



# 02. AI, ML, DL(머신러닝 학습 방법, Ragularization, Sigmoid, ReLU, 딥러닝 훈련 원리, Confusion Matrix)

## ▼ 개요

- Function
- Machine Learning
  - 머신러닝의 3요소
  - 머신러닝의 대표적인 학습 방법
  - 머신러닝 Workflow
  - 머신러닝의 목적
  - Supervised Learning(지도학습)
  - Unsupervised Learning(비지도학습)
  - Semi-supervised Learning(준지도학습)
  - Reinforcement Learning(강화학습)
  - 주의해야 할 점
  - How to solve overfitting
  - Ragularization
- Deep Learning
  - Biological Neuron
  - Perceptron
  - Linear Function of Perceptron
  - Activation Function(활성 함수)
  - Step Function
  - Sigmoid
  - ReLU(Rectified Linear Unit)
  - GeLU(Gaussian Error Linear Unit)
  - Softmax
- History of Deep Learning
  - Single Layer Perceptron(SLP, 단층 퍼셉트론)

- Multi Layer Perceptron(MLP, 다층 퍼셉트론)
  - Deep Neural Network
  - Transformer 아키텍처의 등장
  - Deep Learning Training Principles
    - 딥러닝 훈련 원리
    - Forward Propagation(순전파)
    - Loss Calculation(손실 계산)
    - 손실함수(Loss Function) : Error(차이)를 계산하는 방법
    - Gradient Descent(경사하강법)
    - Backpropagation(역전파)
    - Weight Update(가중치 업데이트)
  - Confusion Matrix
- 

## ■ Function

- 1차 함수 그래프 : 선형(Linear) 그래프
- 2차 함수 : 특징(Feature)이 2개

## ■ Machine Learning

- 데이터를 통해 스스로 성능을 개선하는 인공지능 구현 방법

### ▫ 머신러닝의 3요소

- 데이터 기반 학습
- 패턴 인식
- 자율적 개선

### ▫ 머신러닝의 대표적인 학습 방법

- Supervised Learning(지도학습)
- Unsupervised Learning(비지도학습)
- Semi-supervised Learning(준지도학습)
- Reinforcement Learning(강화학습)

## ▣ 머신러닝 Workflow

- Raw Data Set
  - 가공되지 않은 데이터(원시데이터)
- Feature Extraction
  - 특징 추출
  - 복잡한 데이터에 대한 이해도가 떨어짐
  - 표준정규분포표로 가공
- Algorithm Study
  - 알고리즘을 사용해서 학습
- Output
  - 결과 도출

## ▣ 머신러닝의 목적

| 모델의 일반화 능력을 향상시키는 것

- Generalization Ability(일반화 능력)
  - 학습된 모델이 데이터의 패턴을 파악하는 능력

## ▣ Supervised Learning(지도학습)

- 반복 학습을 통해 예러를 줄이며 모델을 개선시키는 학습 방법
  - 학습목적 : 모델의 예러 최소화
- 정답(Label)이 있는 데이터로 학습시키는 방법
- 예러 발생 시 더 많은 데이터로 학습 또는 재학습
  - 예러(Loss) : 모델이 예측 값과 정답 데이터의 차이
- 대표 알고리즘
  - Classification
  - Regression
  - Decision Tree
  - SVM

- K-NN

## ▫ Unsupervised Learning(비지도학습)

- Label이 없는 데이터로 학습시키는 방법
- 패턴, 관계를 찾아내는 학습 방법(최근에는 거의 사용X)
  - 뉴스 주제 분류, 이상거래 탐지
- 대표 알고리즘
  - Clustering
  - Dimensionality Reduction

## ▫ Semi-supervised Learning(준지도학습)

- Label이 있는 일부 데이터와 Label이 없는 대량의 데이터로 학습시키는 방법
  - Label 데이터는 약 20% 정도
- 데이터 양은 많은데, Labeling 비용이 비싼 분야
  - 의료 분야
- 대표 알고리즘
  - Self-Training
  - Co-Training

