

# 2021254010 이지호 딥러닝 실제 11주차 실습 과제

## 동작 설명 (주석으로 설명)

```
1 from sklearn.datasets import fetch_openml
2 from sklearn.neural_network import MLPClassifier
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6 # MNIST 데이터셋을 읽고 훈련 집합과 테스트 집합으로 분할
7 mnist = fetch_openml("mnist_784") # 사이킷런의 fetch_openml을 이용해서 https://openml.org/ 에서 mnist 데이터셋을 가져옵니다.
8 mnist.data = mnist.data / 255.0 # 데이터를 정규화 하여 0~1사이의 값으로 바꿔줍니다.
9 x_train = mnist.data[:60000] # 60000개는 학습 데이터로
10 x_test = mnist.data[60000:] # 나머지는 테스트 데이터로 나눠 줍니다.
11 y_train = np.int16(mnist.target[:60000]) # 레이블도 똑같이 해줍니다.
12 y_test = np.int16(mnist.target[60000:])
13
14 # MLP 분류기 모델을 학습
15 mlp = MLPClassifier( # MLP 분류기 객체를 생성 합니다.
16     hidden_layer_sizes=(100), # 히든레이어는 100으로
17     learning_rate_init=0.001, # 러닝레이트는 0.001로 시작 합니다.
18     batch_size=512, # 배치 사이즈는 512로 설정 합니다.
19     max_iter=300, # 최대 반복은 300회로 설정 합니다.
20     solver="adam", # adam 옵티마이저를 사용합니다.
21     verbose=True, # 진행 메시지를 출력 합니다.
22 )
23 mlp.fit(x_train, y_train) # 학습데이터로 학습을 진행 합니다.
24
25 # 테스트 집합으로 예측
26 res = mlp.predict(x_test) # 테스트셋으로 평가 합니다.
27
28 # 혼동 행렬
29 conf = np.zeros((10, 10), dtype=np.int16) # 혼동 행렬을 생성해줍니다.
30 for i in range(len(res)):
31     conf[res[i][y_test[i]] += 1 # 테스트 데이터 결과를 혼동 행렬에 추가해 줍니다.
32 print(conf) # 결과를 출력해 줍니다.
33
34 # 정확률 계산
35 no_correct = 0 # 결과를 카운트할 변수를 정합니다.
36 for i in range(10):
37     no_correct += conf[i][i] # 정답을 카운트 합니다.
38 accuracy = no_correct / len(res) # (정답수 / 전체)를 계산 합니다.
39 print("테스트 집합에 대한 정확률은", accuracy * 100, "%입니다.")
40
```

**batch size** 를 128 로 하고, 은닉층 사이즈를 50 인 경우에 수행하여 결과를 비교하시오.

히든레이어 100, 배치사이즈 512

```
[[ 970    0    4    0    1    2    6    1    7    0]
 [   0 1122    2    0    0    2    2    2    1    2]
 [   0    4 1005    1    4    0    3    9    1    1]
 [   1    1    4 992    1   10    1    7    4    4]
 [   0    0    1    0 963    2    7    3    4    9]
 [   2    2    0    5    0 868    4    0    2    3]
 [   2    2    2    1    1    2 932    0    1    1]
 [   1    1    7    3    1    2    0 1000    4    6]
 [   4    3    6    3    1    3    3    1 946    4]
 [   0    0    1    5   10    1    0    5    4 979]]
```

테스트 집합에 대한 정확률은 97.77 %입니다. 실행시간 124.0 초. 반복 92회에서 멈춤.

히든레이어 50, 배치사이즈 128

```
[[ 968    0    7    0    1    5    6    0    4    2]
 [   0 1115    1    0    1    0    2    2    0    2]
 [   2    5 994    7    2    0    2    9    4    0]
 [   2    1    4 981    1   16    1    1    8    3]
 [   1    0    1    1 958    2    4    0    3   11]
 [   1    1    0    6    0 856    4    0    3    4]
 [   2    3    3    0    4    4 933    0    1    0]
 [   1    1    8    9    2    4    2 1009    6    4]
 [   3    7   13    3    1    5    4    2 938    5]
 [   0    2    1    3   12    0    0    5    7 978]]
```

테스트 집합에 대한 정확률은 97.3 %입니다. 실행시간 130 초. 반복 80회에서 멈춤.

### **\*결론**

히든레이어가 적으면 더 낮은 정확도를 갖게 되고 학습 속도가 빨라지지만.

배치사이즈가 작아지면 학습시간이 길어지게 된다.