Machine Learning Assignment 1

20186889 권용한

1. Experimental setup

   윈도우 11, VSCode 파이썬 3.10.10에서 실행하였으며, colab에서 코드가 정상적으로 실행되는 것을 확인하였습니다. 다만 실행시간이 길어 colab 환경에서는 적절히 분리해 실행하는 등 주의할 필요가 있습니다.

   현재는 CPU 사용량에 대해 의식하지 못해 실행 시간이 너무 길었는데, 다음엔 이 점을 유의해 더 좋은 환경에서 실험을 진행할 수 있도록 하겠습니다.

2. 설계 및 소스코드 분석

   DT의 경우 depth가 3, 6, 9, 12인 상황에서 params_grid 중에서 가장 성능이 좋게 측정되는 parameter를 선정해야 하기 때문에, GridSearchCV를 사용하였습니다.

```
params_grid = { 'min_samples_split': [2, 5, 10],
               'min_samples_leaf': [1, 2, 4],
               'max_leaf_nodes': [5, 10, None]}

for depth in range (1,5):
    tree = DecisionTreeClassifier(max_depth=3*depth)
    GR=GridSearchCV(tree,param_grid=params_grid,cv=5,scoring='accuracy',refit=
True)
    GR.fit(MNIST_train_images,MNIST_train_labels)
    print("Best param :",GR.best_params_)
    print('Training Accuracy:',
GR.score(MNIST_train_images,MNIST_train_labels))
    print('Test Accuracy:', GR.score(MNIST_test_images,MNIST_test_labels))
```

   for문을 통해 max_depth가 3*1~4까지 반복하였으며, cv를 5로 설정, refit을 True로 설정하였습니다.

   이후 모델을 학습 시키고 정확도를 출력하였습니다. (CIFAR 또한 같은 방식으로 작성하였습니다.)

   SVM의 경우 kernel에 linear와 rbf를 parameter로 주고 학습을 진행하였습니다. (CIFAR 또한 같은 방식으로 작성하였습니다.)

```
svm_clf=svm.SVC(kernel='linear')
svm_clf.fit(MNIST_train_images,MNIST_train_labels)
print('Linear Training Accuracy:
',svm_clf.score(MNIST_train_images,MNIST_train_labels))
```

```
print('Linear Test Accuracy:
',svm_clf.score(MNIST_test_images,MNIST_test_labels))
svm_clf=svm.SVC(kernel='rbf')
svm_clf.fit(MNIST_train_images,MNIST_train_labels)
print('RBF Training Accuracy:
',svm_clf.score(MNIST_train_images,MNIST_train_labels))
print('RBF Test Accuracy:
',svm_clf.score(MNIST_test_images,MNIST_test_labels))
```

3. Result

Table for DT_MNIST

Depth	Training Accuracy	Test Accuracy	Best param
3	0.49151666666	0.4953	'max_leaf_nodes': 10, 'min_samples_leaf': 1, 'min_samples_split': 2
6	0.73825	0.7415	'max_leaf_nodes': None, 'min_samples_leaf': 1, 'min_samples_split': 2
9	0.86548333333	0.8494	'max_leaf_nodes': None, 'min_samples_leaf': 4, 'min_samples_split': 2
12	0.94915	0.8772	'max_leaf_nodes': None, 'min_samples_leaf': 1, 'min_samples_split': 2

Table for DT_CIFAR

Depth	Training Accuracy	Test Accuracy	Best param
3	0.23762	0.2394	'max_leaf_nodes': 10, 'min_samples_leaf': 1, 'min_samples_split': 2
6	0.29588	0.2812	'max_leaf_nodes': None, 'min_samples_leaf': 1, 'min_samples_split': 2
9	0.38212	0.3042	'max_leaf_nodes': None, 'min_samples_leaf': 4, 'min_samples_split': 10
12	0.521	0.3044	'max_leaf_nodes': None, 'min_samples_leaf': 4, 'min_samples_split': 5

Table for SVM_MNIST

Kernel	Training Accuracy	Test Accuracy
Linear	0.97073333333	0.9403
RBF	0.98991666666	0.9792

Table for SVM_CIFAR

Kernel	Training Accuracy	Test Accuracy
Linear	0.5749	0.3755
RBF	0.70286	0.5436

DT와 SVM을 비교해보자면 MNIST와 CIFAR 데이터에서 모두 SVM을 사용하는 것이 성능이 더 월등히 좋은 것을 확인할 수 있습니다. MNIST 데이터의 경우엔 max_depth가 증가

함에 따라 DT또한 우수한 성적을 보여준다는 것을 볼 수 있습니다.

CIFAR 데이터의 경우엔 MNIST 데이터와 비교해 보았을 때 숫자 이미지를 분류하는 데이터와는 달리 사진에서 object를 구분하는 데이터이기 때문에 데이터가 더 덜 정형화 되었다고 볼 수 있었습니다. 때문에 DT와 SVM에서 좋지 못한 성능을 볼 수 있었습니다.

다만 SVM에서 RBF kernel을 사용하였을 때 이전과 비교했을 때 매우 큰 성능향상을 확인할 수 있었습니다. 즉, 현재 모델 중에서 CIFAR 데이터의 경우 SVM RBF kernel 모델을 사용하는 것이 가장 적절한 것을 알 수 있습니다.

CIFAR 데이터의 DT 모델에서 max_depth가 9, 12인 경우에 test와 training accuracy가 큰 차이가 나오는 것을 볼 수 있는데, 이는 training 데이터에 overfitting 되는 것으로 판단됩니다.

MNIST 데이터는 SVM에서 모두 우수한 성능을 확인 가능하며, DT에서도 SVM에 비해선 부족하지만 훌륭한 성능을 볼 수 있습니다. 다만 max_depth가 12로 증가할 때, training 결과의 정확도 증가량에 비해 test 결과의 정확도 증가량이 다소 떨어지는 것을 볼 수 있으며, max_depth를 더 증가시키게 된다면 overfitting이 발생할 확률이 높을 것이라 판단됩니다.

MNIST 데이터 또한 CIFAR 데이터와 마찬가지로 SVM RBF kernel 모델을 사용하는 것이 가장 성능을 기대하기 좋으며, 성능보다 시간을 더 중시한다면 DT에 max_depth를 적절한 값을 결정해 사용하는 것도 좋은 절충안이 될 수 있을 것입니다.