

머신 러닝

2023 Fall

강미선

01. 인공지능이 내린 결과에 대한 신뢰성

- 미국 전기전자학회인 IEEE는 인공지능과 자율시스템의 윤리적 · 사회적 이슈를 논의하기 위한 보고서에서 인공지능이 가져올 변화를 다음과 같이 기술하였음



그림 4-1 인류의 삶과 산업혁명

미래의 인공지능 시스템은 세상에 농업혁명이나 산업혁명과 맞먹는 정도의 영향을 끼칠 역량을 가지고 있을지 모릅니다.

(Future AI systems may have the capacity to impact the world on the scale of the agricultural or industrial revolutions.)

01. 인공지능이 내린 결과에 대한 신뢰성

- 인공지능 기술은 4차 산업혁명을 이끄는 핵심 동인

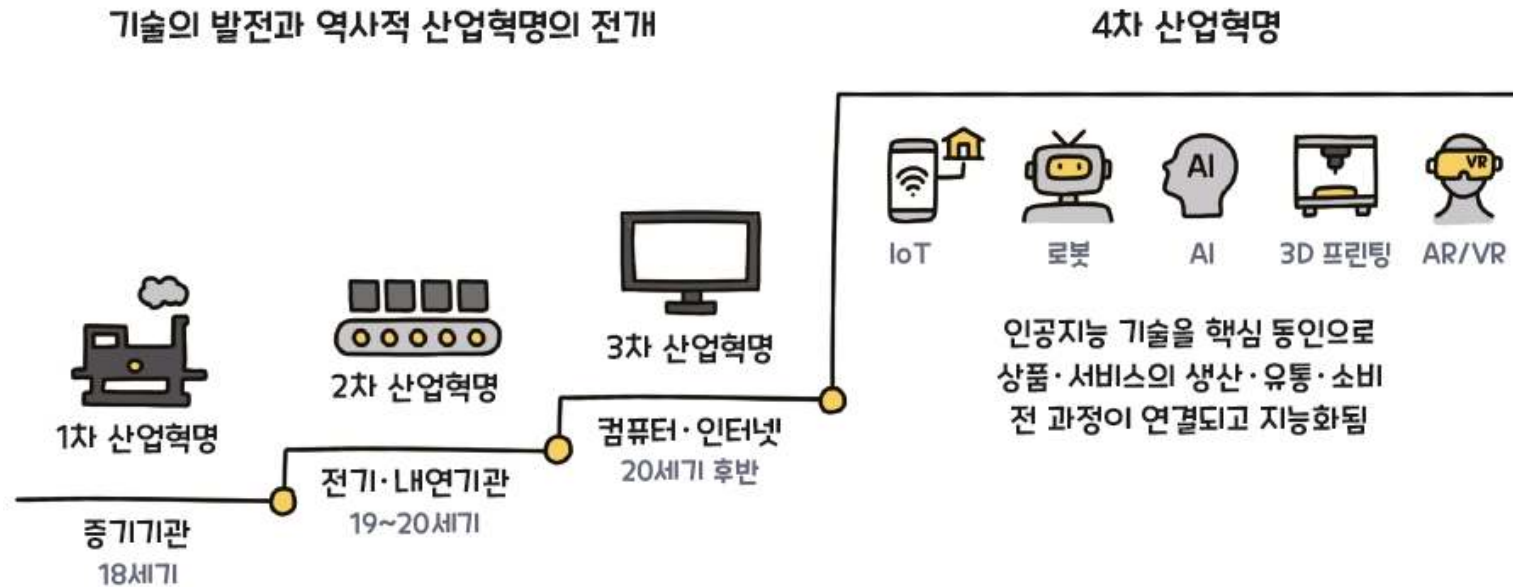


그림 4-2 산업혁명 진행 단계

- 인류의 삶을 바꿀 수 있을 정도의 파괴력을 가진 인공지능이 내놓는 결과는 과연 신뢰할 만할까?

I. 인공지능의 불완전한 판단

- 인공지능의 불완전한 판단으로 인해 세계 곳곳에서 사건 · 사고가 발생
- (예) 스코틀랜드 축구 경기에서 인공지능 카메라가 민머리인 심판을 축구공으로 오인하여 축구공이 아닌 심판만 계속 쫓아다닌 사건



그림 4-3 인공지능 카메라가 민머리 심판을 축구공으로 인식한 사건

I. 인공지능의 불완전한 판단

1) 자율주행차 사례

- 우버(Uber)의 자율주행차에 자전거 이용자가 치여 사망한 사고
- 테슬라의 자율주행차가 도로에서 이탈해 운전자가 사망한 사고



그림 4-4 우버의 자율주행차 관련 사고



그림 4-5 테슬라의 자율주행차 관련 사고

I. 인공지능의 불완전한 판단

1) 자율주행차 사례

- 최근 자율주행차의 시범 운행이 활발히 진행되자 추돌사고와 사망사고 기사가 늘기 시작함

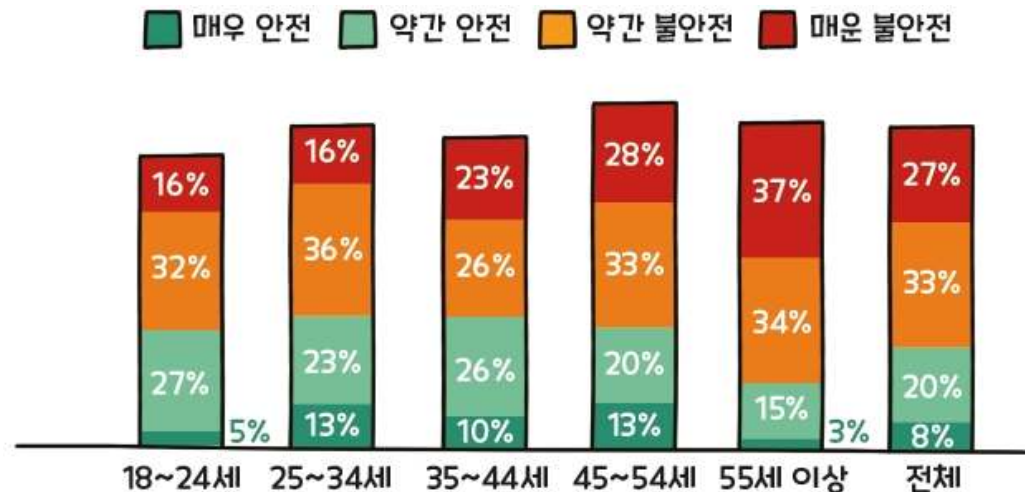


그림 4-6 연령대별 자율주행차에 대한 안전 인식률

I. 인공지능의 불완전한 판단

2) 의료 인공지능 사례

- 의료 인공지능 IBM 왓슨

- 인도 마니팔 병원에서 85% 일치율로 직장암 판단
- 왓슨의 폐암 진단율은 불과 17.8% 불과
- 유방암의 경우 비전이성은 80% 일치하지만, 전이성은 45%만 일치
- 결론 : 왓슨의 암 치료 프로젝트+신약개발을 위한 인공지능 플랫폼
→ 중단/축소



그림 4-7 의료 인공지능 왓슨(Watson)의 낮은 암 진단 일치율

II. 인공지능과 신뢰성

- 인공지능의 신뢰성 문제는 인공지능의 학습 방법에서부터 시작
 - 데이터를 이용한 학습 : 주어진 대용량의 데이터를 학습하여 모델을 만듦
 - 예측 및 분류 : 모델이 만들어진 이후, 또 다른 신규 데이터를 모델에 투입하여 인사이트(예측 및 분류)를 도출

