



인공지능(Artificial Intelligence)의 정의

Artificial

- 인공의, 인조의, 거짓된, 가식의

Intelligence

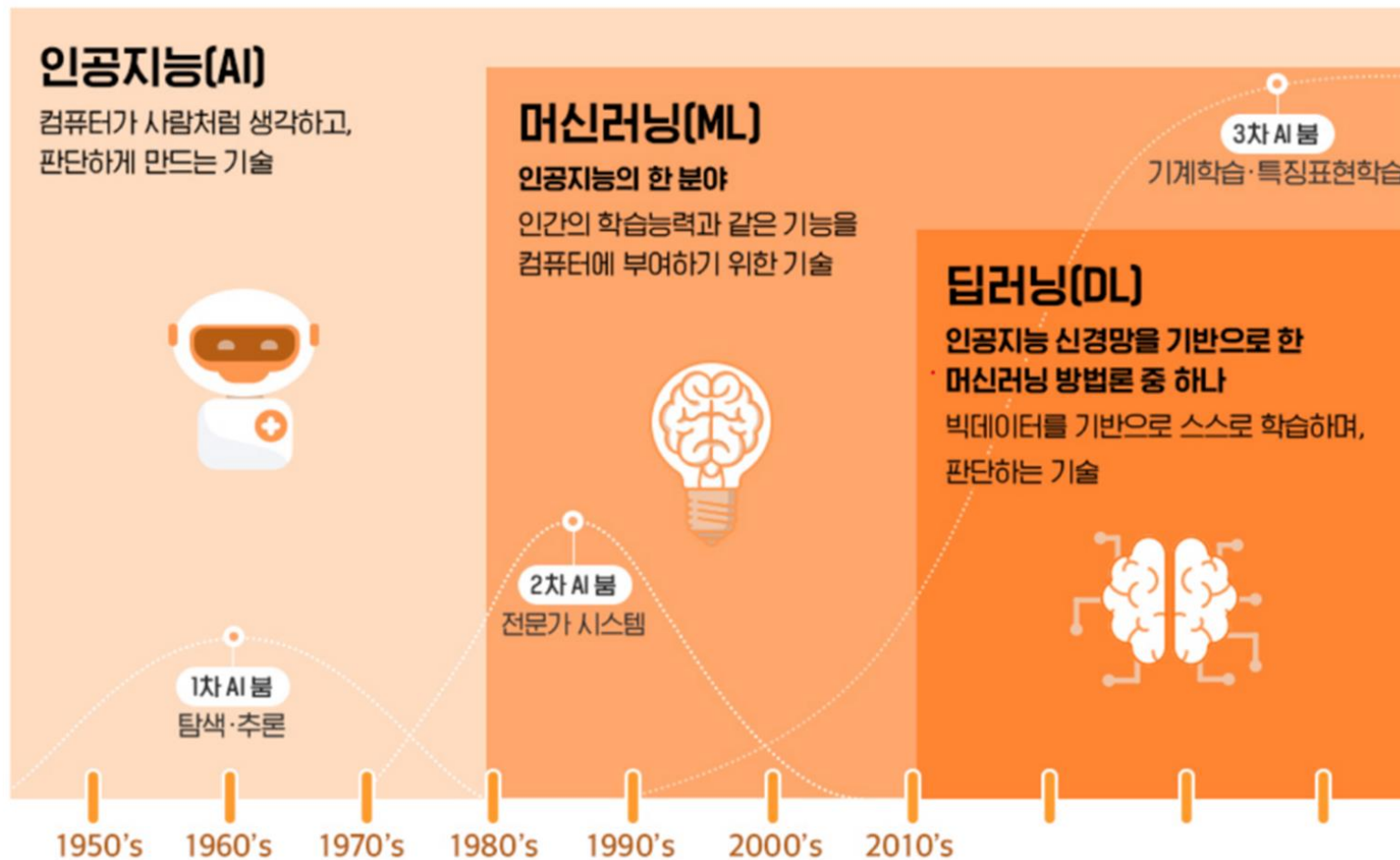
- 지능, 지성, 기밀, 정보; 정보요원

❖ 지능

인식, 추리, 학습 따위의 능력
새로운 사물이나 현상의 의미를 **이해하고 처리**하는 방법을
알아내는 지적활동 능력
(= 사고력, 관찰력, 상상력, 기억력)

❖ 지성

새로운 상황에 부딪혔을 때에 맹목적이거나 본능적 방법에
의하지 아니하고, **지적인 사고에 근거하여, 그 상황에**
적응하고, 과제를 해결하는 성질

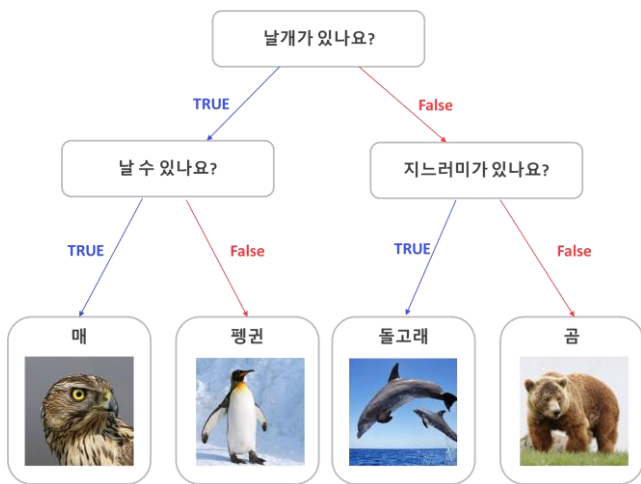


[그림 1] 인공지능 / 머신러닝 / 딥러닝의 관계

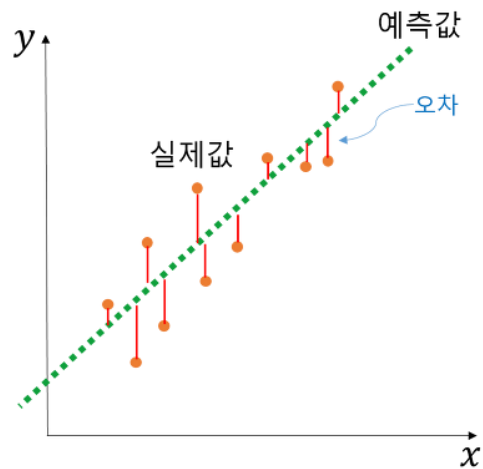


머신러닝이란?

