Neural Network

ANN, DNN AND CNN

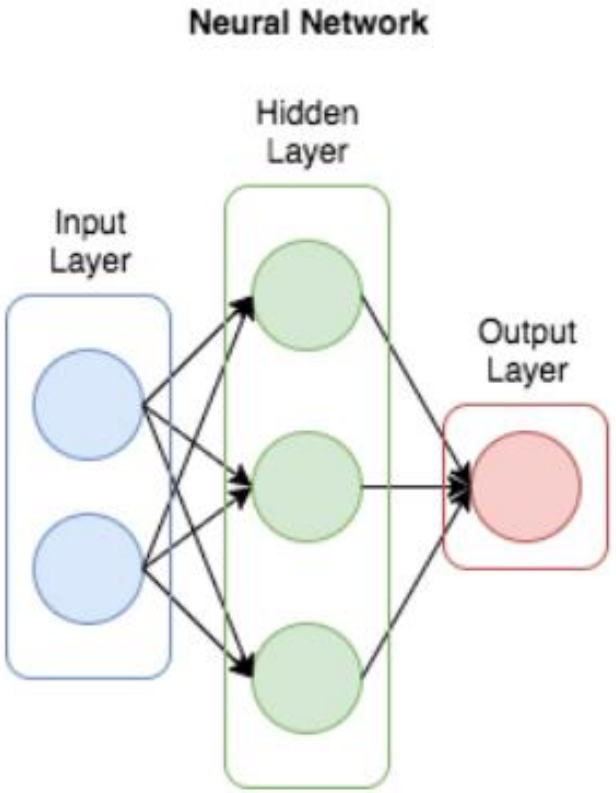
ANN

- ANN (Artificial Neural Network) 이하 NN

: 사람의 신경망 원리와 구조를 모방하여 만든 기계 학습 알고리즘

- 구조

1. 입력층 -> 은닉층 -> 출력층으로 구성
2. 각 층의 노드들은 완전히 연결(이전 층의 모든 노드로부터 입력을 받음), 각 노드는 활성화 함수를 통해 입력 신호를 변환
3. 역전파 알고리즘을 사용해 가중치를 업데이트



