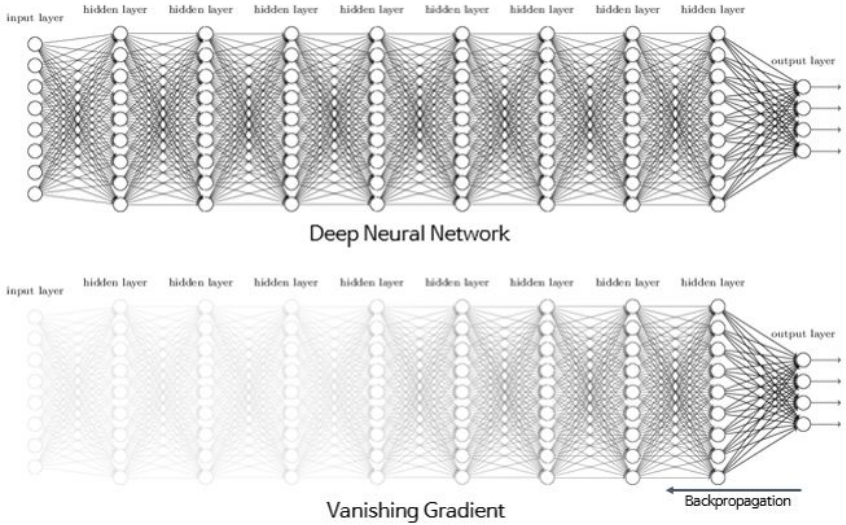
학습 관련 기술들

신경망 학습의 어려움

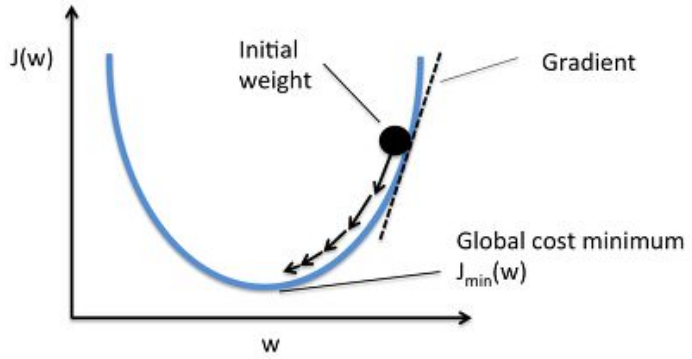
- 그래디언트 소실(vanishing gradient) 또는 폭주(exploding)가 발생할 수 있음
- 모델이 복잡하고 커질수록 학습시간이 매우 느려짐
- 모델이 복잡할수록 오버피팅(overfitting)될 위험이 큼



목차

매개변수 갱신

경사하강법 복습



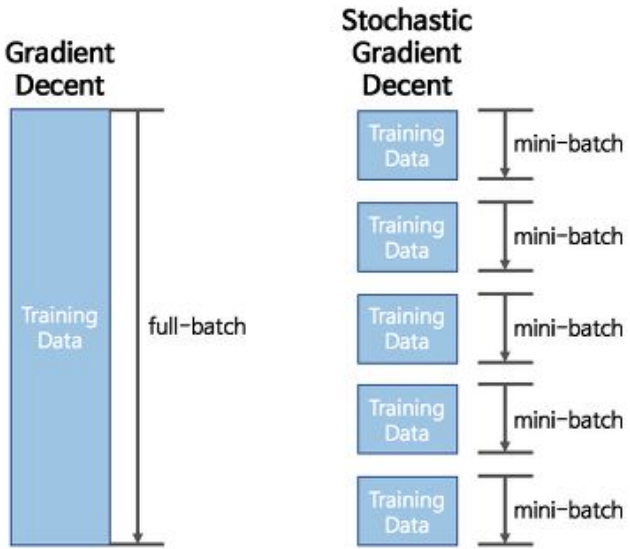
weight의 업데이트 = 에러 낮추는 방향 (decent) X 한발자국 크기 (learning rate) X 현 지점의 기울기 (gradient)

