Ch.6
학습 관련 기술들

딥러닝 기초 c조 / 9기 조세린

목차

1. 매개변수 갱신
2. 가중치의 초깃값
3. 배치 정규화
4. 바른 학습을 위해
5. 적절한 하이퍼파라미터 값 찾기

1. 매개변수 갱신

1) 모멘텀
2) AgaGrad
3) Adam

최적화 optimizaton

- 신경망 학습의 목적 : 손실함수의 값을 최소로 만드는 매개변수를 찾는 것
- 즉, 매개변수의 최적값을 찾는 문제

확률적 경사 하강법 (SGD)

: 매개변수의 기울기를 통해 매개변수 값을 반복 갱신

➡ 기울어진 방향으로 일정거리 정도만 갱신

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

$\mathbf{W}$: 갱신할 가중치 매개변수

$\frac{\partial L}{\partial \mathbf{W}}$: W에 대한 손실 함수의 기울기

η : 학습률 (미리 정해서 사용)

SGD의 단점

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

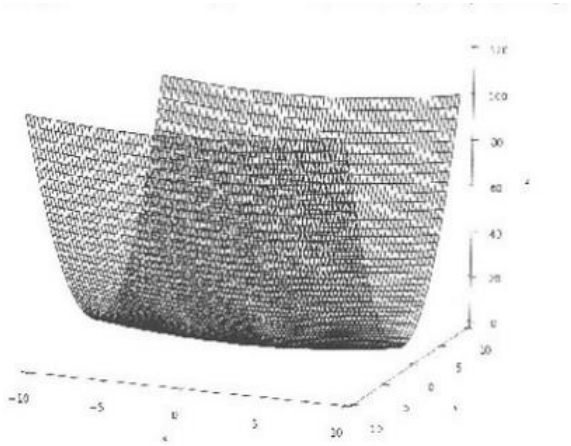
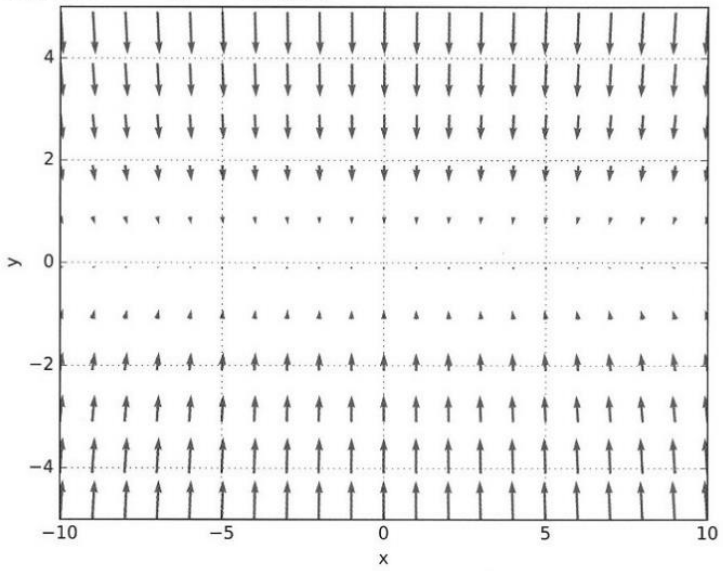
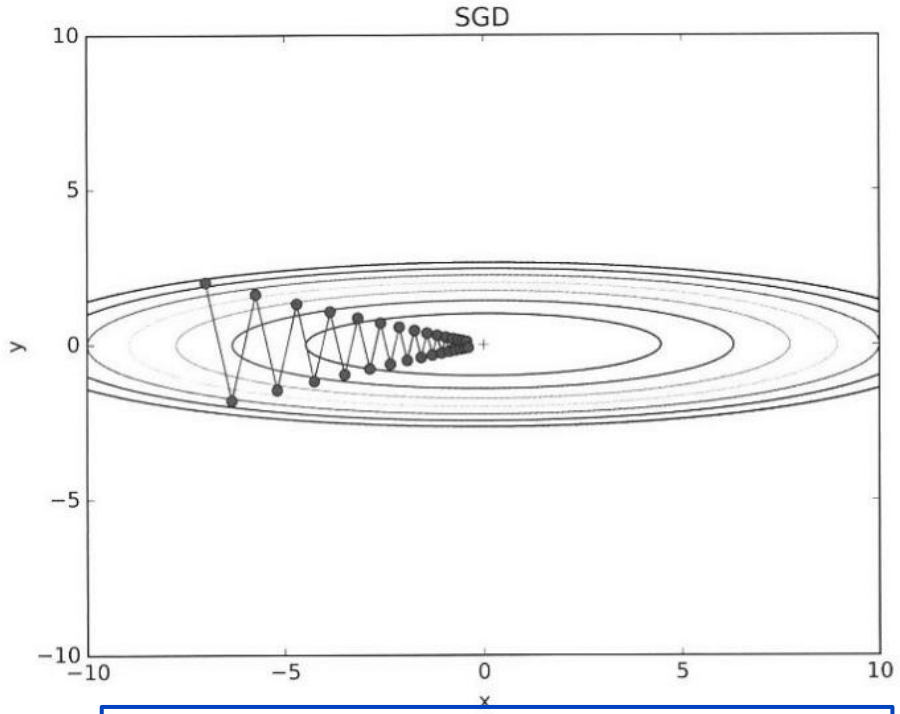


그림 6-2 $f(x,y) = \frac{1}{20}x^2 + y^2$의 기울기



기울기의 y축 방향은 크고,
x축 방향은 작다.



