exam 개인학습

Lec 4

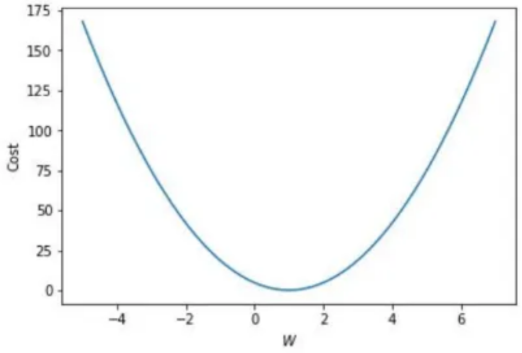
Type something...

01) Gradient Descent

01_Cost Function

Cost function: Intuition

- $W = 1$ 일 때 $cost = 0$
- 1 에서 멀어질수록 높아진다.



- cost를 최소환으로 하는 것이 gradient descent의 목적이다.

Cost function: MSE

Mean Squared Error (MSE)

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

Mean Prediction Target

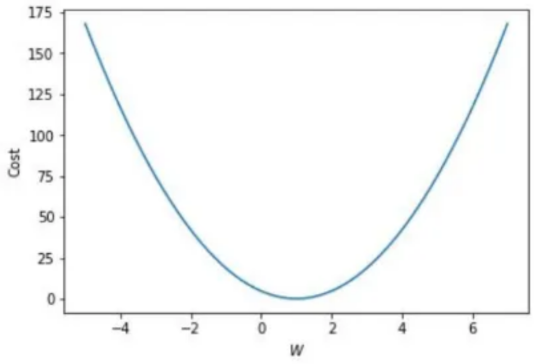
```
cost = torch.mean((hypothesis - y_train) ** 2)
```

- cost function의 대표적인 예시 중 하나인 **Mean Squared Error(MSE)**를 사용해보자.

Gradient Descent: Intuition

- 곡선을 내려가자
- 기울기가 클수록 더 멀리!
- **“Gradient”** 를 계산하자

$$\frac{\partial cost}{\partial W} = \nabla W$$



- 기울기가 음수라면 해당 방향으로 weight를 update 시키고, 기울기가 양수이면 해당 반대방향으로 weight를 update 시켜준다.

Gradient Descent: Code

$$\nabla W = \frac{\partial cost}{\partial W} = \frac{2}{m}\sum_{i=1}^{m}\left(W x^{(i)} - y^{(i)}\right)x^{(i)}$$

$$W := W - \alpha \nabla W$$

```
gradient = 2 * torch.mean((W * x_train - y_train) * x_train)
lr = 0.1
W -= lr * gradient
```

- MSE의 경우에는 gradient를 쉽게 계산해줄 수 있다. 그 후 `W = W - lr*gradient` 로 weight를 update시켜준다.