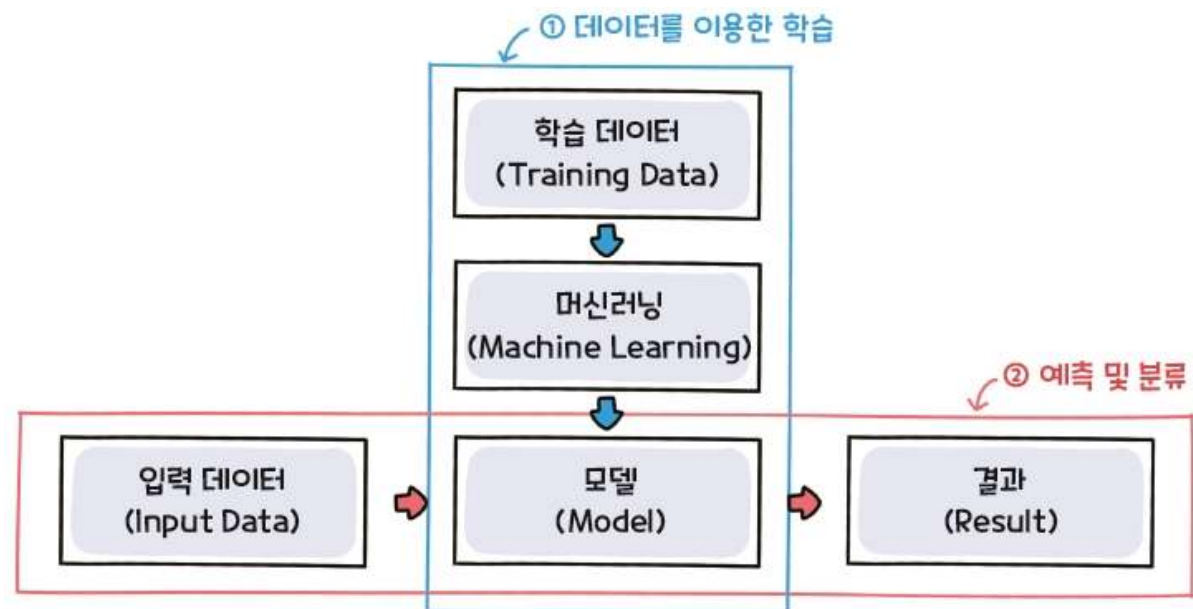
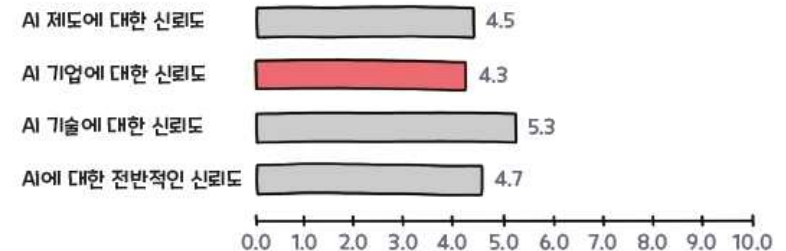


그림 4-8 인공지능 학습 방법

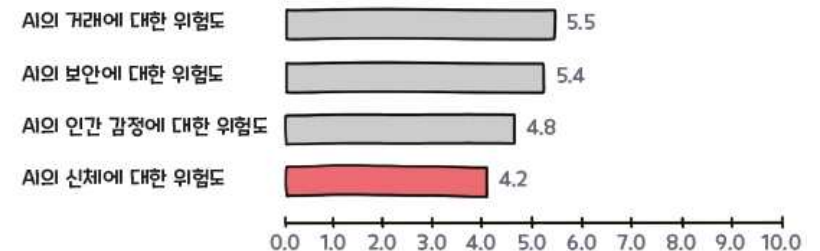
II. 인공지능과 신뢰성

- 이렇게 만들어진 모델의
정확도가 100%였다면
결과에 대한 의구심을 가질
필요가 없음
- 하지만 인공지능이 인간의
생명과 직결되는 판단들을
내놓으면서, 인공지능을
전적으로 믿을 수 있는지에
대한 의구심 증폭

인공지능에 대한 신뢰도



인공지능에 대한 위험도



인공지능에 대한 신뢰 형성

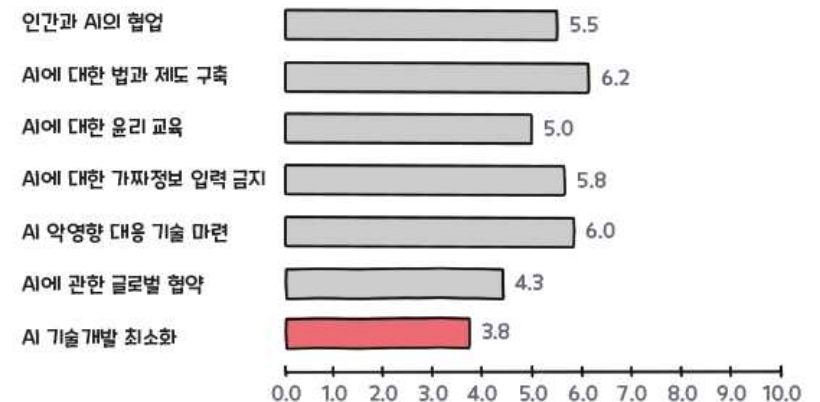


그림 4-9 국내 인공지능 신뢰도 인식 수준

I. 블랙박스의 개념

- 인공지능이 사물을 인식하는 방식은 인간과 크게 다르지 않는데, 그 이유는 인공지능도 결국 인간의 신경망을 모방한 것이기 때문
- 인공지능이 내놓은 결과는, 어떻게 또는 무엇을 근거로 그러한 결과가 나왔는지 정확하게 알 수 없는 것 아닐까?
(인간이 어떤 사물을 '고양이'라고 인식할 때 어떤 과정을 거쳤는지 설명할 수 없는 것과 같이)

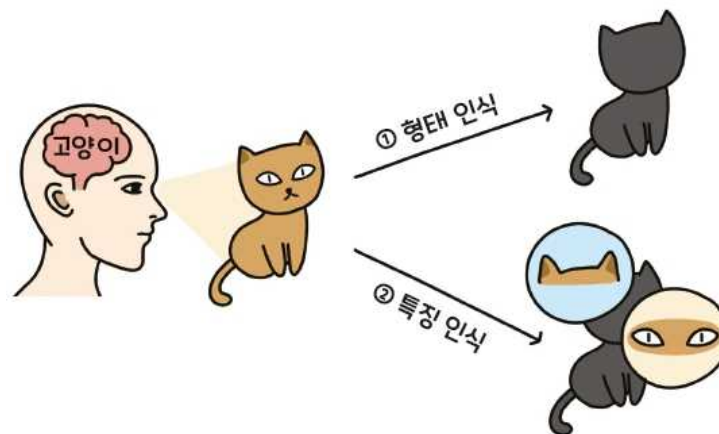


그림 4-10 인간의 고양이 인식

I. 블랙박스의 개념

- 블랙박스(Black Box)
 - 인공지능 시스템이 내놓은 판단이나 결정의 과정 또는 방법에 대해 적절한 설명할 수 없는 상태
 - 인공지능은 매우 복잡해서 전문가조차 이해하기 어렵다는, 이른바 블랙박스 문제가 발생함



그림 4-11 인공지능 블랙박스 문제

II. 블랙박스가 발생하는 이유

1) 주어진 데이터의 불완전성

- 인공지능의 학습 데이터는 인간으로부터 오기 때문에 인공지능은 객관성 ↓
- 데이터 원료 자체에 편향이 개입되므로 인공지능이 편향을 가지는 것은 불가피

