201600282 엄기산

CNN으로 패션 아이템 구분하기

In [1]:

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

In [2]:

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

In [3]:

```
EPOCHS     = 40
BATCH_SIZE = 64
```

데이터셋 불러오기

In [4]:

```
train_loader = torch.utils.data.DataLoader(
    datasets.FashionMNIST('./.data',
                   train=True,
                   download=True,
                   transform=transforms.Compose([
                       transforms.ToTensor(),
                       transforms.Normalize((0.2860,), (0.3205,))
                   ])),
    batch_size=BATCH_SIZE, shuffle=True)
test_loader = torch.utils.data.DataLoader(
    datasets.FashionMNIST('./.data',
                   train=False,
                   transform=transforms.Compose([
                       transforms.ToTensor(),
                       transforms.Normalize((0.2860,), (0.3205,))
                   ])),
    batch_size=BATCH_SIZE, shuffle=True)
```

```
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-id
x3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-id
x3-ubyte.gz to ./.data₩FashionMNIST₩raw₩train-images-idx3-ubyte.gz

Extracting ./.data₩FashionMNIST₩raw₩train-images-idx3-ubyte.gz to ./.data₩FashionMNIST
₩raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-id
x1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-id
x1-ubyte.gz to ./.data₩FashionMNIST₩raw₩train-labels-idx1-ubyte.gz

Extracting ./.data₩FashionMNIST₩raw₩train-labels-idx1-ubyte.gz to ./.data₩FashionMNIST
₩raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx
3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx
3-ubyte.gz to ./.data₩FashionMNIST₩raw₩t10k-images-idx3-ubyte.gz

Extracting ./.data₩FashionMNIST₩raw₩t10k-images-idx3-ubyte.gz to ./.data₩FashionMNIST
₩raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx
```

```
1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx
1-ubyte.gz to ./.data₩FashionMNIST₩raw₩t10k-labels-idx1-ubyte.gz

Extracting ./.data₩FashionMNIST₩raw₩t10k-labels-idx1-ubyte.gz to ./.data₩FashionMNIST
₩raw

Processing...
C:₩anaconda₩lib₩site-packages₩torchvision₩datasets₩mnist.py:502: UserWarning: The give
n NumPy array is not writeable, and PyTorch does not support non-writeable tensors. Th
is means you can write to the underlying (supposedly non-writeable) NumPy array using
the tensor. You may want to copy the array to protect its data or make it writeable be
fore converting it to a tensor. This type of warning will be suppressed for the rest o
f this program. (Triggered internally at  ..₩torch₩csrc₩utils₩tensor_numpy.cpp:143.)
  return torch.from_numpy(parsed.astype(m[2], copy=False)).view(*s)
Done!
```

뉴럴넷으로 Fashion MNIST 학습하기

In [5]:

```python
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 10, kernel_size=5)
        self.conv2 = nn.Conv2d(10, 20, kernel_size=5)
        self.conv2_drop = nn.Dropout2d()
        self.fc1 = nn.Linear(320, 50)
        self.fc2 = nn.Linear(50, 10)

    def forward(self, x):
        x = F.relu(F.max_pool2d(self.conv1(x), 2))
        x = F.relu(F.max_pool2d(self.conv2_drop(self.conv2(x)), 2))
        x = x.view(-1, 320)
        x = F.relu(self.fc1(x))
        x = F.dropout(x, training=self.training)
        x = self.fc2(x)
        return x
```

하이퍼파라미터

In [6]:

```python
model     = Net().to(DEVICE)
optimizer = optim.SGD(model.parameters(), lr=0.01, momentum=0.5)
```

학습하기

In [7]:

```python
def train(model, train_loader, optimizer, epoch):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        data, target = data.to(DEVICE), target.to(DEVICE)
        optimizer.zero_grad()
        output = model(data)
        loss = F.cross_entropy(output, target)
        loss.backward()
        optimizer.step()

        if batch_idx % 200 == 0:
            print('Train Epoch: {} [{}/{} ({:.0f}%)]₩tLoss: {:.6f}'.format(
                epoch, batch_idx * len(data), len(train_loader.dataset),
                100. * batch_idx / len(train_loader), loss.item()))
```

