

 x 

The main title of the slide, featuring the CODINGO logo on the left, a large black "x" in the center, and the POSCO logo on the right. The POSCO logo is in a blue, lowercase, sans-serif font.

 스마트 팩토리 3기

The subtitle of the slide, featuring the "K-Digital Training" logo on the left and the Korean text "스마트 팩토리 3기" (Smart Factory 3rd Generation) on the right. The "K-Digital Training" logo has "K" in green, "Digital" in blue, and "Training" in purple.

회귀(Regression)

회귀(Regression)

지도학습
정답이 있음

- 회귀는 머신러닝에서 주로 사용되는 통계 도구 중 하나.
- 회귀는 데이터의 특성과 관측된 연속 값 응답 간의 관계를 학습하여 여러분의 데이터에서 예측할 수 있다.
- 회귀분석의 개념은 주어진 피처와 결정 값 데이터 기반에서 학습을 통해 최적의 회귀 계수를 찾아내는 것.
- $Y = W1 * X1 + W2 * X2$
 - 위의 식에서 $W1, W2$ 가 회귀 계수

회귀(Regression)

- 회귀 모델의 종류
 - 선형 회귀, 비선형 회귀, 단일 회귀, 다중 회귀 등
 - 이 중 선형 회귀는 실제 값과 예측값의 차이를 최소화하는 직선형 회귀 선을 최적화하는 방식으로 가장 많이 사용됨.
- 회귀 모델의 성능 평가 지표
 - MAE, MSE, RMSE, R^2 등
 - 이들 지표들은 실제값과 예측값의 차이를 기반으로한 지표가 중심.

선형 회귀(Linear Regression)

- 선형 회귀는 종속 변수 y 와 하나 이상의 독립 변수 x 와의 선형 상관관계를 모델링하는 기법이다.
- 만약 독립 변수 x 가 1개라면 단순 선형 회귀라고 하고, 2개 이상이면 다중 선형 회귀라고 한다.

선형 회귀(Linear Regression)

- **단순 선형 회귀**

- $y=Wx+b$ W : 기울기, b : y절편
- W 를 가중치(Weight), b 를 편향(bias)라고 부른다
- 그래프의 형태는 **직선**