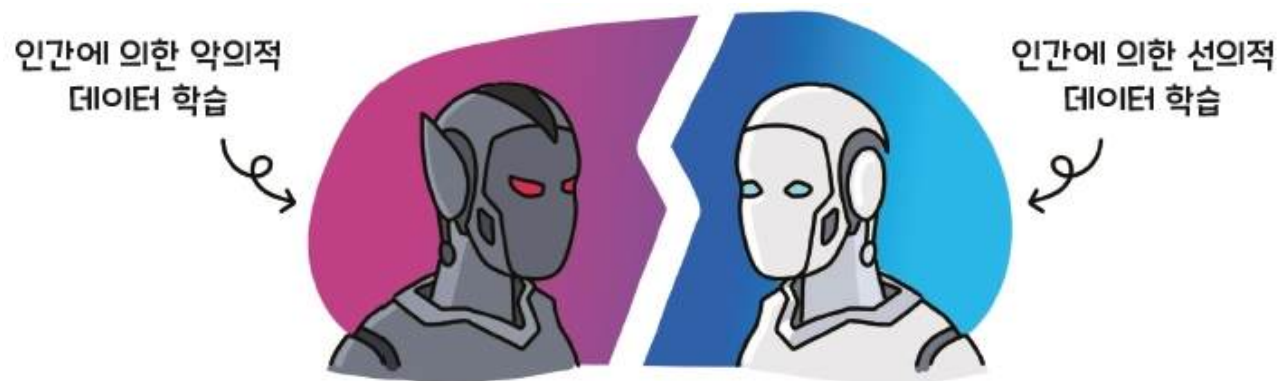


그림 4-12 데이터의 편향에 따른 인공지능의 학습 결과 차이

II. 블랙박스가 발생하는 이유

2) 불확실한 데이터 학습 과정

- 불확실한 학습

- 인공지능의 학습 과정을 인간이 이해할 수 없음
- 인공지능의 신경망은 서로 복잡하게 연결된 수백 개의 계층에서 수백만 개의 매개변수들이 상호작용하는 구조 → 사람이 인지하는 것이 사실상 불가능
- 인공지능의 알고리즘은 불투명해진 상태에서 인간 인지 영역을 넘어선 단계

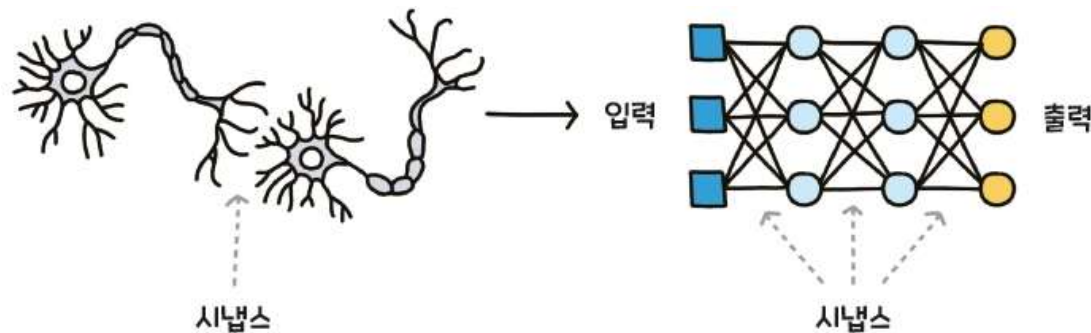


그림 4-13 인간의 뇌 구조를 모방한 인공지능의 신경망

II. 블랙박스가 발생하는 이유

하나 더 알기

인공신경망 기반 연구의 잘못된 판단 사례

- 미국 대학병원에서 폐렴 환자 사망률 낮추기 위한 인공신경망 기반 연구 진행
 - 연구 결과 천식 증상이 나타나면 폐렴이 호전된다는 이상한 결론이 나옴
 - 이러한 결론이 도출된 배경은 폐렴 환자에게 천식이 나타나면 위중해지므로, 이를 대비해 천식 환자에게 폐렴을 대비한 집중진료를 했기 때문
 - 즉, 인과관계를 고려하지 않은 상태에서 결과에만 의존해 내려진 결론
 - 결론 : 인공지능이 아무리 효율적이라고 해도 판단의 근거를 설명할 수 없다면 현장에서 사용되기 어려움

II. 블랙박스가 발생하는 이유

2) 불확실한 데이터 학습 과정

– (예) 범죄자의 재범 가능성을 예측하는 알고리즘
콤파스(COMPAS)

- » 콤파스에서 산출된 자료에 기초하여 형을 선고(인공지능 결과에 대해 전적인 신뢰 입장)
- » 하지만 콤파스는 흑인 범죄자의 재범률을 백인 범죄자보다 2배 높게 예측

주요 범죄	1건의 강도	1건의 체포 불응
기타 범죄	3건의 마약 소지	없음
콤파스 판단	 저위험 3	 고위험 10

그림 4-14 콤파스 알고리즘의 잘못된 판단

II. 블랙박스가 발생하는 이유

2) 불확실한 데이터 학습 과정

- 유럽연합은 개인정보보호 규정인 GDPR(General Data Protection Regulation)에서 유럽연합 시민은 프로파일링 등 자동화된 처리의 적용을 받지 않을 권리를 갖는다고 규정



그림 4-15 유럽연합(EU)의 GDPR

II. 블랙박스가 발생하는 이유

2) 불확실한 데이터 학습 과정

- 인공지능의 판단 과정을 이해할 수 없다면 알고리즘에 오류가 있는지, 어떤 의도로 그러한 판단을 내렸는지 알 수 없음
- 현실적으로 정확성은 떨어지더라도 그 과정을 설명할 수 있는 방식이 필요

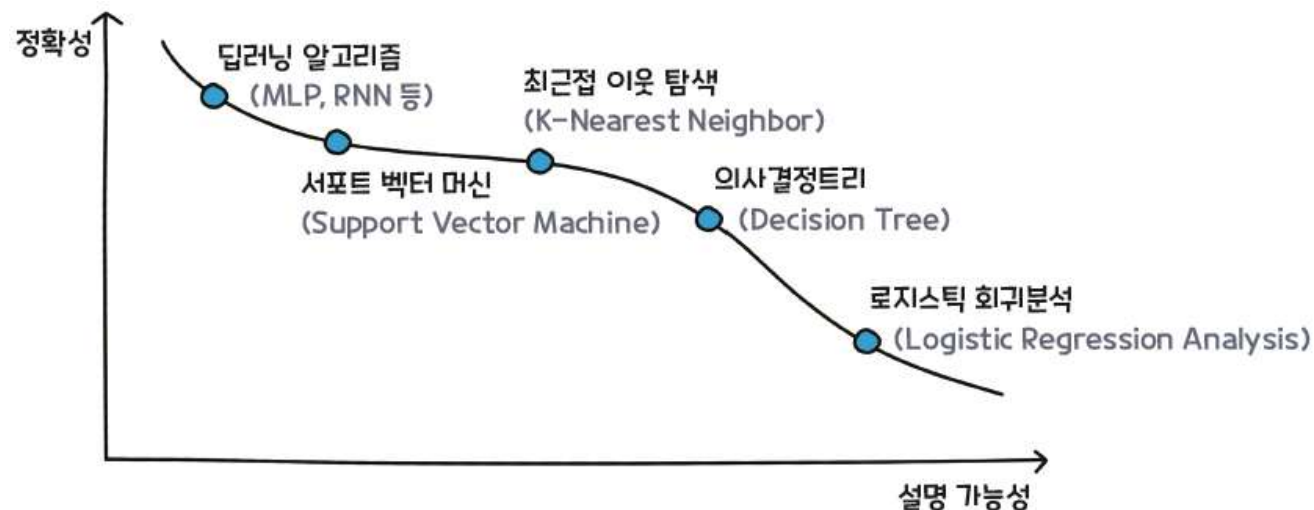


그림 4-16 인공지능의 설명 가능성과 정확성 관계 : 트레이드 오프

I. 설명가능 인공지능의 등장 배경

- SF영화 《채피(Chappie)》에서 주인공 빈센트는 '인공지능의 문제는 예측이 불가능하다는 점'이라고 지적함
- 인공지능은 이미 인간의 연산 및 추론 능력을 뛰어넘었기 때문에, 인공지능이 어떻게 연산하고 왜 그런 결과를 내놨는지는 블랙박스(Black Box)에 남아 있음



그림 4-17 영화 《채피》: “인공지능의 문제는 예측이 불가능하다는 점입니다.”

