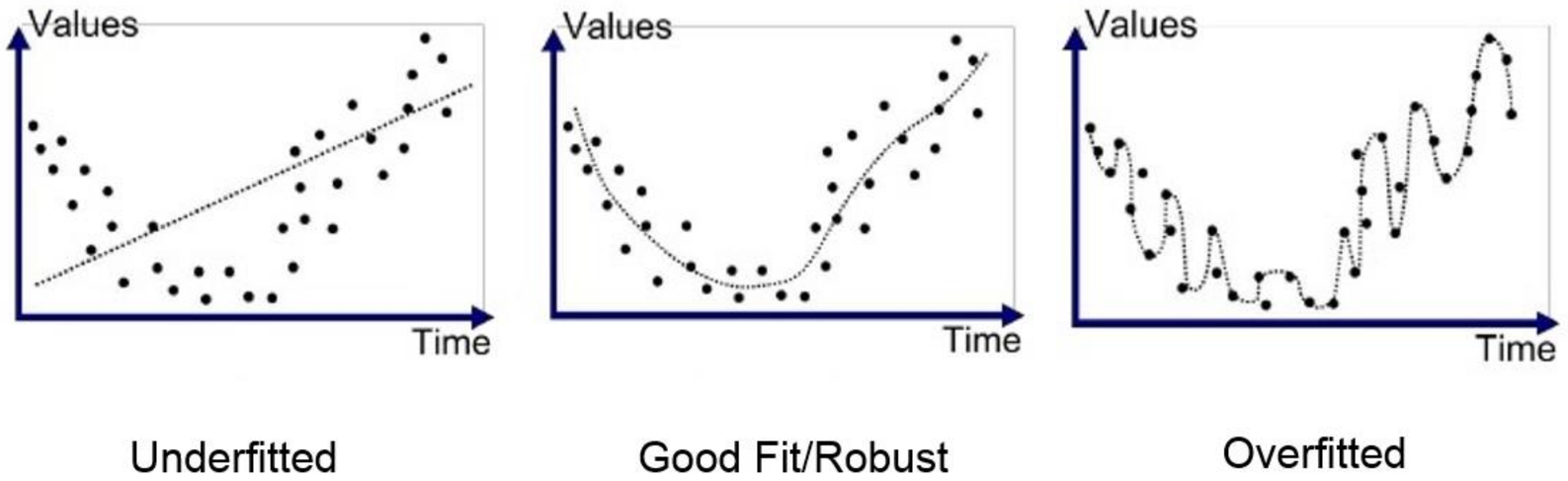
가중치 감소 (Weight decay)



훈련 데이터가 매우 많다면, Overfitting을 줄이는 것이 가능
하지만 현실적으로는 불가능, Overfitting을 줄이는 방법으로 사용
학습 과정에서 큰 가중치에 그에 상응하는 패널티를 부과하여
Overfitting을 방지하는 방법

L2 Regularization

$$Cost = \frac{1}{n}\sum_{i=1}^{n}\{L(y_i, \hat{y}_i) + \frac{\lambda}{2}|w|^2\}$$

가중치의 곱이 아닌 가중치의 합을 더한 규제 강도 (λ)를 곱함
λ를 크게 (규제 중시), 작게 (규제를 중요시 X)
가중치가 크면 손실함수가 커지고, 다음 가중치가 크게 감소