최솟값 (0,0) 까지 지그재그로 이동

- 비등방성 함수(기울기가 달라지는 함수)에서는 탐색 경로가 비효율적
  -기울어진 방향이 본래의 최솟값과 다른 방향을 가리킴

모멘텀 Momentum

$$\mathbf{v} \leftarrow \alpha\mathbf{v} - \eta \frac{\partial L}{\partial \mathbf{W}}$$ ➡ 기울기 방향으로 힘을 받아 물체가 가속

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

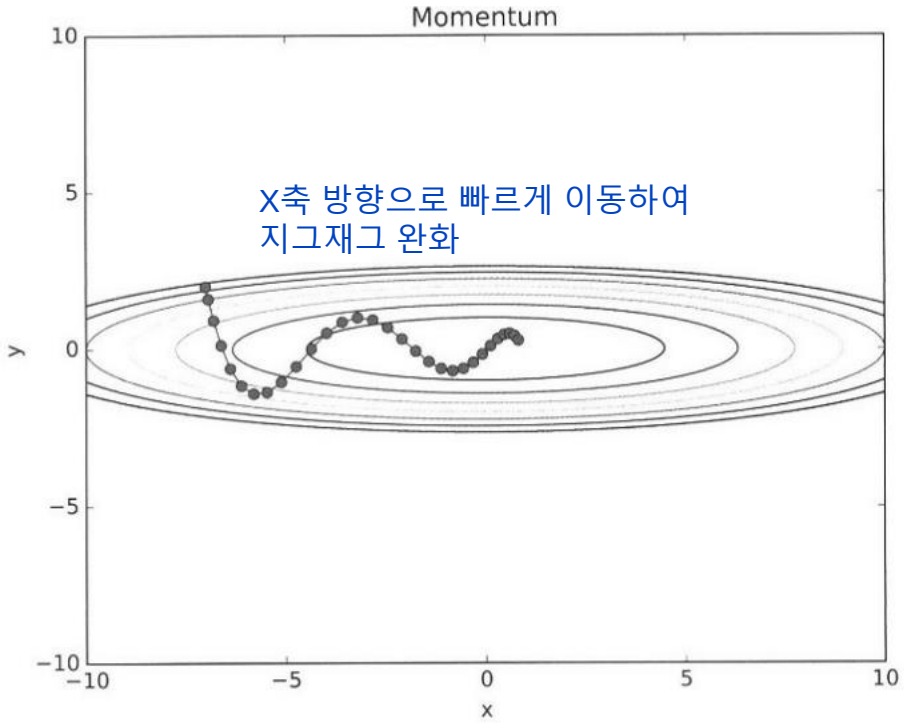
$\mathbf{W}$: 갱신할 가중치 매개변수

$\frac{\partial L}{\partial \mathbf{W}}$: W에 대한 손실 함수의 기울기

η : 학습률 (미리 정해서 사용)

$\mathbf{v}$: 속도

$\alpha\mathbf{v}$: 물체가 아무런 힘을 받지 않을 때 서서히 하강시키는 역할



- x축의 힘이 아주 작지만, 방향 변화x
  => 한 방향으로 일정하게 가속

AdaGrad (Adaptive Gradient)

- **학습률 감소 learning rate decay**

: 학습을 진행하면서 학습률을 점차 줄여가는 방법

- **AdaGrad**

: '각각의' 매개변수에 '맞춤형'으로, 개별 매개변수의 학습률을 조정

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

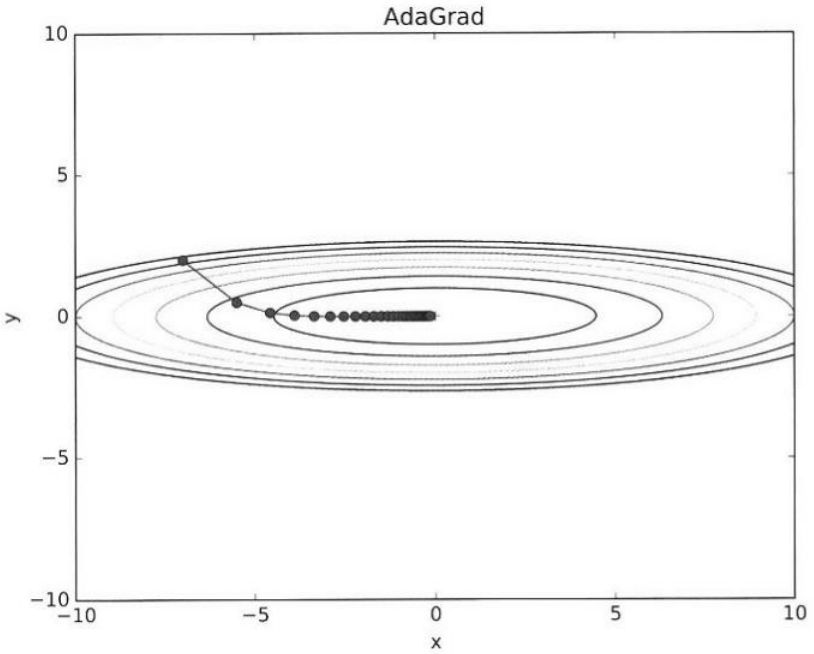
$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{1}{\sqrt{\mathbf{h}}} \frac{\partial L}{\partial \mathbf{W}}$$

h: 기존 기울기 값을 제곱해서 sum

$1/\sqrt{h}$: 매개변수 갱신할 때 곱해준다.