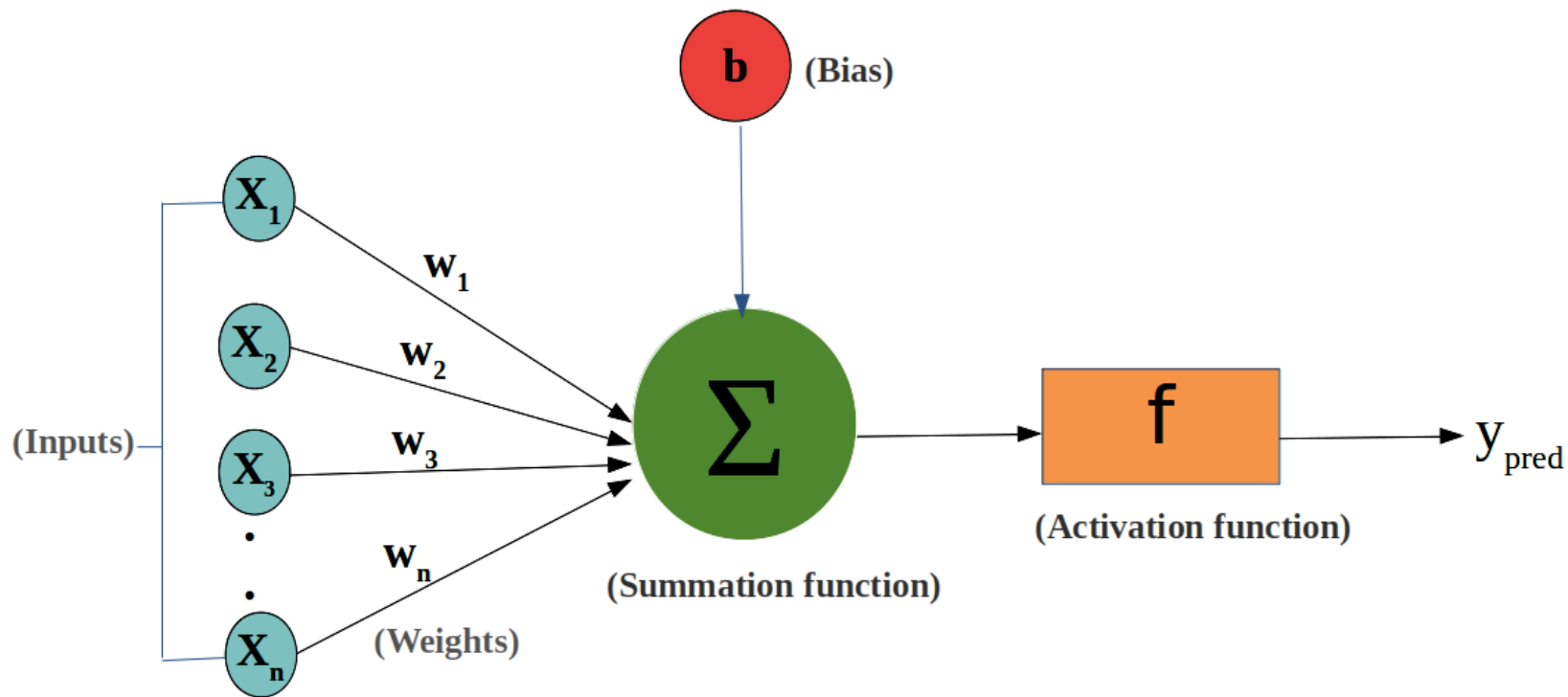
- **다중 선형 회귀**

- $y=W_1x_1 + W_2x_2 + \dots + W_nx_n + b$
- 만약 2개의 독립 변수면 그래프는 **평면**으로 나타남

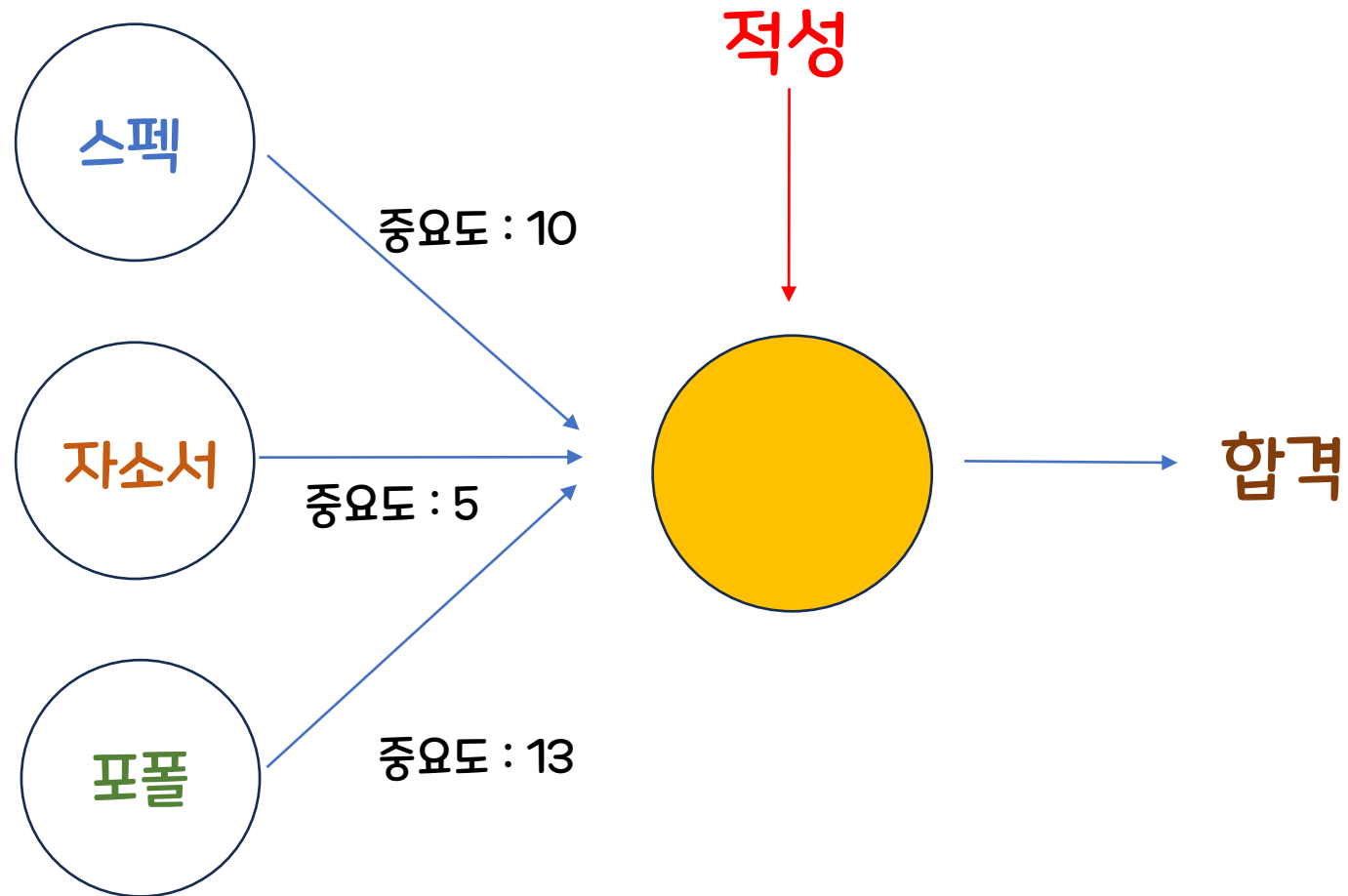
가중치와 편향

- **가중치(Weight)** W(가중치)를 구하는 것이 핵심
 - 입력 데이터의 중요성을 나타내는 매개변수
 - 가중치가 높을수록 해당 뉴런이 중요하다는 것을 뜻한다
- **편향(bias)**
