

인공지능의 개요 및 심층 신경망(DNN)

Hanjin Cho



Electronic & Electrical Convergence Engineering
Hongik University
Republic of Korea

- 인공지능의 개요
- 심층 신경망(DNN)의 개요
- 심층 신경망(DNN)의 학습 과정

■ 규칙 기반 방법론과 학습 기반 방법론의 전환

- 규칙 기반 방법은 사람이 영상을 보고 규칙을 정의하고 직접 구현하는 방식.
- 데이터에 따라 최적의 규칙이나 필터를 매번 설계해야 하므로 유연성이 부족.
 - 예: 스케일 공간에서 가우시안보다 나은 필터는 없는지, 데이터마다 다르게 설계할 필요가 있는지 판단 어려움.
- 인공지능은 데이터를 통해 최적의 필터나 규칙을 스스로 학습함.
- 하지만 현재 규칙 기반의 방법론은 전처리 및 후처리에 적용하여 기계학습 방법론의 성능향상을 목적으로 사용.

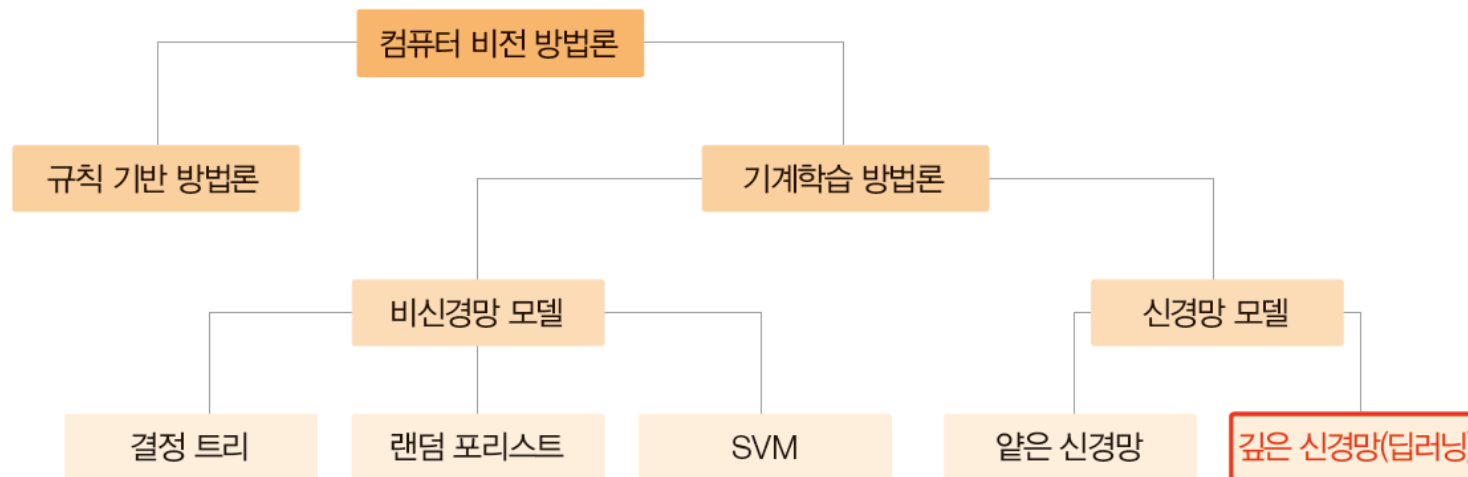
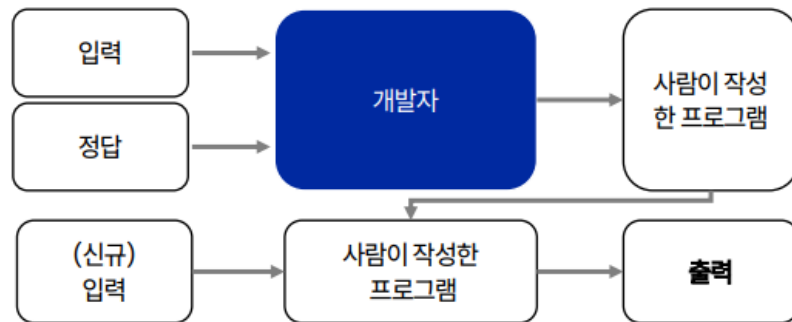


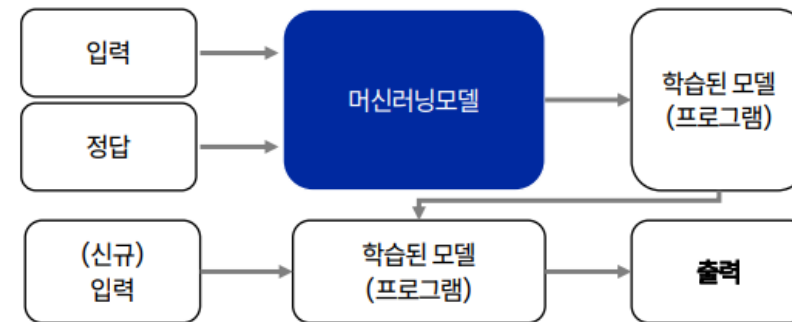
그림 7-2 컴퓨터 비전 방법론의 열개

■ 전통적 프로그래밍 vs 머신러닝

- 전통적 프로그래밍 방식
 - 개발자가 문제 해결 방법을 명시적으로 규정함.
 - 규칙과 조건을 코드로 직접 구현함.
- 머신러닝
 - 데이터 기반으로 규칙을 스스로 학습하고 일반화함.
 - 개발자는 모델과 알고리즘을 선택하여 사용.