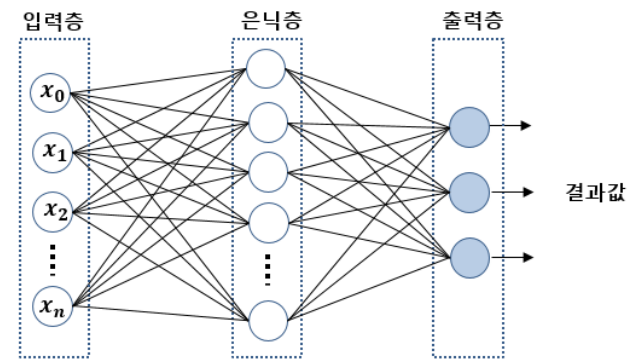
- 머신러닝(Machine Learning)은 컴퓨터가 명시적으로 프로그래밍되지 않아도 데이터로부터 학습하고 예측 또는 결정을 내릴 수 있도록 하는 인공지능 연구의 한 분야
=> 인간의 학습 능력을 컴퓨터로 실현시키기 위한 연구
- 컴퓨터가 데이터 속에서 다양한 것들을 학습하고 예측할 수 있도록 하는 인공지능의 한 형태
- 다양한 알고리즘을 통해 데이터 속 패턴을 찾아내고 이를 바탕으로 모델을 생성함



[그림 2] 의사결정나무



[그림 3] 선형회귀분석



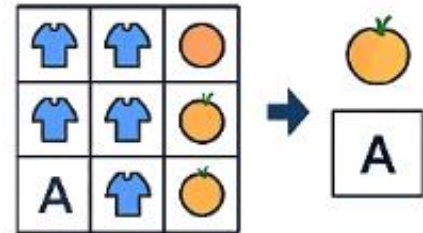
[그림 4] 인공신경망 알고리즘



머신러닝의 학습 방법

1. 지도학습

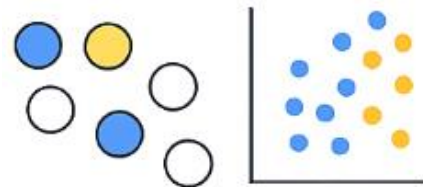
- 지도학습은 **입력과 정답(레이블)이 있는 데이터를 이용해 학습**하는 방식
- 입력값과 결과값을 함께 주고 학습을 시키는 방법으로, 과거의 데이터를 기반으로 미래의 이벤트를 예측할 때 유용함
- 예: 스팸 분류기, 질병 진단



[그림 5] 지도학습

2. 비지도 학습

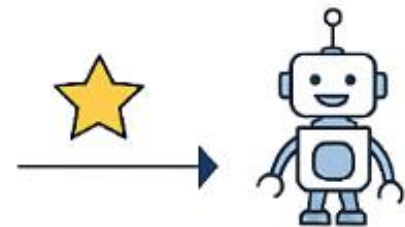
- 비지도학습은 **정답(레이블)이 없는 데이터를 기반으로 학습**하는 방식
- 데이터를 탐색하여 내부 구조와 유사한 속성들을 파악하여 분류하기 때문에 분류 기준들의 속성을 파악하는 데 효과적
- 예: 고객 군집화, 특징 추출



[그림 6] 비지도학습

3. 강화학습

- 강화학습은 에이전트가 환경과 상호작용하면서 **보상을 최대화하는 방향으로 학습**하는 방식
- 주요 개념: 상태(State), 행동(Action), 보상(Reward), 정책(Policy)
- 게임, 네비게이션 등에서 활용되며 일정 시간 내에 예상되는 보상을 극대화시킬 수 있는 동작을 선택할 수 있도록 학습함
- 예: 게임 AI, 로봇 제어, 자율주행 시스템 등

