

인공지능 개론

L09.1 MNIST with DNN

국민대학교

소프트웨어융합대학원

박하명

Contents

- ❖ Softmax 로 MNIST 문제 풀어보기
- ❖ GPU 사용하기
- ❖ Neural Net. 으로 MNIST 문제 풀어보기

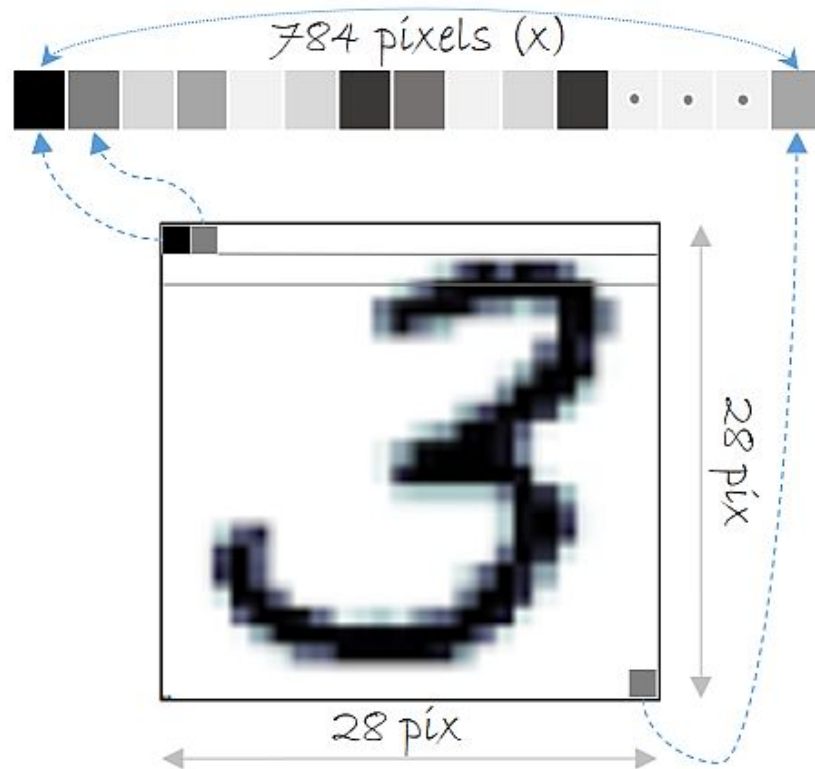
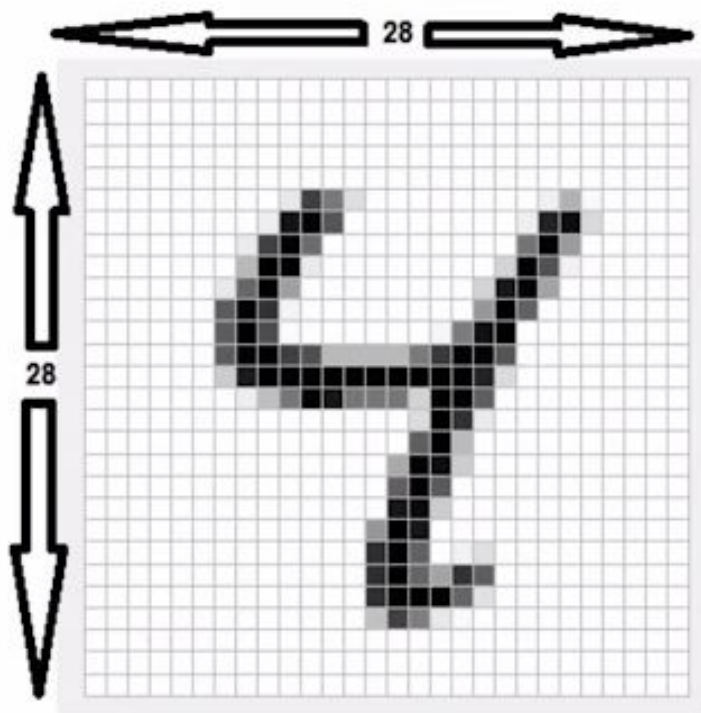
MNIST 데이터 (복습)

- 0부터 9까지의 손글씨 이미지 데이터
- 훈련데이터 6만 개, 테스트데이터 1만 개
- MNIST 문제: 글자 이미지가 주어지면, 0~9 중에 어떤 숫자인지 맞추기