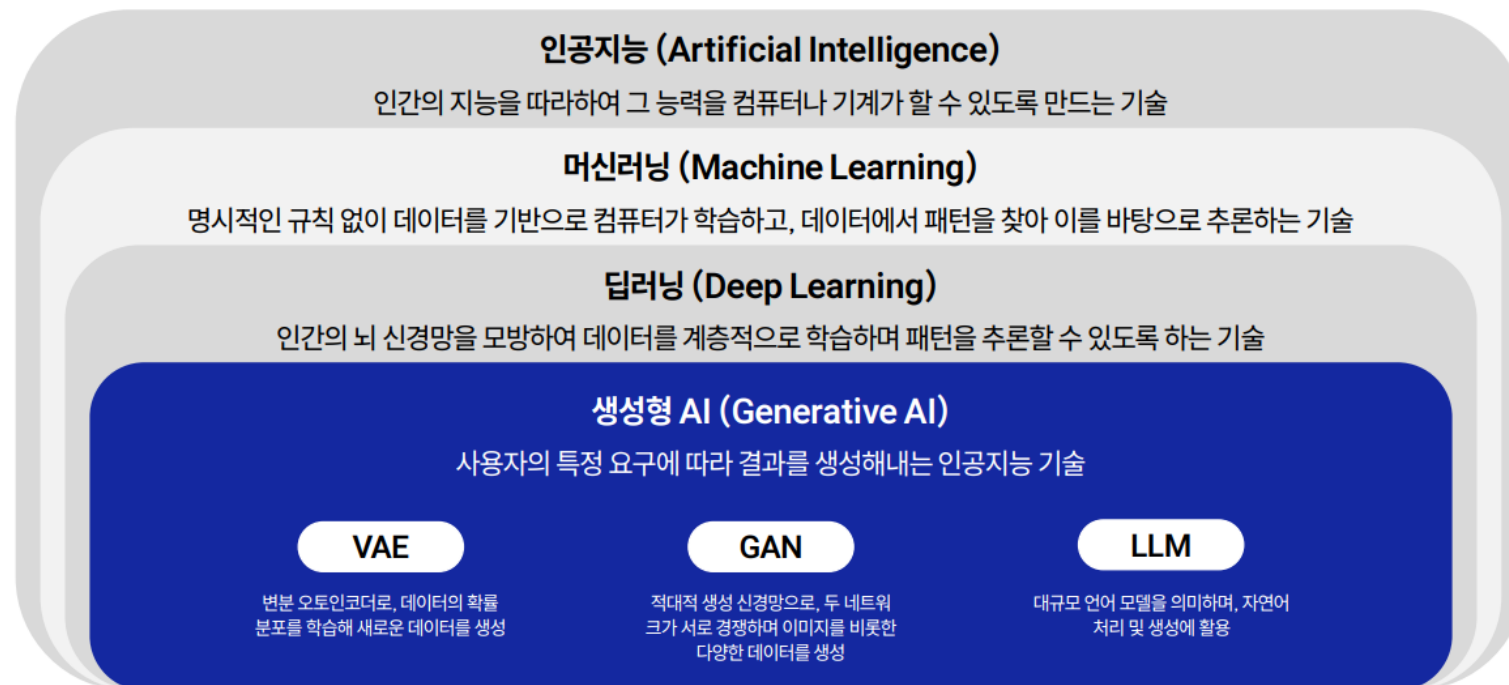
전통적 프로그래밍 방식



머신러닝

■ 인공지능의 계층적 구분

- 인공지능(AI): 인간의 지능을 흉내내도록 컴퓨터를 설계하는 기술.
- 머신러닝(ML): 명시적 규칙 없이 데이터를 기반으로 학습하는 기술.
- 딥러닝(DL): 인간의 신경망을 모방해 계층적으로 학습하는 머신러닝의 하위 영역.
- 생성형 AI(Generative AI): 주어진 조건에 따라 새로운 결과물을 생성하는 AI.



■ 머신러닝에서의 학습이란?

- 학습은 데이터를 통해 규칙이나 패턴을 스스로 찾아내는 과정.
- 머신러닝에서의 학습이란 참고할 수 있는 데이터를 바탕으로 인공지능 모델에 의해 참값이 나오도록 모델의 파라미터를 최적화 하는 과정을 의미.
- 학습의 목표는 미래의 데이터에 대해서도 높은 예측력을 갖는 모델을 만드는 것으로, 이때 과거 데이터에만 맞춘 모델은 실제 환경에서 작동하지 않을 수 있음.

■ 머신러닝의 학습 과정

- **문제 정의** : 분류, 예측, 생성 등 문제 유형을 결정하고 목표를 설정함.
- **데이터 수집** : 텍스트, 이미지, 센서 등 다양한 소스에서 데이터를 확보함.
- **데이터 전처리** : 정제, 분석, 변환 등 학습에 적합한 형식으로 가공함.
- **모델 선택** : 문제 해결에 적합한 알고리즘과 구조를 선택함.
- **모델 학습** : 훈련 데이터를 통해 패턴을 학습하고 파라미터를 최적화함.
- **모델 평가** : 테스트 데이터를 사용해 정확도, 정밀도, 재현율 등으로 성능 측정함.
- **모델 개선** : 하이퍼파라미터 튜닝, 데이터 확장 등을 통해 성능 향상.
- **예측 및 적용** : 학습된 모델을 실제 데이터에 적용하여 문제 해결 또는 의사결정 지원.

■ 머신러닝의 학습 유형

- **지도학습(Supervised Learning)** : 정답이 있는 데이터를 기반으로 학습.
(예) 이미지 분류, 가격 예측.
- **비지도학습(Unsupervised Learning)** : 정답 없이 데이터의 구조나 패턴을 학습.
(예) 클러스터링, 차원 축소.
- **자기지도학습(Self-supervised Learning)** : 일부 정보를 기준으로 나머지를 예측.
(예) 트랜스포머, 언어 모델 사전학습 등.