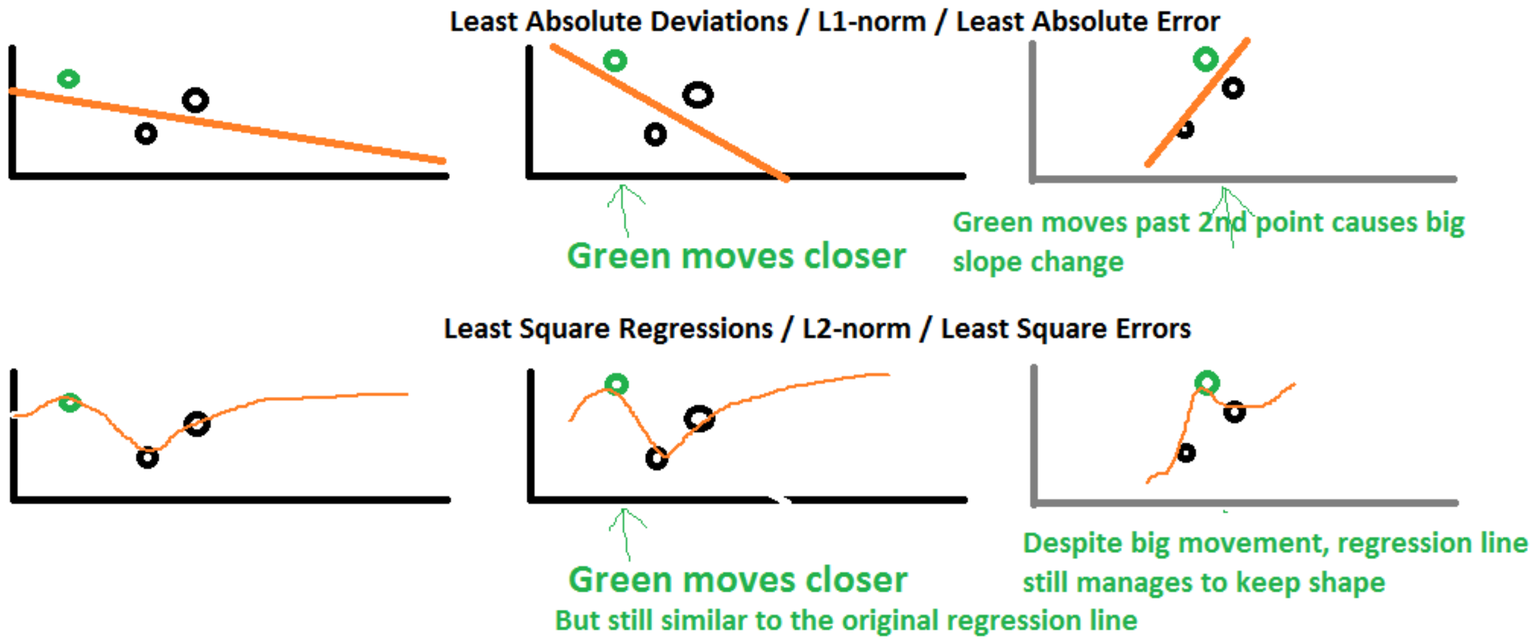
L1 Regularization

$$Cost = \frac{1}{n}\sum_{i=1}^{n}\{L(y_i, \hat{y}_i) + \frac{\lambda}{2}|w|\}$$

$L(y_i, \hat{y}_i)$: 기존의 Cost function

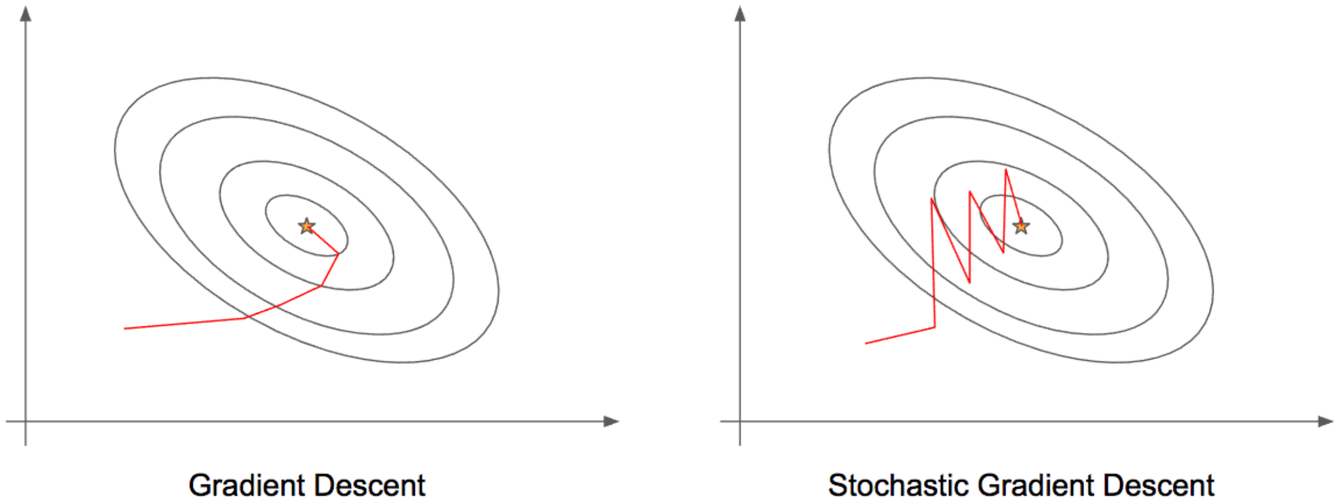
가중치의 곱이 아닌 가중치의 합을 더한 규제 강도 (λ)를 곱함
가중치의 크기가 포함, 가중치가 너무 크기 않은 방향으로 학습

