23 - 1 딥러닝 및 응용 과제

AutoEncoder with CNN

조교 김수형
ksh970404@hanyang.ac.kr

과제 개요

- **수업시간에 배운 다양한 방법들을 이용해서 AutoEncoder 네트워크를 완성하고, 이미지 classification 수행**
- **세부사항**

1. 지금까지 **실습 때 배운** ~~코드~~**네트워크만** 사용하여 Autoencoder 네트워크를 완성하여 encoder 를 학습시키고 encoder 뒤에서 classifier 를 추가로 학습하여 이미지 분류
    - 실습때 배운 neural network 만 사용가능 (linear, CNN)
    - 모든 layer는 2개 이상의 layer로 구성되어야함
2. Oxford flower 102 dataset 으로 진행
3. 모델의 성능향상을 위한 여러가지 기법을 적용하고 변화를 보고서로 서술한다. (3가지 이상)

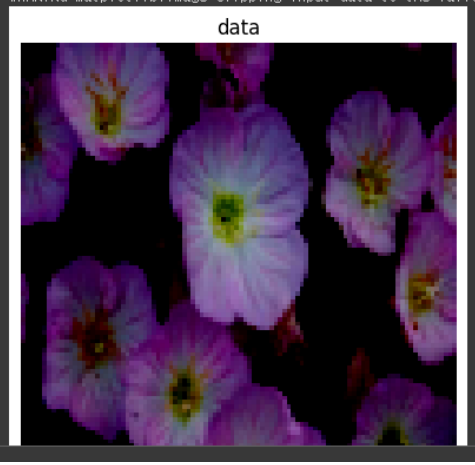
제공된 코드

- **assignment.ipynb : 데이터셋 다운로드 및 기본구조**

```
[4]  1 transform = transforms.Compose([transforms.Resize((100, 100)),
     2                                  transforms.ToTensor(),
     3                                  transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])
     4
     5 train_dataset = torchvision.datasets.Flowers102(root="StanfordCars/",
     6                     split='train',
     7                     transform=transform,
     8                     download=True)
     9 test_dataset = torchvision.datasets.Flowers102(root="StanfordCars/",
    10                     split='test',
    11                     transform=transform,
    12                     download=True)
```

```
1 import matplotlib.pyplot as plt
2
3 sample = train_dataset[0][0].numpy()
4 sample = np.transpose(sample,(1,2,0))
5
6 fig, ax = plt.subplots(1,1)
7 ax.set_title('data')
8 ax.set_axis_off()
9 ax.imshow(sample)
10 plt.show()
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



제공된 코드

- **assignment.ipynb : 데이터셋 다운로드 및 기본구조**

```
[ ]
```

```python
##### AutoEncoder 모델 코드 #####

class Encoder(nn.Module):
    def __init__(self,):
        super(Encoder, self).__init__()
        self.encode = nn.Sequential(

        )
    def forward(self, input):
        return self.encode(input)

class Decoder(nn.Module):
    def __init__(self, ):
        super(Decoder, self).__init__()
        self.decode = nn.Sequential(

        )
    def forward(self, input):
        return self.decode(input)

class AutoEncoder(nn.Module):
    def __init__(self):
        super(AutoEncoder, self).__init__()
        self.encoder = Encoder()
        self.decoder = Decoder()

    def forward(self, input):
        z = self.encoder(input)
        x_hat = self.decoder(z)
        return z, x_hat
```

```
[ ]
```

```python
##### AutoEncoder 학습 코드 #####

```

```
[6]
```

```python
##### Classifier 모델 코드 #####

class Classifier(nn.Module):
    def __init__(self, ):
        super(Classifier, self).__init__()
        self.classify = nn.Sequential(

        )
    def forward(self, input):
        return self.classify(input)
```

```
[7]
```

```python
##### Classifier 학습 코드 #####


```

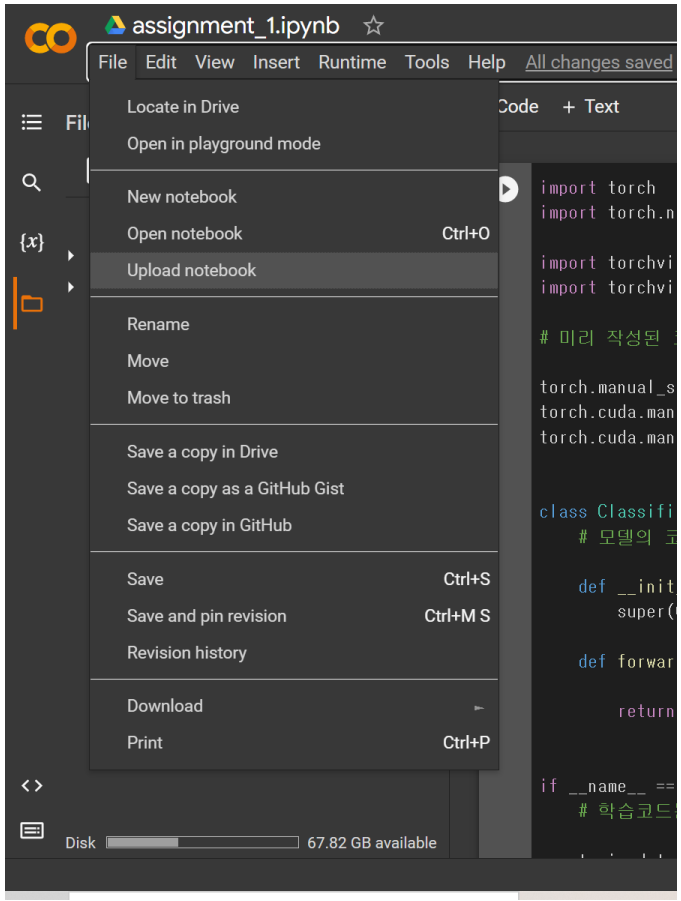
```
[ ]
```

