

DEEPLARNING WITH PYTORCH

PART I

ABOUT DEEP LEARNING

ABOUT DEEP LEARNING

3

◆ 딥러닝이란

- 머신 러닝의 한 분야
 - 인간의 신경망인 뉴런(nerun)을 기반으로 구성된 인공 신경망
 - 학습 데이터를 구분하는 층(Layer)을 많이 만들어 그 정확도를 올리는 방법
-
- 2016년 알파고에 적용
 - 영상 인식, 음성 인식, 자연 언어 처리 등 분야에서 우수한 성능 발휘

ABOUT DEEP LEARNING

4

◆ H/W 사양 - CPU & GPU

CPU	<ul style="list-style-type: none">- Central Processing Unit 약자- 명령어 입력된 순서대로 데이터 처리하는 직렬(순차) 처리 방식에 특화된 구조- 명령어 : 한 번에 한 가지 처리- 실수 저장 방식 : 고정소수점 방식- 연산 : 더하기, 빼기 연산 빠르게 수행
GPU	<ul style="list-style-type: none">- Graphics Processing Unit 약자- 여러 명령어를 동시에 처리하는 병렬 처리 방식- 연산 수행 ALU가 코어당 수백~수천개 장착- 실수 저장 방식 : 부동소수점 연산 수행- 연산 : 그래픽, 음성 등 멀티미디어, 3차원 데이터 빠르게 처리

ABOUT DEEP LEARNING

5

◆ H/W 사양 - CPU & GPU

GPGPU	<ul style="list-style-type: none">- General Perpose computing on GPU 약자- GPU를 프로그램 가능하도록 구성한 구조- 그래픽처리, 행렬, 벡터연산, 병렬처리를 묶어 고속 계산 수행
CUDA	<ul style="list-style-type: none">- Compute Unified Device Architecture 약자- 그래픽 처리 장치에서 수행하는 알고리즘을 C프로그래밍 언어를 비롯한 산업 표준언어 사용하여 작성할 수 있도록 하는 GPGPU기술- 2006년 NVIDIA에서 개발한 GPGPU 프로그래밍 도구- NVIDIA GPU와 특별한 스트림 처리 드라이버 필요
cuDNN	<ul style="list-style-type: none">- cuda Deep Neural network Library 약자- TensorFlow, PyTorch, Theano 등 대중적으로 널리 사용되는 딥러닝 프레임워크 가속화시켜주는 라이브러리

ABOUT DEEP LEARNING

6

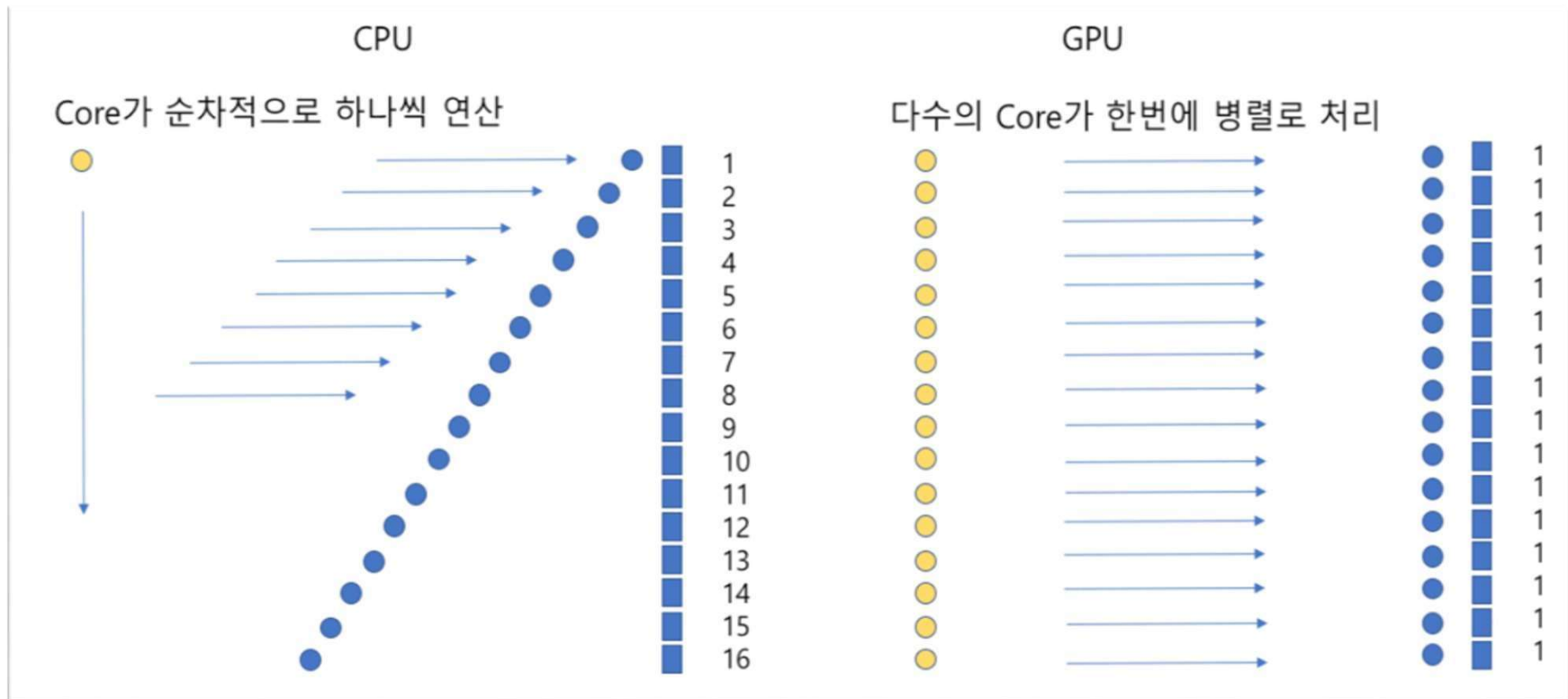
◆ H/W 사양 - CPU & GPU



ABOUT DEEP LEARNING

7

◆ H/W 사양 - CPU & GPU



ABOUT DEEP LEARNING

9

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 퍼셉트론(Perceptron)

- 1945년 개념 제안, 1958년 구체적인 공학적 구현 제안
- 생물학적 뉴런을 공학적인 구조로 변형한 것
- 1969년 XOR문제 해결할 수 없는 단순선형분리기 불과함 증명 → 인기하락
- 논리적인 추론으로 인공지능 트렌드 변화
- 최초 인공신경망 개념을 공학적 구조로 구현한 것으로 큰 의미

