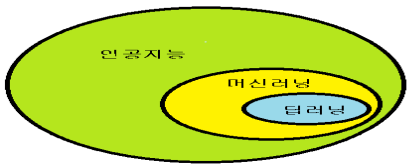
인공 지능(Artificial Intelligence)

인공 지능은 1950년대에 초기 컴퓨터 과학 분야의 일부 선각자들이 "컴퓨터가 '생각'할 수 있는가?"라는 질문을 하면서 시작되었다. 이 질문의 답은 오늘날에도 여전히 찾고 있다. 이 분야에 대한 간결한 정의는 다음과 같다. **보통의 사람이 수행하는 지능적인 작업을 자동화하기 위한 연구 활동**이다. 이처럼 AI는 머신 러닝과 딥러닝을 포괄하는 종합적인 분야이다. 또 학습 과정이 전혀 없는 다른 방법도 많이 포함하고 있다. 예를 들어 초기 체스 프로그램은 프로그래머가 만든 하드코딩된 규칙만 가지고 있었고 머신 러닝으로 인정받지 못했다. 아주 오랜 기간 동안 많은 전문가는 프로그래머들이 명시적인 규칙을 충분하게 많이 만들어 지식을 다루면 인간 수준의 인공 지능을 만들 수 있다고 믿었다. 이런 접근 방법을 심볼릭 AI(symbolic AI)라고 하며 1950년대부터 1980년대까지 AI 분야의 지배적인 패러다임이었다. 1980년대 전문가 시스템(expert system)의 호황으로 그 인기가 절정에 다다랐다.

심볼릭 AI가 체스 게임처럼 잘 정의된 논리적인 문제를 푸는 데 적합하다는 것이 증명되었지만, 이미지 분류, 음성 인식, 언어 번역 같은 더 복잡하고 불분명한 문제를 해결하기 위한 명확한 규칙을 찾는 것은 아주 어려운 일이다. 이런 심볼릭 AI를 대체하기 위한 새로운 방법이 등장했는데, 바로 **머신 러닝**(machine learning)이다.

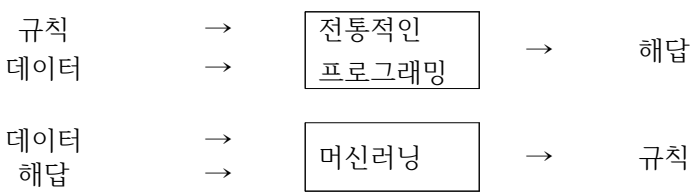
딥러닝(deep learning)은 머신 러닝의 특정한 한 분야로서 연속된 **층**(layer)에서 점진적으로 의미 있는 표현을 배우는 데 강점이 있으며, 데이터로부터 표현을 학습하는 새로운 방식이다. 딥러닝의 딥(deep)이란 연속된 층으로 표현을 학습한다는 개념을 나타낸다. 데이터로부터 모델을 만드는 데 얼마나 많은 층을 사용했는지가 그 모델의 **깊이**가 된다.



머신 러닝(machine learning)

머신러닝은 이런 질문에서부터 시작된다. "우리가 어떤 것을 작동시키기 위해 '어떻게 명령할지 알고 있는 것' 이상을 컴퓨터가 처리하는 것이 가능한가? 그리고 특정 작업을 수행하는 법을 스스로 학습할 수 있는가? 컴퓨터가 우리를 놀라게 할 수 있을까? 프로그래머가 직접 만든 데이터 처리 규칙 대신 컴퓨터가 데이터를 보고 자동으로 이런 규칙을 학습할 수 있을까?"

이 질문은 새로운 프로그래밍 패러다임의 장을 열었다. 전통적인 프로그래밍인 심볼릭 AI의 패러다임에서는 규칙(프로그램)과 이 규칙에 따라 처리될 데이터를 입력하면 해답이 출력된다(그림 1-2 참고). 머신 러닝에서는 데이터와 이 데이터로부터 기대되는 해답을 입력하면 규칙이 출력된다. 이 규칙을 새로운 데이터에 적용하여 창의적인 답을 만들 수 있다.