ANN

- 활성화함수 : 입력된 값을 변환하여 신경망의 다음 층으로 전달

-> 모델에 비선형성을 부여하여, 신경망이 복잡한 패턴을 학습할 수 있도록 도와줌

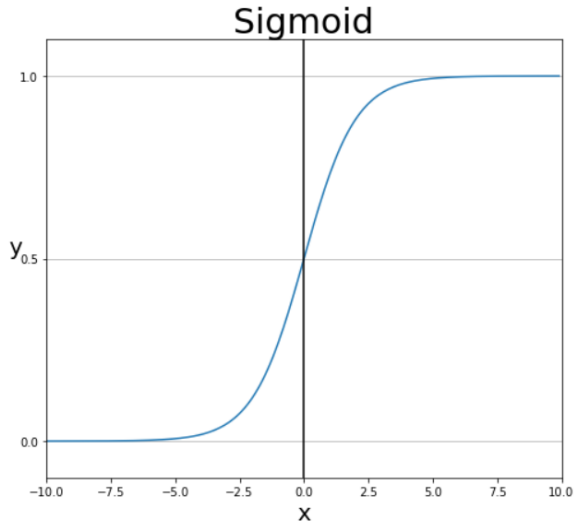
- 종류

1. sigmoid 함수

(1) 수식 : $\frac{1}{1+e^{-x}}$

(2) 특징 : 입력 값을 0과 1 사이의 값으로 변환

-> 출력이 확률값처럼 해석될 수 있어, 이진 분류(단일 노드)에 사용



-> 입력 값이 매우 크거나 작을 경우, 출력의 기울기가 0에 가까워져 학습이 잘 이루어지지 않음

ANN

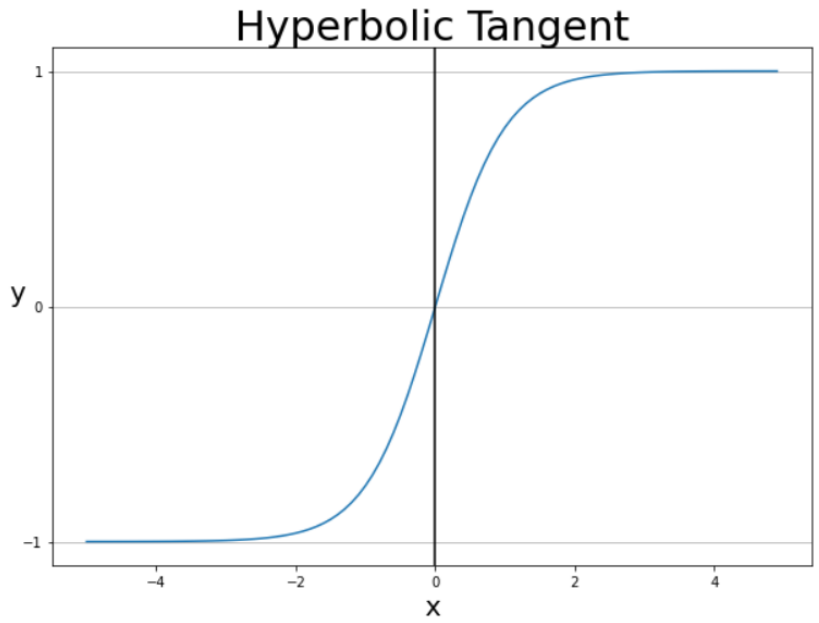
2. 하이퍼볼릭 탄젠트 함수

(1) 수식 : $\frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$

(2) 특징 :

-> 출력이 -1 에서 1 사이에 분포

-> 시그모이드 함수에 비해 기울기의 최대값이 커 학습 속도가 더 빠르지만, 입력 값이 매우 크거나 작을 경우 기울기 소실 문제 발생



ANN

3. ReLU 함수

(1) 수식 : max(0, x)

(2) 특징 :

