

Homework #3

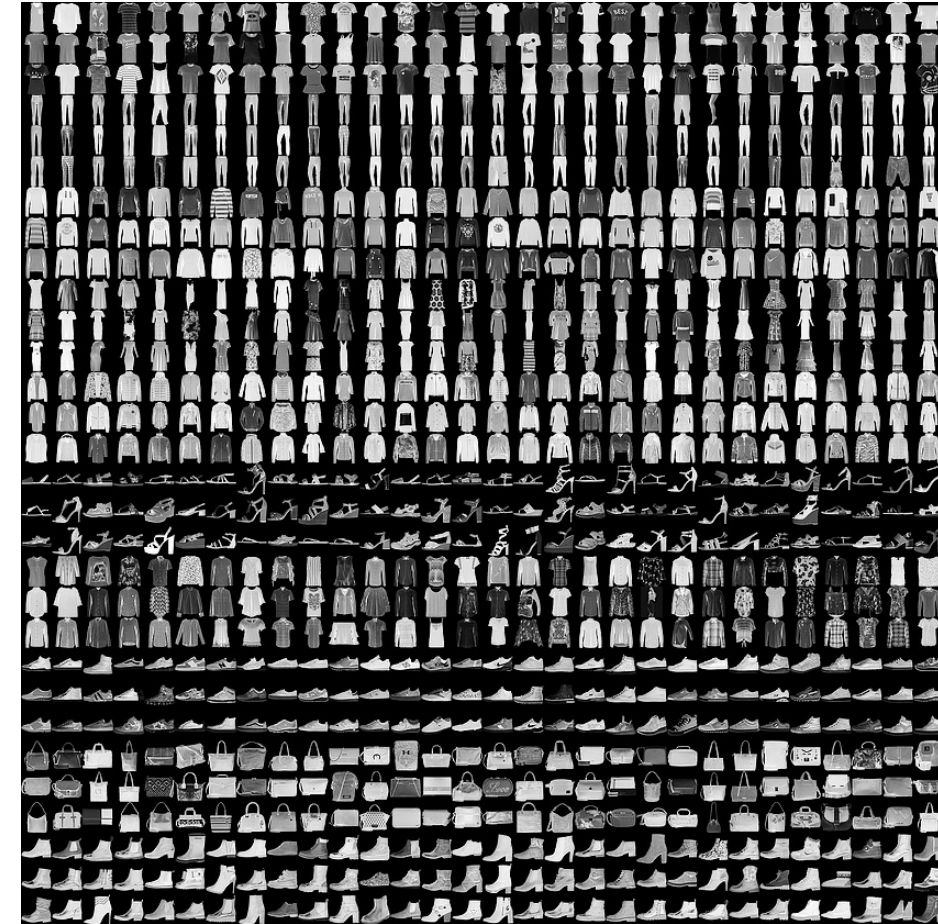
Nov. 2023

<http://link.koreatech.ac.kr>

Fashion MNIST

◆ Fashion MNIST Dataset

- It consists of 70,000 28x28 handwritten digit images in 10 classes, with 7,000 images per class (ranging from 0 to 9)
 - 60,000 training images
 - 10,000 test images
- Labels
 - 0: T-shirt/top, 1: Trouser, 2: Pullover, 3: Dress, 4: Coat
 - 5: Sandal, 6: Shirt, 7: Sneaker, 8: Bag, 9: Ankle boot
- Source
 - <https://github.com/zalandoresearch/fashion-mnist>
- Pytorch Class
 - <https://pytorch.org/vision/stable/generated/torchvision.datasets.FashionMNIST.html#torchvision.datasets.FashionMNIST>



Fashion MNIST

◆ [참고 코드] _02_homeworks/_03_fashion_mnist/a_fashion_mnist_data.py (1/4)

```
import os
from pathlib import Path
import torch
import wandb
from torch import nn

from torch.utils.data import DataLoader, random_split
from torchvision import datasets
from torchvision.transforms import transforms

BASE_PATH = str(Path(__file__).resolve().parent.parent.parent)
print(BASE_PATH)  # >>> BASE_PATH: /Users/yhhan/git/link_dl

import sys
sys.path.append(BASE_PATH)

from _01_code._99_common_utils.utils import get_num_cpu_cores, is_linux, is_windows
```