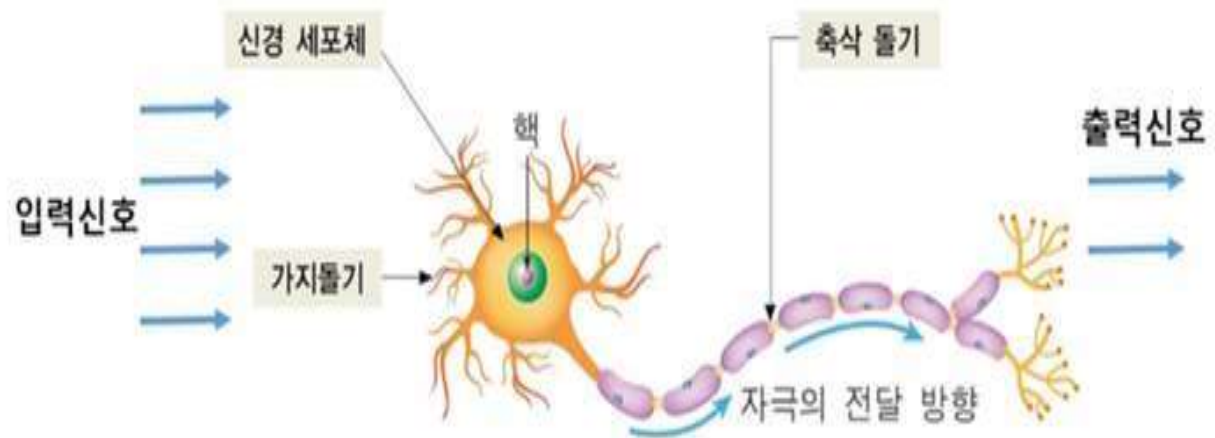
ABOUT DEEP LEARNING

10

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 뉴런(Neuron) 구조

- 600억~1,000억 개 뉴런 + 수 백조 이상의 시냅스



ABOUT DEEP LEARNING

11

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 뉴런(Neuron) 동작 원리

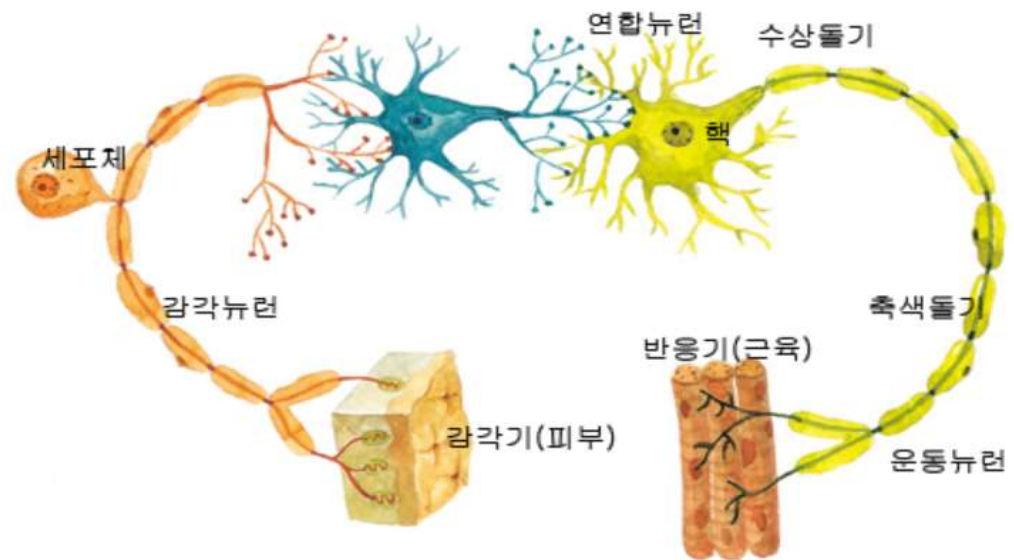
- ① 자극이 들어왔을 때 가지돌기와 축삭돌기 거쳐 시냅스 통해 전기적인 신호 사용 통신
- ② 들어오는 자극이 일정하더라도 각 신경 세포마다 자극에 반응하는 정도가 조금씩 다름
- ③ 자극을 강화시켜 전달하기도 하고 축소시켜 전달하기도 함

ABOUT DEEP LEARNING

12

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 뉴런(Neuron) 자극 반응

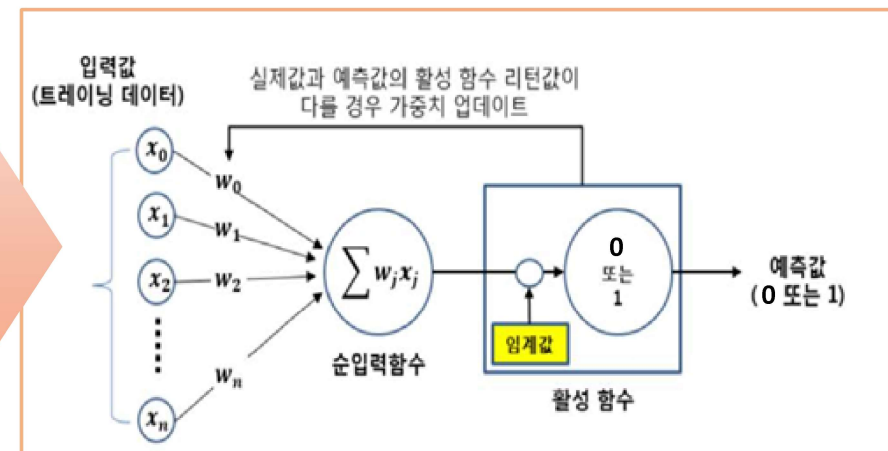
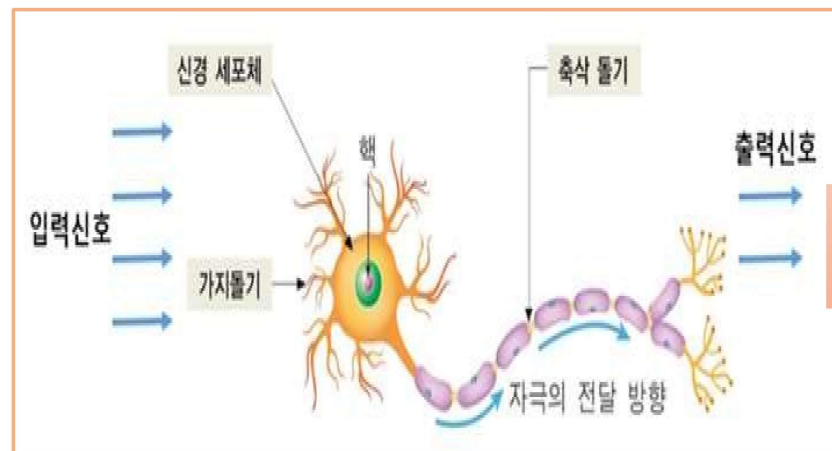


