
아롬 AI 예비대학 OT

2026 아롬 17기 AI 예비대학

Department of Intelligent Mechatronics Engineering

Ji Jun Gu

jun9u20@gmail.com





01. Intro

아롬 AI 예비대학의 목표 및 커리큘럼

01. Intro

예비대학의 목표

1. 이미지 분류의 가장 대표적인 입문 문제 경험

컴퓨터비전 학습에서 가장 널리 사용되는 기초 문제를 통해 이미지 분류의 기본 개념과 전체 흐름을 이해

2. CV 맛보기 + 활용하기

이론적 개념이 실제 성능 향상으로 이어지는 과정을 직접 경험

3. 대학생에게 가장 친숙한 술 문화를 바탕으로 대학 생활의 흐름을 이해

~~아름에서 가장 널리 사용되는 술 문화를 통해 대학 생활의 기본 개념과 전체 흐름을 이해~~



익숙한 사례를 통해 복잡한 개념을 자연스럽게 이해하는 것을 목표로 함

01. Intro

커리큘럼

주차	내용	진행 방식	진행자
1-1	[이론] OT / 딥러닝 개요(경사하강법, 오차역전파, 활성화 함수, 손실 함수, 회귀) / 프레임워크 소개	대면	지준구
1-2	[코딩] Kaggle 데이터 불러오기 / 주요 모듈 소개 / torch.tensor / pytorch로 회귀 구현	비대면	조치연
2-1	[이론] CNN 기초(합성곱 연산, 패딩, 스트라이드)	대면	서동주
2-2	[코딩] pytorch로 CNN 이해 / pytorch로 학습 메커니즘 구현1	비대면	조치연
3-1	[코딩] pytorch로 학습 메커니즘 구현2 / 예측	대면	서동주
3-2	[코딩] 최종 복습	비대면	조치연
4-1	[프로젝트] 팀 구성 및 아이디어(데이터셋) 선정 및 계획	대면	이정연
4-2	[프로젝트] 팀별 진행	팀별	이정연
5-1	[프로젝트] 팀별 진행	팀별	이정연
5-2	[프로젝트] 발표 및 회식	대면	이정연