L1 vs L2



L1은 더 안정적으로 아웃라이어에 영향을 덜 받는 경향
L2는 아웃라이어 값에 영향을 많이 받음 (아웃라이어에 민감)
L2가 L1에 비해 더 안정적이라 일반적으로 L2가 더 많이 사용됨

GD vs SGD

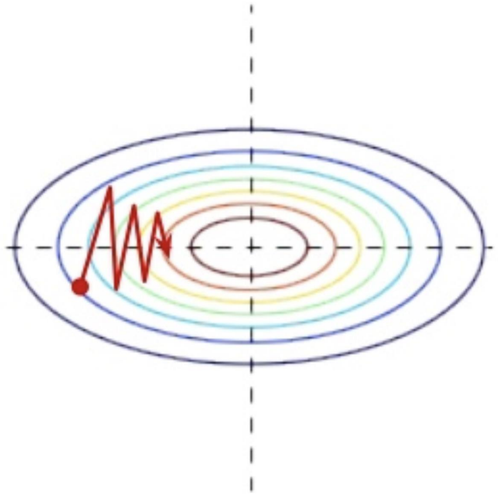


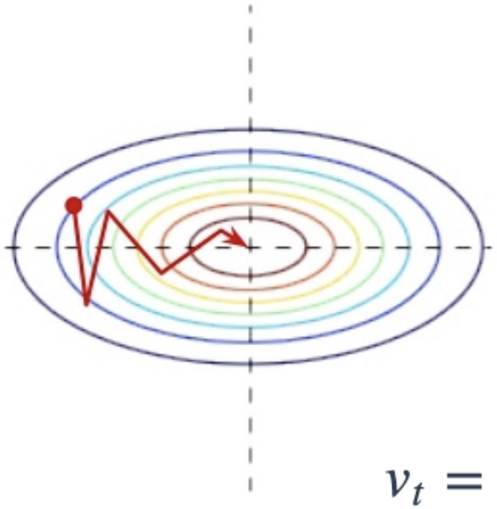
GD: 1 epoch 동안 모든 트레이닝 셋에 대한 gradient sum
SGD는 샘플 단위로 gradient를 업데이트

