# Deep Learning 기초 4

# (딥러닝 학습의 효율과 정확도를 높이는 기술들 2)

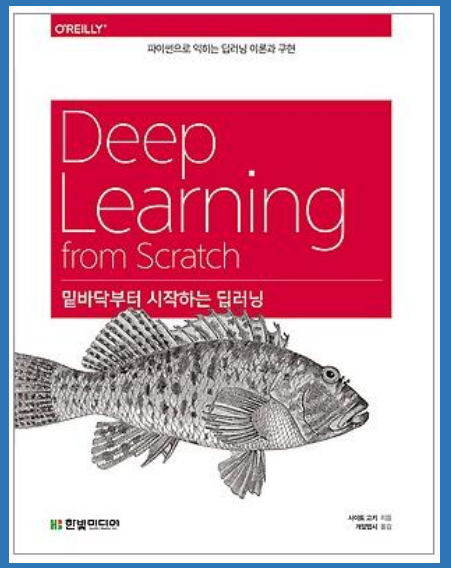
임세진

https://youtu.be/0wfxN46EuOU

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

# Contents




CryptoCraft LAB

# 01. 배치 정규화

<지난 세미나에서 마지막으로 설명한 부분>

- ✓ 가중치의 초깃값도 신경망 학습에서 중요
- ✓ 적절한 가중치 초깃값 ➔ 각 층의 활성화값 분포가 고르게 퍼짐 ➔ 원활한 학습
- ✓ 활성화함수의 종류에 따라 Xavier 초깃값과 He 초깃값이 있음

<의문사항>

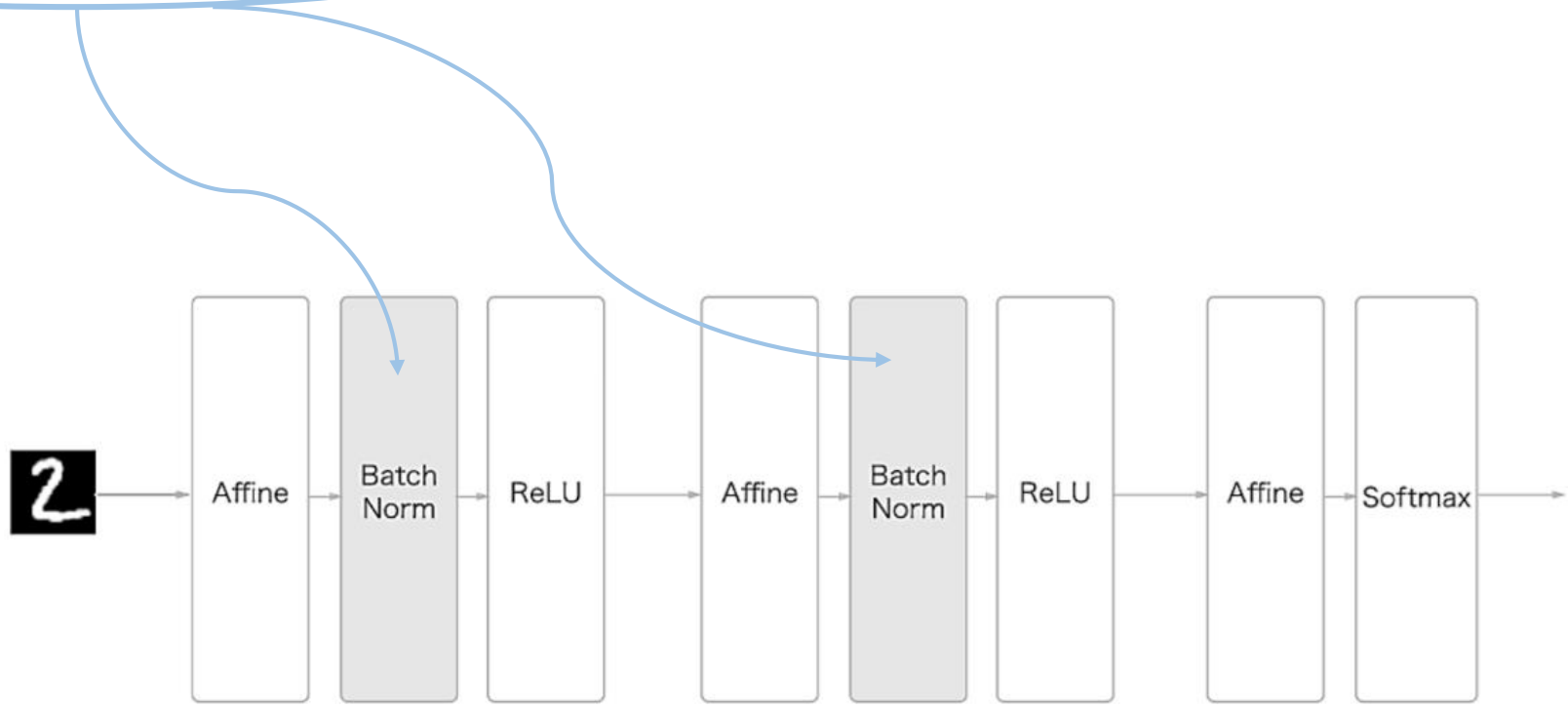
가중치 초깃값에 의존하지 않고 학습을 잘 진행할 수 있는 방법은 없을까?

