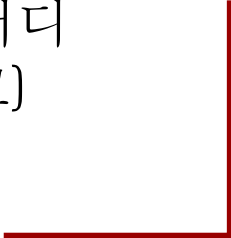


# TRAINING NEURAL NETWORK

딥러닝 개론 스터디  
(뉴럴 네트워크)



1

## INTRODUCTION

- overfitting 문제 및 해결방안
- too slow 문제 및 해결방안
- vanishing gradient 문제 및 해결방안

2

## 학습을 위한 기술

- 미니배치
- 가중치의 초기값
- 배치 정규화
- 오버피팅 방지

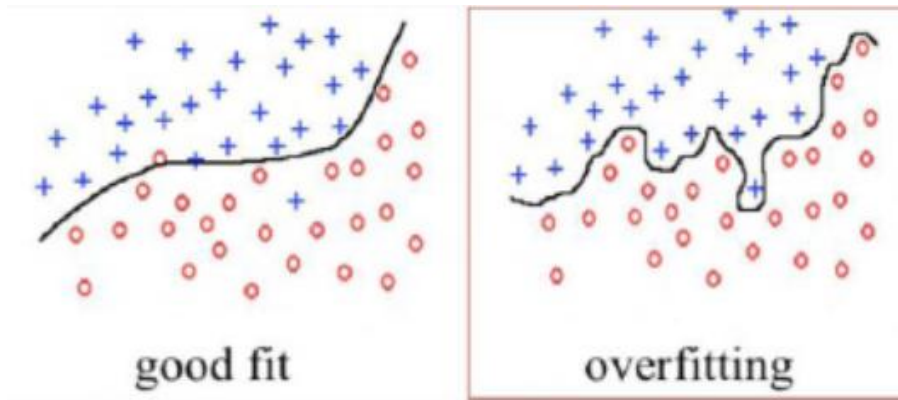
# 1. INTRODUCTION

---

## 1.1 overfitting: too many neurons

Overfitting

Train data만 잘 맞게 형성 되어 그 외의 데이터에 제대로 대응하지 못하는 상태



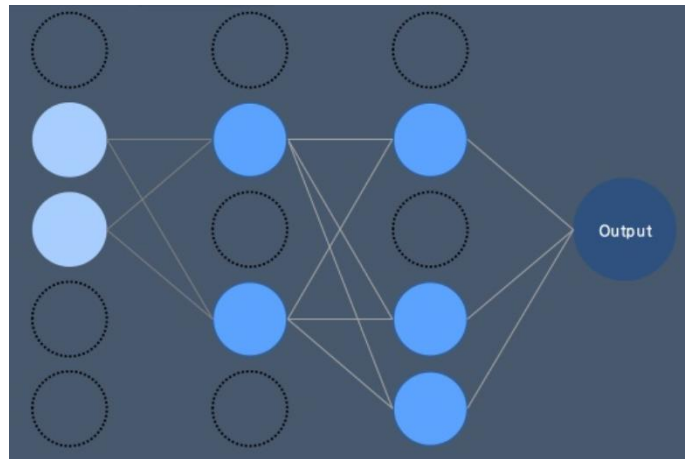
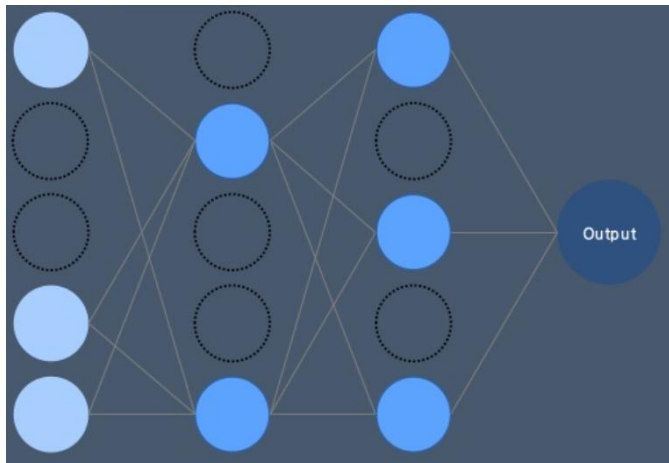
# 1. INTRODUCTION

---

## 1.1 overfitting: too many neurons

### DROP OUT

뉴런들을 랜덤하게 버린다.



