Gradient Descent

2021.10.15

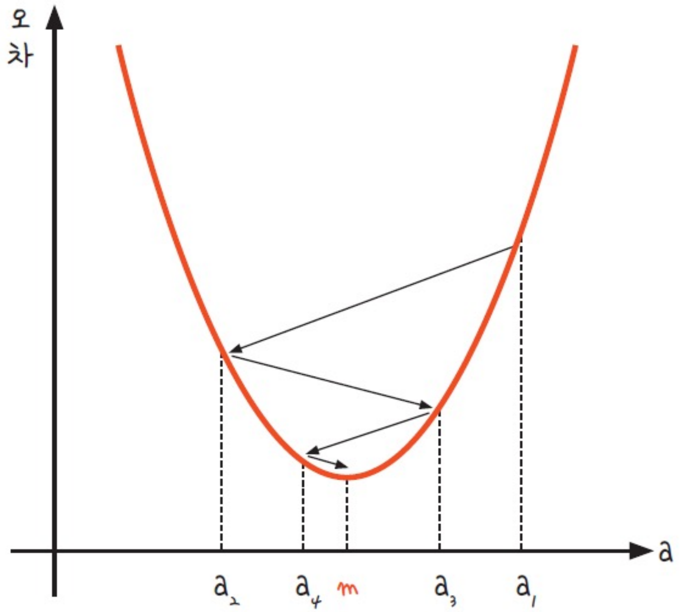
LG

KAIST AI
Graduate School of AI

Gradient Descent (1)

- 복잡한 모델의 경우, 최적의 파라미터를 찾는 것이 매우 어려움

- **경사 하강법**을 이용하자!
  : 손실 함수의 값을 최소화시키는 방향으로 모델 파라미터를 업데이트

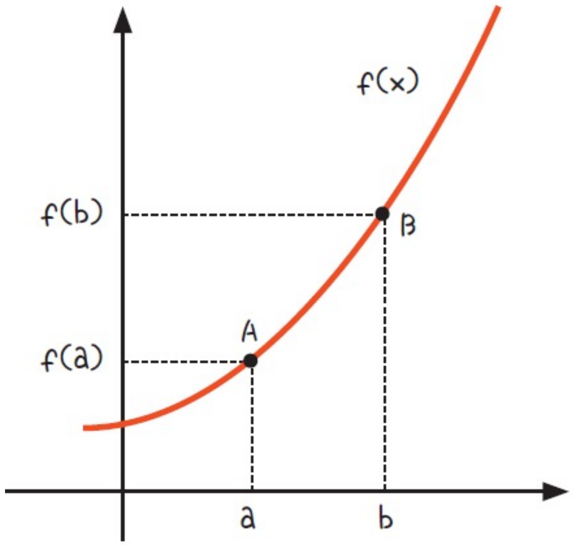


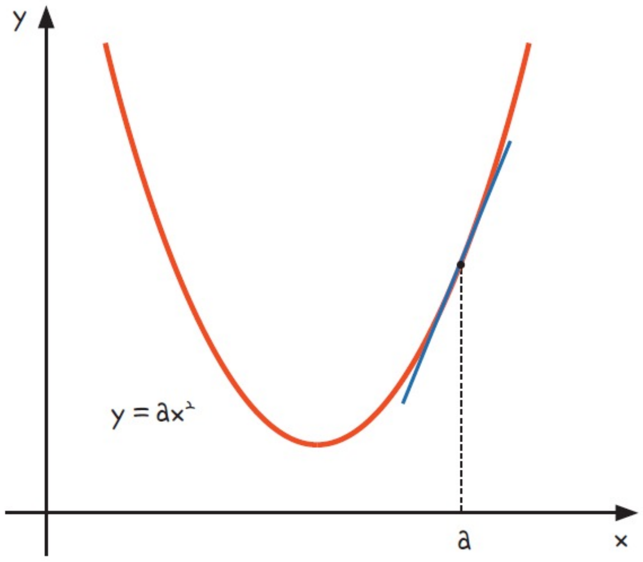
Gradient Descent (2)

