

## 스터디 주간 활동 보고서

팀명	Robo:Loop	제출자 성명	홍송은
참여 명단	전효재, 홍송은, 김사웅		
모임 일시	2025년 05월 08일 17시 30분 ~ 18시 30분		
장소	온라인 구글 미팅	출석 인원	3
학습목표	<ul style="list-style-type: none"><li>출력층 구조, 활성화 함수, 손실 함수의 조합 원리를 정리한다.</li><li>PyTorch로 이진 분류, 다중 분류, 다중 레이블, 회귀 수식 및 모델 구성을 살펴본다.</li><li>BCE, CrossEntropy, MSE 등의 손실 함수 동작 원리를 코드로 비교 분석한다.</li><li>Sigmoid와 Softmax의 차이 및 위치(Sigmoid는 출력단, Softmax는 Loss 내부)를 명확히 이해한다.</li><li>실전 예제에 각 유형을 적용하며 오차 지표와 모델 동작을 직접 확인한다.</li></ul>		
학습내용	<p>컴퓨터비전 1~3주차 강의 기반 핵심 개념 및 중요 부분 정리 및 공유</p> <ul style="list-style-type: none"><li>홍송은<ul style="list-style-type: none"><li>requires_grad=True 꼭 설정해야 미분 가능<ul style="list-style-type: none"><li>실수로 빠뜨리면 backward()가 작동하지 않음</li></ul></li><li>.backward() 전 .sum() 필요</li><li>벡터나 행렬 결과는 바로 미분 불가 → 스칼라로 바뀌어야 함</li></ul></li></ul>		

- .backward() 호출할 때 x.grad.zero\_() 필수
- 사용하지 않을 시 기울기 계속 누적됨 → 학습 결과 왜곡

○ CNN 구조 중 Flatten 위치 헛갈리기 쉬움

- Conv + Pooling 끝나고 → FC 전에 Flatten 해야 함
- 예시)

`x = conv(x)`

`x = pool(x)`

`x = x.view(x.size(0), -1) # Flatten`

`x = fc(x)`

○ 손실 함수 vs 비용 함수

: CrossEntropy도 미니배치면 Loss, 전체 평균이면 Cost

- Loss: 한 샘플 기준
- Cost: 전체 평균

○ Sigmoid vs Softmax

- Sigmoid: 이진 분류 (출력 1개)
  - Softmax: 다중 분류 (출력 여러 개, 확률 합 1)
- : Softmax 후 CrossEntropy 같이 사용

○ Precision vs Recall

- Precision: 예측한 것 중 맞은 비율 (FP 줄이기)
  - Recall: 실제 정답 중 맞춘 비율 (FN 줄이기)
- : 암 진단 → Recall 중요, 스팸 필터 → Precision 중요

○ Batch vs SGD vs Mini-batch

- Batch: 안정적 but 느림
- SGD: 빠르지만 noisy
- Mini-batch: 일반적으로 가장 많이 사용

• 전효재

○ (3, 1) / (3,) 의 차이

`[[1][2][3]]`

`[1 2 3]`

○ 행렬의 곱(Dot Product)

- 파이토치에서 사용되는 자료 구조
  - 스칼라 - 0차원
  - 벡터 - 1차원
  - 행렬 - 2차원
  - 텐서 - 3차원
- 머신러닝 종류
  - 지도학습 : 피쳐와 라벨이 주어진 학습
  - 비지도학습 : 정답없이 패턴을 학습(그룹화, 이상탐지에 사용)
  - 강화학습 : 알파고 같은 ai (혼자 실행하고 학습)
- 인공지능, 머신러닝, 딥러닝의 관계  
: 인공지능 > 머신러닝 > 딥러닝
- 함수 종류  
: sigmoid, Tanh, ReLU
- 경사하강법  
: 함수의최솟값또는최댓값을찾는방법

- 김사웅

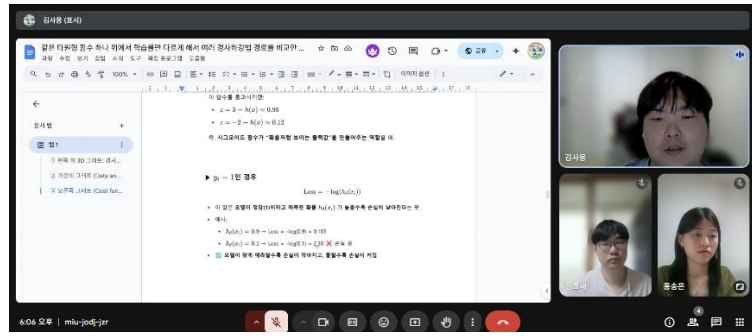
- 경사하강법은 함수의 기울기를 따라 비용 함수 값을 줄이도록 반복 이동하며 최소값을 찾는 알고리즘이다.
- 분석에 사용된 함수는  $f(x,y)=ax^2+by^2$   $f(x, y) = ax^2 + by^2$  형태의 타원형 함수로, 중심점이 전역 최소값이다.
- 왼쪽 위 3D 그래프는 경사하강법이 볼록한 곡면을 따라 이동하며 최솟값에 접근하는 과정을 보여준다.
- 이 3D 그래프에서 함수의 형태는 MSE 기반의 비용 함수(Cost Function)로, 중심이 최솟값이다.
- 가운데 그래프는 실제 데이터와 예측 모델의 직선을 시각화한 것으로, 학습 중 파라미터 변화에 따른 예측 변화가 나타난다.
- 검은 x표시는 실제 데이터 포인트, 색깔 직선은 경사하강법 중 특정 단계에서의 예측 결과이다.

