

# 241031 AI Study about Neural Network

## A. 알고리즘 설명 정리

### Neural Network의 정의

- Neural Network란?
  - : 인간의 뇌가 정보를 처리하는 방식을 모방한 인공지능 모델
  - Node 또는 Neuron이라 불리는 여러 개의 단위로 구성된 구조
  - 각 neuron은 입력을 받아 계산한 후 출력하는 역할을 하며 neuron간의 연결은 가중치(weight)로 표현됨
  - weight를 업데이트 하는 것을 학습이라고 함
- Neural Network의 계층(Layer)
  - 입력층 (Input Layer)
    - : 데이터를 네트워크에 입력하는 역할을 하는 계층
    - Input data는 이 층에서 각 neuron으로 전달됨
  - 은닉층 (Hidden Layer)
    - : 입력을 처리하고 변환하는 중간 단계 역할의 계층
    - 여러 개의 은닉층이 존재할 수 있음
    - 복잡한 패턴을 학습하는 층으로 역할함
  - 출력층 (Output Layer)
    - : 최종 결과를 출력하는 층
    - 분류 결과, 혹은 회귀 값을 출력함
- Neural Network의 구성 요소
  - 활성화 함수 (Activation Function)
    - : Neuron의 출력 값을 non-linear값으로 변환하여 다음 층의 입력으로 전달
    - ReLU, Sigmoid, Tanh 등이 이용됨
  - ReLU (Rectified Linear Unit)
    - :
    - $f(x) = \max(0, x)$ 로 표현됨
    - 계산이 간단하여 대규모 neural network에서 사용됨
    - 학습을 빠르게 진행할 수 있다는 장점이 존재

- 음수 영역에서 0을 출력하기 때문에 weight update가 되지 않는 Dead ReLU Problem이 존재

- Sigmoid

:

$f(x) = \frac{1}{1+e^{-x}}$ 로 표현됨

- 출력값을 확률과 같이 해석할 수 있음
- Deep Neural Network에서 입력값이 매우 크거나 작을 경우 gradient가 0에 수렴

- Tanh

:

$f(x) = \tanh(x)$ 로 표현됨

- 중심이 0인 출력을 이용하므로 학습 속도가 좋음
- Gradient 소실 문제 존재

- 가중치 (Weight)

: Neuron간 연결 강도를 나타냄

- 학습 중 update됨으로써 network가 더 정확한 예측을 하게 함

- 바이어스 (Bias)

: Neuron의 임계값을 설정함

## Neural Network의 학습 과정

- Neural Network는 Forward Propagation, Backward Propagation을 통해 예측 및 학습을 진행함
  - 순전파 (Forward Propagation)
    - : Input data가 input layer에서 output layer까지 전달되며 weight와 bias를 통해 출력값을 결정
  - 손실 함수 (Loss Function)
    - : 출력값과 실제 값의 차이를 계산하는 함수
  - 역전파 (Backward Propagation)
    - : Loss function의 결과를 바탕으로 output layer부터 input layer까지 올라가며 weight를 업데이트함
    - Neural Network의 학습 과정이라 할 수 있음

## Code 실험

- Torch와 torchvision을 이용해 실험해 볼 수 있었음
- MNIST dataset을 통해 학습 및 실험을 진행했으며  
28\*28 image set을 받아 숫자를 출력하는 neural network를 제작함
- 자세한 내용은 코드로 첨부

```
▶ train(model, train_loader, criterion, optimizer)
[8] ✓ 1m 42.5s

... Epoch [1/10], Loss: 1.0511
    Epoch [2/10], Loss: 0.3863
    Epoch [3/10], Loss: 0.3266
    Epoch [4/10], Loss: 0.2935
    Epoch [5/10], Loss: 0.2671
    Epoch [6/10], Loss: 0.2445
    Epoch [7/10], Loss: 0.2241
    Epoch [8/10], Loss: 0.2052
    Epoch [9/10], Loss: 0.1883
    Epoch [10/10], Loss: 0.1745

▶ test(model, test_loader)
[9] ✓ 1.6s

... Test Accuracy: 95.16%
```