# Deep Learning 기초 3

## (딥러닝 학습의 효율과 정확도를 높이는 기술들 1)

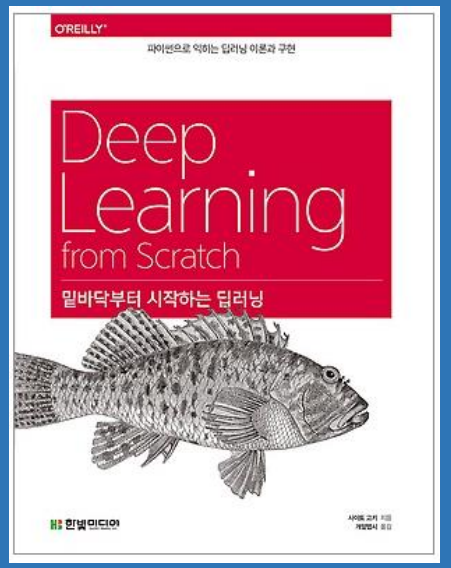
임세진

https://youtu.be/meOdnvGsqj0

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

# Contents




CryptoCraft LAB

# 01. 매개변수 갱신

- Optimization (최적화)

  : 손실 함수의 값을 최대한 낮추는 매개변수를 찾는 것

- **매개변수의 최적값을 찾는 법** (기준으로 삼는 단서에 따라 분류)

  ✓ 확률적 경사 하강법 (SGD)

  ✓ 모멘텀

  ✓ AdaGrad

  ✓ Adam

# 01. 매개변수 갱신

- **확률적 경사 하강법 (SGD)**

  : 매개변수의 기울기(미분) ➔ 기울어진 방향으로 매개변수 값 갱신

<모험가 이야기>

✓ 깊은 골짜기를 찾는 것이 목적

✓ 지도를 보지 않고, 눈가리개를 써서 찾기

**현재 서있는 곳에서**

**가장 크게 기울어진 방향으로 가자**

[이타] 학습률 (보통 0.01, 0.001 사용)

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial L}{\partial \mathbf{W}}$$