## ▫ Reinforcement Learning(강화학습)

- 에이전트가 환경속에서 시행착오를 통해 보상을 최대화하는 최적의 행동을 학습하는 머신러닝 기법
  - 에이전트(Agent) : 학습하고 행동하는 주체
  - 환경(Environment) : 에이전트가 상호작용하는 세계
  - 보상(Reward) : 특정 행동에 대한 긍정적 또는 부정적 피드백(점수)
  - 행동(Action) : 에이전트가 특정 상태에서 취하는 선택
- 사용사례
  - 자율주행, 게임 AI, 시뮬레이션 등
- 강화학습의 목적 : 에이전트의 **보상 최대화**

- 강화학습 모델의 성능 평가 : 에이전트의 행동 최적화

## ▫ 주의해야 할 점

- Overfitting(과적합)
  - 모델이 학습 데이터를 지나치게 외우는 현상
  - 학습한 데이터에 대해서는 성능이 매우 좋지만, 새로운 데이터에서는 성능이 급격하게 떨어짐 → 일반화 능력이 떨어짐
- Black Box 문제
  - 결과는 맞는데 이유를 설명할 수 없음
- Dependent on data quality(데이터 품질에 의존적)
  - GIGO : Garbage In, Garbage Out
  - 데이터 수집, 정제, Labeling이 모델 설계보다 중요한 경우가 많음

## ▫ How to solve overfitting

- Regularization(정규화)
  - 모델의 복잡도에 패널티를 부여하여, weight(가중치)가 커지는 것을 방지
- Feature Selection
  - Feature 수 줄이기
  - Feature 수가 많으면 가설 함수의 차수가 높아짐
  - Underfitting 조심

## ▫ Ragularization

- 손실 함수에 가중치의 합을 더하는 것
- Norm : 가중치 측정 방법
- L1 : Lasso Ragularization
  - 가중치의 절대값 합에 비례하는 패널티를 손실함수에 추가
  - 일부 가중치를 0으로 만듦
  - 모델의 일부 특성을 사실상 제거하며, Feature Selection에 유용
- L2 : Ridge Ragularization

- 가중치의 제곱 합에 비례하는 패널티를 손실함수에 추가
- 가중치 값이 전체적으로 너무 커지지 않도록 제한

## ■ Deep Learning

| 여러 layer의 신경망을 사용하여 학습하는 방법

### ▫ Biological Neuron

- Dendrite(수상돌기) : 전기신호 입력 층
- Axon(축삭돌기) : 신호 전달 통로
- Synapse(시냅스) : 뉴런 간 연결 지점
  - 자주 사용하는 부분은 증폭

### ▫ Perceptron

| 뉴런을 따라 만든 인공신경망의 최소 단위



입력값( $x$ )에 가중치( $w$ )를 곱하고 편향( $b$ )를 더한 뒤,  
활성화 함수 필터를 통과시켜 최종 신호를 만들어내는 수학적 모델

### ▫ Linear Function of Perceptron

$$y = \sum_{i=1}^n w_i x_i + b$$

- Activation Function에 전달하기 전에 Raw Score를 생성하는 과정
- **Weight**(가중치) : 데이터의 중요도
- **Bias**(편향) : 데이터의 민감도
- $w_i x_i + b = y = ax + b$ 
  - 1차 함수 : Linear Function
  - 가중합 + 편향
- Linear Combination(선형결합)

- Hyperplane(초평면) : n-1 차원에 존재

## ▫ Activation Function(활성 함수)

- Linear Function의 점수를 보고 다음 Perceptron으로 보낼지 말지 결정하는 함수
  - 데이터를 얼마나 전달할지 결정
  - 비선형성 부여
  - 결과를 결정하는데 기여(출력층, 분류 시)
- Activation Function의 종류
  - Step Function, Sigmoid, ReLU, Softmax

## ▫ Step Function

- 1 또는 0(Binary Step Function)
- 계단형 그래프

## ▫ Sigmoid

- 0과 1사이의 부드러운 확률 표현
- S 모양 그래프
- 이진분류 시 사용

## ▫ ReLU(Rectified Linear Unit)

- 종합점수가 마이너스면 0, 플러스면 점수 그대로
- 0이하는 0, 0초과 시 기울기가 1인 직선

## ▫ GeLU(Gaussian Error Linear Unit)

- GPT, BERT 모델에서 사용
- 입력값( $x$ )가 얼마나 큰지에 따라 통과시킬 확률 결정

## ▫ Softmax

- 출력층에서 사용되는 활성함수(다중분류)
- 확률화

- 모든 출력값을 0과 1사이로 압축
- 모든 출력값의 합이 1이 되게 만듦

## ■ History of Deep Learning

### ▫ Single Layer Perceptron(SLP, 단층 퍼셉트론)

- 1958년 Frank Rosenblatt이 구현
- 인간의 뉴런을 성공적으로 모방
- AND, OR와 같은 기본적인 논리연산은 잘함
- 1969년 Marvin L. Minsky에 의해 단층 퍼셉트론은 XOR 연산을 못함이 증명됨

