인공지능 개론
L12.1 Autoencoder with Pytorch

국민대학교
소프트웨어융합대학원
박하명

필요한 모듈 불러오기

```
import torch
import torch.nn.functional as F
from torch import nn, optim
from torchvision import transforms, datasets
import matplotlib.pyplot as plt
```

데이터셋 불러오기

- FashionMNIST 데이터 불러오기
  - 28*28 크기의 옷 이미지 데이터셋

```
trainset = datasets.FashionMNIST(
   root = './.data/',
   train = True,
   download = True,
   transform = transforms.ToTensor()
)

CLASSES = {
   0: 'T-shirt/top', 1: 'Trouser', 2: 'Pullover', 3: 'Dress', 4: 'Coat',
   5: 'Sandal', 6: 'Shirt', 7: 'Sneaker', 8: 'Bag', 9: 'Ankle boot'
}
```

CUDA 사용 설정

```python
USE_CUDA = torch.cuda.is_available()
DEVICE = torch.device('cuda' if USE_CUDA else 'cpu')
```

Autoencoder

- encoder: 784 → 128 → 64 → 12 → 2, fully connected layers
- decoder: 2 → 12 → 64 → 128 → 784, fully connected layers

```
relu = nn.ReLU()

encoder = nn.Sequential(
   nn.Linear(28*28, 128), relu,
   nn.Linear(128, 64), relu,
   nn.Linear(64,12), relu,
   nn.Linear(12, 2)
   )
```

```
decoder = nn.Sequential(
   nn.Linear(2, 12), relu,
   nn.Linear(12, 64), relu,
   nn.Linear(64, 128), relu,
   nn.Linear(128, 28*28),
   nn.Sigmoid()
   )
```

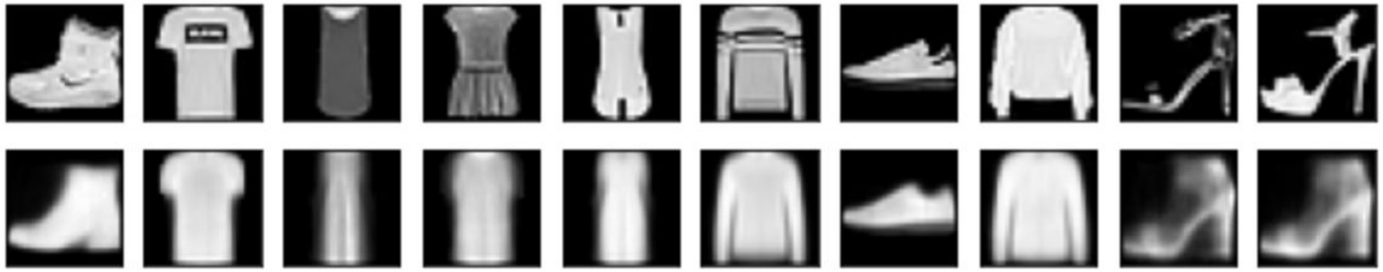
```
autoencoder = nn.Sequential(encoder, decoder).to(DEVICE)
```

학습하기

```python
# x에서 각각 값을 0~255에서, 0~1로 변환
x = (trainset.data.view(-1, 28*28).float()/255).to(DEVICE)
y = x # autoencoder에서는 x와 y가 동일하다

for epoch in range(1001):
    autoencoder.train()
    encoded = encoder(x)
    decoded = decoder(encoded)

    cost = mse(y, decoded)

    optimizer.zero_grad()
    cost.backward()
    optimizer.step()

    if epoch % 100 == 0:
        autoencoder.eval()
        print("epoch: {}, cost: {:.6f}".format(epoch, cost.item()))
```

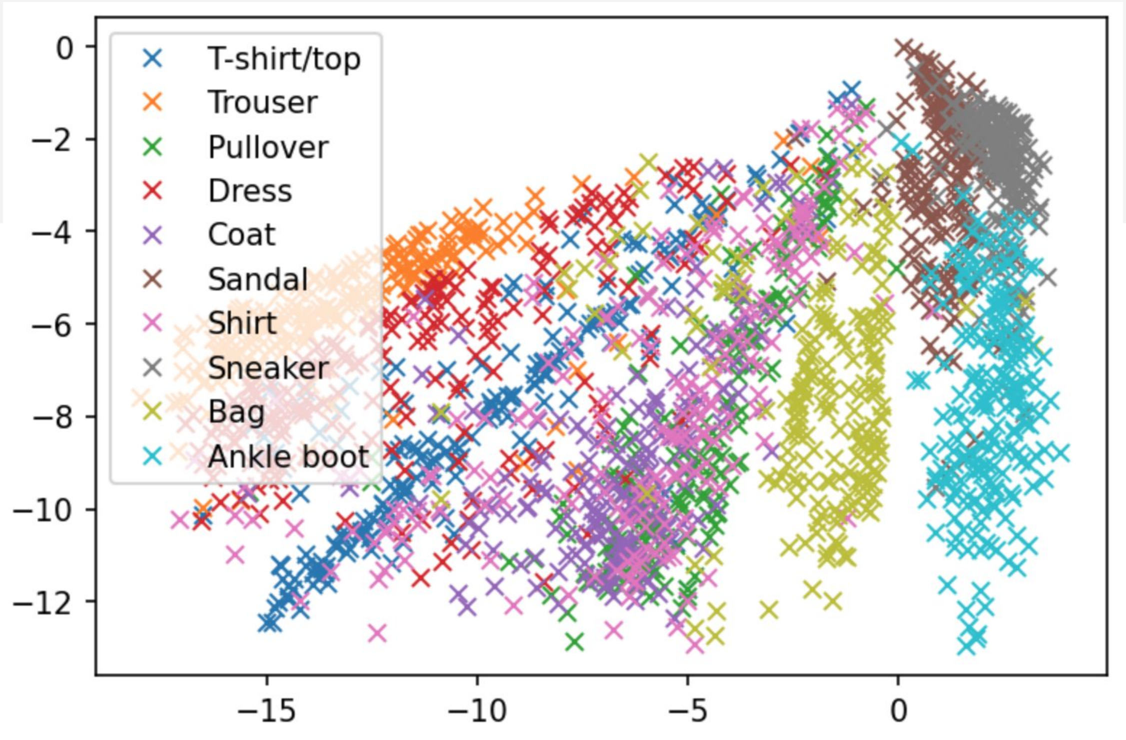
입력 출력 비교

```
f, a = plt.subplots(2, 10, figsize=(10,2))
for i in range(10):
    img = x[i].view(28,28).cpu().numpy()
    a[0][i].imshow(img, cmap='gray')

    img = decoded[i].view(28,28).cpu().numpy()
    a[1][i].imshow(img, cmap='gray')

plt.show()
```



차원축소 결과 시각화

```python
for i in range(10):
    vals = encoder(x)[trainset.targets == i][:200].cpu()
    plt.plot(vals[:,0], vals[:,1], 'x', label=CLASSES[i])

plt.legend()
plt.show()
```



Question?