<그림 1-2> 머신러닝: 새로운 프로그래밍 패러다임

머신 러닝 시스템은 명시적으로 프로그램되는 것이 아니라 **훈련**(training)된다. 작업과 관련 있는 많은 샘플을 제공하면 이 데이터에서 통계적 구조를 찾아 그 작업을 자동화하기 위한 규칙을 만들어 낸다. 예를 들어 여행 사진을 태깅하는 일을 자동화하고 싶다면, 사람이 이미 태그해 놓은 다수의 사진 샘플을 시스템에 제공해서 특정 사진에 태그를 연관시키기 위한 통계적 규칙을 학습할 수 있을 것이다.
머신 러닝은 1990년대 들어와서야 각광을 받기 시작했지만, 고성능 하드웨어와 대량의 데이터셋이 가능해지면서 금방 AI에서 가장 인기 있고 성공적인 분야가 되었다. 머신 러닝은 수리통계와 밀접하게 관련되어 있지만 통계와 다른 점이 몇 가지 있다. 먼저 머신 러닝은 통계와 달리 보통 대량의 복잡한 데이터셋(예를 들어 몇 만 개의 픽셀로 구성된 이미지가 수백만 개가 있는 데이터셋)을 다루기 때문에 베이지안 분석(Bayesian analysis) 같은 전통적인 통계 분석 방법은 현실적으로 적용하기 힘들다. 이런 이유 때문에 머신러닝, 특히 딥러닝은 수학적 이론이 비교적 부족하고 엔지니어링 지향적이다. 이런 실천적인 접근 방식 때문에 이론보다는 경험을 바탕으로 아이디어가 증명되는 경우가 많다.

AI에 대한 전망

AI에 대한 단기간의 기대는 비현실적일지도 모르지만, 장기적인 전망은 매우 밝다. 의료 진단에서부터 디지털 비서까지 확실히 이전과는 다른 여러 중요한 문제에 AI를 적용하기 시작했다. AI 역사상 유례를 찾아볼 수 없는 수준의 투자에 크게 힘입어 AI 연구는 지난 5년간 놀라울 정도로 매우 빠르게 발전해 왔다. 하지만 이런 발전 중에서 비교적 아주 일부만이 현실 세계의 제품과 프로세스에 적용되었다. 딥러닝 연구 성과의 대부분은 아직 적용되지 않았거나, 적어도 전체 산업계를 통틀어서 딥러닝이 풀 수 있는 다양한 종류의 문제에는 적용되지 않았다. 일반 의사들은 아직 AI를 사용하지 않고, 회계사들도 마찬가지이다. 아마 여러분도 일상생활에서 AI 기술을 사용하지 않고 있을 것이다. 물론 스마트폰에 간단한 질문을 해서 그럴싸한 대답을 얻거나 Amazon.com에서 유용한 상품 추천을 받고, Google Photos에서 '생일'을 검색해서 지난달의 딸아이 생일 파티 사진을 바로 찾을 수 있다. 이런 기술은 이전에 비해 많이 발전되었다. 하지만 이런 도구는 여전히 우리 일상생활의 액세서리일 뿐이다. AI는 우리가 일하고 생각하고 생활하는 것의 중심에 들어오지 않았다.
AI가 아직 폭넓게 적용되지 못했기 때문에 지금 당장은 AI가 이 세상에 큰 영향을 줄 수 있으리라고 믿기 힘들지도 모릅니다. 비슷하게 1995년으로 돌아가 보면, 그때는 인터넷이 미래에 미칠 영향을 믿기 힘들었을 것이다. 그 당시에 대부분의 사람들은 인터넷이 자신과 어떻게 연관이 있을지, 우리의 일상생활을 어떻게 바꿀지 이해하지 못했다. 오늘날 딥러닝과 AI도 동일한다. 그러므로 실수를 범하지 말아야 한다. 결국 AI의 시대는 도래할 것이다. 그리 멀지 않은 미래에 AI가 우리의 비서가 되고, 심지어 친구가 될 것이다. 우리의 질문에 대답하고 아이의 교육을 도와주고 건강을 보살펴 줄 것이다. 식료품을 문 앞에 배달해 주고 A부터 B 지점까지 차를 운전해 줄 것이다. 점점 더 복잡해지고 정보가 넘쳐 나는 세상에 대한 인터페이스(interface)가 될 것이다. 더욱 중요한 것은 AI가 유전학에서부터 수학까지 모든 분야의 과학자들을 도와 새롭고 놀라운 발견을 이루어 냄으로써 인류 전체를 발전시킬 것이란 점이다.
이 와중에 몇 번의 난관을 만날 수 있고 새로운 AI 겨울이 올 수도 있다. 마치 인터넷 업체가 1998~1999년 사이에 매우 과열되었다가 2000년대 초에 몰락하면서 투자가 멈추어 고통을 받았던 것과 같다. 하지만 결국 AI 시대는 올 것이다. 오늘날의 인터넷처럼 우리 사회와 일상생활을 구성하는 거의 모든 과정에 AI가 적용될 것이다.

