Review

Gradient ≠ Gradient Descent

(미분값) (현재 위치에서의 미분값을 이용해 그래프의 최소점을 찾는 과정 혹은 기법)

Like, 김 ≠ 김밥 만들기

(그냥 김 자체) ("활용"해 음식 만들기)

Gradient Descent

(손실함수의 최저점을 찾는 가장 베이직한 방법)

100개의 샘플을 가진 input 가정

- Batch GD : 전체 샘플을 모두 훑고 100개 샘플의 평균 오차에 대해 1번 가중치를 업데이트 한다.
- Stochastic GD : 각각의 샘플에 대하여 바로 발생하는 오차만큼 100번 가중치 업데이트를 한다.
- Mini-batch GD : 사용자가 사전정의한 m을 배치 사이즈로 설정하여 m개의 입력마다 그 평균오차에 대해 (100 // m) + 1 번 가중치를 업데이트 한다.

빠르다? 느리다?

= 업데이트를 1번 하는데 걸리는 시간,

BGD SGD win!

1 epoch 에서 (모든 데이터를 한번씩 훑기까지)

총 몇번의 업데이트가 이루어 지는가?

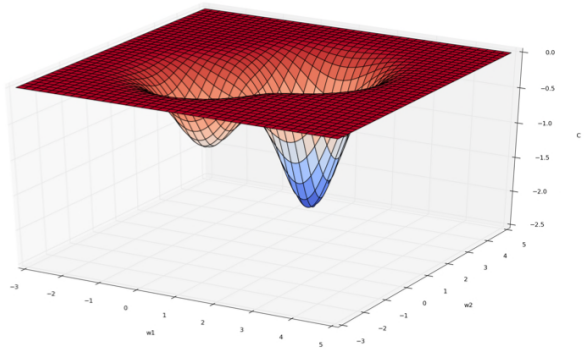
= m개를 한 묶음으로 생각할때 전체 데이터 샘플 수로부터 몇 묶음이 만들어지는가?

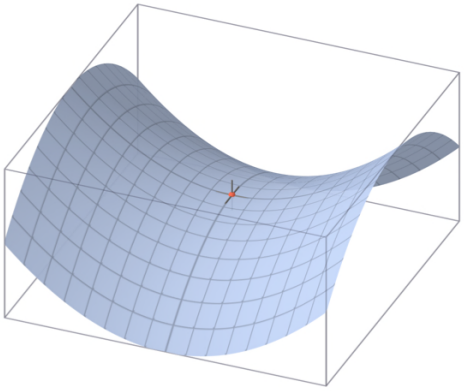
= iteration 수

ex) 100개 샘플, m = 32 ⇒ 4 iteration (나머지도 하나의 묶음으로)

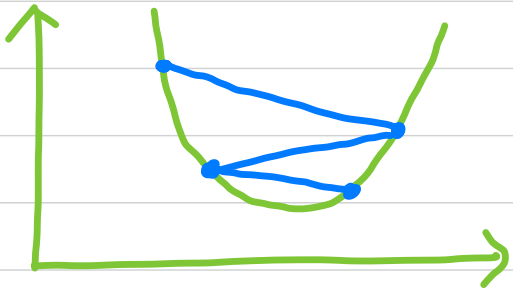
Basic opt idea → Advanced opt idea 로 넘어가기 전, 짚고 넘어가야 할 내용

Local minimum, Saddle point ??





↳ 왜 우리가 아는 예쁜 그림이 아닌 것인가 ?



1. Local minima

= MSE 만 생각해서 잘 상상이 가지 않았던 것.

우리는 최종 결과 $\hat{y}$ 를 얻기까지 여러 activation을 거치는데

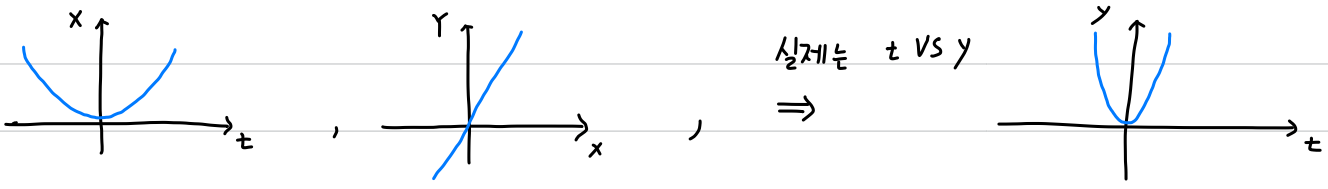
이런 동작이 손실 함수를 Non-Convex 함수로 만들어 버린다.

↳ 참고링크로 설명 : What is convex func ?, Derivative of Logistic Regression

2. Saddle point

= 그래프를 어떤것 VS. 어떤것 으로 그리냐에 따라 다른 것을 이해.

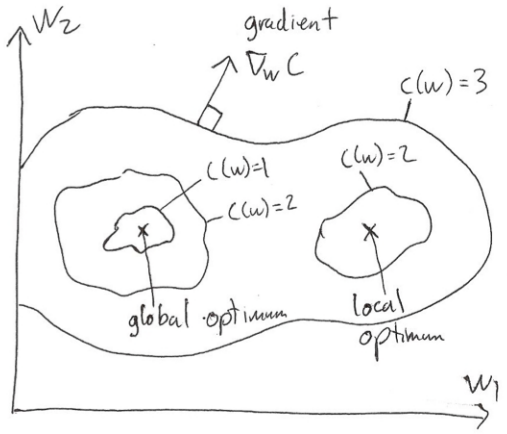
ex) $x = t^2$, $y = 3x$ 라면



⇓ Cost (y , $\hat{y}$) 로 나타낸 MSE 그래프는 2차원 이지 만

실제로 $\hat{y}$ 는 h (w_1 , w_2 , w_3 , ...) 아주 많은 변수로 이루어진 매개변수

참고링크 : Cost - function



Advanced opt idea 는 Basic 에서 무엇을 개선 했는가 ?

Basic) $w_{new} := w_{prev} - \alpha \cdot \nabla w$

2. 어느 방향으로 움직여야 하는지를 개선해보자 !

1. 얼만큼 적용할지를 개선해보자 !