

25p

알고리즘은 어떠한 문제를 해결하기 위해 정해진 일련의 절차나 방법을 이야기한다. 이런 알고리즘을 컴퓨터에 전달하여 문제를 해결하기 위한 방법을 제시하고 이를 컴퓨터가 학습하는 것이다.

이러한 알고리즘의 한 종류에는 분류(**Classification**) 알고리즘이 있다. 분류 알고리즘의 예로 사진 속에서 얼굴을 보고 남자인지 여자인지 분류하는 것과 손으로 쓴 숫자를 인식하여 각각의 숫자를 분류하는 것, 이메일에서 스팸과 일반(정상) 메일을 분류하는 것 등을 예로 들 수 있다.

26p

결론적으로 기계학습은 데이터를 이용하여 문제를 해결하는 절차인 알고리즘을 기술하고 이 알고리즘을 토대로 컴퓨터에게 학습을 시키는 것이다.

26p

기계 학습의 종류는 크게 지도학습(**Supervised Learning**)과 비지도 학습(**Unsupervised Learning**)으로 나뉘어진다.

지도학습

지도학습(**Supervised Learning**)은 다른 말로 감독 학습이라고도 부르는데 답안이 달린 시험 정보를 주고 학습시킨다는 개념이다.

여러 장의 고양이 사진과 개 사진을 주고 개와 고양이 사진을 구분하라고 시키는 것이다. 이렇게 학습하도록 제공되는 데이터를 훈련 데이터(**Train Set**)라고 하고 이 훈련 데이터에 어떤 사진이 고양이 사진이고 어떤 사진이 개 사진인지 직접 답을 달아주는 것을 레이블링(**Labeling**)이라고 한다. 이렇게 레이블링 된 훈련 데이터는 신경망 등 다양한 알고리즘을 통해 이 사진들에서 사진의 중요한 정보인 픽셀을 바탕으로 개와 고양이 모양의 정보를 추출하고 이를 개와 고양이를 구분할 수 있는 모델로 만들어 준다. 이 모델을 통해 앞으로 레이블링

27p

이 되지 않은 개나 고양이 사진 데이터인 테스트 데이터가 들어오면 고양이 사진인지 개 사진인지 판단할 수 있게 된다. 이와 같이 지도학습은 특정 입력에 대하여 학습된 모델에서 입력을 비교하여 정답을 제시하는 학습이라고 할 수 있다.

- 비지도 학습에 비해서 지도 학습이 난이도가 낮은 편이고, 비지도 학습은 최종적인 답 보다는 인간의 의사결정을 위한 정보 모음을 주는 쪽이기 때문에 지도 학습을 할 예정이다.
- 생각해둔 학습량은 사물 2종류, 이미지양은 대략 500장 내외 정도.
- 픽셀로 구분해서 학습하는 것이니 최대한 화질이 좋은 사진을 넣어서 학습시키는 것이 좋겠지만, 장비가 그렇게 좋지 않으니 대략 700px 내외로 넣는 게 좋을 것 같다.
- 여러 케이스를 학습시키고, 변화를 줄 때 사물 자체의 바리에이션 외에도 각도, 명도, 채도 등에서도 여러 변화를 줘서 학습시키면 좋을 것 같다.
- 확장, 혼합 현실 아래 단계로 생각하고 출발하는 거기 때문에 주변 환경에서 장애물이 되는 물체(의자, 창문, 책상 등등)로 선정하도록 하자.
- 시간이 많지 않기도 해서 처음에는 모호하지 않고 명확한 사진 위주로 학습시키는 게 좋을 것 같다.

주어진 데이터를 기반으로 모델을 생성하여 새로운 값이 들어왔을 때 결과 값을 예측하는 학습 방법이 지도 학습 방법이다.

책에는 선형회귀 방식의 예제와 클러스터링 예제가 있다.

