1장

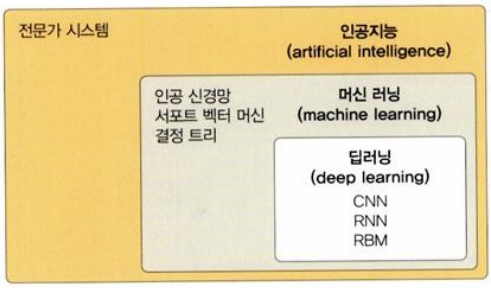
-인공지능, 머신 러닝과 딥러닝

인공지능 : 인간의 지능을 모방하여 사람이 하는 일을 컴퓨터(기계)가 할 수 있도록 하는 기술

머신러닝 : 주어진 데이터를 인간이 먼저 전처리

딥러닝 : 인간이 하던 작업을 생략, 대량의 데이터를 신경망에 적용하면 컴퓨터가 스스로 분석한 후 답을 찾음

❤ 그림 1-1 인공지능과 머신 러닝, 딥러닝의 관계

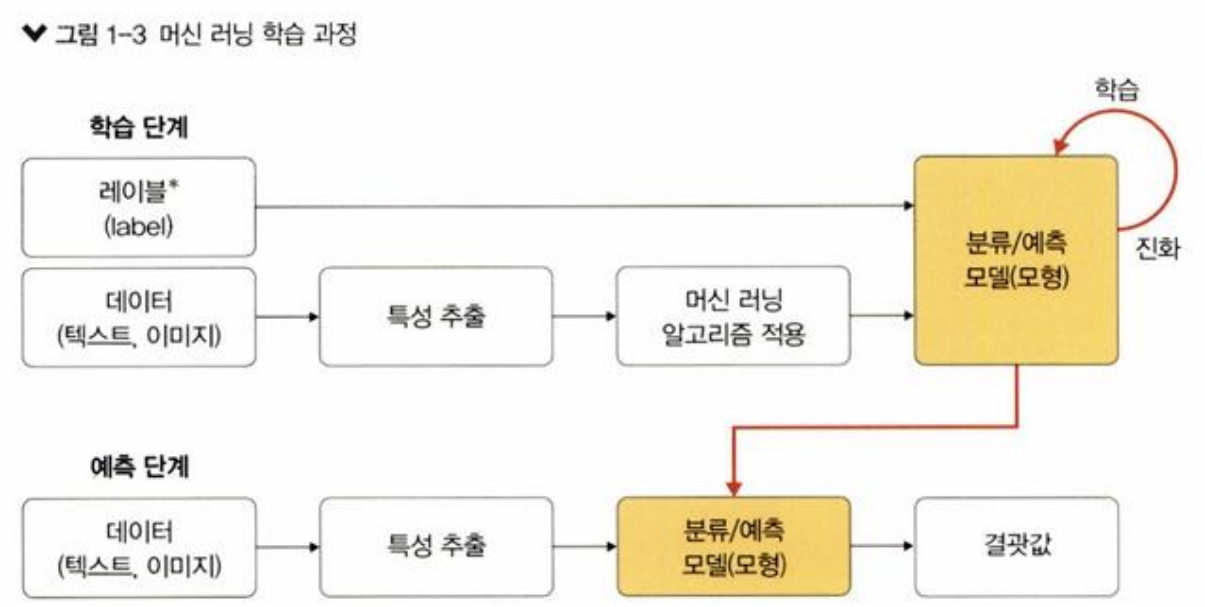


❤ 표 1-1 머신 러닝과 딥러닝

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

-머신러닝이란

머신 러닝은 인공지능의 한 분야로, 컴퓨터 스스로 대용량 데이터에서 지식이나 패턴을 찾아 학습하고 예측을 수행하는 것입니다. 즉, 컴퓨터가 학습할 수 있게 하는 알고리즘과 기술을 개발하는 분야라고 할 수 있습니다.



머신러닝의 주요 구성요소는 데이터와 모델(모형)입니다.

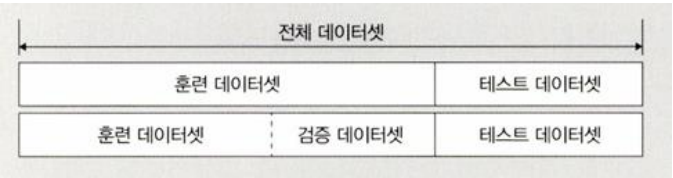
데이터 : 실제 데이터의 특징이 잘 반영되고 편향되지 않는 훈련 데이터를 확보하는 것이 중요

모델 : 머신 러닝의 학습 단계에서 얻은 최종 결과물로 가설이라고도 합니다.

