

인공지능 개론

L04.2 MNIST with Softmax

국민대학교
소프트웨어융합대학원
박하명

Contents

- ❖ Softmax 로 MNIST 문제 풀어보기
- ❖ 학습 결과 테스트 해보기

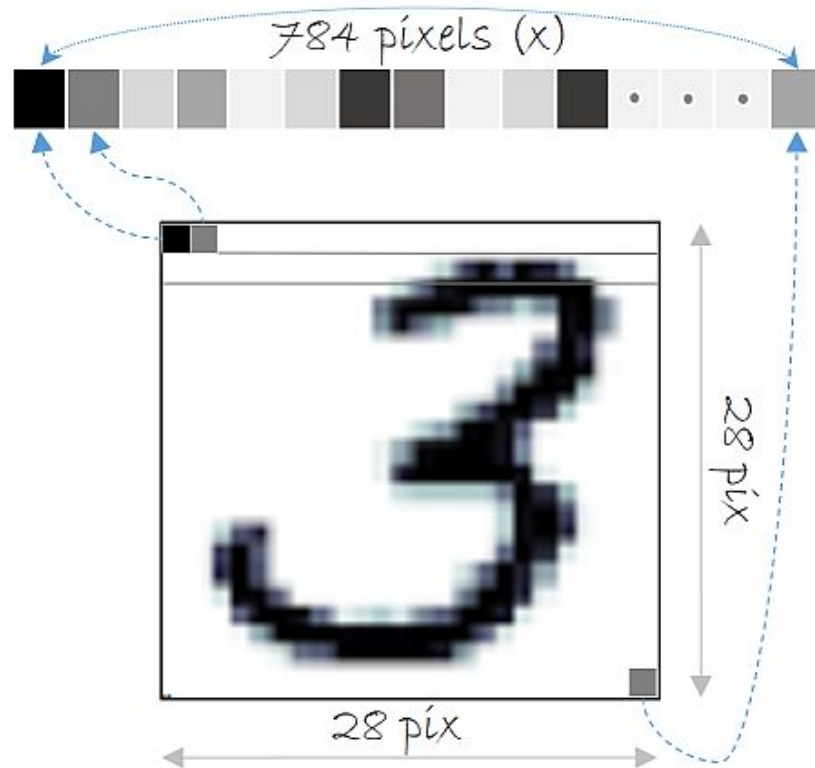
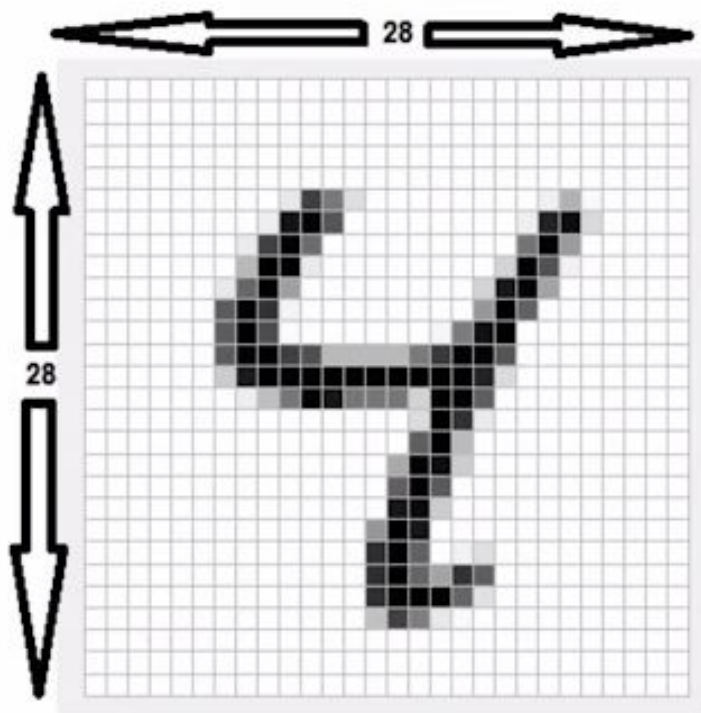
MNIST 데이터

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



MNIST 데이터

- 이미지 크기: 28 픽셀 x 28 픽셀 = 784 픽셀



MNIST 데이터 불러오기

```
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

- 불러온 MNIST 학습 데이터를 학습이 가능한 형태로 변환

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

Softmax Regression

```
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(101):
    z = model(x_train)
    cost = F.cross_entropy(z, y_train)

    optimizer.zero_grad()
    cost.backward()
    optimizer.step()
    if epoch % 10 == 0:
        print("epoch: {}, cost: {:.6f}".format(epoch, cost.item()))
```

Contents

- ❖ Softmax 로 MNIST 문제 풀어보기
- ❖ 학습 결과 테스트 해보기

학습 결과 테스트 해보기

- 테스트 데이터 읽기

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

학습 결과 테스트 해보기

- 테스트 데이터 중 임의로 하나를 골라서 실제값, 예측값 확인
- matplotlib를 이용하여 그림 출력

```
import matplotlib.pyplot as plt
import random

with torch.no_grad():
    r = random.randint(0, len(x_test)-1)
    x_single_test = x_test[r].unsqueeze(0)
    y_single_test = y_test[r].unsqueeze(0)

    single_prediction = model(x_single_test).argmax(1)
    print('True: {}, Pred: {}'.format(y_single_test.item(), single_prediction.item()))

plt.imshow(x_single_test.view(28, 28), cmap='Greys')
plt.show()
```

정확도 구하기

- 정확도: 맞은 개수/ 전체 개수

```
with torch.no_grad():  
    z = model(x_test).argmax(1)  
    accuracy = (y_test==z).sum().item() / len(z)  
    print("accuracy: ", accuracy)
```

Question?