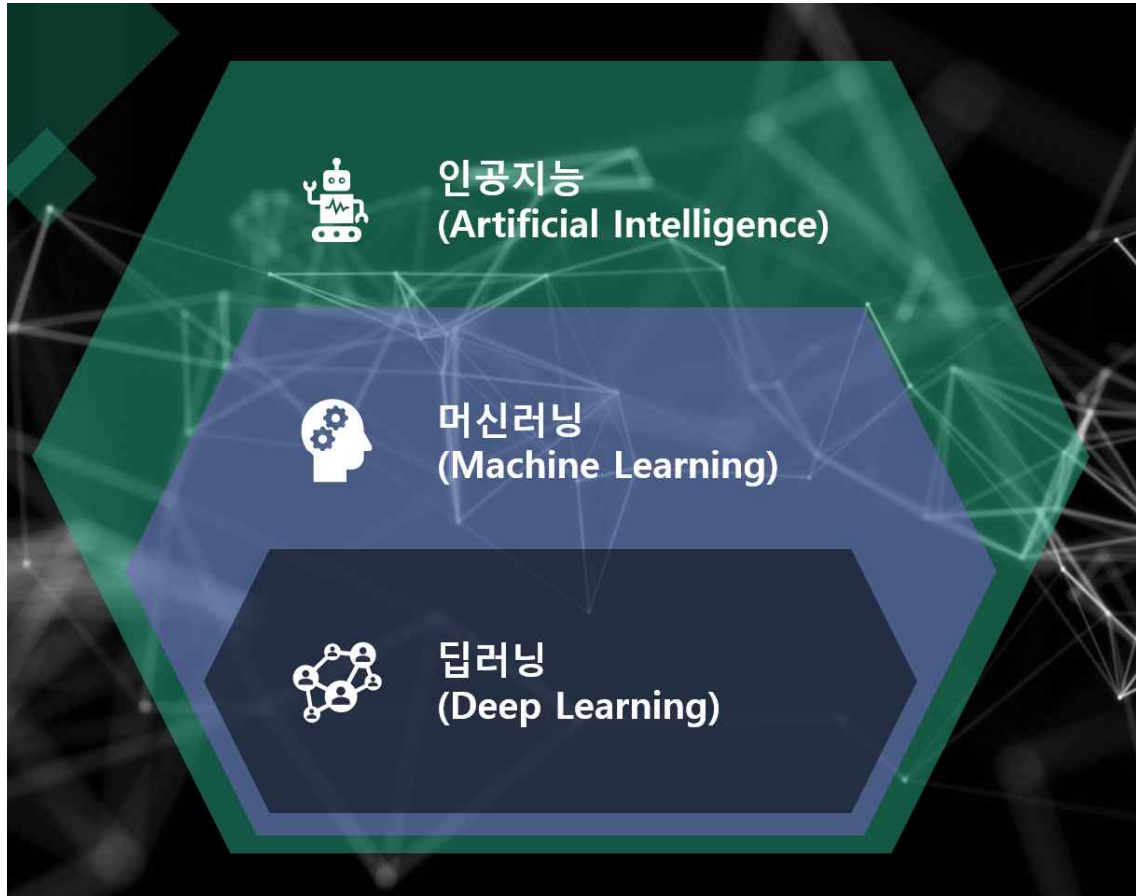


<부록 03. 인공지능기술>

인공 지능(Artificial Intelligence)



인공지능이란 학습, 추론, 인식 등 인간과 관계된 인지 문제를 컴퓨터 프로그램으로 구현한 기술입니다. 인공지능 용어의 첫 등장은 1956년 미국 다트머스 대학교에서 열린 학회에서 처음 사용되었습니다. 스탠퍼드 대학교의 존 매카시(John McCarthy) 교수가 인공지능이란 용어를 처음 사용하였습니다.

기존에 인간만이 실현할 수 있다고 생각한 역할을 컴퓨터가 수행할 수 있도록 구현하여 인위적으로 만들어진 지능을 뜻합니다. 즉, 인간의 지능을 기계나 프로그램 등에 인공적으로 구현한 것을 의미합니다. 인공지능을 크게 두 가지로 나누자면 강인공지능(Strong Artificial Intelligence)과 약인공지능(Weak Artificial Intelligence)이 있습니다.

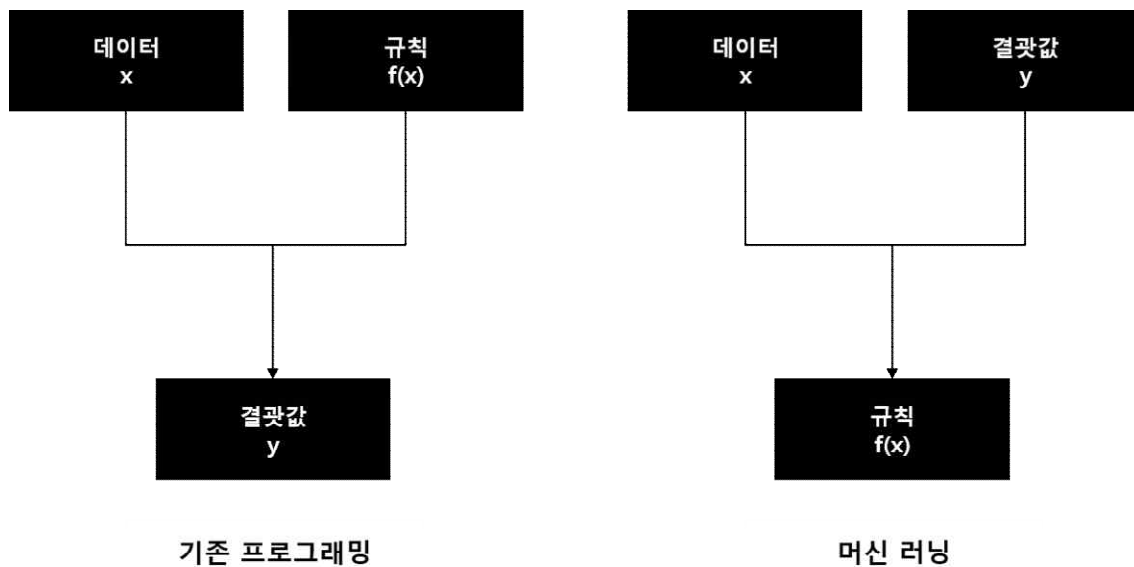
강인공지능은 스스로 학습과 인식 등이 가능하며, 지능 또는 지성의 수준이 인간과 근사한 수준까지 이른 경우를 뜻합니다. 주로 SF 영화 등에 나타나는 휴머노이드나 안드로이드를 생각할 수 있습니다.