모델의 학습 절차

1. 모델(또는 가설) 선책
2. 모델 학습 및 평가
3. 평가를 바탕으로 모델 업데이트

수집된 데이터셋은 크게 훈련과 테스트 데이터셋으로 분리하여 사용



검증 데이터셋을 사용하는 이유는 모델 성능을 평가하기 위해서입니다.

-머신 러닝 학습 알고리즘

지도학습, 비지도 학습, 강화 학습

지도 학습은 정답이 무엇인지 컴퓨터에 알려 주고 학습시키는 방법

비지도 학습은 정답을 알려주지 않고 특징이 비슷한 데이터를 클러스터링(범주화)하여 예측하는 학습 방법

강화학습은 자신의 행동에 대한 보상을 받으며 학습을 진행

❮ 표 1-2 지도 학습, 비지도 학습, 강화 학습

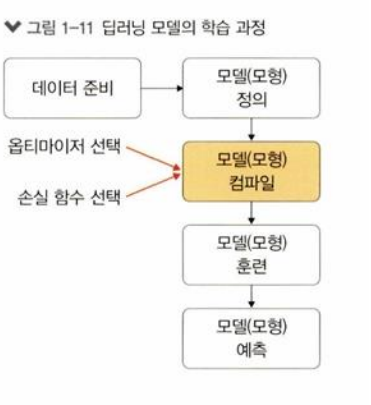
구분	유형	알고리즘
지도 학습 (supervised learning)	분류(classification)	• K-최근접 이웃(K-Nearest Neighbor, KNN) • 서포트 벡터 머신(Support Vector Machine, SVM) • 결정 트리(decision tree) • 로지스틱 회귀(logistic regression)
	회귀(regression)	선형 회귀(linear regression)
비지도 학습 (unsupervised learning)	군집(clustering)	• K-평균 군집화(K-means clustering) • 밀도 기반 군집 분석(DBSCAN)
	차원 축소 (dimensionality reduction)	주성분 분석 (Principal Component Analysis, PCA)
강화 학습 (reinforcement learning)	–	마르코프 결정 과정 (Markov Decision Process, MDP)

-딥러닝이란

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

-딥러닝 학습 과정

❮ 그림 1-11 딥러닝 모델의 학습 과정



딥러닝 학습 과정에서 가장 중요한 핵심 구성 요소는 신경망과 역전파입니다. 딥러닝은 머신 러닝의 한 분야이기는 하지만, 심층 신경망을 사용한다는 점에서 머신러닝과 차이가 있습니다. 심층 신경망에는 데이터셋의 어떤 특성들이 중요한지 스스로에게 가르쳐줄 수 있는 기능이 있습니다.

-딥러닝 학습 알고리즘

지도학습, 비지도학습, 전이학습

지도학습 알고리즘

합성곱 신경망 : 목적에 따라 이미지 분류, 이미지 인식, 이미지 분할로 분류할 수 있습니다.

순환 신경망 : 시계열 데이터를 분류할 때 사용되는 것

비지도 학습

워드 임베딩 : 단어 의미를 벡터화하는 워드투벡터와 글로브

군집 : 아무런 정보가 없는 상태에서 데이터를 분류하는 방법

전이 학습

사전 학습 모델 : 풀고자하는 문제와 비슷하면서 많은 데이터로 이미 학습이 되어 있는 모델

❯ 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

-파이토치 개요

넘파이를 대체하면서 GPU를 이용한 연산이 필요한 경우

최대한의 유연성과 속도를 제공하는 딥러닝 연구 플랫폼이 필요한 경우

-파이토치 특징 및 장점

GPU에서 텐서 조작 및 동적 신경망 구축이 가능한 프레임워크

GPU : 연산 속도를 빠르게 하는 역할

텐서 : 파이토치에서 텐서의 의미는 다음과 같습니다

텐서는 파이토치의 데이터 형태입니다.

텐서는 단일 데이터 형식으로 된 자료들의 다차원 행렬입니다.