# 01. 배치 정규화

- 배치 정규화 (Batch Normalization)

- 학습하는 과정 자체를 전체적으로 안정화하여 학습 속도를 가속시킬 수 있는 근본적인 방법

- 각 layer마다 정규화하는 layer를 두어, 변형된 분포가 나오지 않도록(활성화값이 적절히 분포되도록) 조절함



# 01. 배치 정규화

- **특징** (정규화를 하는 이유)
  - ✓ 학습을 빨리 진행할 수 있음 (학습 속도 개선)
  - ✓ 초깃값에 크게 의존하지 않음 (초깃값 선택을 위한 초기 과정 삭제)
  - ✓ 오버피팅 억제 (드롭아웃 등의 필요성 감소)
  - ➔ 딥러닝 학습에서의 부정적인 요소 제거

# 01. 배치 정규화

- **배치 정규화 알고리즘**

① 미니배치 단위로 정규화 [데이터 분포의 평균이 0, 분산이 1이 되도록 정규화]

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

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

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

이 과정을 활성화 함수의 앞/뒤에 삽입 ➔ 데이터 분포 치우침 방지

아주 작은 값 (0으로 나누게 되는 사태 방지)

② 각 정규화 계층마다 정규화된 해당 데이터에 고유한 확대(Scale)와 이동(Shift) 변환 수행

$y_i \leftarrow \gamma \hat{x}_i + \beta$

이동

확대

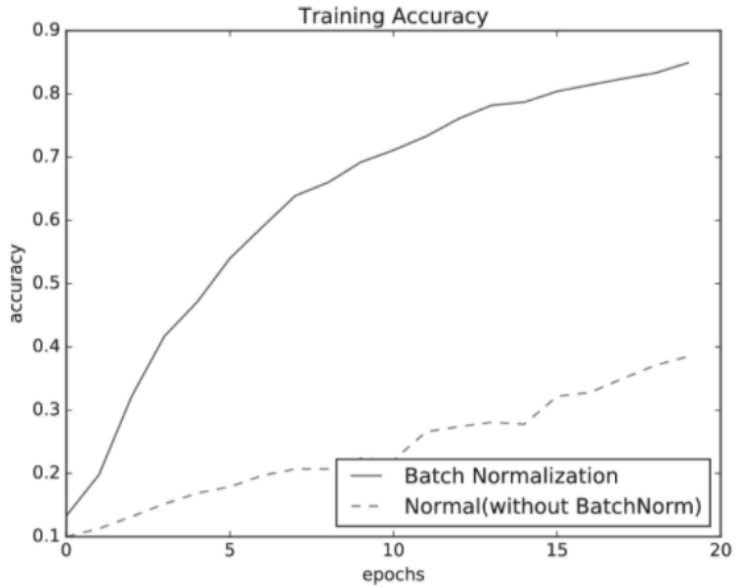
초깃값 $\gamma = 1,\ \beta = 0$ 부터 시작하여

학습하면서 적합한 값으로 조정

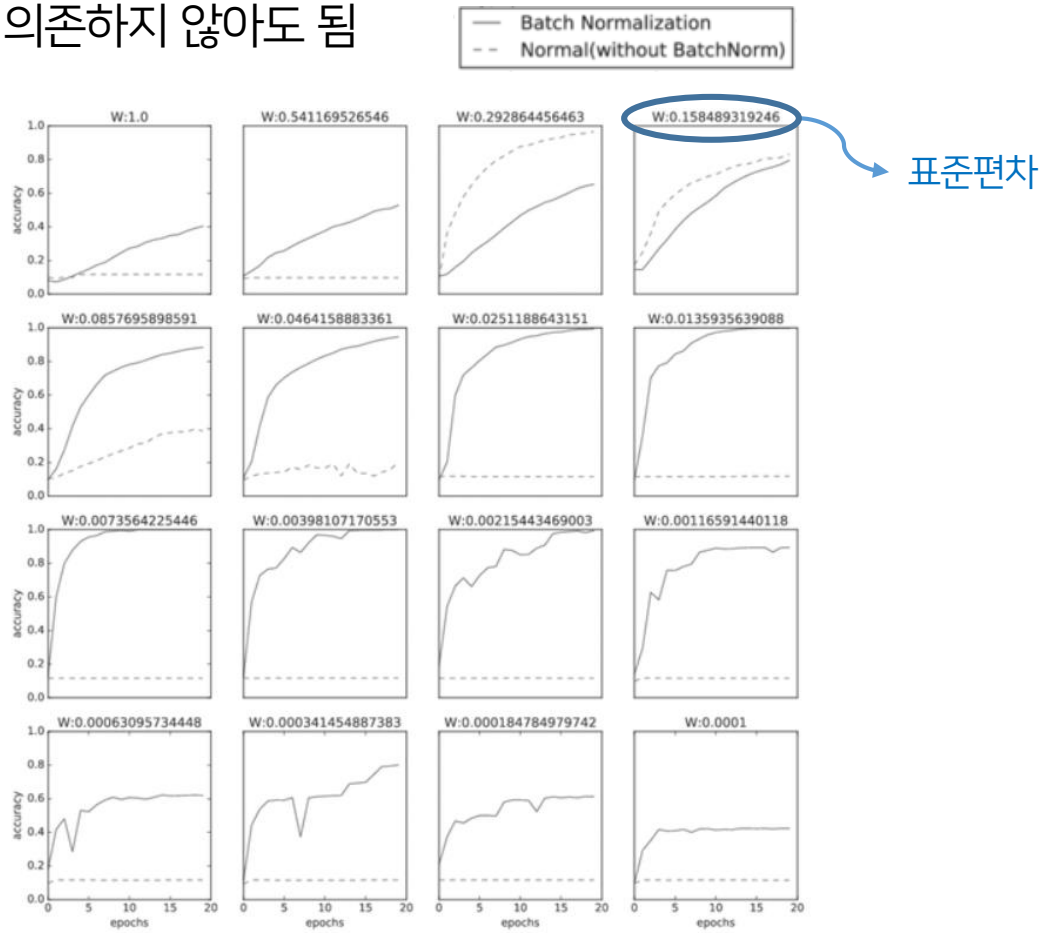
# 01. 배치 정규화

- **배치 정규화의 효과**

➔ 배치 정규화를 사용하면 학습이 빨라지고 가중치 초깃값에 크게 의존하지 않아도 됨




➔ 배치 정규화가 학습을 빨리 진전시킴을 알 수 있음




표준편차

가중치 초깃값의 표준편차에 변화를 주어 학습 경과를 관찰한 그래프

# 02. 오버피팅을 막는 방법

- **오버피팅 (과적합)**

- 신경망이 훈련 데이터에만 지나치게 적응되어 그 외의 데이터에는 제대로 대응하지 못하는 상태

- But 기계학습은 범용 성능을 지향함

  ✓ 훈련 데이터에 포함되지 않은 아직 학습하지 않은 데이터가 주어져도 바르게 식별해야 바람직한 모델임

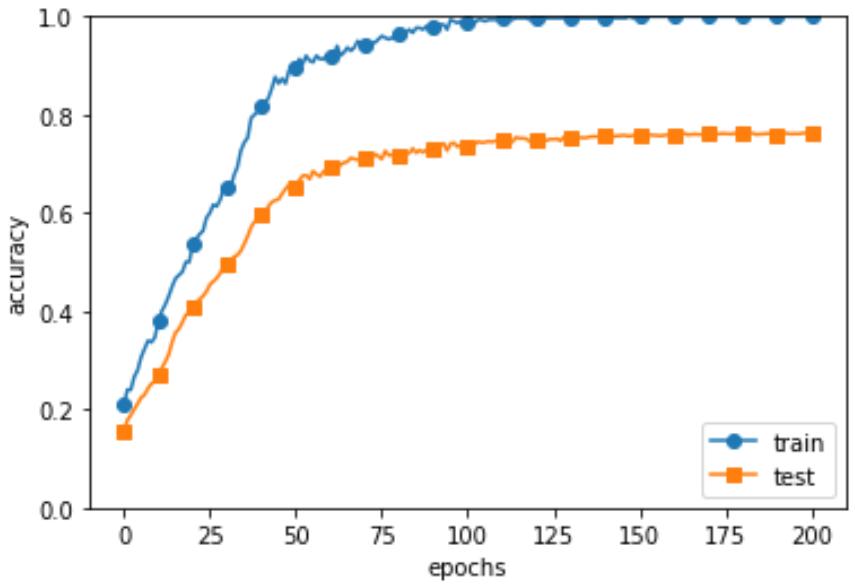
  ✓ 오버피팅을 억제하는 기술 중요 ★

# 02. 오버피팅을 막는 방법

- **오버피팅이 일어나는 경우**

- 매개변수가 많고 표현력이 높은 모델의 경우