테스트하기

In [8]:

```
def evaluate(model, test_loader):
    model.eval()
    test_loss = 0
    correct = 0
    with torch.no_grad():
        for data, target in test_loader:
            data, target = data.to(DEVICE), target.to(DEVICE)
            output = model(data)

            # 배치 오차를 합산
            test_loss += F.cross_entropy(output, target,
                                         reduction='sum').item()

            # 가장 높은 값을 가진 인덱스가 바로 예측값
            pred = output.max(1, keepdim=True)[1]
            correct += pred.eq(target.view_as(pred)).sum().item()

    test_loss /= len(test_loader.dataset)
    test_accuracy = 100. * correct / len(test_loader.dataset)
    return test_loss, test_accuracy
```

코드 돌려보기

In [9]:

```
for epoch in range(1, EPOCHS + 1):
    train(model, train_loader, optimizer, epoch)
    test_loss, test_accuracy = evaluate(model, test_loader)

    print('[{}] Test Loss: {:.4f}, Accuracy: {:.2f}%'.format(
          epoch, test_loss, test_accuracy))
```

```
Train Epoch: 1 [0/60000 (0%)]   Loss: 2.302486
Train Epoch: 1 [12800/60000 (21%)]      Loss: 1.491955
Train Epoch: 1 [25600/60000 (43%)]      Loss: 0.948550
Train Epoch: 1 [38400/60000 (64%)]      Loss: 1.094360
Train Epoch: 1 [51200/60000 (85%)]      Loss: 0.724710
[1] Test Loss: 0.6436, Accuracy: 75.01%
Train Epoch: 2 [0/60000 (0%)]   Loss: 1.082100
Train Epoch: 2 [12800/60000 (21%)]      Loss: 1.003064
Train Epoch: 2 [25600/60000 (43%)]      Loss: 0.841521
Train Epoch: 2 [38400/60000 (64%)]      Loss: 0.644387
Train Epoch: 2 [51200/60000 (85%)]      Loss: 0.760181
[2] Test Loss: 0.5645, Accuracy: 77.84%
Train Epoch: 3 [0/60000 (0%)]   Loss: 0.685786
Train Epoch: 3 [12800/60000 (21%)]      Loss: 0.562801
Train Epoch: 3 [25600/60000 (43%)]      Loss: 0.622043
Train Epoch: 3 [38400/60000 (64%)]      Loss: 0.733132
Train Epoch: 3 [51200/60000 (85%)]      Loss: 0.608670
[3] Test Loss: 0.5237, Accuracy: 80.07%
Train Epoch: 4 [0/60000 (0%)]   Loss: 0.601722
Train Epoch: 4 [12800/60000 (21%)]      Loss: 0.467859
Train Epoch: 4 [25600/60000 (43%)]      Loss: 0.506944
Train Epoch: 4 [38400/60000 (64%)]      Loss: 0.578434
Train Epoch: 4 [51200/60000 (85%)]      Loss: 0.598627
[4] Test Loss: 0.4906, Accuracy: 80.62%
Train Epoch: 5 [0/60000 (0%)]   Loss: 0.596970
Train Epoch: 5 [12800/60000 (21%)]      Loss: 0.525554
Train Epoch: 5 [25600/60000 (43%)]      Loss: 0.623707
Train Epoch: 5 [38400/60000 (64%)]      Loss: 0.742038
Train Epoch: 5 [51200/60000 (85%)]      Loss: 0.588632
[5] Test Loss: 0.4711, Accuracy: 82.63%
Train Epoch: 6 [0/60000 (0%)]   Loss: 0.519649
Train Epoch: 6 [12800/60000 (21%)]      Loss: 0.744802
Train Epoch: 6 [25600/60000 (43%)]      Loss: 0.615717