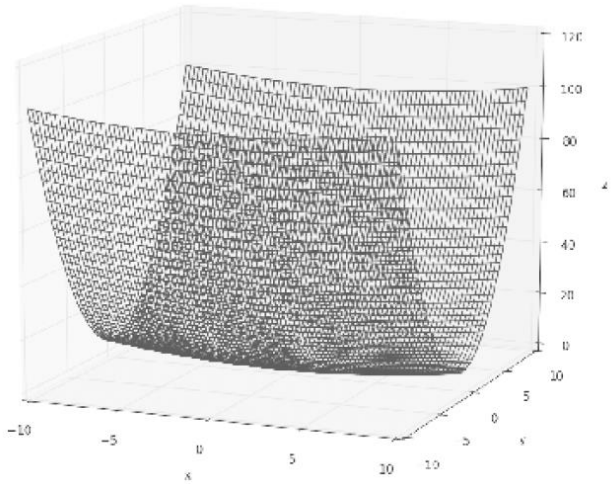
갱신할 가중치 매개변수

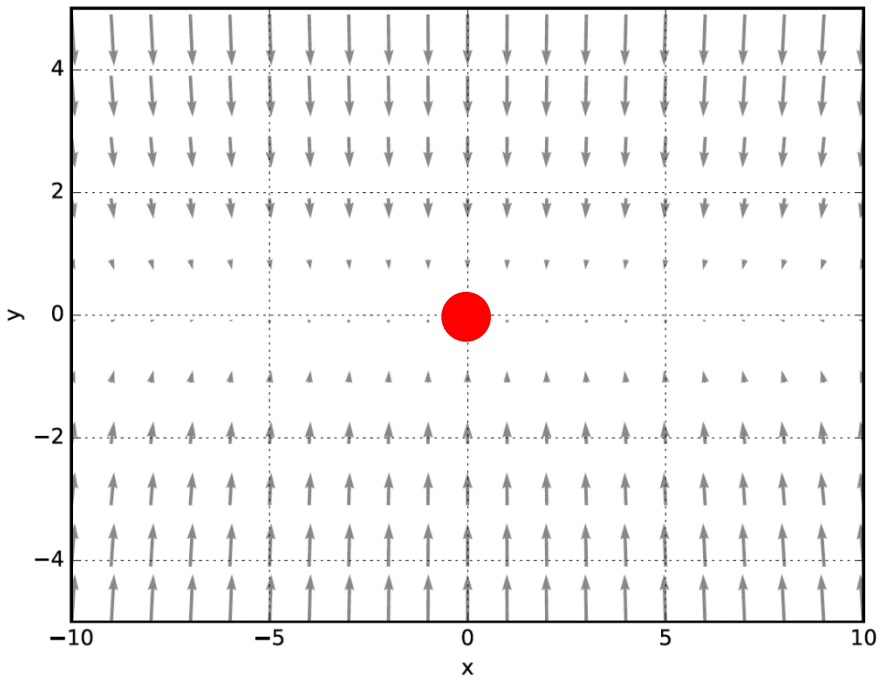
W에 대한 손실 함수의 기울기

# 01. 매개변수 갱신

- SGD의 단점

함수 예 : $f(x,y) = \frac{1}{20}x^2 + y^2$





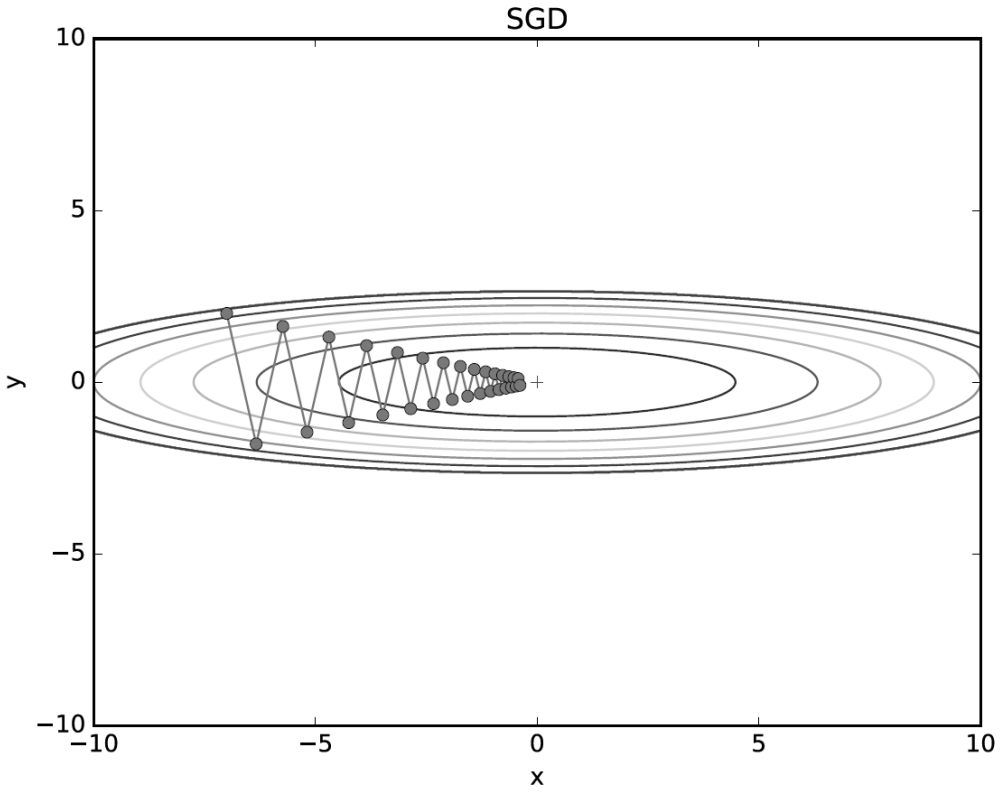

최솟값은 (0, 0) 인데

그림의 기울기는 대부분 (0, 0) 방향을 가리키지 않음

# 01. 매개변수 갱신

- SGD의 단점 : 비등방성(anisotropy) 함수에서 탐색 경로가 비효율적임

  ↳ 방향에 따라 성질(기울기)이 달라지는 함수




앞의 함수에 SGD를 적용한 결과

**심하게 굽이진 움직임을 보여줌**

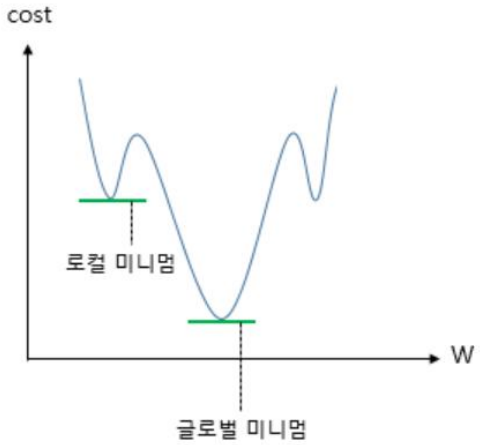
**(기울어진 방향이 본래의 최솟값과 다른 방향을 가리키기 때문)**

**➔상당히 비효율적임**

# 01. 매개변수 갱신

- **모멘텀 (Momentum)**

- '운동량'을 뜻하는 단어로 물리와 관련 있음
- 기울기 값만 반영하는 SGD 개선 ( 기존의 SGD에 관성을 더해줌 ➔ 수렴속도 개선)
- 관성의 법칙 이용 (이전 물체의 속도는 현재 물체의 속도와 관련 있음)