### ▫ Multi Layer Perceptron(MLP, 다층 퍼셉트론)

- 여러 층의 퍼셉트론을 순차적으로 쌓아 올린 인공신경망
  - Input Layer(입력층)
  - Hidden Layer(은닉층)
  - Output Layer(출력층)
- 복잡한 비선형 문제 해결 가능
- 1974년 Backpropagation(역전파) 알고리즘 논문 등장
- 1986년 Geoffrey Hinton이 Backpropagation으로 MLP 학습 증명
- 1990년대 Vanishing Gradient(기울기 소실) 문제로 한계를 느낌
- 데이터 부족 문제

### ▫ Deep Neural Network

- 2000년대 정보화 시대가 되면서 데이터 폭증
- 2010년 Geoffrey Hinton이 학습에 ReLU 사용하여 논문 발표
- 2012년 GPU의 재발견
- 컴퓨터 클러스터 기술의 발전

### ▫ Transformer 아키텍처의 등장

- 2017년 Google DeepMind 팀 Attention Is All You Need 논문 발표
- GPT(Generative Pre-trained Transformer)
- BERT(Pre-training of Deep Bidirectional Transformers for Language Understanding)

## ■ Deep Learning Training Principles

### ▫ 딥러닝 훈련 원리

- Forward Propagation(순전파)
- Loss Calculation(손실 계산)
- Backpropagation(역전파)
- Weight Update(가중치 업데이트)

### ▫ Forward Propagation(순전파)

- 입력 데이터가 가중치와 활성 함수를 거쳐 은닉층을 통과하고 출력층에서 예측결과 (Prediction)를 생성

### ▫ Loss Calculation(손실 계산)

- 생성된 예측결과와 실제 정답을 비교해서 Loss Function(손실함수)을 통해 Error를 계산
  - Error(오차) : 정답 데이터와 예측 데이터 사이의 차이(정답 - 예측)  $y - \hat{y}$

### ▫ 손실함수(Loss Function) : Error(차이)를 계산하는 방법

- MAE (Mean Absolute Error : 평균 절대 오차)
- MSE (Mean Squared Error : 평균 제곱 오차)

### ▫ Gradient Descent(경사하강법)

- 오차를 줄이면서 모델이 더 나은 예측을 할 수 있도록 최적화하는 과정

### ▫ Backpropagation(역전파)

- 계산된 오차를 줄이기 위해, 오차를 신경망 역방향으로 이동하며, 각 층의 가중치에 대한 오차 기울기를 계산
- Chain Rule에 의거해서 각 가중치에 대한 미분
  - Chain Rule : 합성함수의 미분을 계산하는 수학적 규칙

## ▫ Weight Update(가중치 업데이트)

- 역전파 단계에서 계산된 기울기를 이용해서 경사하강법으로 가중치 조정

## ■ Confusion Matrix

| 분류 모델의 예측 성능을 평가하기 위해,  
실제 값과 모델의 예측 값을 비교하여 표 형태로 시각화한 행렬

- T(True) : 실제 정답
- F(False) : 실제 오답
- P(Positive) : 인공지능이 정답이라고 답변
- N(Negative) : 인공지능이 오답이라고 답변

	실제 정답	실제 오답
예측 정답	True Positive(TP)	False Positive(FP)
예측 오답	False Negative(FN)	True Negative(TN)

- Accuracy(정확도) :  $(TP+TN) / \text{전체}$
- Recall(재현율, 민감도) :  $TP / (TP+FN)$
- Precision(정밀도) :  $TP / (TP+FP)$
- F1-Score =  $2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$

| Recall과 Precision은 trade-off 관계