



[인공지능 입문]

Part 02. 인공지능과 기술

Chapter 09. 인공신경망과 딥러닝

목차

1. 인공지능경망의 역사
2. 딥러닝의 개요
3. 딥러닝의 유형

01

인공신경망의 역사

01. 인공지능경망의 역사

I. 퍼셉트론의 등장

- 1957년 프랭크 로젠블라트가 퍼셉트론(Perceptron)이라는 신경망 모델 발표
- 인간의 두뇌 움직임을 수학적으로 구성하여 당시 굉장한 이슈가 되었던 모델
- 퍼셉트론이 발표된 후 인공지능경망에 대한 기대 폭증



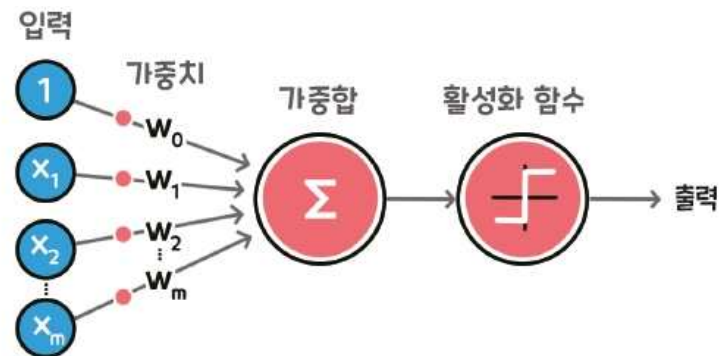
그림 9-1 퍼셉트론 관련 뉴욕타임즈 기사

01. 인공지능의 역사

I. 퍼셉트론의 등장

- 퍼셉트론(Perceptron)

- 신호(x_1, x_2)를 입력으로 받아 하나의 신호(y)를 출력하는 기계
- 신경망의 기원이 되는 알고리즘

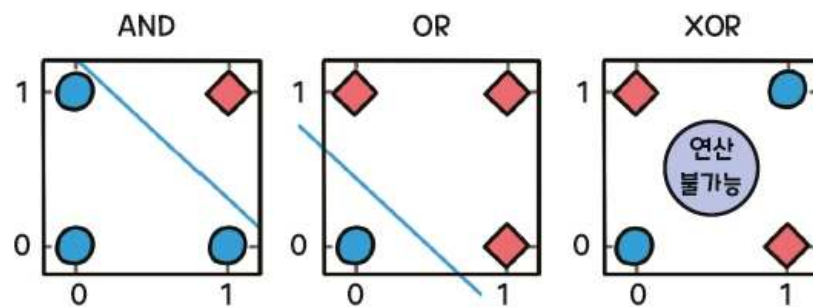


- 입력(Input) : AND 또는 OR 연산을 위한 입력 신호
- 가중치(Weight) : 입력 신호에 부여되는 중요도로, 가중치가 크다는 것은 그 입력이 출력을 결정하는 데 큰 역할을 한다는 의미
- 가중합(Weighted Sum) : 입력값과 가중치의 곱을 모두 합한 값
- 활성화 함수(Activation Function) : 어떠한 신호를 입력받아 이를 적절하게 처리하여 출력해 주는 함수로, 가중합이 임계치(Threshold)를 넘어가면 1, 그렇지 않으면 0을 출력함
- 출력(Output) : 최종 결과(분류)

01. 인공지능망의 역사

I. 퍼셉트론의 등장

- 초기 퍼셉트론을 이용한 문제해결은 AND와 OR 같은 간단한 연산이었음
- 당시 기계가 AND와 OR 연산을 스스로 풀 수 있으면 이를 조합해 어떤 문제든 풀어낼 수 있다고 생각했음
- [그림 9-3]을 보면, AND와 OR 연산의 경우 퍼셉트론을 통한 선형 분리가 가능한 것을 확인 가능



선형 분리(Linearly Separable)는 AND, OR 연산과 같이 직선 하나로 데이터들이 두 개의 그룹으로 나누어지는 것을 의미합니다.

그림 9-3 퍼셉트론의 한계

01. 인공지능경망의 역사

I. 퍼셉트론의 등장

- 1969년 마빈 민스키와 시모어 페퍼트가 현재의 퍼셉트론으로는 XOR 연산이 절대 불가능하다는 가설을 수학적으로 증명
- 퍼셉트론을 여러 개 쌓아 올린 다층 퍼셉트론(MLP)을 통해 XOR 연산에 대한 문제는 해결될 수 있지만, '각각의 가중치(Weight)와 편향(Bias)을 학습시킬 방법이 없다'라는 결론을 내놓으면서 퍼셉트론을 이용한 학습의 한계를 언급
- 이후 인공지능경망 연구는 침체기를 맞이하게 됨

01. 인공지능망의 역사

II. 역전파의 고안

- 역전파(Backpropagation)

- 가중치와 편향을 학습하기 위한 방법으로 고안
- 신경망의 오차(예측값과 실제값의 차이)를 출력층에서부터 입력층으로 거꾸로 전파시켜 각 층(Layer)의 가중치와 편향을 업데이트함

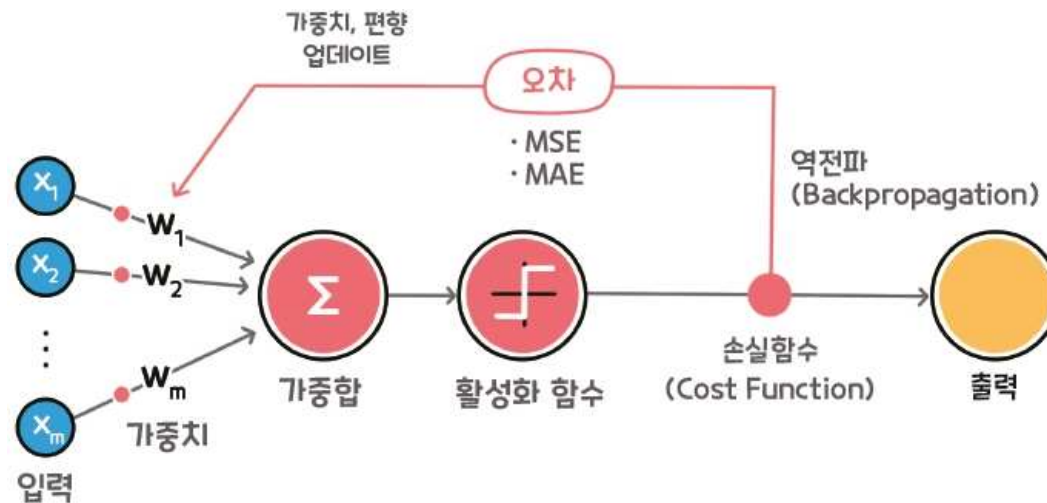


그림 9-4 역전파 과정

01. 인공신경망의 역사

II. 역전파의 고안

- 1974년 폴 워보스가 박사과정 논문에서 역전파를 처음 제안
- 1986년, 제프리 힌튼이 또 다른 역전파 방법을 고안
 - » 제프리 힌튼이 제안한 역전파가 XOR 문제뿐만 아니라 더 복잡한 문제도 해결할 수 있음이 증명되자 인공신경망은 다시 사람들의 관심을 끌기 시작
- 이후 인공신경망 연구는 큰 진전이 있었지만, 다음 이유로 긴 침체기 돌입
 - » 신경망의 깊이가 깊어질수록 원하는 결과를 얻을 수 없음
 - » 신경망 학습에 최적화된 하이퍼파라미터에 대한 이론적인 근거가 없음

01. 인공지능망의 역사

II. 역전파의 고안

- 신경망이 깊어질수록 학습력이 좋아져야 하는데 기대하는 결과가 나오지 않는 일이 지속적으로 발생
- 역전파를 수행할 때 출력층에서 멀리 떨어진 층에서는 기울기가 급속히 작아지는 기울기 소멸 문제(Vanishing Gradient)와 기울기가 너무 커지는 기울기 발산(Exploding Gradient) 문제가 발생해 학습력이 떨어진 것임