지면 마찰, 공기 저항의 역할

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

$$\mathbf{W} \leftarrow \mathbf{W} + \mathbf{v}$$

속도

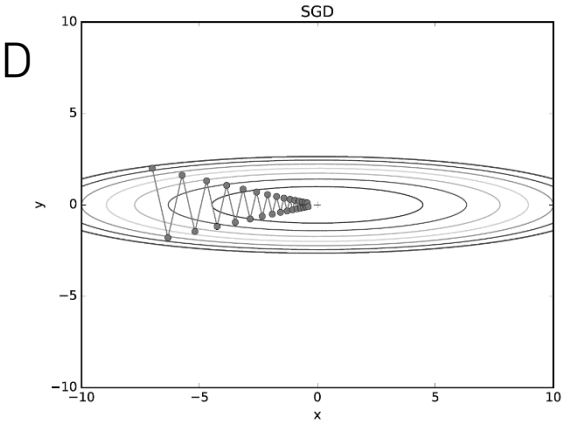
**기울기 방향으로 힘을 받아 물체가 가속되는 물리법칙 표현**

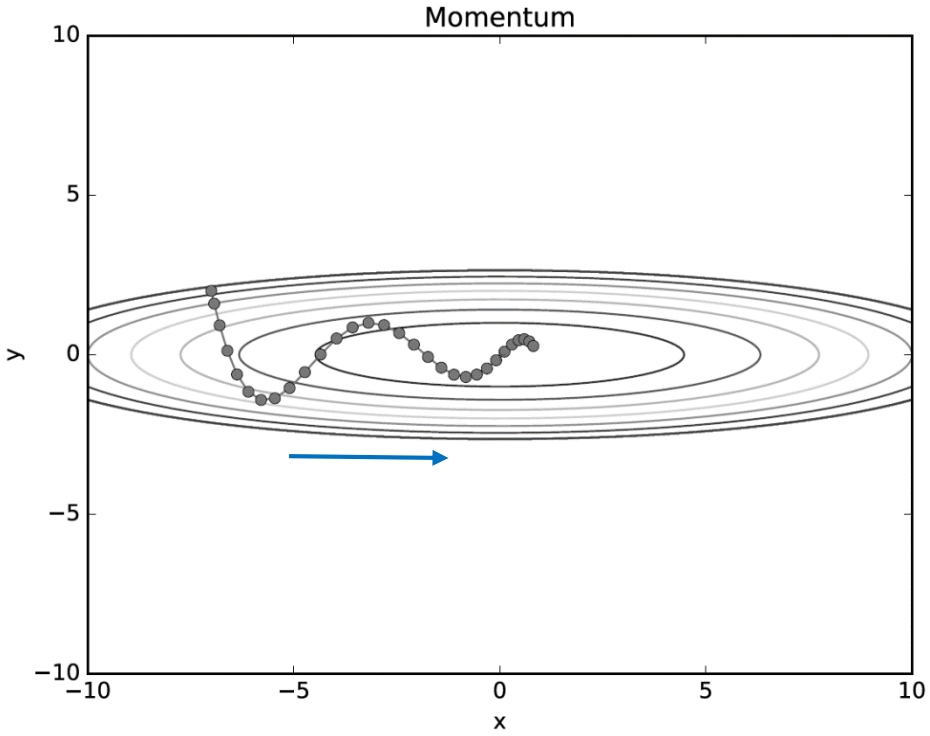
**(속도가 클수록 기울기가 크게 update 됨)**

# 01. 매개변수 갱신

- 모멘텀 (Momentum)

SGD




✓ 속도는 방향성을 포함

✓ X축의 힘은 아주 작지만 방향이 바뀌지 않아서 한 방향으로 일정하게 가속

➔ SGD보다 X축 방향으로 빠르게 다가갈 수 있음

✓ Y축의 힘은 크지만 방향이 계속 바뀌어서 상충하므로 속도가 일정하지 않음

# 01. 매개변수 갱신

- 신경망 학습에서 학습률 값은 ★ 중요★

이 값이 너무 작으면 학습시간이 길어짐
너무 크면 발산하여 학습이 제대로 X

- 학습률을 정하는 효과적 기술

  학습률 감소 (learning rate decay) : 학습을 진행하면서 학습률을 점차 줄여가는 방법

## • Adagrad

- 각각의 매개변수에 맞춤형 값

  개별 매개변수에 적응적으로 학습률을 조정하며 학습 진행

- 과거의 기울기를 제곱하여 계속 더해감

  ➔ 학습을 진행할수록 갱신 강도 약해지는 단점

  ➔ RMSProp 으로 개선

  먼 기울기는 서서히 잊고
  새로운 기울기 정보는 크게 반영

기울기의 값을 제곱하여 계속 더해줌

$$\mathbf{h} \leftarrow \mathbf{h} + \frac{\partial L}{\partial \mathbf{W}} \odot \frac{\partial L}{\partial \mathbf{W}}$$