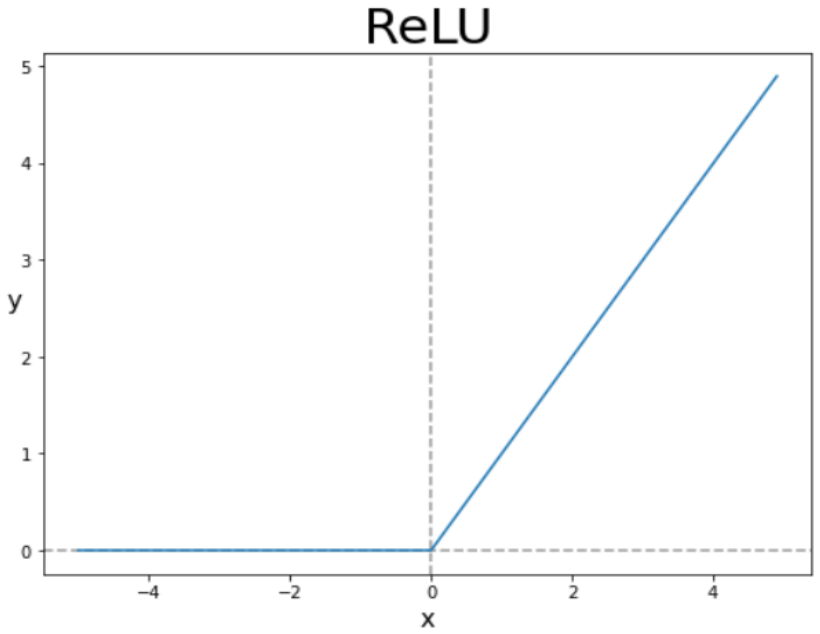
-> 입력이 0보다 크면 그대로 통과, 0보다 작으면 0

-> 간단한 계산으로 인해 매우 빠르고 효과적

-> 기울기 소실 문제가 덜 발생, 음의 출력이 없어

비선형성 유지가 쉬워지므로 복잡한 패턴 학습 가능

-> 대부분의 입력 값이 0 이하인 경우, 해당 노드가 업데이트 되지 않음



ANN

4. Softmax 함수

(1) 수식 : $\frac{e^{x_i}}{\sum_j e^{x_j}}$

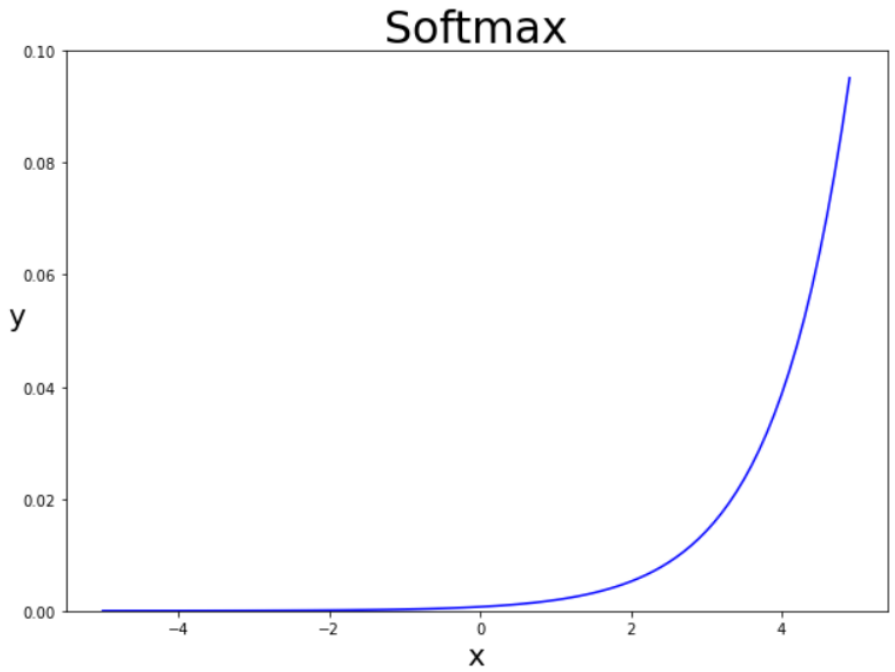
(2) 특징 :

-> 여러 출력값을 0과 1 사이의 확률로 변환해,

전체 출력의 합이 1이 되도록 함

-> 다중 클래스 분류 문제에 사용, 각 출력 노드가

특정 클래스에 속할 확률을 나타냄



-> 계산 비용이 높고, 많은 클래스가 있을 경우 학습 속도가 느려질 수 있음

ANN

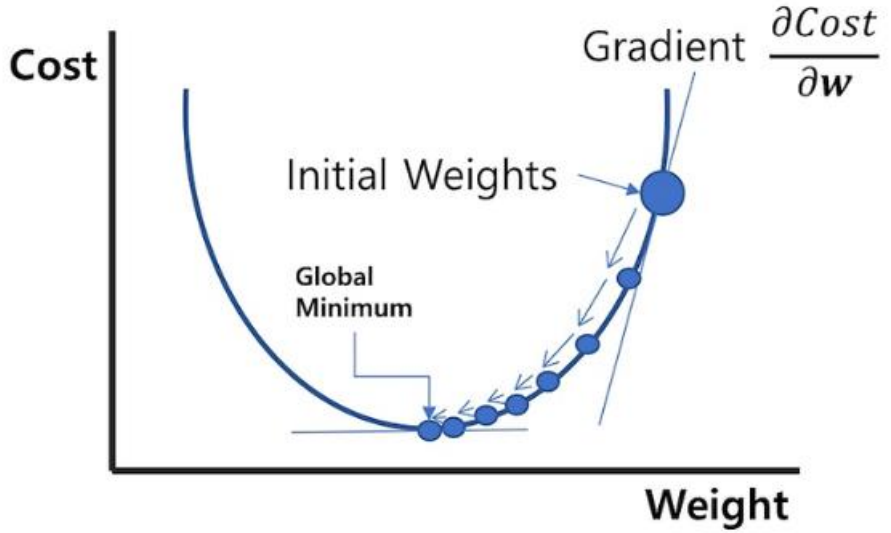
- 역전파 알고리즘 : 네트워크의 가중치 (weights) 를 조정해 예측 오차를 최소화하는 데 사용