ABOUT DEEP LEARNING

13

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 퍼셉트론(Perceptron) 구조

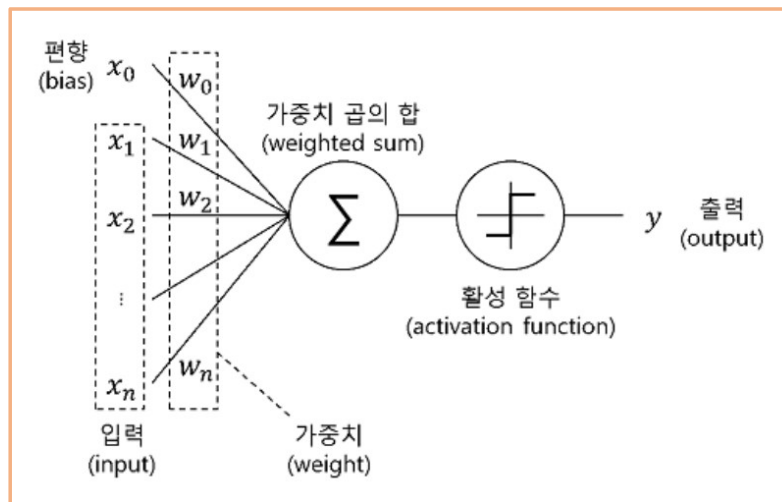


ABOUT DEEP LEARNING

14

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 퍼셉트론(Perceptron) 구조



$$\sum_{i=0}^{N-1} w_i x_i = \mathbf{w} \cdot \mathbf{x} = \mathbf{w}^T \mathbf{x}$$
$$y = a \left(\underbrace{\sum_{i=0}^{N-1} w_i x_i + b}_{\text{활성함수 두 벡터의 내적}} \right) \quad \text{편향(Bias)}$$

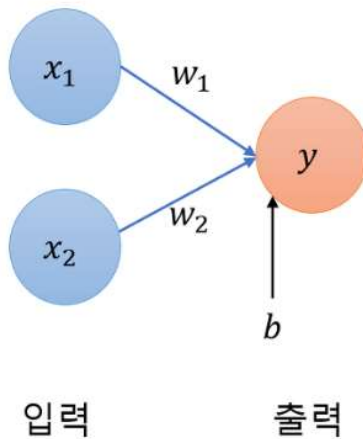
$$y = a(\mathbf{w}^T \mathbf{x} + b)$$

ABOUT DEEP LEARNING

15

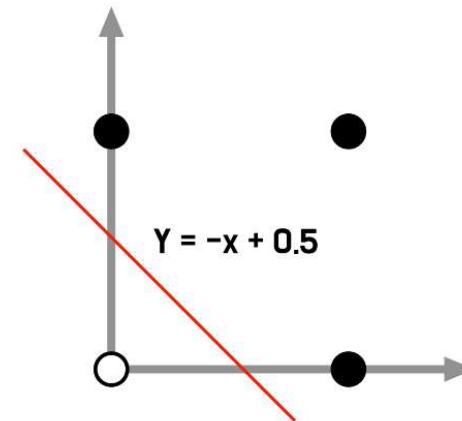
◆ 인공신경망 (ANN : Artificial Neural Network)

■ 퍼셉트론(Perceptron) 증명



OR Gate

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

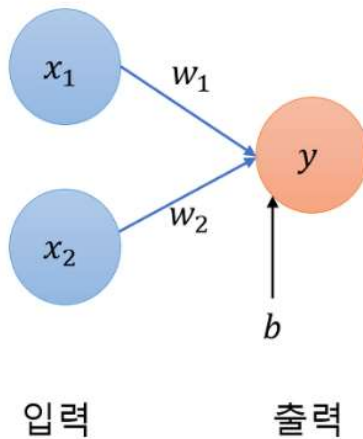


ABOUT DEEP LEARNING

16

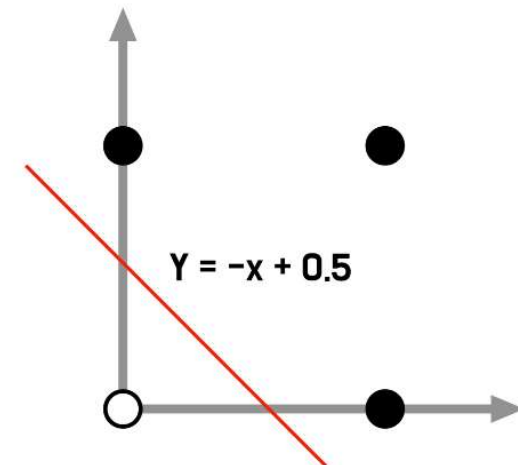
◆ 인공신경망 (ANN : Artificial Neural Network)

■ 퍼셉트론(Perceptron) 증명



OR Gate

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

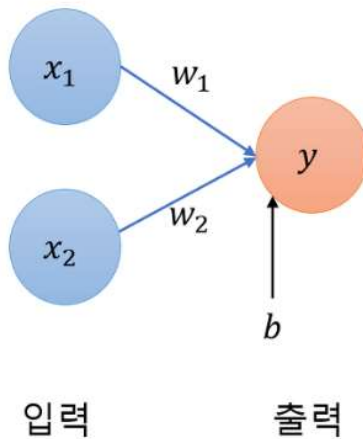


ABOUT DEEP LEARNING

17

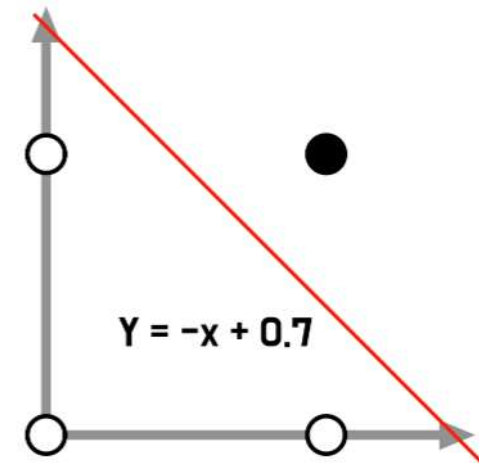
◆ 인공신경망 (ANN : Artificial Neural Network)

■ 퍼셉트론(Perceptron) 증명



AND Gate

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

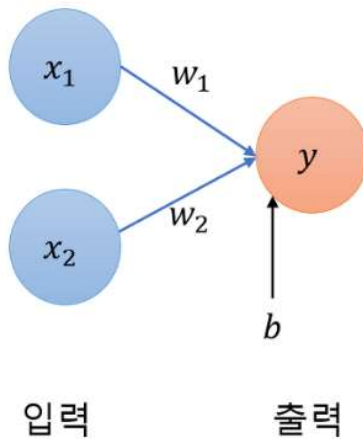


ABOUT DEEP LEARNING

18

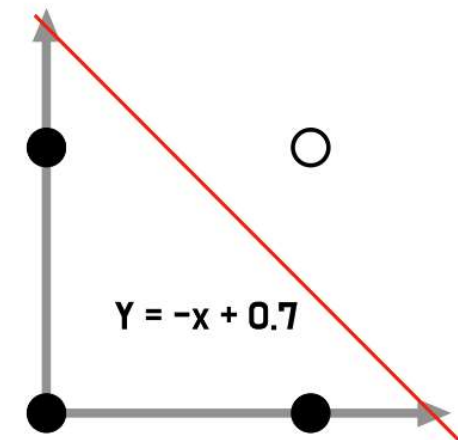
◆ 인공신경망 (ANN : Artificial Neural Network)