약인공지능은 인간이 해결할 수 있으나, 기존의 컴퓨터로 처리하기 힘든 작업들을 처리하기 위한 일련의 알고리즘을 의미합니다. 현재 많은 곳에서 활용되고 있는 A.I. 서비스라고 볼 수

있습니다. 다시 말해, 주어진 시스템에서 입력을 조절해 출력을 원하는 대로 조절하는 제어기로부터 측정가능한 경험적(heuristic) 속성을 학습해 스스로 판단하는 심층 학습까지의 전반을 의미합니다.

인공지능은 인간과 비슷하거나 합리적 행동을 통해 특정한 문제를 해결하는데 중점을 두고 있습니다. 인공지능 분야에서 파생된 컴퓨터 과학 분야로는 머신 러닝(기계 학습, Machine Learning)과 딥 러닝(심층 학습, Deep Learning) 등이 있습니다.

머신 러닝(Machine Learning)

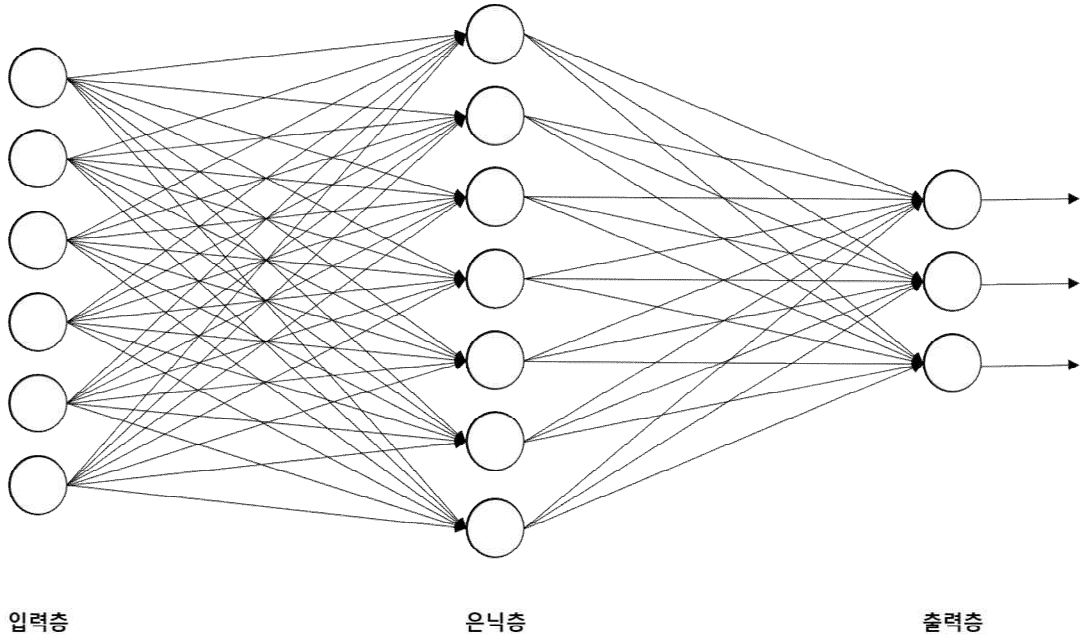


머신 러닝(Machine Learning)이란 데이터를 기반으로 컴퓨터를 프로그래밍하는 연구 분야입니다. 기존 프로그래밍은 명시적인 프로그래밍을 통해서 시스템을 구축하였지만, 머신 러닝은 데이터를 기반으로 학습하거나 시스템의 성능을 개선하는 데 중점을 두고 있습니다. 즉, 기존의 프로그래밍은 규칙과 데이터를 기반으로 결괏값을 예측했지만 머신 러닝은 데이터와 결괏값으로 규칙을 찾아내는 분야입니다.

머신 러닝은 학습 데이터를 분석하여 일정한 규칙이나 패턴을 찾아 예측 알고리즘을 생성합니다. 이 예측 알고리즘을 모델(Model)이라 하며, 모델에 새로운 데이터가 입력되었을 때 모델의 예측값으로 결과를 추론할 수 있습니다. 데이터를 기반으로 알고리즘을 구성하므로, 통계적인 접근 방법을 사용한다 볼 수 있습니다.

머신 러닝 알고리즘에는 지도 학습 (Supervised Learning), 비지도 학습(Unsupervised Learning), 강화 학습(Reinforcement learning), 심층 학습(Deep Learning) 등이 있습니다.

딥 러닝(Deep Learning)



딥 러닝(Deep Learning)은 머신 러닝의 기법 중 하나로, 모델이 스스로 데이터의 관계를 파악해 학습합니다. 여러 층(Layer)을 가진 인공 신경망(Artificial Neural Network, ANN)을 사용하여 머신 러닝 학습을 수행합니다. 여기서 인공 신경망은 인간의 뇌에서 있는 뉴런의 네트워크에서 영감을 얻은 통계학적 학습 알고리즘입니다.

생물학적 뉴런은 단순하게 다른 뉴런에게 신호를 받아 또 다른 뉴런에게 신호를 전달합니다. 뉴런은 수십억 개 이상 구성된 네트워크로 이뤄져 있어 신호의 흐름으로 복잡하고 다양한 활동을 할 수 있게합니다.

결국, 신경망은 서로 연결된 노드(Node)의 집합으로 구성되어 있으며 여러 층으로 이뤄집니다. 층(Layer)은 크게 입력층(Input Layer), 은닉층(Hidden Layer), 출력층(Output Layer)이 존재합니다. 입력층으로 학습하고자 하는 데이터를 받고, 여러 개의 은닉층을 지나 단계를 거쳐 출력층에서 결과를 반환합니다.

인공 신경망에 학습 알고리즘과 데이터를 지속적으로 제공함으로써, 학습 능력과 사고 능력을 지속적으로 개선됩니다.