01. Intro

진행 방식

스터디 정보

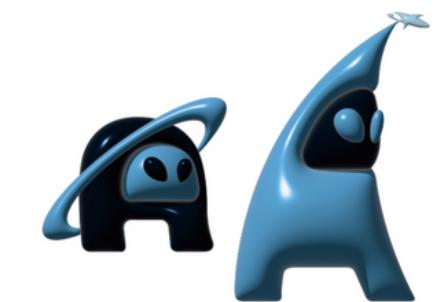
- 온·오프라인 병행
- 매주 2회차 수업으로 구성

수업 구성

- 1회차 (대면) - 학교 강의실에서 이론 중심 강의 진행
- 2회차 (비대면) - Discord 를 통해 실습 중심 코딩 세션 진행

강의 자료

- 멘토진 제작 강의 자료 (PPT)
- Google Colab 실습 노트북



01. Intro

멘토단 소개

김기현

AI로봇학과 25학번

서동주

경제학과 23학번

조치연

지능기전공학과 23학번

지준구

지능기전공학부 무인이동체학전공 20학번

이정연

데이터사이언스학과 23학번





02. 딥러닝 학습의 기본 구조

경사하강법, 오차역전파, 활성화 함수, 손실함수, 회귀

02. 딥러닝 학습의 기본 구조

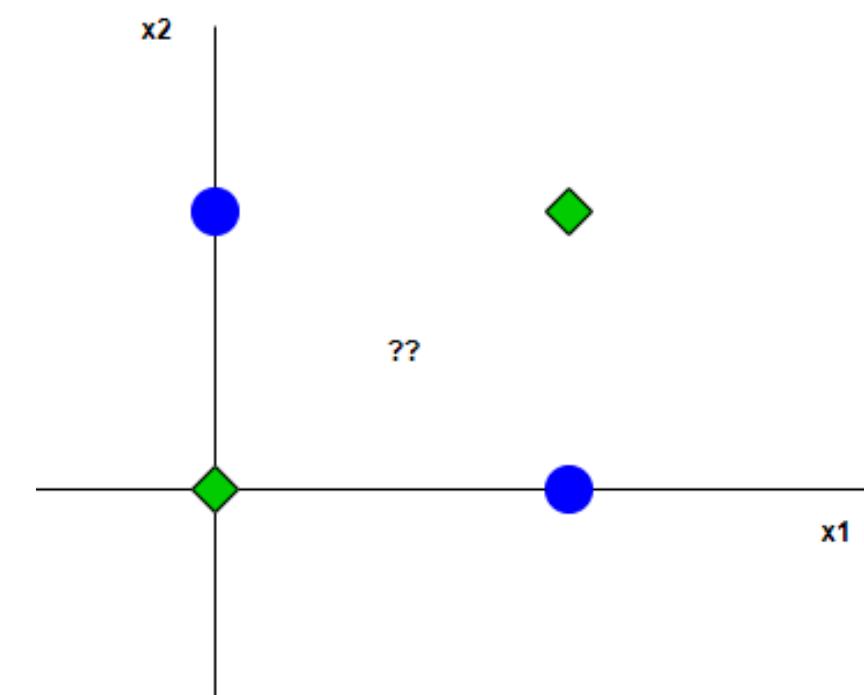
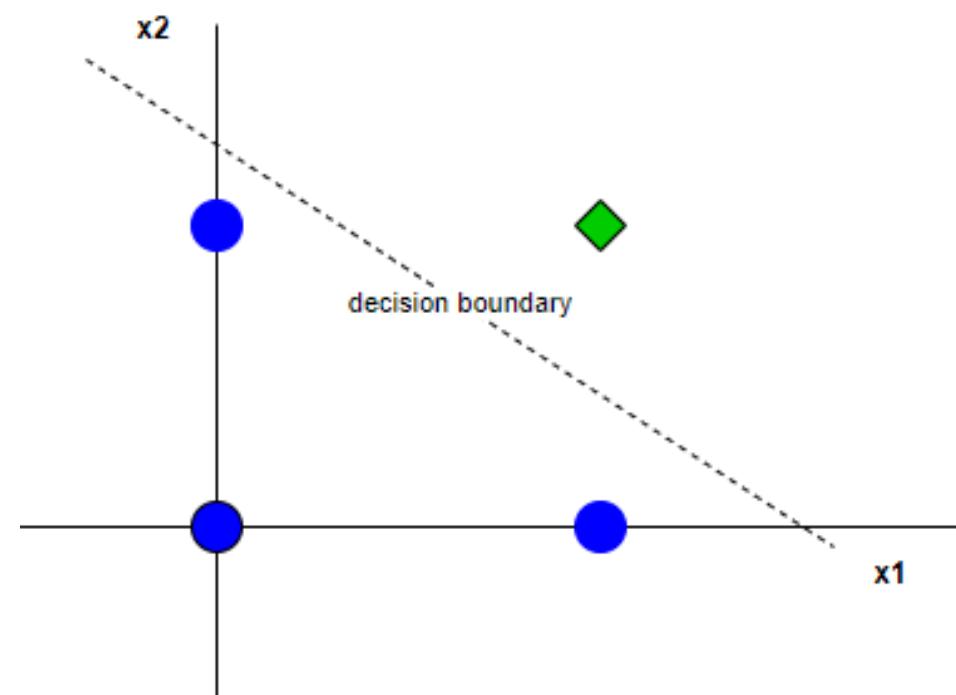
선형 모델의 한계

선형(Linear) 모델이란?

아무리 복잡한 데이터를 넣어도, 직선 하나로만 맞추려고 하는 모델

문제점 ①

선형 모델은 직선으로만 데이터를 구분할 수 있다.