MNIST 데이터 (복습)

- 이미지 크기: 28 픽셀 x 28 픽셀 = 784 픽셀



MNIST 데이터 불러오기 (복습)

```
import torchvision.datasets as datasets
import torchvision.transforms as transforms

mnist_train = datasets.MNIST(root= 'MNIST_data/', # 데이터 위치
                             train=True, # True: 훈련 데이터, False: 테스트 데이터
                             transform=transforms.ToTensor(), # pytorch Tensor로 변환
                             download=True # 데이터가 없을 경우 다운로드
                             )

mnist_test = datasets.MNIST(root= 'MNIST_data/',
                             train=False,
                             transform=transforms.ToTensor(),
                             download=True
                             )
```

MNIST 데이터 불러오기 (복습)

- 불러온 MNIST 학습 및 테스트 데이터를 학습이 가능한 형태로 변환

```
x_train = mnist_train.data.view(-1, 28*28).float()  
y_train = mnist_train.targets  
x_test = mnist_test.data.view(-1, 28*28).float()  
y_test = mnist_test.targets
```

Softmax Regression (복습)

```
import torch
import torch.nn as nn
import torch.nn.functional as F

model = nn.Linear(28*28, 10)

optimizer = torch.optim.Adam(model.parameters())

for epoch in range(201):
    z = model(x_train)
    cost = F.cross_entropy(z, y_train)

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

Softmax Regression (복습)

- 10 epoch 마다 cost, test cost, accuracy 출력하기

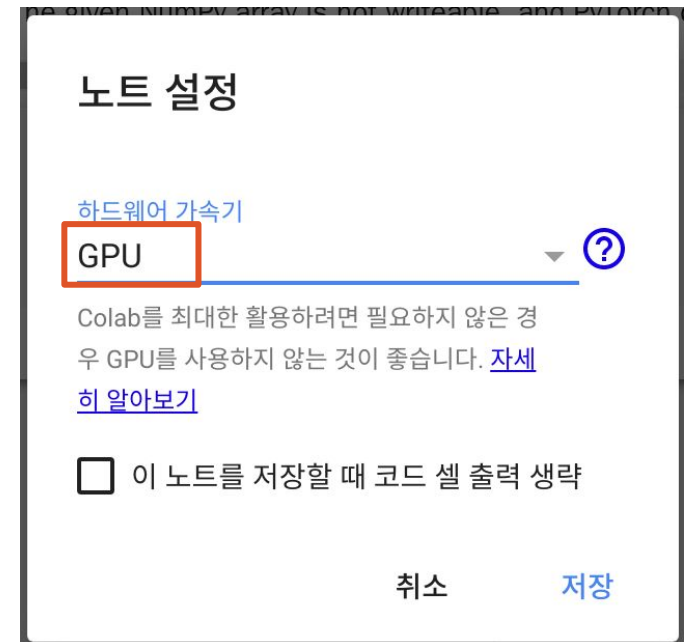
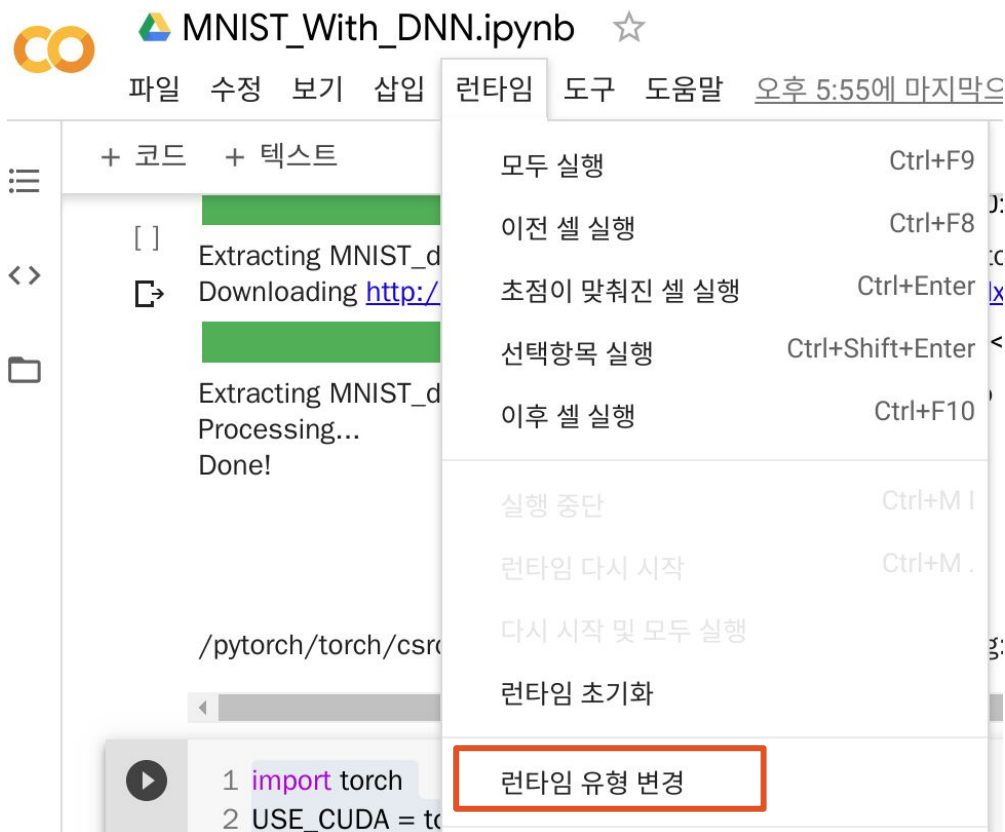
```
if epoch % 10 == 0:
    with torch.no_grad():
        z_test = model(x_test)
        cost_test = F.cross_entropy(z_test, y_test)
        accuracy = 100*(y_test == z_test.argmax(1)).sum().item()/len(z_test)
        print("epoch: {}, cost: {:.6f}, test: {:.6f}, acc: {:.1f}" .format(
            epoch, cost, cost_test, accuracy))
```


Contents

- ❖ Softmax 로 MNIST 문제 풀어보기
- ❖ **GPU 사용하기**
- ❖ Neural Net. 으로 MNIST 문제 풀어보기

GPU 사용하기

- Colab에서 GPU를 사용하기 위해 런타임 유형을 GPU로 변경



GPU 사용하기

- pytorch에서 CUDA 사용 설정

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

GPU 사용하기

- 데이터와 모델을 GPU의 메모리에 올리기

