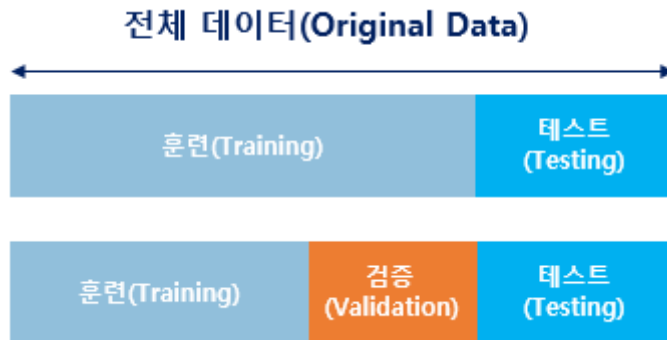


# 머신 러닝 훑어보기

머신 러닝의 특징에 대해 간략하게 설명

## 머신 러닝 모델의 평가












- 모델 평가를 위한 데이터 : 훈련 데이터, 검증 데이터, 테스트 데이터. (각 데이터를 비유하자면 훈련 데이터는 학습지, 검증 데이터는 모의고사, 테스트 데이터는 수능시험에 해당)
- 검증 데이터의 필요성 : 모델 성능을 조정하기 위한 용도 → 과적합이 되고 있는지 판단하거나 하이퍼파라미터의 조정을 위한 용도
  - hyperparameter : 값에 따라서 모델의 성능에 영향을 주는 매개변수로 모델링할 때 사용자가 직접 세팅하는 매개변수
  - parameter : weight와 bias와 같이 기계의 학습을 통해 바뀌는 변수
- 검증이 끝났다면, 테스트 데이터를 가지고 모델의 진짜 성능 평가
- 만약, 검증 데이터와 테스트 데이터를 나눌 만큼 데이터의 양이 충분하지 않다면, **K-Fold Cross Validation**을 사용하는 것이 좋음

## 분류(Classification)와 회귀(Regression)

머신 러닝의 많은 문제는 분류 또는 회귀에 속함

- 이진 분류 문제(Binary Classification)  
주어진 입력에 대해서 둘 중 하나의 선택지를 답으로 정하는 문제  
example : 스팸 메일 분류기(주어진 메일이 정상 메일인지, 스팸 메일인지 분류)
- 다중 클래스 분류 문제(Multi-class Classification)  
두 개 이상의 정해진 선택지 중에서 주어진 입력에 대한 한 선택지를 답으로 정하는 문제  
example : 과학, 영어, IT, 만화, 수학이라는 분야 중에서 주어진 한 책이 어느 분야(class)에 해당되는지 판단
- 다중 레이블 분류 문제(Multi-label Classification)  
두 개 이상의 정해진 선택지 중에서 주어진 입력에 대한 여러 개의 선택지를 답으로 정하는 문제  
example : 로맨스, 스릴러, 액션이라는 분야 중에서 주어진 한 영화가 해당되는 모든 분야(class) 선택

택

	Multi-Class	Multi-Label
<b>C = 3</b>		
	<b>Samples</b>  <b>Labels (t)</b> $[0 \ 0 \ 1]$	<b>Samples</b>  <b>Labels (t)</b> $[1 \ 0 \ 1]$
		
		
	<b>Samples</b>  <b>Labels (t)</b> $[1 \ 0 \ 0]$	<b>Samples</b>  <b>Labels (t)</b> $[0 \ 1 \ 0]$
	<b>Samples</b>  <b>Labels (t)</b> $[0 \ 1 \ 0]$	<b>Samples</b>  <b>Labels (t)</b> $[1 \ 1 \ 1]$

- 회귀 문제(Regression)  
분류문제처럼 분리된 답이 아니라 **연속된 값을 결과로 가짐**. 즉, **연속적인 숫자(실수)**를 예측.  
example : 시계열 데이터를 이용한 주가 예측, 생산량 예측