Train Epoch: 6 [38400/60000 (64%)]      Loss: 0.615347
Train Epoch: 6 [51200/60000 (85%)]      Loss: 0.827481
[6] Test Loss: 0.4541, Accuracy: 81.88%
```

```
Train Epoch: 7 [0/60000 (0%)]   Loss: 0.562150
Train Epoch: 7 [12800/60000 (21%)]      Loss: 0.393152
Train Epoch: 7 [25600/60000 (43%)]      Loss: 0.568423
Train Epoch: 7 [38400/60000 (64%)]      Loss: 0.423217
Train Epoch: 7 [51200/60000 (85%)]      Loss: 0.511879
[7] Test Loss: 0.4379, Accuracy: 83.67%
Train Epoch: 8 [0/60000 (0%)]   Loss: 0.564042
Train Epoch: 8 [12800/60000 (21%)]      Loss: 0.435000
Train Epoch: 8 [25600/60000 (43%)]      Loss: 0.496346
Train Epoch: 8 [38400/60000 (64%)]      Loss: 0.807330
Train Epoch: 8 [51200/60000 (85%)]      Loss: 0.350991
[8] Test Loss: 0.4244, Accuracy: 83.87%
Train Epoch: 9 [0/60000 (0%)]   Loss: 0.632995
Train Epoch: 9 [12800/60000 (21%)]      Loss: 0.673107
Train Epoch: 9 [25600/60000 (43%)]      Loss: 0.562891
Train Epoch: 9 [38400/60000 (64%)]      Loss: 0.385264
Train Epoch: 9 [51200/60000 (85%)]      Loss: 0.508223
[9] Test Loss: 0.4127, Accuracy: 85.00%
Train Epoch: 10 [0/60000 (0%)]  Loss: 0.558427
Train Epoch: 10 [12800/60000 (21%)]     Loss: 0.523113
Train Epoch: 10 [25600/60000 (43%)]     Loss: 0.448160
Train Epoch: 10 [38400/60000 (64%)]     Loss: 0.446747
Train Epoch: 10 [51200/60000 (85%)]     Loss: 0.467985
[10] Test Loss: 0.4018, Accuracy: 84.89%
Train Epoch: 11 [0/60000 (0%)]  Loss: 0.634658
Train Epoch: 11 [12800/60000 (21%)]     Loss: 0.496063
Train Epoch: 11 [25600/60000 (43%)]     Loss: 0.315328
Train Epoch: 11 [38400/60000 (64%)]     Loss: 0.257633
Train Epoch: 11 [51200/60000 (85%)]     Loss: 0.445980
[11] Test Loss: 0.3934, Accuracy: 85.81%
Train Epoch: 12 [0/60000 (0%)]  Loss: 0.625790
Train Epoch: 12 [12800/60000 (21%)]     Loss: 0.616683
Train Epoch: 12 [25600/60000 (43%)]     Loss: 0.365179
Train Epoch: 12 [38400/60000 (64%)]     Loss: 0.556488
Train Epoch: 12 [51200/60000 (85%)]     Loss: 0.582889
[12] Test Loss: 0.3871, Accuracy: 85.50%
Train Epoch: 13 [0/60000 (0%)]  Loss: 0.354353
Train Epoch: 13 [12800/60000 (21%)]     Loss: 0.602171
Train Epoch: 13 [25600/60000 (43%)]     Loss: 0.583165
Train Epoch: 13 [38400/60000 (64%)]     Loss: 0.400351
Train Epoch: 13 [51200/60000 (85%)]     Loss: 0.479636
[13] Test Loss: 0.3729, Accuracy: 86.52%
Train Epoch: 14 [0/60000 (0%)]  Loss: 0.447943
Train Epoch: 14 [12800/60000 (21%)]     Loss: 0.551259
Train Epoch: 14 [25600/60000 (43%)]     Loss: 0.551808
Train Epoch: 14 [38400/60000 (64%)]     Loss: 0.387308
Train Epoch: 14 [51200/60000 (85%)]     Loss: 0.585400
[14] Test Loss: 0.3719, Accuracy: 86.28%
Train Epoch: 15 [0/60000 (0%)]  Loss: 0.561793