⇒크게 갱신된 원소는 학습률이 낮아짐

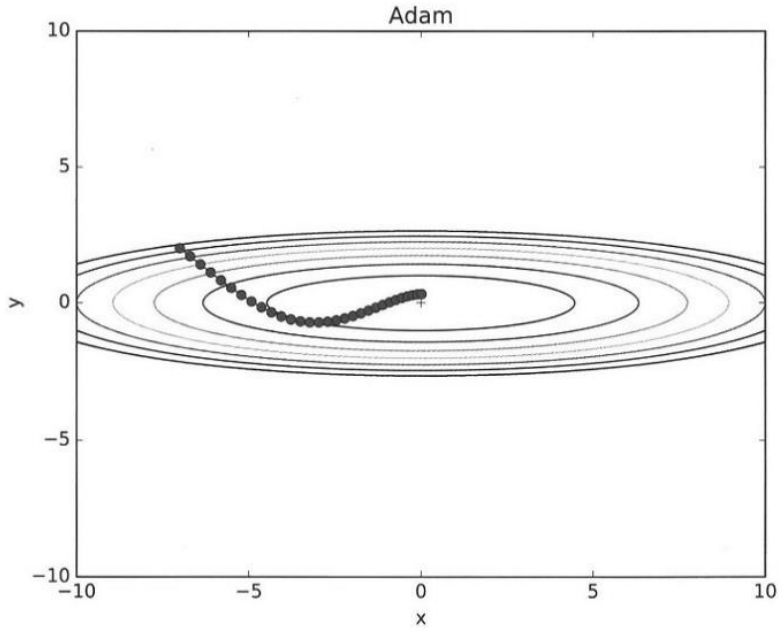
- 학습이 진행될수록, 갱신강도가 약해짐



- y축의 경우, 기울기가 커서 처음엔 크게 움직이지만, 갱신 정도도 큰 폭으로 작아짐

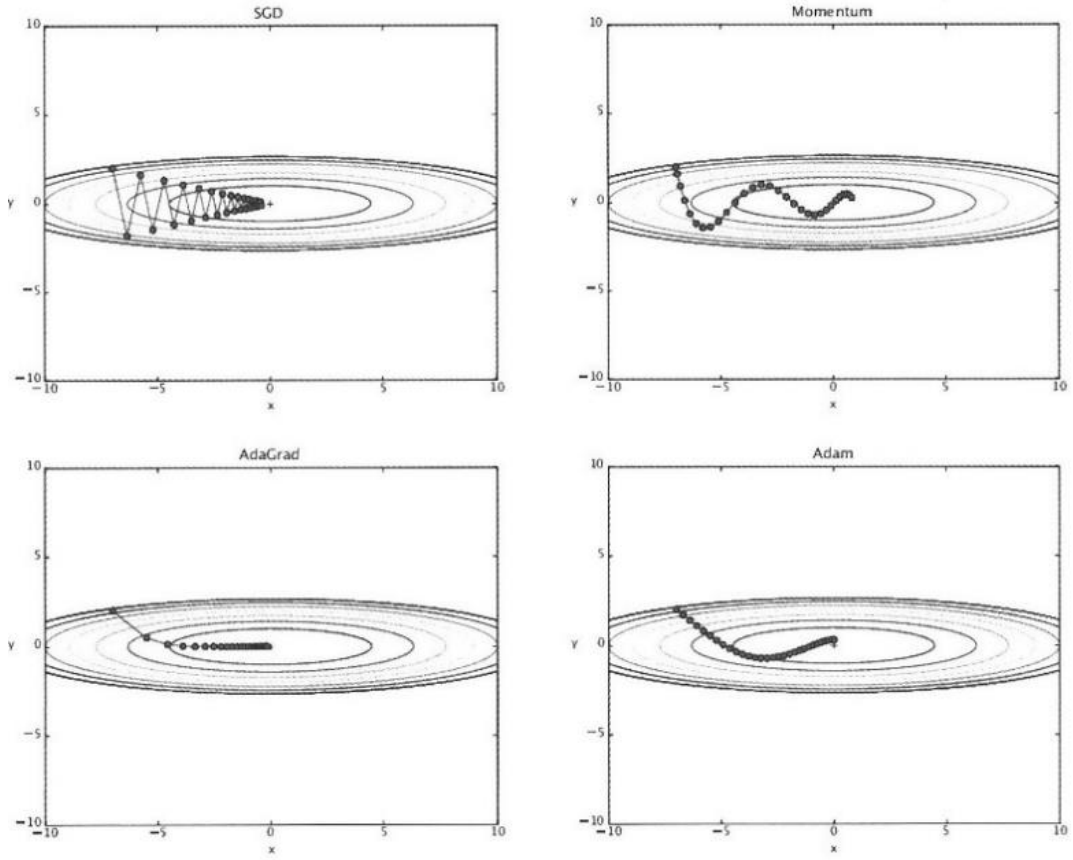
Adam

- 모멘텀 + AdaGrad



모멘텀 때보다 공의 좌우 흔들림이 ↓
∵학습의 갱신 강도를 적응적으로 조정

어느 갱신 방법을 이용?



2. 가중치의 초깃값

가중치 감소 기법 weight decay

- **가중치 매개변수의 값이 작아지도록 학습하는 방법**
- 오버피팅을 억제해 범용 성능을 높이는 기술

가중치를 작게 만든다?
→초깃값도 최대한 작은 값에서 시작? = 0에서 시작?

