

# Week1\_예습과제\_팽소원

## 1장 머신러닝과 딥러닝

### 1.1 인공지능, 머신러닝과 딥러닝

**인공지능** : 인간의 지능을 모방하여 사람이 하는 일을 기계가 할 수 있도록 하는 기술

인공지능 > 머신 러닝 > 딥러닝

머신 러닝과 딥러닝 모두 학습 모델을 제공하여 데이터를 분류할 수 있는 기술이다

**머신러닝** : 주어진 데이터를 인간이 먼저 처리

데이터의 특징을 스스로 추출하지 X → 이 과정을 사람이 처리

**딥러닝**: 대량의 데이터를 신경망에 적용하면 컴퓨터가 스스로 분석한 후 답을 찾음

♥ 표 1-1 머신 러닝과 딥러닝

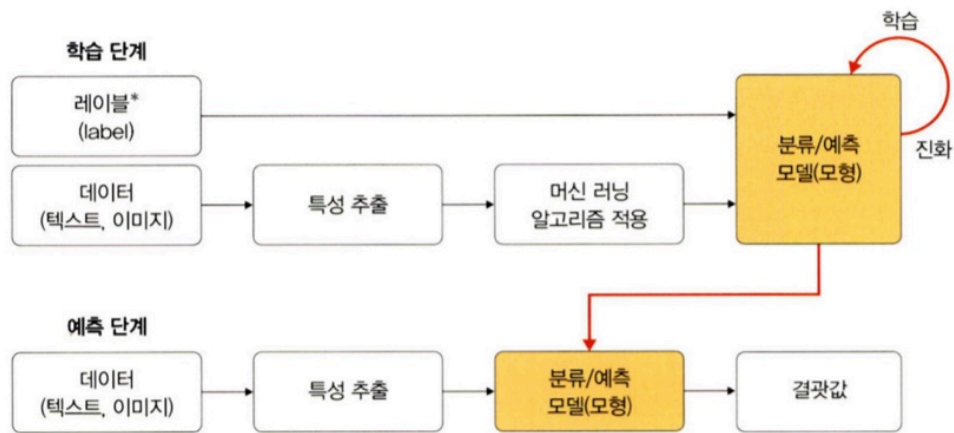
구분	머신 러닝	딥러닝
동작 원리	입력 데이터에 알고리즘을 적용하여 예측을 수행한다.	정보를 전달하는 신경망을 사용하여 데이터 특징 및 관계를 해석한다.
재사용	입력 데이터를 분석하기 위해 다양한 알고리즘을 사용하며, 동일한 유형의 데이터 분석을 위한 재사용은 불가능하다.	구현된 알고리즘은 동일한 유형의 데이터를 분석하는 데 재사용된다.
데이터	일반적으로 수천 개의 데이터가 필요하다.	수백만 개 이상의 데이터가 필요하다.
훈련 시간	단시간	장시간
결과	일반적으로 점수 또는 분류 등 숫자 값	출력은 점수, 텍스트, 소리 등 어떤 것이든 가능