Train Epoch: 15 [12800/60000 (21%)]     Loss: 0.383034
Train Epoch: 15 [25600/60000 (43%)]     Loss: 0.472087
Train Epoch: 15 [38400/60000 (64%)]     Loss: 0.437218
Train Epoch: 15 [51200/60000 (85%)]     Loss: 0.788938
[15] Test Loss: 0.3600, Accuracy: 86.30%
Train Epoch: 16 [0/60000 (0%)]  Loss: 0.373995
Train Epoch: 16 [12800/60000 (21%)]     Loss: 0.604110
Train Epoch: 16 [25600/60000 (43%)]     Loss: 0.682655
Train Epoch: 16 [38400/60000 (64%)]     Loss: 0.597121
Train Epoch: 16 [51200/60000 (85%)]     Loss: 0.359356
[16] Test Loss: 0.3662, Accuracy: 86.57%
Train Epoch: 17 [0/60000 (0%)]  Loss: 0.594334
Train Epoch: 17 [12800/60000 (21%)]     Loss: 0.426389
Train Epoch: 17 [25600/60000 (43%)]     Loss: 0.403571
Train Epoch: 17 [38400/60000 (64%)]     Loss: 0.393097
Train Epoch: 17 [51200/60000 (85%)]     Loss: 0.462633
[17] Test Loss: 0.3533, Accuracy: 86.95%
Train Epoch: 18 [0/60000 (0%)]  Loss: 0.395199
Train Epoch: 18 [12800/60000 (21%)]     Loss: 0.386346
Train Epoch: 18 [25600/60000 (43%)]     Loss: 0.621879
Train Epoch: 18 [38400/60000 (64%)]     Loss: 0.603223
Train Epoch: 18 [51200/60000 (85%)]     Loss: 0.481655
[18] Test Loss: 0.3483, Accuracy: 87.18%
Train Epoch: 19 [0/60000 (0%)]  Loss: 0.564377
Train Epoch: 19 [12800/60000 (21%)]     Loss: 0.467231
Train Epoch: 19 [25600/60000 (43%)]     Loss: 0.380121
Train Epoch: 19 [38400/60000 (64%)]     Loss: 0.388848
Train Epoch: 19 [51200/60000 (85%)]     Loss: 0.296173
[19] Test Loss: 0.3461, Accuracy: 87.04%
Train Epoch: 20 [0/60000 (0%)]  Loss: 0.502762
Train Epoch: 20 [12800/60000 (21%)]     Loss: 0.685857
Train Epoch: 20 [25600/60000 (43%)]     Loss: 0.367786
Train Epoch: 20 [38400/60000 (64%)]     Loss: 0.412866
Train Epoch: 20 [51200/60000 (85%)]     Loss: 0.301552
[20] Test Loss: 0.3463, Accuracy: 87.07%
Train Epoch: 21 [0/60000 (0%)]  Loss: 0.477714
Train Epoch: 21 [12800/60000 (21%)]     Loss: 0.431276
Train Epoch: 21 [25600/60000 (43%)]     Loss: 0.433807
Train Epoch: 21 [38400/60000 (64%)]     Loss: 0.456175
Train Epoch: 21 [51200/60000 (85%)]     Loss: 0.350189
[21] Test Loss: 0.3412, Accuracy: 87.29%
Train Epoch: 22 [0/60000 (0%)]  Loss: 0.533464
Train Epoch: 22 [12800/60000 (21%)]     Loss: 0.594296
Train Epoch: 22 [25600/60000 (43%)]     Loss: 0.473888
Train Epoch: 22 [38400/60000 (64%)]     Loss: 0.519490
Train Epoch: 22 [51200/60000 (85%)]     Loss: 0.522969
[22] Test Loss: 0.3432, Accuracy: 87.21%
Train Epoch: 23 [0/60000 (0%)]  Loss: 0.365857
Train Epoch: 23 [12800/60000 (21%)]     Loss: 0.258735
Train Epoch: 23 [25600/60000 (43%)]     Loss: 0.431100
Train Epoch: 23 [38400/60000 (64%)]     Loss: 0.262442