오차역전파법에서 모든 가중치의 값이 똑같이 갱신되어 버림
→가중치를 여러 개 갖는 의미가 사라짐

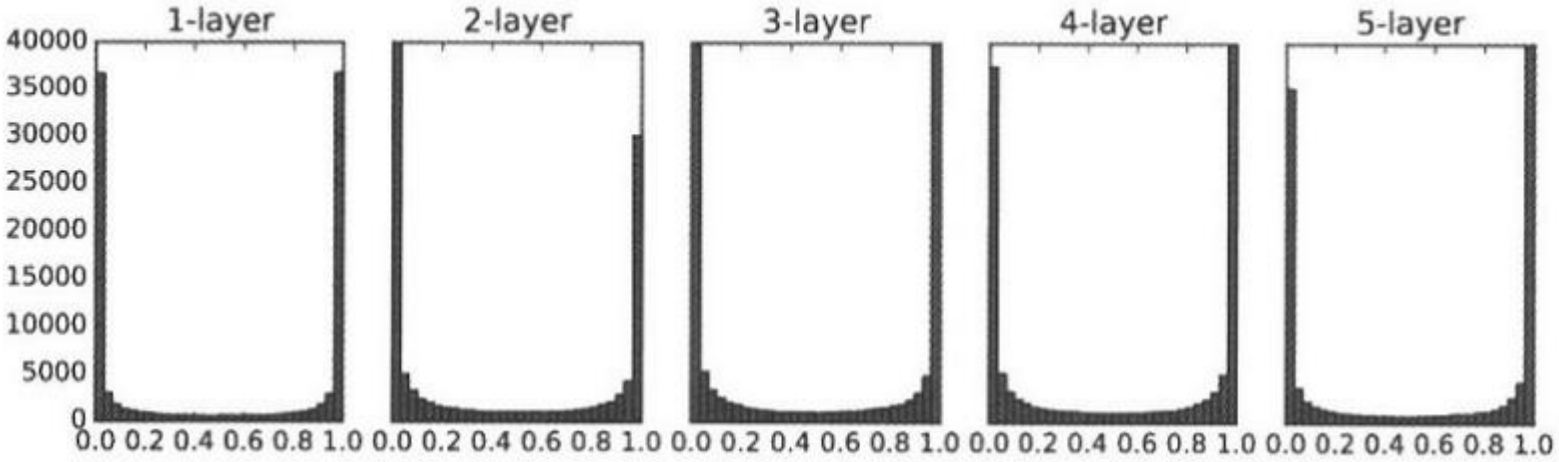
은닉층의 활성화값 분포

가중치의 초깃값에 따라 은닉층 활성화값들이 어떻게 변화?

⇒표준편차 (가중치의 분포)를 바꿔가며 활성화값의 분포 확인

- 층 5개
- 각 층의 뉴런 100개
- 1000개의 정규분포 입력데이터
- 활성화 함수 : 시그모이드 함수

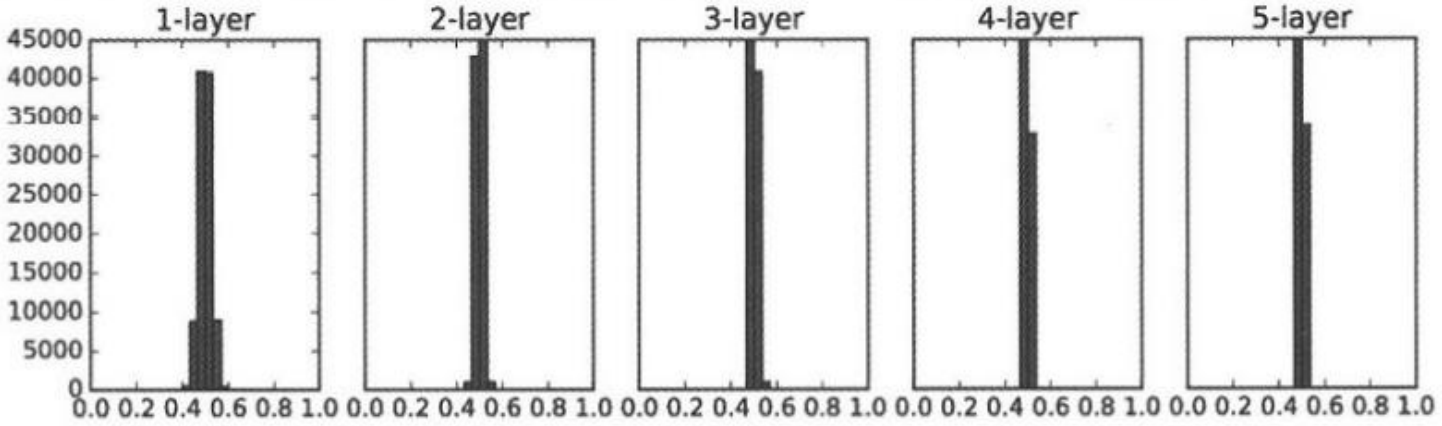
그림 6-10 가중치를 표준편차가 1인 정규분포로 초기화할 때의 각 층의 활성화값 분포



데이터가 0과 1에 치우쳐져 분포
→ 역전파 기울기 값이 작아지다가 사라짐 : **기울기 소실**

은닉층의 활성화값 분포

그림 6-11 가중치를 표준편차가 0.01인 정규분포로 초기화할 때의 각 층의 활성화값 분포



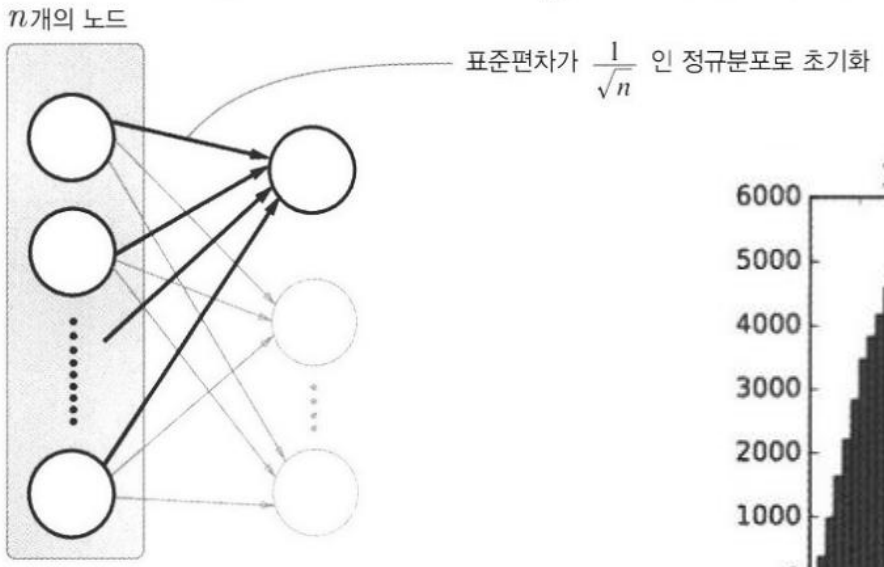
다수의 뉴런이 거의 같은 값 출력
→ 뉴런을 여러 개 둔 의미가 사라짐 : **표현력을 제한한다**