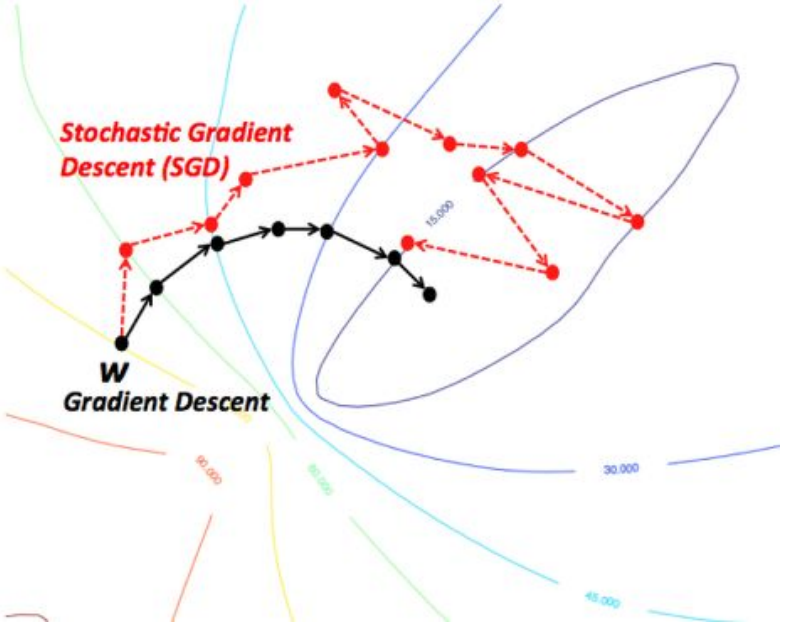
$$-\gamma \nabla F(\mathbf{a^n}) \quad - \quad \gamma \quad \nabla F(\mathbf{a^n})$$

https://seamless.tistory.com/38

매개변수 갱신

경사하강법 vs 확률적 경사하강법





매개변수 갱신

확률적 경사 하강법 (SGD)

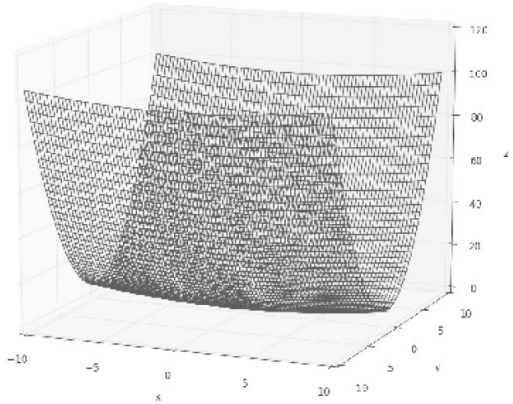
$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial L}{\partial \mathbf{W}}$$

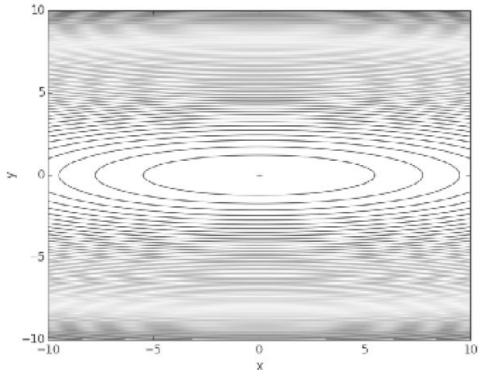
SGD는 단순하고 구현이 쉽지만, 문제에 따라 비효율적일 때가 있음

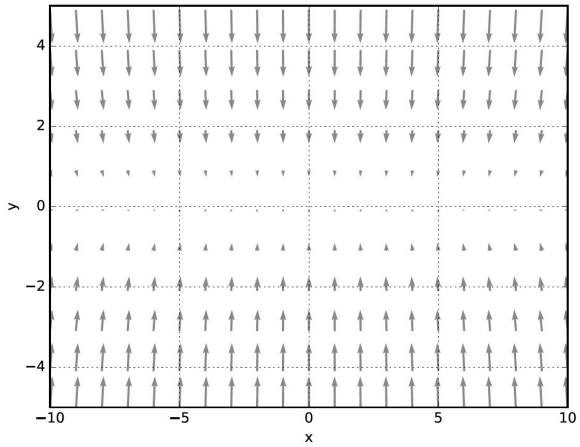
매개변수 갱신

확률적 경사 하강법 (SGD)