 - 뉴런의 활성화 여부를 결정
 - 성향

가중치와 편향

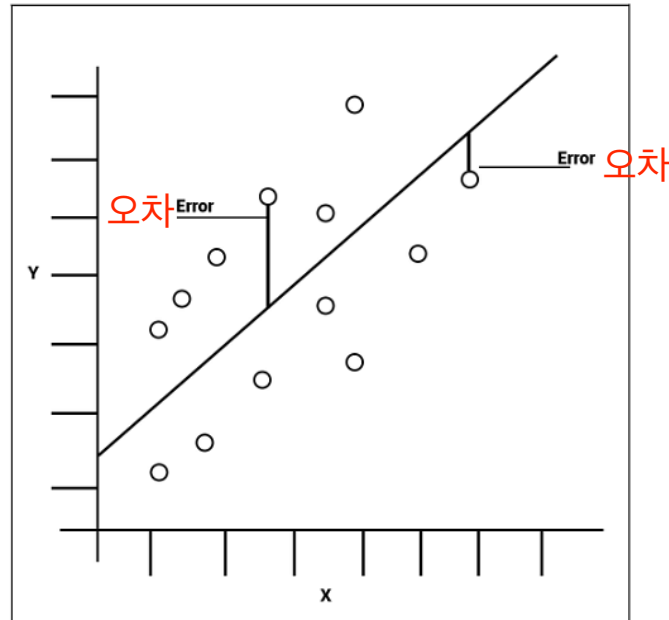


가중치와 편향



선형 회귀(Linear Regression)

- 선형회귀는 $y=Wx+b$ 방정식의 해를 찾는 과정
- 임의의 W 와 b 를 설정하고 오차를 줄이도록 학습
- 오차를 최소화하는 W 와 b 를 찾는것이 목표



오차(error)

- 0에 가까울 수록 원본에 가까운 것 (정확도가 높다)
- 아래의 함수들을 **Cost function**, **Loss function** 으로 부른다.