## 지도 학습(Supervised Learning)과 비지도 학습(Unsupervised Learning)

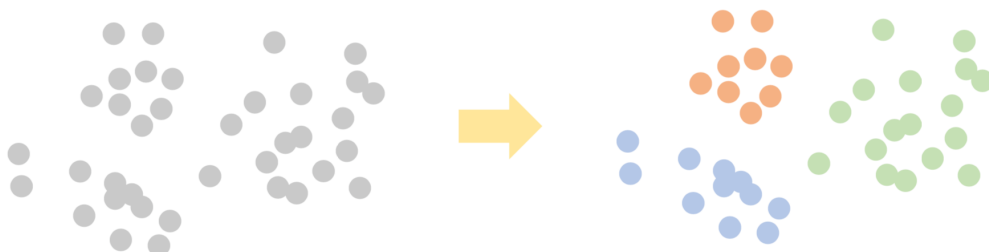
머신러닝은 크게 지도 학습, 비지도 학습, 강화 학습으로 나뉨

- 지도 학습  
레이블(label)과 함께 학습. (예측모델 등에 사용)
- 비지도 학습  
레이블(label) 없이 학습. (군집화, 차원축소 등에 사용)  
LDA, Word2Vec 또한 비지도 학습이다.

지도 학습(Supervised Learning) : 입력과 함께 ‘정답’을 알려주고 그 정답을 맞추도록 하는 학습 방법



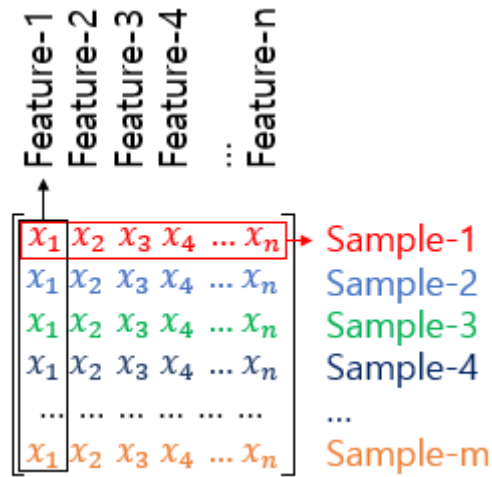
비지도 학습(Unsupervised Learning) : 정답의 제공 없이 학습 데이터로부터 유용한 정보를 추출하는 학습 방법



- 강화 학습  
시뮬레이션 반복 학습. (성능 강화 등에 사용)

## 샘플(Sample)과 특성(Feature)

인공신경망 모델은 연산을 주로 행렬 연산을 이용.



- 하나의 데이터, 즉 하나의 행을 **샘플(sample)**이라고 부름 (데이터베이스에서 레코드라고 부르는 단위)
- 종속 변수  $y$ 를 예측하기 위한 각각의 독립 변수  $x$ 를 **특성(feature)**라고 부름

## 혼동 행렬(Confusion Matrix)

- 머신러닝 성능 평가를 위한 지표

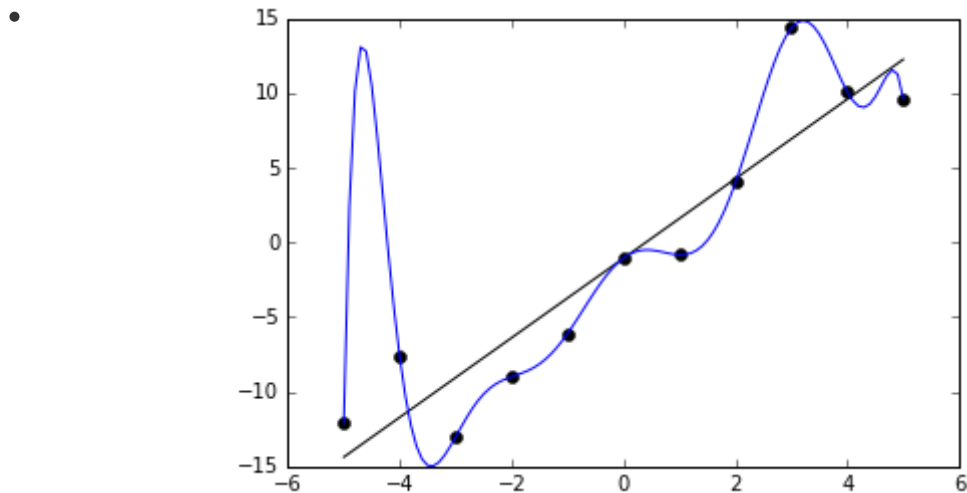
	positive (predicted)	negative(predicted)
positive (actual)	True Positive	False Negative
negative(actual)	False Positive	True Negative