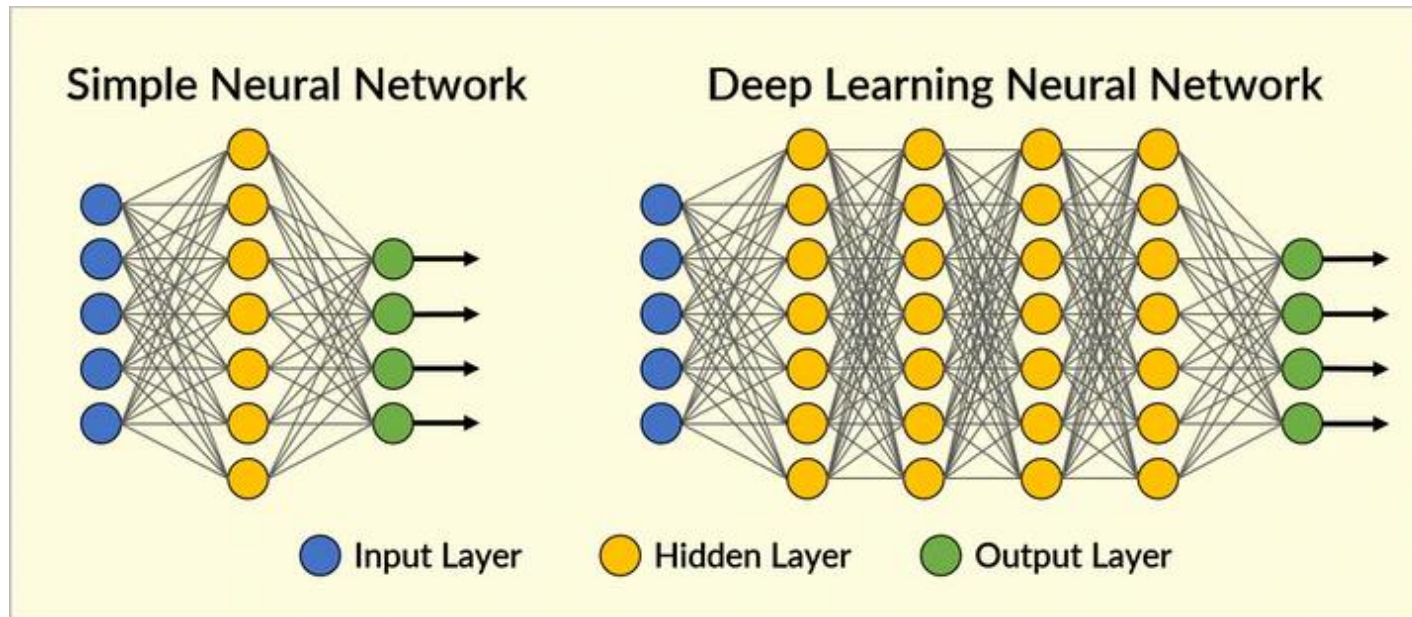


[그림 7] 강화학습



딥러닝(Deep Learning)이란?

- 머신러닝에 속하는 한 분야로, 인공 신경망(Artificial Neural Network)의 층을 연속적으로 깊게 쌓아 올려 데이터를 학습하는 방식
ANN 은닉층 레이어 두 개 이상 => 심층 신경망(Deep Neural Network, DNN) => 축약하여 딥러닝(Deep Learning)
- 머신러닝의 학습 방법(지도학습, 비지도학습, 강화학습) 모두 적용 가능
- 대표적인 예로 딥 페이크, 이미지나 영상 복원, LLM, 자율주행 자동차 등이 있음

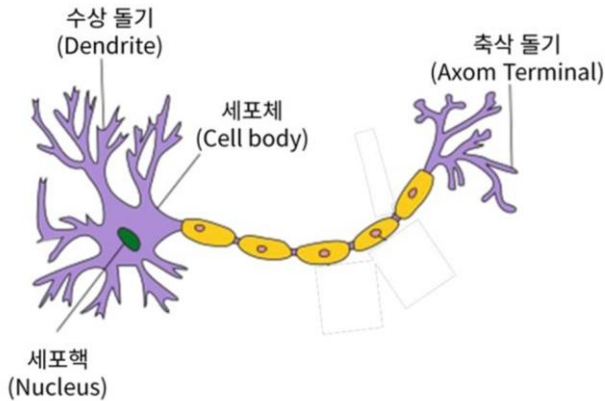


[그림 8] 인공 신경망과 심층 신경망

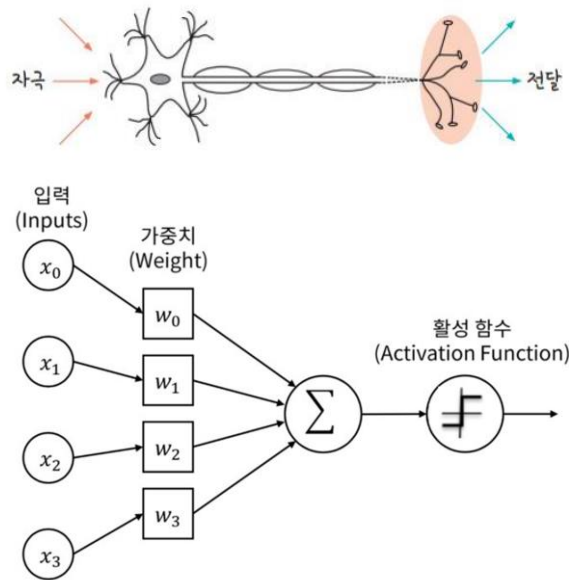


인공신경망의 시작

- 1943년, 워런 맥컬럭(Warren McCulloch)과 월터 피트(Walter Pitts) 인간 두뇌에 대한 최초의 논리적 모델 제안
‘켜고 끄는 기능이 있는 신경’을 그물망 형태로 연결하면 사람의 뇌처럼 동작할 수 있다는 가능성을 주장
- 1957년, 프랭크 로젠블랫(Frank Rosenblatt)이 최초의 신경망 모델 ‘퍼셉트론(Perceptron)’ 발표
- 퍼셉트론은 인공 신경망의 구성요소로 뉴런의 자극 전달 구조를 모방함



[그림 9] 신경세포(nerve cell, neuron)