Fashion MNIST

◆ [참고 코드] _02_homeworks/_03_fashion_mnist/a_fashion_mnist_data.py (2/4)

```
def get_fashion_mnist_data():
    data_path = os.path.join(BASE_PATH, "_00_data", "j_fashion_mnist")
    f_mnist_train = datasets.FashionMNIST(
        data_path, train=True, download=True, transform=transforms.ToTensor()
    )
    f_mnist_train, f_mnist_validation = random_split(f_mnist_train, [55_000, 5_000])

    print("Num Train Samples: ", len(f_mnist_train))           # >>> 55000
    print("Num Validation Samples: ", len(f_mnist_validation)) # >>> 5000
    print("Sample Shape: ", f_mnist_train[0][0].shape)        # torch.Size([1, 28, 28])

    num_data_loading_workers = get_num_cpu_cores() if is_linux() or is_windows() else 0
    print("Number of Data Loading Workers:", num_data_loading_workers) # >>> 0

    train_data_loader = DataLoader(
        dataset=f_mnist_train, batch_size=wandb.config.batch_size, shuffle=True,
        pin_memory=True, num_workers=num_data_loading_workers
    )
```

Fashion MNIST

◆ [참고 코드] _02_homeworks/_03_fashion_mnist/a_fashion_mnist_data.py (3/4)

```
def get_fashion_mnist_data():  
    ...  
  
    validation_data_loader = DataLoader(  
        dataset=f_mnist_validation, batch_size=wandb.config.batch_size,  
        pin_memory=True, num_workers=num_data_loading_workers  
    )  
  
    mnist_transforms = nn.Sequential(  
        transforms.ConvertImageDtype(torch.float),  
        transforms.Normalize(mean=0.0, std=0.1),  
    )  
  
    return train_data_loader, validation_data_loader, mnist_transforms
```

1번 문제. 정확한 mean 값과 std 값을 찾으시오.

Fashion MNIST

◆ [참고 코드] _02_homeworks/_03_fashion_mnist/a_fashion_mnist_data.py (4/4)

```
def get_fashion_mnist_test_data():
    data_path = os.path.join(BASE_PATH, "_00_data", "j_fashion_mnist")

    f_mnist_test_images = datasets.FashionMNIST(data_path, train=False, download=True)
    f_mnist_test = datasets.FashionMNIST(
        data_path, train=False, download=True, transform=transforms.ToTensor()
    )
    print("Num Test Samples: ", len(f_mnist_test))      # >>> 10000
    print("Sample Shape: ", f_mnist_test[0][0].shape)   # torch.Size([1, 28, 28])

    test_data_loader = DataLoader(dataset=f_mnist_test, batch_size=len(f_mnist_test))

    f_mnist_transforms = nn.Sequential(
        transforms.ConvertImageDtype(torch.float),
        transforms.Normalize(mean=0.0, std=0.1),
    )

    return f_mnist_test_images, test_data_loader, f_mnist_transforms
```

1번 문제. 정확한 mean 값과 std 값을 찾으시오.