- **MSE(Mean Squared Error)** 모든 오차를 합한다

- 평균 제곱오차
- 오차의 제곱을 평균으로 나눈 것

$$MSE = \frac{\sum_{i=1}^n (y - \hat{y})^2}{n}$$

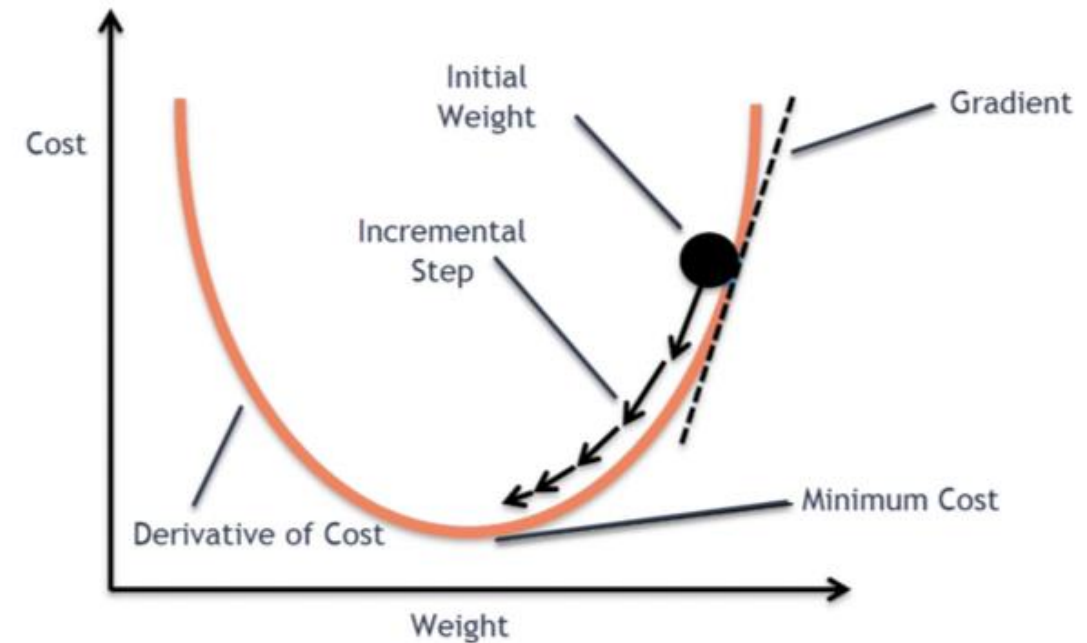
- **MAE(Mean Absolute Error)**

- 평균 절대 오차
- 오차의 차이를 절대값으로 변환한 뒤 합산

$$MAE = \frac{\sum |y - \hat{y}|}{n}$$

경사 하강법(Gradient Descent)

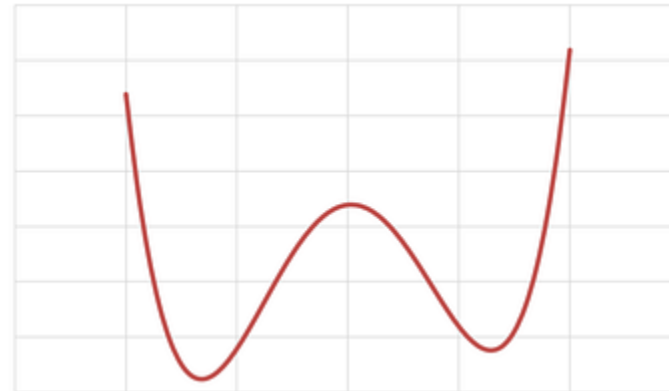
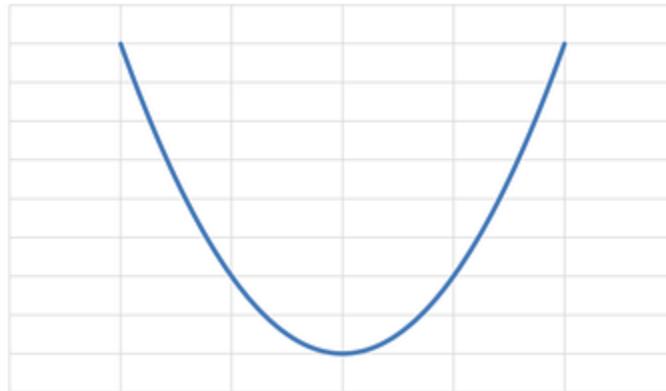
- 1차 근삿값 발견용 최적화 알고리즘
- 함수의 기울기(경사)를 구하고 기울기의 반대 방향으로 계속 이동시켜서 최소값에 이를 때까지 반복시키는 것



