Machine Learning Lab 8

Simran Kaur

311443

Exercise 1: Optical Character Recognition via Neural Networks

Importing Required libraries

In [7]:

```
from sklearn import datasets
from sklearn.model_selection import train_test_split, cross_val_score
import warnings
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Loading MNIST Dataset

In [8]:

```
mnist = datasets.load_digits()
```

In [9]:

```
X = mnist.data
y = mnist.target
```

Splitting the data into train and test

In [14]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

In [12]:

```
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import GridSearchCV
```

Choosing different hyperparameters for training Neural Network

In [19]:

```
parameter_space = {
    'hidden_layer_sizes': [(20,),(40,), (60,)],
    'activation': ['tanh', 'relu', 'logistic'],
    'solver': ['lbfgs', 'sgd', 'adam']}
nn = MLPClassifier(max_iter = 500, alpha = 0)
clf = GridSearchCV(nn, parameter_space, n_jobs=-1, cv=5)
clf.fit(X_train, y_train)
```

Out[19]:

```
GridSearchCV(cv=5, estimator=MLPClassifier(alpha=0, max_iter=500), n_jobs=-1,
             param_grid={'activation': ['tanh', 'relu', 'logistic'],
                         'hidden_layer_sizes': [(20,), (40,), (60,)],
                         'solver': ['lbfgs', 'sgd', 'adam']})
```

Best Hyperparameters found:

In [20]:

```
print('Best parameters found:\n', clf.best_params_)
print('Best score:\n', clf.best_score_)
```

```
Best parameters found:
 {'activation': 'logistic', 'hidden_layer_sizes': (60,), 'solver': 'adam'}
Best score:
 0.9763404955478127
```

Taking the hyperparameters found above and finding the accuracy on the Test set.

In [21]:

```
nn_test = MLPClassifier(solver = 'adam', alpha = 0, hidden_layer_sizes = (60,), max_
    X_train ,y_train)
print(f'Accuracy on the test set is {nn_test.score(X_test,y_test)}')
```

```
Accuracy on the test set is 0.975
```

Since Regularization is not there so it is possible that model performs better on training than test because of overfitting.