딥러닝 알고리즘에는 합성곱 신경망(Convolutional Neural Network, CNN), 순환 신경망(Recurrent Neural Network, RNN), 심층 신경망(Deep Neural Network, DNN) 등이 있습니다.

Tip : 딥(Deep)이란, 지속적인 개선으로 얻어지는 신경망의 은닉 층(Hidden Layer)을 의미합니다.

지도 학습(Supervised Learning)



지도 학습이란 컴퓨터(알고리즘)에 훈련 데이터(train data)와 레이블(Label)을 포함시켜 학습을 하는 방법입니다. 여기서 훈련 데이터는 입력 데이터(input data)가 되며, 레이블은 정답(label data)이 됩니다. 즉, 훈련 데이터로부터 하나의 함수를 유추해내기 위한 기계 학습의 한 방법입니다.

지도 학습의 훈련 데이터(train data)는 일반적으로 입력 데이터(input data)에 대한 속성을 벡터 형태로 포함하고 있으며, 각각의 벡터에 대해 원하는 결과(label data)가 무엇인지 표시되어 있습니다. 지도 학습에는 크게 회귀 분석(Regression)과 분류(Classification) 등이 있습니다.

회귀 분석(Regression)

회귀 분석은 연속형 변수들에 대해 두 변수 사이의 모형을 구한 뒤 적합도를 측정해 내는 분석 방법입니다. 로버스트(Robust), 라쏘(Lasso), 로지스틱(Logistic) 등을 비롯해, 합성곱 신경망(CNN), 순환 신경망(RNN)까지 회귀분석에 포함할 수 있습니다. 회귀 분석은 크게, 선형(Linear)과 비선형(Non-Linear)으로 나눌 수 있습니다.

여기서 선형은 직선의 특징을 갖고 있음을 뜻하며, 중첩의 원리(superposition principle)가 적용되는 것을 의미합니다. 즉, 비선형은 예측하기 힘든 값이 나와 함수의 수식을 예측하기가 어렵습니다. 그러므로, 비선형 회귀 분석을 진행할 때에는 CNN, RNN, DNN 등을 이용합니다.

선형 회귀는 다시 크게 단변량(Univariate)과 다변량(Multivariate)으로 나뉩니다. 단변량은 종속 변수(Y)가 하나일 때를 의미하며, 다변량은 종속 변수(Y)가 두 개 이상일 때를 의미합니다.

단변량에서는 종속 변수(Y)가 하나만 존재하므로, 독립 변수(X)가 하나 이상의 값을 사용할 수 있습니다. 여기서 독립 변수(X)가 하나라면 단순(Simple)이 되며, 두 개 이상이 되면 다중(Multiple)이 됩니다.

이를 다시 정리하자면, 다음과 같이 설명할 수 있습니다. 하나의 종속 변수와 하나의 독립 변수 사이의 관계를 분석하는 경우에는 단순 선형 회귀 분석(Simple Linear Regression Analysis)이라 합니다.

만약, 하나의 종속 변수와 여러 개의 독립 변수 사이의 관계를 분석한다면 다중 선형 회귀 분석(Multiple Linear Regression Analysis)이라 합니다. 단순 선형 회귀 분석을 수식으로 나타낸다면 $Y = Wx + b$ 의 형태가 되며, 다중 선형 회귀 분석을 수식으로 나타낸다면 $Y = W1x1 + W2x2 + \dots + b$ 의 형태가 됩니다.

Y는 결과로 종속 변수를 의미합니다.

X는 결과에 영향을 미치는 요소로 독립 변수를 의미합니다.

W는 독립 변수에 영향을 미치는 가중치(Weight)를 의미합니다.

b는 외부에서 영향을 미치는 값으로 편향성(Bias)을 의미합니다.

Tip : 독립 변수(X)는 예측 변수(Predictor Variable) 또는 원인 변수, 설명 변수(Explanatory Variable)라고도 부릅니다.

Tip : 종속 변수(Y)는 결과 변수(Outcome Variable) 또는 반응 변수, 목적 변수(Response Variable)라고도 부릅니다.

예를 들어, 나의 출근 시간을 예측하는 모델을 만든다 가정한다면 Y는 출근 시간, X는 출근 시간에 영향을 주는 요소입니다. W는 출근 시간에 영향을 주는 요소의 가중치를 의미합니다. 출근 시간에 영향을 주는 요소를 일어난 시간, 아침 식사 시간, 아침 출근 준비로 가정한다면 각각, X1, X2, X3로 볼 수 있습니다.

만약, 단순 선형 회귀 분석으로 출근 시간을 예측할 때 X은 일어난 시간이며, W은 해당 독립 변수의 영향력입니다. 다시 수식으로 표현한다면, 출근 시간 = 영향력 * 일어난 시간 + 편향

성으로 정의할 수 있습니다.

이번에는 다중 선형 회귀 분석으로 출근 시간을 예측한다면 출근 시간 = 영향력1 * 일어난 시간 + 영향력2 * 아침 식사 시간 + 영향력3 * 아침 출근 준비 시간 + 편향성이 됩니다.

Tip : 여기서 일어난 시간, 아침 식사 시간, 아침 출근 준비는 특성(feature)이 되며, 출근 시간은 목표(target)가 됩니다.

분류(Classification)

분류는 훈련 데이터(train data)에서 지정된 레이블(label)과의 관계를 분석해, 새로운 데이터의 레이블을 스스로 판별하는 방법입니다. 즉, 새로운 데이터를 대상으로 카테고리(category)를 스스로 판단합니다. 새로운 데이터를 대상으로 참인지 거짓인지 분류할 수 있다면, 이진 분류(Binary Classification)라 합니다. 만약, 새로운 데이터를 대상으로 두 개 이상의 카테고리를 나눠 분류 할 수 있다면, 다중 분류(Multi-label Classification)라 합니다.

예를 들어, 시험 성적으로 합격 여부를 판단한다면 합격(참)과 불합격(거짓)으로 구분할 수 있으므로 이진 분류가 됩니다. 다중 분류는 동물 이미지를 입력했을 때 강아지, 고양이, 새 등으로 분류하는 것을 의미합니다.