### 1.2 머신 러닝 이란

#### 1.2.1 머신 러닝 학습 과정

머신 러닝은 크게 학습단계와 예측 단계로 구분

▼ 그림 1-3 머신 러닝 학습 과정



\* 레이블은 지도 학습에서 정답을 의미

## 머신 러닝의 주요 구성 요소

### • 데이터

머신 러닝이 학습 모델을 만드는 데 사용  
데이터의 80%: 훈련용, 20%: 테스트용

### • 모델

머신 러닝의 학습 단계에서 얻은 최종 결과물 = 가설  
모델의 학습절차

1. 모델 선택
2. 모델 학습 및 평가
3. 평가를 바탕으로 모델 업데이트

## 1.2.2 머신 러닝 학습 알고리즘

### • 지도 학습

정답이 무엇인지 컴퓨터에게 알려 주고 학습시키는 방법

### • 비지도 학습

정답을 알려 주지 않고 특징을 클러스터링하여 예측하는 방법

### • 강화학습

자신의 행동에 대한 보상을 받으며 학습을 진행

ex) 쿠키런 게임

▼ 표 1-2 지도 학습, 비지도 학습, 강화 학습

구분	유형	알고리즘
지도 학습 (supervised learning)	분류(classification)	<ul style="list-style-type: none"> <li>• K-최근접 이웃(K-Nearest Neighbor, KNN)</li> <li>• 서포트 벡터 머신(Support Vector Machine, SVM)</li> <li>• 결정 트리(decision tree)</li> <li>• 로지스틱 회귀(logistic regression)</li> </ul>
	회귀(regression)	선형 회귀(linear regression)
비지도 학습 (unsupervised learning)	군집(clustering)	<ul style="list-style-type: none"> <li>• K-평균 군집화(K-means clustering)</li> <li>• 밀도 기반 군집 분석(DBSCAN)</li> </ul>
	차원 축소 (dimensionality reduction)	주성분 분석 (Principal Component Analysis, PCA)
강화 학습 (reinforcement learning)	-	마르코프 결정 과정 (Markov Decision Process, MDP)

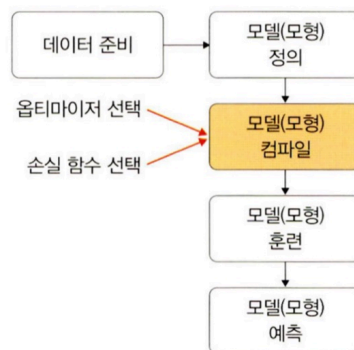
## 1.3 딥러닝이란

딥러닝은 인간의 신경망 원리를 모방한 심층 신경망 이론을 기반으로 고안된 머신 러닝 방법의 일종

머신 러닝과의 차이점은 인간의 뇌를 기초하여 설계했다는 것

### 1.3.1 딥러닝 학습 과정

▼ 그림 1-11 딥러닝 모델의 학습 과정



- 데이터 준비

초보자가 데이터를 쉽게 구할 방법

1. 파이토치나 케라스에서 제공하는 데이터셋 사용
2. 캐글에서 공개된 데이터 사용

- 모델 정의

신경망 생성

은닉층 수 ↑ ⇒ 성능 🍌

but 과적합 발생 확률 ↑

- 모델 컴파일

활성화 함수, 손실 함수, 옵티마이저 선택

- 모델 훈련

한 번에 처리할 데이터양 지정

- 모델 예측

검증 데이터셋을 생성한 모델에 적용하여 실제 예측을 진행

딥러닝은 심층 신경망을 사용한다는 점에서 머신 러닝과 차이가 있음

### 1.3.2 딥러닝 학습 알고리즘

#### 지도학습

- 합성곱 신경망

이미지 분류, 이미지 인식, 이미지 분할에 사용

- 순환 신경망(RNN)

시간에 따른 데이터가 있을 때 사용

역전파 과정에서 기울기 소멸 문제가 발생하여 세 개의 게이트를 추가 (LSTM)

-망각 게이트, 입력 게이트, 출력 게이트

#### 비지도 학습

- 워드 임베딩: 단어를 벡터로 표현
- 군집: 아무런 정보가 없는 상태에서 데이터를 분류하는 방법

#### 전이학습

사전에 학습이 완료된 모델을 가지고 우리가 원하는 학습에 미세 조정 기법을 이용하여 학습

사전 학습 모델 - 풀고자 하는 문제와 비슷하면서 많은 데이터로 이미 학습이 되어 있는 모델

#### 강화학습

머신러닝과 동일

▼ 표 1-3 지도 학습, 비지도 학습, 강화 학습

구분	유형	알고리즘
지도 학습(supervised learning)	이미지 분류	• CNN • AlexNet • ResNet
	시계열 데이터 분석	• RNN • LSTM
비지도 학습(unsupervised learning)	군집(clustering)	• 가우시안 혼합 모델(Gaussian Mixture Model, GMM) • 자기 조직화 지도(Self-Organizing Map, SOM)
	차원 축소	• 오토인코더(AutoEncoder) • 주성분 분석(PCA)
전이 학습(transfer learning)	전이 학습	• 버트(BERT) • MobileNetV2
강화 학습(reinforcement learning)	-	마르코프 결정 과정(MDP)