# 1. INTRODUCTION

---

## 1.2 too slow: 계산이 오래 걸린다

- 딥러닝에서는 gradient descent를 이용하면 시간이 너무 오래 걸림

- 다른 optimizer들 사용

- ex)SGD: 확률적경사하강법(무작위로 고른 데이터에만 경사하강법 시행)  
adam, adagrad, adadelata, RMSprop, momentum 등등

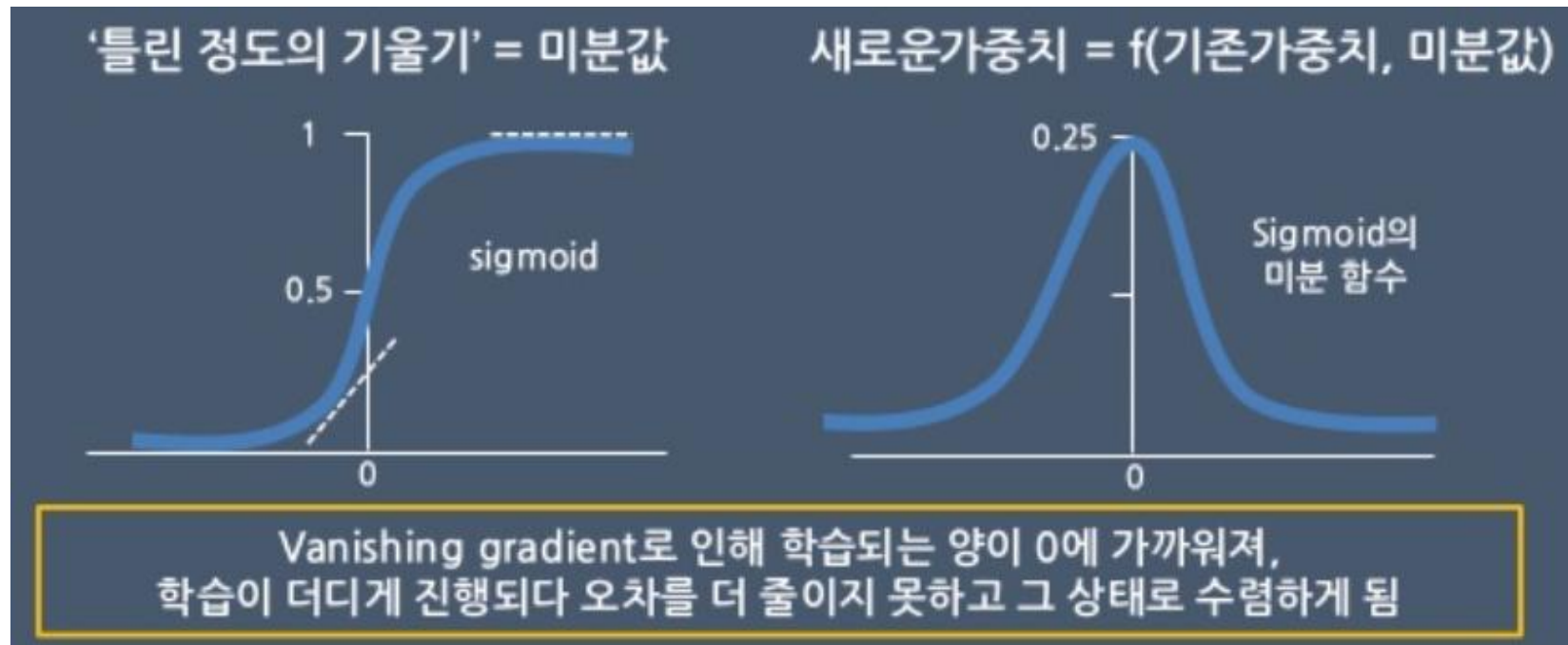
# 1. INTRODUCTION

## 1.3 vanishing gradient: gradient가 사라진다(기울기가 사라진다)



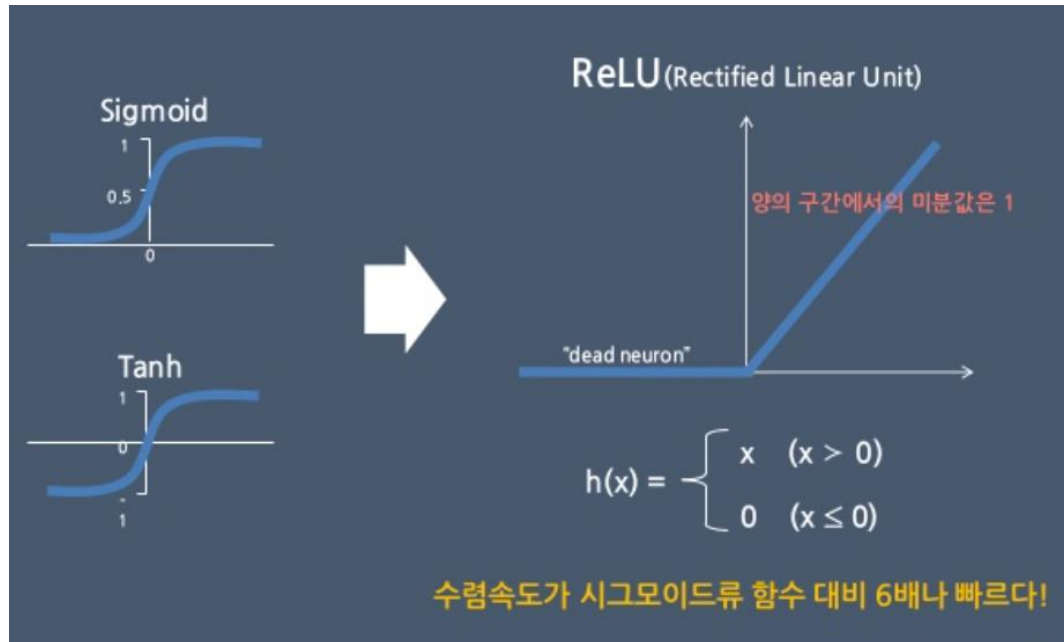
# 1. INTRODUCTION

## 1.3 vanishing gradient: gradient가 사라진다(기울기가 사라진다)



# 1. INTRODUCTION

## 1.3 vanishing gradient: gradient가 사라진다(기울기가 사라진다)



-sigmoid 함수를 활성화 함수로 사용하면 기울기가 사라져서 학습이 더디게 되면서 문제가 발생

-활성화 함수로 ReLU 함수를 사용해서 문제를 해결 가능  
but 음수 입력값이 주어지면 미분계수가 0이 되어 학습이 이루어지지 않음



## 2. 학습을 위한 기술

---

### 2.1 미니배치

-미니 배치란?

**train data**에서 일부를 추린 것(수십, 수백 개의 데이터)

-**train data**에 **data**가 너무 많아서 각각의 손실함수 모두 구해서  
합치기는 거의 불가능하기 때문에 미니 배치를 뽑아서 학습하는 것