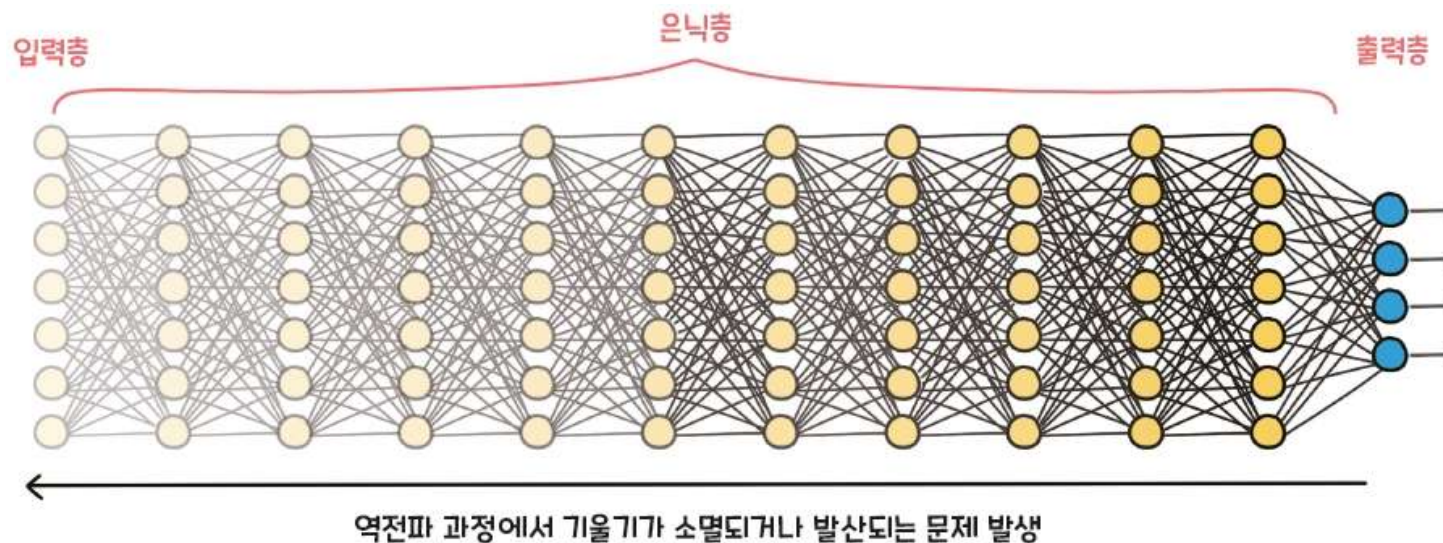


그림 9-5 기울기 소멸, 발산 문제

01. 인공지능경망의 역사

II. 역전파의 고안

하나 더 알기 오차 계산

회귀모델에서 오차를 계산하는 방법으로는 MSE와 MAE가 많이 사용

1. MSE(Mean Squared Error)

- 의미 : 예측값과 실제값의 차이인 오차들의 제곱에 대한 평균
- 수식 : $\frac{\sum(\text{예측값} - \text{실제값})^2}{n}$
- 장점 : 직관적이고 단순함
- 단점 : ① 오류를 제곱하기 때문에 1 미만의 오류는 더 작아지고 그 이상의 오류는 더 커지는 값의 왜곡이 발생
② 스케일에 의존적

01. 인공지능경망의 역사

II. 역전파의 고안

하나 더 알기

오차 계산

2. MAE(Mean Absolute Error)

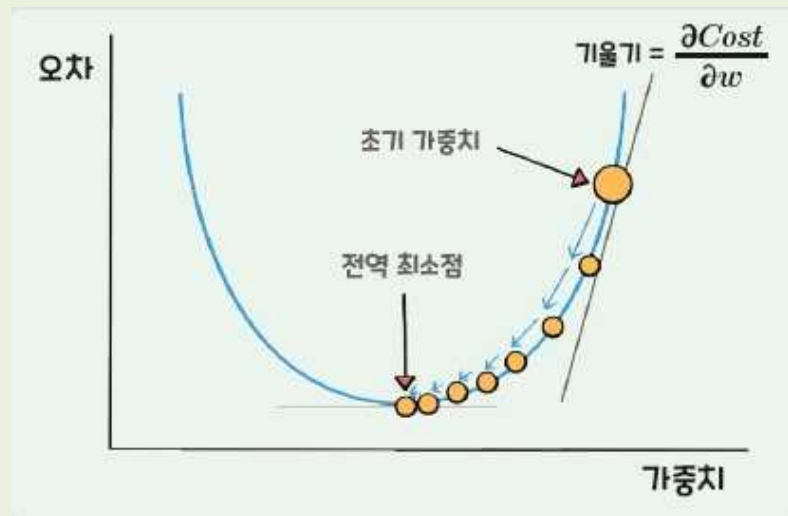
- 의미 : 예측값과 실제값의 차이인 오차들의 절댓값에 대한 평균
- 수식 : $\frac{\sum(|\text{예측값} - \text{실제값}|)}{n}$
- 장점 : 매우 직관적인 지표
- 단점 : 스케일에 의존적

01. 인공지능경망의 역사

II. 역전파의 고안

하나 더 알기 기울기 계산

- 경사 하강법은 오류가 작아지는 방향으로 가중치(w)값을 보정하기 위해 사용
- 경사 하강법은 최초 가중치에서부터 미분을 적용한 뒤 이 미분값이 계속 감소하는 방향으로 순차적으로 가중치값을 업데이트한 후 업데이트가 끝나는 전역 최소점에서의 w 를 반환

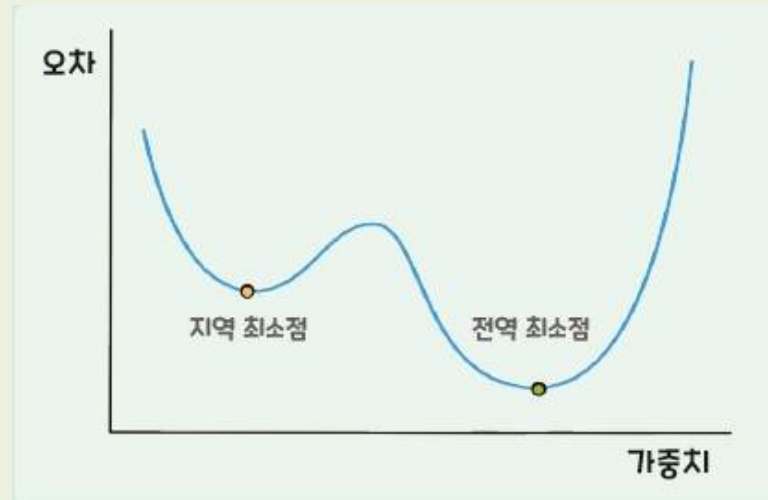


01. 인공지능경망의 역사

II. 역전파의 고안

하나 더 알기 기울기 계산

- 손실함수에는 지역 최소점(Local Minimum)과 전역 최소점(Global Minimum)이 있는데, 지역 최소점은 함수 일부 구간의 최솟값이며, 전역 최소점은 전체 구간에서의 최솟값임



01. 인공지능망의 역사

III. 딥(Deep)의 출현

- 2006년, 역전파를 고안했었던 제프리 힌튼이 가중치의 초기값을 제대로 설정하면 깊이가 깊은 신경망도 학습이 가능하다는 연구를 선보임
- 2007년, 벤지오(Bengio) 팀이 오토인코더(Autoencoder)를 사용한 좀 더 간단한 사전훈련 방법을 제안
- 이때부터 인공지능망(Neural) 대신 '딥(Deep)'이라는 용어를 사용하기 시작