출처 : <https://vitalflux.com/gradient-descent-explained-simply-with-examples/>

볼록함수(convex function)

- 아래로 볼록한 함수
- 함수 위의 임의의 두 점을 연결하는 선을 그래프에 그었을때, 그 선이 그래프의 위쪽만을 지나가는 함수
- 반대는 Non-Convex function
- 경사하강법은 Convex function 에서 잘 작동한다.



경사 하강법(Gradient Descent)

- BGD(Batch Gradient Descent, BGD)
 - 전체 데이터 셋에 대한 에러를 구한 뒤 기울기를 한번만 계산하여 모델을 업데이트
 - Optimal 로 안정적으로 수렴
 - 메모리가 많이 필요, 학습이 오래 걸림
 - Local optimal 에 빠지기 쉬움

경사 하강법(Gradient Descent)

가장 많이 사용함

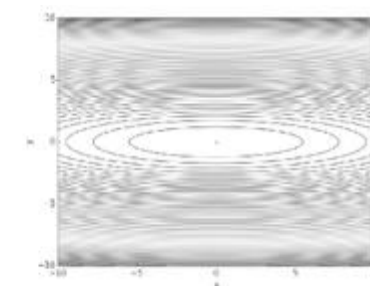
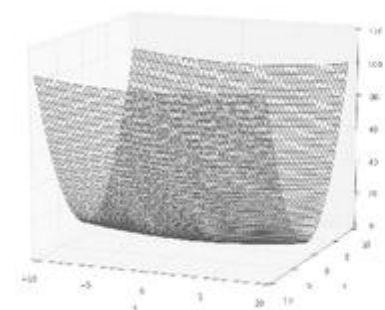
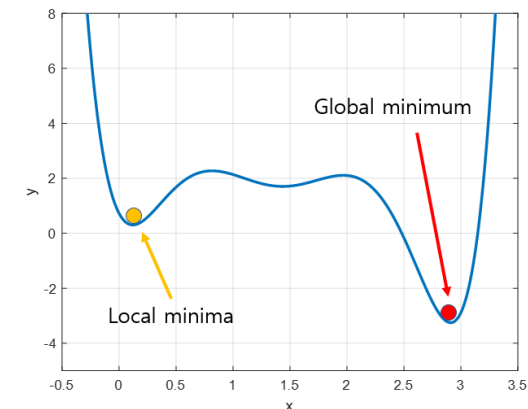
- 확률적 경사 하강법(Stochastic Gradient Descent, SGD)
 - 임의로 선택한 데이터에 대해 기울기를 계산하는 방법
 - 적은 데이터를 사용하므로 BGD보다 계산이 빠름
 - 정확도는 낮을 수 있음

경사 하강법(Gradient Descent)

- 미니 배치 경사 하강법(Mini-batch Gradient Descent, MGD)
 - 전체 데이터셋을 미니 배치 여러 개로 나눈 뒤, 미니 배치 한 개마다 기울기를 구함
 - 그 평균 기울기를 이용하여 모델을 업데이트
 - 계산이 빠름
 - SGD보다 안정적