[그림 10] 신경세포 구조와 퍼셉트론 알고리즘

```
import numpy as np

class 퍼셉트론:

    def __init__(self, w1, w2, bias):
        self.w = np.array([w1, w2])
        self.b = bias

    def 실행(self, x1, x2):
        x = np.array([x1, x2])

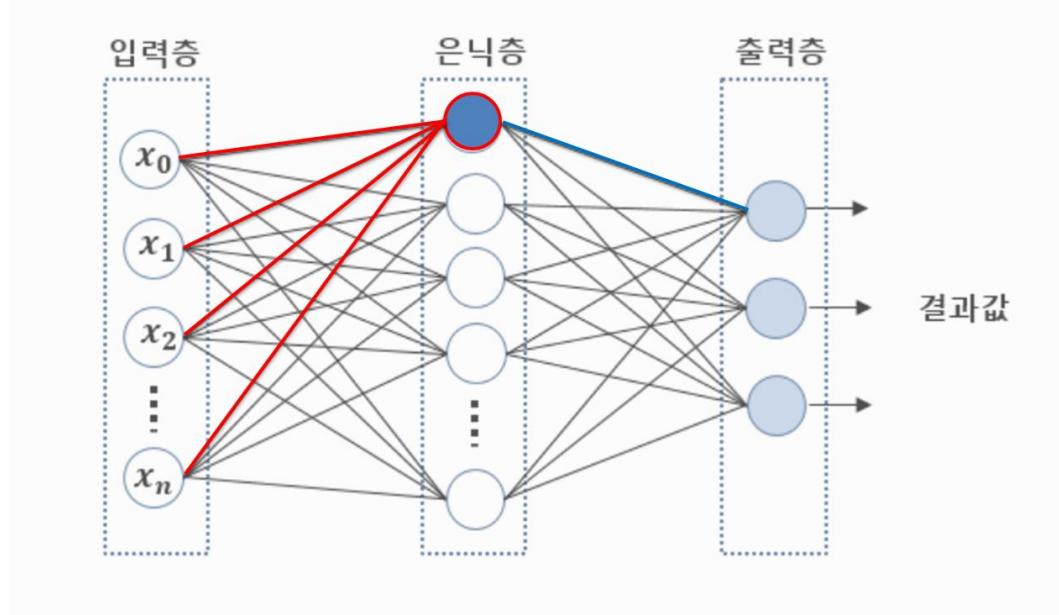
        tmp = np.sum(self.w*x) + self.b
        if tmp > 0:
            return 1
        else:
            return 0
```

[그림 11] 퍼셉트론 알고리즘 구현 코드

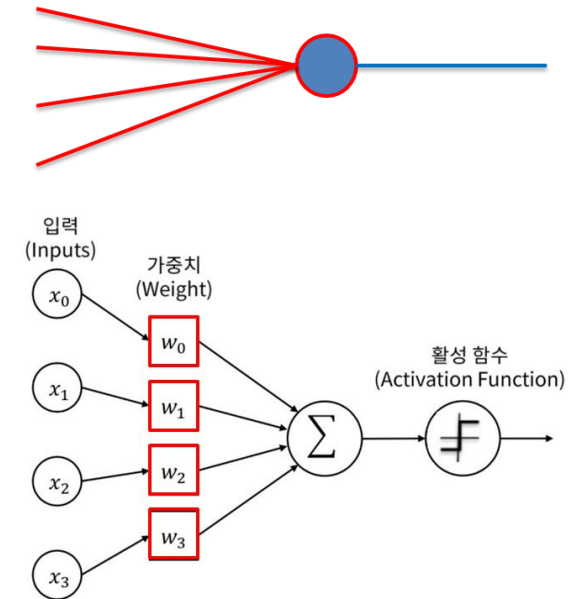


인공신경망(Artificial Neural Network, ANN)

- 입력층 : 입력 데이터의 특성을 입력받는 층 (입력값은 실수값임)
- 은닉층 : 입력층에서 전달받은 특성들을 가지고 계산한 뒤에 다음 층으로 전달, 실질적인 학습이 이루어지는 층, 학습시에 가중치와 편향 값을 업데이트함
- 출력층 : 은닉층에서 예측한 정답을 출력하는 층



[그림 12] 은닉층이 하나인 인공신경망



[그림 13] 신경망과 퍼셉트론 관계도



딥러닝에서의 블랙박스 문제란?

- 과학, 기술 공학 등의 분야에서의 블랙박스 = 내부 작업의 어떤 이해 없이 입력과 결과의 측면을 볼 수 있는 시스템
- 블랙박스 문제를 해결하는 설명 가능한 인공지능(eXplainable Artificial Intelligence, XAI)라는 개념 등장