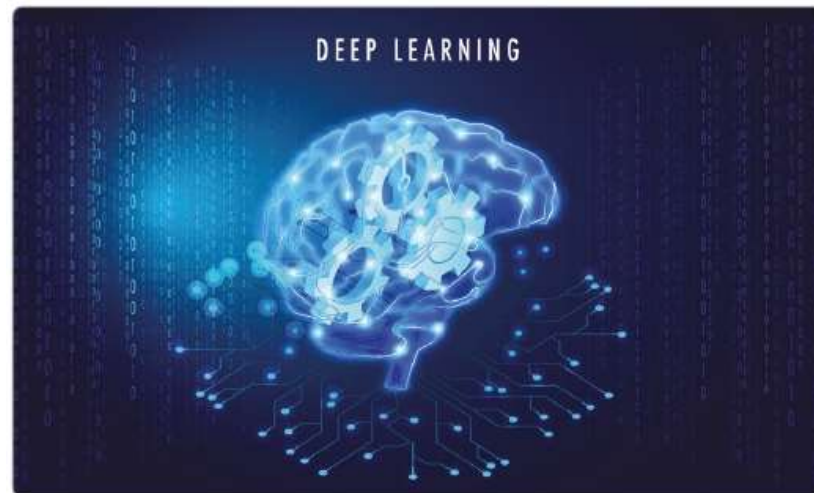


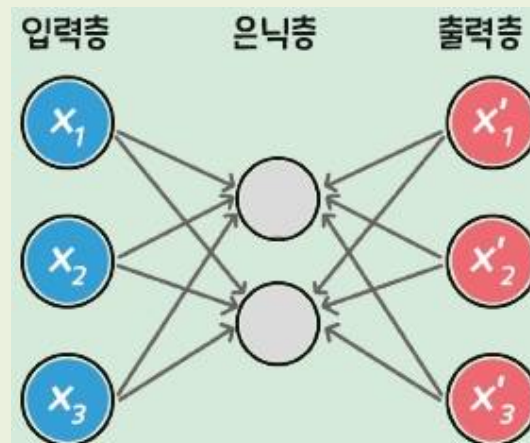
그림 9-6 딥러닝의 출현

01. 인공지능망의 역사

III. 딥(Deep)의 출현

하나 더 알기 오차 계산

- 오토인코더(Autoencoder) : 입력층과 출력층이 동일한 네트워크에 데이터를 입력하여 비지도학습을 하는 것
- 인코더를 통해 입력 데이터에 대한 특징을 추출하고, 디코더를 통해 원본 데이터를 재구성하는 학습
- 가중치의 좋은 초기값을 얻는 목적으로 이용됨



02

딥러닝의 개요

02. 딥러닝의 개요

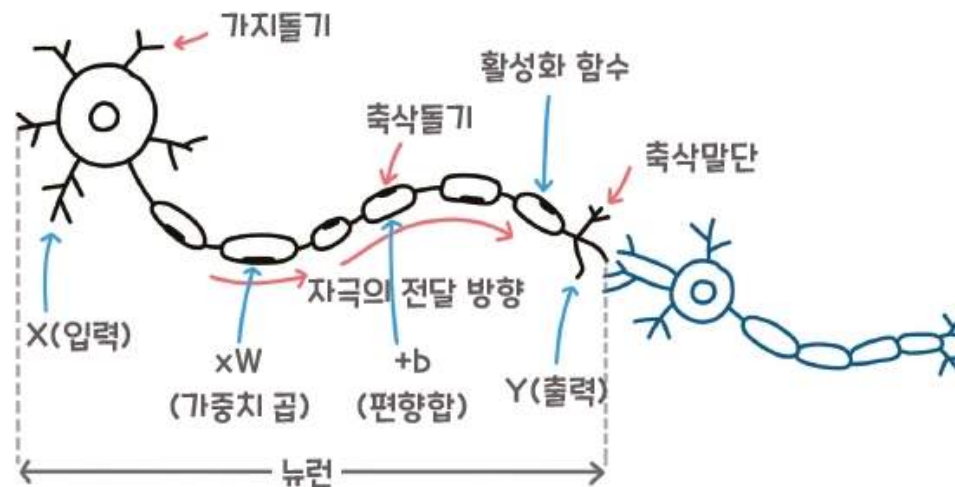
I. 인공신경망의 개념

- 딥러닝의 기원은 인공신경망
- 인간의 뇌에는 수많은 뉴런이 존재하고 그 뉴런들은 시냅스로 서로 연결되어 있는데, 이를 신경망(Neural Network)이라고 부름
- 인공신경망은 사람의 신경망 구조에서 착안해 만들어졌기 때문에 뉴런들의 연결, 즉 신경망을 인공적으로 흉내 낸 것

02. 딥러닝의 개요

I. 인공신경망의 개념

- 인간의 신경망을 이루는 기본 동작



- ① 가지돌기에서 신호를 받아들임
- ② 신호가 축삭돌기를 지나 축삭말단으로 전달됨
- ③ 축삭돌기를 지나는 동안 신호가 약해져서 축삭말단까지 전달되지 않거나 강하게 전달되기도 함
- ④ 축삭말단까지 전달된 신호는 다음 뉴런의 가지돌기로 전달됨
- ⑤ 수억 개의 뉴런 조합을 통해 손가락을 움직이거나 물체를 판별하는 등 다양한 조작과 판단 수행 가능

그림 9-7 인간의 신경망 동작 과정

02. 딥러닝의 개요

I. 인공신경망의 개념

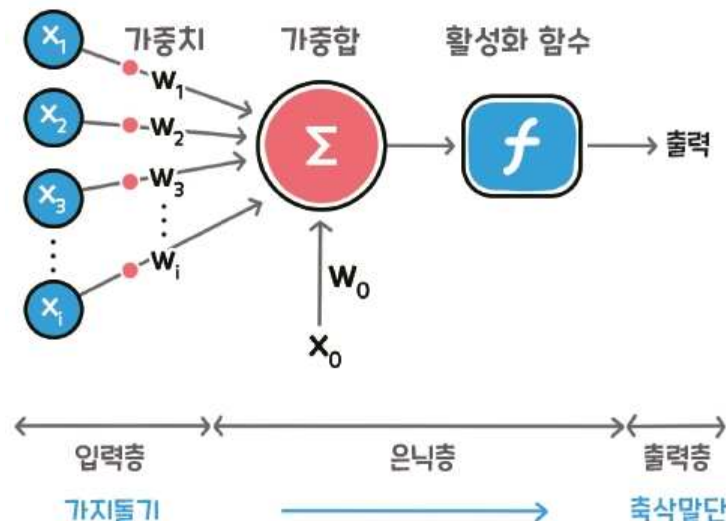
- 인공신경망(ANN, Artificial Neural Network)

- 인간의 신경망을 흉내 낸 머신러닝 기법

- » 가지돌기 : 인공신경망의 입력층

- » 축삭말단 : 인공신경망의 출력층

- » 축삭말단에 이르기까지의 신호의 크기 : 인공신경망에서 가중치



인공신경망	인간의 뇌
Σ	뉴런
\longrightarrow	전기·화학적 신호가 흐르는 방향
가중치	신호의 크기

그림 9-8 인공신경망

02. 딥러닝의 개요

II. 딥러닝의 개념

- 딥러닝(Deep Learning)