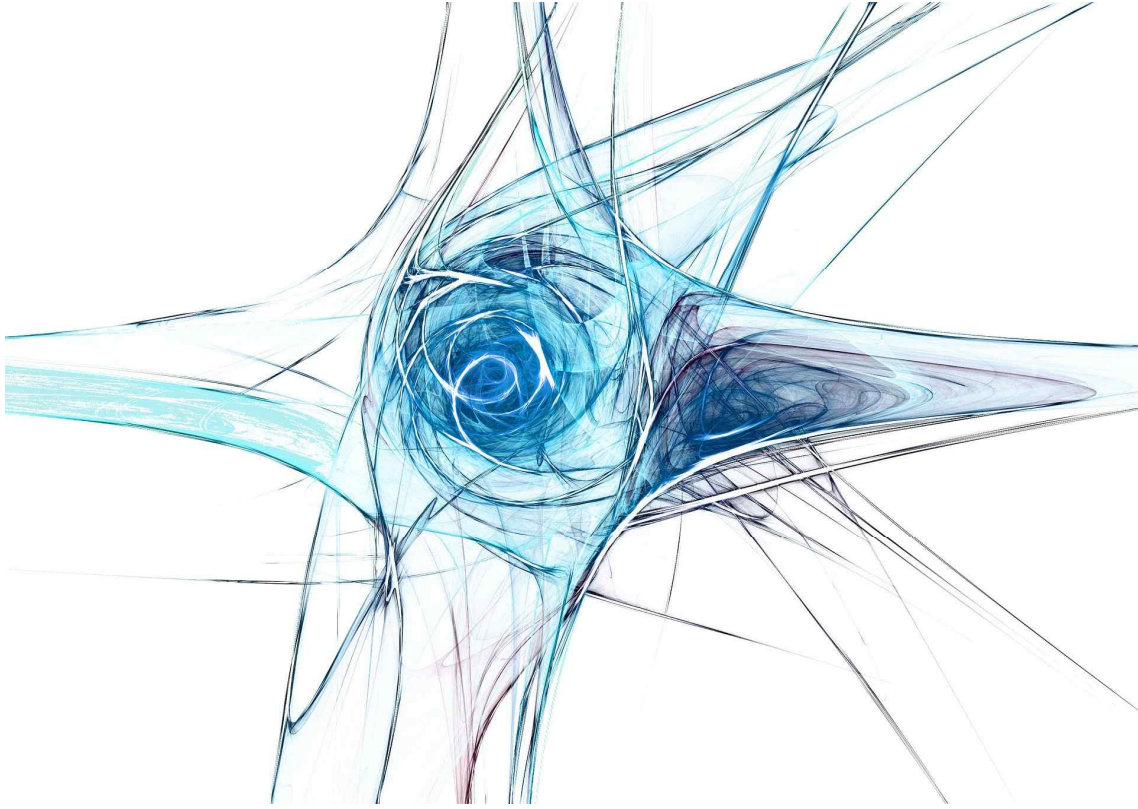
이진 분류의 대표적인 알고리즘은 로지스틱 회귀(Logistic Regression)가 있습니다. 분류에도 회귀가 사용될 수 있는데, 로지스틱 회귀는 종속변수에 로짓 변환을 실시해 편향성(bias)이 없는 타당한 계수를 추정할 수 있습니다. 간단하게 설명하자면 시험 성적을 그래프화 하여 합격 여부에 참과 거짓을 나타낸다면 극단적인 그래프가 그려질 수 있습니다. 이 그래프에서 특정 점수를 대입해 합격 여부를 확인할 수 있습니다.

다중 분류의 대표적인 알고리즘은 소프트맥스 회귀(Softmax Regression)가 있습니다. 이진 분류에서 참/거짓으로 분류할 때 참일 경우 1, 거짓일 경우 0으로 분류했습니다. 즉, 나온 결과값을 모두 합한다면 1이 됩니다. 이와 같이 다중 분류도 합이 1이 되는 확률 분포를 구성할 수 있습니다.

소프트맥스 회귀는 가중치를 정규화해 나온 결과값을 모두 더할 때 1이 되게끔 구성하는 것을 의미합니다. A, B, C를 분류하는 알고리즘을 만든다면 A는 0.1, B는 0.1, C는 0.8로 나오게 될 수 있습니다. 이는 A일 확률은 10%, B일 확률은 10%, C일 확률은 80%를 의미합니다. 이 중 가장 높은 확률인 C를 선택하게 됩니다. 이 확률 분포를 통해서 다중 분류를 진행할 수 있습니다.

Tip : 레이블의 범주를 클래스(class)로 부르며, Python의 Class와는 다른 의미입니다.

비지도 학습(Unsupervised Learning)



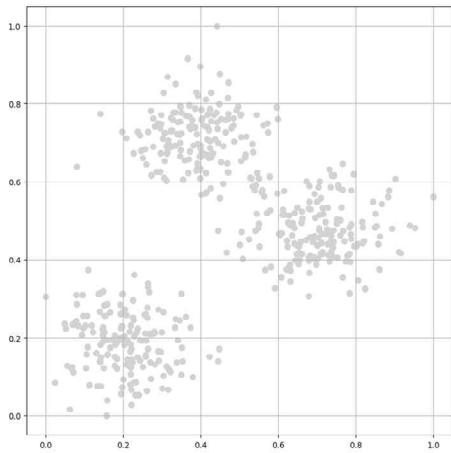
비지도 학습이란 레이블(Label)을 포함시키지 않고 데이터에 의해 컴퓨터(알고리즘)이 스스로 학습하는 방법입니다. 레이블이 존재하지 않기 때문에 특정한 패턴이나 규칙을 지정하여 모델을 생성합니다. 지도 학습에서는 훈련 데이터(train data)와 레이블(Label)이 각각 X와 Y의 역할을 했다면, 비지도 학습은 데이터(Data)로만 결과를 유추합니다. 즉, X를 일련의 규칙($f(x)$)을 통해 숨겨진 패턴을 찾는 것을 목표로 합니다.

훈련 데이터 없이 데이터를 대상으로 수행하므로, 목표값이 없어 지도 학습과 다르게 사전 학습을 필요로 하지 않습니다. 그러므로, 지도 학습과는 다르게 레이블이 없기 때문에 결과에 대한 성능평가가 어렵습니다. 비지도 학습에는 크게 군집화(Clustering), 이상치 탐지(Outlier Detection, Anomaly Detection), 차원 축소(Dimensionality Reduction) 등이 있습니다.

