

22 - 1 딥러닝 및 응용 과제 1

이미지 분류

조교 김수형, 조환희

ksh970404@hanyang.ac.kr

straw88@hanyang.ac.kr

과제 개요

- 수업시간에 배운 다양한 방법들을 이용해서 이미지 분류 task 수행
- 세부사항
 1. 지금까지 **실습 때 배운 코드만** 사용하여 이미지 분류하는 것이 목표.
CNN, Transformer 등 사용불가능 오직 Linear Layer만 사용
 2. 실습 때 진행한 **CIFAR10 dataset** 을 분류하는 것이 목표.
 3. 학습된 모델의 test dataset을 분류한 **accuracy 성능으로 성적반영**

제공된 코드

- assignment_1.ipynb : 모델 학습과 학습된 모델을 평가하기 위한 코드

```
import torch
import torch.nn as nn

import torchvision
import torchvision.transforms as transforms

# 미리 작성된 코드들은 수정할 수 없으며, 이외의 코드를 작성하시면 됩니다.

torch.manual_seed(0)
torch.cuda.manual_seed(0)
torch.cuda.manual_seed_all(0)

class Classifier(nn.Module):
    # 모델의 코드는 여기서 작성해주세요

    def __init__(self):
        super(Classifier, self).__init__()

    def forward(self, x):

        return output

if __name__ == "__main__":
    # 학습코드는 모두 여기서 작성해주세요

    train_dataset = torchvision.datasets.CIFAR10(root="CIFAR10/",
                                                train=True,
                                                transform=transforms.ToTensor(),
                                                download=True)

    test_dataset = torchvision.datasets.CIFAR10(root="CIFAR10/",
                                                train=False,
                                                transform=transforms.ToTensor(),
                                                download=True)

    model = Classifier()

    torch.save(model.state_dict(), 'model.pt') # 학습된 모델을 저장하는 코드입니다.
```

```
# 아래의 코드로 평가를 진행할 예정이므로 아래의 코드가 정상 동작 해야하며, 제출된 모델의 성능을 확인하시면 됩니다.

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

test_dataset = torchvision.datasets.CIFAR10(root="CIFAR10/",
                                            train=False,
                                            transform=transforms.ToTensor(),
                                            download=True)

test_dataloader = torch.utils.data.DataLoader(test_dataset, batch_size=10000)

classifier = Classifier().to(device)
classifier.load_state_dict(torch.load('model.pt'))
classifier.eval()

for data, label in test_dataloader:
    data = data.view(-1, 32 * 32 * 3).to(device)

    with torch.no_grad():
        logits = classifier(data)

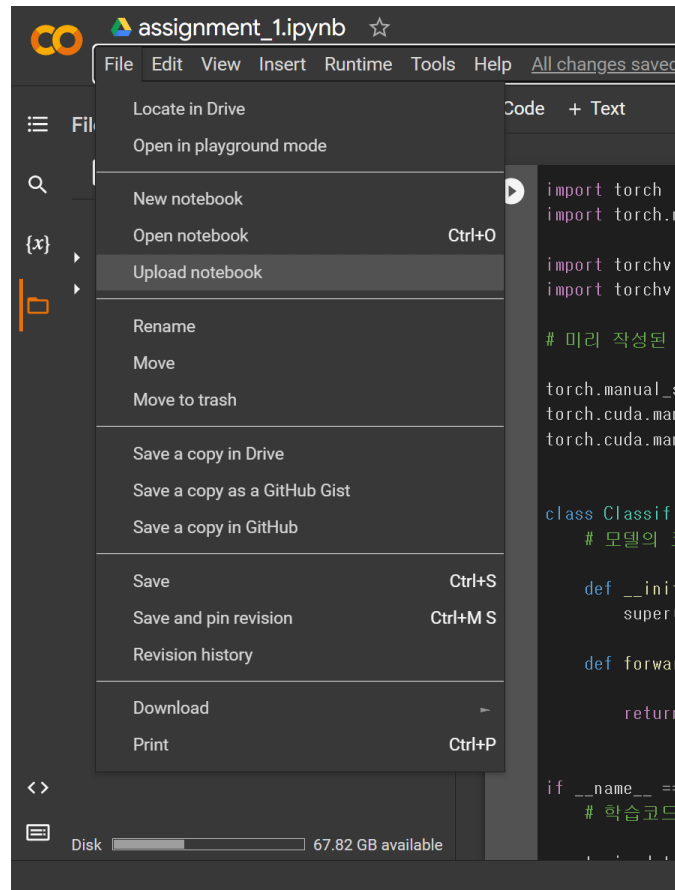
        pred = torch.argmax(logits, dim=1)

        total = len(label)
        correct = torch.eq(pred, label.to(device)).sum()

        print("Accuracy on test set : {:.4f}%".format(100 * correct / total))
```