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



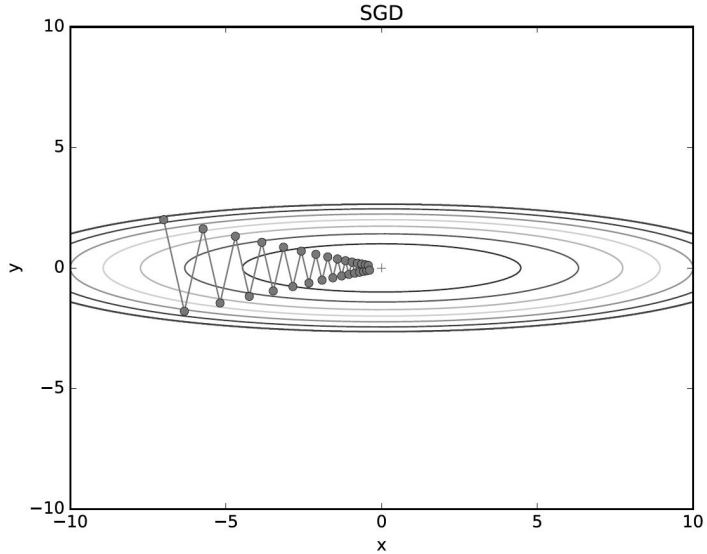




기울기

매개변수 갱신

확률적 경사 하강법 (SGD)



SGD에 의한 최적화 갱신 경로
지그재그로 이동하니 비효율적으로 탐색을 하는 것을 알 수 있음

매개변수 갱신

모멘텀 (Momentum)

모멘텀은 '운동량, 관성, 탄력'등으로 해석할 수 있음

가중치를 수정하기 이 전 기울기까지 고려하여 같은 방향으로 일정한 비율만 수정되도록 한 것

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

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

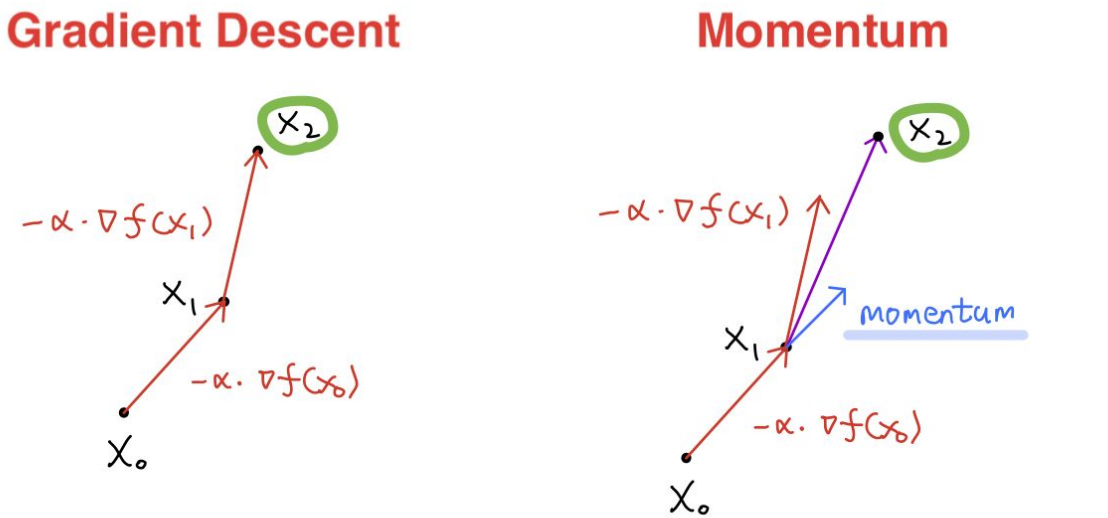
쉽게 말하면, **원래 이동하던 방향에 대한 관성**을 어느정도 반영하겠다는 것

https://icim.nims.re.kr/post/easyMath/428

https://icim.nims.re.kr/post/easyMath/428

매개변수 갱신

모멘텀 (Momentum)