- 여러 층(특히 은닉층이 여러 개)을 가진 인공신경망을 사용하여 머신러닝 학습을 수행하는 것
- 심층학습이라고도 부름

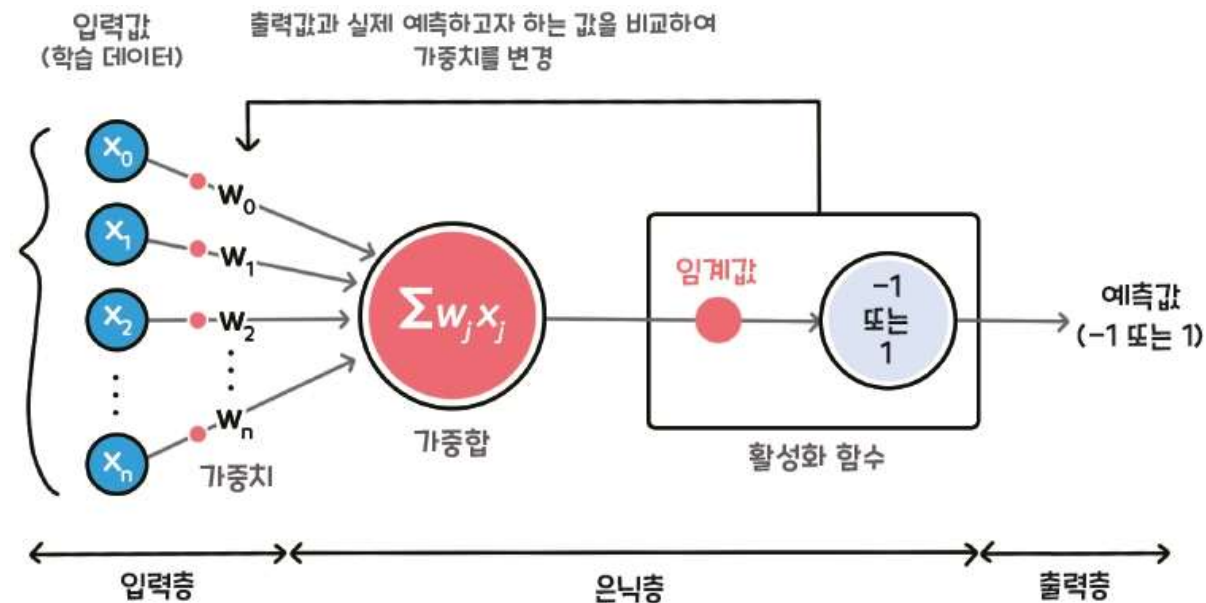


그림 9-9 인공신경망의 구조

02. 딥러닝의 개요

II. 딥러닝의 개념

표 9-1 딥러닝 용어 설명

구분	설명
입력층	학습하고자 하는 데이터를 입력받음
은닉층	모든 입력 노드로부터 입력값을 받아 가중합을 계산하고, 이 값을 활성화 함수에 적용하여 출력층에 전달
출력층	최종 결과 출력
가중치	입력 신호가 출력에 미치는 영향을 조절하는 매개변수로, 입력값의 중요도를 결정
편향	가중합에 더하는 상수로, 하나의 뉴런에서 활성화 함수를 거쳐 최종적으로 출력되는 값을 조절

02. 딥러닝의 개요

II. 딥러닝의 개념

1) 가중합

- 가중합(Weighted Sum)

- 입력값과 가중치를 곱한 뒤 편향(Bias)을 더한 값
- [그림 9-10]을 예로 들어 하나의 노드에 대한 가중합 구하기

» 가중합(y) = $(x_0 \times w_0) + (x_1 \times w_1) + (x_2 \times w_2) + b$

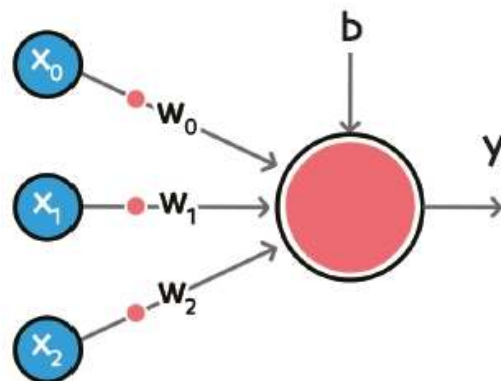


그림 9-10 하나의 노드에 대한 가중합

- 원: 노드
- 화살표: 신호의 흐름
- x_n : 외부에서 들어오는 신호(Input)
- w_n : 신호를 연결하는 가중치(Weight)
- b : 출력값을 조절하는 편향(Bias)
- y : 외부로 나가는 신호(Output)

02. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

- **활성화 함수(Activation Function)**
 - 입력 신호가 출력 결과에 미치는 영향도를 조절하는 매개변수
- **활성화 함수를 사용하는 이유**
 - 출력값을 0~1 사이의 값으로 반환해야 하는 경우에 사용
 - 비선형을 위해 사용

02. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

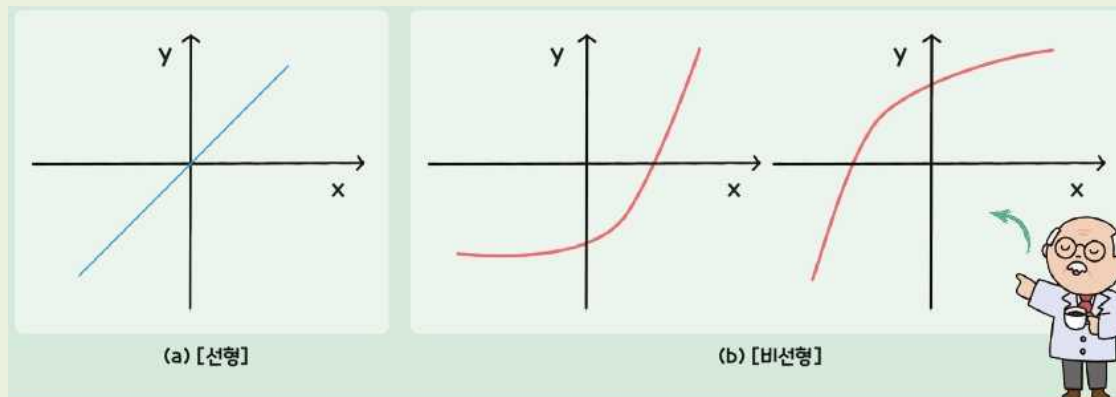
- 선형함수인 $[f(x) = W \times x]$ 식이 있다고 가정
 - 이때 은닉층을 2개 사용한다면, 식은 $[f(f(f(x))) = W \times W \times W \times x = kx]$ 로 단순히 변형됨
 - 은닉층을 20개 사용한다고 해도 동일한 결과가 나옴
 - 즉, 선형함수로는 은닉층을 여러 개 추가해도 은닉층을 1개 추가한 것과 차이가 없기 때문에 정확한 결과를 위해선 비선형 함수, 즉 활성화 함수가 필요함

02. 딥러닝의 개요

II. 딥러닝의 개념

하나 더 알기 선형과 비선형

- **선형(Linear)** : 그림 (a)처럼 그래프가 직선(line)의 모양을 취하는 것으로, 1차 함수의 그래프 모양을 유지하는 것
- **비선형(Non-linear)** : 직선이 아닌 것으로, 그림 (b)와 같이 그래프에서 곡선 모양을 취하는 것을 비선형이라고 함



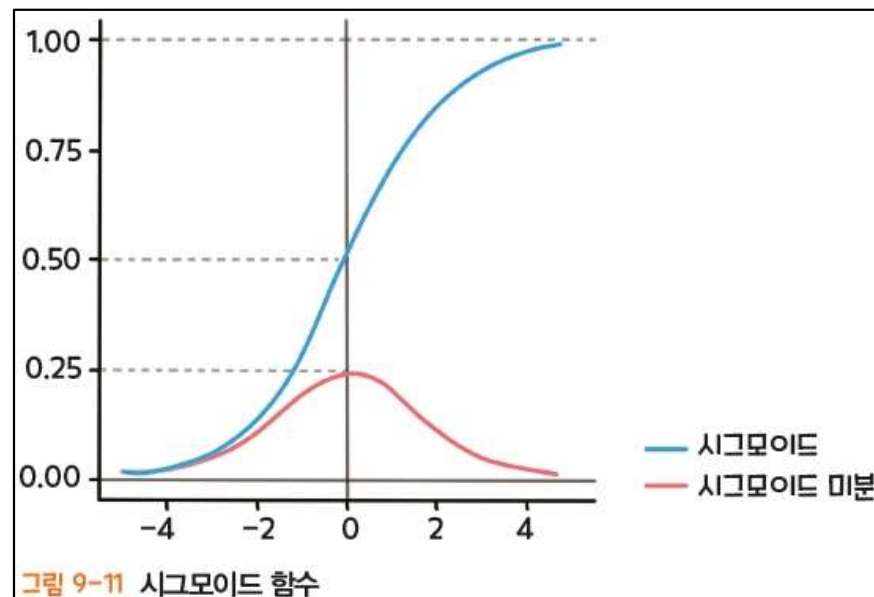
02. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