		<ul style="list-style-type: none"> <li>○ 범례에는 각 시점에서의 파라미터 값이 기록되어 있어 학습 과정을 추적할 수 있다.</li> <li>○ 오른쪽 등고선 그래프는 비용 함수의 등고선을 따라 파라미터가 이동하는 경로를 시각화한 것이다.</li> <li>○ 학습률이 다른 경우 각각의 경로가 다르게 수렴하거나 진동하는 경향을 보인다.</li> <li>○ 타원형 함수와 등고선 그래프는 학습 경로의 수렴 여부와 속도를 파악하는 데 중요하다.</li> </ul>
활동평가	전효재	중간 점검으로 배운 내용을 이해하고 다시 보면서 잘 몰랐던 부분을 정리함. 학습을 하는 과정을 이해하는데 수식이 많이 사용되어 이 수식들을 이해하려고 노력했으나 어려운 부분이 있었음. 추가적인 학습으로 수식들을 이해해보고 이 수식들의 사용 방법을 알고 학습과정을 이해할 예정.
	홍승은	이번 중간 점검에서는 지금까지의 내용을 전반적으로 복습하며 흐름을 정리함. 특히, 실수하기 쉬운 개념과 혼동되는 내용을 중심으로 핵심 포인트를 정리함. 개념 간 비교를 통해 단순 암기가 아닌 구조적인 이해를 돕고자 함. 정리한 내용을 통해 이후 학습에서도 혼란 없이 따라갈 수 있도록 기반을 다지고자 하였으며 다음 스터디에서 모델 구성 기본 틀, 모델 간의 비교, 각 수식 분석 등에 집중할 예정.
	김사웅	개념 정리에 충실하며 경사하강법의 이론과 시각화 자료를 연계해 이해하려하였음. 수식과 그래프 해석을 결합하여 학습 내용을 입체적으로 파악하려고 시도함. 학습률에 따른 수렴 경로 차이까지 관찰한 것은 실제 모델 튜닝에 필요한 통찰을 확보하는 과정을 진행함. 최대한 표현하려고 노력하였고 내용을 논리적으로 정리하여, 혼자 공부해도 체계적으로 정리할 수 있는 역량을 기를 다만, 추가적으로 직접 실습 코드나 수치 실험을 병행하면 이해가 더 확실해지고 실무에도 더 유용할 것임.

과제	<ul style="list-style-type: none"> <li>컴퓨터비전 3~4주차 강의 내용을 정리하고, 핵심 개념 및 중요 부분을 중심으로 학습             <ul style="list-style-type: none"> <li>스터디 시 주요 개념 설명 및 질의사항 답변할 수 있도록 확실하게 이해한 후 정리</li> </ul> </li> <li>정리본: 스터디 전까지 Slack에 공유</li> </ul>
향후 계획	<ul style="list-style-type: none"> <li>VGG와 ResNet의 전체 아키텍처 구조 차이점(단순 vs. 잔차 연결)을 파악한다.</li> <li>각 모델의 Conv, Pooling, FC 레이어 구성 방식과 위치를 정확히 분석한다.</li> <li>ResNet의 핵심인 Residual Block 수식과 구조(identity mapping, skip connection)를 이해한다.</li> <li>VGG의 반복적 Conv + MaxPool 패턴이 특징인 이유와 장단점을 비교한다.</li> <li>실제 구현 코드를 통해 입력→출력까지 흐름을 시각적으로 추적하고 계층별 역할을 해석한다.</li> </ul>
첨부 자료	<div data-bbox="368 1290 571 1684"> <p>스터디 화면</p> <p>1</p> </div> <div data-bbox="584 1319 1329 1655"> <p>The screenshot shows a Zoom meeting interface. The main window displays a document with handwritten notes. The notes include a definition of 'Cost Function' and a formula for 'J(theta)'. The formula is <math>J(\theta) = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2</math>. The document also includes a definition of 'Cost Function' and a formula for 'J(theta)'. The formula is <math>J(\theta) = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2</math>. The document also includes a definition of 'Cost Function' and a formula for 'J(theta)'. The formula is <math>J(\theta) = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2</math>.</p> </div>

스터디 화면

2



Github 주소:

[https://github.com/yellowHSE/ds\\_rokey4\\_study\\_team2/tree/main/codes/week8\\_ComputerVision\\_1\\_3](https://github.com/yellowHSE/ds_rokey4_study_team2/tree/main/codes/week8_ComputerVision_1_3)

결과물