02. 딥러닝 학습의 기본 구조

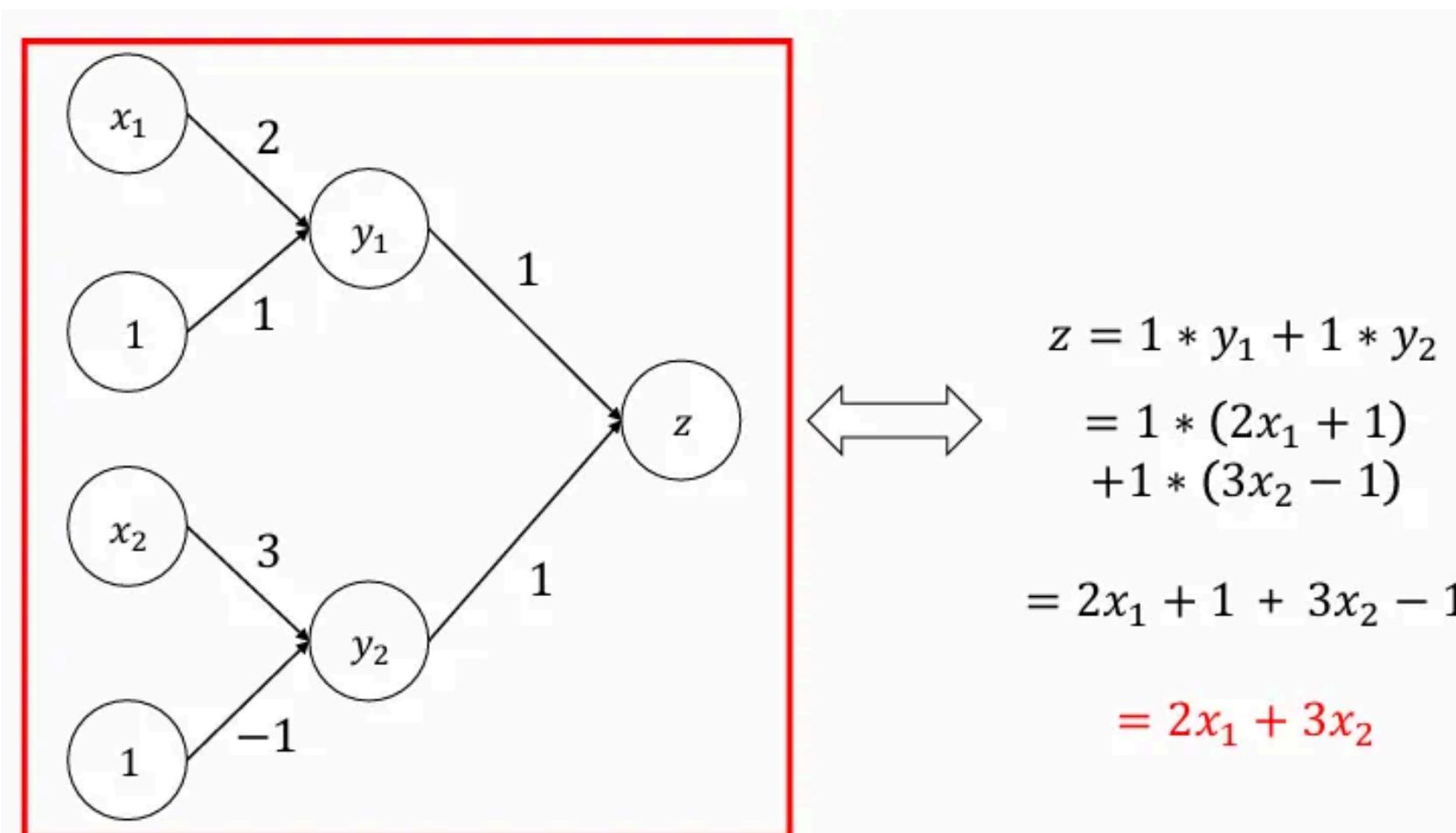
선형 모델의 한계

선형(Linear) 모델이란?

아무리 복잡한 데이터를 넣어도, 직선 하나로만 맞추려고 하는 모델

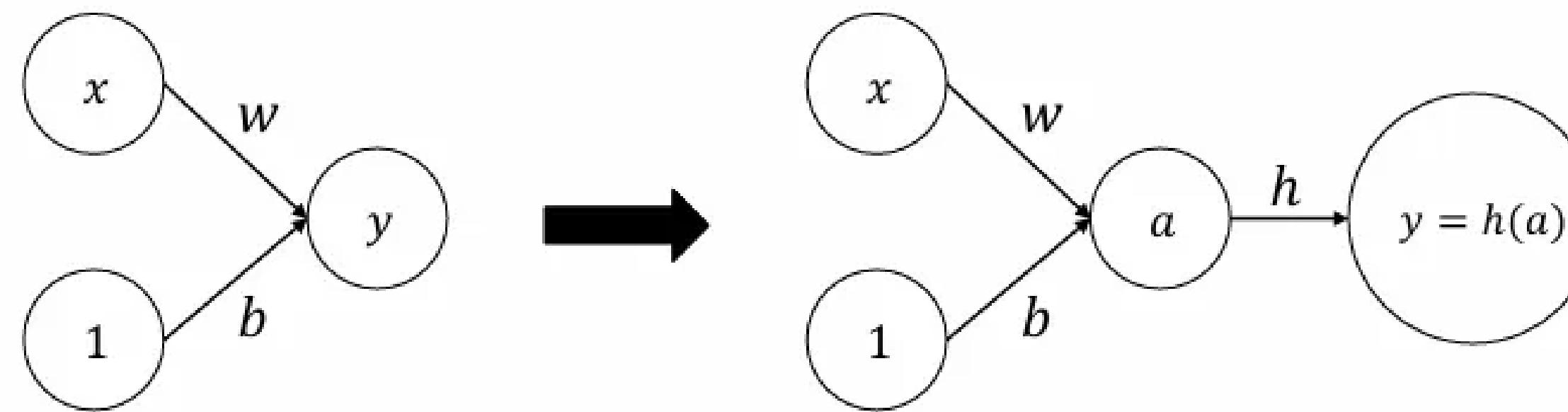
문제점 ②

겉보기엔 복잡해 보여도 실제로는 직선 하나로 계산하는 것과 같다.



02. 딥러닝 학습의 기본 구조

선형 모델의 한계



곱하고 더하는 계산만 반복하면 표현력이 제한되기 때문에 중간 결과를 변형하는 과정이 필요

02. 딥러닝 학습의 기본 구조

활성화 함수(activation function)

활성화 함수(activation function)란?

모델이 계산한 값을 다음 단계로 그대로 보낼지, 어떻게 바꿔서 보낼지 결정하는 함수

왜 필요한가?

