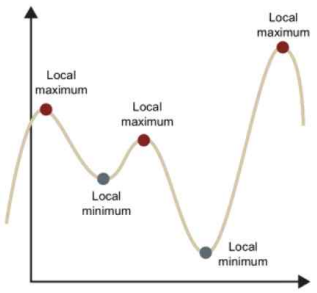
mlads 6주차 정리

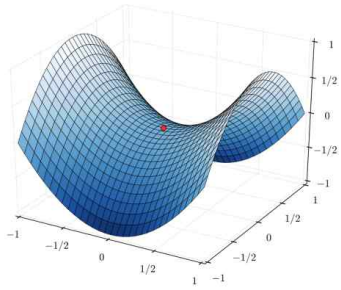
1. gradient
: 스칼라장의 최대의 증가율을 보여주는 벡터장

2. gradient descent(경사 하강법)
: gradient를 계산해 minima를 찾는 방법으로 gradient를 계산하고 그 반대 방향으로 간다.

$$w := w - \alpha \frac{\partial C}{\partial w}$$

local minnimum, saddle point(미분계수가 0이지만 극대 극소가 아님)에 빠질 가능성이 있다.

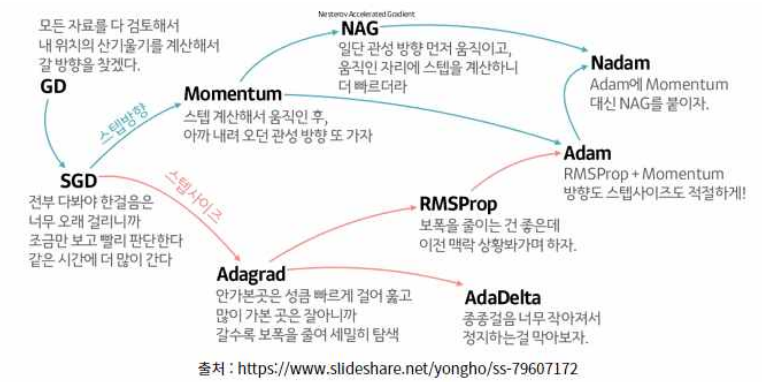




3. GD, SGD와 mini-batch GD 세가지가 있다.
첫 번째는 모든 점 이용
두 번째는 점 하나만 이용, 단 noise가 클 수 있다.
세 번째는 점 일부만 이용

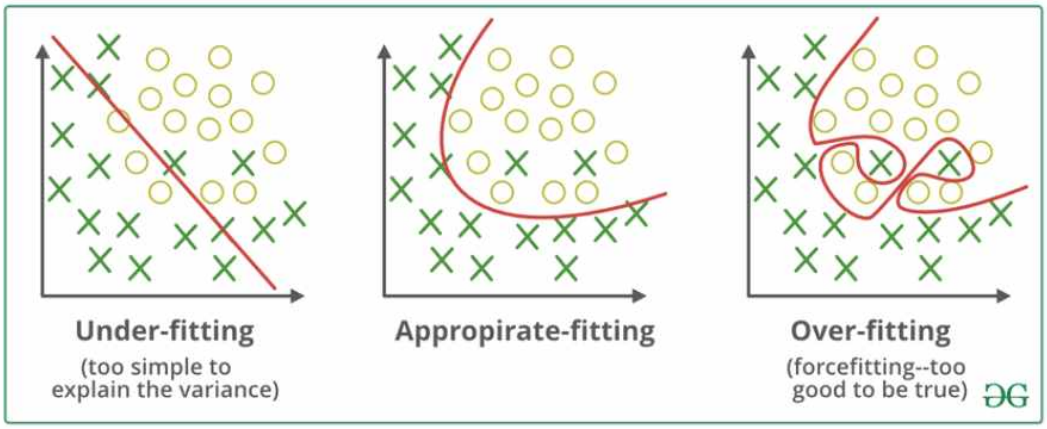
난 이렇게 배웠지만 사람마다 다르게 쓰이기도 한다.