Train Epoch: 23 [51200/60000 (85%)]     Loss: 0.524415
[23] Test Loss: 0.3363, Accuracy: 87.64%
Train Epoch: 24 [0/60000 (0%)]  Loss: 0.326386
Train Epoch: 24 [12800/60000 (21%)]     Loss: 0.391044
Train Epoch: 24 [25600/60000 (43%)]     Loss: 0.444202
Train Epoch: 24 [38400/60000 (64%)]     Loss: 0.510439
Train Epoch: 24 [51200/60000 (85%)]     Loss: 0.376127
[24] Test Loss: 0.3377, Accuracy: 87.62%
Train Epoch: 25 [0/60000 (0%)]  Loss: 0.451990
Train Epoch: 25 [12800/60000 (21%)]     Loss: 0.440438
Train Epoch: 25 [25600/60000 (43%)]     Loss: 0.381405
Train Epoch: 25 [38400/60000 (64%)]     Loss: 0.575301
Train Epoch: 25 [51200/60000 (85%)]     Loss: 0.403210
[25] Test Loss: 0.3300, Accuracy: 87.50%
Train Epoch: 26 [0/60000 (0%)]  Loss: 0.524264
Train Epoch: 26 [12800/60000 (21%)]     Loss: 0.455129
Train Epoch: 26 [25600/60000 (43%)]     Loss: 0.371129
Train Epoch: 26 [38400/60000 (64%)]     Loss: 0.505873
Train Epoch: 26 [51200/60000 (85%)]     Loss: 0.565969
[26] Test Loss: 0.3292, Accuracy: 87.79%
Train Epoch: 27 [0/60000 (0%)]  Loss: 0.733786
Train Epoch: 27 [12800/60000 (21%)]     Loss: 0.370577
Train Epoch: 27 [25600/60000 (43%)]     Loss: 0.592899
Train Epoch: 27 [38400/60000 (64%)]     Loss: 0.395124
Train Epoch: 27 [51200/60000 (85%)]     Loss: 0.415988
[27] Test Loss: 0.3298, Accuracy: 87.65%
Train Epoch: 28 [0/60000 (0%)]  Loss: 0.299635
Train Epoch: 28 [12800/60000 (21%)]     Loss: 0.403142
Train Epoch: 28 [25600/60000 (43%)]     Loss: 0.313814
Train Epoch: 28 [38400/60000 (64%)]     Loss: 0.341872
Train Epoch: 28 [51200/60000 (85%)]     Loss: 0.654585
[28] Test Loss: 0.3286, Accuracy: 87.81%
Train Epoch: 29 [0/60000 (0%)]  Loss: 0.551997
Train Epoch: 29 [12800/60000 (21%)]     Loss: 0.474852
Train Epoch: 29 [25600/60000 (43%)]     Loss: 0.528627
Train Epoch: 29 [38400/60000 (64%)]     Loss: 0.428916
Train Epoch: 29 [51200/60000 (85%)]     Loss: 0.511192
[29] Test Loss: 0.3322, Accuracy: 87.38%
```

```
Train Epoch: 30 [0/60000 (0%)]  Loss: 0.358097
Train Epoch: 30 [12800/60000 (21%)]     Loss: 0.313225
Train Epoch: 30 [25600/60000 (43%)]     Loss: 0.669700
Train Epoch: 30 [38400/60000 (64%)]     Loss: 0.352734
Train Epoch: 30 [51200/60000 (85%)]     Loss: 0.489272
[30] Test Loss: 0.3200, Accuracy: 88.24%
Train Epoch: 31 [0/60000 (0%)]  Loss: 0.415663
Train Epoch: 31 [12800/60000 (21%)]     Loss: 0.441079
Train Epoch: 31 [25600/60000 (43%)]     Loss: 0.376632
Train Epoch: 31 [38400/60000 (64%)]     Loss: 0.429233
Train Epoch: 31 [51200/60000 (85%)]     Loss: 0.439819
[31] Test Loss: 0.3228, Accuracy: 87.88%
Train Epoch: 32 [0/60000 (0%)]  Loss: 0.379033
Train Epoch: 32 [12800/60000 (21%)]     Loss: 0.488387