■ 퍼셉트론(Perceptron) 증명



NAND Gate

X1	X2	Y
0	0	1
0	1	1
1	0	1
1	1	0

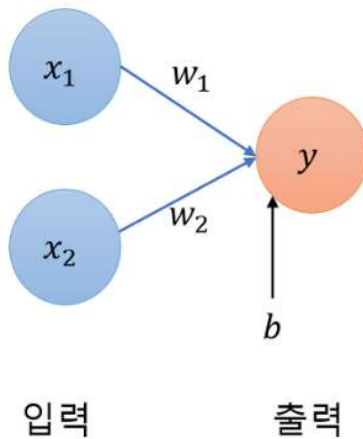


ABOUT DEEP LEARNING

19

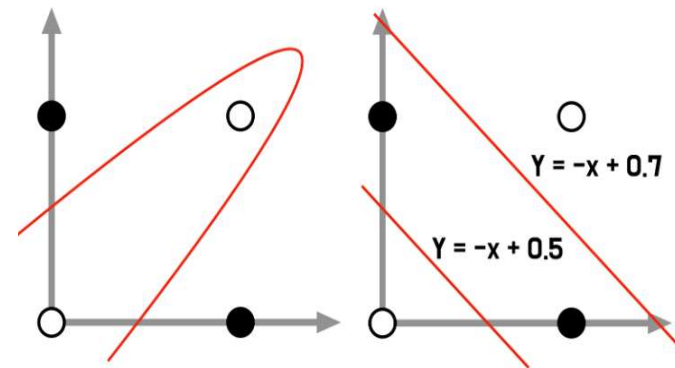
◆ 인공신경망 (ANN : Artificial Neural Network)

■ 퍼셉트론(Perceptron) 증명



XOR Gate

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0



ABOUT DEEP LEARNING

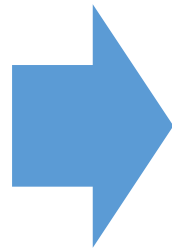
20

◆ 인공신경망 (ANN : Artificial Neural Network)

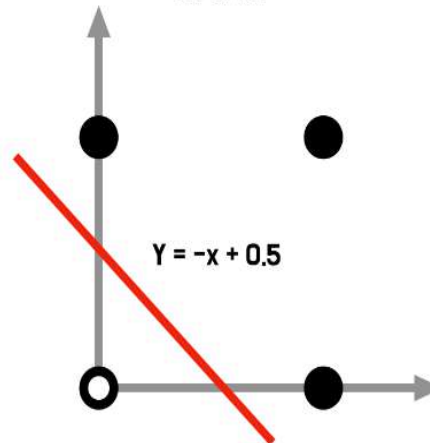
■ 퍼셉트론(Perceptron) 증명

XOR Gate

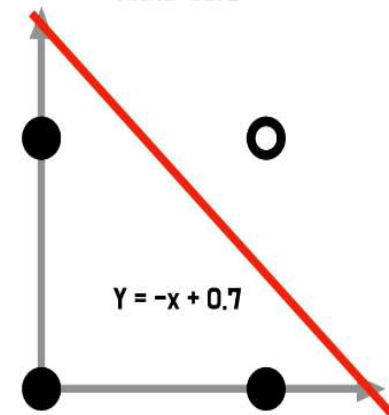
X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0



OR Gate



NAND Gate

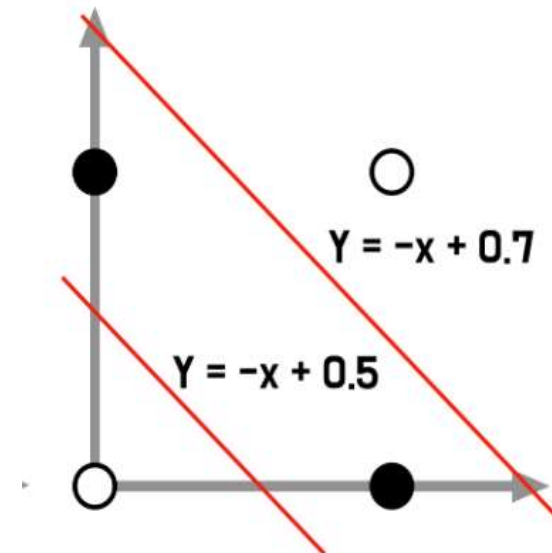
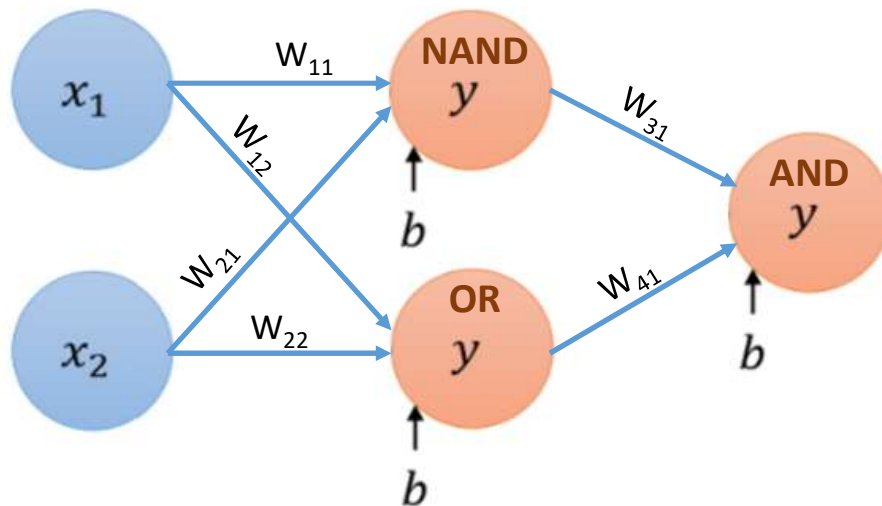


ABOUT DEEP LEARNING

21

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 퍼셉트론(Perceptron) 증명



ABOUT DEEP LEARNING

22

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 다층 퍼셉트론(MLP : Multi Layer Perceptron)