Randomly choose mini-batch

수능 언어영역 시험을 볼 때 GD는 긴 지문을 다 읽고 풀고 SGD는 필요한 부분만 골라 문제를 푸는 방법

Momentum



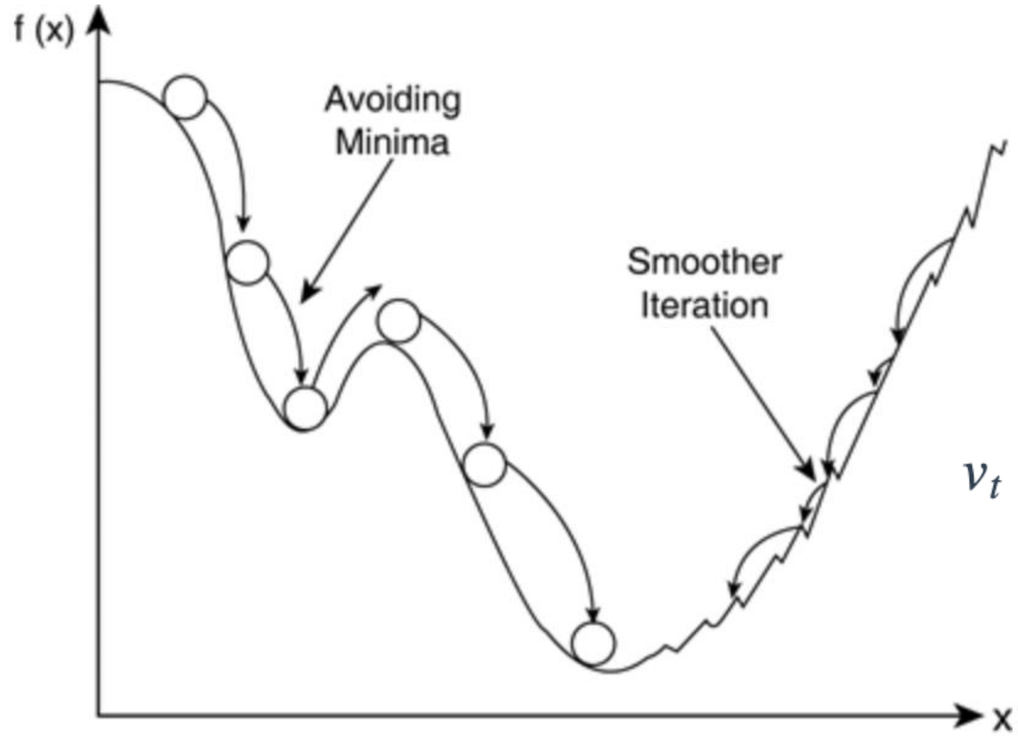


$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

$$\theta = \theta - v_t$$

Gradient Descent를 통해 이동하는 과정에서 '관성'을 주는 방법
과거에 이동했던 방식을 기억하면서 그 방향으로 일정 정도를
추가적으로 이동하는 방식