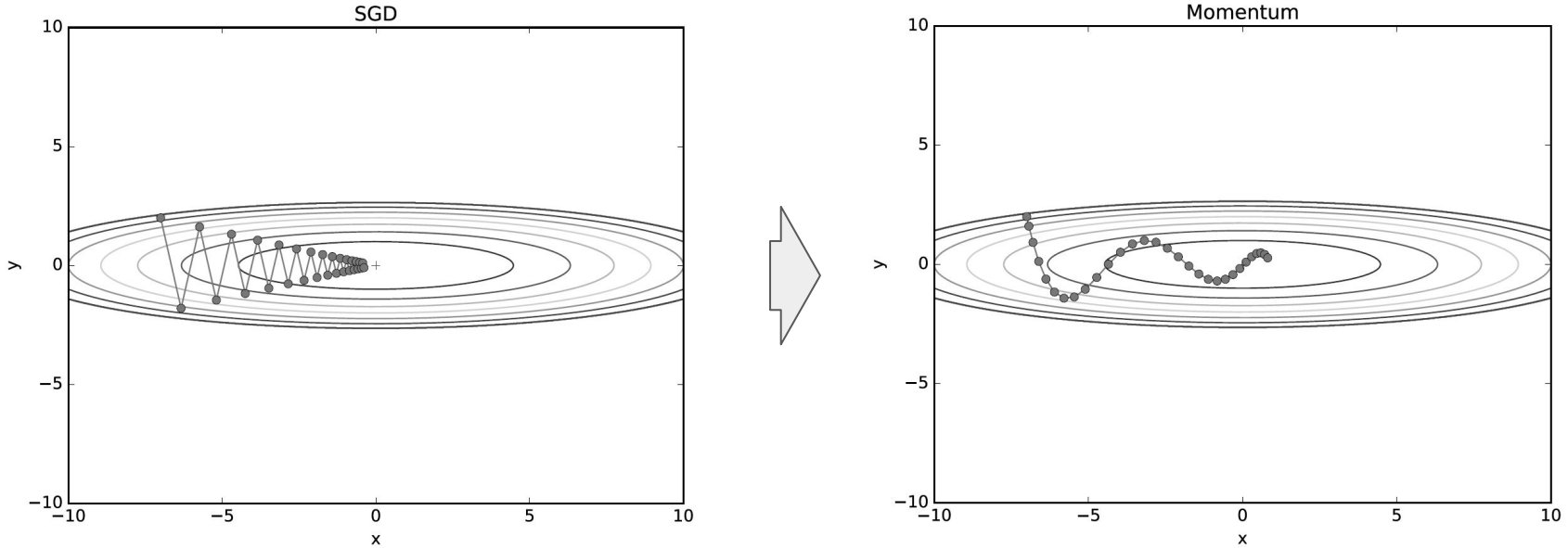
$$x_1 = x_0 - \alpha \cdot f(x_0)$$
$$x_2 = x_1 - \alpha \cdot f(x_1)$$

$$x_1 = x_0 - \alpha \cdot f(x_0)$$
$$x_2 = x_1 - \alpha \cdot f(x_1) + \text{momentum}$$

https://icim.nims.re.kr/post/easyMath/428

매개변수 갱신

모멘텀 (Momentum)



https://icim.nims.re.kr/post/easyMath/428

매개변수 갱신

AdaGrad (Adaptive Gradient)

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

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

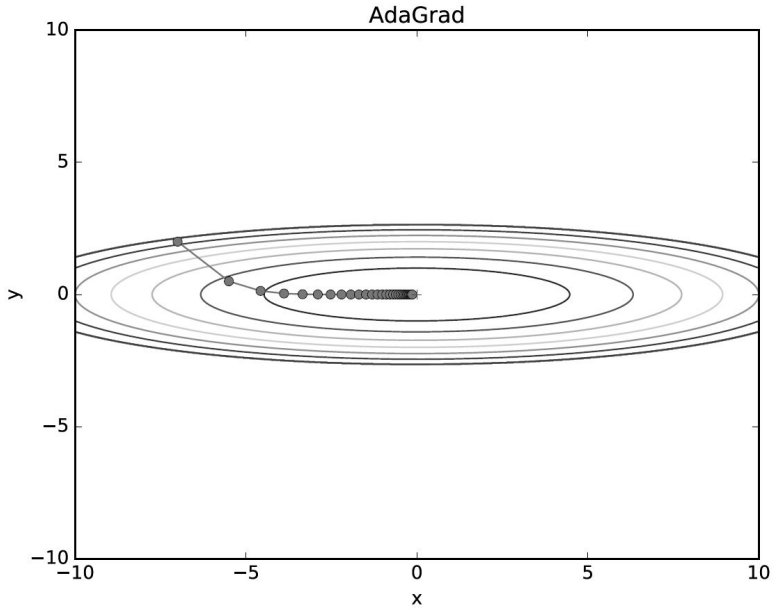
학습률이 너무 작으면 학습 시간이 길어지고, 너무 크면 수렴하지 않는 문제가 있음

학습률 감소(learning rate decay)를 통해 해결함

https://icim.nims.re.kr/post/easyMath/428

매개변수 갱신

AdaGrad (Adaptive Gradient)