■ 시그모이드(Sigmoid) 함수

- x 값의 변화에 따라 0에서 1까지의 값을 출력하는 S자형 함수
- 로지스틱 함수라고도 부름



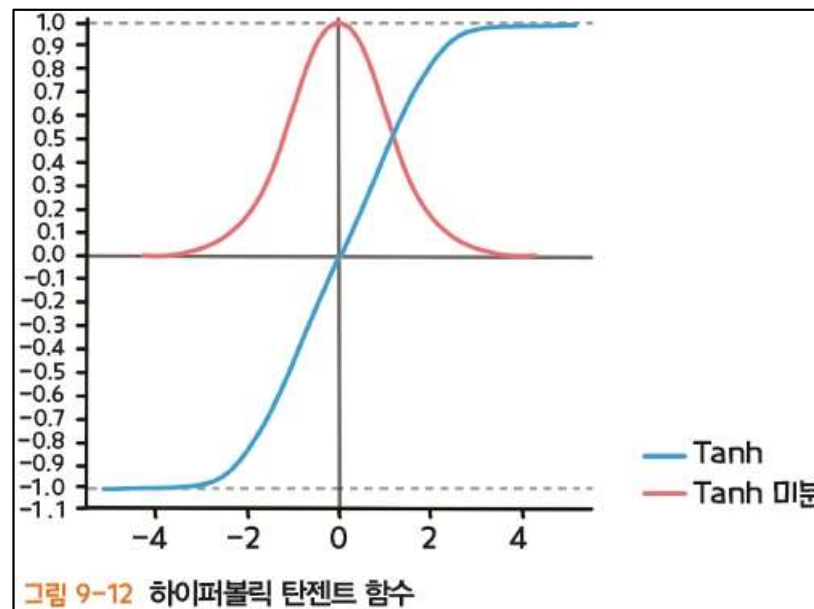
02. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

- 하이퍼볼릭 탄젠트(Hyperbolic Tangent) 함수

- 시그모이드 함수와 유사
- 하지만 $-1 \sim 1$ 의 값을 가지면서 데이터의 평균이 0이라는 점이 다름



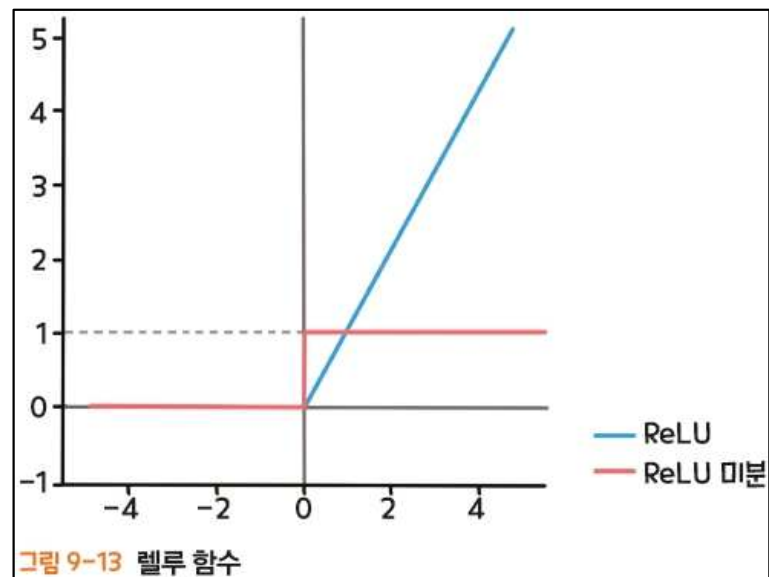
02. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

- 렐루(ReLU) 함수

- x 가 음의 값을 가지면 0을 출력하고, 양의 값을 가지면 x 를 그대로 출력
- 함수 형태도 $\max(0, x)$ 로 계산이 간단하여 학습 속도가 매우 빠름



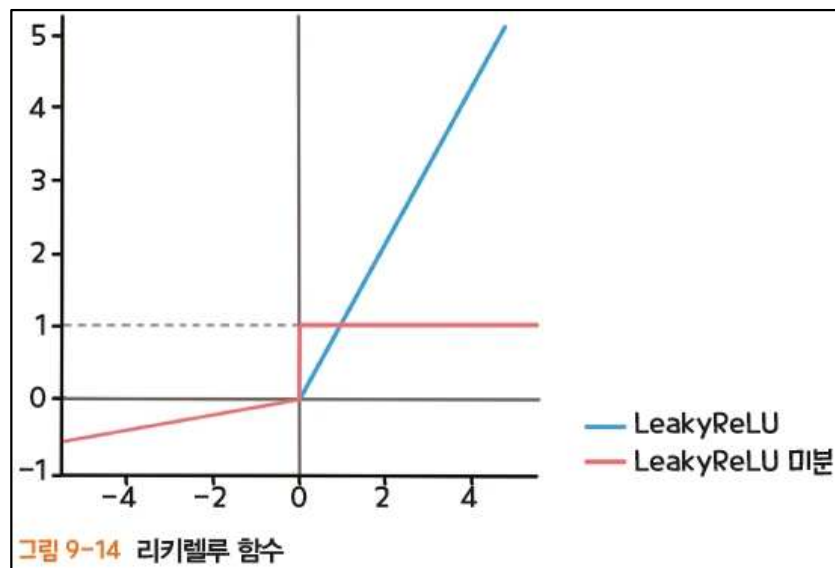
02. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

■ 리키렐루(LeakyReLU) 함수

- 렐루 함수와 유사
- 차이점은 가중치 곱의 합이 0보다 작을 때의 값도 고려한다는 점
- 이 함수는 렐루 함수의 '죽은 렐루' 현상을 보완



02. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

■ 소프트맥스(SoftMax) 함수

- 입력받은 값이 0~1 사이의 값으로 출력되도록 정규화하여 출력의 총 합이 1이 되는 특성을 갖는 함수
- 딥러닝에서 출력 노드의 활성화 함수로 많이 사용



그림 9-15 소프트맥스 함수

02. 딥러닝의 개요

III. 딥러닝 학습

1) 순전파

- 순전파(Forward Propagation)

- 입력층에서 출력층 방향으로 연산이 진행되면서 최종 출력값(예측값)이 도출되는 과정

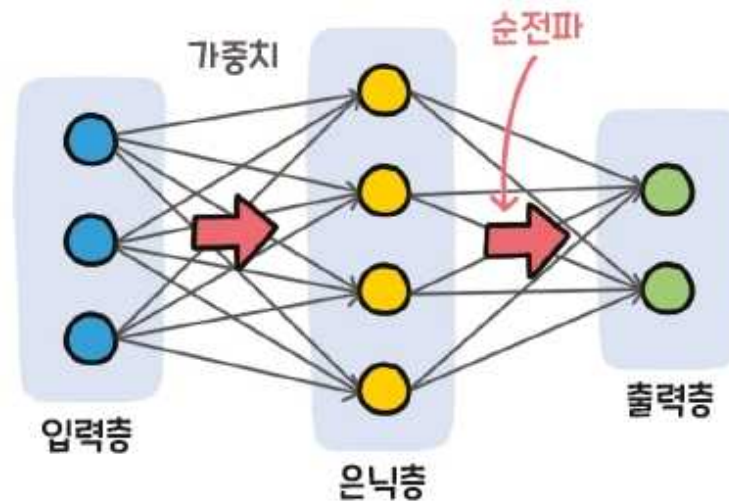


그림 9-16 순전파

02. 딥러닝의 개요

III. 딥러닝 학습

2) 손실함수

- 손실함수(Loss Function)