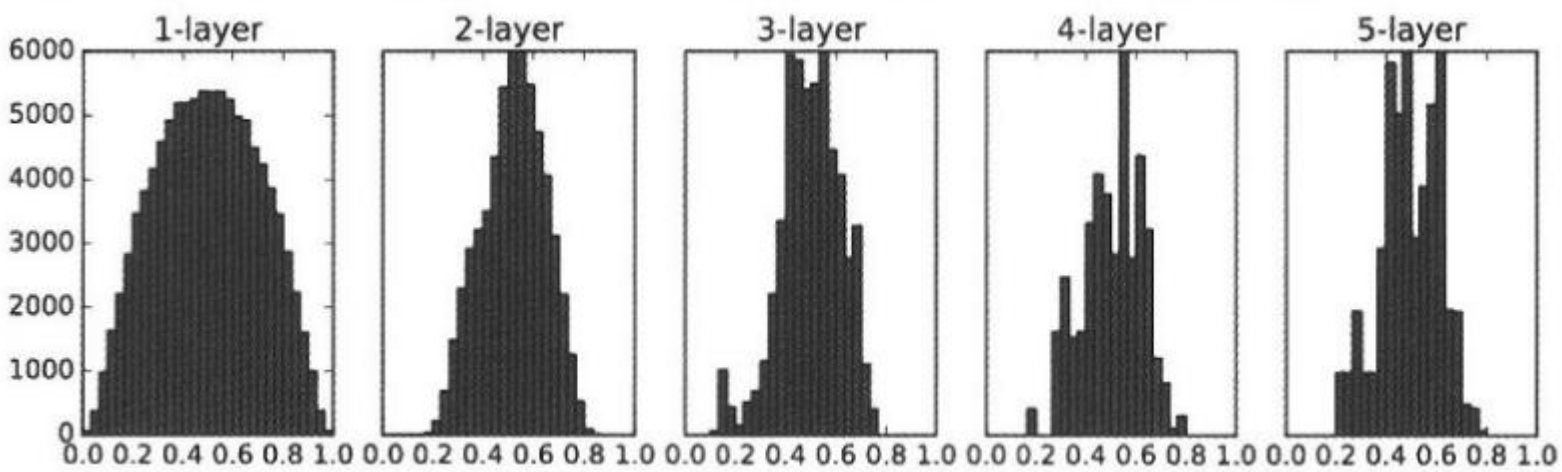
➡ **각 층의 활성화 값이 적당히 잘 분포되어 있어야**
다양한 데이터가 층과 층 사이에 흘러 신경망 학습이 효율적으로 이루어진다.