머신러닝 주요 용어 정리

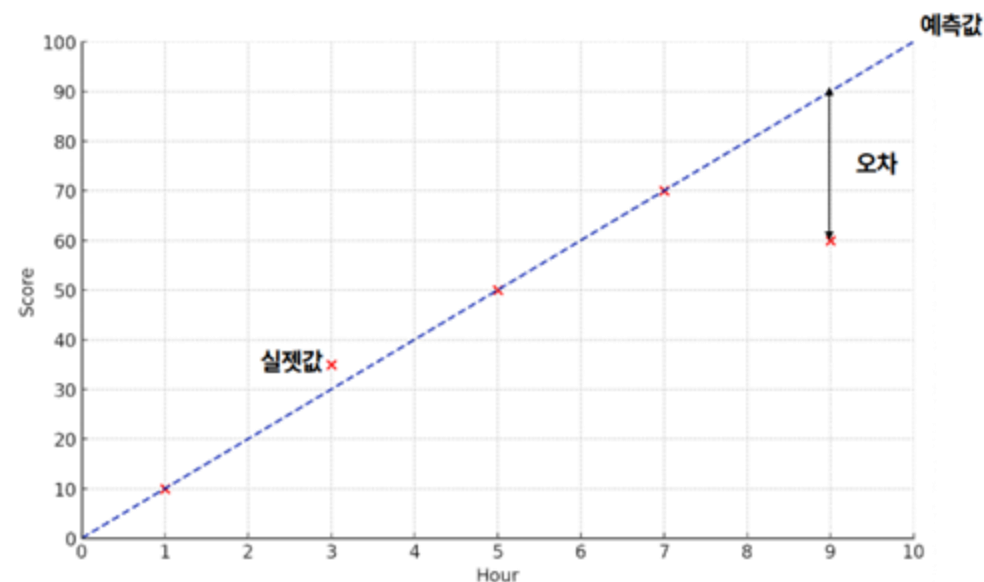
- 데이터셋(Dataset)
 - 머신러닝 모델 학습에 사용하는 데이터 모음.
 - 훈련 데이터셋(Training Dataset): 모델을 학습시키는 데 사용하는 데이터.
 - 평가 데이터셋(Test Dataset): 학습된 모델의 성능을 평가하는 데 사용하는 데이터.
- 특성(Feature)
 - 입력 데이터에서 유의미한 정보를 나타내는 속성 또는 변수.
- 레이블(Label)
 - 지도 학습에서 모델이 예측해야 하는 정답값.

※ 우측 예시처럼 ‘생존여부’가 레이블, 나머지는 특성으로 분류됨.

데이터 셋						
레이블	특성					
생존여부	구역	객실등급	성별	나이	동승자 수	운행 요금
0	A	3	male	22	1	7.25
1	B	1	female	38	1	71.2833
1	C	3	female	26	0	7.925
1	D	1	female	35	1	53.1
0	A	3	male	35	0	8.05

■ 머신러닝 주요 용어 정리 (계속)

- 실제값(True Value)
 - 관측된 참값 또는 모델이 예측해야 할 정답값.
 - 보통 훈련 데이터나 테스트 데이터에 포함된 레이블.
- 예측값(Predicted Value)
 - 학습된 모델이 새로운 데이터에 대해 출력한 값.
 - 모델 성능은 예측값과 실제값의 비교를 통해 평가함.
- 두 값의 관계
 - 머신러닝 모델은 두 값의 차이를 최소화하는 방향으로 학습.
 - 이 차이를 오차(Error)라고 하며, 성능 평가의 핵심 지표.



■ 딥러닝(Deep Learning)이란?

- 딥러닝은 인공 신경망(Artificial Neural Networks)을 기반으로 복잡한 패턴을 인식하는 기술.
- 여러 층으로 구성된 신경망 구조를 통해 비정형 데이터를 효율적으로 분석.
- 이미지, 음성, 자연어 처리에서 뛰어난 성능을 보임.
- 사람의 뇌처럼 뉴런(Neuron)이 연결되어 층(Layer)을 구성함.