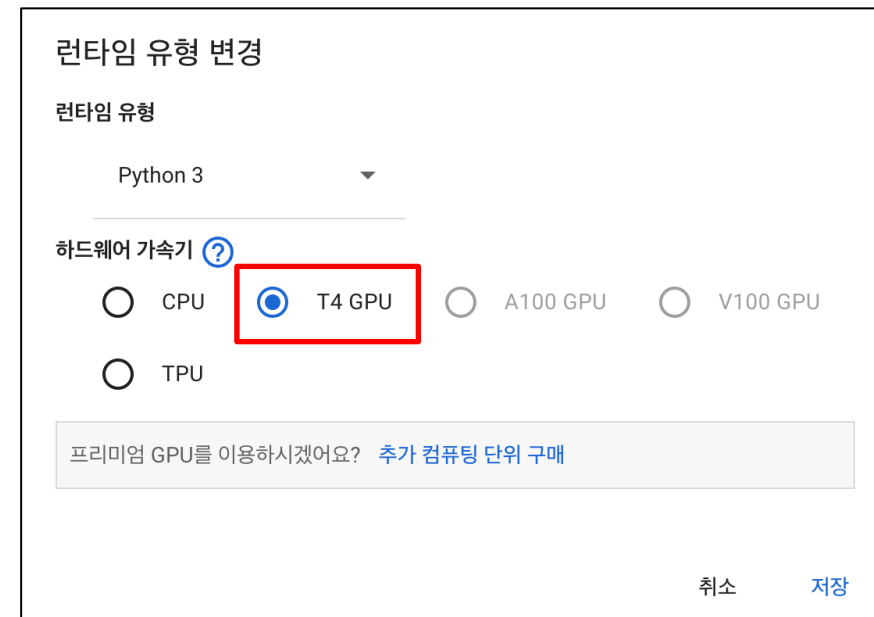
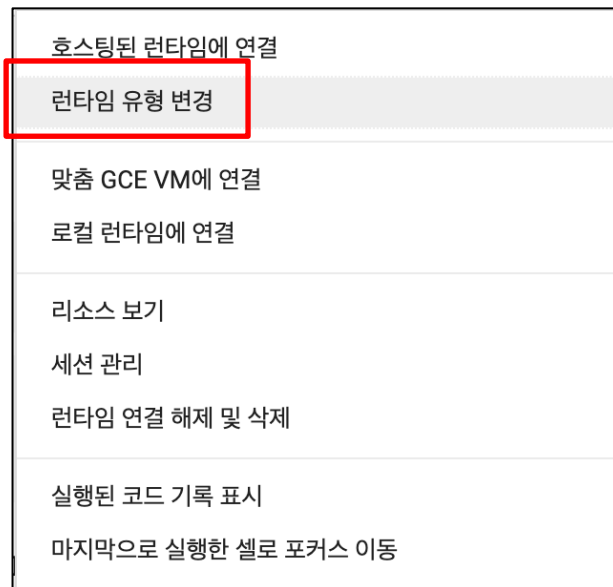
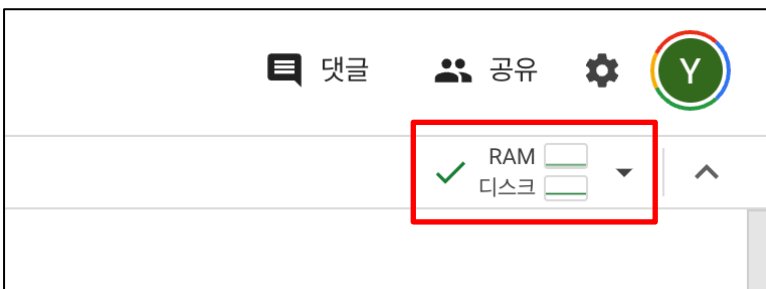
Google Colab

◆ Goggle Colab

- <https://colab.research.google.com/?hl=ko>

◆ Goggle Colab 사용법

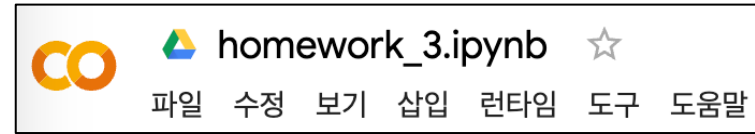
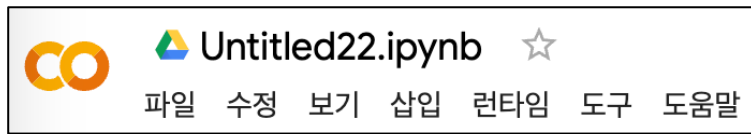
- 구글 코랩 사용하기: 마크 다운 연습(글자 크기, 리스트 만들기 등)
 - <https://shorturl.at/jITZ3>
- 런타임 유형 변경 → 하드웨어 가속시: GPU 사용