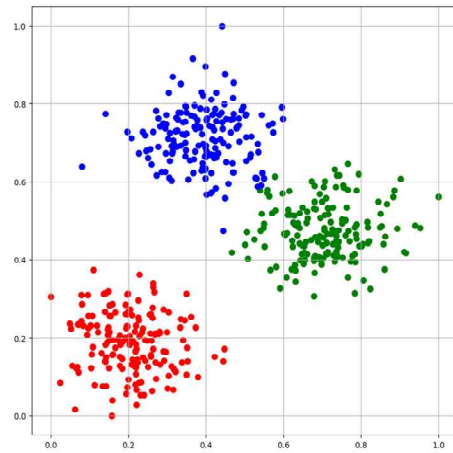
군집화(Clustering)

군집화란 입력 데이터를 기준으로 비슷한 데이터끼리 몇 개의 군집(cluster)으로 나누는 알고리즘입니다. 입력 데이터들의 특성을 고려하여 데이터를 분류하고, 같은 그룹으로 분류된 데이터끼리는 서로 비슷한 성질(위치, 평균, 편차 등)을 갖습니다. 반대로 서로 다른 그룹으로 분류된 데이터는 서로 다른 성질을 갖습니다.

위 좌측 이미지는 제공된 데이터(Data)이며, 우측 이미지는 사전 학습 없이 일련의 규칙을 통해 분류한 결과입니다. 데이터의 비슷한 성질로 세 종류의 군집을 분류하였습니다. 군집화는 일반적인 정의가 존재하지 않기 때문에, 알고리즘마다 서로 다른 군집을 분류할 수 있습니다.



Data

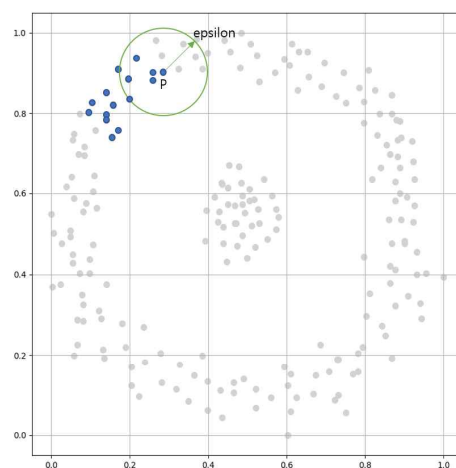
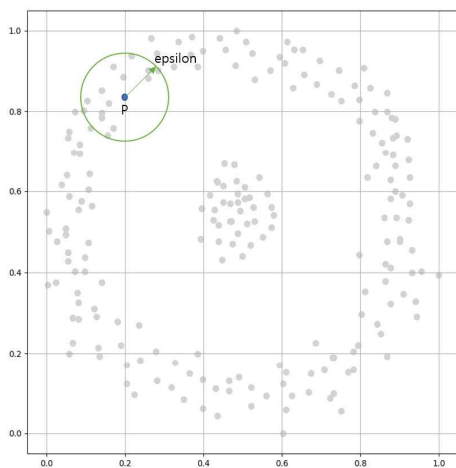


Clustering

예를 들어, K-평균 군집화(K-Means Clustering)는 임의의 중심점(Centroid)을 기준으로 최소 거리에 기반한 군집화를 진행합니다. 각각의 데이터는 가장 가까운 중심에 군집을 이루며, 같은 중심에 할당된 데이터는 하나의 군집군으로 형성됩니다.

여기서 K는 군집의 갯수를 의미하며, k가 3일 때는 3개의 군집군을 형성합니다. 데이터들의 군집 중심에서 가장 가까운 군집으로 뭉쳐지게 됩니다.

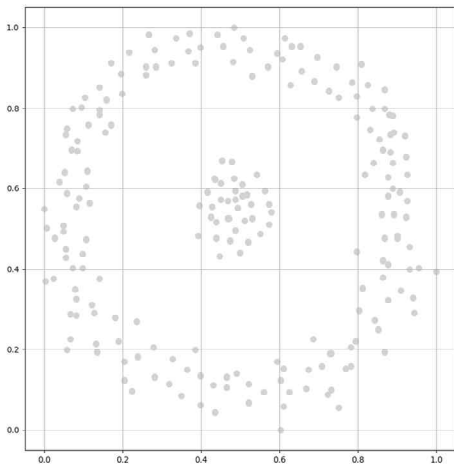
K-평균 군집화는 중심의 초깃값이 무작위로 정해지며, K의 갯수만큼 군집을 이루게 됩니다. 중심점과 군집 개수로 나누기 때문에, 군집의 크기, 밀도, 형태가 특이하거나 서로 다를 경우 좋지 않은 결과가 나타날 수 있습니다.



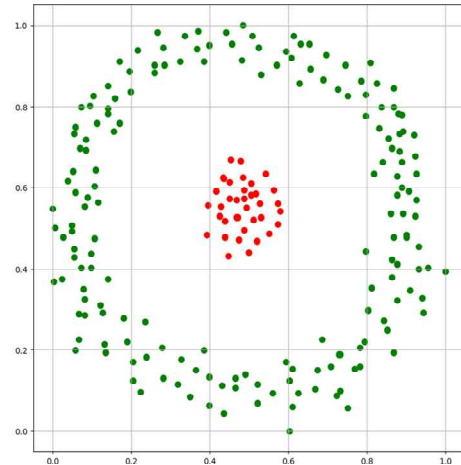
밀도 기반 군집화(Density-based spatial clustering of applications with noise, DBSCAN)는 K-평균 군집화 방식과는 다르게 군집화를 진행합니다. 밀도 기반 군집화는 특정 공간 내에 데이터가 많이 몰려있는 부분을 대상으로 군집화 하는 알고리즘 입니다.

즉, 임의의 P점(sample)을 기준으로 특정 거리(epsilon) 내에 점이 M(MinSamples)개 이상 있다면 하나의 군집으로 간주합니다. 이 지역(특정 거리 안쪽)을 ϵ -이웃(epsilon-neighborhood)이라 부릅니다.