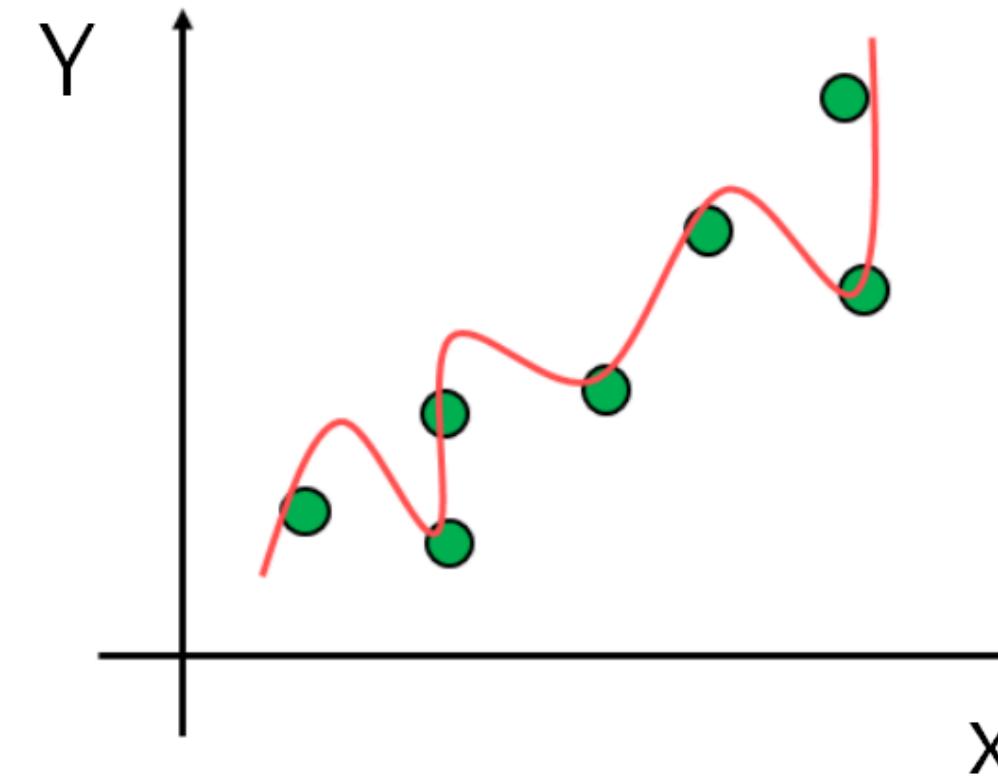
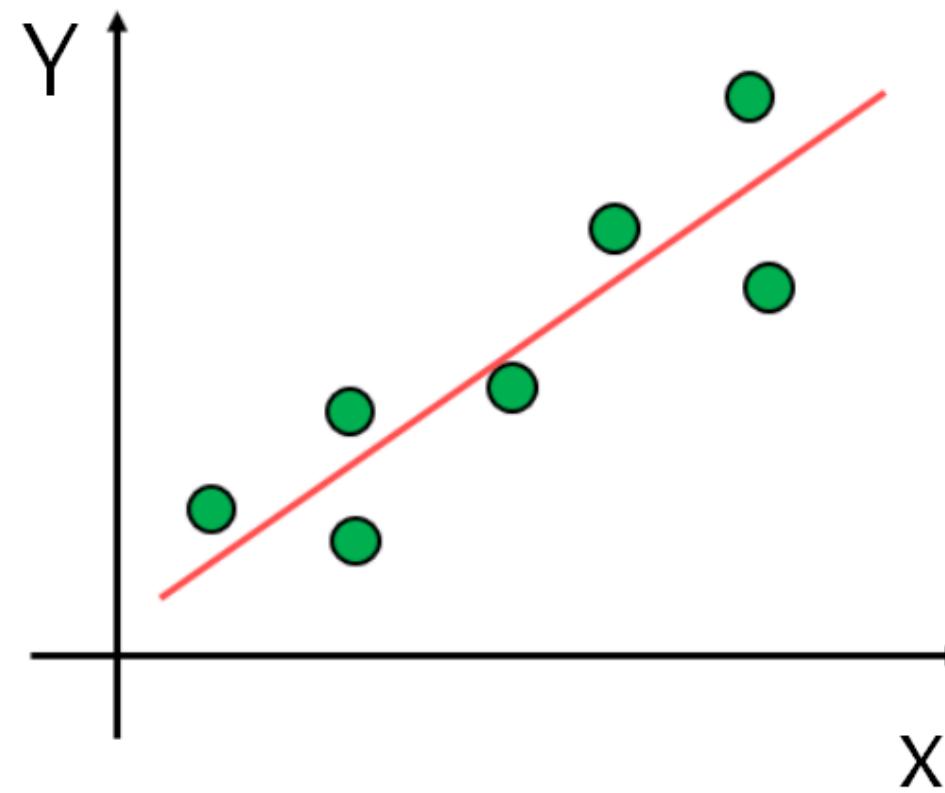
- 활성화 함수가 없으면 모델은 **복잡한 패턴을 학습할 수 없음**
- 활성화 함수는 모델에 **비선형성(곡선적인 판단)**을 추가함
- 이를 통해 **복잡한 이미지 특징을 학습 가능**

직관적 이해

- “이 정보가 중요한지 아닌지”를 판단하는 스위치 역할
- 어떤 값은 강조하고, 어떤 값은 약하게 전달

02. 딥러닝 학습의 기본 구조

활성화 함수(activation function)



계산 결과를 그대로 전달하면 표현력이 제한되지만, 중간에서 값을 변형하면 더 복잡한 관계를 학습할 수 있다.

02. 딥러닝 학습의 기본 구조

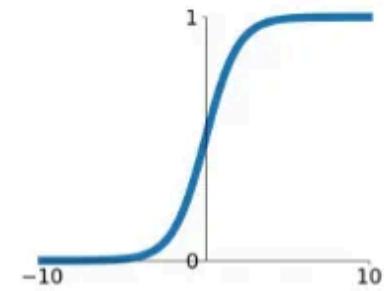
활성화 함수(activation function)

대표적인 활성화 함수들

Activation Functions

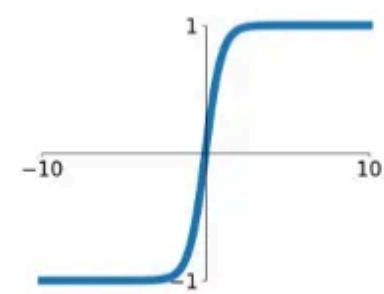
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