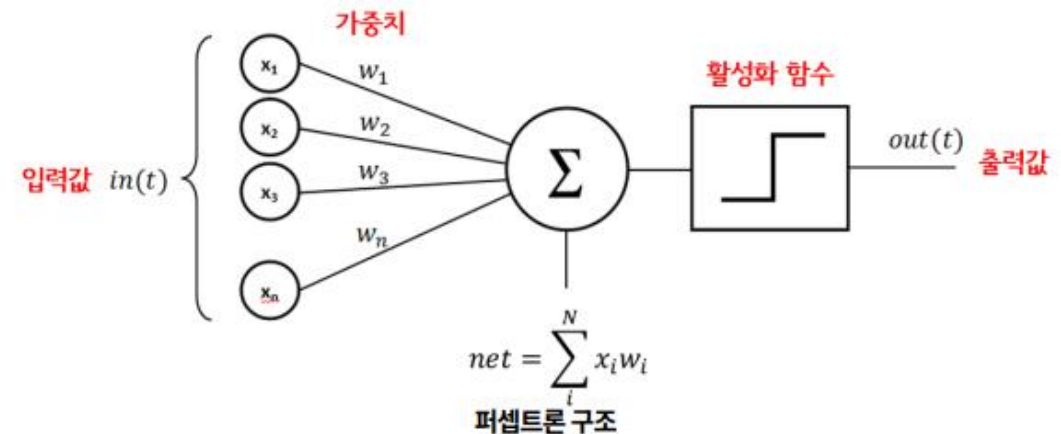
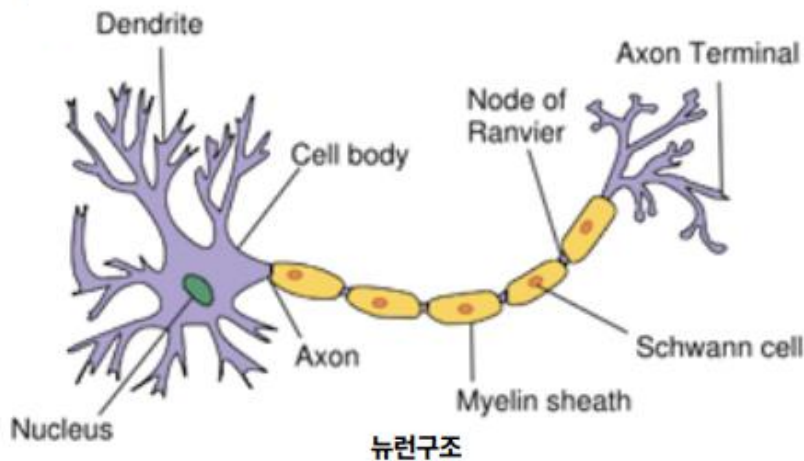
전통적인 머신러닝과 딥러닝의 비교

비교 항목	전통적인 머신러닝 (Traditional Machine Learning)	딥러닝 (Deep Learning)
정의	학습을 기반으로 하는 대부분의 AI 모델 생성 기법. 일반적으로 특징 추출은 수동으로 전처리하고 모델은 추출된 특징의 패턴과 관계를 찾는 것을 담당함	다양한 구조의 인공 신경망을 다층 구조로 사용하여 데이터 특징 추출, 패턴 및 관계 파악 등의 전반적인 프로세스를 모델에서 처리하는 머신러닝의 하위 분야
데이터 종류	비교적 소량의 정형 데이터	대량의 비정형 데이터
특징 추출	전처리를 통해 수동으로 데이터에서 중요한 특징(변수)을 추출	모델이 스스로 데이터를 분석하고, 특징을 자동으로 추출함
구조	비교적 간단한 구조의 알고리즘 (예: 의사결정 트리, SVM, KNN)	다층 인공 신경망을 활용한 복잡한 구조의 알고리즘 사용
모델 복잡성	상대적으로 단순한 알고리즘을 사용	수백, 수천 개의 뉴런으로 구성된 매우 복잡한 신경망 사용
요구 자원	비교적 적은 컴퓨팅 자원 필요	고성능 GPU 및 많은 컴퓨팅 자원이 요구됨
해석 가능성	상당수의 모델이 최소한의 해석이 가능함. 일부는 구체적인 해석이 가능함.	신경망 구조가 복잡하여 모델 해석이 어려움
응용 분야	가격 예측, 고장 예측, 등급 분류 등	자율 주행, 이미지 인식, 음성 인식, 자연어 처리 등 고차원 문제 해결

■ 인공 신경망 기본 구조

- 인공 신경망(ANN)은 사람의 뇌 신경망 구조를 모방한 컴퓨팅 모델.
- 입력을 받아 가중치를 적용해 출력값을 생성함.
- 퍼셉트론(Perceptron)은 1957년 제안된 초기 형태로, 단일 층 구조를 가짐.
- 각 입력값은 가중치와 곱해진 후 합산되고, 활성화 함수를 통해 출력이 결정됨.

※ 아래 그림은 생물학적 뉴런 구조와 퍼셉트론 수학적 모델을 비교한 예시.

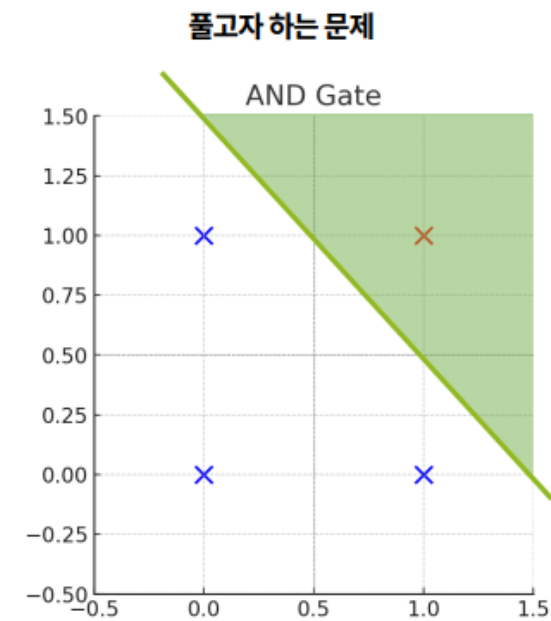
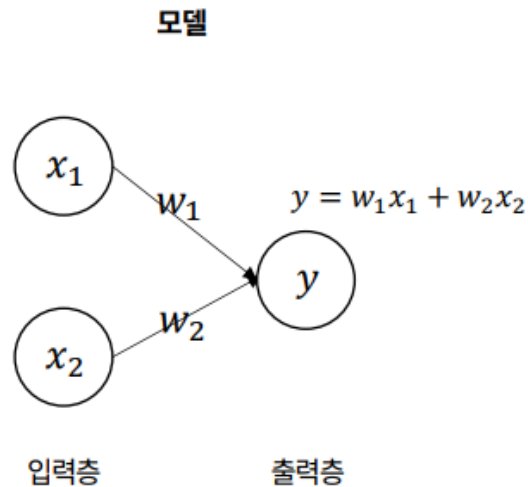


■ 인공 신경망 - 단층 퍼셉트론(AND 문제)

- 단층 퍼셉트론(Single Layer Perceptron)은 입력층과 출력층만으로 구성됨.
- 선형 분리 가능한 문제만 해결 가능.
- AND 게이트처럼 직선으로 두 클래스를 나눌 수 있는 경우에만 학습 가능.