ϵ -이웃 안에 샘플이 M개 이상이라면, 핵심 샘플(core sample) 또는 핵심 지점(core point)이라 부르며, 이 과정을 반복합니다. 만약, 핵심 샘플이 아닌 영역은 이상치로 간주합니다.

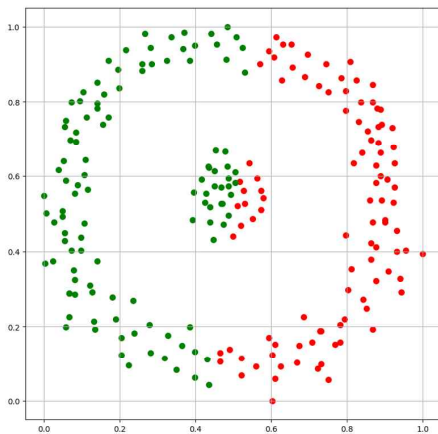


Data

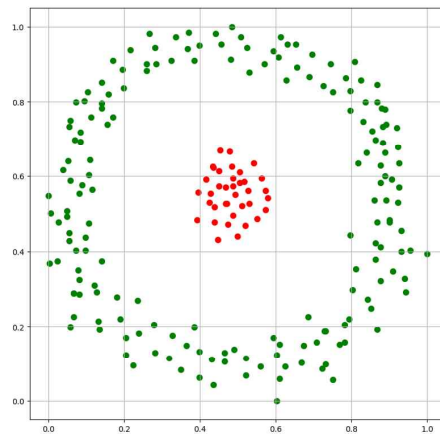


Clustering

밀도 기반 군집화는 특정 거리와 최소 샘플 갯수로 군집화를 이루므로, 사전에 군집 갯수를 설정하지 않아도 됩니다. 또한, 밀도를 기준으로 군집화하기 때문에 복잡한 분포의 데이터를 분석할 수 있으며, 어떤 군집에도 속하지 않는 이상치를 구분할 수 있습니다.



K-Means



DBSCAN

위 이미지는 같은 데이터를 K-평균 군집화와 밀도 기반 군집화로 군집화한 결과입니다. 군집화 알고리즘마다 군집화 하는 방식이 다르므로, 데이터의 특성과 수행할 알고리즘 또는 원하는 결과물에 따라 가장 효율적인 군집화 알고리즘을 선택해야 합니다.

군집화 알고리즘은 K-평균 군집화, 밀도 기반 군집화 이외에도 병합 군집화(Agglomerative Clustering), 평균 이동 군집화(Mean-Shift Clustering), 계층적 군집화(Hierarchical Clustering), 스펙트럼 군집화(Spectral Clustering) 등이 있습니다.

이상치 탐지(Outlier Detection, Anomaly Detection)

이상치 탐지(Outlier Detection, Anomaly Detection)는 어떤 군집에도 포함되지 않는 샘플을 의미합니다. 학습 데이터(train data)나 입력 데이터(input data)에 비정상적인 값을 갖는 데이터가 있다면, 이를 이상 데이터라 부릅니다.

정제되지 않은 빅데이터는 정상적인 데이터와 정상적이지 않은 데이터가 포함되어 있을 확률이 매우 높습니다. 정상적이지 않은 데이터를 이상 데이터로 간주합니다.

만약, 이상 데이터가 많다면 학습 모델이나 알고리즘에 정확도와 신뢰도를 낮추게 됩니다. 이상치 탐지는 결함이 있는 데이터나 제품을 찾거나, 시계열 데이터(Time Series data)에서 일반적인 패턴을 벗어난 패턴 등을 찾습니다.

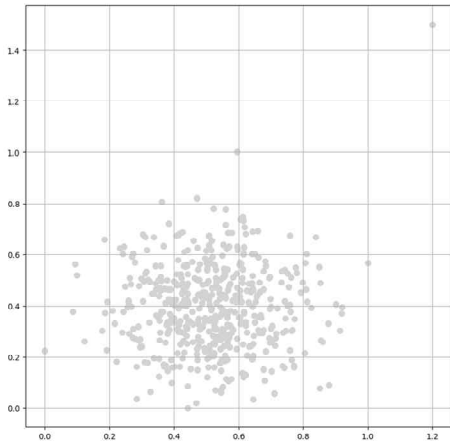
이상치 탐지는 크게 이상치(Outlier)와 이상(Anomaly) 탐지가 있습니다.

이상치(Outlier)는 횡단면 데이터(Cross-sectional data)에서 비정상적인 데이터를 찾는 것을 의미합니다. 이상(Anomaly)는 시계열 데이터에서 비정상적인 데이터를 찾는 것을 의미합니다.

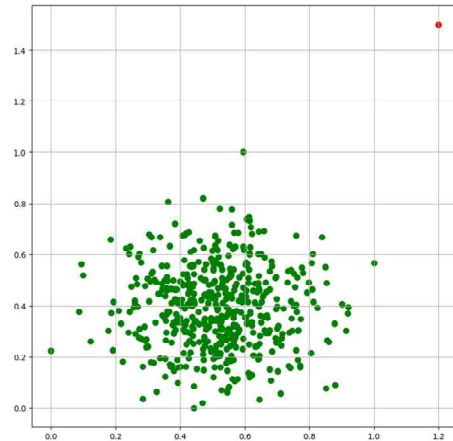
여기서, 일반적인 데이터는 정상치(Inlier)로 부르며 비정상적인 데이터는 이상치(Outlier)로 부릅니다.

Tip : 시계열 데이터란 일정 시간을 간격으로 배치된 데이터를 의미합니다.

Tip : 횡단면 데이터란 동일한 시간, 동일 기간에 여러 변수에 대하여 수집된 데이터를 의미합니다.



Data



Outlier

이상치는 위의 이미지에서 붉은색 지점이 이상치로 볼 수 있습니다. 주요한 군집에 포함되지 않으며, 갑자기 데이터가 크게 달라지는 부분을 확인할 수 있습니다.

이런 부분은 특정 알고리즘을 진행하기 전에, 사전에 제거하거나 별도의 분류를 진행해야 합니다. 만약, 이상치를 제거하지 않고 군집화 등을 진행한다면 의도하지 않은 결과가 나타날 수 있습니다.