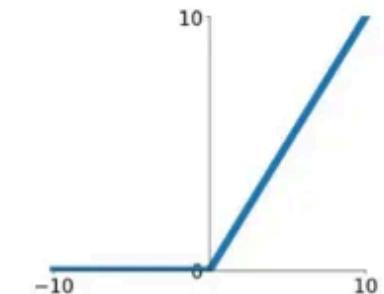
tanh

$$\tanh(x)$$



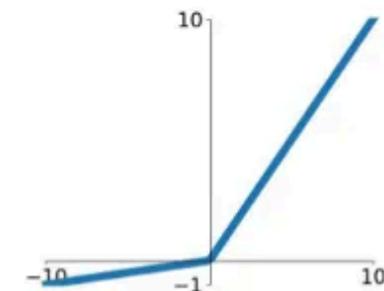
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$



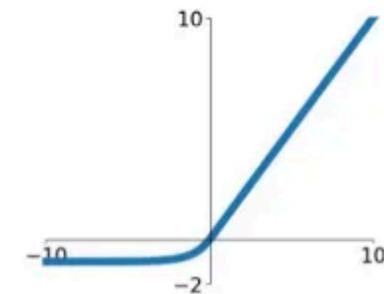
Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

- 어떤 함수는 출력 범위를 제한해 안정성을 높이고,
- 어떤 함수는 기울기를 유지해 학습을 빠르게 하며
- 또 어떤 함수는 더 복잡한 패턴을 유연하게 표현함

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



02. 딥러닝 학습의 기본 구조

회귀(regression)

회귀(regression)란?

입력 데이터를 바탕으로 하나의 숫자 값을 예측하는 문제 유형

어떤 문제를 말하는가?

- 시험 점수 예측
- 집값·가격 예측
- 온도·수치 예측



출력값이

숫자 하나로 나오는 것이 특징

왜 중요한가?

모델이 틀렸는지, 얼마나 틀렸는지를 숫자로 바로 확인할 수 있음

02. 딥러닝 학습의 기본 구조

분류(classification)

분류(classification)란?

입력 데이터를 바탕으로 정해진 여러 범주 중 하나를 선택하는 문제 유형

어떤 문제를 말하는가?

- 스팸 메일 분류 (스팸 / 정상)
- 이미지 분류 (개 / 고양이)
- 감정 분류 (긍정 / 부정)



출력값이 ‘라벨’로 나오는 것이 특징

왜 중요한가?

- 컴퓨터비전에서 가장 많이 다루는 문제 유형
- 실제 서비스에서 활용도가 매우 높음
- 회귀와 달리 맞았는지 / 틀렸는지를 명확히 판단 가능

02. 딥러닝 학습의 기본 구조

손실함수(loss function)

손실함수(loss function)란?

모델의 예측이 정답에서 얼마나 벗어났는지를 숫자로 알려주는 기준

왜 필요한가?

- 모델은 스스로 “잘했는지, 못했는지”를 알 수 없음
- 손실함수는 모델에게 지금 답이 얼마나 틀렸는지 알려주는 역할을 함
- 학습은 이 숫자를 줄이는 방향으로 진행됨

아이디어

- 예측이 맞을수록 손실값은 작아짐
- 예측이 틀릴수록 손실값은 커짐
- 목표는 손실값을 점점 줄이는 것



손실값을 기준으로 **조금씩 더 나은 방향으로
모델을 수정**

02. 딥러닝 학습의 기본 구조

손실함수(loss function)

손실함수(loss function)란?

모델의 예측이 정답에서 얼마나 벗어났는지를 숫자로 알려주는 기준

회귀 문제에서는

예측한 값이 정답에서 얼마나 멀리 떨어졌는지를 숫자로 확인

분류 문제에서는

정답일 확률을 얼마나 높게 예측했는지를 기준으로 학습

02. 딥러닝 학습의 기본 구조

오차역전파(backpropagation)

오차역전파(backpropagation)란?

모델이 얼마나 틀렸는지(손실)를 기준으로, 그 원인이 어느 부분에 있는지 거꾸로 따라가며 각 파라미터를 조금씩 수정하는 방법

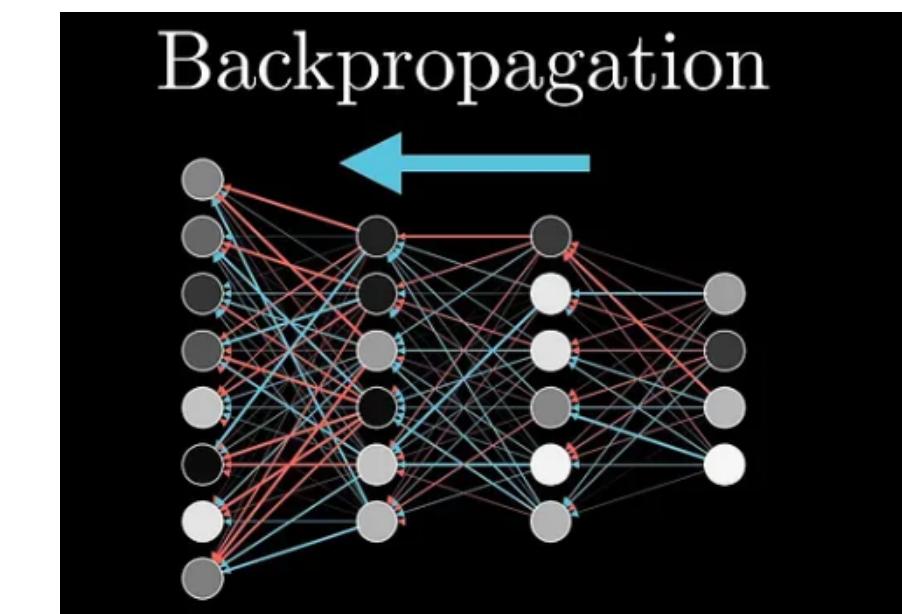
*파라미터(parameter): 모델이 학습을 통해 값을 스스로 조정하는 변수로, 입력 데이터를 어떻게 처리하고 판단할지를 결정하는 기준