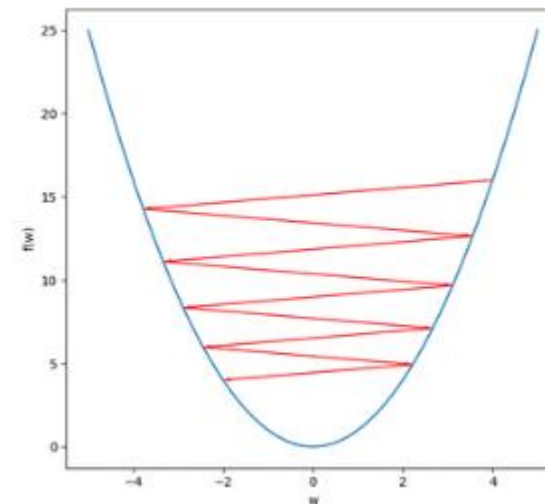
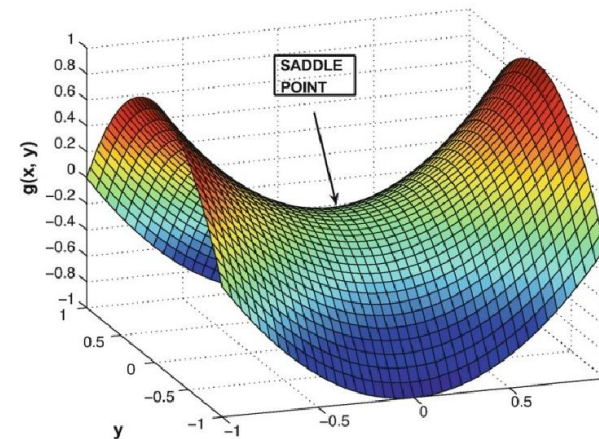
경사 하강법의 문제

- Local minimum
 - 전역 최소값이 있지만 지역 최솟값에 안착하는 문제
- 비등방성 함수
 - 기울기가 가리키는 지점이 하나가 아니라 여러 가지
 - 지그재그 형태가 발생해 학습률의 성장이 낮아짐



경사 하강법의 문제

- Saddle point(안장점)
 - 최댓값인지 최솟값인지 구분할 수 없음
- Overshooting
 - 학습률이 너무 클 경우에 minima를 찾지 못하고 발산하는 문제



SGD의 단점을 개선(optimizer)

- 학습률(Learning rate)
 - 최적화 할 때의 속도를 조절하는 parameter
 - Step size라고도 함
 - 너무 작으면 시간이 오래 걸리고 너무 크면 학습이 제대로 이루어지지 않음

SGD의 단점을 개선(optimizer)

- 학습률(Learning rate)
 - 최적화 기법
 - learning rate decay
 - 초반에는 큰 폭으로 이동하여 빠르게 내려가고, 점차 학습률을 줄여서 조심스럽게 내려가는 방법
 - Adaptive learning rate
 - 경사가 가파르면 조금 이동하고 완만하면 크게 이동
 - Momentum
 - 가던 길로 조금 더 가는 방법

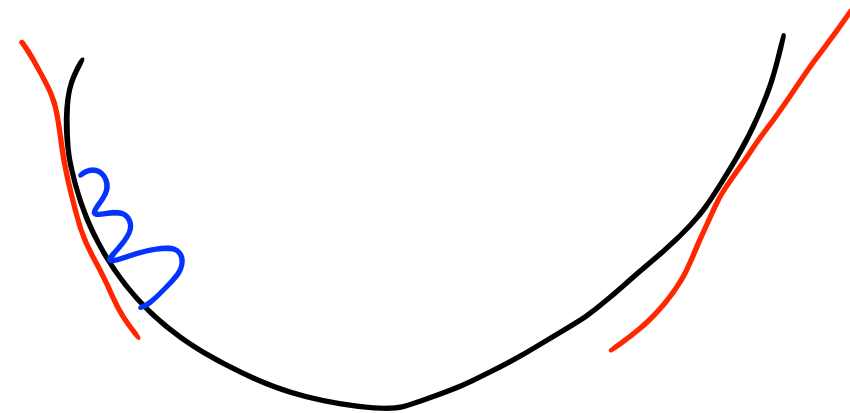
Momentum

- 진행하던 속도에 관성을 주는 방법 (경사진 곳에서 돌을 굴리면 계속해서 아래로 굴러가려는 성질)
- 느린 속도, saddle point, 진동이 심한 점 등을 해결한 알고리즘

$$v \leftarrow \alpha v - \eta \frac{\partial L}{\partial W}$$

$$W \leftarrow W + v$$

학습률 η
 기울기가 마이너스 일 때는 + 쪽으로 이동(\)
 기울기가 플러스 일 때는 - 쪽으로 이동(/)
 기울기하기 위하여 -를 붙여줌



AdaGrad(Adaptive Gradient)