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

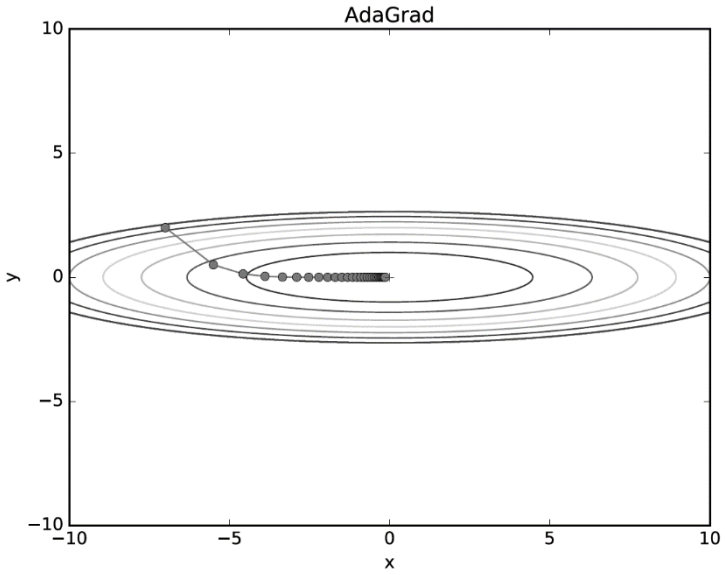
학습률 조정

**매개변수의 원소 중 많이 움직인(갱신된) 원소는 학습률이 낮아짐**

**➔ 학습률 감소가 매개변수의 원소마다 다르게 적용**

# 01. 매개변수 갱신

- Adagrad




✓ 최솟값을 향해 효율적으로 움직임

✓ Y축 방향은 기울기가 커서 처음에는 크게 움직이지만 Y축 방향으로 갱신 강도가 빠르게 약해짐

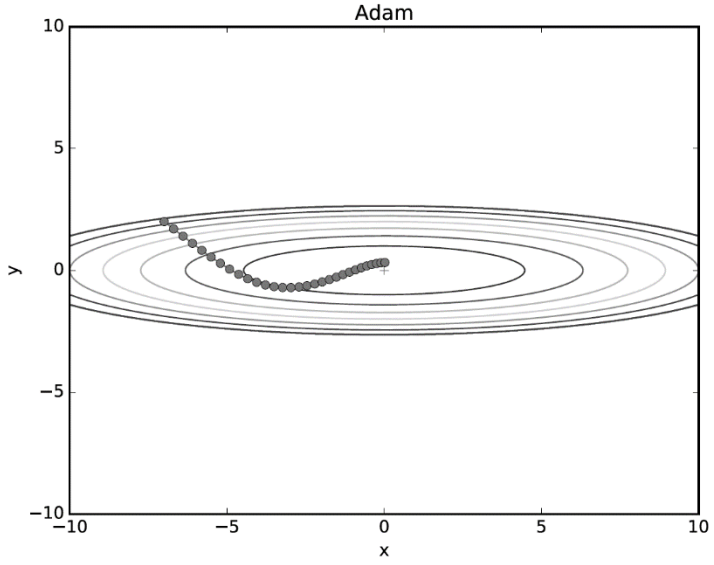
# 01. 매개변수 갱신

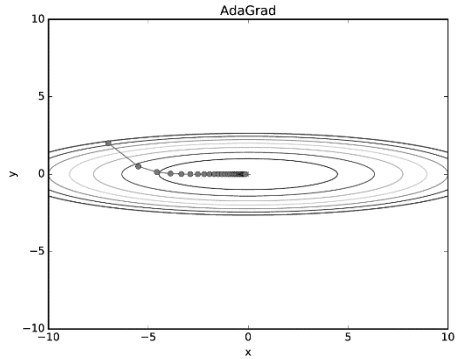
- Adam

- 모멘텀 + AdaGrad

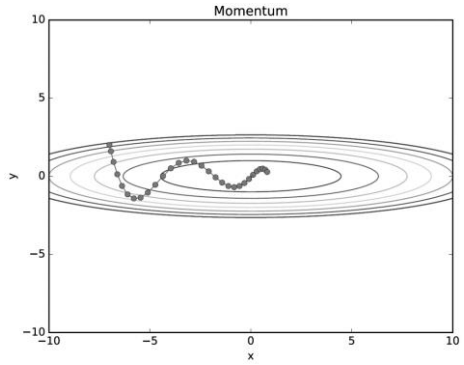
- 학습률을 줄여나가고 속도를 계산하여 학습의 갱신 강도를 적응적으로 조정해 나가는 방법

- 하이퍼파라미터의 '편향 보정'이 진행됨




AdaGrad




모멘텀

# 01. 매개변수 갱신

- 4가지 Optimizer 비교 (MNIST 데이터셋에 대한 학습 진도 비교)