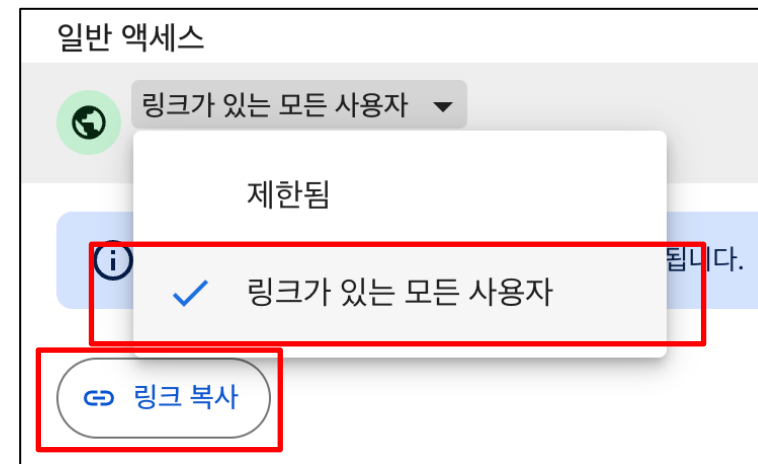
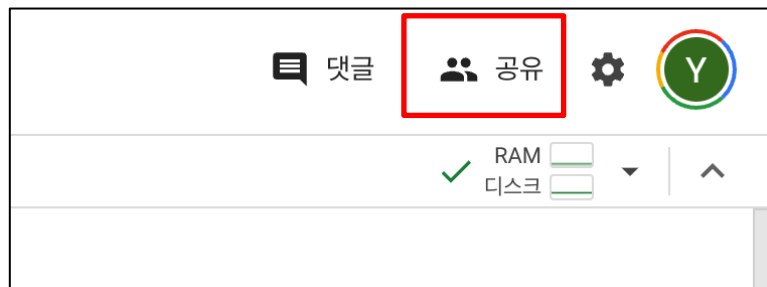
Google Colab

◆ Goggle Colab을 사용한 숙제 제출을 위한 처리 작업

– 1. 파일명부터 변경하기



– 2. 공유 설정 → 링크가 있는 모든 사용자



복사된 링크를 EL2 숙제 게시물에 기입

문제 1

◆ [문제 1] Fashion MNIST 데이터 정규화를 위한 Mean과 Std. 값 찾기

- 10점 중 1점 할당
- 문제 풀이시 참고할 코드
 - _01_code/_06_fcn_best_practice/a_data_mnist.py
 - _02_homeworks/_03_fashion_mnist/a_fashion_mnist_data.py
- Colab에서의 풀이 과정 일부

▼ [문제 1] Fashion MNIST 데이터 정규화를 위한 Mean과 Std. 값 찾기

```
1 from torch.utils.data import random_split, DataLoader
2 from torchvision import datasets
```

```
[15] 1 data_path = "."
      2 f_mnist_train = datasets.FashionMNIST(data_path, train=True, download=True, transform=transforms.ToTensor())
      3 f_mnist_train, f_mnist_validation = random_split(f_mnist_train, [55_000, 5_000])
```

```
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.gz to ./FashionMNIST/raw/train-images-idx3-ubyte.gz
100%|██████████| 26421880/26421880 [00:01<00:00, 16746830.46it/s]
Extracting ./FashionMNIST/raw/train-images-idx3-ubyte.gz to ./FashionMNIST/raw
```

```
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.gz to ./FashionMNIST/raw/train-labels-idx1-ubyte.gz
100%|██████████| 29515/29515 [00:00<00:00, 272770.28it/s]
Extracting ./FashionMNIST/raw/train-labels-idx1-ubyte.gz to ./FashionMNIST/raw
```

```
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz to ./FashionMNIST/raw/t10k-images-idx3-ubyte.gz
100%|██████████| 4422102/4422102 [00:00<00:00, 5066056.98it/s]
Extracting ./FashionMNIST/raw/t10k-images-idx3-ubyte.gz to ./FashionMNIST/raw
```

```
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz to ./FashionMNIST/raw/t10k-labels-idx1-ubyte.gz
100%|██████████| 5148/5148 [00:00<00:00, 6535192.79it/s]Extracting ./FashionMNIST/raw/t10k-labels-idx1-ubyte.gz to ./FashionMNIST/raw
```

