

# 1부 딥러닝의 기초 - 1장 딥러닝이란 무엇인가

📅 시작일	@2022년 7월 18일
🔗 링크	<a href="https://github.com/SEC-WBdle/Deep-Learning-with-Python">https://github.com/SEC-WBdle/Deep-Learning-with-Python</a>
📅 종료일	@2022년 7월 18일

## ▼ 1장 딥러닝이란 무엇인가?

### ▼ 1.1 인공 지능과 머신 러닝, 딥러닝

#### ▼ 1.1.1 인공 지능

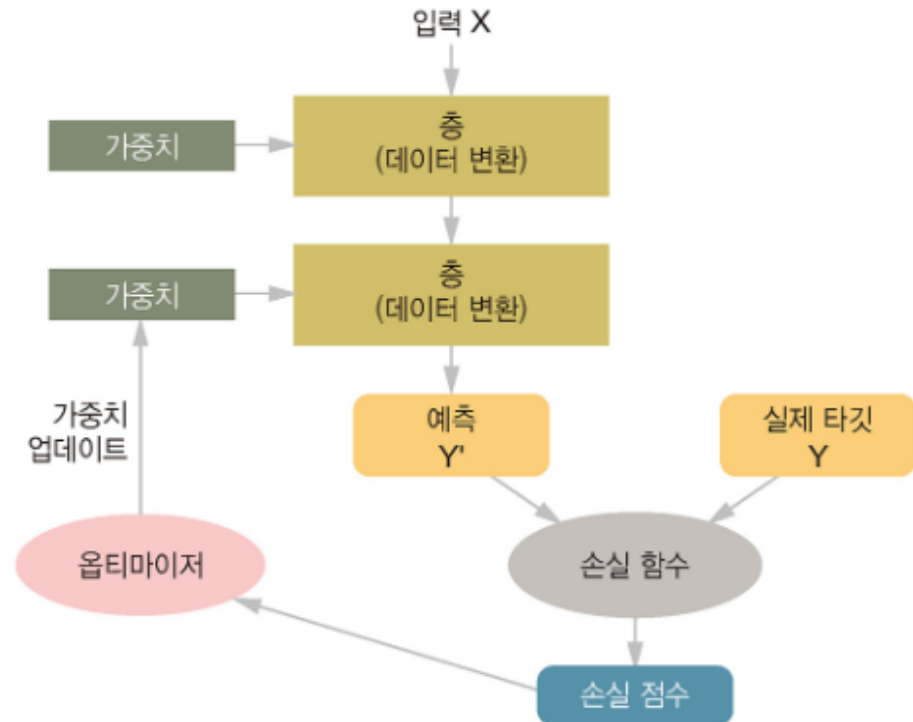
#### ▼ 1.1.2 머신 러닝

#### ▼ 1.1.3. 데이터에서 표현을 학습하기

#### ▼ 1.1.4 딥러닝에서 '딥'이란 무엇일까?

#### ▼ 1.1.5 그림 3개로 딥러닝의 작동 원리 이해하기

- 입력과 타겟을 매핑(mapping)하여 학습
- 가중치(weight)을 조정, 가중치를 파라미터로 가지는 함수로 표현됨
- 학습: 주어진 입력, 정확한 타겟에 매핑하기 위해 가중치 값을 찾는 것
- 출력 제어 - 신경망의 손실 함수(loss function) 또는 목적 함수(object function)
- 손실 점수가 감소되는 방향으로 가중치 값을 조금씩 수정함
- 수정 과정 : 역전파(Backpropagation)알고리즘을 구현한 옵티마이저(optimizer\_)가 담당함



▲ 그림 3-1 네트워크, 층, 손실 함수, 옵티마이저 사이의 관계

- ▼ 1.1.6 지금까지 딥러닝의 성과
- ▼ 1.1.7 단기간 과대 선전을 믿지 말자
- ▼ 1.1.8 AI에 대한 전망
- ▼ 1.2 딥러닝 이전
  - ▼ 1.2.1 확률적 모델링
    - ▼ 나이브 베이즈(Naive Bayes) 알고리즘
      - 입력 데이터의 특성이 모두 독립적이라고 가정하고 베이즈 정리를 적용하는 머신 러닝 분류 알고리즘
      - 밀접하게 연관된 모델이 로지스틱 회귀(logistic regression)
  - ▼ 1.2.2 초창기 신경망
  - ▼ 1.2.3 커널 방법
    - ▼ 커널 방법(Kernel method)
      - 분류 알고리즘의 한 종류
      - 서포트 벡터 머신(Support Vector Machine, SVM) 이 대표적

- 2개의 다른 범주에 속한 데이터 포인트 그룹 사이 좋은 결정 경계 (decision boundary) 를 찾음
- SVM이 결정 경계 찾는 과정
  - 결정 경계가 하나의 초평면(hyperplane) 으로 표현될 수 있도록 데이터 매핑
  - 데이터 포인트 사이의 거리가 최대가 되도록 마진 최대화 (maximizing the margin)
- 커널 기법(kernel trick)
  - 새롭게 표현된 공간에서 좋은 결정 초평면을 찾기 위해 새로운 공간에 대응하는 데이터 포인트의 좌표를 실제로 구할 필요가 없음
  - 커널 함수(Kernel function) 사용하여 효율적으로 계산



#### ▼ 1.2.4 결정 트리, 랜덤 포레스트, 그래디언트 부스팅 머신

##### ▼ 결정 트리

##### ▼ 랜덤 포레스트

- 결정 트리의 앙상블

##### ▼ 그래디언트 부스팅 머신

- 약한 예측 모델인 결정 트리를 앙상블 (놓친 데이터 포인트를 보완하여 훈련)

#### ▼ 1.2.5 다시 신경망으로

#### ▼ 1.2.6 딥러닝의 특징

- 머신 러닝: 특성 공학(feature engineering)을 통해 데이터의 좋은 표현을 수동으로 만들어야 함
- 딥러닝은 특성을 직접 찾는 대신, 한 번에 모든 특성을 학습
- 엔드-투-엔드(end-to-end) 딥러닝 모델로 대체
- 층을 거치면서 점진적으로 더 복잡한 표현이 만들어진다 : 이런 점진적인 중간 표현이 공동으로 학습된다

#### ▼ 1.2.7 머신 러닝의 최근 동향

#### ▼ 1.3 왜 딥러닝일까? 왜 지금일까?

### ▼ 1.3.1 하드웨어

### ▼ 1.3.2 데이터

### ▼ 1.3.3 알고리즘

- 신경망 층에 더 잘 맞는 **활성화 함수(activation function)**
- 층별 사전 훈련(pretraining)을 불필요하게 만든 **가중치 초기화(weight initialization) 방법**
- RMSProp 과 Adam 과 같은 더 좋은 **최적화 방법**

### ▼ 1.3.4 새로운 투자의 바람

### ▼ 1.3.5 딥러닝의 대중화

### ▼ 1.3.6 지속될까?

- 단순함 : 딥러닝 특성 공학이 필요하지 않아 복잡하고 불안정한 많은 엔지니어링 과정을 엔드-투-엔드로 훈련시킬 수 있는 모델로 바꾸어T口
- 확장성: 딥러닝 GPU/TPU에서 병렬화할 수 있음
- 다용도와 재사용성: 딥러닝 모델은 처음부터 다시 시작하지 않고 추가되는 데이터로도 훈련할 수 있음`