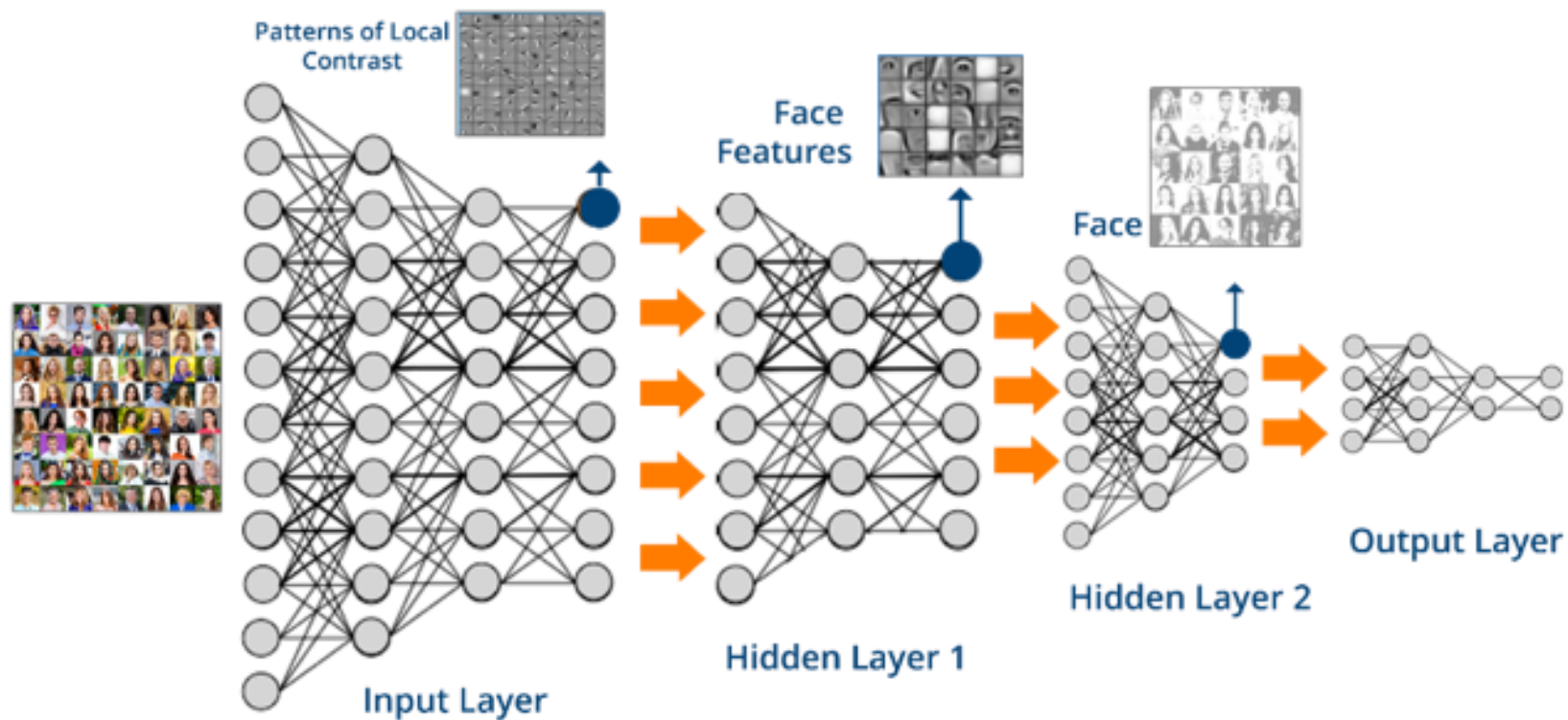


[그림 14] 공학 분야의 블랙박스



딥러닝에서의 블랙박스 문제가 나타나는 이유는?

- 블랙박스 문제는 딥러닝 모델의 히든 레이어를 분석하는 것이 너무 어렵기 때문에 발생

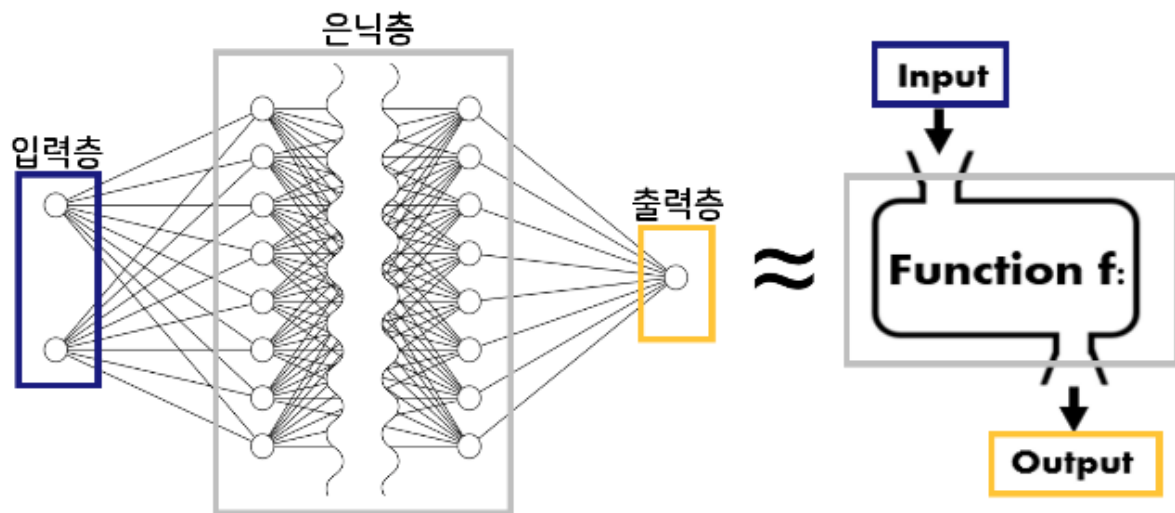


[그림 15] 얼굴 인식에 활용된 레이어구조 (Bakshi, 2017)



딥러닝 모델에 대한 가장 단순한 이해

- 인공신경망은 복잡한 구조를 가지고 있지만, 사용자 입장에서선 입력과 출력이 중요한 함수로 치환하여 생각할 수 있음
- 마찬가지로 추상화 시킨 그림 17의 수식으로 보면 원하는 출력값을 얻기 위해 수식(인공신경망)의 가중치(w)와 편향(b) 값을 변경해야 함을 알 수 있음
- 원하는 출력값이 나오도록 수식(인공신경망)을 학습 시키는 것이 중요



[그림 16] 인공신경망과 함수(Function) 비교

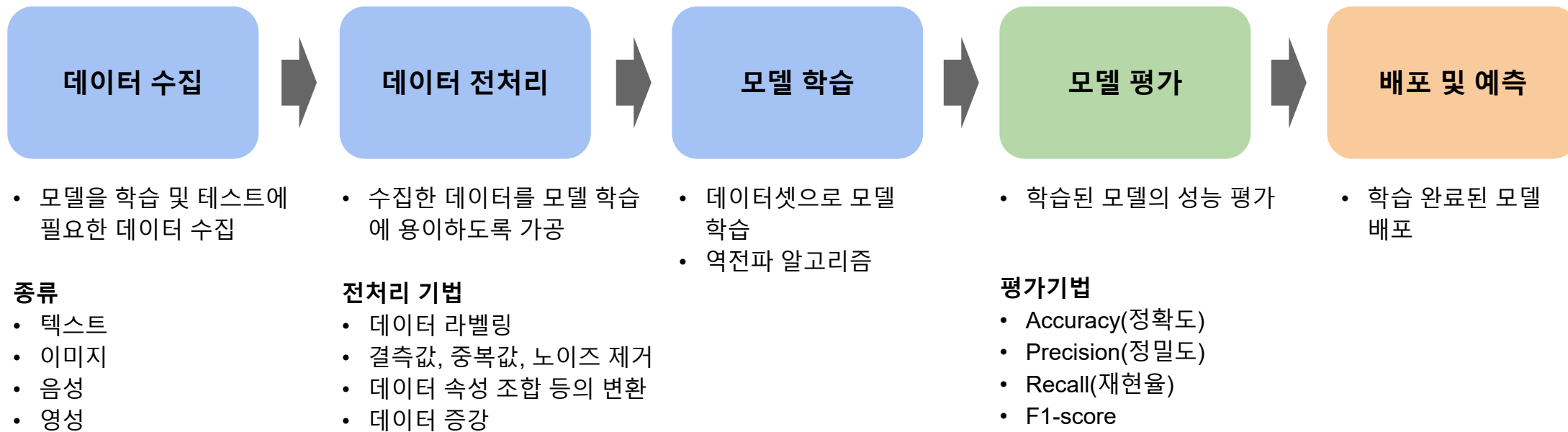
$$f(x) = \text{가중치}(w) \times \text{입력값}(x) + \text{편향}(b)$$