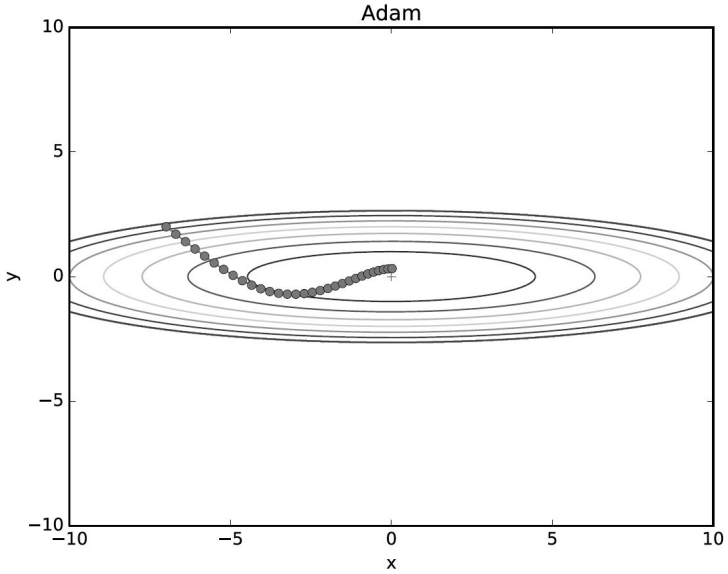


https://icim.nims.re.kr/post/easyMath/428

매개변수 갱신

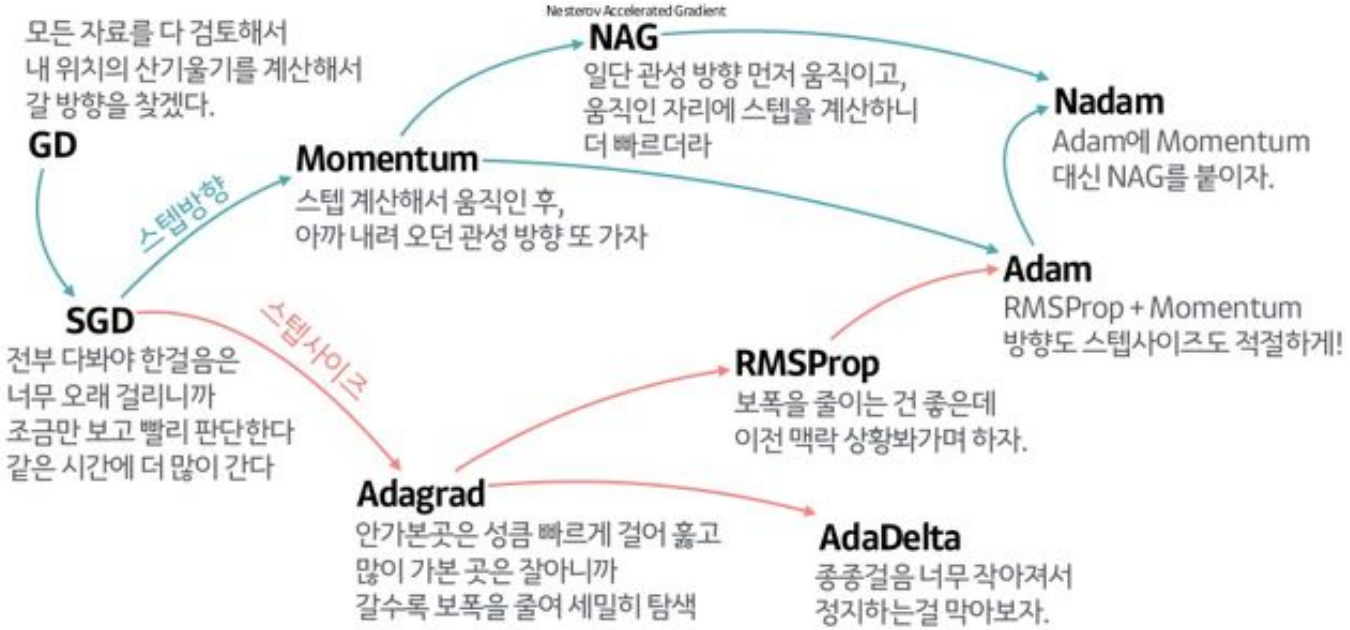
Adam

Momentum, AdaGrad(decay)를 융합한 방법



매개변수 갱신

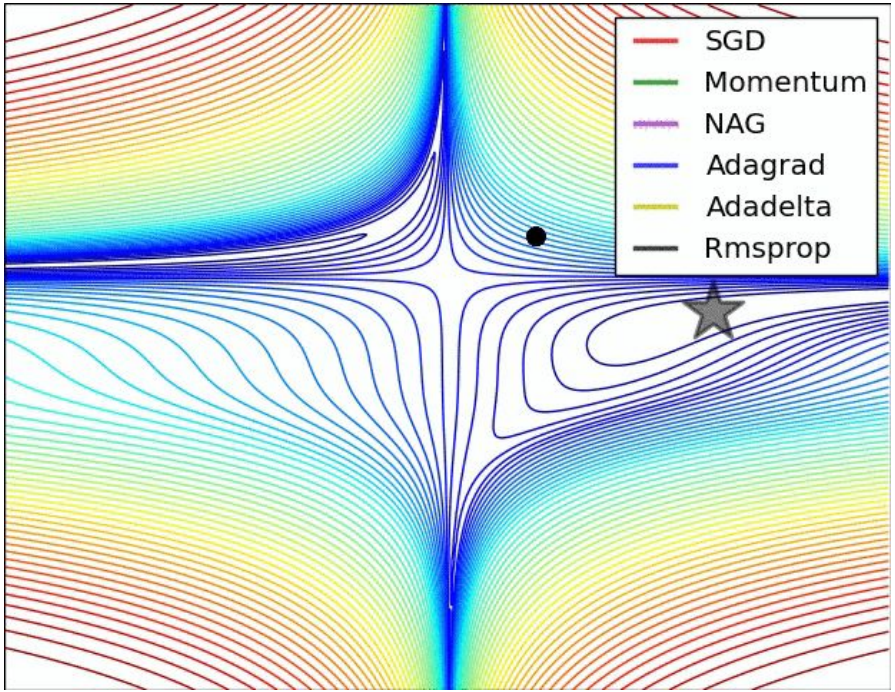
그 외 Optimizer들을 정리하면...



https://light-tree.tistory.com/141?category=755497

매개변수 갱신

그 외 Optimizer들을 정리하면...



https://light-tree.tistory.com/141?category=755497

Tensorflow Optimizer

Classes

`class Adadelta` : Optimizer that implements the Adadelta algorithm.