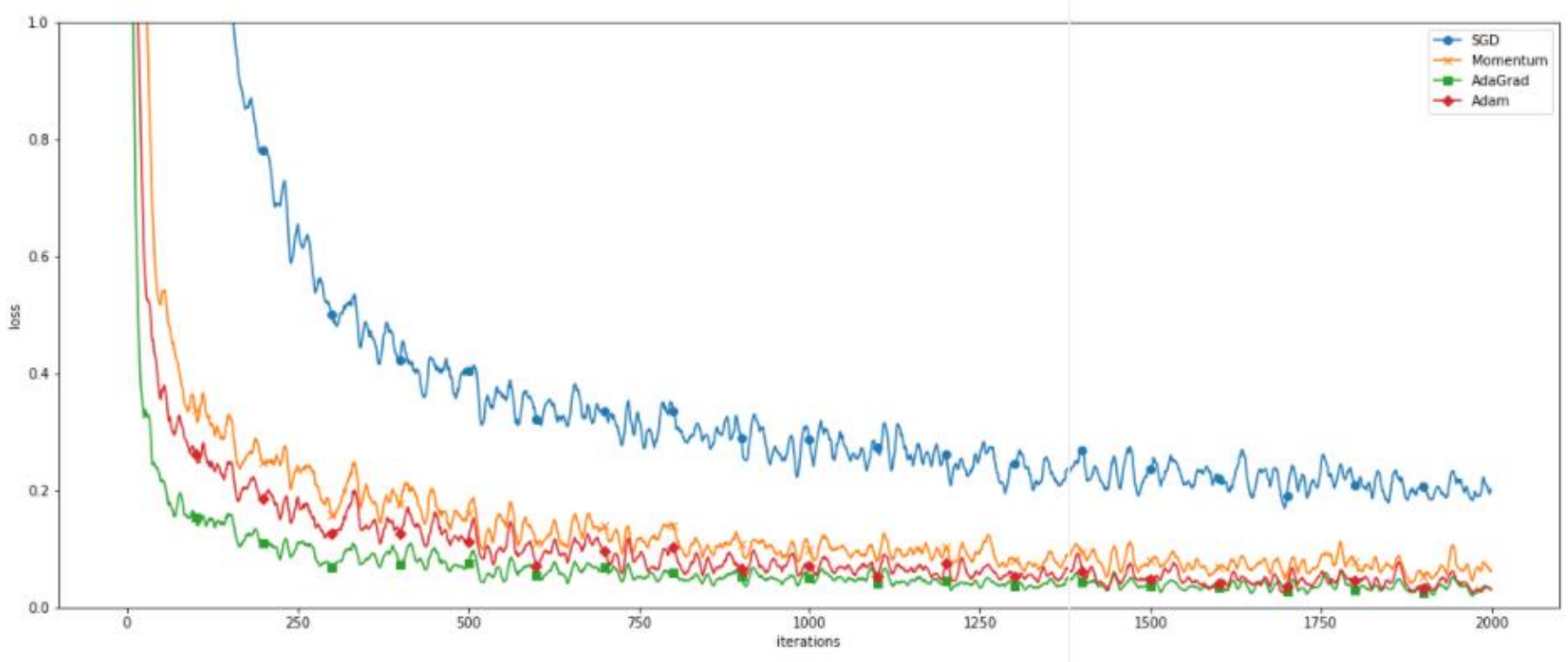
- 모든 문제에서 뛰어난 성능을 보이는 기법은 없음

- Adam을 많이 사용함

✓ **SGD의 학습 진도가 가장 느림**

✓ **AdaGrad가 빠른편**

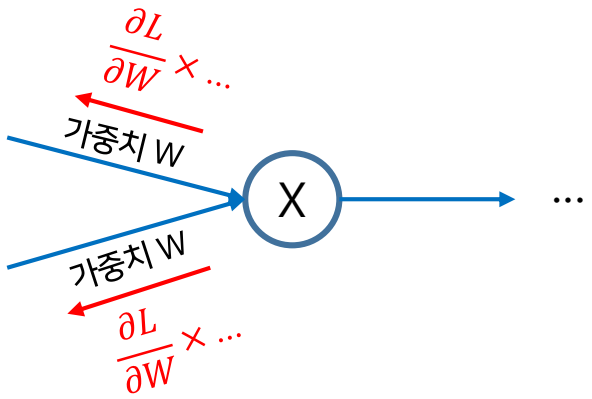
✓ **하이퍼파라미터 (학습률, 신경망의 구조) 에 따라 결과가 달라짐**