- 예측값과 실제값의 차이를 구하는 함수

- » 두 값의 차이가 클 수록 손실함수의 값은 커짐

- » 두 값의 차이가 작을수록 손실함수의 값도 작아짐

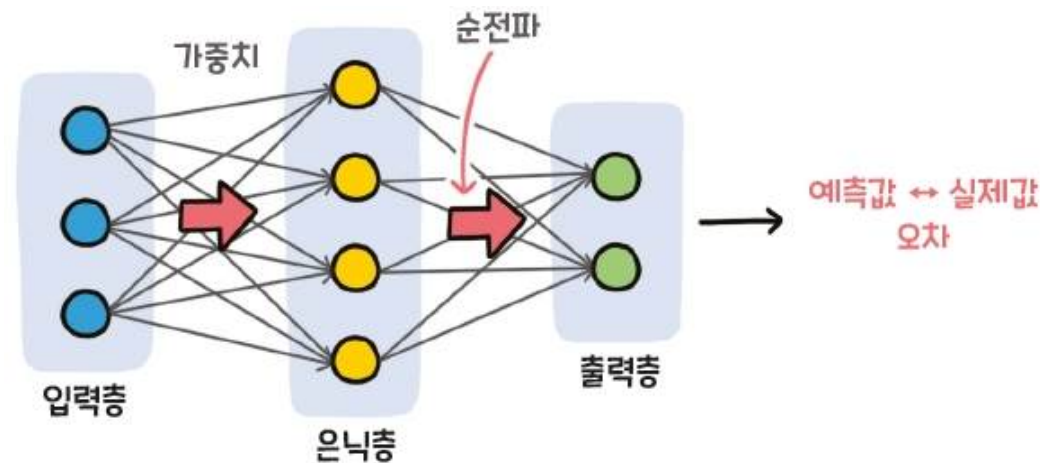


그림 9-17 손실함수

02. 딥러닝의 개요

III. 딥러닝 학습

3) 옵티마이저

- 옵티마이저(Optimizer)

- 딥러닝에서 학습 속도를 빠르고 안정적이게 만드는 것
- 전체 데이터(전체 배치)를 가지고 매개변수(가중치, 편향)의 값을 조정할 수 있고, 정해진 양의 데이터(미니 배치)만 가지고 매개변수의 값을 조정할 수 있음
 - » 배치(Batch) : 가중치 등 매개변수 값을 조정하기 위해 사용하는 데이터의 양

02. 딥러닝의 개요

III. 딥러닝 학습

3) 옵티마이저

- 일반적으로 전체 데이터를 한번에 학습하게 되면 계산량이 많아질 뿐만 아니라 속도도 느려질 수 있기 때문에 미니 배치를 이용한 학습을 함

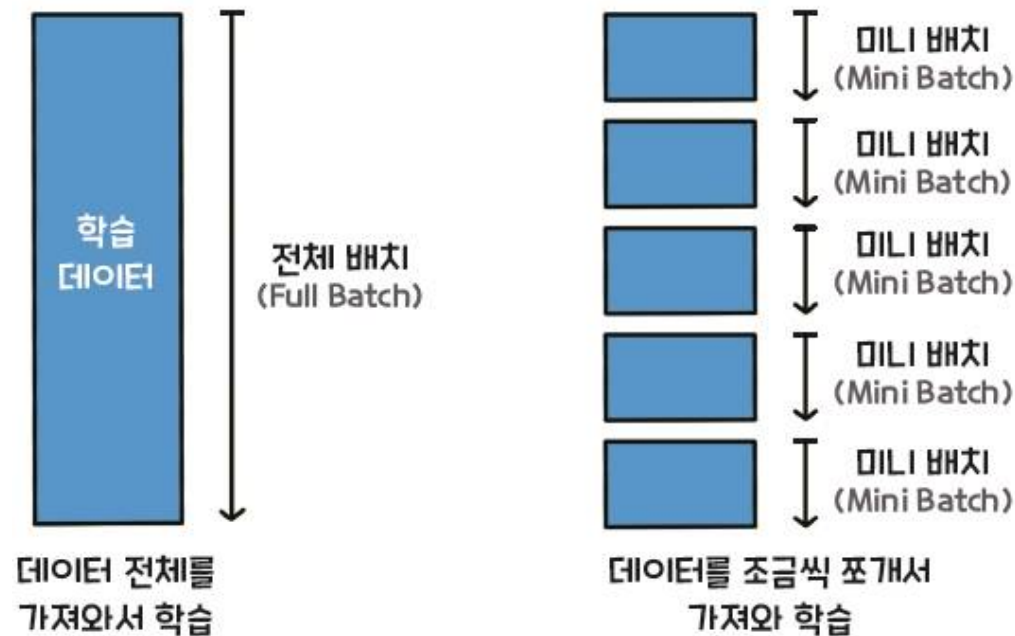


그림 9-18 옵티마이저

02. 딥러닝의 개요

III. 딥러닝 학습

4) 역전파

- 역전파(Backpropagation)

- 오차예측값과 실제값의 차이를 역방향으로 전파시키면서(출력층 → 은닉층 → 입력층) 가중치를 업데이트하는 것

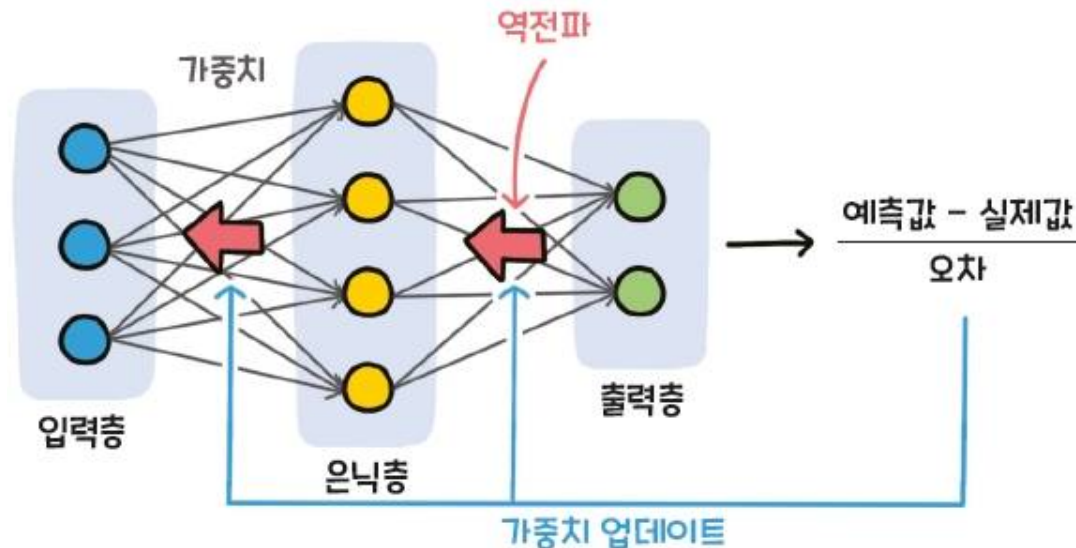


그림 9-19 역전파

02. 딥러닝의 개요

III. 딥러닝 학습

4) 역전파

- 역전파는 경사 하강법을 이용해 오차를 줄이는 방향으로 가중치를 수정함
- 가중치를 수정할 때에는 순전파에서 계산한 결과 $y = f(x)$ 의 편미분값을 오차에 곱해 출력층 \rightarrow 은닉층 \rightarrow 입력층 순서로 전달함