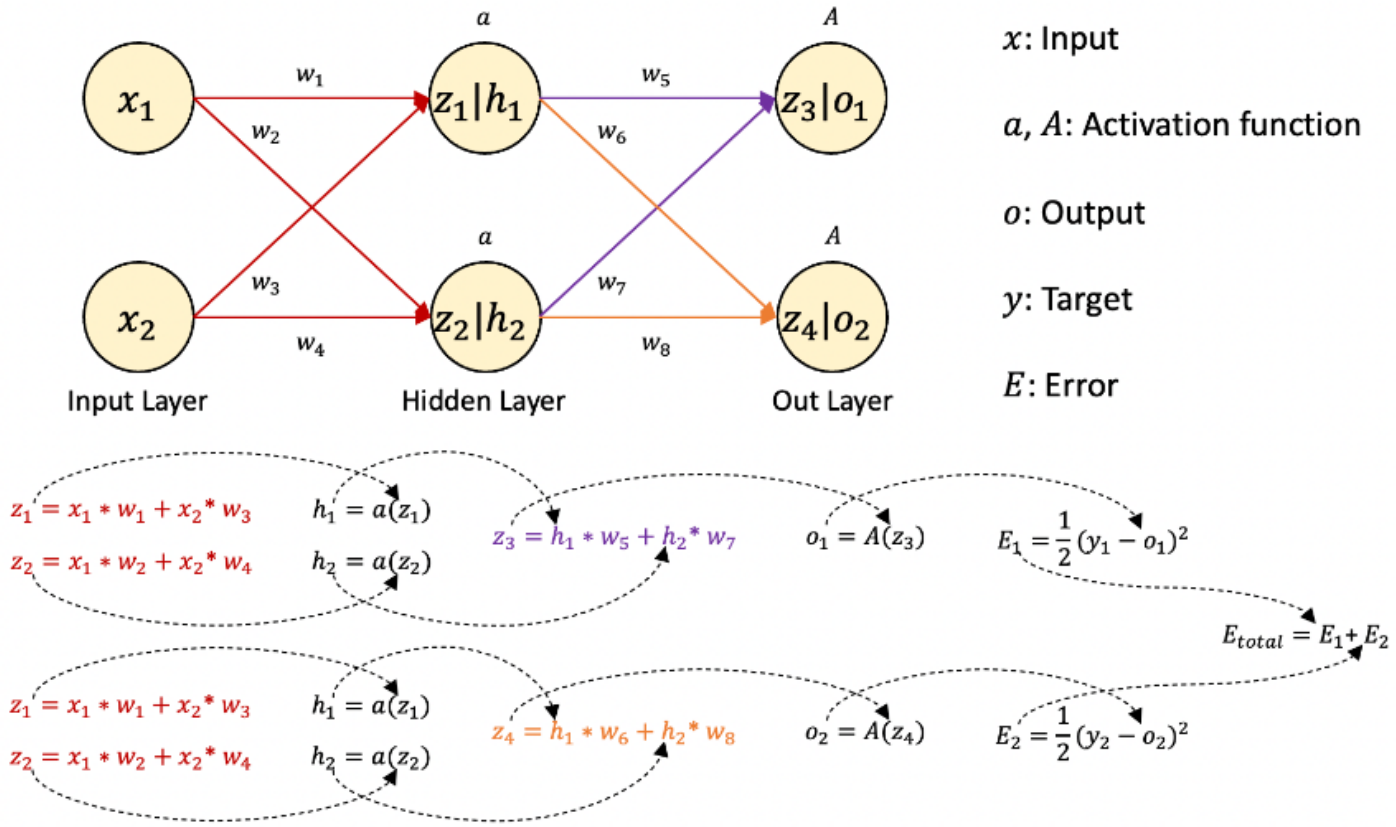
-> 신경망이 입력 데이터를 통해 예측을 수행할 때, 예측 값과 실제 값 사이의 차이를 최소화하는 것이 목표