[그림 17] 이해를 돕기 위한 수식 형태로의 표현



딥러닝 워크플로우

- 딥러닝 모델의 학습부터 시작하여 실제 서비스에 배포하기까지의 흐름





모델 평가 방법

• Accuracy(정확도)

- 분류 모델을 평가하기에 가장 단순한 지표
- 불균형한 클래스를 가진 데이터셋을 평가하기는 어렵다는 단점이 있음

• Precision(정밀도)

- 분류 모델이 Positive 판정하는 것, 실제로 Positive인 샘플의 비율
- Positive로 검출된 결과가 얼마나 정확한지 나타냄
- 동어 : PPV(Positive Predictive Value)

• Recall(재현율)

- 실제 Positive 샘플 중 분류 모델이 Positive로 판정한 비율
- 분류 모델이 실제 Positive 클래스를 얼마나 빠지지 않고 잘 잡아내는지 나타냄
- 동어 : TPR(Ture Positive Rate, 양성률), Sensitivity(민감도)

Accuracy $\frac{TP + TN}{TP + FP + FN + TN}$		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

[그림 18] Accuracy 수식

Precision $\frac{TP}{TP + FP}$		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

$$\text{Precision} = \frac{TP}{TP + FP}$$

[그림 19] Precision 수식

Recall $\frac{TP}{TP + FN}$		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

$$\text{Recall} = \frac{TP}{TP + FN}$$

[그림 20] Recall 수식



모델평가 방법(Precision-Recall의 관계)

• F1-score

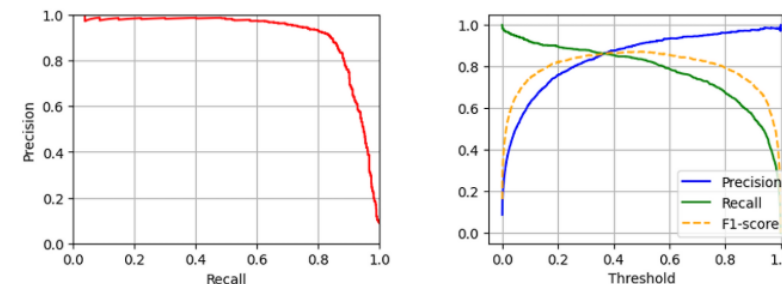
- Precision과 Recall의 조화평균으로 정의됨
- F1-score는 0과 1사이 값이며, 1에 가까울수록 분류 성능이 좋음을 나타냄

$$F1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

[그림 21] F1-score 수식

• Precision-Recall Curve

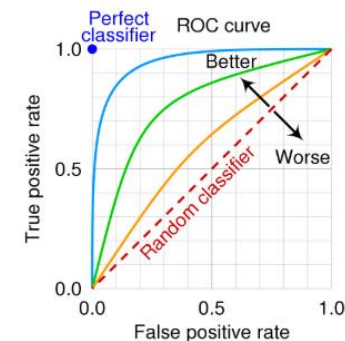
- Precision과 Recall은 trade-off 관계임
- 분류 모델의 decision threshold를 통해 trade-off 관계를 조절
- decision threshold란 분류 모델의 결과인 0~1사이의 값을 positive 또는 negative로 결정하는 경계임
- Precision-recall curve 위의 한 점은 특정 threshold에 해당하는 Precision과 Recall 값을 의미함



[그림 22] Precision-Recall Curve 그래프

• ROC(receiver operating characteristic) Curve

- threshold에 따른(=Recall)과 FPR(=Fall-out)을 나타낸 그래프
- 대각선을 기준으로 좌상단으로 붙어있는 ROC curve일수록 좋은 분류 성능을 나타냄



[그림 23] ROC Curve 그래프



Vertex AI

- 머신러닝 모델과 AI 애플리케이션을 교육하고 대규모로 배포하기 위해 구글 클라우드에서 만든 중앙 집중식 플랫폼
- AI 응용 프로그램에서 사용할 대규모 언어 모델(LLM)을 사용자 지정 가능
- 머신러닝 워크플로를 간소화하고 데이터 세트 생애 주기를 단순화 시킬 수 있도록 데이터 관리, 모델 배포 등에 도움을 줌
- 데이터 엔지니어링, 데이터 과학, ML 엔지니어링 워크플로를 결합함

Custom ML Platform

Fast track ML model development & deployment with managed data science tools, workflows, & infra

Model Garden

Jump-start ML projects with access to a variety of pre-built models including APIs & foundation models

Generative AI Studio

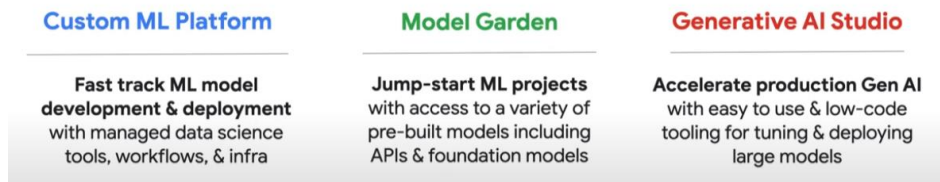
Accelerate production Gen AI with easy to use & low-code tooling for tuning & deploying large models