## 2장 실습 환경 설정과 파이토치 기초

### 2.1 파이토치 개요

파이토치는 파이썬 기반의 과학 연산 패키지로 다음 두 집단을 대상으로 함

1. 넘파이를 대체하면서 GPU를 이용한 연산이 필요한 경우
2. 최대한의 유연성과 속도를 제공하는 딥러닝 연구 플랫폼이 필요한 경우

#### 2.1.1 파이토치 특징 및 장점

특징 : GPU에서 텐서 조작 및 동적 신경망 구축이 가능한 프레임워크

GPU: 연산 속도를 빠르게 하는 역할

텐서 : 파이토치의 데이터 형태

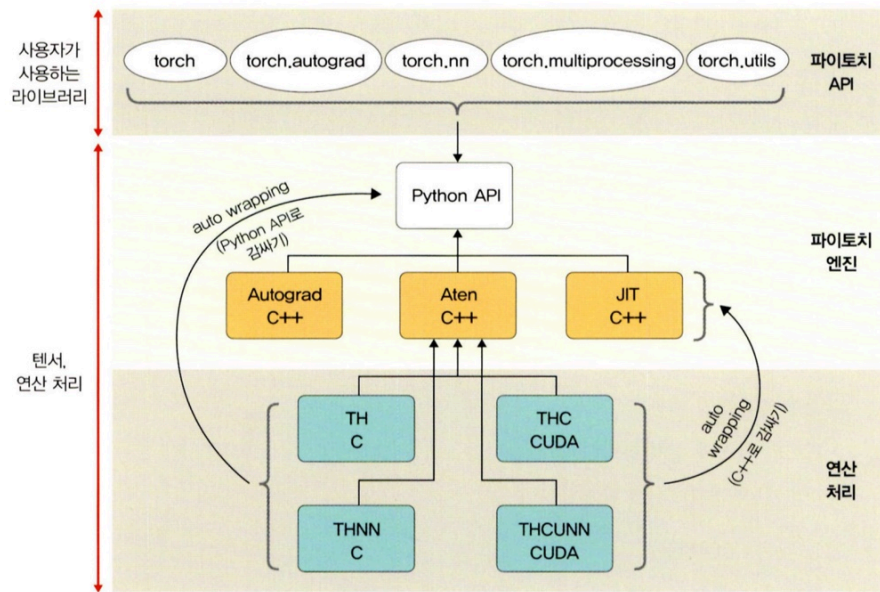
동적 신경망: 훈련을 반복할 때마다 네트워크 변경이 가능한 신경망

파이토치의 장점

1. 단순함 = 효율적인 계산
2. 성능 = 낮은 CPU 활용
3. 직관적인 인터페이스

#### 2.1.2 파이토치의 아키텍처

▼ 그림 2-4 파이토치의 아키텍처



## 파이토치 API

: 사용자가 이해하기 쉬운 API를 제공하여 텐서에 대한 처리와 신경망을 구축하고 훈련할 수 있도록 도움

실제 계산 수행 X

C++로 작성된 파이토치 엔진으로 작업을 전달하는 역할

## 파이토치 엔진

Autograd C++ : 가중치, 바이어스를 업데이트 하는 과정에서 필요한 미분을 자동으로 계산

Aten C++: C++ 텐서 라이브러리 제공

JIT C++: C++ 계산을 최적화하기 위한 JIT 컴파일러

## 연산 처리

가장 아래 계층에 속하는 C 또는 CUDA 패키지는 상위 API에서 할당된 거의 모든 계산을 수행