

## Lab-03 Deeper Look at Gradient Descent

Dummy Data: input = output

모델의 좋고 나쁨을 평가하는 방법: Cost가 작을수록 좋음

MSE (Mean Squared Error): Cost (loss)  $= \frac{1}{m} \sum_{i=1}^m (H(x^{(i)}) - y^{(i)})^2$

Gradient Descent: 가중치가 가파를수록 W를 많이 하려고  
가중치가 완만하면 W를 더 크게,  
가중치가 완만하면 W를 더 작게 한다.

$$\Delta W = \frac{\partial \text{Cost}}{\partial W} = \frac{2}{m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)}) x^{(i)}$$

$$W := W - \underbrace{\Delta W}_{\text{learning rate}} \quad \text{Gradient}$$

epoch: 데이터로 학습한 횟수

\* torch.optim으로 gradient descent 가능

optimizer = optim.SGD([W], lr=0.15) *optimizer 정의*

optimizer.zero\_grad() *gradient를 0으로 초기화*

cost.backward() *gradient 계산*

optimizer.step() *gradient descent*