그림 9-20 가중치 수정 방법

03

딥러닝의 유형

03. 딥러닝의 유형

I. 심층 순방향 신경망(DFN)

- 심층 순방향 신경망(Deep Feedforward Network, 이하 DFN)
 - 딥러닝에서 가장 기본으로 사용하는 인공신경망(심층신경망이라고도 부름)
 - DFN은 입력층, 은닉층, 출력층으로 이루어져 있음.
 - » 이때 중요한 것은 은닉층이 2개 이상이어야 한다는 점
 - 순방향이라 부르는 이유 : 데이터가 입력층→은닉층→출력층 순으로 전파되므로

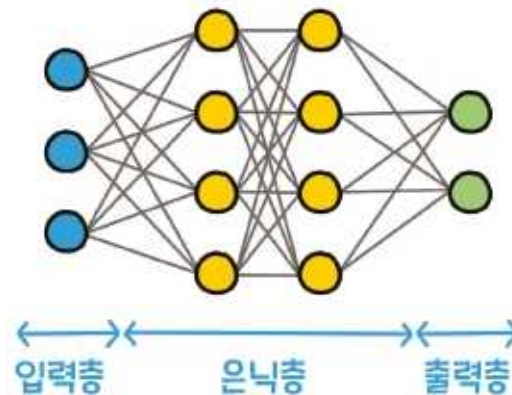


그림 9-21 DFN

03. 딥러닝의 유형

I. 심층 순방향 신경망(DFN)

- DFN의 은닉층이 수십 개에서 수백 개라고 할 때, 입력 데이터가 시간 순서에 따른 종속성을 가질 경우 시계 열데이터 처리에 한계가 있음
- 그래서 이러한 문제점을 해결하기 위해 제안된 것이 순환 신경망

03. 딥러닝의 유형

II. 순환 신경망(RNN)

- 순환 신경망(Recurrent Neural Network, 이하 RNN)

- 시계열 데이터와 같이 시간적으로 연속성이 있는 데이터를 처리하기 위해 고안된 인공신경망

- » 시계열 데이터 : 일정한 시간 동안 관측되고 수집된 데이터



그림 9-22 시계열 데이터의 예 : 주식 데이터

03. 딥러닝의 유형

II. 순환 신경망(RNN)

- 시계열 데이터가 딥러닝 신경망의 입력값으로 사용될 때, 데이터의 특성상 앞에 입력된 데이터가 뒤에 입력된 데이터에 영향을 미침
- 그래서 DFN으로는 시계열 데이터를 정확히 예측할 수 없음

- RNN 구조와 DFN의 차이점

- RNN은 은닉층의 각 뉴런에 순환 구조를 추가하여 이전에 입력된 데이터가 현재 데이터를 예측할 때 다시 사용될 수 있도록 함
- 따라서 현재 데이터를 분석할 때 과거 데이터를 고려한 정확한 데이터 예측

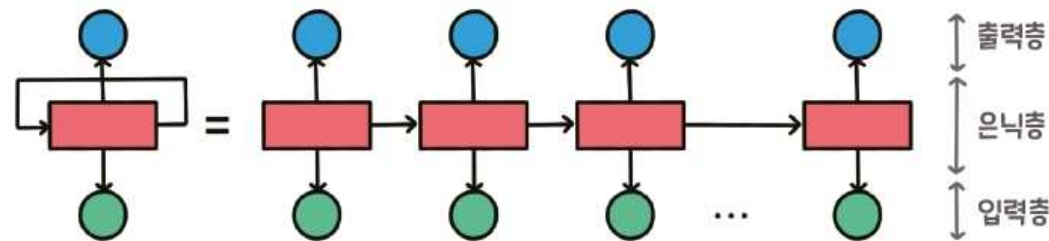


그림 9-23 RNN

03. 딥러닝의 유형

II. 순환 신경망(RNN)

- 하지만 RNN 역시 신경망 층이 깊어질수록(은닉층 수가 많을수록) 먼 과거의 데이터가 현재에 영향을 미치지 못하는 문제가 발생함
- 이를 '장기 의존성(Long-Term Dependency)' 문제라고 하며, 이를 해결하기 위해 제안된 것이 **LSTM**

03. 딥러닝의 유형

III. LSTM

- LSTM(Long Short-Term Memory)

- RNN과는 다르게 신경망 내에 메모리를 두어 먼 과거의 데이터도 저장할 수 있도록 함
- 입출력을 제어하기 위한 소자를 두었는데, 이것을 게이트(Gate)라고 함
 - » 게이트는 입력 게이트, 출력 게이트, 망각 게이트가 있음

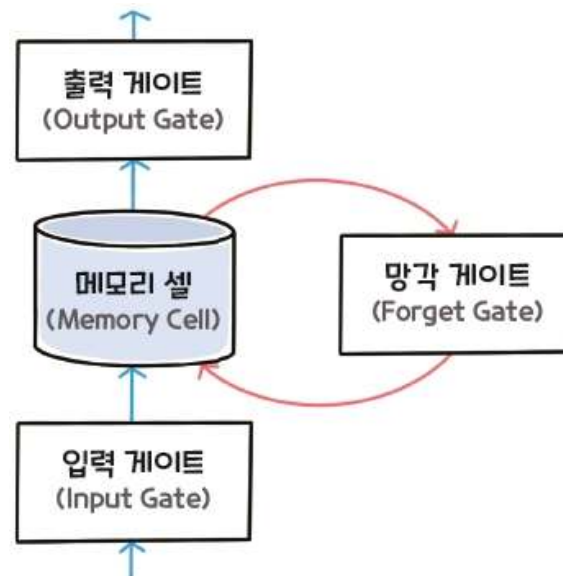


그림 9-24 LSTM

03. 딥러닝의 유형

III. LSTM

- LSTM에서 각 게이트의 역할

- **입력 게이트** : 현재의 정보를 기억하기 위한 소자임. 과거와 현재 데이터가 시그모이드 함수와 하이퍼볼릭 탄젠트 함수를 거치면서 현재 정보에 대한 보존량을 결정
- **망각 게이트** : 과거의 정보를 어느 정도까지 기억할지 결정하는 소자
- **출력 게이트** : 출력층으로 출력할 정보의 양을 결정하는 소자

03. 딥러닝의 유형

IV. 합성곱 신경망(CNN)

- 합성곱 신경망(Convolutional Neural Network, 이하 CNN)
 - 인간의 시각 처리 방식을 모방한 신경망
 - 이미지 처리가 가능하도록 합성곱(Convolution) 연산 도입