Momentum

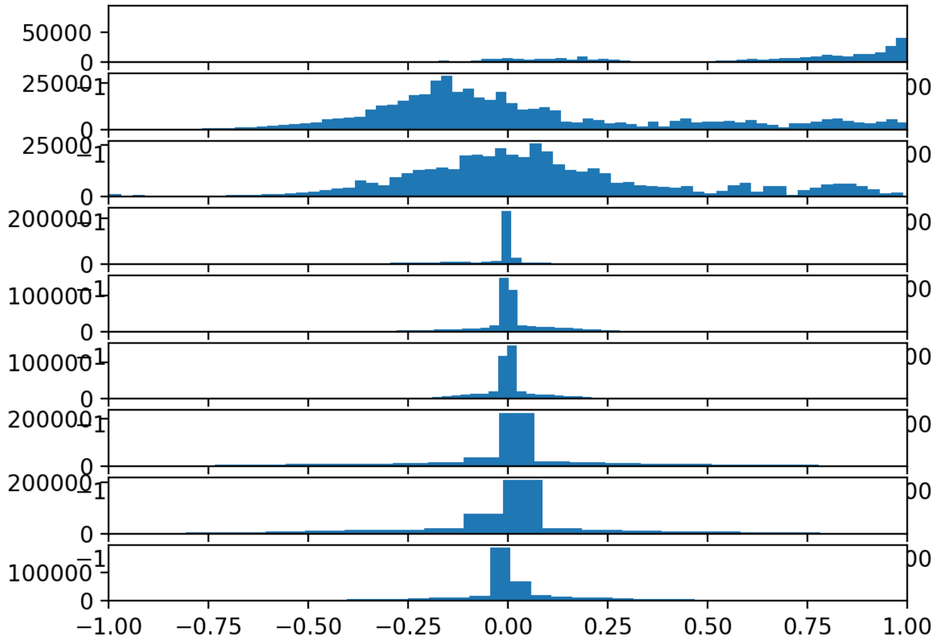


$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

$$\theta = \theta - v_t$$

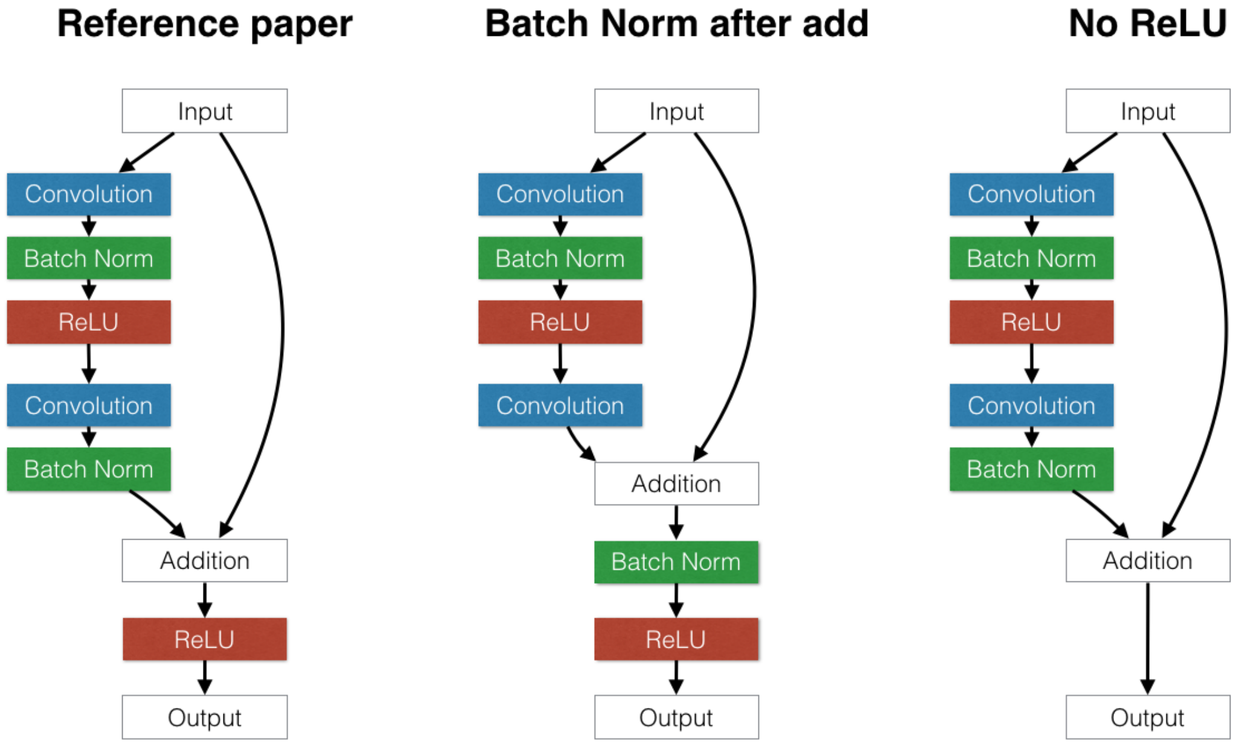
Momentum은 Local minima를 빠져나오는 효과가 있음
기존의 GD/SGD에서는 Local minima에 빠지면
Gradient가 0이 되어 이동 불가능
관성이 존재하기 때문에 더 좋은 minima로 이동할 수 있음

Batch Normalization



Internal Covariance Shift: 네트워크의 각 레이어나 활성화 함수 마다 Input의 분포가 달라지는 현상

Batch Normalization



Batch Normalization은 평균과 분산을 조절하는 과정이 신경망 안에 포함된 과정