- **미분**을 통해 함수에서의 **순간 변화율**을 구할 수 있음
- 순간 변화율은 x의 값이 미세하게 변화할 때, y의 변화율을 의미

$$\lim_{\Delta x \to 0} \frac{f(a + \Delta x) - f(a)}{\Delta x}$$

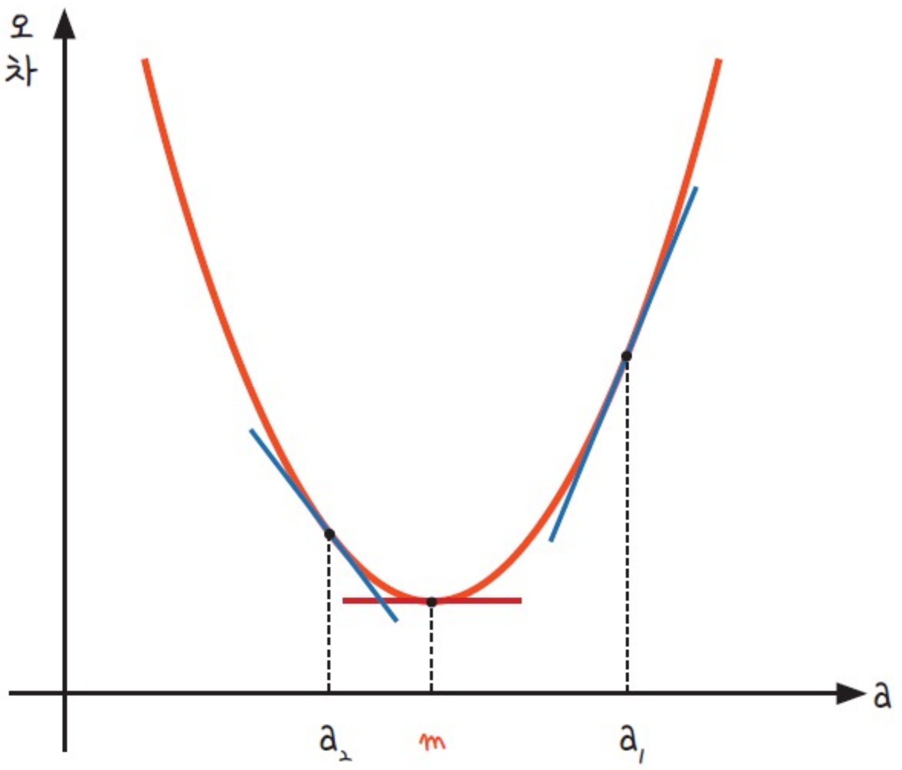
- 어떤 x값(=a)에서의 그래프와 맞닿는 접선의 기울기





Gradient Descent (3)

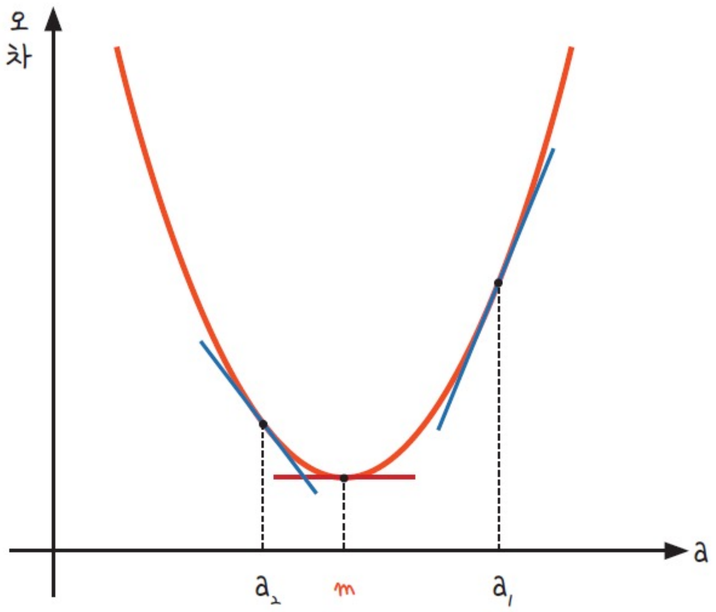
- 함수의 최솟값에서는 순간 변화율이 무조건 0임



Gradient Descent (4)

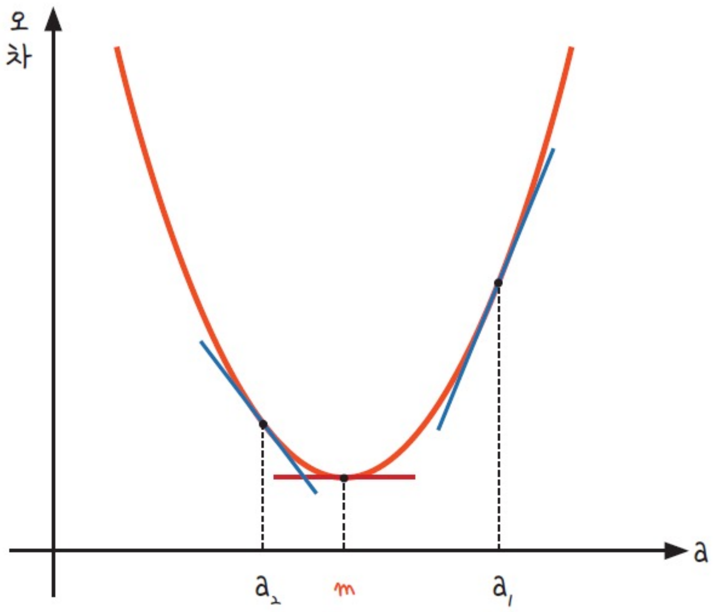
- 함수의 최솟값, 즉 **손실값을 최소화시키는 최적의 파라미터에서 미분값은 0**이어야 함

- 손실 함수를 파라미터에 대해 미분했을 때, 미분값이 0이 되는 방향으로 업데이트 방향을 결정



Gradient Descent (5)

- 경사하강법 슈도코드
  1. 현재 파라미터에서의 손실 함수에 대한 미분값을 구함
  2. 미분값의 반대 방향으로 파라미터값을 업데이트
  3. 미분값이 0이 될 때까지 1~2번을 에폭(Epoch)만큼 반복



Gradient Descent (6)

- 이번 실습에서는 경사하강법을 **Numpy**와 **Pytorch** 패키지를 각각 이용하여 구현해볼 예정
- 이를 통해, Pytorch 패키지의 장점들을 살펴볼 계획

Thank you

LG

KAIST AI
Graduate School of AI