왜 필요한가?

- 딥러닝 모델은 여러 층으로 이루어져 있음
- 결과만 보고는 어디를 얼마나 고쳐야 할지 알 수 없음
- 오차역전파를 통해 각 층이 얼마나 책임이 있는지 계산

직관적 이해

- 시험을 보고나서
 - 틀린 문제를 확인
 - 어디서 실수했는지 거꾸로 돌아가며 복습
- 오차역전파도
 - 결과의 오차를 거꾸로 전달해 수정



틀린 정도를 뒤에서부터 전달해
모델을 고치는 방법

02. 딥러닝 학습의 기본 구조

경사하강법(gradient descent)

경사하강법(gradient descent)이란?

손실 함수(Loss)를 기준으로, 손실이 가장 작아지는 방향으로 파라미터를 반복적으로 업데이트하는 최적화 방법

왜 필요한가?

- 딥러닝 모델은 수많은 파라미터를 가짐
- 모든 경우를 직접 계산해 최적값을 찾는 것은 현실적으로 불가능
- 경사하강법을 통해 조금씩 더 나은 방향으로 학습을 진행

아이디어

기울기(Gradient)는 함숫값이 가장 빠르게 증가하는 방향을 의미함
→ 기울기의 반대 방향으로 이동하면 함숫값이 감소함



기울기의 반대 방향으로 **얼마나 이동할지(학습률)**를 결정함



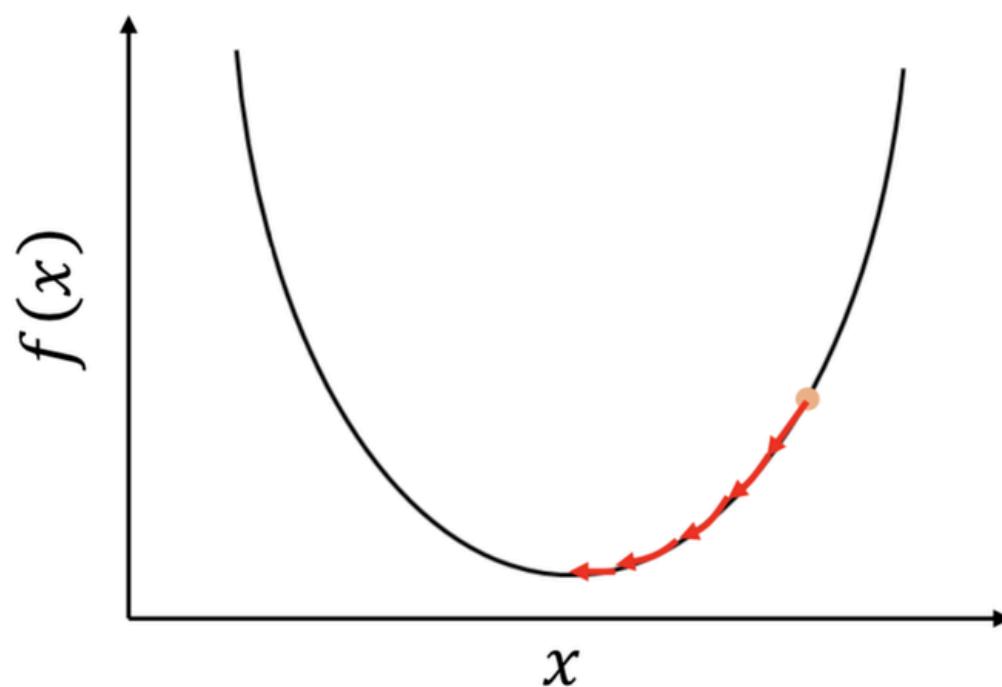
반복하다 보면 함수의 최저점을 찾을 수 있음

02. 딥러닝 학습의 기본 구조

경사하강법(gradient descent)

경사하강법(gradient descent)이란?

손실 함수(Loss)를 기준으로, 손실이 가장 작아지는 방향으로 파라미터를 반복적으로 업데이트하는 최적화 방법



경사를 따라 여러 번의 스텝을 통해 최적점으로 다가간다.

경사는 기울기(미분, Gradient)를 이용해 계산한다.