# 02. 가중치의 초깃값

- **★ 가중치의 초깃값 ★ 도 신경망 학습에서 중요**

- 초깃값을 0(균일한 값)으로 하면?
  - ➢ 학습이 올바르게 이루어지지 않음
  - ➢ 오차역전파법에 의해 모든 가중치의 값이 똑같이 갱신되기 때문 (같은 초깃값에서 시작하고 갱신을 거쳐도 여전히 같은 값)
  - ➢ 가중치를 여러 개 갖는 의미가 없음



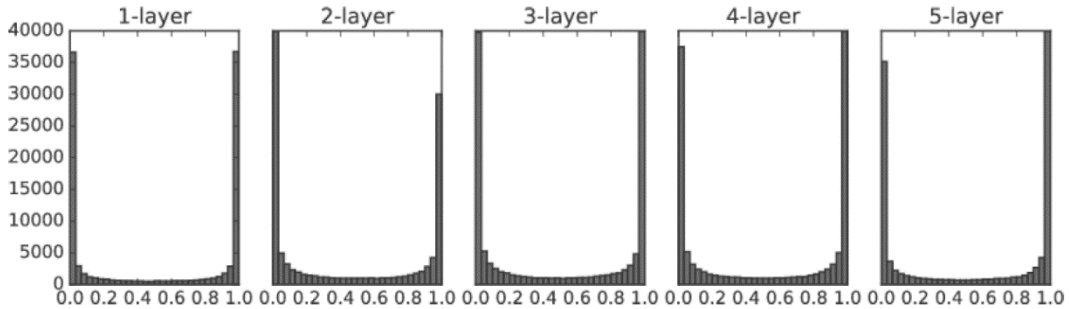

- 가중치의 대칭적인 구조를 무너뜨리려면 초깃값을 무작위로 설정해야 함

# 02. 가중치의 초깃값

- **표준편차에 따른 활성화 값의 분포 비교**
  - 분포된 정도

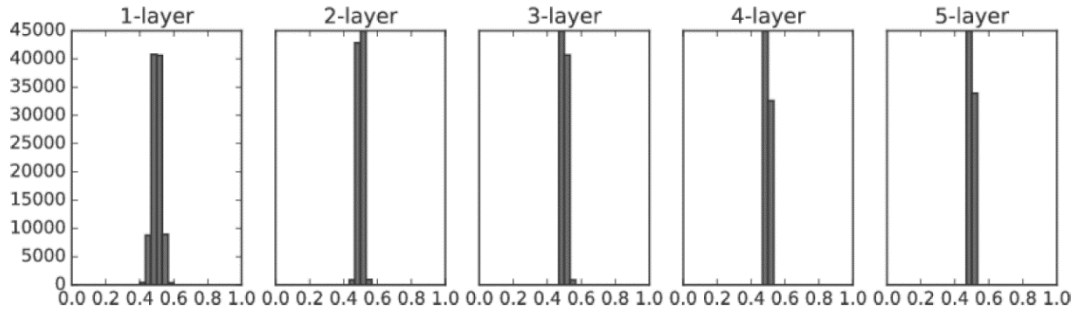
✓ **층이 5개 있고, 각 층마다 뉴런이 100개씩 있음**

✓ **활성화 함수로 시그모이드 함수 사용**




가중치를 표준편차가 1인 정규분포로 초기화할 때 활성화값 분포

- Sigmoid 함수의 출력값이 0과 1에 치우침
- 역전파의 기울기 값이 점점 작아지다 사라짐
- 기울기 소실 문제 발생




가중치를 표준편차가 0.01인 정규분포로 초기화할 때 활성화값 분포

- 0.5 중심으로 값이 몰림
- 활성화 값이 치우침 ➔ 다수의 뉴런을 둔 의미 X
  (표현력 제한 문제)

