

## ▼ Deep Learning

```
1 from google.colab import drive
2 drive.mount('/content/drive')
3
4 basicpath = '/content/drive/MyDrive/LG_DIC_lecture/1일 차/실습'
5
```

```
1 import pandas as pd
2 import os
3 import numpy as np
4 import matplotlib
5 import matplotlib.pyplot as plt
6 import torch
7 import torch.nn as nn
8 import torch.optim as optim
```

```
1 device = 'cuda' if torch.cuda.is_available() else 'cpu'
```

## ▼ Data import

```
1 path = os.path.join(basicpath, 'Dataset/Classification')
2 train_csv = 'mnist_train.csv'
3 test_csv = 'mnist_test.csv'
4 train_data = pd.read_csv(os.path.join(path, train_csv))
5 test_data = pd.read_csv(os.path.join(path, test_csv))
```

## ▼ Deep learning - classification 모델

### ▼ Pytorch 모델에 입력하기 위한 데이터 변환

```
1 train_data = torch.from_numpy(train_data.values).float()
2 test_data = torch.from_numpy(test_data.values).float()
```

## ▼ 과제 3-1. Training data를 pytorch 모델에 적용하기 위한 data loader를 만드시오

```
1 BATCH_SIZE = 15
```

```
1 #답변
2 #답변
3
4
5
6
7
```

과제 3-2. mnist 전체 분류를 위한 딥러닝 모델을 만들고, 훈련시켜 보시오. 모델과 각종 파라미터는 자유입니다.

```
1 epochs =
2 learning_rate =
3 drop_prob =

1 # Make deeplearning model
2 class DNNModel(nn.Module):
3
4
5
6
7
8
9
10
11
12
13
14

1 # Training model
2 for epoch in range(epochs):
3
4
5
6
7
8
9
10
11
12
13
14
15
```

## ▼ 정확도 판단

## ▼ Confusion matrix

```
1 from sklearn.metrics import confusion_matrix
2 from sklearn.metrics import precision_score, recall_score

1 with torch.no_grad():
2     X_test = test_data[:, 1:].view(-1, 28 * 28).float().to(device)
3     y_test = test_data[:, 0].float()
4
5     prediction = model(X_test).cpu()
6     print(confusion_matrix(torch.argmax(prediction, 1), y_test))
7     print("==Precision==")
8     print(precision_score(torch.argmax(prediction, 1), y_test, average=None))
9     print(precision_score(torch.argmax(prediction, 1), y_test, average='weight
10    print("Recall")
11    print(recall_score(torch.argmax(prediction, 1), y_test, average=None))
12    print(recall_score(torch.argmax(prediction, 1), y_test, average='weighted')
```