Xavier 초깃값

- 앞 계층의 노드가 n개, **표준편차가 $1/\sqrt{n}$**인 분포 사용

  => 각 층의 활성화값들을 광범위하게 분포시키는 것이 목적



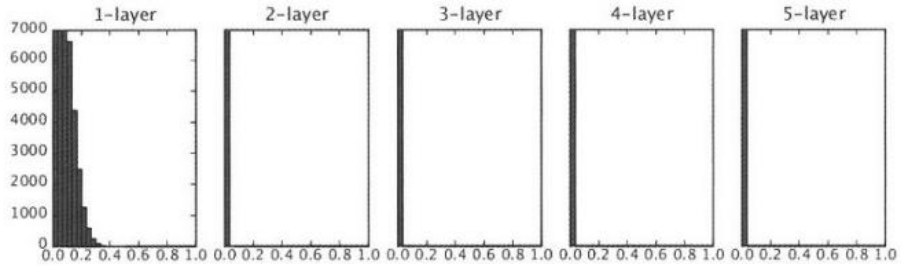


활성화 값이 넓게 분포됨을 확인 가능

ReLU : He 초깃값

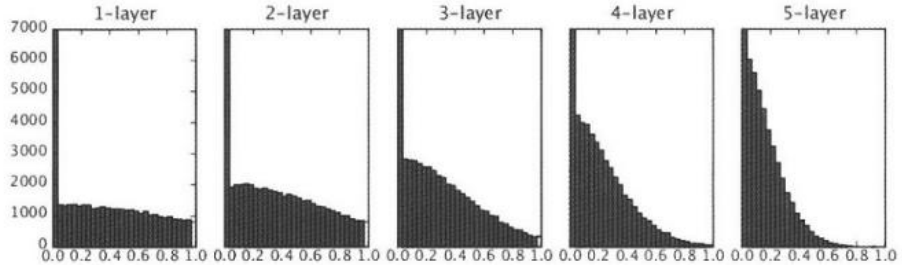
- Xavier 초깃값 : 활성화 함수가 선형인 것이 전제

  →ReLU 이용시, He 초깃값 사용



표준편차가 0.01인 정규분포를 가중치 초깃값으로 사용한 경우

기울기가 매우 작아짐

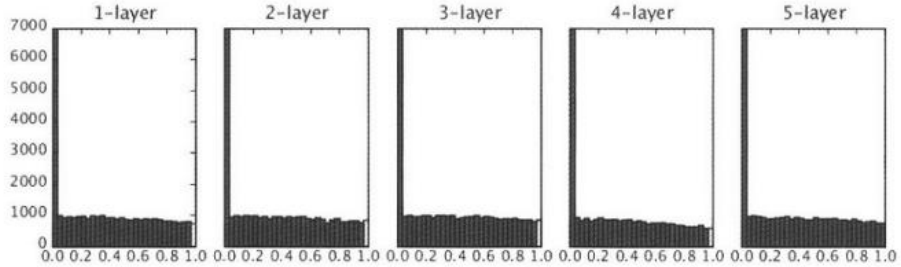


Xavier 초깃값을 사용한 경우

기울기가 치우침

He 초깃값

: 앞 층의 노드가 n개일 때, **표준편차가 $\sqrt{2/n}$인 정규분포** 사용



He 초깃값을 사용한 경우

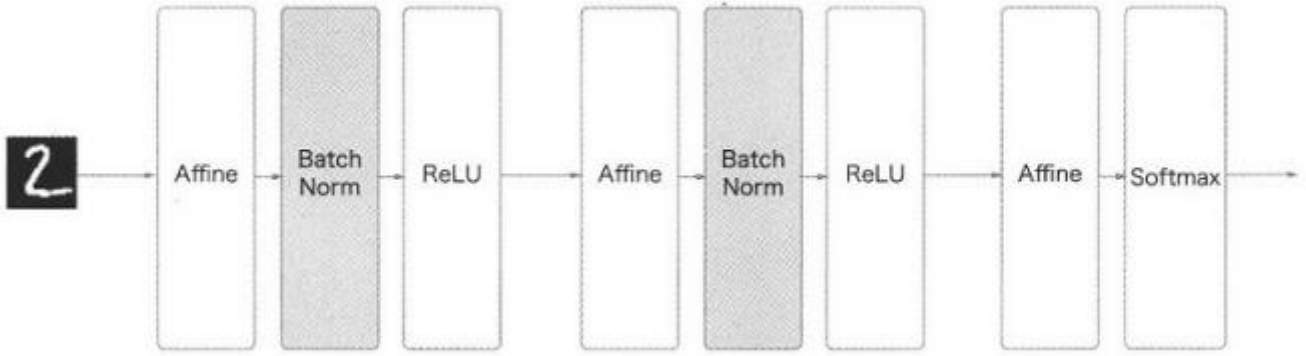
3. 배치 정규화

배치 정규화 Batch Normalization

각 층이 활성화를 적당히 퍼뜨리도록 '강제'