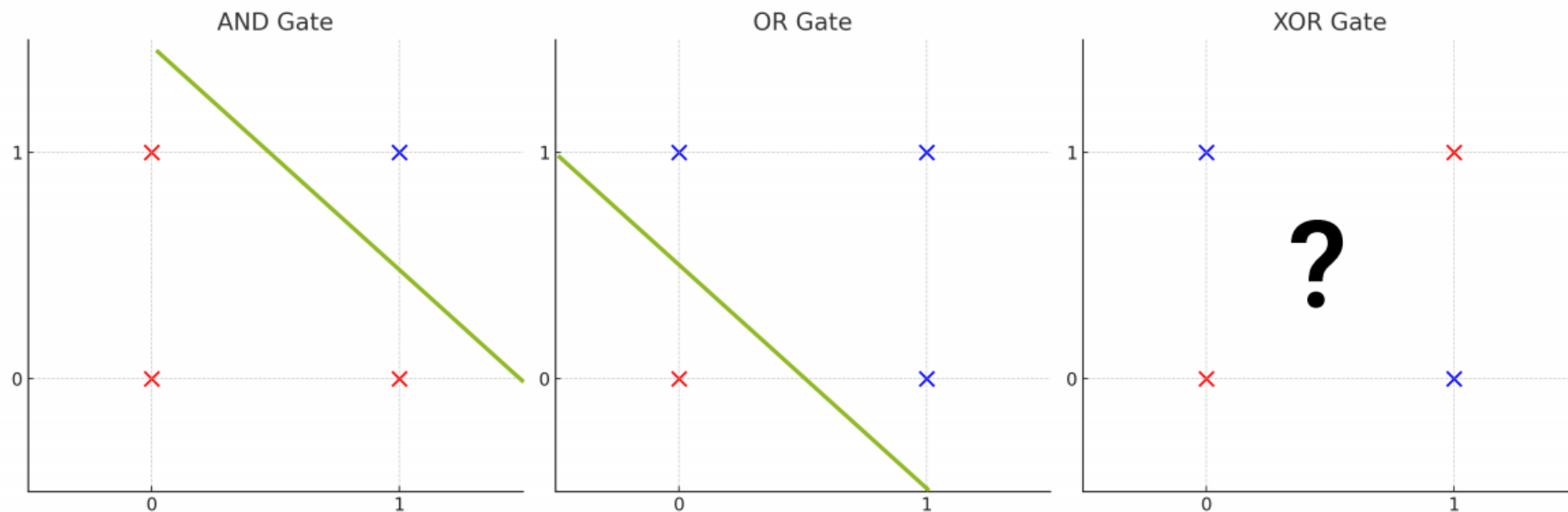
데이터세트

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1



■ 인공 신경망 - 단층 퍼셉트론의 한계

- 단층 퍼셉트론은 AND, OR 게이트처럼 선형 분리가 가능한 문제는 해결 가능.
- 그러나 XOR 문제처럼 비선형 문제는 해결할 수 없음.
- 이는 단층 구조의 표현력 한계를 의미함.

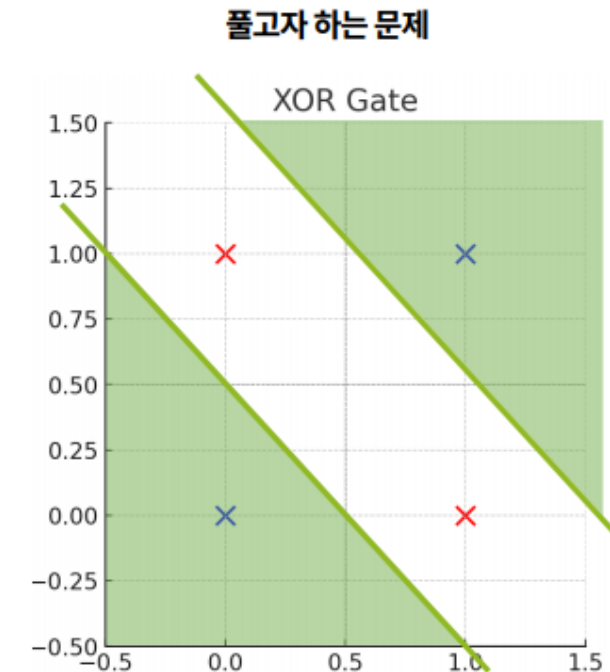
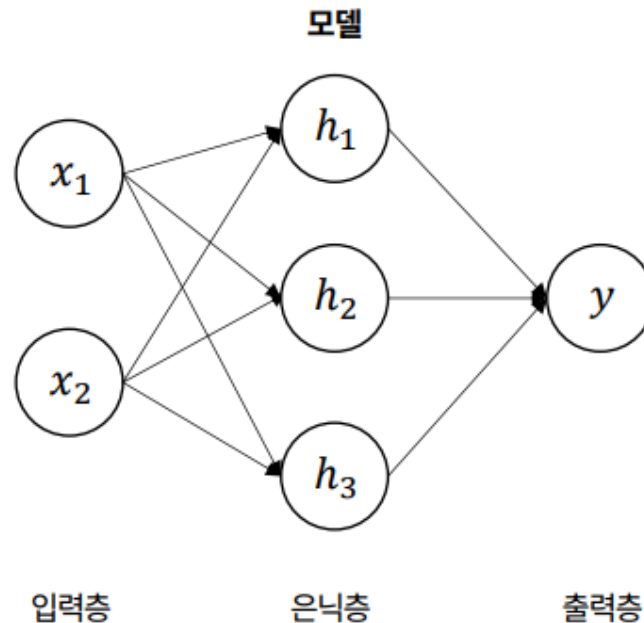


■ 인공 신경망 - 다층 퍼셉트론(Multilayer Perceptron, MLP)

- MLP는 입력층, 하나 이상의 은닉층, 출력층으로 구성됨.
- 은닉층이 추가되면서 비선형 분류 및 복잡한 함수 근사가 가능해짐.
- XOR 문제처럼 단층 퍼셉트론이 해결하지 못하는 문제도 처리 가능.