문제 2

◆ [문제 2] Fashion MNIST 데이터에 대하여 CNN 학습시키기 (1/3)

- 학습 데이터, 검증 데이터, 테스트 데이터를 가져오는 코드는 반드시 다음 코드를 Colab으로 가져와 활용하기
 - `_02_homeworks/_03_fashion_mnist/a_fashion_mnist_data.py`
 - 따라서, Training, Validation, Test 데이터의 수는 각각 다음과 같음
 - 55,000, 5,000, 10,000
 - 다만, 다음 코드의 Mean과 Std값은 이번 [문제 1]에서 찾은 값으로 변경하기

```
f_mnist_transforms = nn.Sequential(  
    transforms.ConvertImageDtype(torch.float),  
    transforms.Normalize(mean=0.0, std=0.1),  
)
```

- 위 `f_mnist_transforms` 객체에 추가적인 transform 객체를 넣어 활용하는 것 허용
 - 예를 들어 Image Augmentation 기법 활용 가능

문제 2

◆ [문제 2] Fashion MNIST 데이터에 대하여 CNN 학습시키기 (2/3)

– Wandb에 다음 4가지 그래프 출력하여 EL2 숙제 게시물에 Wandb URL 제시

- Training Loss, Validation Loss, Training Accuracy, Validation Accuracy

– *주의*: Overview 메뉴 페이지에서 접근 권한 설정

Privacy **PRIVATE**

Last active 2023. 10. 3. 오후 1:23:55

Author **K** link-koreatech

Contributors 1 user

Total runs 2

Total compute 40 minutes



What problem does this project tackle? ✎

Privacy **PUBLIC**

Last active 2023. 10. 3. 오후 1:23:55

Author **K** link-koreatech

Contributors 1 user

Total runs 2

Total compute 40 minutes

Project Access

Privacy settings affect your whole project, including runs, reports, artifacts, etc.

☐ **Private** DEFAULT
Only you can view and contribute

☒ **Public**
Anyone can view

+ Show advanced settings

Create a team to collaborate on private projects.

Create Team

Cancel Save

문제 2

◆ [문제 2] Fashion MNIST 데이터에 대하여 CNN 학습시키기 (3/3)