그림 9-25 이미지 인식

03. 딥러닝의 유형

IV. 합성곱 신경망(CNN)

- CNN의 구조

- 합성곱층
- 풀링층
- 완전연결층

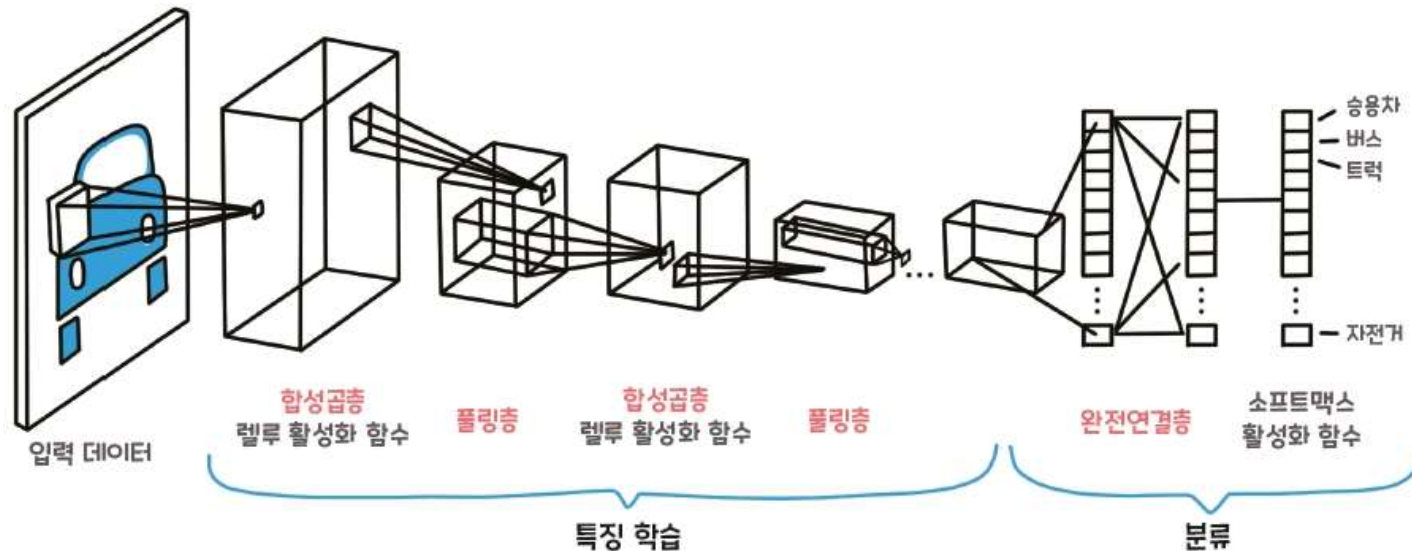


그림 9-26 CNN 구조

03. 딥러닝의 유형

IV. 합성곱 신경망(CNN)

■ 합성곱층(Convolutional Layer)

- 이미지를 분류하는 데 필요한 특징(Feature) 정보들을 추출하는 역할
- 특징 정보는 필터를 이용해 추출
- 합성곱층에 필터가 적용되면 이미지의 특징들이 추출된 '특성 맵'이라는 결과를 얻을 수 있음



그림 9-27 필터가 적용된 합성곱층

03. 딥러닝의 유형

IV. 합성곱 신경망(CNN)

■ 풀링층(Pooling Layer)

- 합성곱층의 출력 데이터(특성 맵)를 입력으로 받아서 출력 데이터인 활성화 맵의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용
- 풀링층을 처리하는 방법 : 최대 풀링, 평균 풀링, 최소 풀링(Min Pooling)
- [그림 9-28]과 같이 정사각행렬의 특정 영역에서 최댓값을 찾거나 평균값을 구하는 방식으로 동작함

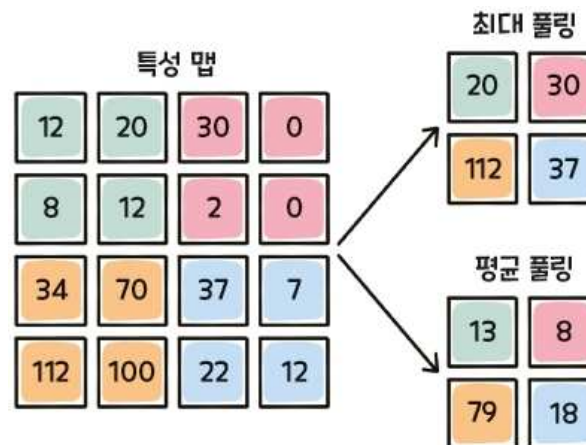


그림 9-28 풀링층 처리 방법

03. 딥러닝의 유형

IV. 합성곱 신경망(CNN)

- 완전연결층(Fully-connected Layer)

- 합성곱층과 풀링층으로 추출한 특징을 분류하는 역할
- CNN은 합성곱층에서 특징만 학습하기 때문에 DFN이나 RNN에 비해 학습해야 하는 가중치의 수가 적어 학습 및 예측이 빠르다는 장점이 있음
- 최근에는 CNN의 강력한 예측 성능과 계산상의 효율성을 바탕으로 이미지뿐만 아니라 시계열 데이터에도 적용해 보는 연구가 활발히 진행되고 있음

03. 딥러닝의 유형

V. 워드 임베딩

- 워드 임베딩(Word Embedding)
 - 단어를 벡터로 표현하는 방법
 - 기계가 사람의 언어를 이해할 수 있도록 변환

03. 딥러닝의 유형

V. 워드 임베딩

1) 원-핫 인코딩(One-hot Encoding)

- N개의 단어를 각각 N차원의 벡터로 표현하는 방식
- 단어가 포함되는 자리엔 1을 넣고 나머지 자리에는 0을 넣는 식

color	color_red	color_blue	color_green
red	1	0	0
green	0	0	1
blue	0	1	0
red	1	0	0

그림 9-29 원-핫 인코딩

03. 딥러닝의 유형

V. 워드 임베딩

2) 워드투벡터(Word2Vec)

- 비슷한 콘텍스트(context)에 등장하는 단어들은 유사한 의미를 지닌다는 이론에 기반하여 단어를 벡터로 표현해 주는 기법
- '주변 단어를 알면 특정 단어를 유추할 수 있다'라는 원리를 기반
- 대표적인 모델로는 CBOW와 Skip-gram이 있음
 - » **CBOW** : 전체 콘텍스트로부터 특정 단어를 예측하는 것
 - » **Skip-gram** : 특정 단어로부터 전체 콘텍스트의 분포(확률)를 예측하는 것

03. 딥러닝의 유형

V. 워드 임베딩

3) TF-IDF

- 단어마다 가중치를 부여하여 단어를 벡터로 변환하는 방법
 - » TF(Term Frequency) : 특정 문서에서 특정 단어가 등장하는 횟수
 - » DF(Document Frequency) : 특정 단어가 등장한 문서의 수
 - » IDF(Inverse Document Frequency) : DF에 반비례하는 수

03. 딥러닝의 유형

V. 워드 임베딩

4) Fasttext

- 페이스북에서 개발한 워드 임베딩 방법
- 단어를 벡터로 변환하기 위해 부분 단어(Sub Words)라는 개념을 도입
 - » 부분 단어를 이해하기 위해서는 N-gram을 먼저 이해해야 함
 - » N-gram : 문자열에서 N개의 연속된 요소를 추출하는 방법
- 부분 단어를 사용하면 워드투벡터에서 문제가 되는 '모르는 단어' 문제를 해결할 수 있기 때문에 임베딩에서 많이 사용되는 모델 중 하나

03. 딥러닝의 유형

VI. 적대적 생성 신경망(GAN)

- 적대적 생성 신경망(Generative Adversarial Network, 이하 GAN)
 - 2개의 신경망 모델이 서로 경쟁하면서 더 나은 결과를 만들어 내는 강화학습
 - 특히 이미지 생성 분야에서 뛰어난 성능을 보이고 있음
 - 기존 인공신경망과는 다르게 두 개의 인공신경망이 서로 경쟁하며 학습 진행
 - 이를 생성 모델(Generator Model)과 판별 모델 (Discriminator Model)이라고 하며, 각각은 서로 다른 목적을 가지고 학습
 - » 생성 모델 : 주어진 데이터와 최대한 유사한 가짜 데이터를 생성
 - » 판별 모델 : 진짜 데이터와 가짜 데이터 중 어떤 것이 진짜 데이터인지를 판별

03. 딥러닝의 유형

VI. 적대적 생성 신경망(GAN)

- GAN의 동작 방식은 [그림 9-30]의 위조지폐범 판별 문제로 이해 가능
 - 위조지폐범은 판별 모델을 속이기 위한 진짜 같은 위조지폐를 만들고, 판별 모델은 위조지폐범이 만든 위조 지폐를 찾아내기 위해 서로 경쟁적으로 학습함
 - 이 과정을 통해 두 모델의 성능은 꾸준히 향상됨



그림 9-30 적대적 생성 신경망

Thank You !