1. 기본 개념 : 네트워크의 출력에 대한 오차를 각 가중치에 대해 미분하여, 이 미분값을 사용해 가중치를 조정

-> 손실함수를 최소화 하기 위해 경사 하강법 진행

$$w_{new} = w_{old} - \eta \cdot \frac{\partial L}{\partial w}$$





ANN

2. 역전파 진행 과정 : 기본적으로 미분의 연쇄법칙을 활용

(1) Feed-Forward 계산

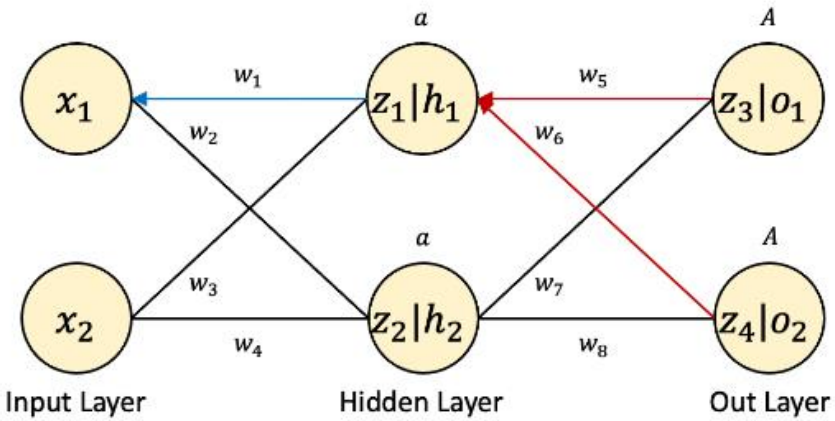
: 인공 신경망의 예측 값 o 생성

-> 실제 값 y 와 비교해 오류 계산

ANN

(2) BackPropagation 수행

-> W 업데이트



x: Input

a, A: Activation function

o: Output

y: Target

E: Error

$$w_1 \leftarrow w_1 - \eta \frac{\partial E_{total}}{\partial w_1}$$

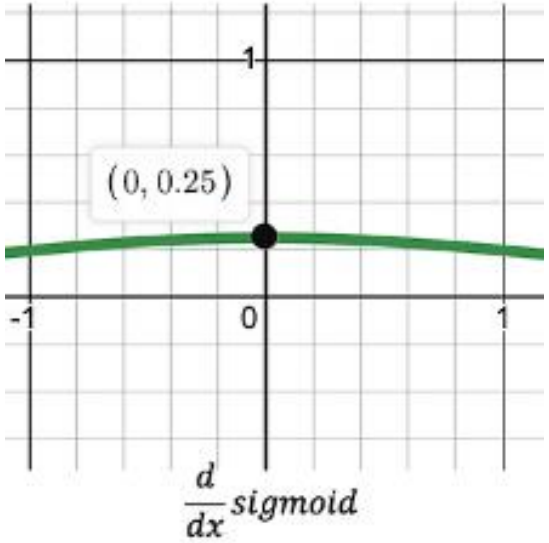
$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial h_1} \times \frac{\partial h_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1} = \left(\frac{\partial E_1}{\partial h_1} + \frac{\partial E_2}{\partial h_1}\right) \times \frac{\partial h_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1} = \left(\frac{\partial E_1}{\partial o_1} \times \frac{\partial o_1}{\partial z_3} \times \frac{\partial z_3}{\partial h_1} \times \frac{\partial h_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}\right) + \left(\frac{\partial E_2}{\partial o_2} \times \frac{\partial o_2}{\partial z_4} \times \frac{\partial z_4}{\partial h_1} \times \frac{\partial h_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}\right)$$

$$\frac{\partial E_{total}}{\partial h_1} = \frac{\partial E_1}{\partial h_1} + \frac{\partial E_2}{\partial h_1}$$

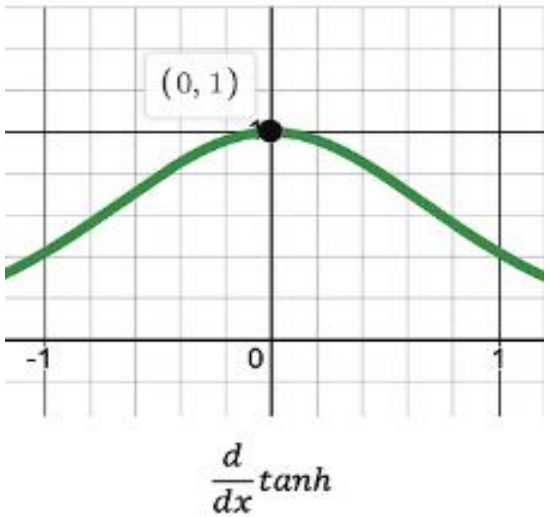
$$\frac{\partial E_1}{\partial h_1} = \frac{\partial E_1}{\partial o_1} \times \frac{\partial o_1}{\partial z_3} \times \frac{\partial z_3}{\partial h_1} \qquad \frac{\partial E_2}{\partial h_1} = \frac{\partial E_2}{\partial o_2} \times \frac{\partial o_2}{\partial z_4} \times \frac{\partial z_4}{\partial h_1}$$