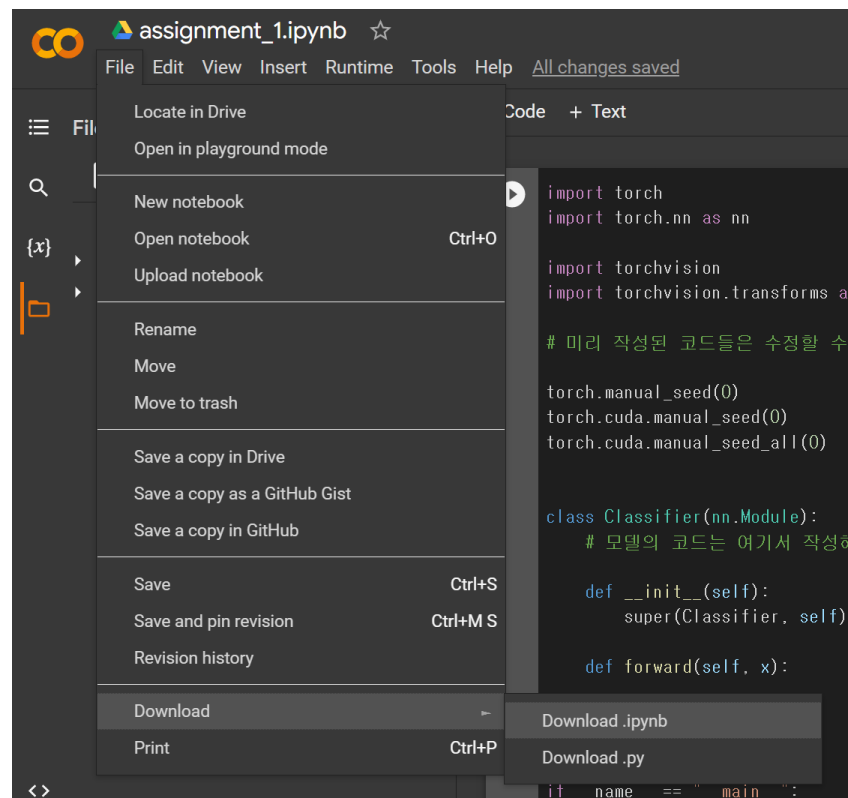
Colab ipynb 업로드

- File – Upload notebook



Colab ipynb 다운로드

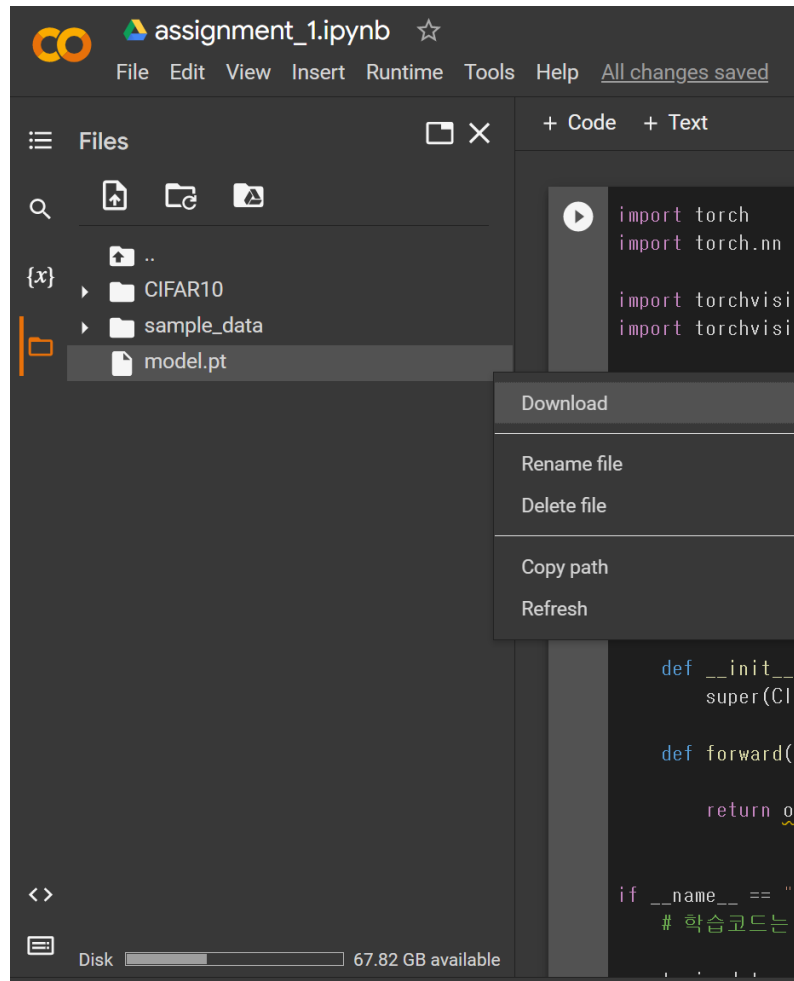
- File – Download – Download .ipynb



Colab 파일 다운로드



- 왼쪽 바의 폴더 아이콘 – 파일 점 세 개 클릭 - Download



점수 산출

- 코드 (70%)

- 지정한 파일 명으로 실행여부 (5%)
- 모델구현 및 실행 (45%)
- 성능향상 정도를 평가하여 점수 부여 (20%)

Accuracy(%)	점수(%)
$x < 52$	0
$52 \leq x < 55$	10
$55 \leq x < 58$	15
$58 \leq x$	20

- 보고서 (30%)

- 코드 설명
 - 모델(코드)에 대한 **설명** 명시 (15%)
- 실험결과
 - 성능향상을 위해 진행한 실험들의 **성능** 비교 (15%)

ex) dropout 추가, optimizer 변화, hyperparameter 조정 등

과제 조건

- 환경
 - 프로그래밍 언어 : **Python 3.7, pytorch 3.7~버전**
 - OS : **Windows**
 - 보고서 : **PDF**
- 제출 사항
 - **assignment_1.zip**
 - 파이썬 파일 : **assignment_1.ipynb** 혹은 **assignment_1.py**
 - 결과 보고서 : **본인학번_assignment_1.pdf**
 - 학습된 모델 체크포인트 파일: **model.pt**

주의 사항

- **파일명** 반드시 준수
 - assignment_1.zip
 - assignment_1.ipynb 혹은 assignment_1.py
 - 본인학번_assignment_1.pdf
 - model.pt
- 제출 기한 : **2022.04.15 (23:59)**
- 추가 제출 기한 **없음**.
- 점수 비중 : **코드 70% 보고서 30%**

Thank you!