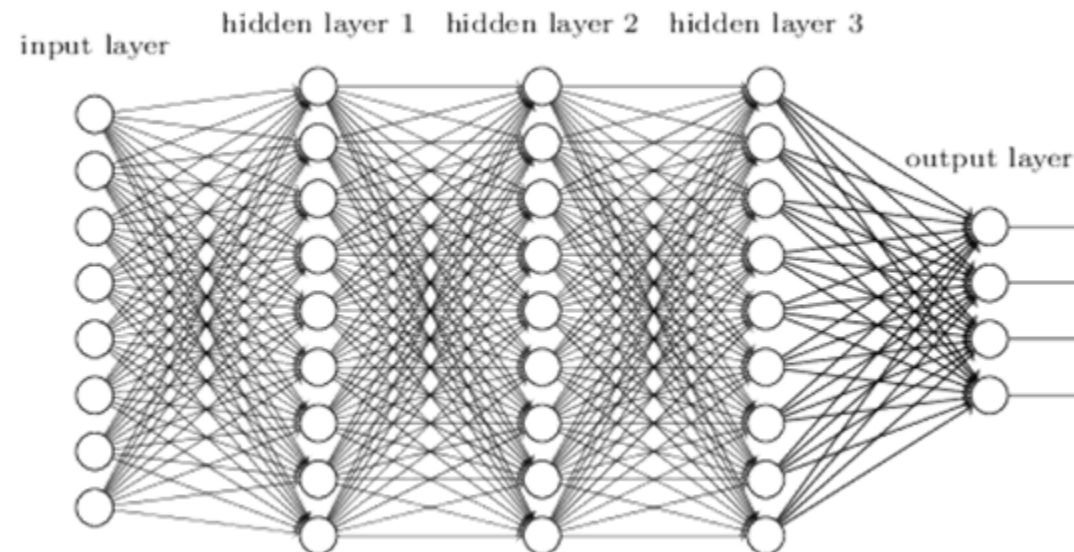
데이터세트

x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0



■ 심층 신경망(Deep Neural Network)

- 심층 신경망(DNN)은 딥러닝에서 사용되는 대표적인 모델.
- 입력층, 여러 은닉층, 출력층으로 구성되고 각 층은 뉴런 단위로 구성.
 - 입력층: 데이터를 받아들임.
 - 은닉층: 특성 추출 및 패턴 학습.
 - 출력층: 최종 예측 결과 출력.
- 가중치와 편향을 통해 데이터를 처리하여 복잡한 데이터 분류 및 회귀 문제 해결에 효과적임.



■ 심층 신경망(DNN)의 학습과정 요약

- 심층 신경망의 학습과정은 아래와 같은 총 6단계로 이루어져있음.
- 6가지의 단계를 거치고 나면 이를 한 에폭(Epoch)이라고 하며 한 에폭은 모든 학습 데이터를 한번씩 모델에 통과시킨 과정 전체를 의미.

[1단계] **순전파 (Forward Propagation)** : 입력 데이터를 받아 가중치와 편향을 이용해 출력을 계산함.

[2단계] **출력층 처리 (Output Layer)** : 문제 유형(회귀, 이진/다중 분류)에 맞게 최종 예측값 생성함.

[3단계] **손실 계산 (Loss Calculation)** : 예측값과 실제 정답(label) 간의 차이를 수치로 계산함.

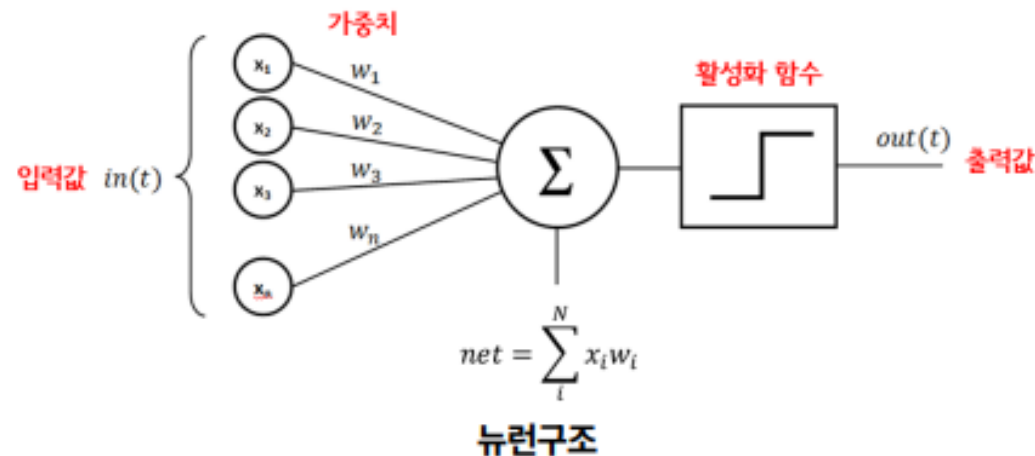
[4단계] **역전파 (Backpropagation)** : 손실 값 기준으로 각 가중치가 얼마나 손실에 영향을 줬는지 미분을 통해 계산함.