✓ **각 층의 활성화 값은 적당히 고루 분포되어야함**

✓ **층과 층 사이에 다양한 데이터가 흘러야 신경망 학습이 효율적으로 이루어짐**

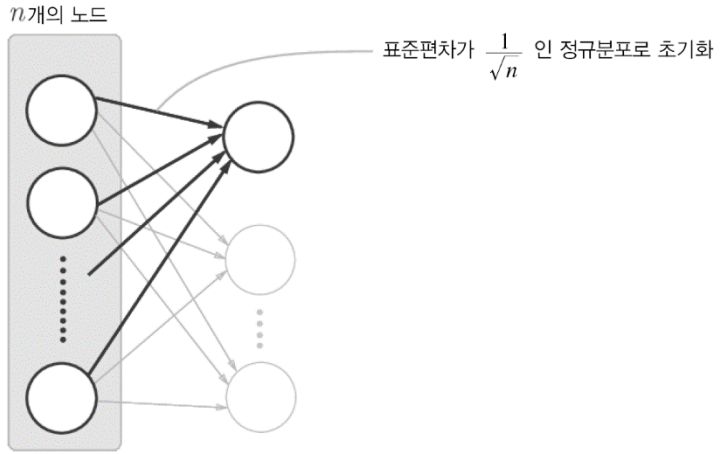
# 02. 가중치의 초깃값

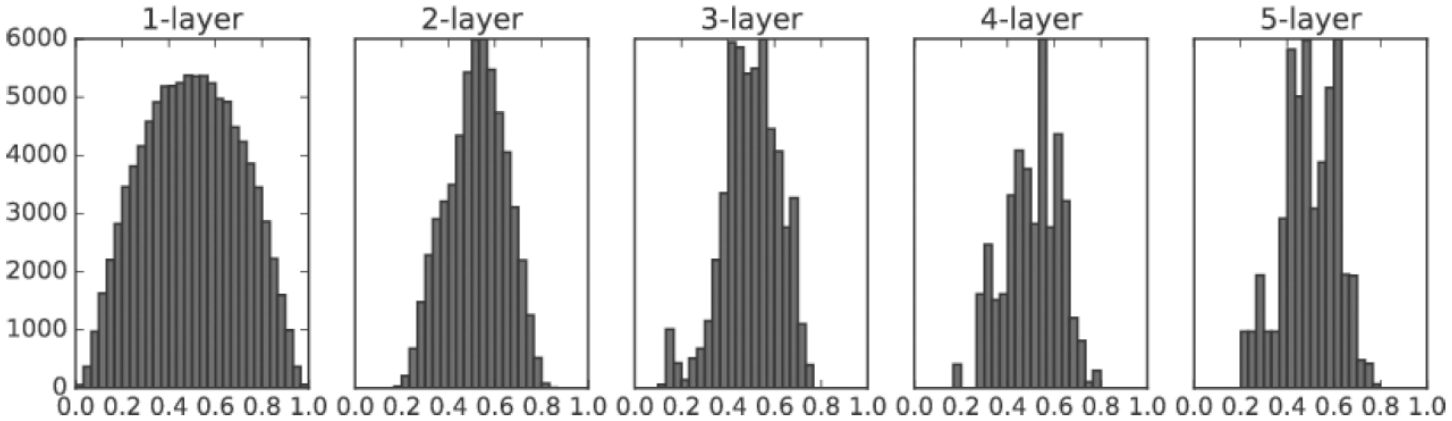
- Xavier 초깃값

- 각 층의 활성화값들을 광범위하게 분포시키는, 가중치의 적절한 분포 찾기 목적

  ➔ 앞 계층의 노드가 n개라면 표준편차가 $\frac{1}{\sqrt{n}}$ 인 분포 사용

- 앞 층에 노드가 많을수록 대상 노드의 초깃값으로 설정하는 가중치가 좁게 퍼짐




Xavier 초깃값을 사용했을 때

- 값들이 넓게 분포하며 Sigmoid 함수의 표현력도 제한 X
- 학습이 효율적으로 이루어질 것을 기대할 수 있음
- 일그러짐은 tanh 함수로 개선 가능