- 훈련 데이터가 적은 경우




오버피팅이 발생한 상황

✓ MNIST 데이터셋의 훈련 데이터(6만개) 중 300개 사용

✓ 7층 네트워크를 사용하여 네트워크 복잡도 ↑

✓ 각 층 뉴런 100개, 활성화 함수 ReLU

➔ 훈련 데이터와 시험 데이터의 정확도 차이가 큼

➔ 오버피팅 발생 (모델이 훈련데이터에만 적응해버림)

# 02. 오버피팅을 막는 방법 – 가중치 감소

- **가중치 감소 (Weight decay)**

- 큰 가중치는 그에 상응하는 큰 페널티 부과 ➔ 오버피팅 억제

  ✓ 가중치 매개변수 값이 커서 오버피팅이 발생하는 경우 多

$W = (w_1, w_2, \ldots, w_n)$ 이 있을 때

L2 노름은 $\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}$

- 가중치의 제곱 노름(L2 노름)을 손실함수에 더해 가중치가 커지는 것을 억제

$$W \Rightarrow \frac{1}{2}\lambda W^2$$

L2 노름에 따른 가중치 감소

정규화의 세기를 조절하는 하이퍼파라미터

이 값이 클수록 큰 가중치에 대한 페널티도 ↑

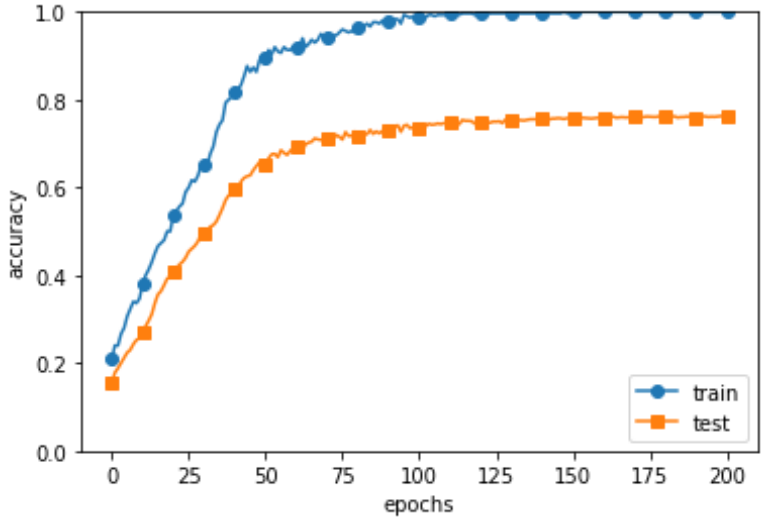
- 모든 가중치 각각의 손실함수에 $\frac{1}{2}\lambda W^2$ 을 더함