I. 설명가능 인공지능의 등장 배경

- [그림 4-18]은 인공지능으로 구현된 프로그램에 '나무늘보' 그림을 '레이싱카'라고 학습시킨 결과, 실제로 '나무늘보'를 '레이싱카'로 잘못 판단한 사례

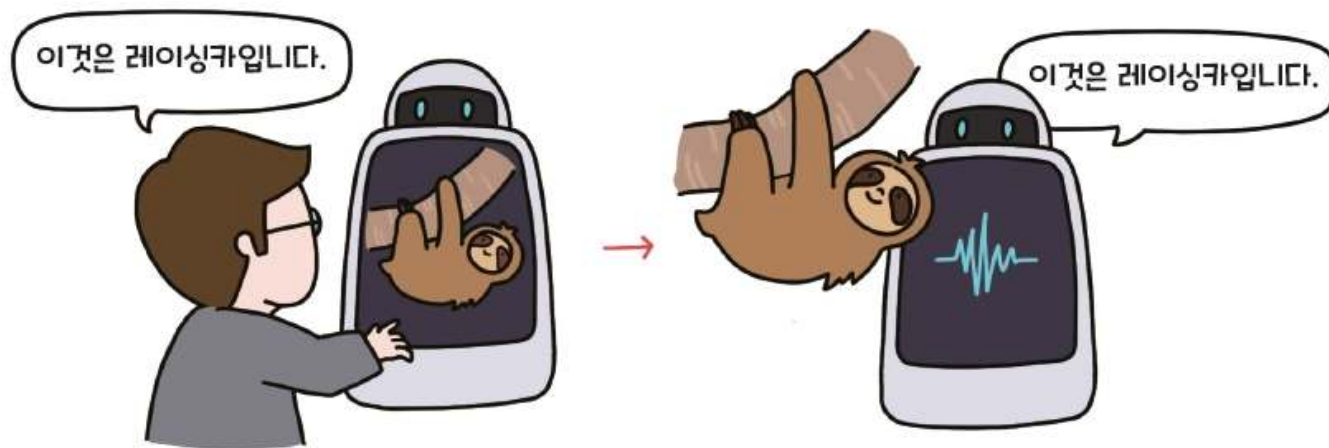


그림 4-18 잘못된 데이터를 학습한 인공지능의 판단

I. 설명가능 인공지능의 등장 배경

- [그림 4-19]는 '정지(Stop)' 표지판을 보면 멈추는 자율주행차가 노이즈 데이터가 추가된 '정지' 표지판을 보고는 '최고속도 45(Speed Limit 45)'로 잘못 판단한 사례



그림 4-19 '정지'를 '최고속도 45'로 잘못 인식한 인공지능

- 이러한 오류를 수정하려면 인공지능이 결과를 판단하는 근거 및 이유를 알아야 함

II. 설명가능 인공지능의 개념

- 설명가능 인공지능(XAI, eXplainable Artificial Intelligence)
 - 사용자가 인공지능 시스템의 동작과 최종 결과를 이해하고 올바르게 해석하도록 결과물이 생성되는 과정을 설명해주는 기술
 - 인공지능의 행위와 판단을 사람이 이해하는 형태로 설명할 수 있는 인공지능

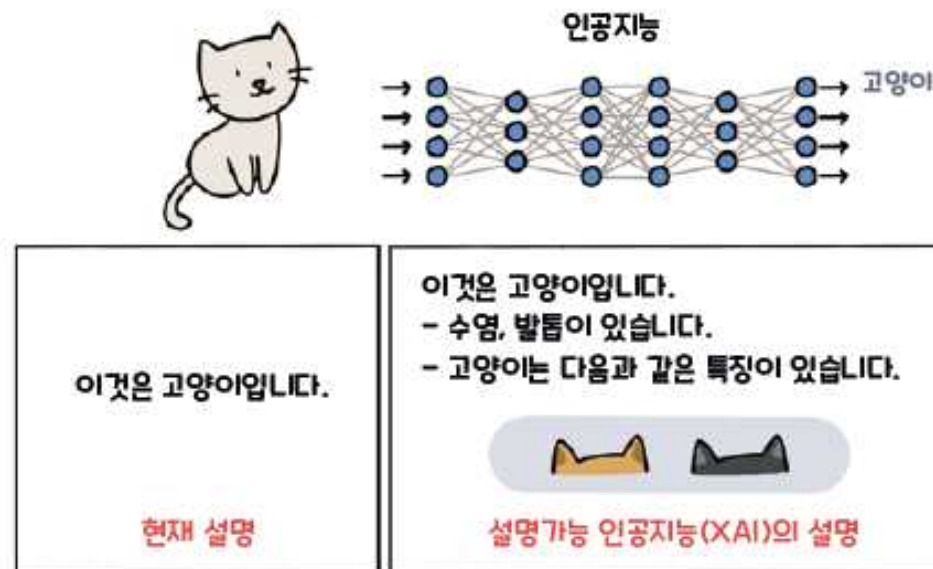


그림 4-20 설명가능 인공지능의 설명 방법

II. 설명가능 인공지능의 개념

- 현재 인공지능은 학습용 데이터를 학습 모델에 투입하여 분석한 후 그 결과를 이용자에게 전달하는데, 이때 학습 영역에서는 확률값을 계산한 후 이용자에게 전달
- [그림 4-21]은 객체(고양이) 분석 확률이 93%이기 때문에 '고양이'임을 사용자에게 제시하고 있지만, 이 과정에서 사람은 왜 인공지능이 이런 결과를 도출했는지에 대한 근거를 알 수

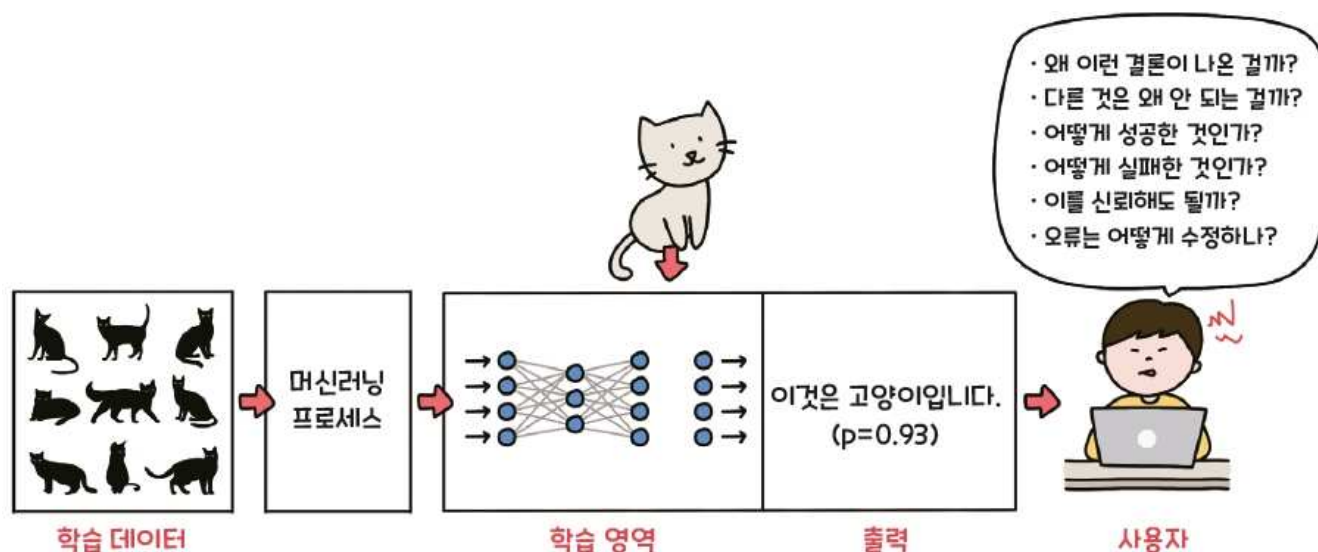


그림 4-21 현재의 인공지능 분석 과정

II. 설명가능 인공지능의 개념

- 설명가능 인공지능은 사용자에게 전달한 결과가 왜 '고양이'인지에 대한 이유를 설명함
- 단순히 그 이유를 보여주는 것을 넘어, '설명 인터페이스'를 통해 인간이 즉시 이해하고 해석할 수 있도록 도움