- False Positive는 양성이라고 대답하였는데 실제 값이 음성이라서 틀린 경우
- False Negative는 음성이라고 대답하였는데 실제 값이 양성이라서 틀린 경우
- False Positive와 False Negative에 각각 초점을 맞춘 성능평가방법 : Precision, Recall (각각 False Positive, False Negative가 많아질수록 성능이 낮아짐)
- **정밀도(Precision)**
  - $TP / (FP + TP)$
  - $TP$  : 실제 Positive를 잘 판단한 경우
  - $(FP + TP)$  : 예측을 Positive로 한 모든 경우
  - **잘못된 positive를 줄이는 데에 초점**
  - example : 스팸메일 분류
    - 스팸을 스팸메일로 분류하지 않는 것(FN)은 큰 문제가 없음
    - 스팸메일이 아닌 것을 스팸메일로 분류하면(FP) 업무 차질 발생
    - 이와 같이 FN보단 FP를 줄이는 것이 중요한 경우 Precision 사용
- **재현율(Recall)**
  - $TP / (FN + TP)$
  - $TP$  : 실제 Positive를 잘 판단한 경우
  - $(FN + TP)$  : 실제 값이 Positive인 모든 경우
  - **잘못된 Negative를 줄이는 데에 초점**
  - example : 악성코드 판별
    - 악성코드가 아닌데 악성코드로 분류하면(FP) 사용자가 확인하고 예외처리 하면 됨

- 악성코드인데 악성코드가 아닌 것으로 분류하면(FN) 악성코드에 감염되어 위험 노출
- 이와 같이 FP보단 FN을 줄이는 것이 중요한 경우 Recall 사용

## 과적합(Overfitting)과 과소 적합(Underfitting)

- 과적합 : 모델이 훈련 데이터에 너무 잘 맞지만 일반성이 떨어지는 경우. 이러한 모델은 테스트 데이터에 대해 높은 성능을 보여줄 확률 낮음 ← 훈련데이터에 너무 맞춰져 훈련 데이터 이외의 다양한 변수에는 대응하기 어렵.



파란색 선은 overfitted model을 보여주고, 검은색 선은 regularized model을 의미

- 과소적합 : 테스트 데이터의 성능이 올라갈 여지가 있음에도 훈련을 덜 한 경우, 모델이 너무 단순해서 데이터의 내재된 구조를 학습하지 못하는 경우
- 과"적합", 과소"적합"이라고 부르는 이유 → 머신러닝에서 학습 또는 훈련 과정을 적합(fitting)이라고 부를 수 있기 때문.
- 과적합 예방법
  - Drop out
  - 더 많은 훈련데이터
  - 조기종료(Early Stopping)
  - 정규화

## 출처

- <https://wikidocs.net/32012>
- [http://itwiki.kr/w/혼동\\_행렬](http://itwiki.kr/w/혼동_행렬)
- <https://mc.ai/building-a-multi-label-text-classifier-using-bert-and-tensorflow/>
- [https://heung-bae-lee.github.io/2019/12/08/deep\\_learning\\_02/](https://heung-bae-lee.github.io/2019/12/08/deep_learning_02/)
- <https://en.wikipedia.org/wiki/Overfitting>