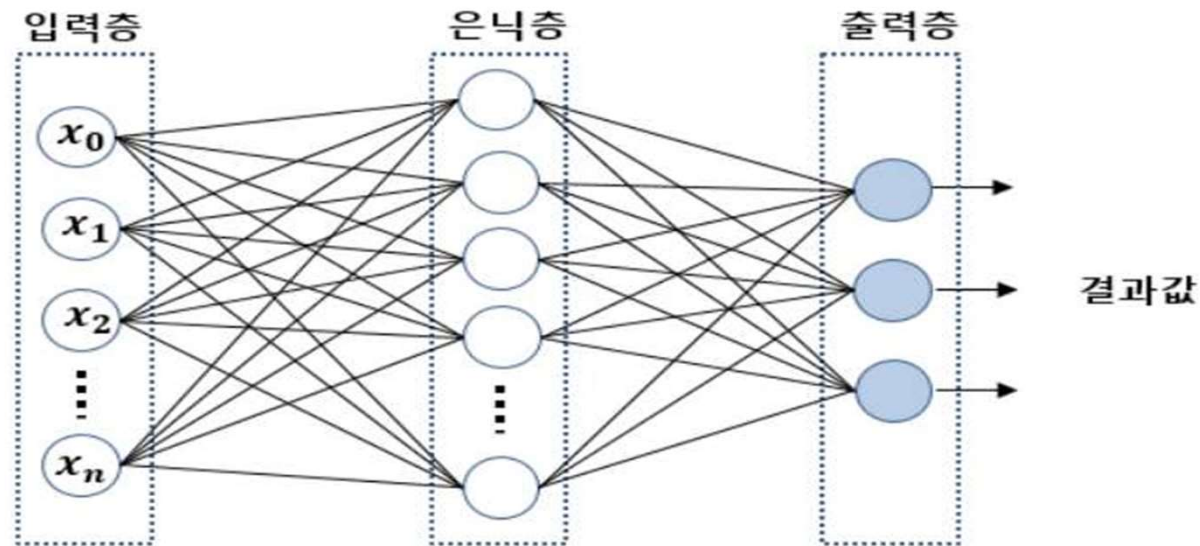
- 퍼셉트론을 여러층 쌓아 올린 구조
- 선형분리 불가능한 문제도 해결 가능
- 인공신경망(ANN)을 의미함
- 경사하강법 손실 함수 사용

ABOUT DEEP LEARNING

23

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 다층 퍼셉트론(MLP : Multi Layer Perceptron)

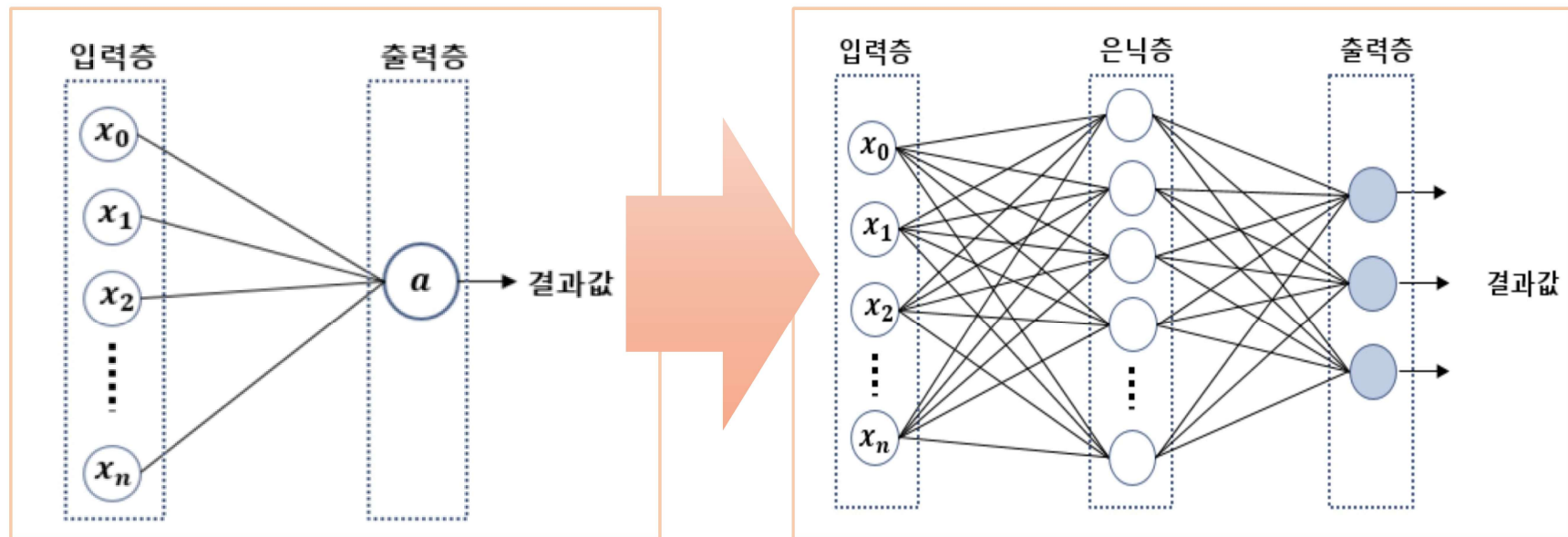


ABOUT DEEP LEARNING

24

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 다층 퍼셉트론(MLP : Multi Layer Perceptron)



ABOUT DEEP LEARNING

25

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 동작원리

[STEP1] 순전파 FORWARD PROPAGATION

- 입력층 =====> 출력층 방향 계산 과정
- 신호와 가중치 곱한 값 출력층까지 차례대로 계산
- 피쳐1*가중치1 + 피쳐2*가중치2 + + 피쳐n*가중치n + b
→ 활성화함수AF(피쳐가중치합+b) → 결과값

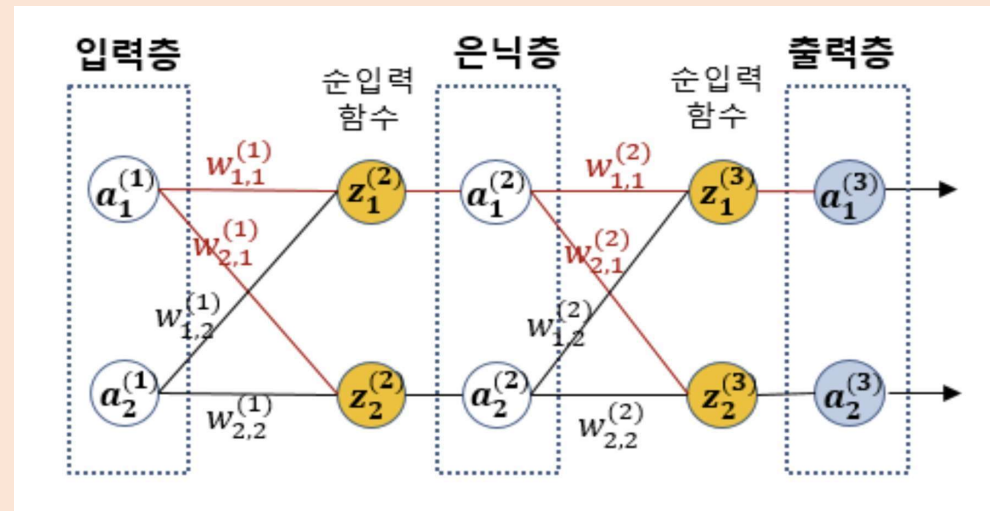
ABOUT DEEP LEARNING

26

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 동작원리

[STEP1] 순전파 FORWARD PROPAGATION



ABOUT DEEP LEARNING

27

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 동작원리

[STEP2] 역전파 BACKWARD PROPAGATION

- 입력층 <==== 출력층 방향 계산 과정
- 오류에 대한 미분값, 학습률, 경사하강법 이용 최적화 값 입력층으로 전달
- 현재 W, b - (정답-예측값) \rightarrow 미분값 * 학습률 \rightarrow 새로운 W, b 업데이트
 \rightarrow 이전 층으로 전달