단기간의 과대 선전은 믿지 말고 장기 비전을 믿어야 한다. AI가 아직 아무도 감히 생각하지도 못했던 완전한 모습으로 진정한 잠재성을 발휘하려면 어느 정도의 시간이 걸릴지 아무도 모른다. 하지만 AI의 시대는 올 것이고 이 세상을 환상적인 방식으로 변모시킬 것이다.

머신러닝의 간략한 역사

딥러닝은 AI 역사에서 찾을 수 없을 만큼 대중에게 많은 관심과 업계의 투자를 받고 있다. 하지만 이것이 머신 러닝의 첫 번째 성공은 아니다. 오늘날 산업계에서 사용하는 대부분의 머신 러닝 알고리즘은 딥러닝 알고리즘이 아니다. 또 딥러닝이 모든 작업에 맞는 만능 도구는 아니다. 때로는 딥러닝을 적용하기에 데이터가 충분하지 않거나 다른 알고리즘이 문제를 더 잘 해결할 수도 있다.
다음의 1~4에서 전통적인 머신 러닝 방법을 간단하게 소개하고 지금까지의 역사적 배경을 설명한다..

1.확률적 모델링(probabilistic modeling)

확률적 모델링은 통계학 이론을 데이터 분석에 응용한 것이다. 초창기 머신 러닝 형태 중 하나고 요즘도 널리 사용된다. 가장 잘 알려진 알고리즘 중 하나는 **나이브 베이즈**(Naive Bayes) 알고리즘이다.
나이브 베이즈는 입력 데이터의 특성이 모두 독립적이라고 가정하고 베이즈 정리(Bayes' theorem)를 적용하는 머신 러닝 분류 알고리즘이다. (강한 또는 '순진한'(naive) 가정이다. 여기에서 이름이 유래되었다.) 이런 형태의 데이터 분석은 컴퓨터보다 앞서 있었기 때문에 첫 번째 컴퓨터가 등장하기 수십년 전에는 수작업으로 적용했다(거의 1950년대로 거슬러 올라간다). 베이즈 정리와 통계학의 토대는 18세기까지 거슬러 올라간다.
*베이즈 정리(Bayes' Theorem)
$P(Y_1),\cdots,P(Y_m),P(X|Y_1),\cdots,P(X|Y_m)$ given (표본공간)$S=\cup_{i=1}^{m} Y_i$ and Y_i's 상호배타적).

$$\Rightarrow P(Y_k|X)=\frac{P(Y_k)P(X|Y_k)}{\sum_{i=1}^{m}P(Y_i)P(X|Y_i)},\ k=1,\cdots,m$$

이와 밀접하게 연관된 모델이 **로지스틱 회귀**(logistic regression)이다. 이 모델은 현대 머신 러닝의 "hello world"로 여겨진다.
* "hello world" : 많은 프로그래밍 언어에서 가장 처음 만들어보는 기본 예제
로지스틱 회귀는 회귀(regression) 알고리즘이 아니라 분류(classification) 알고리즘이다. 나이브 베이즈와 매우 비슷하게 로지스틱 회귀는 컴퓨터보다 훨씬 오래 전부터 있었다. 하지만 간단하고 다목적으로 활용할 수 있어서 오늘날에도 여전히 유용한다.
데이터 과학자가 분류 작업에 대한 감을 빠르게 얻기 위해 데이터셋에 적용할 첫 번째 알고리즘으로 선택하는 경우가 많다.

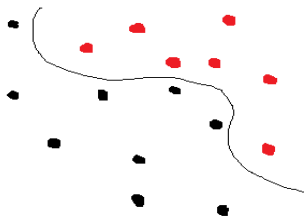
2.신경망(neural Network)

신경망의 핵심 아이디어는 아주 일찍 1950년대에 작으나마 연구되었지만(perceptron, Frank Rosenblatt, 1957) 본격적으로 시작되기까지는 수십 년이 걸렸다. 대규모 신경망을 훈련시킬

수 있는 효과적인 방법을 오랜 기간 동안 찾지 못했기 때문이다. 1980년대 중반에 여러 사람들이 제각기 역전파 알고리즘을 재발견하고 신경망에 이를 적용하기 시작하면서 상황이 바뀌었다. 이 알고리즘은 경사 하강법 최적화를 사용하여 연쇄적으로 변수가 연결된 연산을 훈련하는 방법이다.
성공적인 첫 번째 신경망 애플리케이션은 1989년 벨 연구소(Bell Labs)에서 나왔다. Yann LeCun은 초창기 합성곱 신경망(convolution neural network)과 역전파를 연결하여 손글씨 숫자 이미지를 분류하는 문제에 적용했다. **LeNet**이라 부르는 이 신경망은 우편 봉투의 우편번호 코드를 자동으로 읽기 위해 1990년대 미국 우편 서비스에 사용되었다.