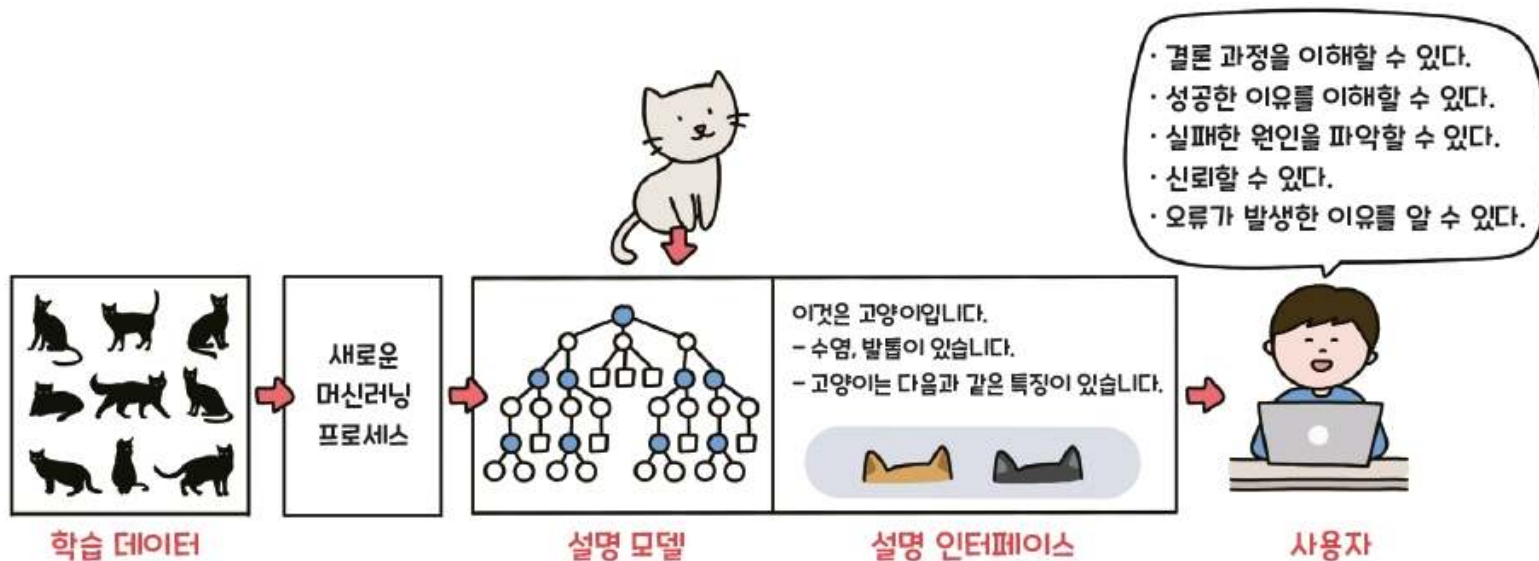


그림 4-22 설명가능 인공지능 분석 과정

II. 설명가능 인공지능의 개념

- AI 심머신(AI simMachines)
 - 설명가능 인공지능의 대표적인 사례
 - 정확한 예측에 이르기까지의 과정을 가시적으로 보여줌
 - 시계열 분석을 통해 시간 경과에 따른 변화와 변화 원인을 간격을 두고 설명
 - [그림 4-23]은 AI 심머신을 마케팅에 적용하여 동적으로 예측을 분류한 사례

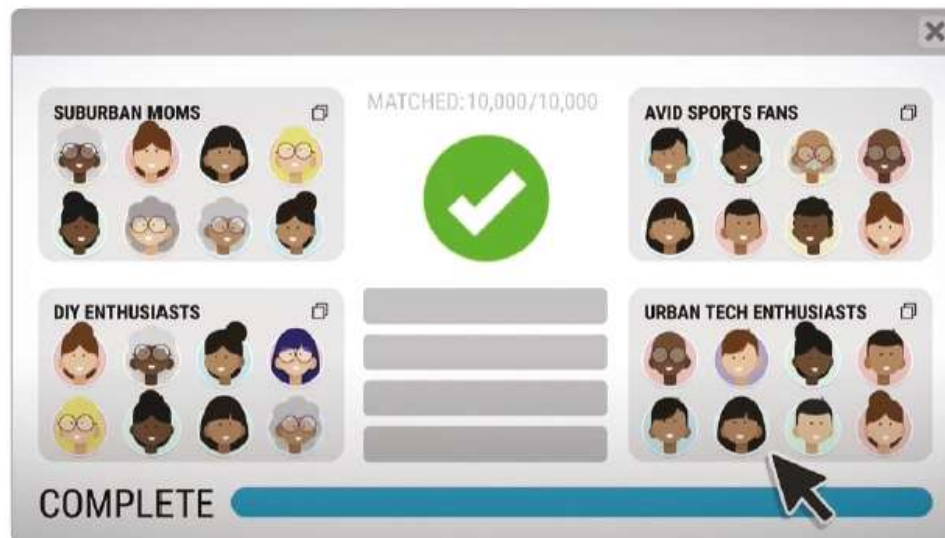


그림 4-23 AI 심머신 적용 사례(<https://bit.ly/3kJ8Px5>)

II. 설명가능 인공지능의 개념

- 자율주행차가 사물을 인식하는 방식을 시각화하는 연구가 많이 진행 중



그림 4-24 자율주행차가 사물을 인식하는 방법

III.인공지능 학습 과정의 시각화 방법

- 미국 DARPA(고등연구계획국)는 설명가능 인공지능(XAI)에 필요한 해석 가능한 모델 중 하나가 그래프 기반 모델이라는 연구 결과를 발표함
- 그래프는 점과 선을 이용한 데이터 시각화 표현으로, 학습 과정을 시각적으로 표현함으로써 설명가능 인공지능을 구현 가능

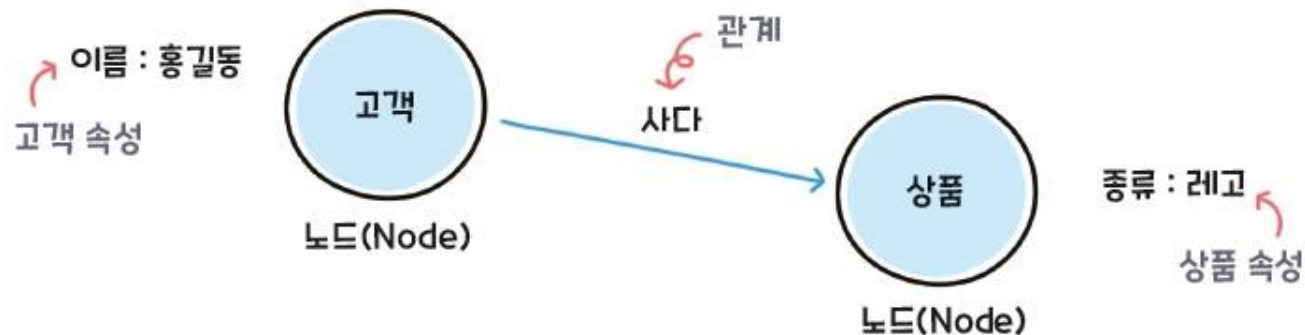


그림 4-25 데이터 시각화

III. 인공지능 학습 과정의 시각화 방법

1) 의사결정트리 이용하기

- 데이터를 분류하거나 원하는 결괏값을 예측하는 분석 방법



그림 4-26 의사결정트리 구조

- 의사결정트리의 기본 아이디어는 주어진 데이터 중 성격이 유사한 것끼리 분류하는 것, 즉 복잡성 (Entropy)이 낮아지도록 만드는 것

III. 인공지능 학습 과정의 시각화 방법

1) 의사결정트리 이용하기

- [그림 4-27]과 같이 흡연자와 비흡연자가 섞여 있는 데이터
- [그림 4-28]과 같이 주어진 데이터를 흡연 기준으로 두 개의 그룹으로 분류한다면, 각 그룹의 복잡도는 [그림 4-27]보다 낮아짐