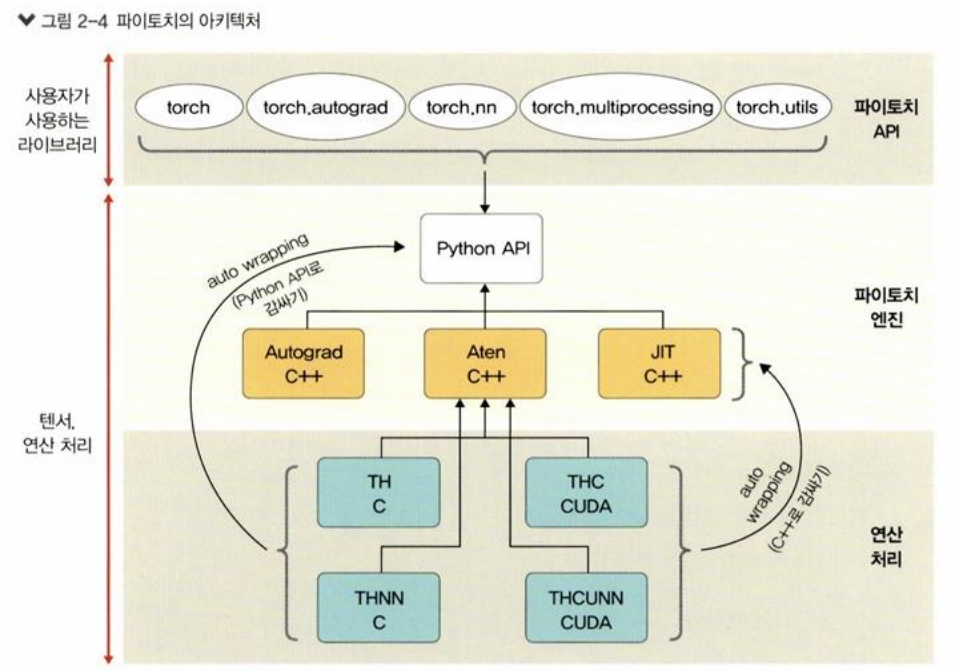
텐서는 간단한 명령어를 사용해서 GPU로 연산을 수행하게 할 수 있스 니다.

동적 신경장 : 훈련을 반복할 때마다 네트워크 변경이 가능한 신경망

파이토치는 효율적인 계산, 낮은 CPU 활용, 직관적인 인터페이스와 낮은 진입 장벽등을 장점으로 꼽을 수 있습니다.

단순함, 성능, 직관적인 인터페이스

-파이토치의 아키텍처



-파이토치 API

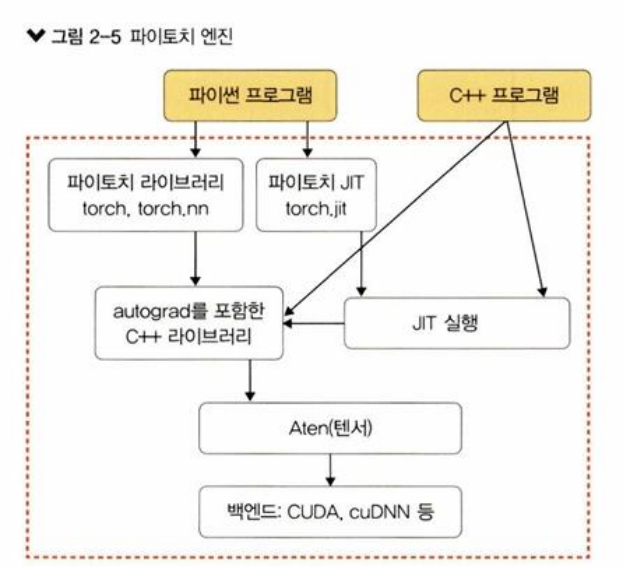
Torch : GPU를 지원하는 텐서 패키지

Torch.autograd : 자동 미분 패키지

Torch.nn : 신경망 구축 및 훈련 패키지

Torch.multoprocessing : 파이썬 멀티프로세싱 패키지

Torch.utils : dataloader 및 기타 유틸리티를 제공하는 패키지



-모델 정의

파이토치에서 모델을 정의하기 위해서는 모듈을 상속한 클래스를 사용합니다.

계층 : 모듈 또는 모듈을 구성하는 한 개의 계층으로 합성곱층, 선형 계층 등이 있습니다.

모듈 : 한 개 이상의 계층이 모여서 구성된 것으로, 모듈이 모여 새로운 모듈을 만들 수 있습니다.

모델 ; 최종적으로 원하는 네트워크로, 한 개의 모듈이 모델이 될 수도 있습니다.