3.커널 방법(Kernel trick)

초기 성공에 힙입어 1990년대에 신경망은 연구자들 사이에서 어느 정도 관심을 얻기 시작했지만, 머신 러닝의 새로운 접근 방법인 커널 방법이 인기를 얻자 신경망은 빠르게 잊혔다. 커널 방법(Kernel method)은 분류 알고리즘의 한 종류를 말하며 그중 Support Vector Machine(SVM)이 가장 유명한다. 현대적인 SVM의 공식은 1990년대초 벨연구소의 Vladimir Vapnik과 Corinna Cortes에 의해 개발되었고 1995년에 공개되었다. Vapnik과 Alexey Chervonenkis가 만든 오래된 선형 공식은 1963년에 공개되었다.
SVM은 분류 문제를 해결하기 위해 2개의 다른 범주에 속한 데이터 포인트 그룹 사이에 좋은 결정 경계(decision boundary)(그림 1-10 참고)를 찾는다. 결정 경계는 훈련 데이터를 2개의 범주에 대응하는 영역으로 나누는 직선이나 표면으로 생각할 수 있다. 새로운 데이터 포인트를 분류하려면 결정 경계 어느 쪽에 속하는지를 확인하기만 하면 된다.

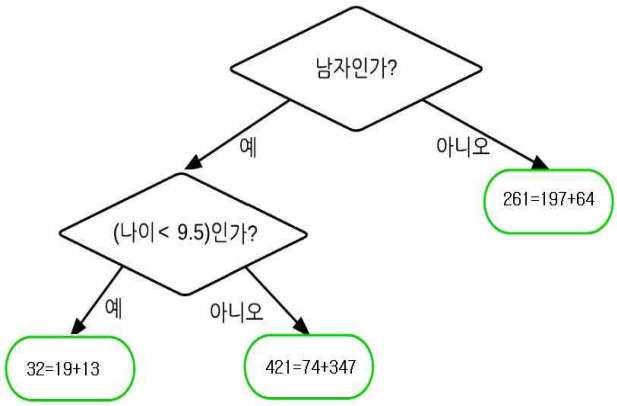


<그림 1-10> 결정 경계
SVM이 결정 경계를 찾는 과정은 두 단계이다.
1. 결정 경계가 하나의 초평면(hyperplane)으로 표현될 수 있는 새로운 고차원 표현으로 데이터를 매핑한다(그림 1-10과 같은 2차원 데이터라면 초평면은 선이 된다).
2. 초평면과 각 클래스의 가장 가까운 데이터 포인트 사이의 거리가 최대가 되는 최선의 결정 경계(하나의 분할 초평면)를 찾다. 이 단계를 마진 최대화(maximizing the margin)라고 부릅니다. 이렇게 함으로써 결정 경계가 훈련 데이터셋 이외의 새로운 샘플에 잘 일반화되도록 도와준다.
분류 문제를 간단하게 만들어 주기 위해 데이터를 고차원 표현으로 매핑하는 기법이 이론상으로는 좋아 보이지만 실제로는 컴퓨터로 구현하기 어려운 경우가 많다. 그래서 커널 기법(kernel trick)이 등장했다(커널 방법의 핵심 아이디어로 여기에서 이름을 따왔다). 요지는 다음과 같다. 새롭게 표현된 공간에서 좋은 결정 초평면을 찾기 위해 새로운 공안에 대응하는 데이터 포인트의 좌표를 실제로 구할 필요가 없다. 새로운 공간에서의 두 데이터 포인트 사이의 거리를 계산할 수만 있으면 된다. **커널 함수**(kernel function)를 사용하면 이를 효율적으로 계산할 수 있다. 커널 함수는 원본 공간에 있는 두 데이터 포인트를 명시적으로 새로운 표현으로 변환하지 않고 타겟 표현 공간에 위치했을 때의 거리를 매핑해 주는 계산 가능한 연산

이다. 커널 함수는 일반적으로 데이터로부터 학습되지 않고 직접 만들어야 한다. SVM에서 학습되는 것은 분할 초평면뿐이다.