02. 딥러닝 학습의 기본 구조

활성화 함수

“모델이 어떻게 판단 구조를 갖는가”

손실함수

“얼마나 틀렸는지 어떻게 재는가”

오차역전파

“누가 얼마나 잘못했는지 어떻게 아는가”

경사하강법

“그래서 어떻게 고치는가”



03. 컴퓨터비전이란?

CV의 개념과 프레임워크

03. 컴퓨터비전이란?

CV의 개념

컴퓨터 비전(Computer Vision)이란?

디지털 이미지와 비디오에서 의미 있는 정보를 추출하고 해석하는 인공지능 기술 분야

무엇을 할 수 있는가?

- 이미지 분류: 사진 속 대상이 무엇인지 판단
- 객체 인식: 이미지 속 여러 객체를 찾아내고 구분
- 얼굴 인식, 자율주행, 의료 영상 분석 등 다양한 분야에 활용

왜 중요한가?

- 이미지와 영상은 가장 많이 생성되는 데이터 형태
- 컴퓨터비전은 현실 세계를 AI가 이해하는 핵심 기술
- 최근 딥러닝(CNN)의 발전으로 성능이 급격히 향상됨

03. 컴퓨터비전이란?

프레임워크(Framework)

프레임워크(Framework)란?

모델을 처음부터 직접 구현하지 않고, 학습·계산·최적화 과정을 체계적이고 효율적으로 구성할 수 있도록 도와주는 도구

프레임워크가 없던 시절(혹은 순수 Python/C++로 구현하던 시절)에는 개발자가 다음 과정을 직접 처리

1. 수학적 유도

손실 함수(Loss Function)에 대한 가중치의 기울기(Gradient)를 미분 공식을 써서 직접 유도

2. 역전파 구현

유도한 공식을 코드로 옮기고, 연쇄 법칙(Chain Rule)을 끊어짐 없이 연결

3. 행렬 연산 최적화

GPU를 사용하기 위해 CUDA 커널을 직접 건드리거나 메모리 관리를 신경 써야함

03. 컴퓨터비전이란?

프레임워크(Framework)

프레임워크(Framework)란?

모델을 처음부터 직접 구현하지 않고, 학습·계산·최적화 과정을 체계적이고 효율적으로 구성할 수 있도록 도와주는 도구

왜 필요한가?

- 복잡한 수식과 미분 과정을 자동으로 처리
- 모델 구조 설계와 실험에 집중 가능
- 학습·평가·추론 과정을 일관된 방식으로 관리
- 코드 재사용이 쉬워 실험 비교 및 확장에 유리

예시

- 모델 구조를 코드 몇 줄로 직관적으로 정의 가능
- 이미지 데이터를 Tensor 형태로 자동 처리
- 손실 계산 → 오차역전파 → 파라미터 업데이트 과정을 자동으로 수행
- 동일한 코드로 학습 · 평가 · 추론 흐름을 일관되게 관리 가능

03. 컴퓨터비전이란?

파이토치(PyTorch)



파이토치(PyTorch)란?

딥러닝 모델의 구조와 학습 과정을 직관적인 코드로 구현할 수 있게 해주는 딥러닝 프레임워크.

파이토치(PyTorch)의 기본 구성

Tensor

데이터를 저장하고 계산하는 기본 단위

Model

입력을 받아 예측을 수행하는 신경망 구조

Loss Function

예측이 얼마나 틀렸는지 계산

Optimizer

손실을 줄이기 위해 파라미터를 업데이트

왜 PyTorch인가?

1. 수식 대신 코드

미분을 직접 계산할 필요 없이 기울기(Gradient)를 자동으로 계산

2. 오차역전파 자동 처리

연쇄 법칙을 직접 구현할 필요 없이, `loss.backward()` 한 줄로 역전파 수행이 가능하다.

3. 연산 최적화와 GPU 지원

대규모 행렬 연산에 최적화되어 있으며, GPU를 간단히 활용할 수 있어 데이터 규모가 커져도 코드 변경을 최소화할 수 있음



04. 기존 신경망의 한계

이미지 데이터의 본질과 접근방식

04. 기존 신경망의 한계

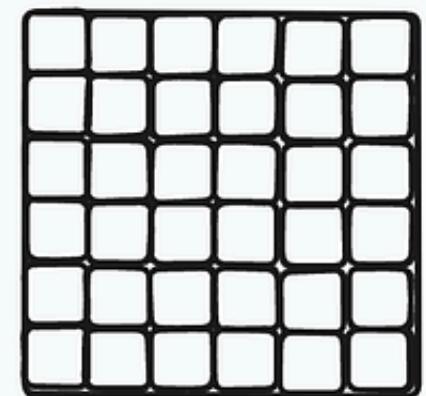
기존 신경망의 접근방식

왜 기존 신경망으로는 이미지가 잘 안 되었을까?