- Learning rate decay 를 적용 한 방법
- 기울기의 제곱을 누적해서 더한후에 학습률을 조정 하는 방법

$$h \leftarrow h + \frac{\partial L}{\partial W} \odot \frac{\partial L}{\partial W}$$
$$W \leftarrow W - \eta \frac{1}{\sqrt{h}} \frac{\partial L}{\partial W}$$

- 학습률이 너무 작아져 학습이 되지 않는 문제

RSMPProp

- 지수 가중 이동 평균을 이용
- AdaGrad에서 학습이 안 되는 문제를 해결
- Hyper parameter인 ρ 가 추가됨(보통 0.9사용)
- 최근 값은 크게 반영, 오래된 값은 작게 반영

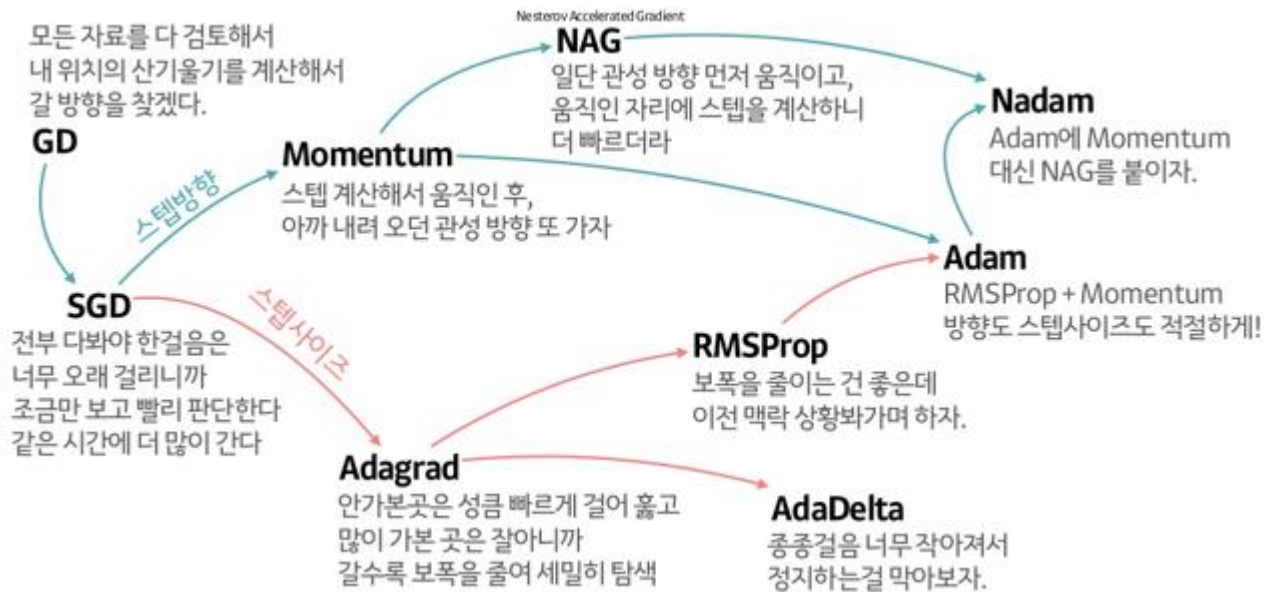
$$h_i \leftarrow \rho h_{i-1} + (1 - \rho) \frac{\partial L_i}{\partial W} \odot \frac{\partial L_i}{\partial W}$$

Adam

- Momentum, RMSProp를 융합한 방법
- 최근에는 Adam을 개량한 AdamW, Nadam, RAdam등도 사용

Optimizer

- <https://keras.io/api/optimizers/>
- <https://pytorch.org/docs/stable/optim.html>