4.결정 트리, 랜덤 포레스트, 그래디언트 부스팅

결정 트리(decision tree)는 플로차트(Flowchart) 같은 구조를 가지며 입력 데이터 포인트를 분류하거나 주어진 입력에 대해 출력 값을 예측한다(그림 1-11 참고). 결정 트리는 시각화하고 이해하기 쉽다. 데이터에서 학습되는 결정 트리는 2000년대부터 연구자들에게 크게 관심을 받기 시작했고 2010년까지는 커널 방법보다 선호하곤 했다.



<그림 1-11>결정 트리 예제: 타이타닉호 탑승객(결측값 없는 714명)의 생존 여부를 나타내는 결정 트리 (탑승객수=생존자수+사망자수)

특히 **랜덤 포레스트**(Random Forest) 알고리즘은 결정 트리 학습에 기초한 것으로 안정적이고 실전에서 유용한다. 서로 다른 결정 트리를 많이 만들고 그 출력을 앙상블하는 방법을 사용한다. 랜덤 포레스트는 다양한 문제에 적용할 수 있다. 얕은 학습에 해당하는 어떤 작업에서도 거의 항상 두 번째로 가장 좋은 알고리즘이다. 잘 알려진 머신 러닝 경연 웹 사이트인 캐글(Kaggle)(http://kaggle.com)이 2010년에 시작되었을 때부터 랜덤 포레스트가 가장 선호하는 알고리즘이 되었다. 2014년에 **그래디언트 부스팅 머신**(gradient boosting machine)이 그 뒤를 이어받았다. 랜덤 포레스트와 아주 비슷하게 그래디언트 부스팅 머신은 약한 예측 모델인 결정 트리를 앙상블하는 것을 기반으로 하는 머신 러닝 기법이다. 이 알고리즘은 이전 모델에서 놓친 데이터 포인트를 보완하는 새로운 모델을 반복적으로 훈련함으로써 머신 러닝 모델을 향상하는 방법인 **그래디언트 부스팅**(gradient boosting)을 사용한다. 결정 트리에 그래디언트 부스팅 기법을 적용하면 비슷한 성질을 가지면서도 대부분의 경우에 랜덤 포레스트의 성능을 능가하는 모델을 만듭니다. 이 알고리즘이 오늘날 지각에 관련되지 않은 데이터를 다루기 위한 알고리즘 중 최고는 아니지만 뛰어나다. 딥러닝을 제외하고 캐글 경연 대회에서 가장 많이 사용되는 기법이다.

*캐글(Kaggle)은 2010년 설립된 예측모델 및 분석 대회 플랫폼이다. 기업 및 단체에서 데이터와 해결과제를 등록하면, 데이터 과학자들이 이를 해결하는 모델을 개발하고 경쟁한다.

머신 러닝의 최근 동향

요즘 머신 러닝 알고리즘과 도구의 동향에 대한 정보를 얻는 좋은 방법은 캐글의 머신러닝 경연을 살펴보는 것이다. 매우 치열하게 경쟁하고 (어떤 대회는 수천 명이 참여하고 상금이 높

다) 다양한 종류의 머신 러닝 문제를 다루고 있기 때문에 캐글은 좋은 것과 나쁜 것을 평가할 수 있는 현실적인 잣대가 된다. “어떤 종류의 알고리즘이 경연 대회이서 우승하는 데 도움이 되는가? 상위에 랭크되어 있는 참가자들은 어떤 도구를 사용하는가?”
2016년과 2017년 캐글에는 그래디언트 부스팅 머신과 딥러닝의 두 가지 접근 방법이 주류를 이루었다. 특히 그래디언트 부스팅은 구조적인 데이터인 경우에 사용되고, 딥러닝은 이미지 분류 같은 지각에 관한 문제에 사용된다. 전자의 경우 항상 XGBoost 라이브러리를 사용한다. 이 라이브러리는 데이터 과학 분야에서 가장 인기 있는 두 언어인 파이썬(Python)과 R을 지원한다.

수업방법

1. 과제관련 동영상: python 및 머신러닝 관련 동영상
2. 강의 내용 파일: 팀즈/파일 폴더
3. 강의 내용 실습 파일: 팀즈/파일 폴더