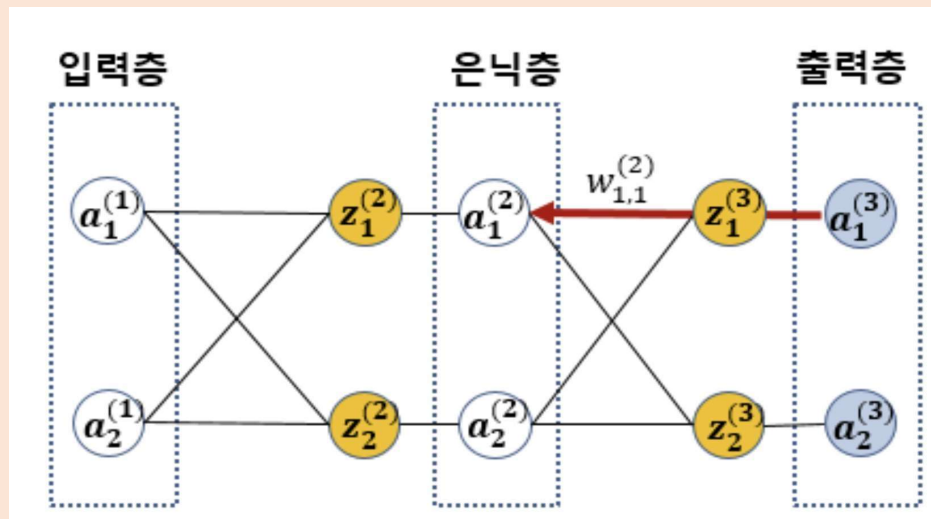
ABOUT DEEP LEARNING

28

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 동작원리

[STEP2] 역전파 BACKWARD PROPAGATION



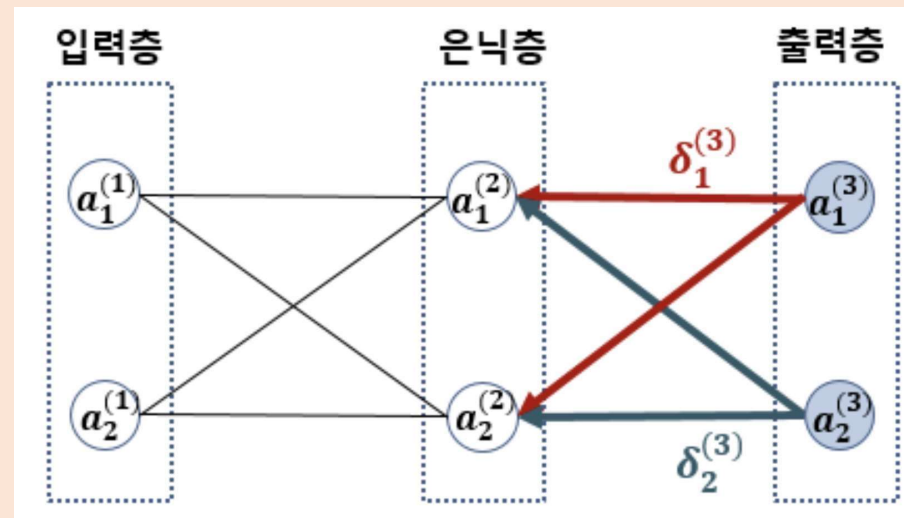
ABOUT DEEP LEARNING

29

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 동작원리

[STEP2] 역전파 BACKWARD PROPAGATION

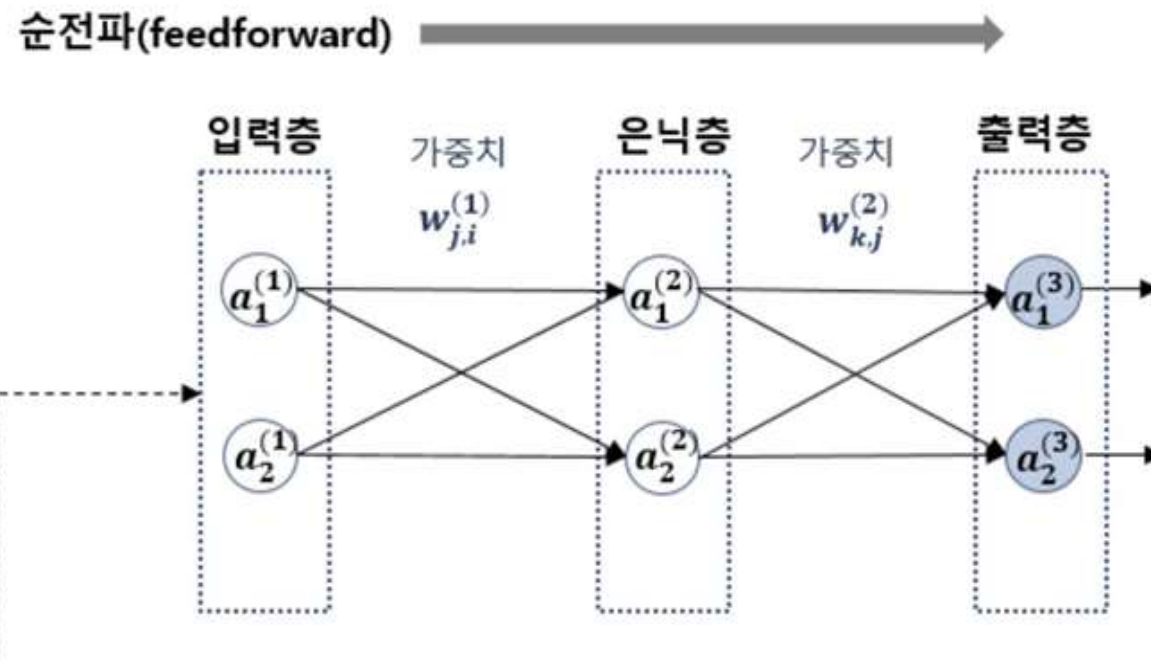


ABOUT DEEP LEARNING

30

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 동작원리

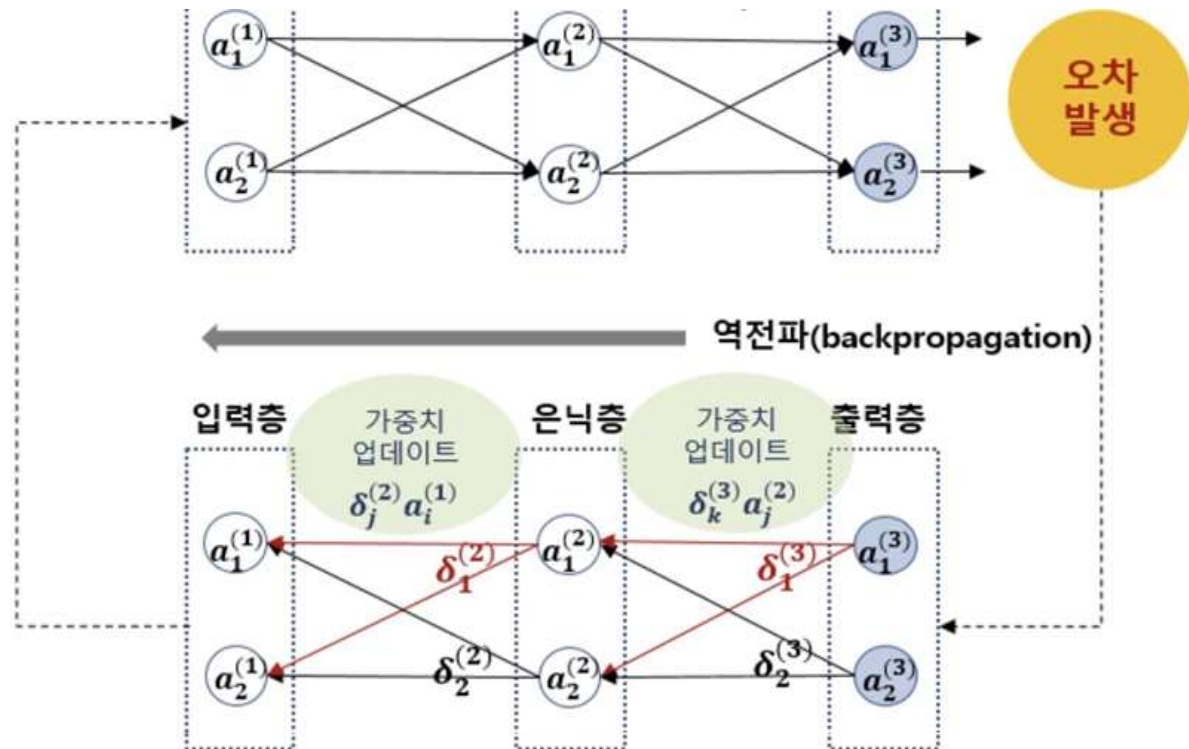


ABOUT DEEP LEARNING

31

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 동작원리

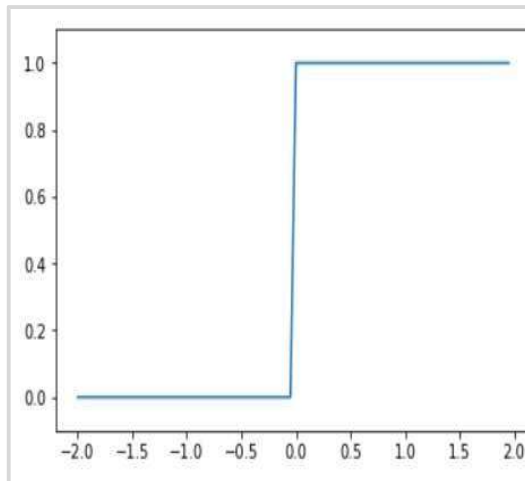


ABOUT DEEP LEARNING

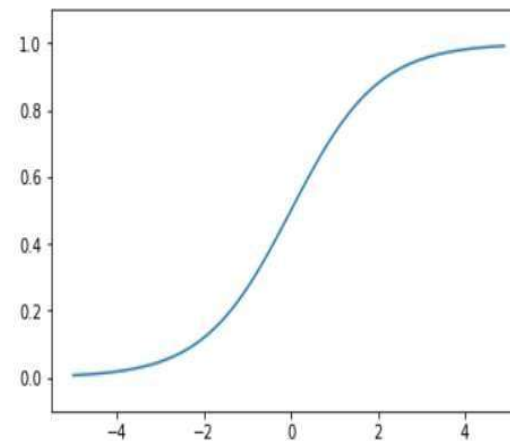
32

◆ 인공신경망 (ANN : Artificial Neural Network)

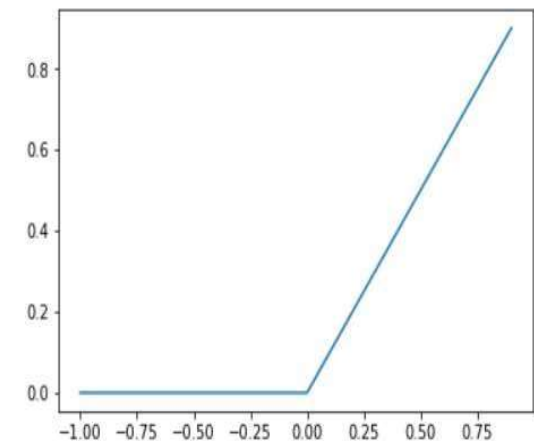
■ 활성화함수 (Activation Function) 가중치합 연산 후 결과값 결정 함수



Step Function



Sigmoid Function



ReLu Function

ABOUT DEEP LEARNING

33

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 오류역전파 알고리즘 (Backward Propagation)