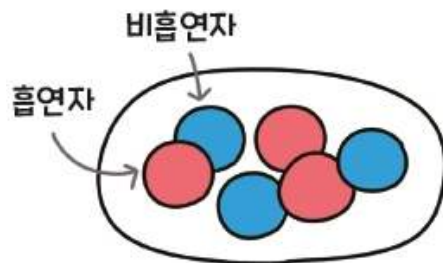


그림 4-27 흡연자와 비흡연자가 섞여 있는 데이터

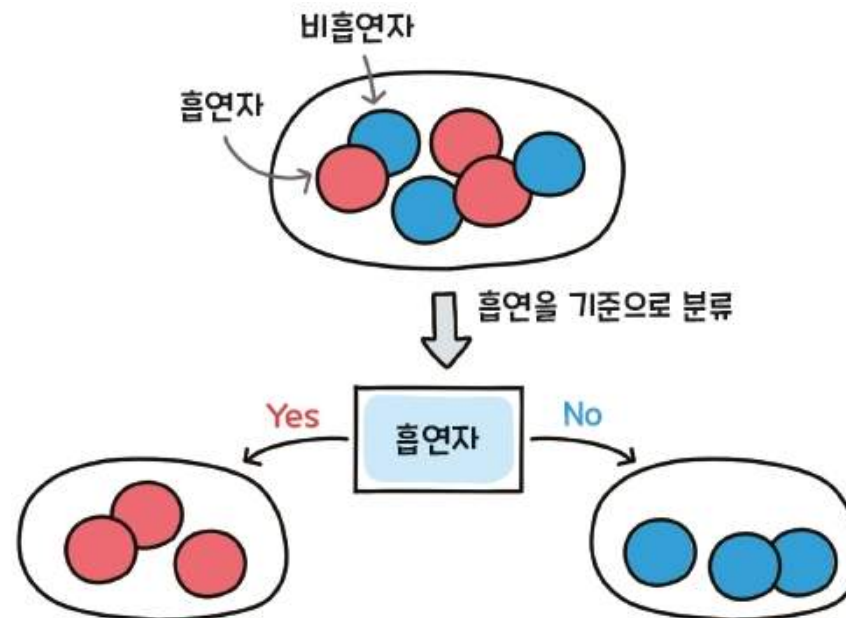


그림 4-28 흡연을 기준으로 분류된 데이터

III.인공지능 학습 과정의 시각화 방법

2) 신경망 노드에 설명 붙이기

- 인공신경망에서 설명 가능한 노드를 찾아 설명을 붙이는 방법
- (예) 인공신경망은 고양이의 수염, 털, 발톱과 같은 이미지의 특정 부분에 특정 노드를 지정하고 퍼즐 맞추기 게임처럼 모든 노드를 조합하여 대상 인식

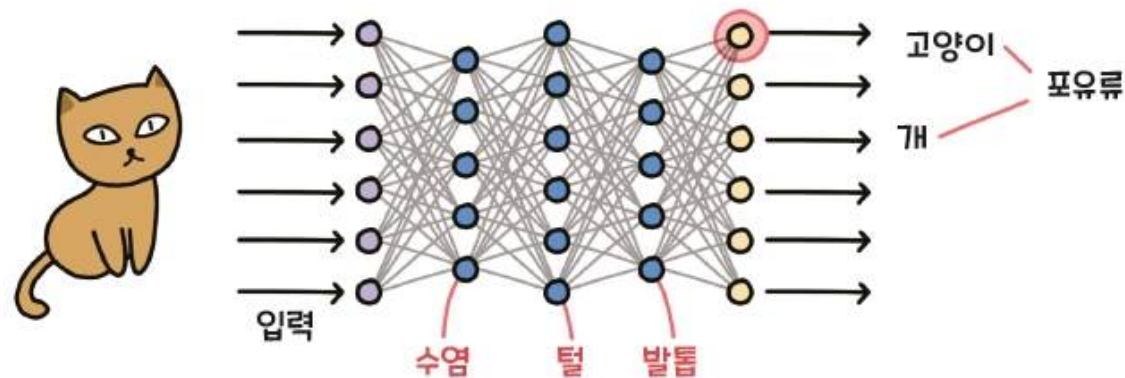


그림 4-29 신경망 노드에 설명을 붙이는 방법

III.인공지능 학습 과정의 시각화 방법

3) 모델 유추하기

- 인공지능 블랙박스에서 설명가능 모델을 유추하는 방법
- 모델 유추의 학습 진행
 - » 설명 딱지가 붙어 있는 네트워크 학습
 - » 이후 딥러닝 시스템을 훈련해 시스템이 어떻게 최종 결론에 도달했는지 설명

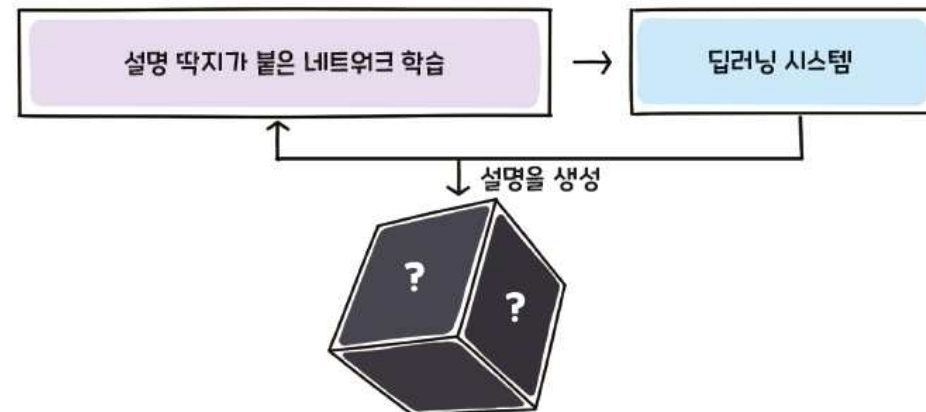


그림 4-30 모델 유추하기

IV. 설명가능 인공지능의 역량평가 요소

- 설명가능 인공지능의 역량평가 요소

표 4-1 DARPA에서 제공하는 설명가능 인공지능 역량 평가지표

항목	평가지표
사용자 만족도	<ul style="list-style-type: none">• 설명이 얼마나 명확한가?• 설명이 얼마나 유용한가?
설명모델 수준	<ul style="list-style-type: none">• 개별 의사결정 이해도• 전체 모델에 대한 이해도• 장단점 평가• '미래 해동' 예측• '개입 방법' 예측
업무수행 향상도	<ul style="list-style-type: none">• 설명이 사용자 의사결정, 업무수행 능력을 향상시켰는가?• 사용자의 이해도 평가를 위한 실험적 업무
신뢰성 평가	<ul style="list-style-type: none">• 미래에도 사용할 만큼 신뢰하는가?
오류 수정 수준(가점)	<ul style="list-style-type: none">• 인식 오류 수준• 인식 오류 수정을 위한 지속적인 훈련

V. 설명가능 인공지능의 효과

1) 기술적 관점

- 인공신경망의 내부를 들여다보면서 인공지능이 낸 결과의 인과관계 이해 가능
- 설명가능 인공지능은 설계자와 개발자에게 데이터 편향성 탐지·제거, 모델의 정확성·성능 개선 등의 효과 제공



그림 4-31 기술적 관점에서의 설명가능 인공지능 효과

V. 설명가능 인공지능의 효과

2) 비즈니스 관점

- 투명성 보장으로 감사에 대비
- 기업의 신뢰성 향상
- 직원들과의 협업



투명성 보장



기업 신뢰성 향상



직원들과의 협업

그림 4-32 비즈니스 관점에서의 설명가능 인공지능 효과

V. 설명가능 인공지능의 효과

3) 법과 제도적 관점

- 인공지능으로 인한 분쟁이 발생할 경우, 문제의 원인을 파악하는 것이 중요
- 이때 설명가능 인공지능을 활용한다면 분쟁의 원인 파악 및 중재까지 가능