# 02. 오버피팅을 막는 방법 – 가중치 감소

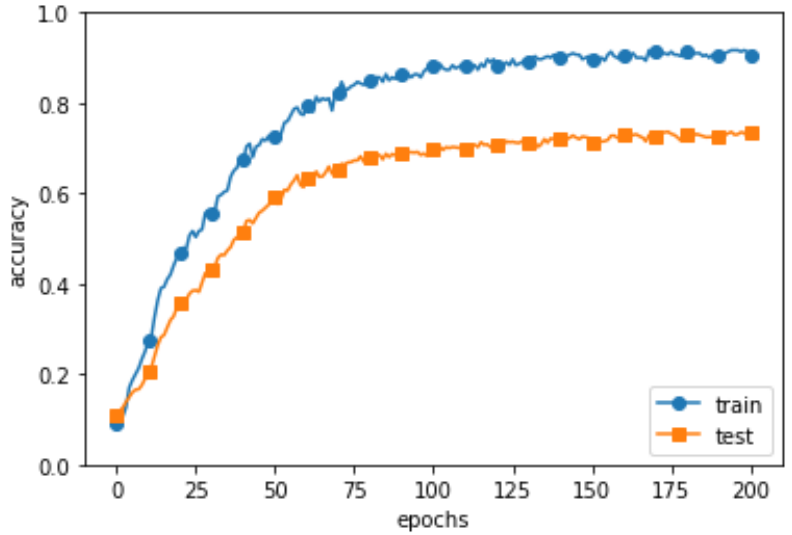
- 특징

  ✓ 구현이 간단

  ✓ 지나친 학습 억제




오버피팅이 발생한 모습




가중치 감소를 적용한 모습

➔ 훈련 데이터와 시험 데이터의 정확도 차이가 줄었음 (오버피팅 억제 효과)

➔ 훈련 데이터에 대한 정확도가 100%에 도달하지 못했음

# 02. 오버피팅을 막는 방법 – 드롭아웃

- **드롭아웃 (Dropout)**

- 신경망 모델이 복잡해질 경우 가중치 감소만으로는 오버피팅을 억제하기 어려움 ➔ 드롭아웃 기법 이용

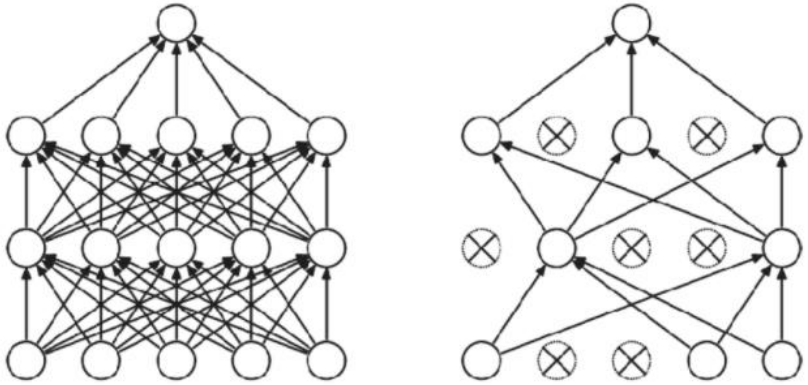
- 뉴런을 임의로 삭제하면서 학습하는 방법

  √ 훈련 시 은닉층의 뉴런을 무작위로 골라 삭제 ➔ 삭제된 뉴런은 신호 전달 X (순전파, 역전파 모두 적용 )