- 1986년 제안된 효율적 최적화 알고리즘
- 경사하강법 이용 파라미터(w , b) 업데이트하며 학습 진행
- 각 Layer에서 손실함수 미분값 계산에 어려움을 해결한 알고리즘
- 미분학의 '체인룰' 착안 → 연쇄법칙
- 출력층에서 입력층로 오류를 전달하며 파라미터(w , b) 업데이트

ABOUT DEEP LEARNING

34

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 최적화 방법 - 경사하강법(Gradient Descent)

- 손실함수의 값을 최소화하기 위해 기울기를 이용하는 사용
- step-size 간격으로 수행, 보통 0.1~0.001 속도가 적당
- 학습 데이터 수, 학습률에 따른 다양한 방법 존재

ABOUT DEEP LEARNING

35

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 최적화 방법 - 경사하강법(Gradient Descent)

Batch Gradient Descent	전체 데이터 학습 후 검증 및 업데이트 진행, 많은 시간 및 계산량 소요
SGD (Stochastic Gradient Descent)	일부 학습 데이터(mini-batch) 선택 후 진행 BGD에 비해 다소 부정확, 속도 빠름
Momentum	업데이트 시 이전 값과 비교하여 같은 방향으로 업데이트 진행 -> 관성
AdaGrad (Adaptive Gradient)	변수들 update 시 각각의 변수마다 lr 즉 step size 다르게 설정 갱신 정도 약해져 전혀 갱신되지 않는 경우 발생
RMSProp / AdaDelta	AdaGrad의 단점 보완, 과거 기울기는 조금 반영 + 최신 기울기 많이 반영
Adam (Adaptive Moment Estimation)	과거 미분값 계속 가중평균 내면서 효율적 업데이트 R MSProp + Momentum 방식 결합