기울기 소실 문제

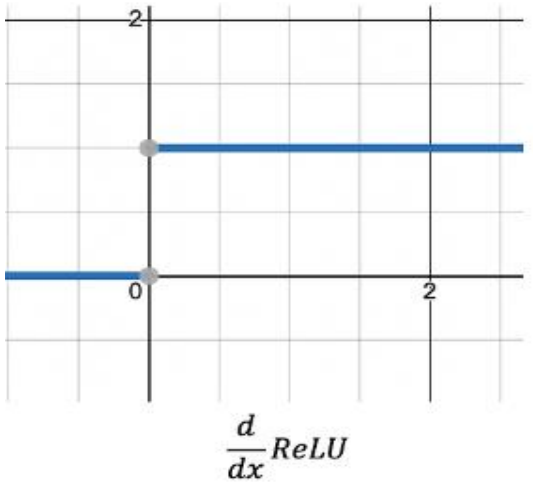
※ 기울기 소실 : 역전파 과정에서 출력층에서 멀어질수록 Gradient 값이 매우 작아지는 현상 -> 노드의 각 가중치의 업데이트 값이 매우 작거나 없어지는 현상



$\frac{d}{dx}sigmoid$



$\frac{d}{dx}tanh$



$\frac{d}{dx}ReLU$

ANN

- ANN 의 장점

1. 단순성 : 구조가 단순해 구현과 이해가 용이함

2. 적용 용이성 : 이미지, 텍스트 등 다양한 형태의 데이터에 적용 가능

3. 비선형 관계 학습

- ANN 의 단점

1. Underfitting (과소적합) : 하나의 은닉층으로는 복잡한 데이터의 패턴을 충분히 학습하기 어려워 훈련 데이터와 테스트 데이터 모두에서 성능이 떨어짐

2. 복잡한 데이터 처리 한계 : 복잡한 패턴 인식에 어려움이 있음

DNN

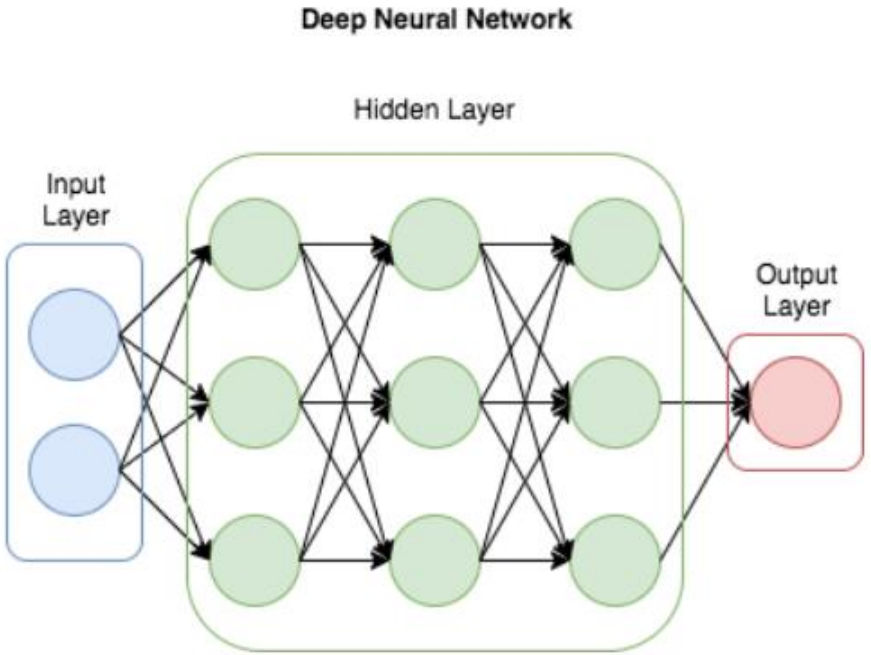
- DNN (Deep Neural Network) : 여러 개의 은닉층을 가진 ANN 의 확장 형태