이상치 탐지 알고리즘은 아이솔레이션 포레스트(Isolation Forest), LOF(Local Outlier Factors), One-class SVM 등이 있습니다.

차원 축소(Dimensionality Reduction)

차원 축소란 다차원의 데이터의 차원을 축소하여 저차원의 새로운 데이터로 변경해 생성하는 것을 의미합니다. 여러 특성과 변수 간에 존재하는 상관관계를 이용하여 주요 구성요소에 대한 분석을 진행합니다.

데이터에 특성(feature)이 매우 많다면 학습 모델을 구성하는 데 오랜 시간이 소요되며, 정확도와 신뢰도를 보장할 수 없습니다. 다차원의 데이터의 특성간에 상관관계가 서로 높다면, 하나의 특성만을 사용하거나 여러 특성을 조합하여 하나의 특성으로 변경할 수 있습니다.

만약, 3차원의 데이터를 2차원으로 축소한다고 가정하면 연산량의 감소, 노이즈 및 이상치 제거, 시각화 등에 이점을 볼 수 있습니다.

차원 축소에는 크게 특성 선택(Feature Selection)과 특성 추출(Feature Extraction)이 있습니다. 특성 선택과 특성 추출은 앞서 설명한 변경법을 의미합니다.

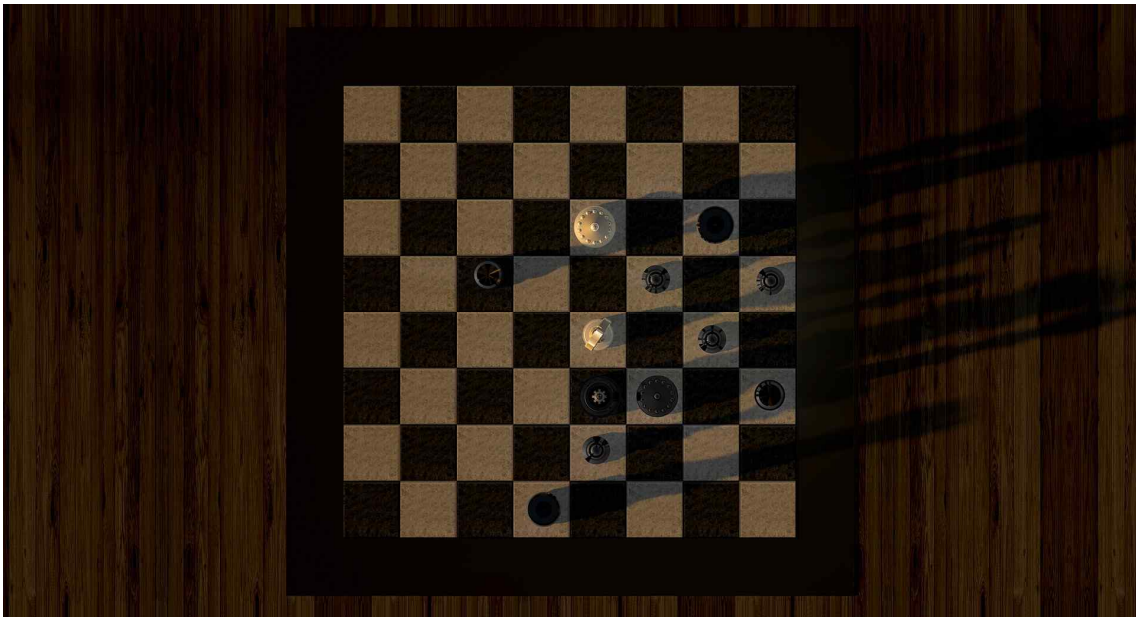
특성 선택은 특정 특성에 종속적인 성향이 강해, 사용하지 않아도 되는 특성을 의미합니다.

예를 들어, 점수(score)와 등급(rank)에 관한 특성이 있다면 등급은 점수에 종속적인 데이터일 가능성이 높습니다. 그렇다면, 둘 중 하나의 특성을 제거하여 차원을 축소할 수 있습니다.

특성 추출은 점수와 등급에 관한 특성을 하나로 압축하여 새로운 특성을 만들어 내는 것입니다. 예를 들어, 점수와 등급을 다시 산정해 기존 A, B, C 분류 방식에서 A+, A, B+, B, C+, C 등으로 새로운 특성을 만들어낼 수 있습니다.

차원 축소 알고리즘은 주성분 분석(Principal Component Analysis, PCA), 특잇값 분해(Singular Value Decomposition, SVD), 음수 미포함 행렬 분해(Non-negative Matrix Factorization, NMF) 등이 있습니다.

강화 학습(Reinforcement Learning)



강화 학습(Reinforcement Learning)은 머신 러닝(Machine Learning)의 분야 중 하나로, 행동주의(Behaviorism) 심리학 이론을 토대로 구현된 알고리즘입니다.

행동주의는 모든 동물은 학습 능력을 가지고 있으므로, 어떤 행동을 수행했을 때 보상(reinforcement)이 있다면 보상을 받았던 행동의 발생 빈도가 높아진다는 이론입니다.

이러한 이론을 토대로 보상을 최적화하기 위한 강화 학습이 생겨났습니다.

강화 학습은 크게, 환경(Environment), 에이전트(Agent), 상태(State), 행동(Action), 보상(Reward), 정책(Policy) 등으로 구성되어 있습니다.

환경(Environment)이란 학습을 진행하는 공간 또는 배경을 의미합니다. 예를 들어, 바둑에서의 환경은 바둑판이며, 게임에서의 환경은 게임 속 세상을 의미합니다.

에이전트(Agent)는 환경과 상호작용하는 프로그램을 의미합니다. 즉, 플레이어(Player)나 관측자(Observer)를 지칭합니다.

상태(State)란 환경에서 에이전트의 상황을 의미합니다. 바둑에서 매 턴 둘의 상태나 게임의 각각의 프레임일수도 있습니다.