[5단계] **기울기 계산 및 경사 하강법 (Gradient Calculation & Descent)** : 계산된 미분값을 기반으로 가중치와 편향을 어느 방향으로 업데이트할지 결정함.

[6단계] **옵티마이저 적용 (Optimizer Update)** : 학습률과 기울기 조정을 통해 파라미터를 실제로 업데이트.

■ [1단계] 순전파 (Forward Propagation)

- 순전파는 입력 데이터를 받아 뉴런을 거쳐 출력 결과를 생성하는 과정.
- 입력값은 각 입력별 가중치(weight)를 곱해 뉴런으로 전달됨.
 - 가중치는 입력의 중요도를 의미하며 학습을 통해 조정됨.
 - 전달된 가중합에 편향(bias)을 더하고, 비선형 활성화 함수를 적용함.
- 활성화 함수는 복잡한(비선형) 패턴을 학습할 수 있도록 변환 기능 수행.



▪ [2단계] 출력층 처리 (Output Layer)

- 출력층은 이전 층의 출력을 입력으로 받아 예측값을 생성함.
- 분류 문제는 확률 값으로, 회귀 문제는 연속적인 수치로 표현됨.
- 출력층의 활성화 함수는 문제 유형에 따라 선택됨.
 - 회귀: 활성화 함수 없이 가중합 그대로 출력.
 - 이진 분류: 시그모이드 함수 사용.
 - 다중 클래스 분류: 소프트맥스 함수 사용 (PyTorch에서는 손실 함수에 포함되기도 함).
- 출력층 결과는 최종 예측값으로 해석 가능.

▪ [3단계] 손실 계산 (Loss Calculation)

- 손실 함수는 예측값과 실제값(label)의 차이를 계산해 모델 성능을 수치화함.
- 손실이 클수록 예측이 부정확함을 의미함.
- 학습 과정에서는 손실을 줄이는 방향으로 가중치와 편향을 업데이트함.
- 손실 함수는 문제 유형에 따라 다르게 설정됨.
 - 회귀: MSE(평균 제곱 오차), MAE(평균 절대 오차) 사용.
 - 분류: 교차 엔트로피(Cross Entropy) 사용.

■ [4단계] 옵티마이저 (Optimizer)

- 옵티마이저는 파라미터를 효율적이고 안정적으로 업데이트(손실을 최소화)하기 위한 알고리즘을 의미.
- 경사 하강법(Gradient Decent)은 손실을 최소화하기 위한 옵티마이저의 기본 개념.
- 손실 함수의 기울기를 계산해 그 반대 방향으로 파라미터 조정.
- 미분값이 작은 방향으로 이동하며 최소값(local minimum)에 수렴함.
- 학습률(learning rate)은 변화량 조절에 사용됨.