이미지 판별을 위한 예제로는

[https://colab.research.google.com/drive/1Lv9rchnTm8diOwO8182ccoqdPWZGhww6?usp=sharing#scrollTo=Z\\_Dp1RQz9taT](https://colab.research.google.com/drive/1Lv9rchnTm8diOwO8182ccoqdPWZGhww6?usp=sharing#scrollTo=Z_Dp1RQz9taT)

를 참고할 것.

tensorflow 패키지 설치 필수(cmd > pip install tensorflow)

약 460MB

32p

비지도 학습

비지도 학습(Unsupervised Learning)은 레이블링이 되어 있지 않는 데이터를 주고 학습시키는 것이다. 학습된 모델을 이용하여 정답을 맞히는 목적으로 학습하는 것이 아닌 어떤 데이터들이 서로 비슷한지 그룹지어주거나, 어떤 성질이 데이터를 잘 정의하는 지를 판단하는 등에 대해 유용한 정보를 제공해주는 방법이다.

지도 학습에서 살펴본 예제를 비지도 학습에서 적용해보면 어느 사진이 개인지 고양이인지 알려주지 않고 군집 분석 알고리즘으로 사진 데이터를 학습하고 고양이처럼 생긴 사진끼리 한 그룹으로, 개처럼 생긴 사진끼리 또 다른 그룹으로 구분해 준다. 이렇게 구분만 하고 이 그룹이 어떤 그룹인지 알려주지 않고 최종 판단은 사람이 하도록 하는 것이 비지도 학습이다. 고양이 사진과 개 사진이 서로 다른 그룹으로 구분은 잘 해냈으므로 사람은 이 결과를 보고 한 그룹은 고양이 사진, 다른 그룹은 개 사진이라고 판단할 수 있는

근거로 사용하게 된다. 이런 그룹화하는 문제를 비지도 학습에서 클러스터링(Clustering)이라고 한다.

33p

이렇게 비지도 학습은 최종적인 답을 알려주는 용도로는 사용하지 않고 사람들에게 의사결정을 위해 유용한 정보를 주는 학습 방법이며 지도 학습과 마찬가지로 널리 사용되고 있다.

- 지도/비지도 학습은 학습 방법 차이 뿐만 아니라 결과물에도 차이가 있어서 만들어진 인공지능의 사용에도 차이가 나게 된다.
- 즉, 학습 방법의 차이만 아니라 결과물도 방식이 달라서 지도 학습을 선택한 것.
- 강화 학습(Reinforcement Learning)도 있지만, 이번 프로젝트에서는 사용 예정 없음.

36p

파이썬 패키지

-**sklearn**(사이킷런)은 파이썬에서 머신러닝 분석을 할 때 유용하게 사용할 수 있는 라이브러리이다. 여러가지 머신러닝 모듈로 구성되어 있다. **from sklearn import**

-**linear\_model**은 사이킷런에서 선형 모델을 활용하게 만들어 준다.

-**numpy**는 파이썬 언어를 위한 행렬, 벡터 등의 수학 계산을 위한 자료구조와 계산 함수를 제공하는 패키지이다.

-**matplotlib**는 그래프 중 플롯 그래프를 그릴 때 주로 쓰이는 2D, 3D 플롯팅 패키지이다.

-**matplotlib.pyplot**은 **matplotlib**의 서브 패키지이다.

