

딥러닝 & DNN

16기 분석 남유지

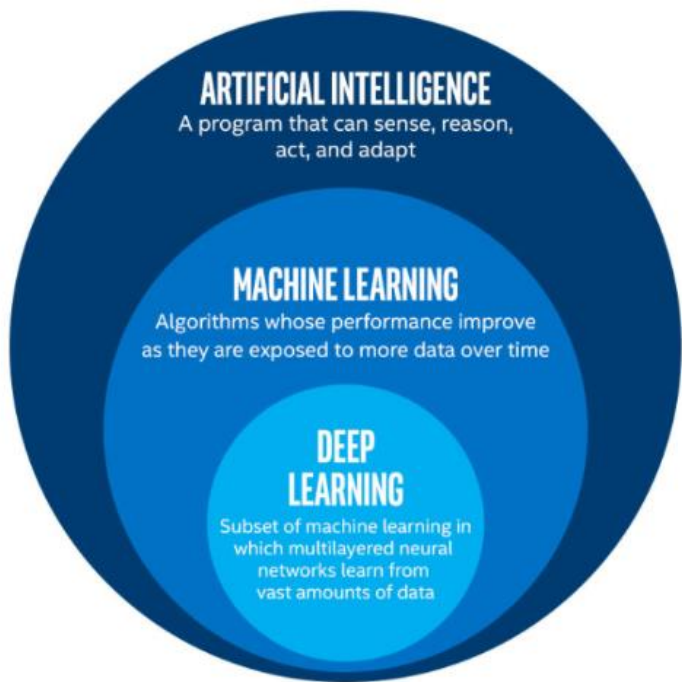
목차

1. 딥러닝

2. ANN

3. DNN

01 딥러닝 - 인공지능, 머신러닝, 딥러닝



인공지능

인간의 지능을 기계 등에 인공적으로 구현한 것

머신러닝

컴퓨터가 명시적으로 프로그램 되지 않고도 학습할 수 있도록 하는 연구 분야

딥러닝

머신러닝의 한 분야로 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화를 시도하는 머신러닝 알고리즘

02 ANN - 인공 신경망

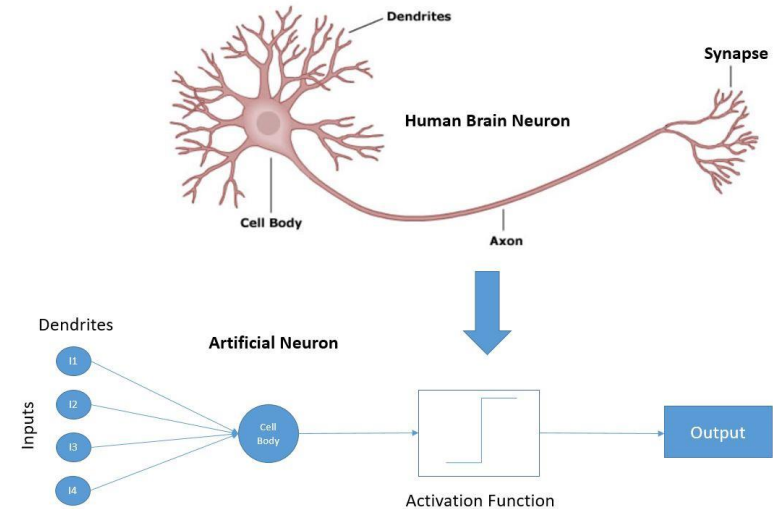
ANN (Artificial Neural Network, Neural Network)

사람의 신경망 원리와 구조를 모방하여 만든 기계학습 알고리즘

- 자극, 신호 = Input data
- 임계값 = 가중치 (weight)
- 전달하는 신호 = Output data
- 비선형 관계 모델링

다양한 ANN의 공통 요소

- Neuron (Node)
- Synapse
- Weight
- Bias
- Activation function

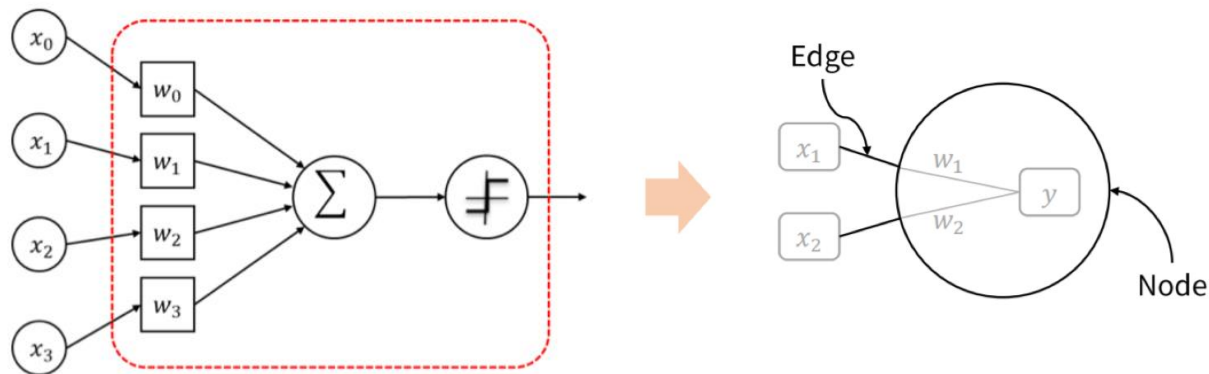


02 ANN - 뉴런

뉴런 (Neuron)