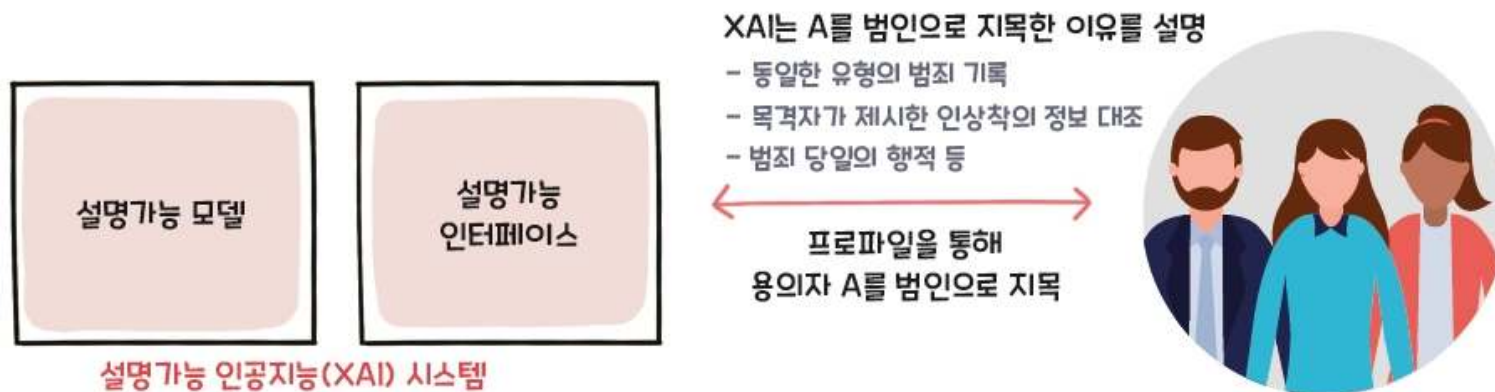


그림 4-33 설명가능 인공지능을 통한 자동 프로파일 정당성 확보

V. 설명가능 인공지능의 효과

4) 인공지능 산업 관점

- 인공지능은 블랙박스라는 한계로 특정 산업 및 분야에 국한해 사용
- 설명가능 인공지능이 활성화된다면 어떤 산업 분야에라도 적용 가능
- 인공지능이 범용적으로 활용됨으로써 인공지능 산업은 절정기에 이를 수 있음

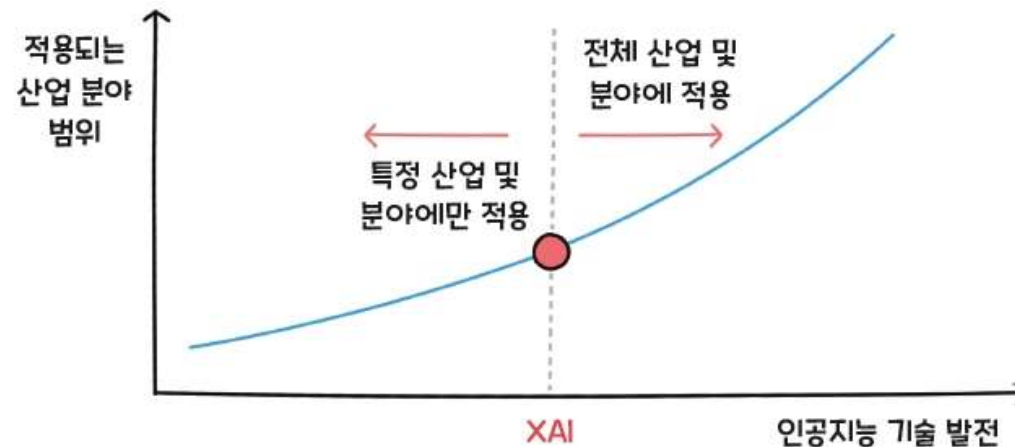


그림 4-34 설명가능 인공지능으로 인한 산업의 발전

1.5.1 지배적인 공학적 관점

- 인공지능 접근 방법

- 과학적: 인간의 지능을 충분히 연구한 다음 그 원리를 충실히 모방하는 지능 기계 제작
- 공학적: 쓸만한 지능 기계를 만들 수 있다면 굳이 인간의 지능 원리를 따르지 않아도 됨(비행기 날개는 새의 날개를 그대로 모방하지 않는다)

→ 현재는 공학적 접근방법이 지배적임

1.5.2 규칙 기반 방법론 vs. 기계학습 방법론

• 규칙 기반 방법론

- 사람이 사용하는 규칙을 수집하여 프로그래밍
- 예) 필기 숫자 인식 프로그램
 - 숫자 3은 "왼쪽에서 보면 위와 아래에 터진 골이 있고, 오른쪽에서 보면 둥근 원호가 중간에서 만나고"와 같은 규칙을 수집
- 한계 노출: 다음과 같이 규칙 위반하는 샘플이 꾸준히 발생

• 기계학습 방법론

- 인공지능 초반에는 규칙 기반이 대세였으나 1990년부터 기계학습으로 주도권이 이동함
- 충분한 데이터를 수집한 다음 기계학습 모델을 학습하는 방법(데이터-주도 패러다임)

3 7 3 3



예) 필기 숫자 인식을 위한 MNIST 데이터셋

• C와 파이썬

- 파이썬은 벡터와 행렬 처리를 코딩하는데 편리한 언어

[C 코드]

```
int a[5]={2,3,1,5,4}
int b[5]={1,2,6,5,7}
int c[5];
for(int i=0; i<5; i++)
    c[i]=a[i]+b[i];
```

[파이썬 코드]

```
import numpy as np
a=np.array([2,3,1,5,4])
b=np.array([1,2,6,5,7])
c=a+b
```

- 기계 학습은 벡터와 행렬 처리를 많이 수행하므로 파이썬을 주로 사용
- 핵심 라이브러리는 효율성때문에 C로 코딩 되어 있음(파이썬은 이들 라이브러리를 호출해 사용하는 인터페이스 언어로 사용됨)

1.6.2 볼거리

• 볼거리

- TED recommends Artificial Intelligence
 - TED가 권유하는 7편의 인공지능 강연
 - <https://cooltool.com/blog/7-best-ted-videos-about-artificial-intelligence>
- The Age of AI: 유튜브
 - <https://www.youtube.com/watch?v=UwsrzCVZAb8>
 - 아이언맨의 배우 로버트 다우니 주니어가 제작한 과학 다큐멘터리
 - 8편의 에피소드로 구성된 시즌 1 1) How far is too far? 2) Healed through AI, 3) Using AI to build a better human, 4) Love, art and stories: decoded, 5) The 'Space Architects' of Mars, 6) Will a robot take my job? 7) Saving the world one algorithm at a time, 8) How AI is searching for Aliens



- 인공지능은 진입장벽이 낮은 편이지만 상품화는 많은 노력과 인내 필요
 - 인공지능 개발에 필요한 강력한 소프트웨어가 모두 무료이고 클라우드 서비스도 있음
 - 시장에 나가려면 인간 수준의 성능 필요. 시장 경쟁이 치열
- 훌륭한 프로그래밍 환경이 제공됨
 - 입증된 논문은 구현되어 라이브러리 형태로 제공되며 꾸준히 업그레이드 중

2.1 프로그램 예제

• 아주 간단한 프로그램

- 연산자의 동작과 함수를 사용하는 방법 예시

프로그램 2-1 덧셈과 문자열 접합을 하고 난수 2개를 더하는 프로그램

```
01 # 연산자 오버로딩 예시
02 print(12+37)           # +는 두 정수를 더하는 연산자
03 print('python'+ ' is exciting.') # +는 두 문자열을 접합하는 연산자
04
05 # 라이브러리 불러오기
06 import random          # random이라는 라이브러리를 불러옴
07
08 # 정수 난수 생성
09 a=random.randint(10,20) # [10,20] 사이의 난수를 생성하고 변수 a에 대입
10 b=random.randint(10,20) # [10,20] 사이의 난수를 생성하고 변수 b에 대입
11
12 # 덧셈을 하고 결과를 출력
13 c=a+b                  # a와 b를 더하여 변수 c에 대입
14 print(a,b,c)           # 변수 a, b, c를 출력
```

```
49
python is exciting.
10 15 25
```

- #으로 시작하는 주석문
- 02~03행: + 연산자(덧셈 또는 문자열 접합). 연산자 오버로딩
- 06행: random 모듈을 불러옴
- 09~10행: 정수 난수를 생성하여 변수에 저장
- 13~14행: 두 변수를 더하고 출력

NOTE 파이썬의 연산자 오버로딩

하나의 연산자가 피연산자에 따라 서로 다른 기능을 하는 특성을 연산자 오버로딩(operator overloading)이라 부른다. 예를 들어 1.5.3항에서 예시한 바와 같이, 피연산자가 벡터이면 벡터 덧셈, 행렬이면 행렬 덧셈을 해준다. 현대적 언어일수록 연산자 오버로딩이 풍부하며, 객체지향 언어에서는 더욱 풍부하다. 파이썬은 현대적 객체지향 언어로서 연산자 오버로딩이 매우 풍부하다. 파이썬의 객체지향 특성과 사용법은 2.7절에서 설명한다.