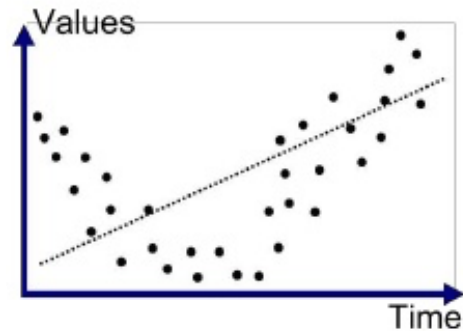
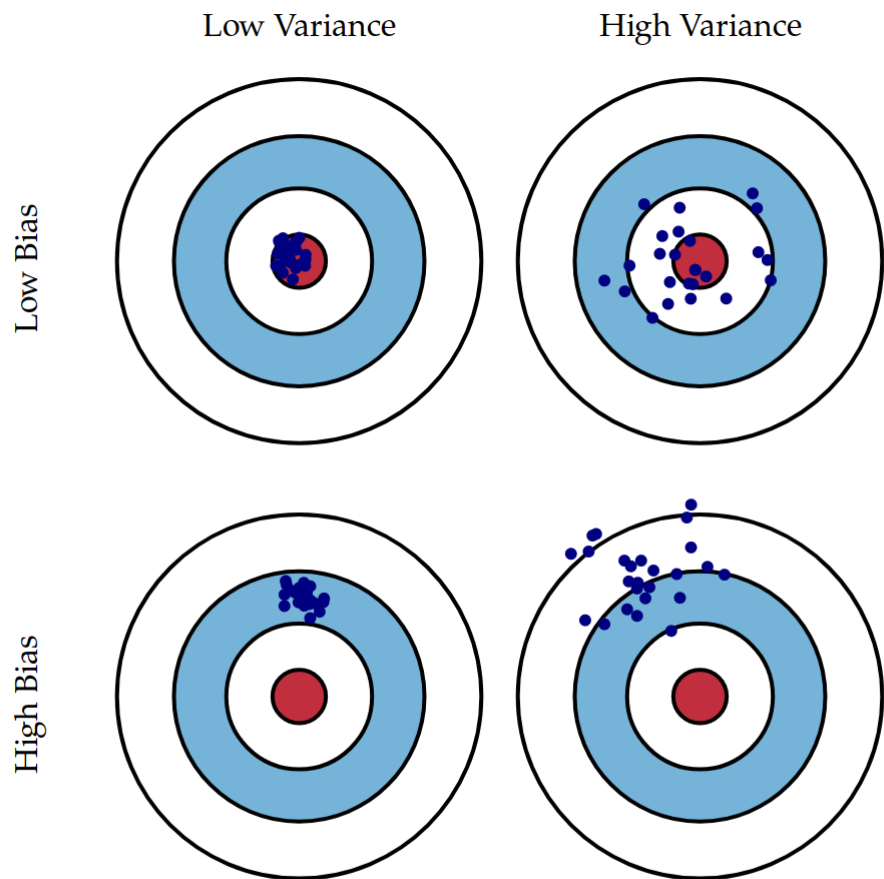


출처: <https://www.slideshare.net/yongho/ss-79607172>

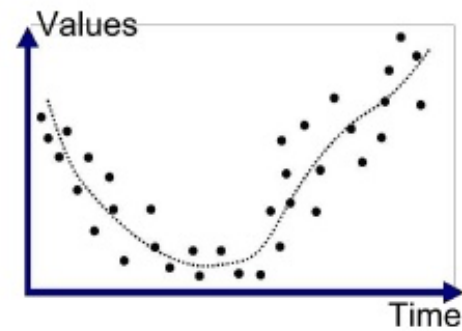
편향과 분산

- **편향(bias)**
 - 실제값에 대한 추정값의 오차
- **분산(Variance)**
 - 데이터들이 퍼져있는 정도

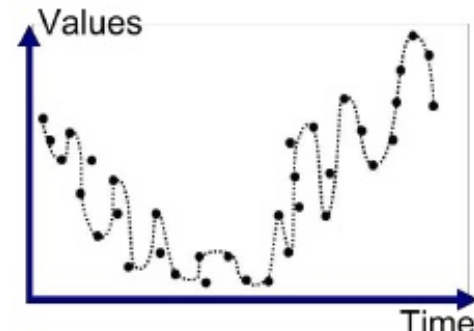
편향과 분산



high bias
low variance



medium bias
medium variance



low bias
high variance

Fig. 1 Graphical illustration of bias and variance.

편향과 분산

