Assignment #1 - Supervised Learning

이름 : 전예찬

학번 : 20194902

소속 : 소프트웨어대학 소프트웨어학부

---

1. 테스트 환경

- Test OS : Windows 11 23H Education edition, Google colab
- python version : 3.12.2

2. 실험 진행

a. Dataset

MNIST dataset

- 클래스 : 0 ~ 9까지 숫자의 총 10개 클래스
- training images / labels : 60000
- test images / labels : 10000
- input image resolution : 28 × 28 with depth 1
- label type : class

CIFAR 10 dataset

- classes : airplane / automobile / bird / cat / deer / dog / frog / horse / ship / truck
- train images / labels : 50000

- test images / labels : 10000
- input image resolution : 32 × 32 with depth 3
- label type : class

```
MNIST original dataset
Train images: torch.Size([60000, 28, 28])
Train labels: (60000,)
Test images: torch.Size([10000, 28, 28])
Test labels: (10000,)

CIFAR-10 original dataset
Train images: (50000, 32, 32, 3)
Train labels: (50000,)
Test images: (10000, 32, 32, 3)
Test labels: (10000,)

MNIST dataset
Train images: (60000, 784)
Train labels: (60000,)
Test images: (10000, 784)
Test labels: (10000,)

CIFAR-10 dataset
Train images: (50000, 3072)
Train labels: (50000,)
Test images: (10000, 3072)
Test labels: (10000,)
```

b. Experimental setup

Decision tree classifier

- classifier 구현체 : scikit learn 라이브러에서 제공하는 DecisionTreeClassifier
- tree depth : 3 / 6 / 9 / 12
- Grid search parameter :
  - min samples split : 2 / 5 / 10
  - min samples leaf : 1 / 2 / 4
  - max leaf nodes : 5 / 10 / None
  - cross validation : 5
- Decision Tree Algorithm for MNIST

Algorithm 1 MNIST Decision Tree with Hyperparameters

Input: MNIST train and test data
Output: Accuracy scores for decision tree on MNIST dataset

Data Loading and Preprocessing:
1: Load MNIST train and test datasets
2: Preprocess images and labels
Algorithm Configuration:
3: Initialize decision tree classifiers with different depths
4: Set hyperparameters grid for grid search
MNIST Dataset Training and Evaluation:
5: **for** i in tree_depth **do**
6: Train decision tree classifier (tree_depth = i)
7: **end for**
8: Print accuracy scores for each decision tree classifier

- Decision Tree Algorithm for CIFAR

Algorithm 1 CIFAR Decision Tree with Hyperparameters

Input: CIFAR-10 train and test data
Output: Accuracy scores for decision tree on CIFAR-10 dataset

Data Loading and Preprocessing:
1: Load CIFAR-10 train and test datasets
2: Preprocess images and labels
Algorithm Configuration:
3: Initialize decision tree classifiers with different depths
4: Set hyperparameters grid for grid search
CIFAR-10 Dataset Training and Evaluation:
5: **for** i in tree_depth **do**
6: Train decision tree classifier (tree_depth = i)
7: **end for**
8: Print accuracy scores for each decision tree classifier

SVM classifier

- classifier 구현체 : scikit learn 라이브러리에서 제공하는 svm 모듈의 SVC사용.
- kernel type : linear / radial basis function
- SVM Train Algorithm for MNIST

Algorithm 1 MNIST SVM with Hyperparameters

Input: MNIST train and test data
Output: Accuracy scores for SVM on MNIST dataset
Data Loading and Preprocessing:
1: Load MNIST train and test datasets
2: Preprocess images and labels
Algorithm Configuration:
3: Initialize SVM classifiers with different kernels
MNIST Dataset Training and Evaluation:
4: Train SVM classifier (kernel = 'linear')
5: Train SVM classifier (kernel = 'rbf')
6: Print accuracy scores for linear and RBF SVM classifiers

- SVM Train Algorithm for CIFAR

Algorithm 1 CIFAR SVM with Hyperparameters

Input: CIFAR-10 train and test data
Output: Accuracy scores for SVM on CIFAR-10 dataset
Data Loading and Preprocessing:
1: Load CIFAR-10 train and test datasets
2: Preprocess images and labels
Algorithm Configuration:
3: Initialize SVM classifiers with different kernels
CIFAR-10 Dataset Training and Evaluation:
4: Train SVM classifier (kernel = 'linear')
5: Train SVM classifier (kernel = 'rbf')
6: Print accuracy scores for linear and RBF SVM classifiers

MNIST dataset에 대해서 local machine에서 decision tree, svm의 학습 동시에 진행

CIFAR dataset에 대해서 local machine(decision tree학습)과 colab(svm 학습)에서 병렬적으로 진행.

c. Results

MNIST

- train / test accuracy on decision tree with depth of 3/6/9/12

```
mnist_dt_df = pd.DataFrame(data={'tree_depth' : tree_depth, 'train_accuracy': mnist_dt_train_accuracy, 'test_accuracy'
mnist_dt_df
#20194902 전예찬
```

[50] Python

	tree_depth	train_accuracy	test_accuracy
0	3	0.491517	0.4953
1	6	0.738250	0.7415
2	9	0.866517	0.8497
3	12	0.938733	0.8783

- train / test accuracy on svm with linear and rbf kernel

```
kernel_type = ["linear", "rbf"]
svm_train_accuracy = [mnist_svm_lin_train_accuracy, mnist_svm_rbf_train_accuracy]
svm_test_accuracy = [mnist_svm_lin_test_accuracy, mnist_svm_rbf_test_accuracy]
mnist_svm_df = pd.DataFrame(data={'kernel_type' : kernel_type, 'train_accuracy' : svm_train_accuracy, 'test_accuracy'
mnist_svm_df
#20194902 전예찬
```

[51] Python

	kernel_type	train_accuracy	test_accuracy
0	linear	0.970733	0.9404
1	rbf	0.989917	0.9792

CIFAR 10

- train / test accuracy on decision tree with depth of 3/6/9/12

```
cifar_dt_df = pd.DataFrame(data={'tree_depth' : tree_depth, 'train_accuracy': cifar_dt_train_accuracy, 'test_accuracy'
cifar_dt_df
#20194902 전예찬
```

[10] Python

	tree_depth	train_accuracy	test_accuracy
0	3	0.23762	0.2394
1	6	0.29588	0.2812
2	9	0.38214	0.3040
3	12	0.52096	0.3031

- train / test accuracy on svm with linear and rbf kernel

```
[ ] #20194902 전예찬
    kernel_type = ["linear", "rbf"]
    svm_train_accuracy = [cifar_svm_lin_train_accuracy, cifar_svm_rbf_train_accuracy]
    svm_test_accuracy = [cifar_svm_lin_test_accuracy, cifar_svm_rbf_test_accuracy]
    cifar_svm_df = pd.DataFrame(data={'kernel_type' : kernel_type, 'train_accuracy' : svm_train_accuracy,
    cifar_svm_df
```

	kernel_type	train_accuracy	test_accuracy
0	linear	0.57484	0.3754
1	rbf	0.70282	0.5436