`class Adagrad` : Optimizer that implements the Adagrad algorithm.

`class Adam` : Optimizer that implements the Adam algorithm.

`class Adamax` : Optimizer that implements the Adamax algorithm.

`class Ftrl` : Optimizer that implements the FTRL algorithm.

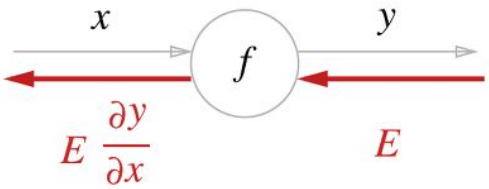
`class Nadam` : Optimizer that implements the NAdam algorithm.

`class Optimizer` : Base class for Keras optimizers.

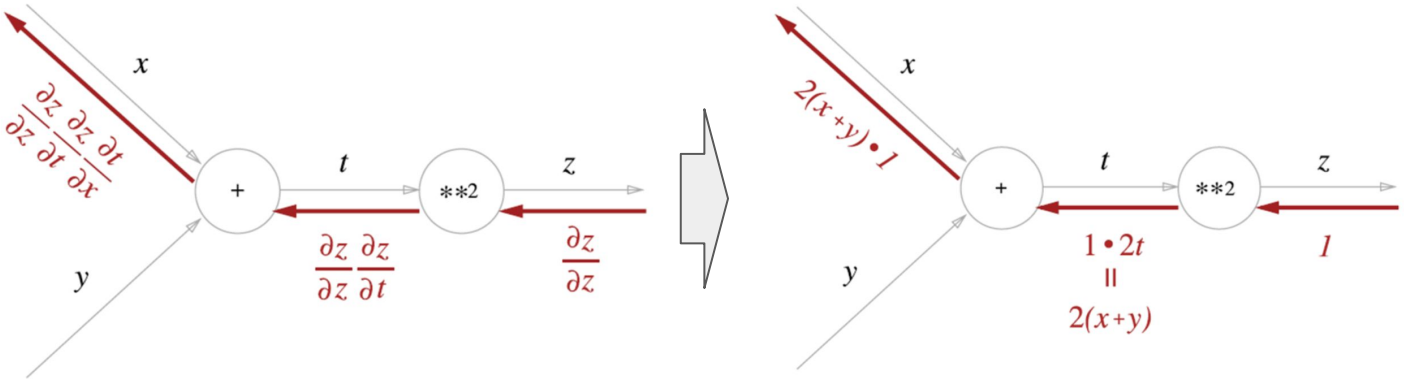
`class RMSprop` : Optimizer that implements the RMSprop algorithm.