문제점 ①

초기 신경망 모델은 이미지를 정해진 길이의 숫자 벡터로 변환하여 입력으로 사용함

예) 28×28 이미지를 784차원 벡터로 펼쳐서 처리



6x6 pixels
Image



Flattened Image

04. 기존 신경망의 한계

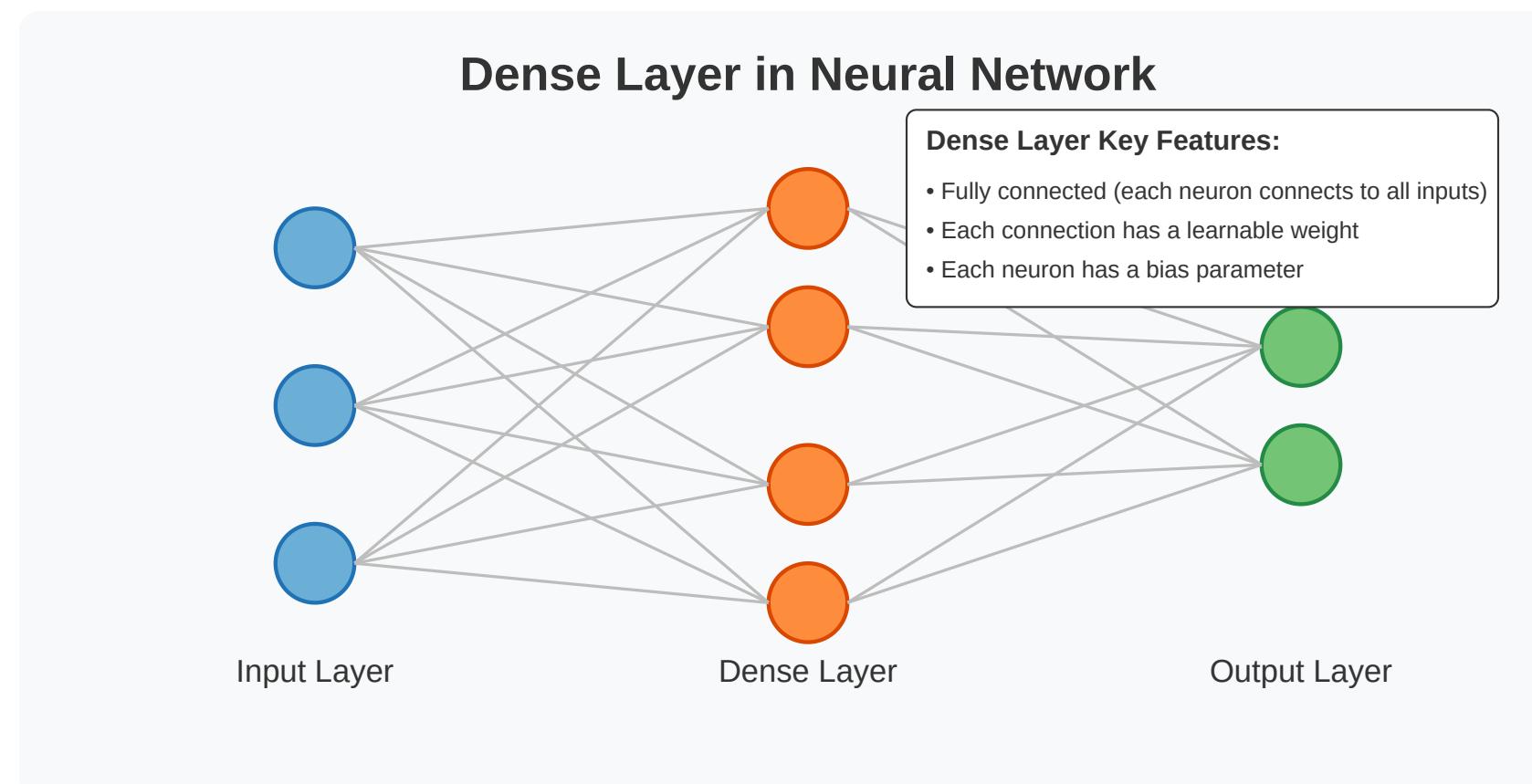
기존 신경망의 접근방식

왜 기존 신경망으로는 이미지가 잘 안 되었을까?

문제점 ②

Dense layer는 입력 벡터의 모든 값과 가중치를 각각 연결하여 계산함

→ 각 픽셀을 서로 독립적인 입력 변수로 학습



- 픽셀들이 격자 형태가 아니라 점(node)으로만 표현됨
- 이웃한 픽셀과 멀리있는 픽셀이 동일한 방식으로 연결



픽셀 간의 위치 관계나 이웃 정보는 반영하지 않음

04. 기존 신경망의 한계

정리

왜 기존 신경망으로는 이미지가 잘 안 되었을까?

지금까지의 신경망은 이미지를 숫자로는 잘 처리했지만, 이미지의 구조 자체를 이해하는 데에는 한계가 있었다.

1. 기존 신경망은 이미지를 2차원 구조가 아닌 1차원 숫자 벡터로 변환하여 처리함

이 과정에서 픽셀 간의 위치 관계가 사라짐

2. 모든 픽셀이 동일한 중요도와 입력값으로 취급됨

모델이 선과 면의 연결, 윤곽과 형태, 반복되는 패턴과 같은 시각적 구조를 직접 학습하기 어려움



따라서 이미지의 공간적 구조를 유지한 채 이해할 수 있는 방식이 필요함

Next Class...

- Kaggle 데이터 불러오기
- PyTorch 기초
- Linear Layer 기반 이미지 분류 모델 구현

진행자: 조치연



2026 아름 17기 AI 예비대학

김기현 (Kim Gi Hyeon)

서동주 (Seo Dong Ju)

조치연 (Jo Chi yeon)

지준구 (Ji Jun Gu)

이정연 (Lee Jung Yeon)