ABOUT DEEP LEARNING

36

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 기울기 소실/폭주 (Gradient Vanishing / Exploding)

[기울기 소실]

- 역전파 과정에서 입력층으로 갈 수록 기울기가 점점 작아지는 현상
- 입력층 가까운 층에서 가중치 업데이트 제대로 되지 않는 문제 발생

[기울기 폭주]

- 역전파 과정에서 입력층으로 갈 수록 기울기가 비정상적으로 커지는 현상
- 순환신경망에서 발생

ABOUT DEEP LEARNING

37

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 기울기 소실/폭주 (Gradient Vanishing / Exploding)

- 해결방법
 - 활성화 함수
 - 가중치 초기화
 - 배치 정규화

ABOUT DEEP LEARNING

38

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 기울기 소실/폭주 (Gradient Vanishing / Exploding)

- 해결방법 → 활성화 함수

입력/은닉층 : ReLu계열 출력층 : Sigmoid/Softmax/Linear 함수	
Step Function	- 퍼셉트론에서 사용 즉, 이진분류에 사용 - 값이 0보다 작으면 0 출력 / 크면 1출력
Sigmoid Function	- 0 ~ 1 사이의 값 출력, 평균 0.5 - 분류에서 확률 표현 위해서 사용 → 출력층
Softmax Function	- 0.0 ~ 1.0 사이 값 출력 - 다중 분류 → 출력층

ABOUT DEEP LEARNING

39

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 기울기 소실/폭주 (Gradient Vanishing / Exploding)

- 해결방법 → 활성화 함수

ReLU Function (Rectified Linear Unit)	<ul style="list-style-type: none">- 입력이 0을 넘으면 그 입력을 그대로 출력- 입력이 0이하면 0을 출력 → 0 : 죽은 ReLU
Leaky ReLU Function	<ul style="list-style-type: none">- 입력값이 음수면 기울기가 0이되는 경우 => 죽은 ReLU- 입력값이 음수일 경우 0이 아니라 0.001과 같은 매우 작은 수 반환
tanh Function	<ul style="list-style-type: none">- -1 ~ 1 범위값 출력

ABOUT DEEP LEARNING

40

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 기울기 소실/폭주 (Gradient Vanishing / Exploding)

- 해결방법 → 가중치 초기화

➤ 세이버/글로트 초기화(Xavier/Glorot Initialization)

- 2010년 세이버 글로트와 요슈아 벤지오 제안
- 여러 층의 기울기 분산 균형 맞춰 특정 층 너무 주목/ 다른 층 뒤쳐지는 것 막음
- 좋은 성능 AF : Sigmoid(), tanh() 같은 S자 형태 계열
- 좋지 않은 AF : ReLu() 계열

ABOUT DEEP LEARNING

41

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 기울기 소실/폭주 (Gradient Vanishing / Exploding)

- 해결방법 → 가중치 초기화

➤ 헤 초기화(He initialization)

- 세이버 초기화와 유사
- 다음 층의 뉴런의 수를 반영하지 않음
- 좋은 성능 AF : ReLU 계열 → ReLU + He 초기화 방법이 좀 더 보편적

ABOUT DEEP LEARNING

42

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 기울기 소실/폭주 (Gradient Vanishing / Exploding)

- 해결방법 → 입력 데이터 정규화(Normalization)

➤ 배치 정규화(Batch normalization)

- 각 층에 입력되는 데이터를 AF 전 평균과 분산으로 정규화 → 학습 효율적
- 작은 미니 배치보다는 크기가 어느정도 되는 미니 배치에서 하는 것이 좋음
- 피쳐 단위로 평균/분산
- RNN 적용 어려움

ABOUT DEEP LEARNING

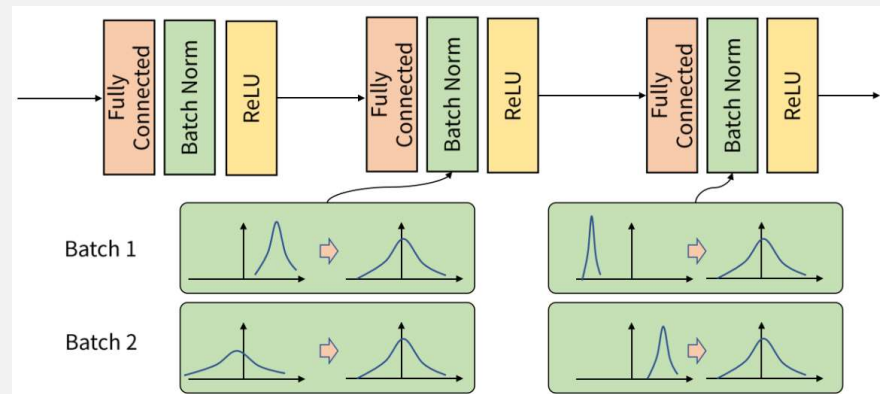
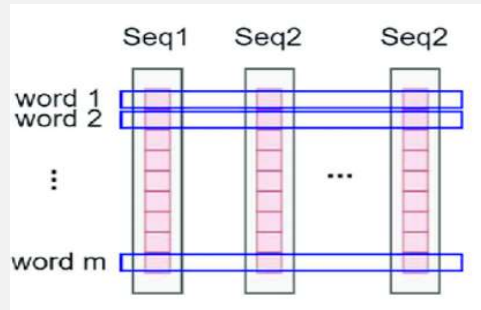
43

◆ 인공신경망 (ANN : Artificial Neural Network)

■ 기울기 소실/폭주 (Gradient Vanishing / Exploding)

- 해결방법 → 입력 데이터 정규화(Normalization)

➢ 배치 정규화(Batch normalization)



ABOUT DEEP LEARNING

44

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 기울기 소실/폭주 (Gradient Vanishing / Exploding)

- 해결방법 → 입력 데이터 정규화(Normalization)

➤ 층 정규화(Layer Normalization)

- data sample 단위로 평균(mean)과 표준편차(std) 계산해서 정규화
- 특성 개수와 상관없이 **batch 내부 데이터 개수의 샘플 평균, 표준편차 값 사용**
- **작은 batch size**에서도 효과적인 이용
- 고정 길이 정규화로 **RNN 사용 가능**

ABOUT DEEP LEARNING

45

◆ 인공신경망 (ANN : Artificial Neural Network)

▪ 기울기 소실/폭주 (Gradient Vanishing / Exploding)

- 해결방법 → 입력 데이터 정규화(Normalization)

➢ 층 정규화(Layer Normalization)