-> 여러 은닉층을 통해 데이터의 비선형성과 복잡성을 더 잘 학습할 수 있음

- 특징

(1) 복잡한 문제 해결 : ANN 에 비해 더 복잡한 데이터 패턴 학습 가능

(2) 학습 어려움 : 층이 깊어질수록 기울기 소실 문제가 발생할 수 있고, 과적합 (테스트 데이터에만 의존) 의 위험이 커짐



CNN

- CNN (Convolutional Neural Network) : 입력층, 합성곱층, 풀링층, 완전 연결층으로 구성

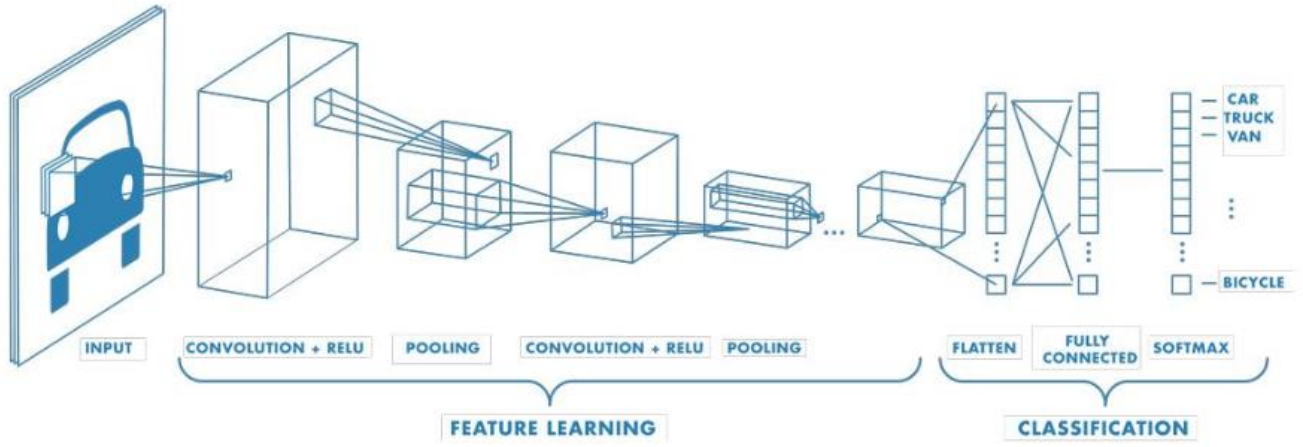
-> 주로 이미지 데이터에 사용되며, 합성곱 연산을 통해 이미지의 공간적 특징 추출

- 특징

(1) 연산 효율 : 필터를 사용해 연산량 줄임, 이미지에서 중요한 특징을 자동으로 추출할 수 있음

(2) overfitting 가능성을 줄임

(3) 고정된 크기의 입력 필요



CNN

- 구조 : 입력값의 특징을 추출하는 부분과, 클래스를 분류하는 부분으로 나눌 수 있음

-> 특징 추출 영역은 Convolutional layer + Pooling Layer (선택사항)

-> 분류 영역은 Fully Connected Layer

-> 특징과 분류 영역 사이에 Flatten Layer

1. Convolutional layer (= Locally Connected Layer)

: 입력값에서 특징을 추출하는 역할

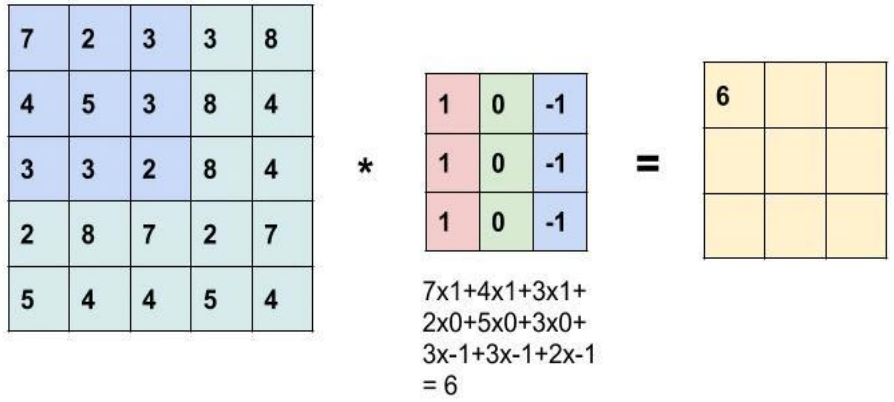
-> 필터 (kernel) 라 불리는 작은 행렬이 입력값에 적용 -> 합성곱 연산 수행