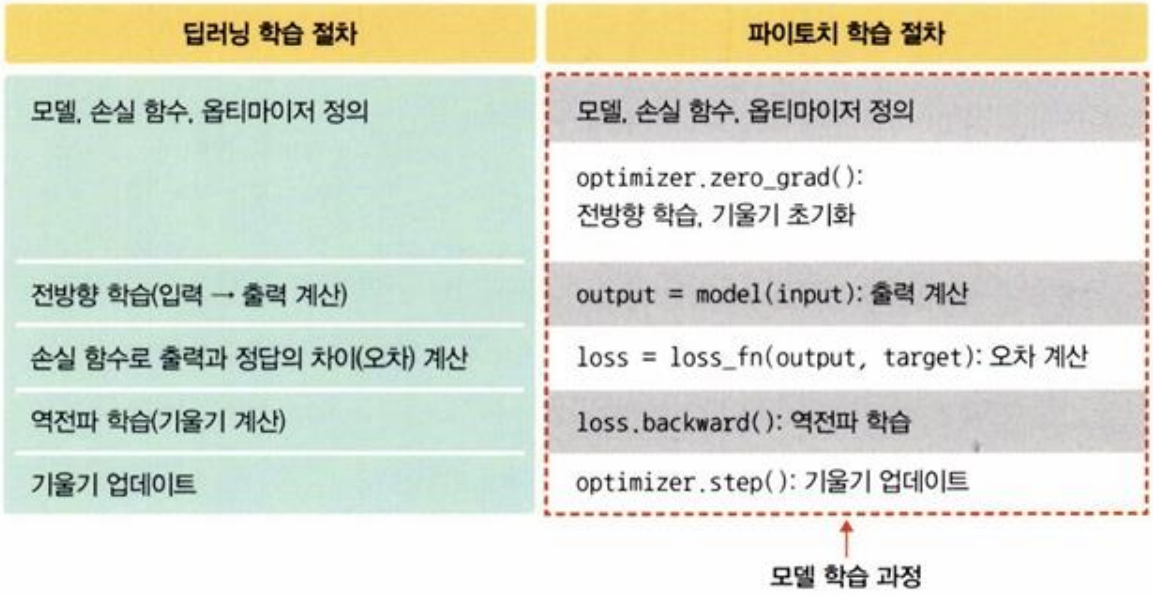
-모델의 파라미터 정의

손실 함수 : 학습하는 동안 출력과 실제 값 사이의 오차를 측정합니다.

옵티마이저 : 데이터와 손실 함수를 바탕으로 모델의 업데이트 방법을 결정합니다.

학습률 스케줄러 : 미리 지정한 횟수의 에포크를 지날 때마다 학습률을 감소시켜 줍니다. 학습률 스케줄러를 이용하면 학습 초기에는 빠른 학습을 진행하다가 전역 최소점 근처에 다다르면 학습률을 줄여서 최적점을 찾아갈 수 있도록 해 줍니다.

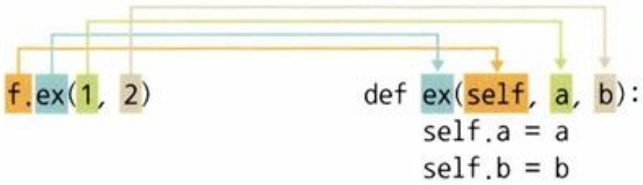
❯ 그림 2-12 파이토치 학습 절차




```
def __init__(self, embedding_size, output_size, layers, p=0.4)
             ⓐ          ⓑ             ⓒ        ⓓ     ⓔ
```

ⓐ self: 첫 번째 파라미터는 self를 지정해야 하며 자기 자신을 의미합니다. 예를 들어 ex라는 함수가 있을 때 self 의미는 다음 그림과 같습니다.

❯ 그림 2-31 self 의미



ⓑ embedding_size: 범주형 칼럼의 임베딩 크기

ⓒ output_size: 출력층의 크기

ⓓ layers: 모든 계층에 대한 목록

ⓔ p: 드롭아웃(기본값은 0.5)

딥러닝 분류 모델의 성능 평가 지표

- True Positive: 모델(분류기)이 '1'이라고 예측했는데 실제 값도 '1'인 경우입니다.
- True Negative: 모델(분류기)이 '0'이라고 예측했는데 실제 값도 '0'인 경우입니다.
- False Positive: 모델(분류기)이 '1'이라고 예측했는데 실제 값은 '0'인 경우로, Type I 오류라고도 합니다.
- False Negative: 모델(분류기)이 '0'이라고 예측했는데 실제 값은 '1'인 경우로, Type II 오류라고도 합니다.

정확도

전체 예측 건수에서 정답을 맞힌 건수의 비율입니다. 이때 맞힌 정답이 긍정(positive)이든 부정(negative)이든 상관없습니다.

$$\frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

재현율

실제로 정답이 1이라고 할 때 모델(분류기)도 1로 예측한 비율입니다. 따라서 처음부터 데이터가 1일 확률이 적을 때 사용하면 좋습니다.

$$\frac{True\ Positive}{True\ Positive + False\ Negative}$$

정밀도

모델(분류기)이 1이라고 예측한 것 중에서 실제로 정답이 1인 비율입니다.

$$\frac{True\ Positive}{True\ Positive + False\ Positive}$$

F1-스코어

일반적으로 정밀도와 재현율은 트레이드오프(trade-off) 관계입니다. 정밀도가 높으면 재현율이 낮고, 재현율이 높으면 정밀도가 낮습니다. 이러한 트레이드오프 문제를 해결하려고 정밀도와 재현율의 조화 평균(harmonic mean)을 이용한 것이 F1-스코어 평가입니다. 이때 조화 평균은 다음 공식으로 구할 수 있습니다.

$$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$