  √ Test 시 모든 뉴런에 신호 전달

  √ 단, Test 시에 각 뉴런의 출력에다가 훈련 때 삭제 안 한 비율을 곱하여 출력

- 무작위 삭제 ➔ 매번 다른 모델을 학습시키는 효과



(a) 일반 신경망

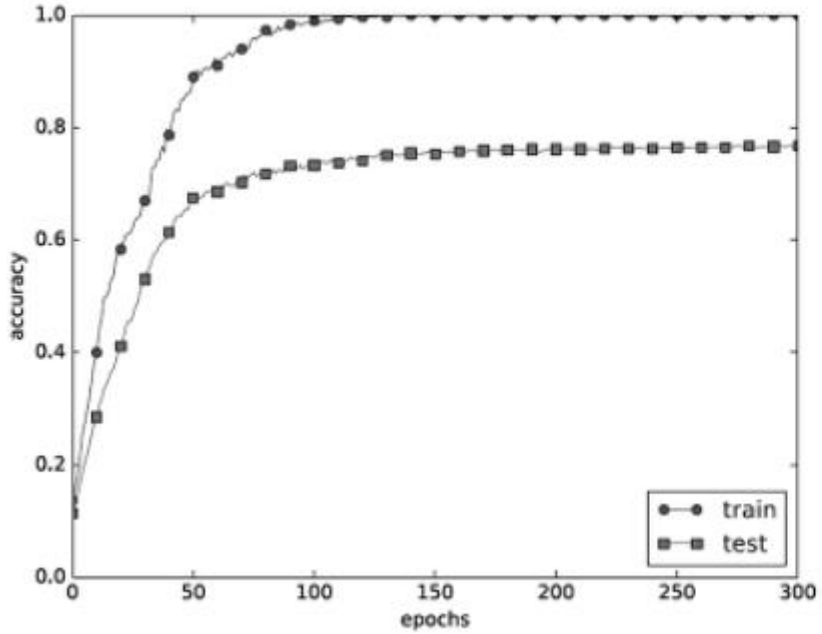


(b) 드롭아웃을 적용한 신경망

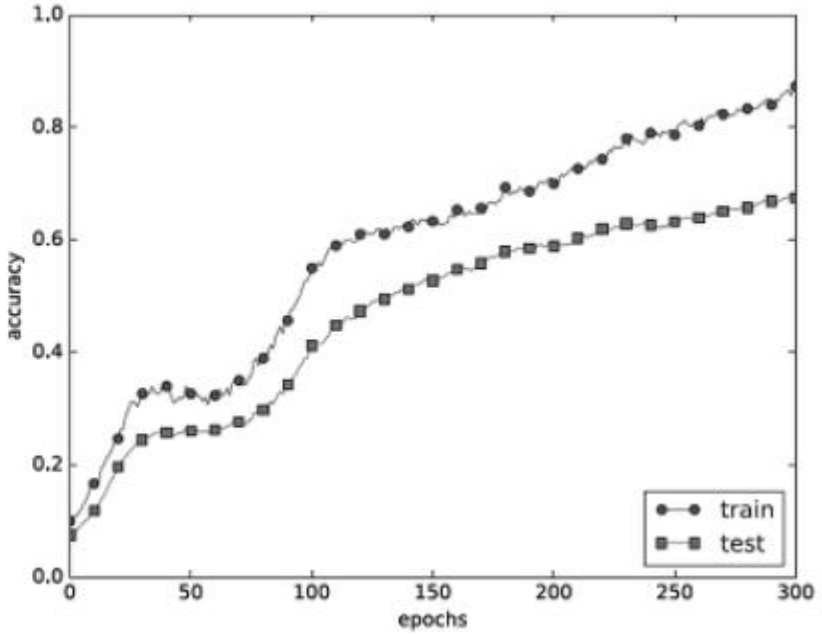
# 02. 오버피팅을 막는 방법 – 드롭아웃

- 드롭아웃을 이용하면 표현력을 높이면서도 오버피팅 억제 가능

✓ MNIST 데이터셋의 훈련 데이터(6만개) 중 300개 사용
✓ 7층 네트워크를 사용하여 네트워크 복잡도 ↑
✓ 각 층 뉴런 100개, 활성화 함수 ReLU




오버피팅이 발생한 모습




드롭아웃을 적용한 모습 (dropout_ratio = 0.15)

