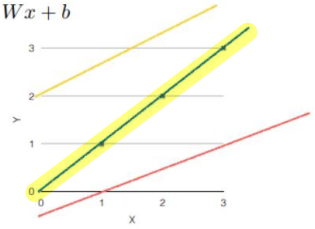


대략 65점?

- Hypothesis (가설 설정) 모델이 선형이라고 가정

(Linear) Hypothesis

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



- Cost function = Loss function = Objective function → Minimize cost

loss를 최소화하는 것이 목적!

우리가 손실 함수 제곱하는 이유 : 예) (-2) + (2) = 0 이런 경우, 음과 양이 서로 상쇄됨

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

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

$H(x) = Wx + b$ 가설 설정 → 우리가 어떤 모델을 사용할지에 따라 바뀜

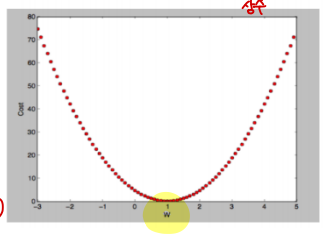
모델 파라미터 학습대상 요소들

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

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

최소화 하는 W, b 찾는 것이 목표 !!

$$cost(W) = \frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$ ⇒ '이차 함수'로 여겨 가능



- How to find W, b ?
  - **Goal: Minimize cost**

$$\underset{W,b}{\text{minimize}}\ cost(W,b)$$

$= \arg\min_{W,b} cost(W,b)$

모델을 학습시키는 방법 中 1개.
↳ cost를 최소로 하는 W와 b를 찾는 방법

- Gradient descent algorithm (경사 하강 알고리즘)

(기울기 = 0)

- 경사도가 없는 부분까지 내려가는 것 → 기울기의 절댓값이 작아지는 방향으로 진행
- 기울기 → "미분" 이용, 이때 미분상의 편의를 위해 cost(W) 함수를 변경

★ Linear Regression
① Hypothesis → 선형
② cost
③ GD

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

분모에 2 추가.

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

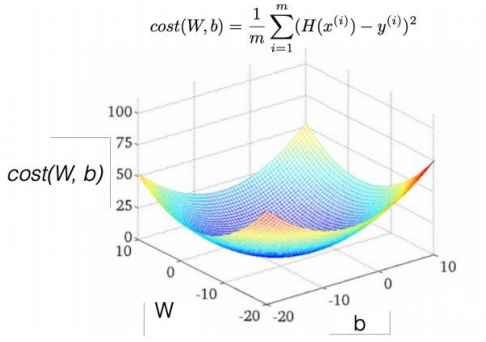
$$W_{(t+1)} := W_{(t)} - \alpha\frac{\partial}{\partial W_{(t)}}cost(W_{(t)})$$

(갱신전) W

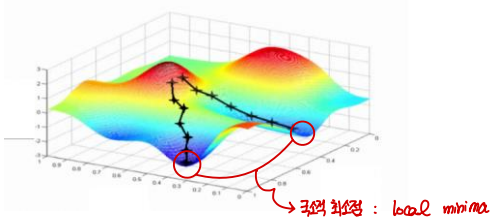
learning rate (학습률)

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

- Convex function → Linear htpothesis 는 Global minimum 만이 존재, ex) 선형회귀



- Non Convex function → 항상 global minimum 을 보장 X



국소 최소점 : local minima

- Summary

- Linear Regression → Hypothesis + Cost function + Gradient descent algorithm

$$H(x) = Wx + b \qquad cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$

- Regression using many inputs (x1, x2, x3, ···)

  - Hypothesis (multi-variable)

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

$$H(x_1, x_2, x_3, ..., x_n) = w_1x_1 + w_2x_2 + w_3x_3 + ... + w_nx_n + b$$

편

너무 길어서, 수식 쓰기 힘들다!! Matrix 란 녀석을 도입해보자.

  - Cost function (multi-variable)

$$cost(W,b) = \frac{1}{m}\sum_{i=i}^{m}\left(H\left(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \ldots, x_n^{(i)}\right) - y^{(i)}\right)^2$$

  - Implementation (Tensorflow)

$$H(X) = XW$$

matrix !!