– 만족해야 할 요구사항

- Validation Accuracy: 91.0% 이상

- Validation Accuracy: 91.0% 미만: 10만점에서 7점 이상 획득 불가

– 수업에서 배운 다음 내용들에 소개된 기법들 최대한 활용

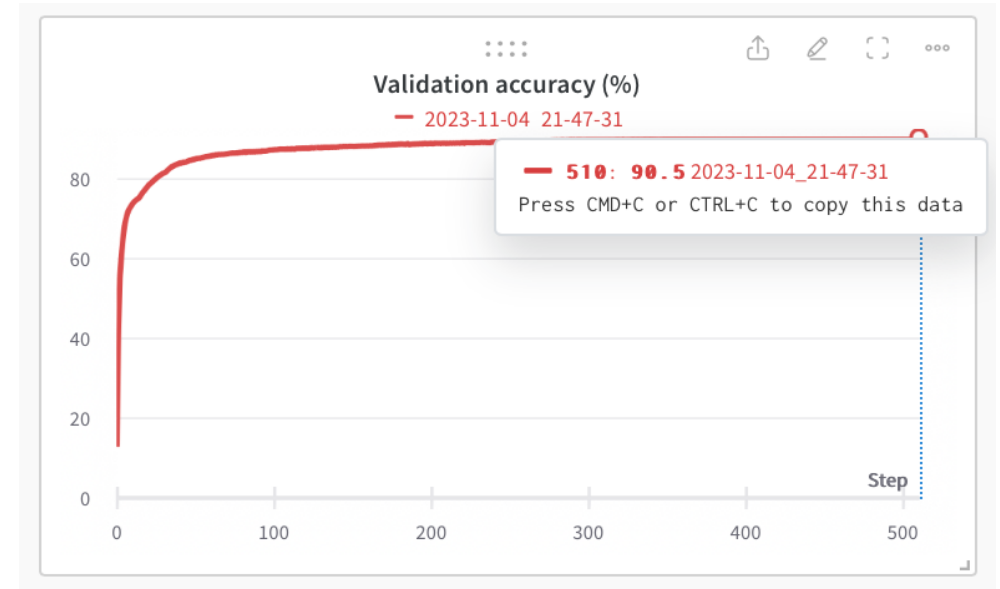
- 08. 다양한 최적화 및 정규화 기법

- 09. 다양한 CNN 모델

– 마크다운 글 작성 방법으로 본인이 찾은 최적의 하이퍼파라미터 정리하여 제시

- Learning Rate
- Batch Size
- Early Stop Patience
- Weight Decay Value
- Normalization Type
- Dropout Rate 등

– torchinfo 모듈 사용한 모델 구조 출력 필요 (from torchinfo import summary)



문제 3 & 4

◆[문제 3] 학습 완료된 모델로 테스트 데이터 Accuracy 확인하기

– 만족해야 할 요구사항

- Test Accuracy: 90.0% 이상
- Test Accuracy: 90.0% 미만: 10만점에서 7점 이상 획득 불가

◆[문제 4] 샘플 테스트 데이터 분류 예측 결과 확인하기

– 10,000개의 테스트 데이터 중 10개를 임의로 선택하여

- 1) 해당 이미지를 출력하고
- 2) 해당 이미지의 레이블을 출력하고
- 3) 학습 완료된 모델을 사용하여 해당 이미지의 분류 예측 결과를 출력하여
- 4) 이미지 레이블과 분류 예측결과가 동일한지 점검

– 최소한 1개의 테스트 이미지에 대해서는 해당 이미지 레이블과 분류 예측 결과가 동일하지 않도록 하며, 이러한 경우 분류 예측 결과가 틀린 이유에 대해서 이미지를 보면서 나름대로의 해석을 작성해야 함.

요구사항

◆ 보고서(Report) 내용에 대한 요구사항

- 프로그램 코드가 실행되는 것을 확인할 수 있도록 Colab 내 셀별로 출력 값들을 정확하게 나타낼 것
 - 모든 코드에 대한 출력이 잘 나와 있는지 확인함
- 핵심 코드라고 생각되는 것들에 대해서 주석(한글 또는 영문)을 넣을 것
 - 다다익선
- 코딩을 통하여 자신이 취득한 기술적 사항/고찰 내용을 생각한 바를 제시할 것
 - 다다익선
- [NOTE] 숙제 마지막에는 "숙제 후기" 라는 섹션 제목으로 본 숙제를 한 이후의 느낀점, 하고싶은 말, 또는 불평 등을 반드시 넣을 것
 - "숙제 후기" 섹션이 없으면 감점 처리

요구사항

◆ 제출형태 및 방법

- Colab 파일명: [homework3.ipynb](#)
 - 각 4가지 문제에 대해 섹션 제목을 정확히 넣고, 섹션이 잘 구분되도록 작성하기
- 숙제 제출 방식
 - <http://el2.koreatech.ac.kr> 의 "딥러닝및실습" 교과목 "과제" 메뉴
 - 게시물 본문에는 Colab URL 및 Wandb URL 넣기 (총 2개 URL 제시)
- 숙제에 관한 질의/응답
 - <http://el2.koreatech.ac.kr> 의 "딥러닝및실습" 교과목 "Q&A" 게시판 활용

◆ 제출기한

- 2023년 11월 18일 (토) 23시 59분
- 지각 제출은 받지 않습니다 (0점 처리)