[그림 24] Vertex AI 주요 서비스



Vertex AI는 모델 학습 및 배포를 위한 여러 옵션을 제공

- AutoML을 사용하면 코드를 작성하거나 데이터 분할을 준비하지 않고도 테이블 형식, 이미지, 텍스트 또는 동영상 데이터를 학습시킬 수 있음
- 이러한 모델은 온라인 예측을 위해 배포되거나 일괄 예측을 위해 직접 쿼리 될 수 있음



[그림 24] Vertex AI 주요 서비스

Custom ML Platform

- 커스텀 학습을 사용하면 선호하는 ML 프레임워크 사용, 자체 학습 코드 작성, 초매개변수 조정 옵션 선택 등 학습 프로세스를 완벽하게 제어할 수 있음
- 커스텀 학습 모델을 Model Registry로 가져와 사전 빌드 된 또는 커스텀 컨테이너를 사용하여 온라인 예측을 위한 엔드포인트에 배포할 수 있음
- 또는 일괄 예측을 위해 직접 쿼리 할 수 있음

Model Garden

- Model Garden을 사용하면 Vertex AI를 검색, 테스트, 맞춤 설정, 배포하고 오픈소스 모델과 애셋을 선택할 수 있음

Generative AI Studio

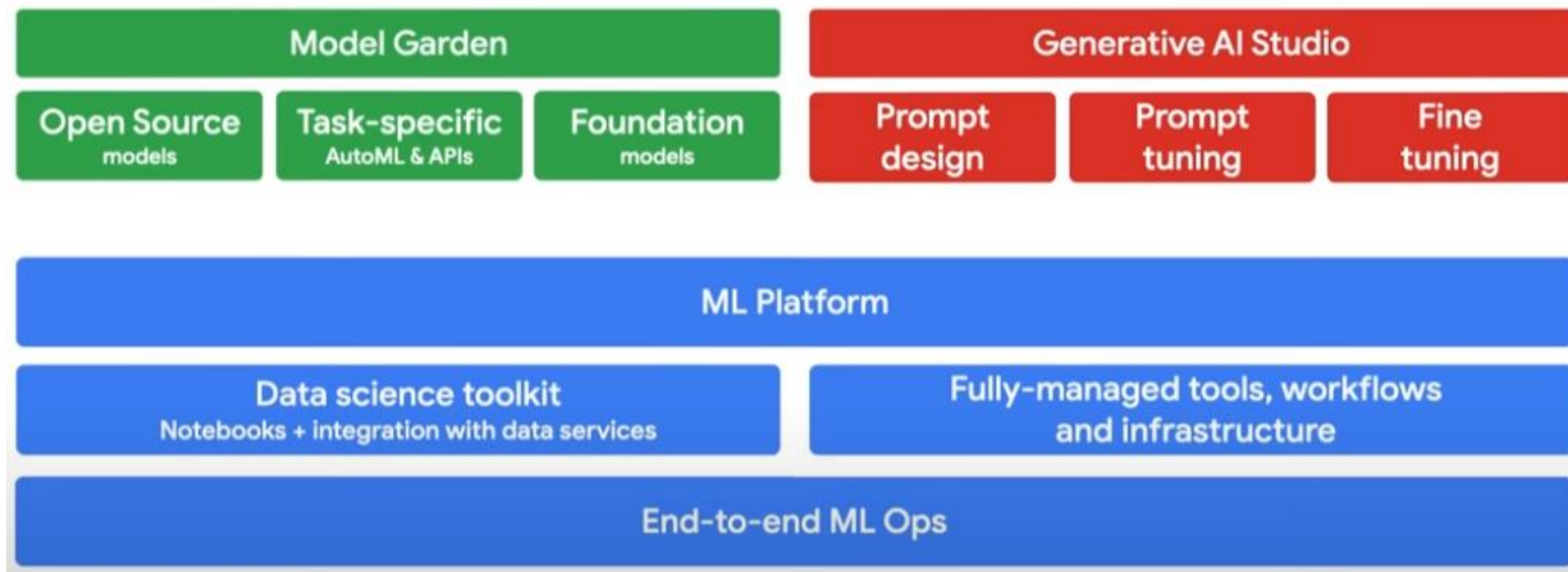
- 생성형 AI는 여러 모달(텍스트, 코드, 이미지, 음성)에 대한 Google의 대규모 생성형 AI 모델에 대한 액세스를 제공함
- 필요에 맞게 Google의 LLM을 조정한 후 AI 기반 애플리케이션에서 사용하도록 배포할 수 있음