4. 최적화 방법들



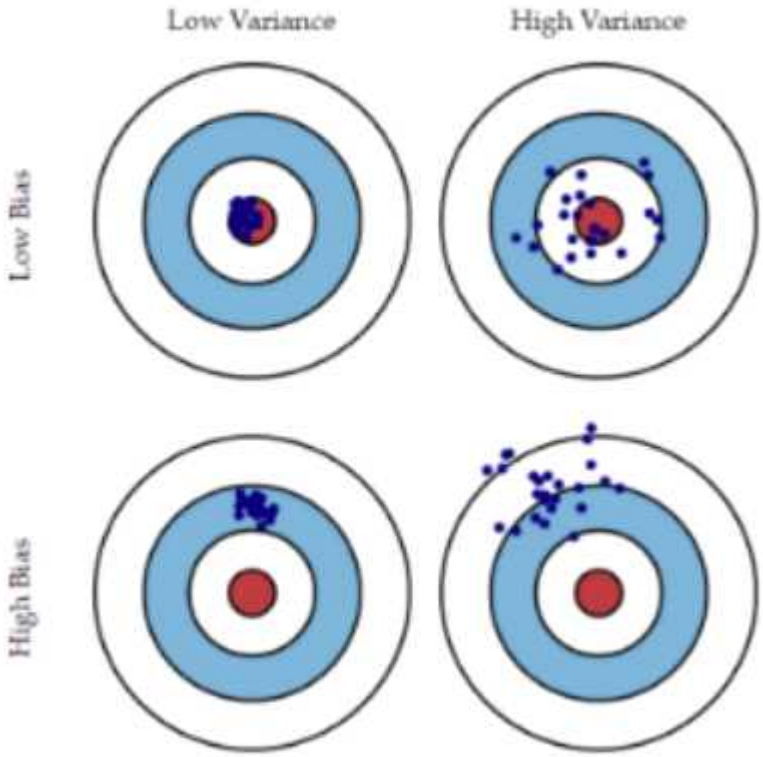
출처 : https://www.slideshare.net/yongho/ss-79607172

5. 과적합과 과소적합

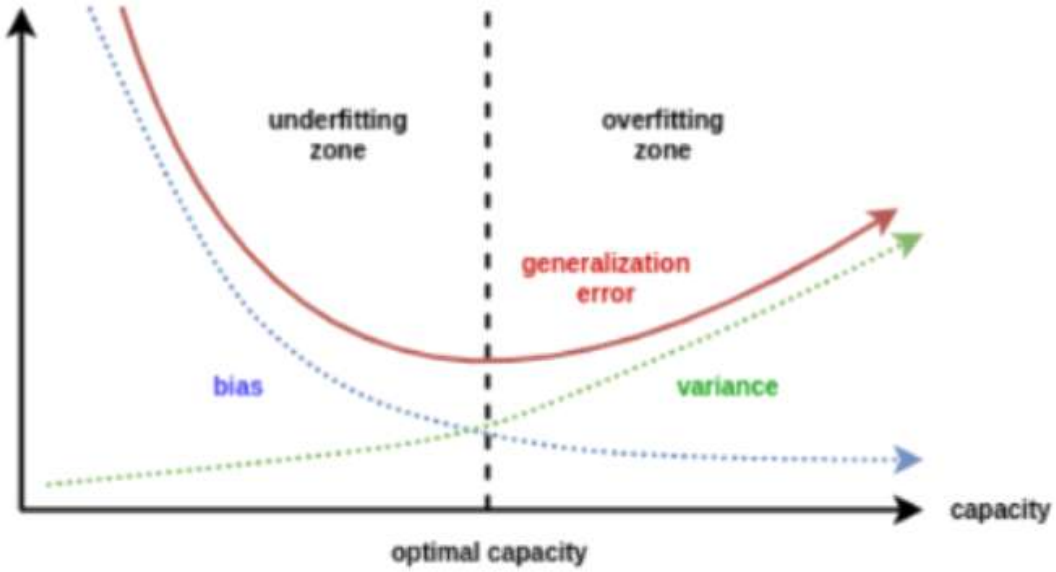


과적합은 데이터의 bias까지 너무 학습해서 일반화가 어려운 상태를 말함.
과소적합은 모델의 성능이 training 단계부터 별로인걸 말함

6. bias와 variance



7. bias-variance trade-off



8. 과적합 막기.

(1) Regularization: term을 추가해 오버피팅 막음

$$L(\mathbf{w}) + \lambda R(\mathbf{w})$$

This is called the regularization term or regularizer or penalty. The squared L2 norm is one kind of penalty, but there are others

λ is called the regularization strength.

Other common names for λ: *alpha* in sklearn, *C* in many algorithms. Usually C actually refers to the inverse regularization strength, 1/λ. Figure out which one your implementation is using (whether this will increase or decrease regularization)

- **Ridge**: $R(w) = \|w\|^2 = w_1^2 + \cdots + w_n^2$
- **Lasso**: $R(w) = \|w\|_1 = |w_1| + \cdots + |w_n|$
- **Elastic Net**: $R(w) = \lambda_2\|w\|^2 + \lambda_1\|w\|_1$

(2) Dropout

DNN에서 링크 몇 개를 랜덤으로 끊기

(3) Batch-normalization

배치 정규화 논문에서는 학습에서 불안정화가 일어나는 이유를 'Internal Covariance Shift' 라고 주장하고 있는데, 이는 네트워크의 각 레이어나 Activation 마다 입력값의 분산이 달라지는 현상을 뜻한다.

이를 해결하기 위해 배치 정규화는 평균과 분산을 조정하는 과정이 별도의 과정으로 떼지 말고 신경망 안에 포함되어 학습 시 평균과 분산을 조정한다.