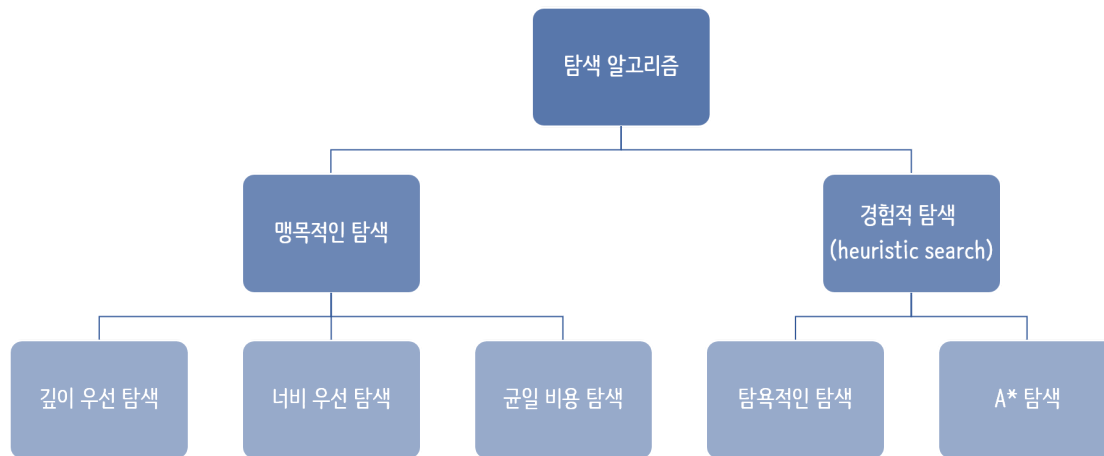
50~51p

탐색

탐색은 어떤 문제를 찾을 때 이를 해결하기 위한 답이 될 수 있는 최적의 해(답)을 찾기 위한 것이다. 최적의 해를 얼마나 잘 찾느냐는 어떠한 순서로 진행을 해야하는지 여러 방법을 통해 가능한 다양한 시도를 해보며 그 안에서 최소의 노력으로 원하는 해를 찾는 것이다.

일반적으로 어떤 문제를 해결하는 방법은 모든 경우의 수를 생각해야 해서 이를 컴퓨터를 이용하여 해결하려면 엄청 복잡하고 많은 메모리를 차지하게 된다. 이렇게 되면 아무리

좋은 방법이라도 효율적이지 못하게 된다. 이를 좀 더 효율적이면서도 좋은 최적의 해를 찾기 위해 우리는 탐색이 필요한 것이다.



▲ 탐색기법

[ 인공지능에서의 탐색 ]

출처: <https://hyeonjiwon.github.io/artificial%20intelligence/AI-1/>

52~53p

맹목적 탐색

가장 단순한 방법으로 **Uninformed Search** 또는 **Blind Search Method**로도 불리는데 이를 해석해보면 정보 없이 무작위적으로 탐색하는 방법이다. 문제를 해결하기 위한 어떤 특성이나 방안 등을 고려하지 않고 아무런 정보도 없이 미리 정해진 순서에 따라 해를 찾는 방법이다.

모든 경우의 수를 고려한다고 했을 때 이를 전부 다 표현하기가 복잡하고 계산 시간 및 메모리에 제한적인 문제가 발생할 수 있다. 따라서 이를 좀 더 효과적으로 한눈에 알아볼 수 있도록 인공지능에서는 탐색의 과정을 형식화해주는 추상화 과정이 필요했고 이를 시각화하기 위해 그래프를 이용하게 되었다.

그래프란 복잡한 문제를 정점과 간선으로 모형화하여 표현한 것으로 각 정점은 여러 개의 다른 정점과 간선을 통해 단방향 또는 양방향으로 연결할 수 있다.

그래프를 이용하여 각 상태의 경로를 한눈에 파악할 수 있다.

53~54p

경험적 탐색

경험적 탐색은 인공지능에서 가장 많이 사용하는 방법으로 탐색 시간을 줄이기 위해 비슷한 문제에 대한 이전 경험이나 지식을 바탕으로 어떤 기준을 결정하는 방식이다. 이때 경험이나 지식을 바탕으로 만들어진 정보를 휴리스틱 정보라고 하는데 이를 이용한 탐색이라고 해서 휴리스틱 탐색(**Heuristic Search**)이라고도 부른다.

어떤 문제를 쫓을 때 기존에 이와 비슷한 문제가 있었는지, 이를 어떻게 해결했는지 등의 경험했던 정보나 지식을 이용하여 직관적으로 판단하고 선택하는 의사결정 방식을 통해 해를 얻는다. 따라서 논리적이거나 최적의 해는 아니더라도 근사치의 값을 이용해 탐색을 빠르게 할 수 있도록 해주는 방법이다.

경험적 탐색은 특히 모호하여 문제의 해가 정확하게 나오지 않을 때 또는 정해진 시간 안에 모든 탐색을 하지 못하는 경우 아주 유용하다. 왜냐하면 이전 경험에 의한 지식이나 정보를 바탕으로 근사치의 값을 제시하기 때문이다.