-> 필터가 입력값 전체를 훑으면서 특징 맵 생성 -> 필터와 입력값의 내적 결과에 해당

CNN

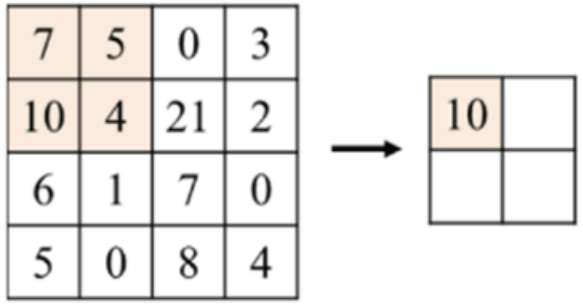
Ex) 입력값 : 5 by 5 행렬, 필터 : 3 by 3 행렬

, stride : 1 (필터의 이동량)



2. Pooling Layer : 특징 맵의 크기를 줄이고, 특정 feature 를 강조하는 역할

-> Max Pooling, Average Pooling 등이 있음



CNN

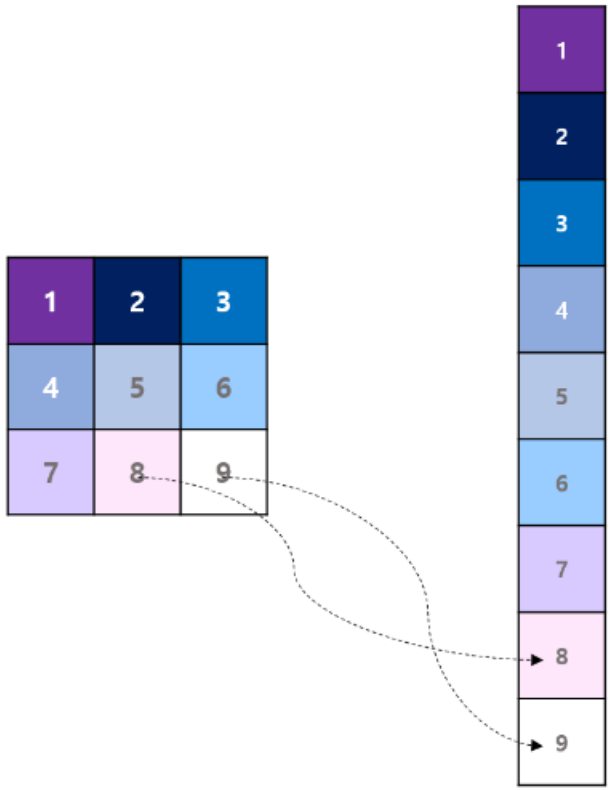
3. Fully Connected Layer (FC Layer) :특징 맵을 벡터로 펼친 후 (Flattening), 이를 완전 연결층의 입력값으로 사용

-> 모든 노드가 연결돼 있으며, 최종적으로 각 클래스에

속할 확률을 예측

-> 출력층에서는 보통 softmax 함수를 사용해 각 클래스에

속할 확률을 계산



CNN

- CNN 을 강화학습에 사용할 때의 이점

1. 공간적 특성 추출

: CNN 은 공간적 정보를 잘 캡처해 다양한 입력에 대한 일반화 성능이 뛰어남

-> 강화학습에서 다루는 환경의 상태가 다양할 때 매우 유용

2. 효율적 학습

: 합성곱 계층은 필터를 사용해 입력 데이터의 모든 부분에서 동일한 가중치를 공유

-> 모델의 파라미터 수를 줄이고, 궁극적으로 학습을 더 효율적으로 만듦

Thank you

2024.08.14