$$w_{new} = w_{old} - \eta \cdot \frac{\partial L}{\partial w_{old}}$$

이전 가중치

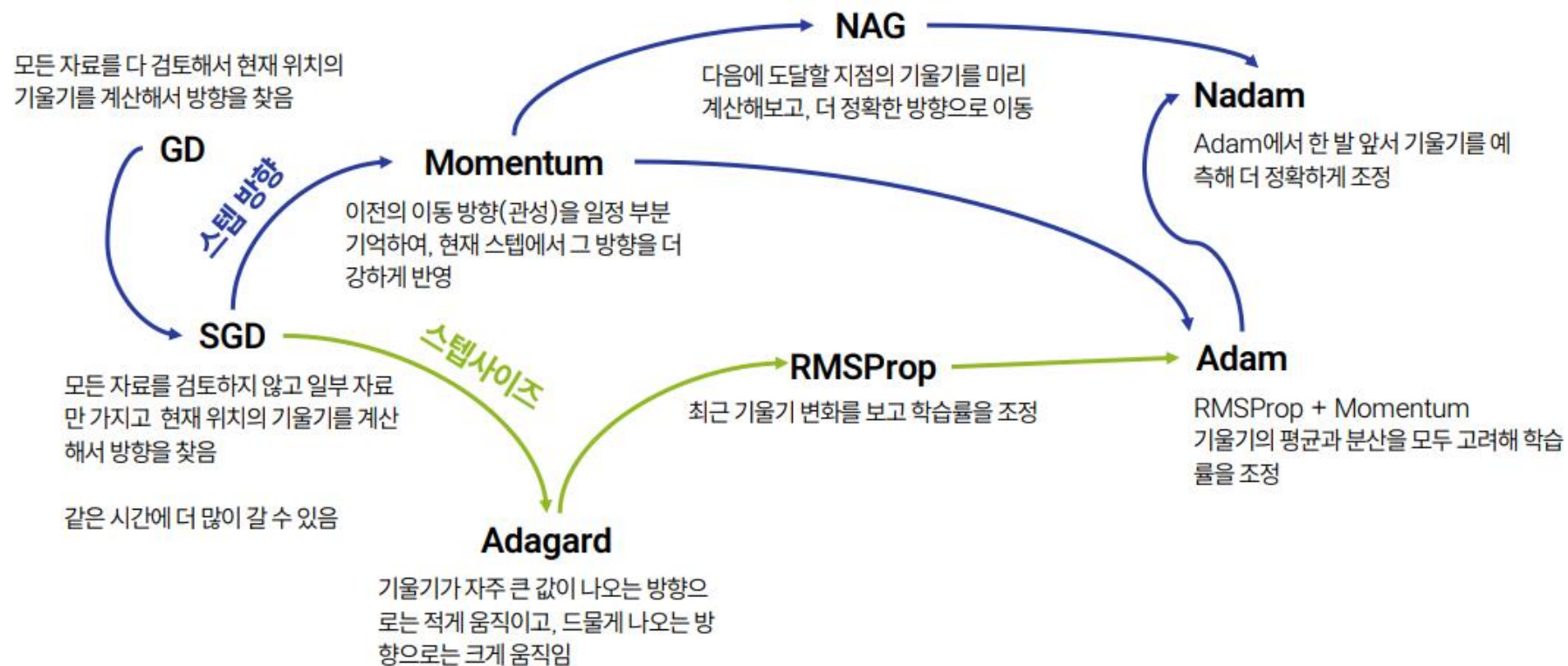
손실 함수 L에 대한 가중치 w의 기울기

업데이트된 가중치

학습률(Learning Rate)

■ [4단계] 옵티마이저 (Optimizer) 비교

- 기본 경사 하강법을 개선하여 학습 속도와 수렴 안정성을 향상시킴.



▪ [4단계] 옵티마이저 (Optimizer) 비교

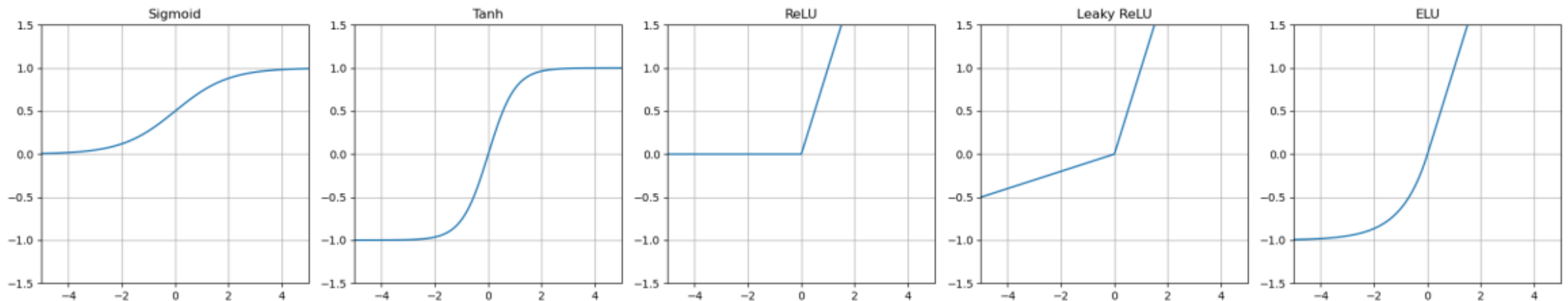
옵티마이저	설명	특징	장점	단점
GD	전체 데이터에 대해 기울기를 계산해 파라미터 업데이트	정확한 방향으로 최적화	전역 최적화 가능, 수렴 안정적	계산 비용이 매우 큼, 대규모 데이터셋에서 느림
SGD	하나의 데이터 포인트에 대해 기울기를 계산해 파라미터 업데이트	빠른 업데이트, 노이즈 포함	계산 비용이 적음, 빠른 학습	기울기의 노이즈로 불안정한 수렴 가능
Momentum	이전 기울기를 기억해 업데이트 시 관성처럼 활용	과거 기울기를 반영해 가속화	빠른 수렴, 진동 억제	적절한 모멘텀 값 설정 필요
NAG	미리 한 발 앞서 기울기를 계산해 더 정확한 방향으로 이동	기울기를 미리 예측	빠르고 정확한 수렴	기울기 계산 추가로 복잡성 증가
Nadam	Adam과 NAG를 결합한 방식	NAG의 예측과 Adam의 적응형 학습률 결합	빠른 수렴과 안정성	비교적 복잡하고 계산량이 많음
AdaGrad	각 파라미터마다 적응형 학습률 적용	자주 업데이트되는 파라미터는 학습률 작아짐	드물게 업데이트되는 파라미터에는 큰 학습률	학습이 진행될수록 학습률이 너무 작아짐
RMSProp	최근 기울기의 제곱 평균을 사용해 학습률 조정	기울기 변화에 따른 학습률 조정	기울기 진동을 억제하고 빠른 수렴	적절한 학습률 설정이 필요
Adam	모멘텀과 RMSProp의 결합, 1차와 2차 모멘트를 모두 추적	적응형 학습률과 모멘텀 적용	빠른 수렴, 안정적인 학습	하이퍼파라미터 설정이 중요, 과적합 가능

▪ [5단계] 역전파 (Backpropagation)

- 역전파는 출력층에서 계산된 손실을 기준으로 오차를 거꾸로 전파하는 과정.
- 각 층의 가중치가 손실에 얼마나 기여했는지 기울기로 계산함.
- 연쇄 미분법(Chain Rule)을 이용해 각 층별로 미분값 계산.
- 계산된 기울기를 바탕으로 경사 하강법 등 최적화 알고리즘으로 파라미터 업데이트 진행.

■ [6단계] 활성화 함수 (Activation Function)

- 활성화 함수는 선형 결합 결과에 비선형성을 부여함.
- 각 층을 거치면서 누적된 비선형성이 모델의 복잡한 표현 가능성 확보.
- 주요 함수:
 - Sigmoid: 0~1 범위로 출력. 현재는 주로 출력층에서 사용됨.
 - Tanh: -1~1 범위. 출력 중심이 0에 가까움.
 - ReLU: 은닉층에서 가장 널리 사용됨.
 - Leaky ReLU, ELU: ReLU의 단점을 보완한 함수들.



Thank you!