```python
##### Classifier 정확도 측정 코드 #####


```

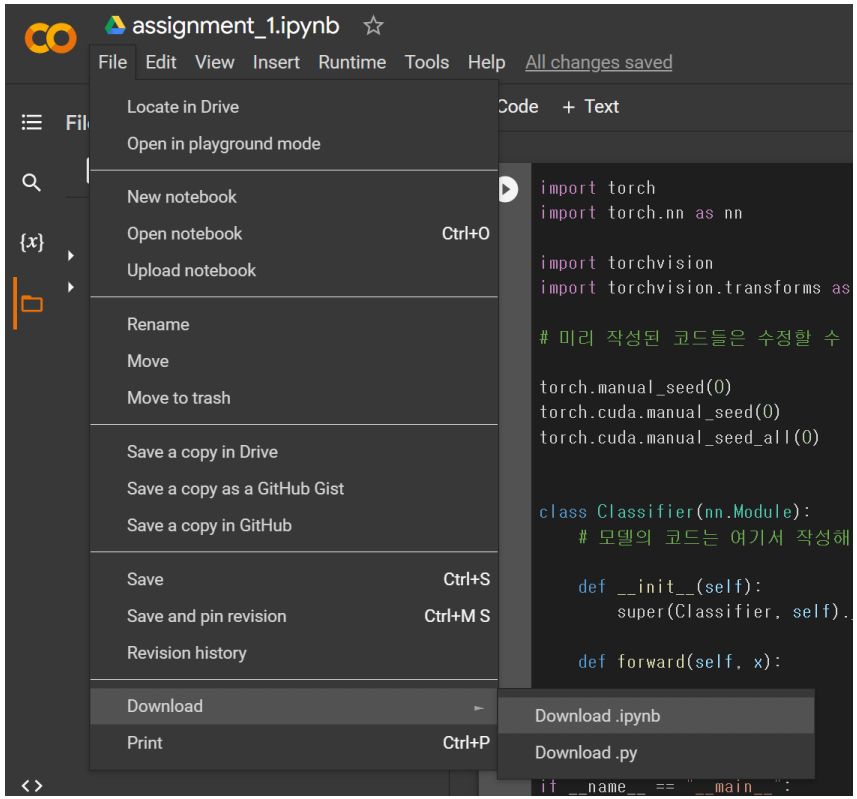
Colab ipynb 업로드

- File – Upload notebook



Colab ipynb 다운로드

- File – Download – Download .ipynb



점수 산출

- **코드 (70%)**
  - 모든 모델을 ~~CNN layer로만 구현~~ (70%) -> CNN layer를 포함하여 구현 및 실행가능
  - Encoder와 classifier만 CNN layer를 포함하여 구현 및 실행가능(30%)
  - 모델구현 및 실행가능 (10%)

- **보고서 (30%)**
  - **코드 설명**
    - 모델(코드)에 대한 **설명** 명시 (10%)
  - **실험결과**
    - 성능향상을 위해 진행한 실험들(3가지 이상)의 **성능** 비교 (20%)
      - ex) dropout 추가, optimizer 변화, hyperparameter, batch norm 등

과제 조건

- **환경**
  - 프로그래밍 언어 : **Python 3.7, pytorch 3.7~버전**
  - OS : **Windows**
  - 보고서 : **PDF**

- **제출 사항**
  - **assignment.zip**
    - 파이썬 파일 : **assignment.ipynb** 혹은 **assignment.py**
    - 결과 보고서 : **본인학번_assignment.pdf**

주의 사항

- **파일명** 반드시 준수
  - assignment.zip
    - assignment.ipynb 혹은 assignment.py
    - 본인학번_assignment.pdf
- 제출 기한 : **2023.05.31 (23:59)**
- 추가 제출 기한 **없음**.
- 점수 비중 : **코드 70% 보고서 30%**

Thank you!