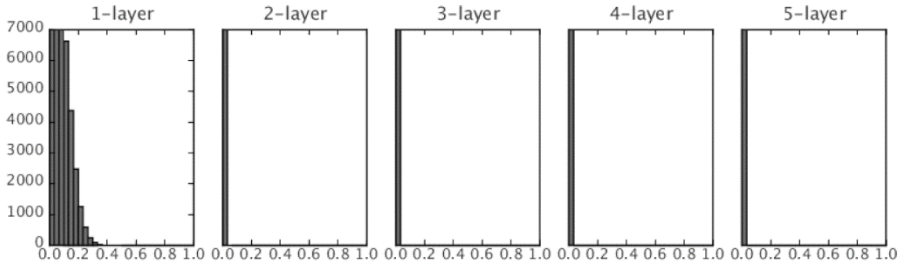
# 02. 가중치의 초깃값

- He 초깃값

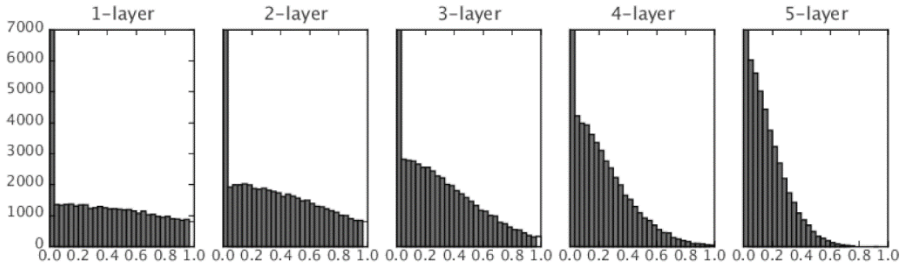
- ReLU에 특화된 초깃값

➔ 앞 계층의 노드가 n개라면 표준편차가 $\sqrt{\frac{2}{n}}$ 인 정규분포 사용

➔ ReLU는 음의 영역이 0이므로 더 넓게 분포시키기 위함으로 볼 수 있음




표준편차가 0.01인 정규분포를 가중치 초깃값으로 사용한 경우

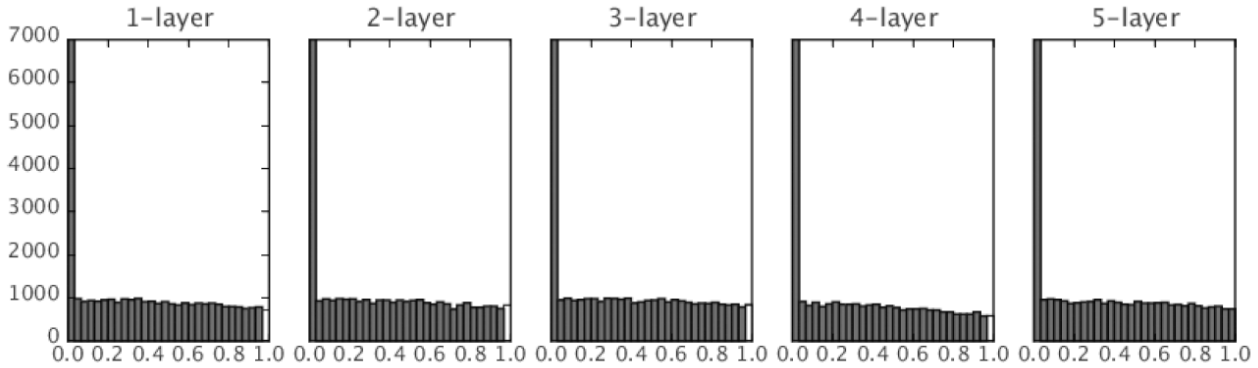



Xavier 초깃값을 사용한 경우

모든 층에서 균일하게 분포

ReLU : He 초깃값

Sigmoid, tanh 등의 S자 모양 곡선 : Xavier 초깃값

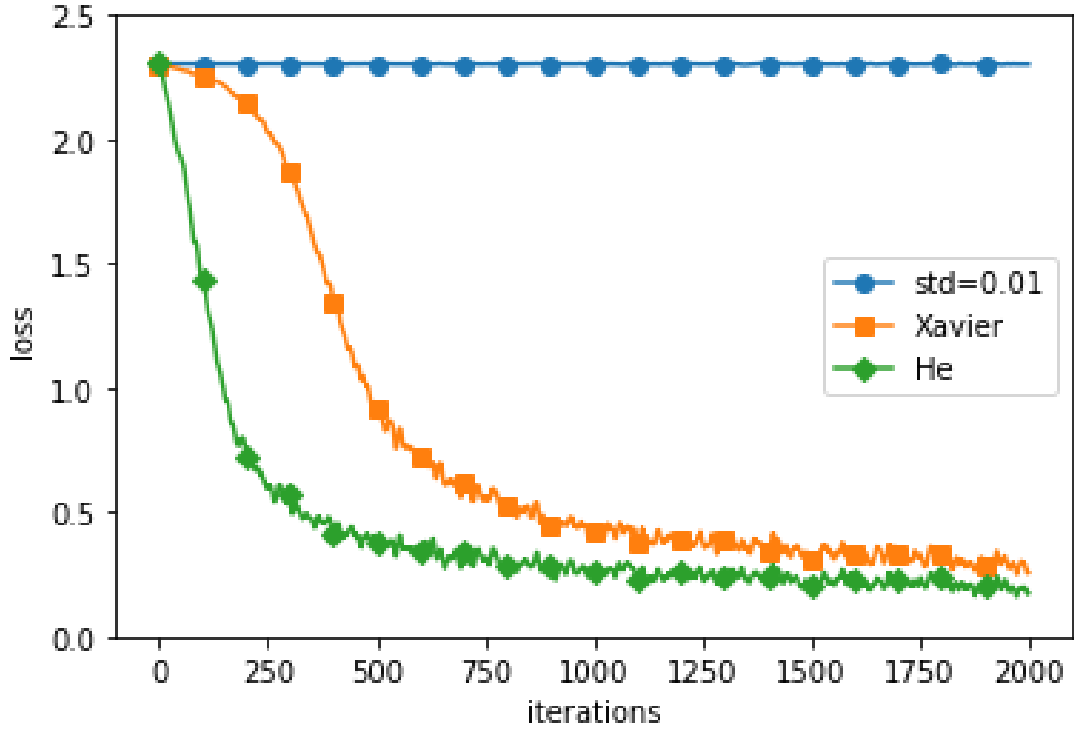



He 초깃값을 사용한 경우

# 02. 가중치의 초깃값

- MNIST 데이터셋으로 본 가중치 초깃값 비교

  ➔ 가중치의 초깃값은 신경망 학습에 아주 중요



감사합니다