node와 edge로 표현

- node: 단일 뉴런 연산
- edge: 뉴런의 연결성



$$y = a\left(\sum_{i=0}^{N-1} w_i x_i + b\right)$$

활성함수 두 벡터의 내적 편향(Bias)

$$\sum_{i=0}^{N-1} w_i x_i = \mathbf{w} \cdot \mathbf{x} = \mathbf{w}^T \mathbf{x}$$

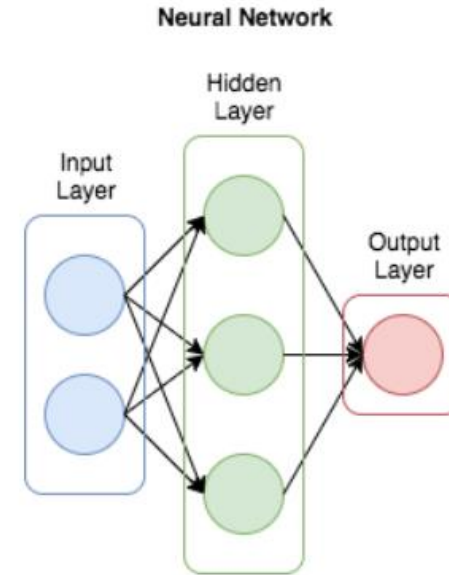
$$y = a(\mathbf{w}^T \mathbf{x} + b)$$

02 ANN - 인공 신경망, 문제점

ANN (Artificial Neural Network, Neural Network)

- 입력층(Input layer)
 - 출력층(Output layer)
 - 은닉층(Hidden layer)
- *layer = 뉴런이 모인 한 단위

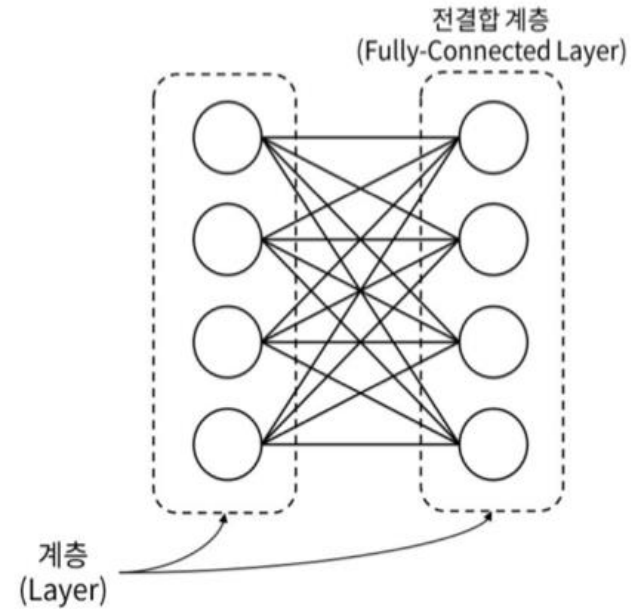
→ 모델을 구성해 학습을 통해서 최적의 weight와 bias를 찾는 것이 목표



02 ANN – Fully connected layer

fully connected layer

이전 계층과 모든 뉴런이 서로 연결된 계층



$$\begin{aligned} W &= [\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{M-1}]^T \\ \mathbf{b} &= [b_0, b_1, \dots, b_{M-1}]^T \end{aligned}$$
$$\left. \begin{aligned} y_0 &= a(\mathbf{w}_0^T \mathbf{x} + b_0) \\ y_1 &= a(\mathbf{w}_1^T \mathbf{x} + b_1) \\ &\vdots \\ y_{M-1} &= a(\mathbf{w}_{M-1}^T \mathbf{x} + b_{M-1}) \end{aligned} \right\} \mathbf{y} = a(W\mathbf{x} + \mathbf{b})$$