## 2. 학습을 위한 기술

### 2.1 미니배치



$$E = -\frac{1}{N} \sum_n \sum_k t_{nk} \log y_{nk}$$

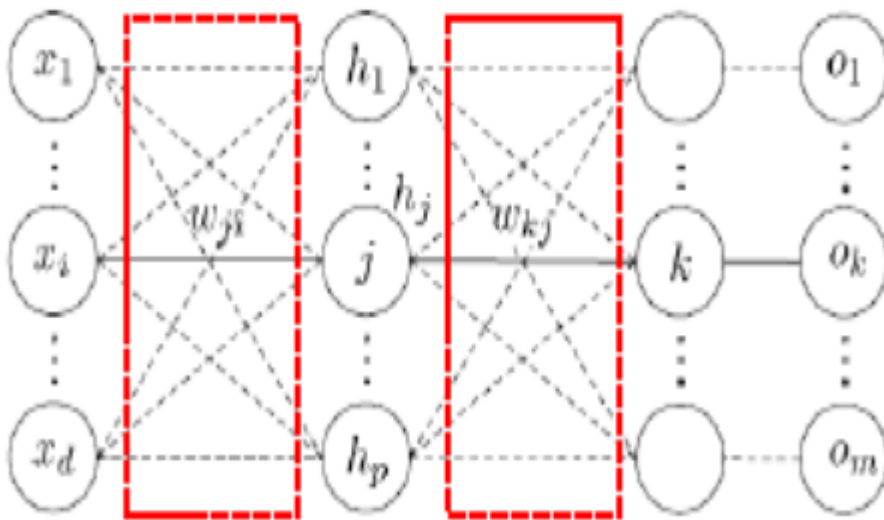
$t_{nk}$  : n번째 데이터의 정답 레이블

$y_{nk}$  : n번째 데이터의 신경망 출력

학습에 드는 시간을 줄이면서 최대한 많은 **data** 반영

## 2. 학습을 위한 기술

### 2.2 가중치의 초기값



-**cost function** 최소화하는 가중치 찾는 것이 목표

-**backpropagation** 통해 가중치 갱신하며 학습

-최초의 가중치는 어떻게 정해야 할까?

## 2. 학습을 위한 기술

---

### 2.2.1 초기값을 동일하게 하면?

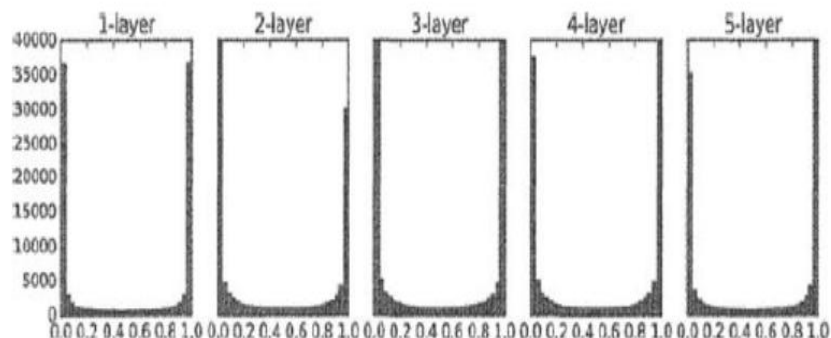
가중치  $W$ 를 모두 같은 값으로 하고 시작하면 두 번째 뉴런에 모두 같은 값이 전달