# 03. 효율적으로 하이퍼파라미터 값 찾기

- **하이퍼파라미터**

- 각 층의 뉴런 수, 배치 크기, 매개변수 갱신 시의 학습률과 가중치 감소 등이 해당됨

- 하이퍼파라미터의 값을 적절히 설정하지 않으면 모델의 성능이 크게 떨어질 수 있음

# 03. 효율적으로 하이퍼파라미터 값 찾기

- 검증 데이터 (Validation data)

- 훈련 데이터 : 매개변수(가중치와 편향) 학습용 , 시험 데이터 : 신경망의 범용 성능 평가

- 하이퍼파라미터의 성능 평가용 데이터 (하이퍼파라미터 전용)

  √ 하이퍼파라미터의 성능 평가 시 시험 데이터를 사용해서는 안됨

  √ 하이퍼파라미터 값이 시험 데이터에 오버피팅 되기 때문 ➔ 범용 성능이 떨어지는 모델이 될 수 있음

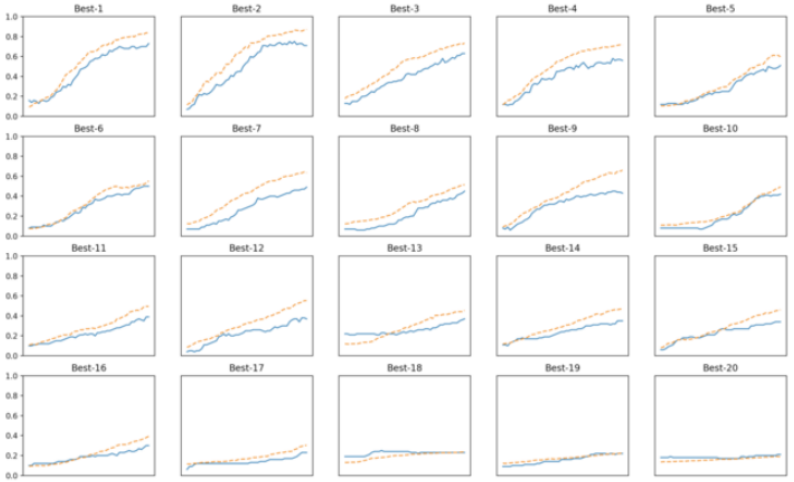
- 보통 훈련 데이터의 20%를 검증 데이터로 분리함

# 03. 효율적으로 하이퍼파라미터 값 찾기

- **하이퍼파라미터 최적화**

- 하이퍼파라미터의 '최적 값'이 존재하는 범위를 줄여나가는 것이 핵심

1. 하이퍼파라미어 값의 대략적인 범위 설정 ($10^{-3}$ ~$10^{3}$ 로그 스케일)
2. 설정된 범위 내에서 무작위로 하이퍼파라미터 값 추출 (샘플링)
3. 해당 값으로 학습 후, 검증 데이터로 정확도를 평가 (시간 단축을 위해 에폭을 작게 설정)
4. 정확도에 따라 위 작업(2-3단계)을 반복하여 범위를 좁혀감
5. 압축된 범위에서 값을 하나 선택




학습이 잘 진행될 때의 학습률 : 0.001 ~ 0.01

가중치 감소 계수 : $10^{-8}$ ~$10^{-6}$

```
val acc:0.83 | lr:0.007930777133653748, weight decay:4.285774741857826e-08
val acc:0.08 | lr:2.622049738234655e-06, weight decay:4.940041697515601e-06
val acc:0.49 | lr:0.001976012154057031, weight decay:1.407891931740619e-08
val acc:0.07 | lr:2.059835391667847e-05, weight decay:1.4266563733468448e-06
val acc:0.56 | lr:0.004430642268141839, weight decay:3.5664565404497015e-05
val acc:0.13 | lr:0.00010846540334534612, weight decay:5.388291372869288e-05
val acc:0.15 | lr:0.000416250351827967, weight decay:1.9089887539335125e-06
val acc:0.14 | lr:1.5146890150683426e-06, weight decay:9.314904481576831e-05
val acc:0.83 | lr:0.006912744489471699, weight decay:9.73346368878762e-08
```

감사합니다