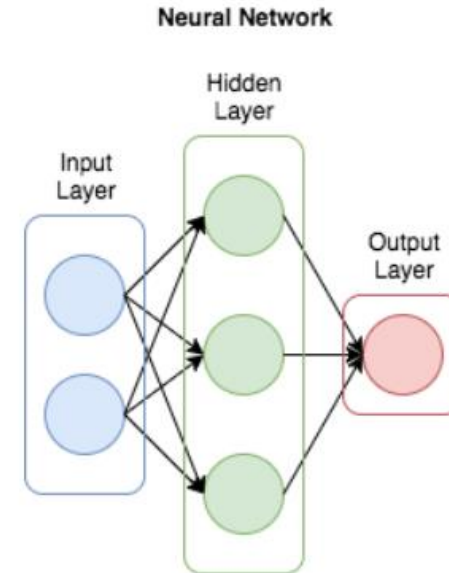
02 ANN - 문제점

NN의 문제점

- **Vanishing Gradient**

층이 깊어질 수록 역전파되는 gradient가 점점 작아져 0에 수렴하여 학습이 잘 되지 않는 문제

- **Overfitting**

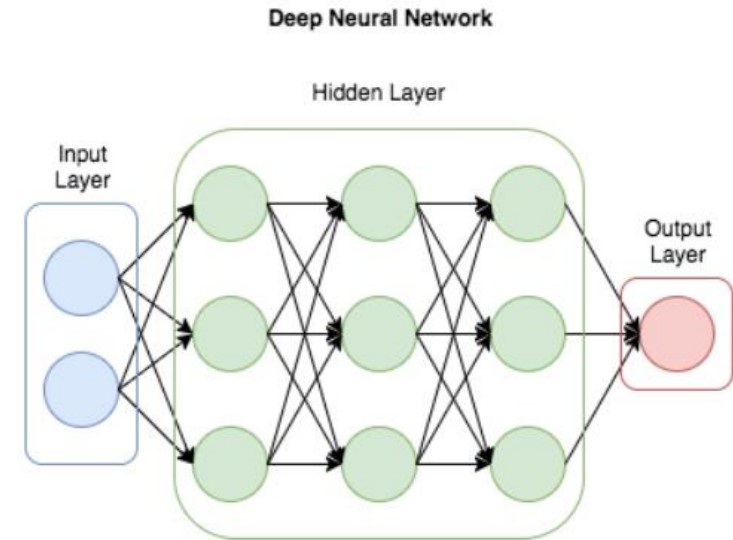


03 DNN – DNN

DNN (Deep Neural Network)

입력층과 출력층 사이에 여러 개의 은닉층들로 이루어진 인공 신경망

- 비선형 관계 모델링
- 최적의 weight을 찾는 것이 목표



03 DNN - DNN 기본 구조

DNN 기본 구조

X : training data , W: weight

순방향

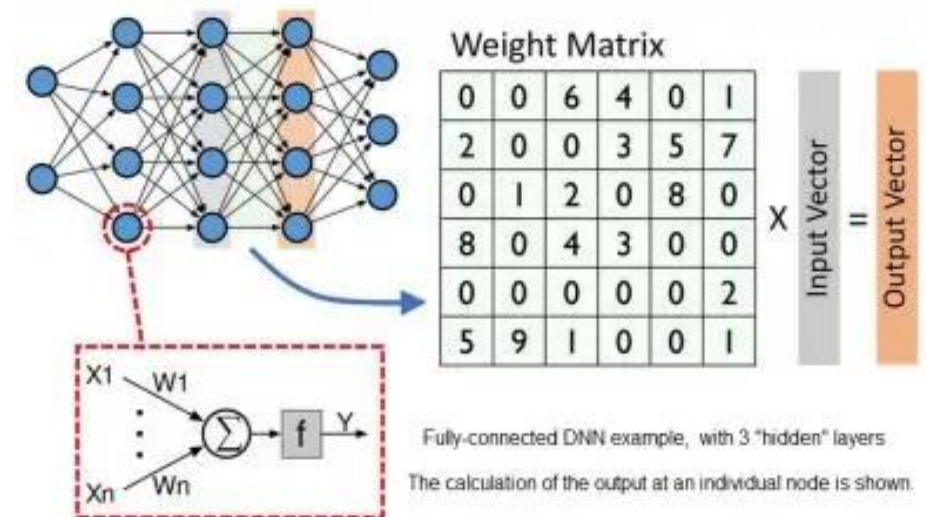
$X \rightarrow WX \rightarrow \text{activation function} \rightarrow Y$ (예측값)

역방향

Loss \rightarrow Backpropagation을 통한 가중치 업데이트

DNN 학습 방법

- 경사 하강법 (SGD, 모멘텀 SGD, ...)



$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial L}{\partial \mathbf{W}}$$

03 DNN – DNN의 문제점

DNN의 문제점

- **Overfitting**

파라미터 수가 많을 수록 training data에 과적합되기 쉽다.

- **높은 시간 복잡도**

심층 신경망을 학습시킬 때 모델 크기(계층 수, 계층 당 유닛 수), 학습률, 초기 가중치 등 많은 매개변수를 고려 → 오차역전파법, 경사 하강법을 사용해 최적의 매개변수를 찾기 위해 매개변수 공간 전부를 확인하는데 많은 시간, 자원 필요

해결

- Regularization (L1, L2)
- Mini batch, drop out, GPU

참고

사진 및 정의

<https://brunch.co.kr/@gdhan/10>

<https://deepai.org/machine-learning-glossary-and-terms/synapse>

<https://semiwiki.com/efpga/flex-logix/7796-architecture-for-machine-learning-applications-at-the-edge/>

http://tcpschool.com/deeplearning/deep_algorithm1

<https://ebbnflow.tistory.com/119>

감사합니다.