- 학습을 빠르게 진행 가능
- 초깃값에 의존X
- 오버피팅 억제

그림 6-16 배치 정규화를 사용한 신경망의 예



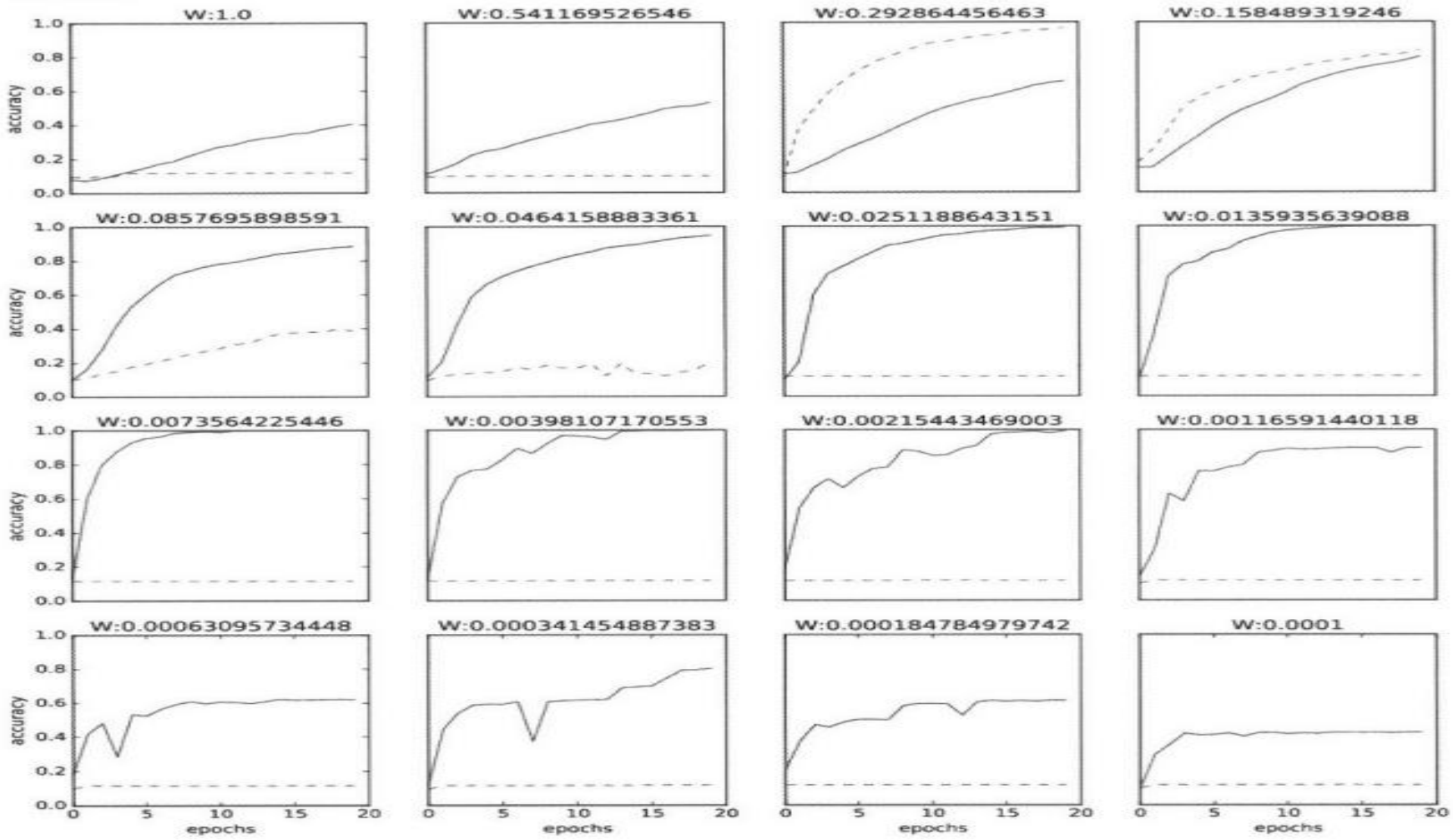
'배치 정규화 계층'을 신경망에 삽입
→미니배치 단위로 정규화

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$ 평균

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i-\mu_B)^2$$ 분산

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2+\varepsilon}}$$ 정규화

배치 정규화 Batch Normalization



실선 : 배치 정규화 O
점선 : 배치 정규화 X

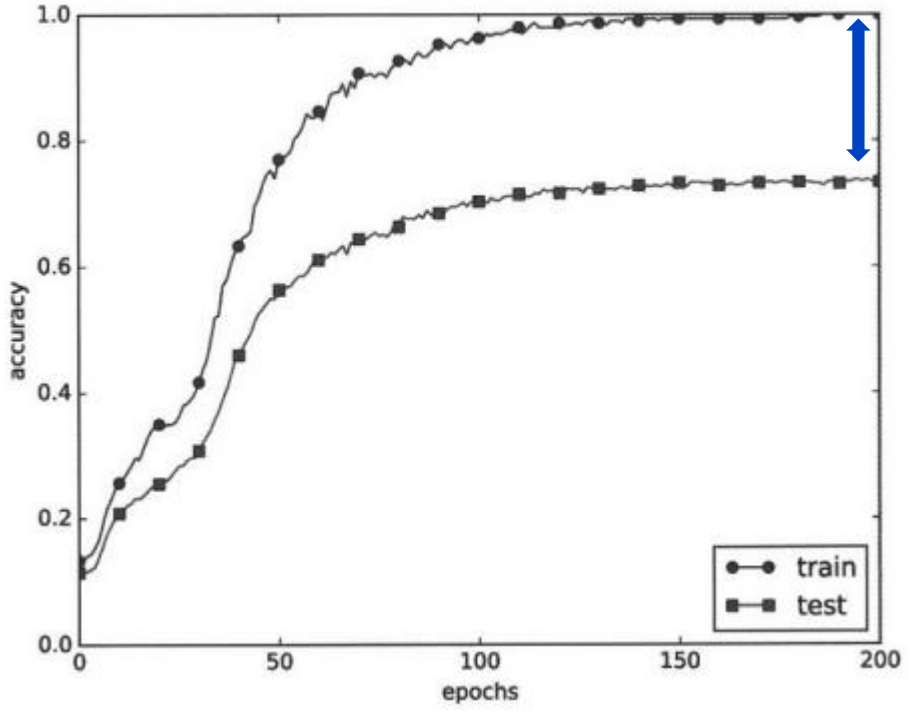
4. 바른 학습을 위해

오버피팅

신경망이 훈련 데이터에만 지나치게 적응하여 그 외의 데이터에는 제대로 대응하지 못하는 상태

- 매개변수가 많고 표현력이 높은 모델
- 훈련 데이터가 적음

그림 6-20 훈련 데이터(train)와 시험 데이터(test)의 에폭별 정확도 추이



가중치 감소

가중치에 비례하는 페널티를 부과하여 오버피팅을 억제하는 방법

∵ 오버피팅은 가중치 매개변수의 값이 커서 발생하는 경우↑

- 가중치의 L2 norm (각 원소 제곱들의 sum)을 손실 함수에 더함

⇒ 가중치가 커지는 것을 억제 가능

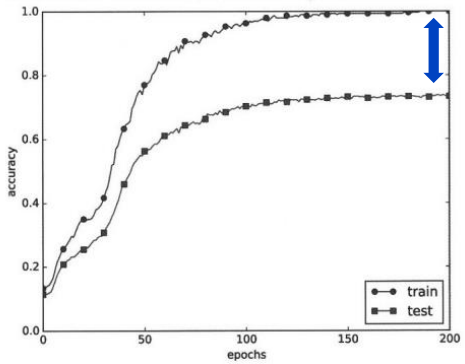
가중치 = W

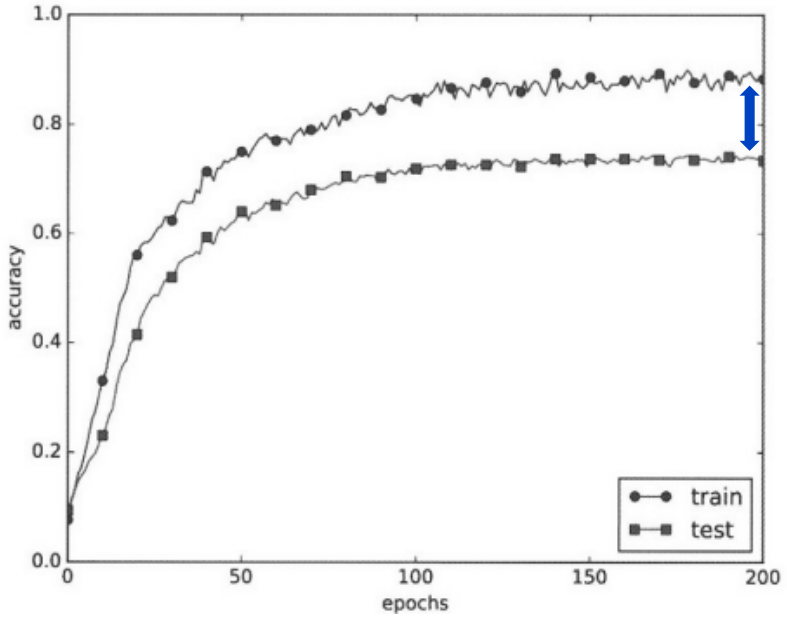
가중치 감소 = $1/2\lambda W^2$ => 손실함수에 더해줌

(λ : 정규화 세기 조절 하이퍼파라미터 , 크게 설정할수록 페널티가 커짐)

가중치 감소는 모든 가중치 각각의 손실함수에 $1/2\lambda W^2$을 더한다.