`class SGD` : Gradient descent (with momentum) optimizer.

가중치의 초기값

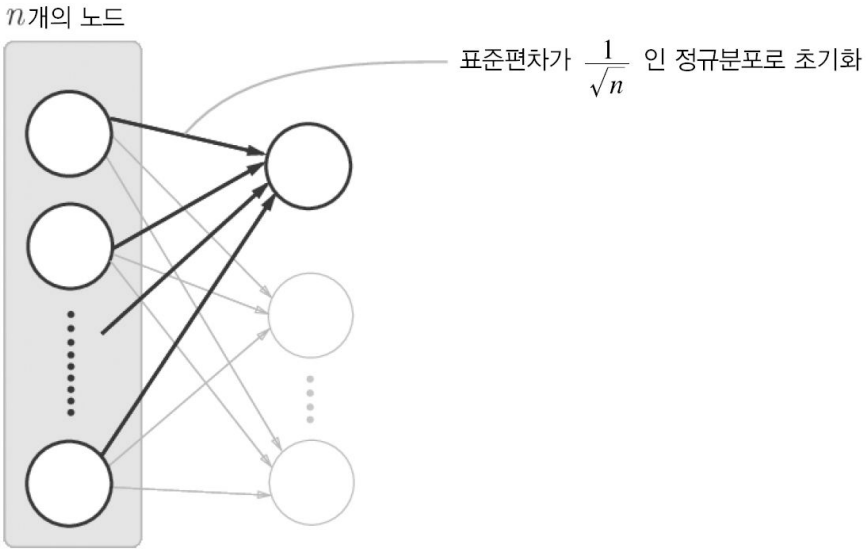


초기값이 0(또는 모두 같은값) 이라면, 역전파 과정에서
동일하게 학습 될 것이기 때문에, 임의의 값으로 설정해주어야 함



가중치 초기화 방법

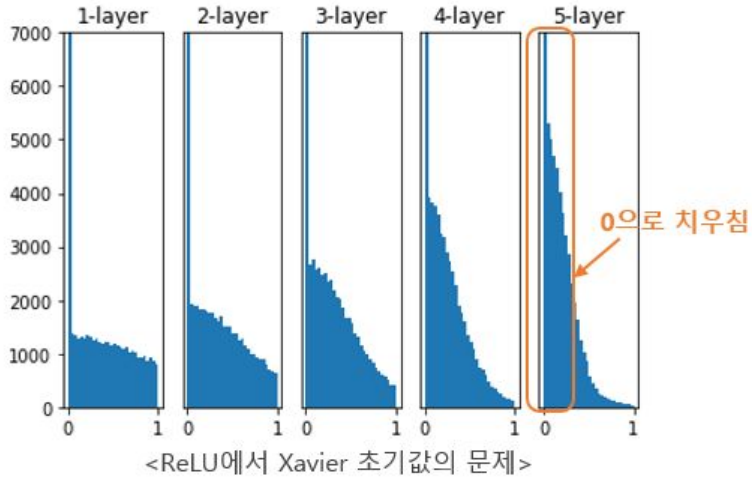
Xavier Initialization



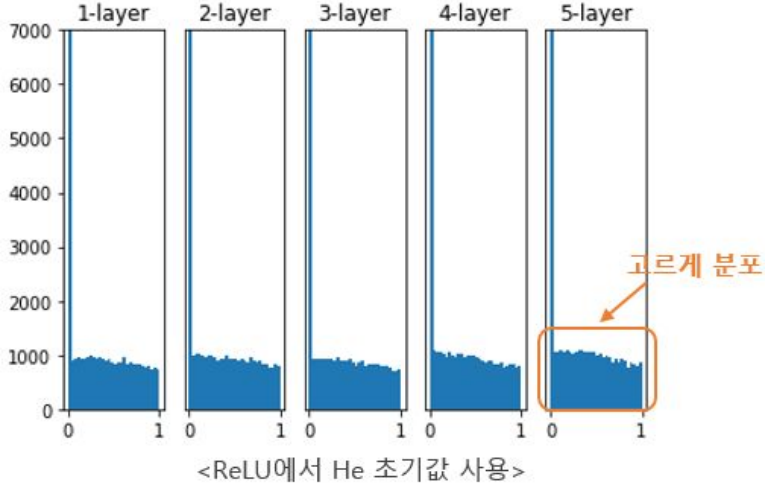
Sigmoid, Tanh 등 비선형 함수에서 Xavier 초기화가 성능이 좋음