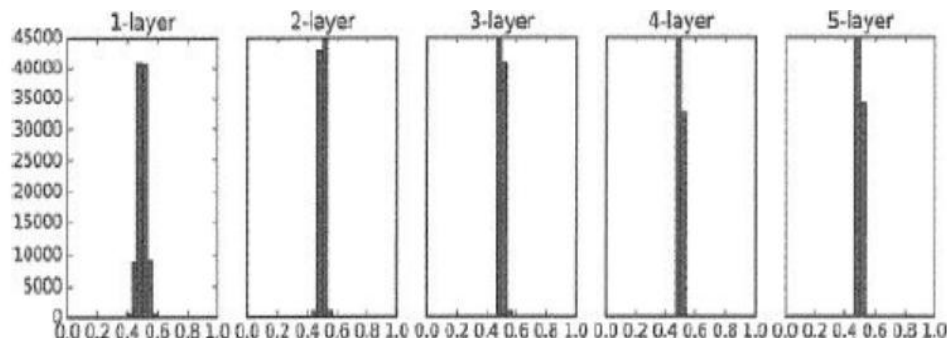
뉴런의 개수와 상관없이 모두 같은 값을 출력하며 의미가 없어짐

## 2. 학습을 위한 기술

### 2.2.2 은닉층의 활성화값 분포



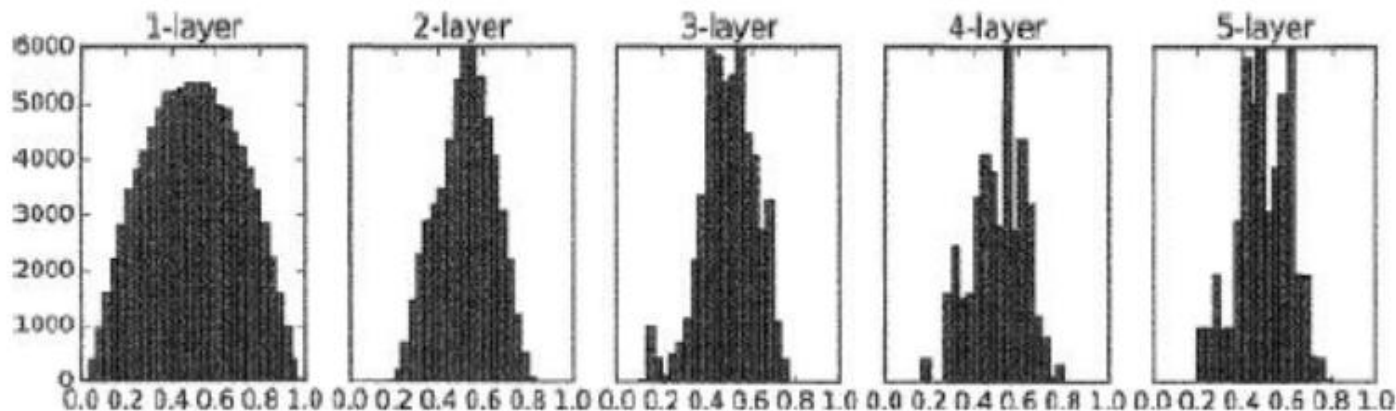
가중치 분포: 표준편차가 1인 정규분포 사용  
➡ 0과 1에 치우침, vanishing gradient



가중치 분포: 표준편차가 0.01인 정규분포 사용  
➡ 0.5부근에 집중, 표현력 제한

## 2. 학습을 위한 기술

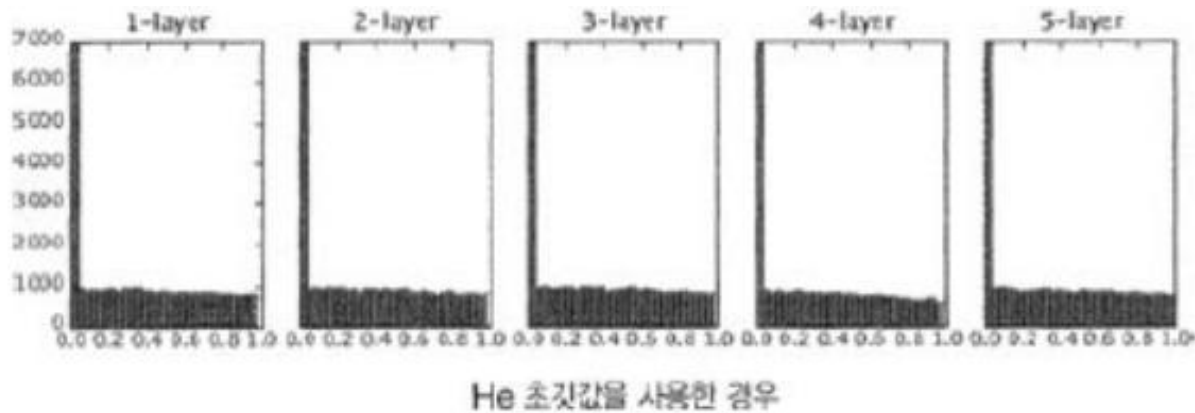
### 2.2.3 Xavier 초기값



**Xavier 초기값:** 앞 계층이  $n$ 개의 노드를 가진다면 가중치의 초기값을  $1/\sqrt{n}$ 을 표준편차로 가지는 정규분포를 사용하는 것 (**sigmoid, tanh** 함수에 적합)

## 2. 학습을 위한 기술

### 2.2.4 He 초기값



**He 초기값:** 더 넓게 분포시키기 위해서 **Xavier** 초기값 표준편차에 2 곱한  $2/\sqrt{n}$ 을 표준편차로 가지는 정규분포 사용(ReLU 함수에 적합)

## 2. 학습을 위한 기술

---

### 2.3 배치 정규화

배치 정규화란?