그림 6-20 훈련 데이터(train)와 시험 데이터(test)의 에폭별 정확도 추이





드롭아웃 Dropout

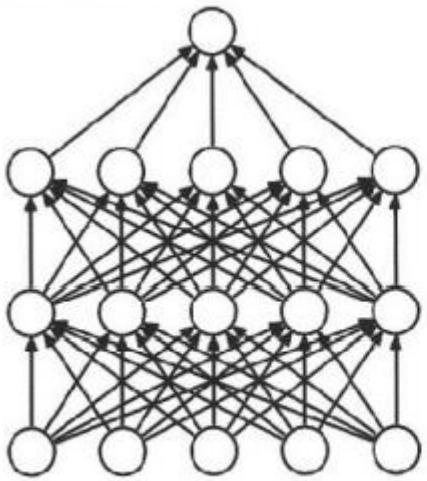
뉴런을 임의로 삭제하면서 학습하는 방법

- 신경망 모델이 복잡해질 때 유용

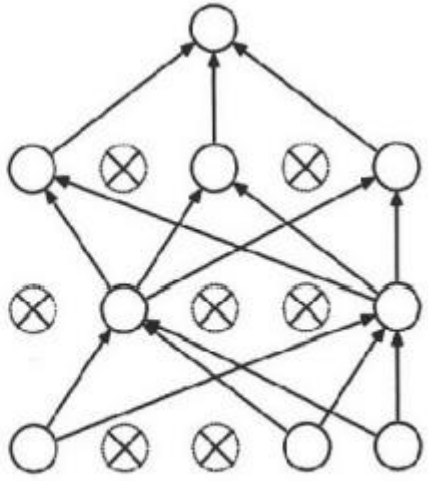
훈련 때, 은닉층의 뉴런을 무작위로 골라 삭제

시험 때, 모든 뉴런에 신호를 전달

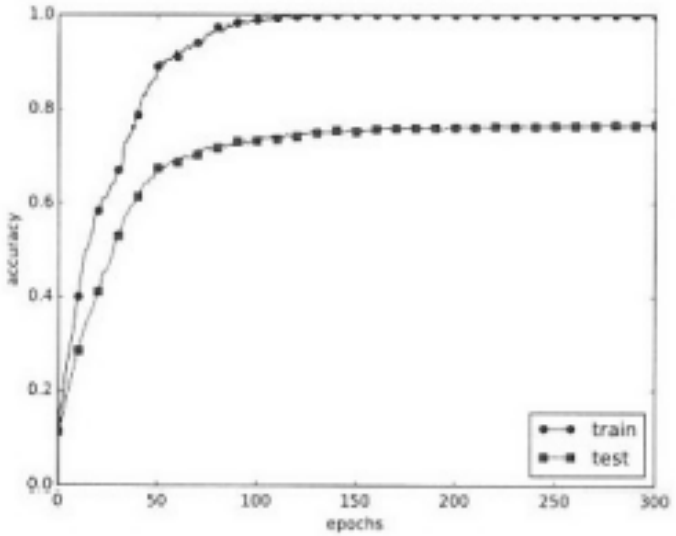
*삭제한 비율을 곱하여 출력

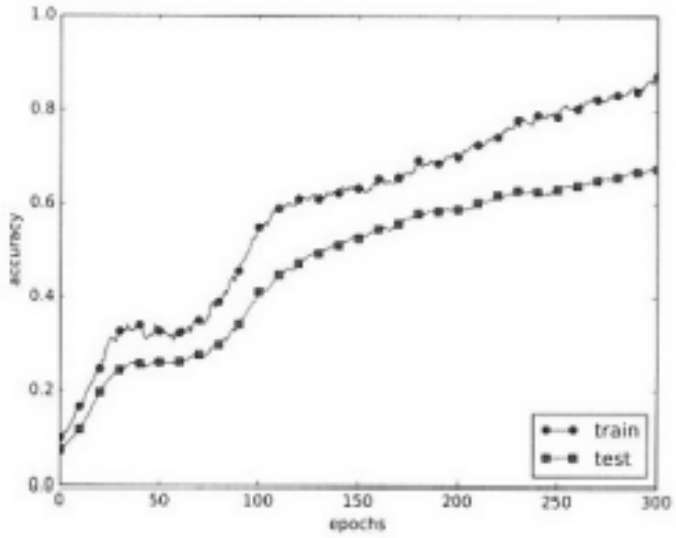


(a) 일반 신경망



(b) 드롭아웃을 적용한 신경망





5. 적절한 하이퍼파라미터 값 찾기

하이퍼파라미터

각 층의 뉴런 수, 배치 크기, 배개변수 갱신 시의 학습률, 가중치 감소 등

검증 데이터
: 하이퍼파라미터 조정용 데이터

- 하이퍼파라미터 성능 평가시, 시험 데이터 사용X

∵ 오버피팅 발생 가능

- 훈련 데이터 : 매개변수 학습
- 검증 데이터 : 하이퍼파라미터 성능 평가
- 시험 데이터 : 신경망의 범용 성능 평가

하이퍼파라미터 최적화

‘최적 값’이 존재하는 범위를 조금씩 줄여간다

-무작위로 샘플링해 탐색하는 것이 더 좋다

- **0단계**
  하이퍼파라미터 값의 범위를 설정합니다.
- **1단계**
  설정된 범위에서 하이퍼파라미터의 값을 무작위로 추출합니다.
- **2단계**
  1단계에서 샘플링한 하이퍼파라미터 값을 사용하여 학습하고, 검증 데이터로 정확도를 평가합니다(단, 에폭은 작게 설정합니다).
- **3단계**
  1단계와 2단계를 특정 횟수(100회 등) 반복하며, 그 정확도의 결과를 보고 하이퍼파라미터의 범위를 좁힙니다.

6장 내용 정리

- 매개변수 갱신 방법 : 확률적 경사 하강법 (SGD), 모멘텀, AdaGrad, Adam
- 가중치 초깃값을 정하는 방법은 올바른 학습을 하는 데에 중요하다.
- 가중치의 초깃값 : Xavier 초깃값, He 초깃값
- 배치 정규화를 이용하면 학습을 빠르게 진행할 수 있으며, 초깃값에 영향을 덜 받게 된다.
- 오버피팅을 억제하는 정규화 기술 : 가중치 감소, 드롭아웃
- 하이퍼파리미터 값 탐색은 최적 값이 존재할 법한 범위를 점차 좁히면서 하는 것이 효과적이다.

감사합니다 ☺