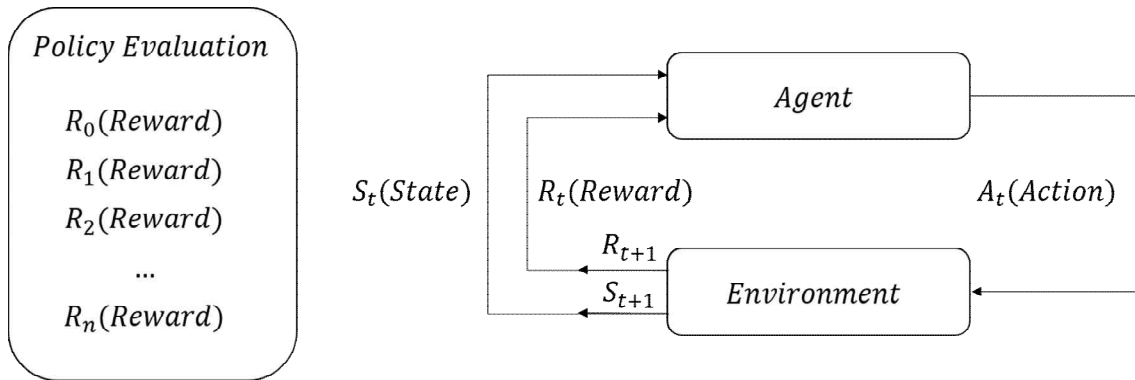
행동(Action)이란 주어진 환경의 상태에서 에이전트가 취하는 행동을 의미합니다. 돌을 놓거나 움직이는 등의 모든 행위를 지칭합니다.

보상(Reward)은 현재 환경의 상태에서 에이전트가 어떠한 행동을 취했을 때, 제공되는 보상을 의미합니다. 양(Positive)의 보상, 음(Negative)의 보상, 0의 보상 등을 돌려받습니다. 양의 보상이 주어진다면, 그 환경의 상태에서 에이전트는 해당 행동을 더 많이 취할 가능성이 높아집니다. 반대로 음의 보상에서는 해당 행동의 발생 빈도가 낮아집니다.

마지막으로 정책(Policy)은 에이전트가 보상을 최대화하기 위해 행동하는 알고리즘을 의미합니다. 즉, 에이전트는 반복되는 학습을 통해 보상을 최대화하는 행동을 취하게 됩니다.

Tip : 심리학에서 reinforcement는 생물이 어떤 자극에 반응해 미래의 행동을 바꾸는 것을 의미합니다.

마르코프 결정 과정(Markov Decision Process, MDP)



강화 학습을 간단한 순서도로 변경한다면 위와 같은 이미지로 표현할 수 있습니다.

위의 순서도는 마르코프 결정 과정에서 행동이라는 요소가 추가된 형태입니다.

에이전트는 S_t 상태에서 A_t 의 행동을 수행합니다. 그러면 환경은 다음 번 상태의 S_{t+1} 과 다음 번 보상인 R_{t+1} 을 에이전트에게 전달합니다.

강화 학습에서 중요한 가정 중 하나는 환경이 마르코프 결정 과정의 마르코프 속성(Markov property)을 가진다는 것 입니다.

마르코프 속성은 과거 상태(S_1, S_2, \dots, S_{t-1})들과 현재 상태(S_t)가 주어졌을 때, 미래 상태(S_{t+1})는 과거 상태보다 현재 상태에 의해서 결정된다는 것을 의미합니다. 즉, 과거 상태와는 별개로 현재 상태에 의해서만 결정된다는 의미입니다.

지속적으로 상태가 변화하게 되는데, 어떤 상태에서 다음 상태로 변화하는 것을 전이(transition)라 합니다.

결국, t시점의 상태 S_t 에서 행동(A_t)을 할 때 수행하는 다음 상태의 S_{t+1} 을 결정하게 됩니다.

이를 상태 전이(state transition)라 하며, 상태 전이에서 받는 보상은 $R(S_{t+1})$ 로 표현할 수 있습니다.

가치 함수(Value Function)

가치 함수(Value Function)란 어떤 상태 S_t 에서 정책에 따라 행동을 할 때 얻게되는 기대 보상(Expected Reward)을 의미합니다.

즉, 상태와 행동에 따라 최종적으로 어떤 보상을 제공해줄지에 대한 예측 함수입니다.

또한, 상태-행동 가치 함수(State-Action Value Function)도 존재하는데 이는 어떤 상태 S_t 에서 행동을 한 다음, 정책에 따라 행동을 할 때 얻게되는 기대 보상을 의미합니다.

이 가치 함수에 따라 학습이 진행되게 됩니다. 최적의 가치 함수를 구현한다면, 효율적인 정책을 구성할 수 있습니다.

가치 함수는 벨만 최적 방정식(Bellman Optimality Equation)을 적용하며, 동적계획법(Dynamic Programming), 몬테 카를로 방법(Monte Carlo Method), 모수적 함수(parameterized function) 등을 사용할 수 있습니다.

결국, 강화 학습은 에이전트의 시행착오(trial and error)를 통해 보상을 최대화 할 수 있는 정책을 찾는 방법으로 학습이 진행되게 됩니다.

강화 학습은 크게 모델 기반의 강화 학습(Model-based Reinforcement Learning)과 모델이 없는 강화 학습(Model-free Reinforcement Learning)이 있습니다.

모델 기반(Model-based) & 모델 프리(Model-free)

모델 기반의 강화 학습과 모델이 없는 강화 학습의 이름에서 알 수 있듯이, 모델을 사용하는 여부에 따라 나뉩니다.

강화 학습에서의 모델(Model)은 데이터와 결괏값에 대한 규칙($f(x)$)이 아닌, 환경에 대한 가정

을 모델로 간주합니다.

즉, 에이전트가 환경의 모델에 상태 전이와 보상을 예측하게 됩니다.

모델 기반의 강화 학습은 에이전트는 어떠한 행동을 할 때, 이미 환경이 어떻게 바뀔지 알 수 있습니다.

그러므로, 에이전트가 행동하기 전에 환경의 변화를 예상하여 최적의 행동을 실행할 수 있습니다.

모델 기반의 강화 학습은 적은 양의 데이터로도 효율적인 학습을 할 수 있지만, 모델이 정확한 환경을 구현하지 않는다면 올바른 학습을 진행할 수 없습니다.

모델을 사용하지 않는 강화 학습은 모델 기반의 강화학습과 정반대의 장/단점을 갖습니다. 즉, 모델을 구현하기 어려운 상황에도 사용할 수 있다는 장점이 있습니다.