2.2.1 클라우드 방식과 스탠드얼론 방식

• 클라우드 방식

- 프로그램과 데이터가 서버에 저장되고 관리
 - 서버에 환경이 대부분 갖추어져 있어 로그인하면 바로 프로그래밍 가능
 - 인터넷 연결만 있으면 어느 곳에서도 개발 가능. 협업 가능
- 구글의 Colab, 아마존의 SageMaker, 마이크로소프트의 Azure
- 내 프로젝트에 최적의 환경을 갖추 수 없는 한계

• 스탠드얼론 방식

- 자신에 최적의 환경 구축 가능. 프로그램과 데이터가 자신의 컴퓨터에 저장
- 소프트웨어를 설치하고 환경을 스스로 구축해야 함



그림 2-1 파이썬 프로그래밍 환경

2.2.1 클라우드 방식과 스탠드얼론 방식

NOTE 프로그래밍 환경에 대한 좋은 태도

여러 가지 프로그래밍 환경이 주어진다는 사실은 인공지능 프로그래머 층이 두터워졌다는 뜻으로 해석할 수 있다. 시범적으로 프로그래밍을 해보려는 사람과 본격적으로 제품을 개발하는 사람의 요구사항은 다를 수밖에 없기 때문에 여러 계층의 프로그래밍 환경이 주어진다. 이 절에서 설명하는 두 가지 프로그래밍 환경 중에 반드시 하나를 선택해야 하는 것이 아니다. 두 환경 사이를 쉽게 전환할 수 있는 수준의 익숙함을 확보하는 것이 현명한 태도이다.

• Colab을 사용하는 절차

1. <http://colab.research.google.com>에 접속한다.
2. 구글 계정으로 로그인한다(구글 계정이 없다면 계정을 만든 다음에 로그인한다).
3. 파이썬 프로그래밍을 한다.
4. 프로그래밍을 마치면 프로그램 파일을 구글 드라이브에 저장한다.

2.2.2 파이썬 시작하기: 클라우드 방식의 colab



(a) 로그인한 화면



(b) 파이썬 코드를 입력하고 실행한 결과 화면

그림 2-2 코랩 화면

2.2.2 파이썬 시작하기: 클라우드 방식의 colab

• 프로그래밍 장면

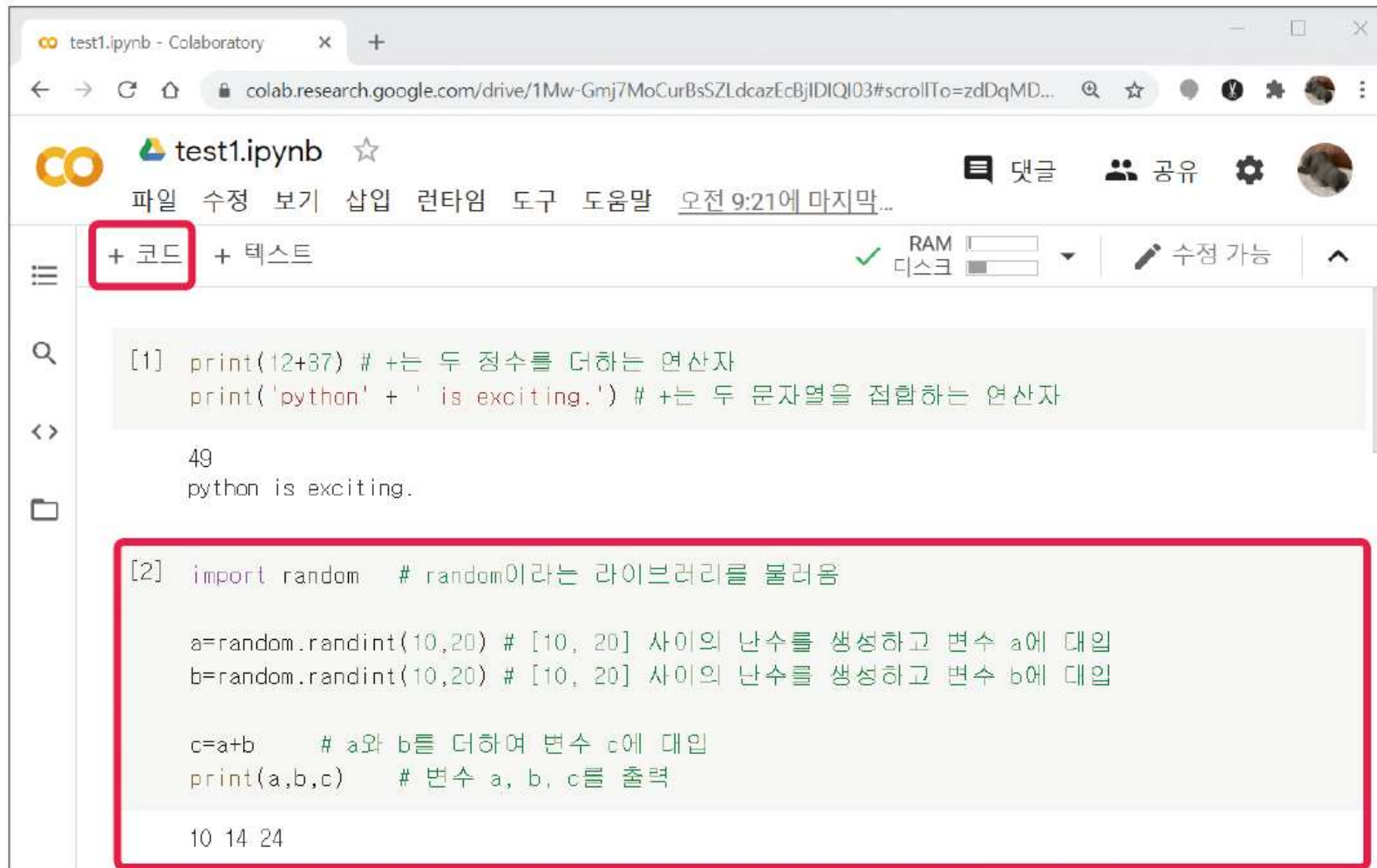


그림 2-3 코랩의 주피터 노트북에서 코드 추가

2.2.2 파이썬 시작하기: 클라우드 방식의 colab

NOTE 주피터 노트북

파이썬을 대화식으로 편리하게 실행할 수 있는 환경을 만드는 IPython 프로젝트를 수행하던 페르난도 페레스(Fernando Perez)는 2014년에 파이썬뿐 아니라 R과 Julia 언어도 지원하는 새로운 프로젝트인 주피터 노트북을 분리하여 진행한다. 나중에 주피터 노트북은 파이썬 프로그래밍을 위한 대화형 인터페이스의 표준으로 자리 잡는다. 갈릴레오가 목성(Jupiter) 주위를 도는 달의 운동을 기록한 노트북을 기리는 의미에서 주피터 노트북이라 이름 지었다고 전해진다.