```
x_train = mnist_train.data.view(-1, 28*28).float().to(DEVICE)
y_train = mnist_train.targets.to(DEVICE)
x_test = mnist_test.data.view(-1, 28*28).float().to(DEVICE)
y_test = mnist_test.targets.to(DEVICE)

model = nn.Linear(28*28, 10).to(DEVICE)
```

Contents

- ❖ Softmax 로 MNIST 문제 풀어보기
- ❖ GPU 사용하기
- ❖ **Neural Net. 으로 MNIST 문제 풀어보기**

Softmax → Neural Net.

- $784 \rightarrow 256 \rightarrow 256 \rightarrow 10$ 의 층으로 구성된 neural net. 모델 생성
- 활성화함수로 ReLU 활용

```
layer1 = nn.Linear(28*28, 256)
```

```
layer2 = nn.Linear(256, 256)
```

```
layer3 = nn.Linear(256, 10)
```

```
relu = nn.ReLU()
```

```
model = nn.Sequential(layer1, relu, layer2, relu, layer3).to(DEVICE)
```

```
epoch: 200, cost: 0.064516, test: 0.136335, acc: 96.1
```

가중치 초기화

- Xavier 초기화 사용하기

```
nn.init.xavier_normal_(layer1.weight, 0.1)
nn.init.xavier_normal_(layer2.weight, 0.1)
nn.init.xavier_normal_(layer3.weight, 0.1)
```

epoch: 200, cost: 0.002548, test: 0.102989, **acc: 97.7**

Dropout

- 모델에 Dropout 추가 (제거할 노드 비율 = 0.2)

```
...
dropout = nn.Dropout(0.2)

model = nn.Sequential(layer1, dropout, relu, layer2, dropout, relu, layer3).to(DEVICE)

...
for epoch in range(201):
    model.train() # 학습 모드로 전환 (dropout 켜기)
    z = model(x_train)
    ...

    if epoch % 10 == 0:
        model.eval() # 평가 모드로 전환 (dropout 끄기 = dropout 비율 0으로 설정)
        with torch.no_grad():
            ...
```

epoch: 200, cost: 0.018812, test: 0.077772, **acc: 98.0**

Question?