가중치 초기화 방법

He Initialization



<ReLU에서 Xavier 초기값의 문제>

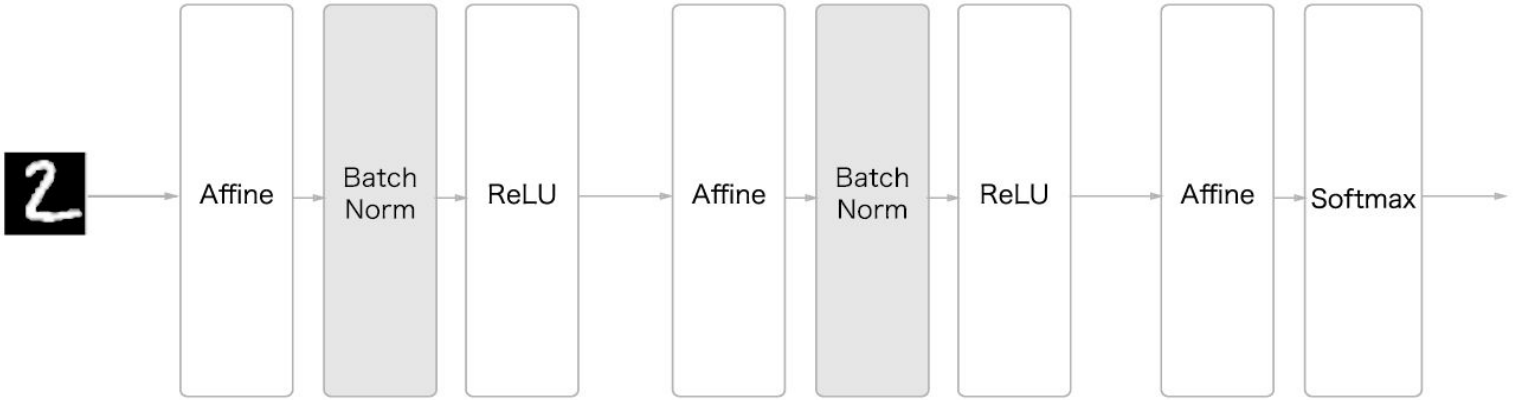


<ReLU에서 He 초기값 사용>

ReLU는 입력이 음수일 때 출력이 전부 0이기 때문에, Xavier 초기값보다 더 넓게 분포를 생성함

https://excelsior-cjh.tistory.com/177

배치 정규화



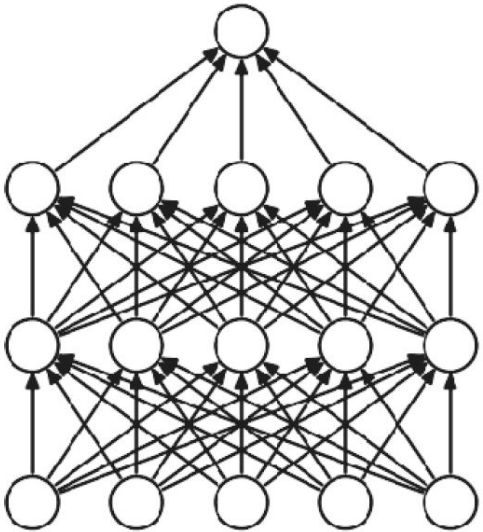
배치정규화의 기본 아이디어는

각 층에서의 활성화 값이 적당히 분포되도록 조정하는 것

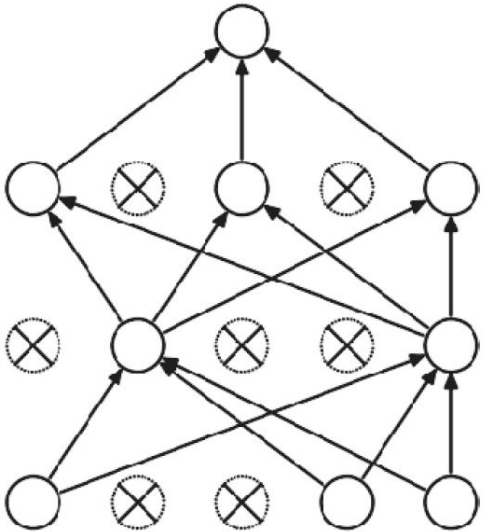
배치 정규화

- 학습을 빨리 진행할 수 있음 (학습 속도 개선)
- 초기값에 크게 의존하지 않음
- 오버피팅을 억제함

드롭아웃 (Dropout)



(a) 일반 신경망



(b) 드롭아웃을 적용한 신경망