Train Epoch: 32 [25600/60000 (43%)]     Loss: 0.386960
Train Epoch: 32 [38400/60000 (64%)]     Loss: 0.351505
Train Epoch: 32 [51200/60000 (85%)]     Loss: 0.454919
[32] Test Loss: 0.3201, Accuracy: 88.21%
Train Epoch: 33 [0/60000 (0%)]  Loss: 0.381017
Train Epoch: 33 [12800/60000 (21%)]     Loss: 0.272054
Train Epoch: 33 [25600/60000 (43%)]     Loss: 0.580554
Train Epoch: 33 [38400/60000 (64%)]     Loss: 0.669154
Train Epoch: 33 [51200/60000 (85%)]     Loss: 0.434783
[33] Test Loss: 0.3239, Accuracy: 87.97%
Train Epoch: 34 [0/60000 (0%)]  Loss: 0.433193
Train Epoch: 34 [12800/60000 (21%)]     Loss: 0.383458
Train Epoch: 34 [25600/60000 (43%)]     Loss: 0.240377
Train Epoch: 34 [38400/60000 (64%)]     Loss: 0.318115
Train Epoch: 34 [51200/60000 (85%)]     Loss: 0.291008
[34] Test Loss: 0.3214, Accuracy: 87.87%
Train Epoch: 35 [0/60000 (0%)]  Loss: 0.261930
Train Epoch: 35 [12800/60000 (21%)]     Loss: 0.309770
Train Epoch: 35 [25600/60000 (43%)]     Loss: 0.360767
Train Epoch: 35 [38400/60000 (64%)]     Loss: 0.382458
Train Epoch: 35 [51200/60000 (85%)]     Loss: 0.520428
[35] Test Loss: 0.3131, Accuracy: 88.49%
Train Epoch: 36 [0/60000 (0%)]  Loss: 0.442000
Train Epoch: 36 [12800/60000 (21%)]     Loss: 0.374192
Train Epoch: 36 [25600/60000 (43%)]     Loss: 0.489146
Train Epoch: 36 [38400/60000 (64%)]     Loss: 0.333775
Train Epoch: 36 [51200/60000 (85%)]     Loss: 0.365784
[36] Test Loss: 0.3277, Accuracy: 87.52%
Train Epoch: 37 [0/60000 (0%)]  Loss: 0.396576
Train Epoch: 37 [12800/60000 (21%)]     Loss: 0.334724
Train Epoch: 37 [25600/60000 (43%)]     Loss: 0.389094
Train Epoch: 37 [38400/60000 (64%)]     Loss: 0.261944
Train Epoch: 37 [51200/60000 (85%)]     Loss: 0.349461
[37] Test Loss: 0.3153, Accuracy: 88.33%
Train Epoch: 38 [0/60000 (0%)]  Loss: 0.301890
Train Epoch: 38 [12800/60000 (21%)]     Loss: 0.368030
Train Epoch: 38 [25600/60000 (43%)]     Loss: 0.510675
Train Epoch: 38 [38400/60000 (64%)]     Loss: 0.357954
Train Epoch: 38 [51200/60000 (85%)]     Loss: 0.587082
[38] Test Loss: 0.3119, Accuracy: 88.39%
Train Epoch: 39 [0/60000 (0%)]  Loss: 0.310375
Train Epoch: 39 [12800/60000 (21%)]     Loss: 0.279881
Train Epoch: 39 [25600/60000 (43%)]     Loss: 0.404495
Train Epoch: 39 [38400/60000 (64%)]     Loss: 0.512330
Train Epoch: 39 [51200/60000 (85%)]     Loss: 0.368709
[39] Test Loss: 0.3197, Accuracy: 88.06%
Train Epoch: 40 [0/60000 (0%)]  Loss: 0.190655
Train Epoch: 40 [12800/60000 (21%)]     Loss: 0.353446
Train Epoch: 40 [25600/60000 (43%)]     Loss: 0.455667
Train Epoch: 40 [38400/60000 (64%)]     Loss: 0.328713
Train Epoch: 40 [51200/60000 (85%)]     Loss: 0.311599
[40] Test Loss: 0.3128, Accuracy: 88.24%
```

201600282 엄기산