II Vertex AI AutoML 소개

교육 서비스



Vertex AI 구성

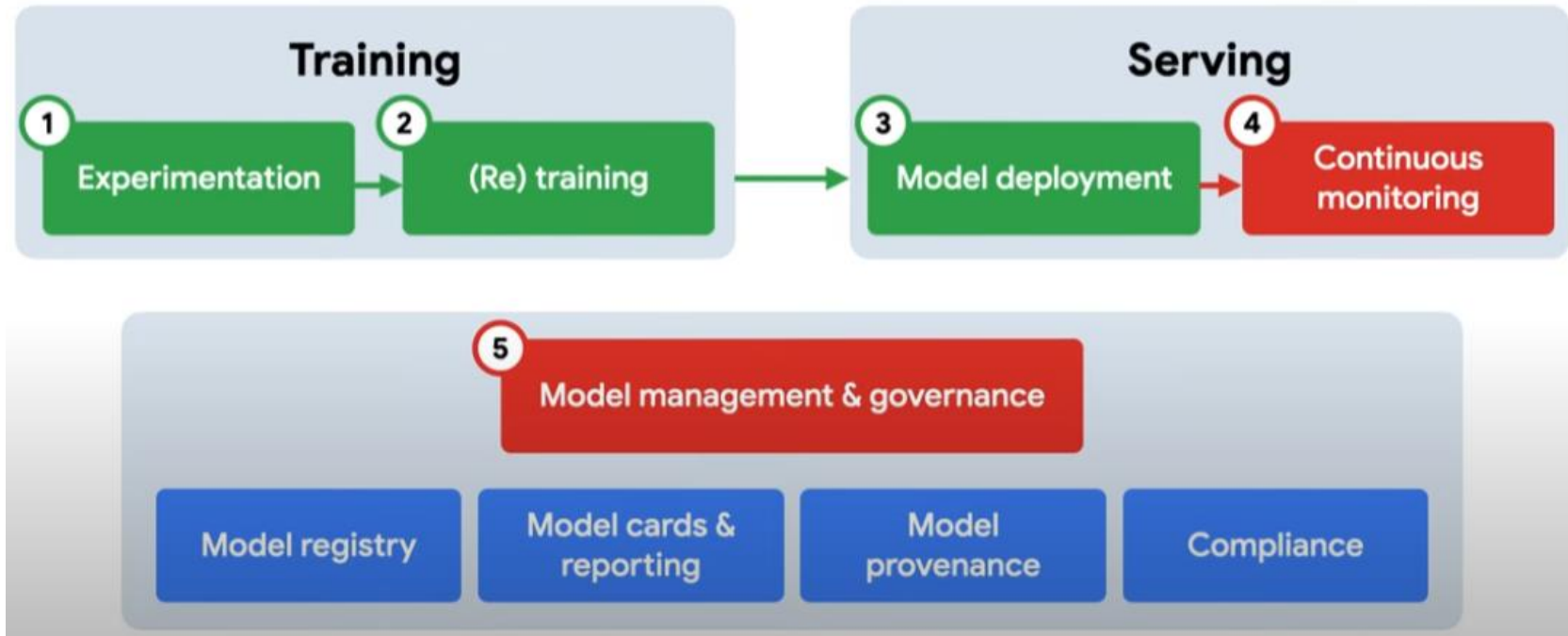


[그림 25] Vertex AI 주요 서비스 구조



Vertex AI의 end-to-end MLOps

- “Data Scientists”와 “ML Engineer”가 전체 개발 수명 주기 동안 ML 프로젝트를 효율적이고 책임감 있게 관리, 모니터링, 관리 및 설명할 수 있도록 함



[그림 26] ML Ops 절차



AutoML 이란?

- AutoML (Automated Machine Learning) ML 전문지식 없이 모델 생성 가능
- 파이프라인 자동화: 전처리 → 학습 → 튜닝 → 평가

Vertex AI AutoML 주요 기능

- 데이터 업로드만으로 모델 학습 가능
- 예측 성능 최적화를 위한 하이퍼 파라미터 자동 탐색
- 학습된 모델을 엔드포인트에 즉시 배포 가능
- UI 또는 API를 통한 접근 가능

Task 유형	지원기능
이미지 분류	• 정적 이미지 분석
객체 감지	• 이미지 내 객체 위치 식별
텍스트 분류	• 뉴스/리뷰 분류 등
텍스트 감정 분석	• 긍정/부정 등 분석
표 형식 예측	• 구조화된 데이터 학습

[표 1] Vertex AI AutoML 지원 타입

II Vertex AI AutoML 소개

교육 서비스



Vertex AI의 AutoML

최소한의 기술적 노력으로 모델을 만들고 학습할 수 있어 개발에 투자하기에 앞서 AutoML을 사용하여 빠르게 모델 프로토타입을 제작 및 새로운 데이터 세트 확인 가능

	AutoML	커스텀 학습
필요한 데이터 과학 전문 지식	<ul style="list-style-type: none"> 아니요 	<ul style="list-style-type: none"> 학습 애플리케이션을 개발하고 특성 추출과 같은 일부 데이터 준비 작업을 수행함
프로그래밍 능력 필요	<ul style="list-style-type: none"> AutoML에서는 코딩이 필요하지 않음 	<ul style="list-style-type: none"> 학습 애플리케이션을 개발하는 데 필요함
학습 모델 소요 시간	<ul style="list-style-type: none"> 상대적으로 적게 걸림 필요한 데이터 준비가 적고 개발이 필요하지 않음 	<ul style="list-style-type: none"> 상대적으로 많이 걸림. 필요한 데이터 준비가 많고 학습 애플리케이션 개발도 필요함
머신러닝 목표 한도	<ul style="list-style-type: none"> AutoML의 사전 정의된 목표 중 하나를 목표로 삼아야 함 	<ul style="list-style-type: none"> 아니요
초매개변수 조정으로 모델 성능을 수동으로 최적화할 수 있음	<ul style="list-style-type: none"> 아니요 AutoML은 일부 자동화된 초매개변수 조정을 수행하지만 사용되는 값을 수정할 수는 없음 	<ul style="list-style-type: none"> 각 학습 실행 중에 실험 및 비교를 위해 모델을 조정할 수 있음
학습 환경의 측면을 제어할 수 있음	<ul style="list-style-type: none"> 제한적으로 가능. 이미지 및 테이블 형식 데이터 세트의 경우 학습시킬 노드 시간 및 학습 초기 중단 허용 여부를 지정할 수 있음 	<ul style="list-style-type: none"> 예. Compute Engine 머신 유형, 디스크 크기, 머신러닝 프레임워크, 노드 수와 같은 환경의 측면을 지정할 수 있음
데이터 크기 제한	<ul style="list-style-type: none"> AutoML은 관리형 데이터 세트를 사용며, 데이터 크기 제한은 데이터 세트 유형에 따라 다름 <ul style="list-style-type: none"> 이미지 학습 데이터 준비 테이블 형식 학습 데이터 준비 동영상 교육 준비 	<ul style="list-style-type: none"> 관리되지 않는 데이터 세트의 경우에는 데이터 크기 제한이 없음 관리형 데이터 세트에는 Vertex AI에서 생성되고 호스팅되는 관리형 데이터 세트 객체와 동일한 한도가 적용되며 AutoML 모델을 학습시키는 데 사용됨

[표 2] AutoML vs CustomML