-각 층이 활성화 값이나 데이터가 표준정규분포를 갖도록 강제하는 것

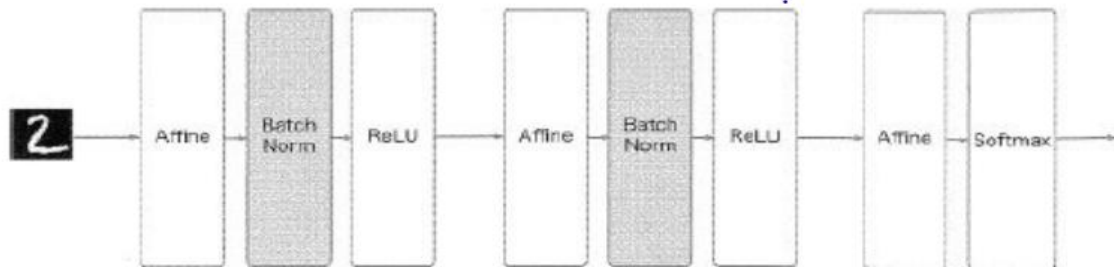
배치 정규화의 장점

1. 학습 속도가 빠르다
2. 초기값에 크게 의존하지 않는다
3. 오버피팅을 억제한다



## 2. 학습을 위한 기술

### 2.3 배치 정규화

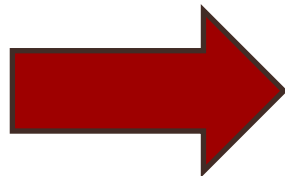
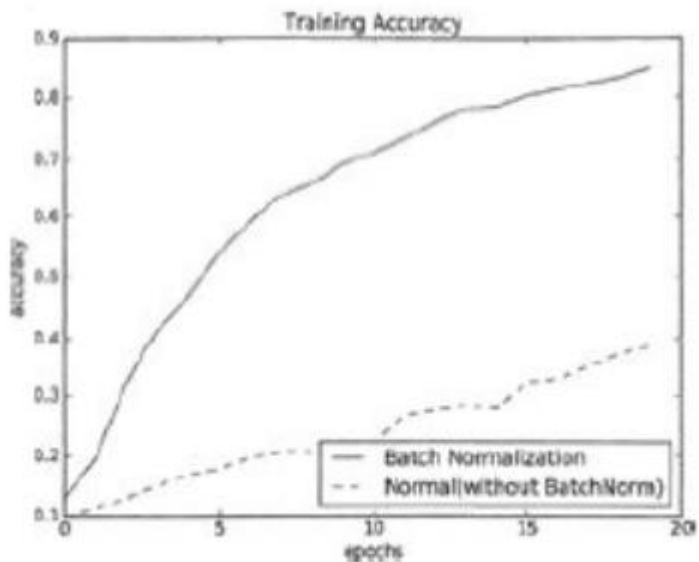


-데이터 분포를 정규화하는 ‘배치 정규화’ 계층을 삽입

-미니배치를 단위로 평균이 0, 분산이 1이 되도록 정규화

## 2. 학습을 위한 기술

### 2.3 배치 정규화



배치 정규화가 학습을  
빨리 진전시킨다.

## 2. 학습을 위한 기술

---

### 2.4 overfitting 해결책-가중치 감소

- 가중치 감소: 큰 가중치에 그에 상응하는 큰 페널티를 부여
- L2 regularization

$$W = [w_1 \quad w_2 \quad \dots \quad w_n]$$

$$L_{new} = L_{old} + \frac{\lambda}{2}(w_1^2 + w_2^2 + \dots + w_n^2)$$

-1/2를 곱한 것은 미분의 편의성 고려

-람다: 페널티의 세기 결정하는  
hyper-parameter



# 감사합니다