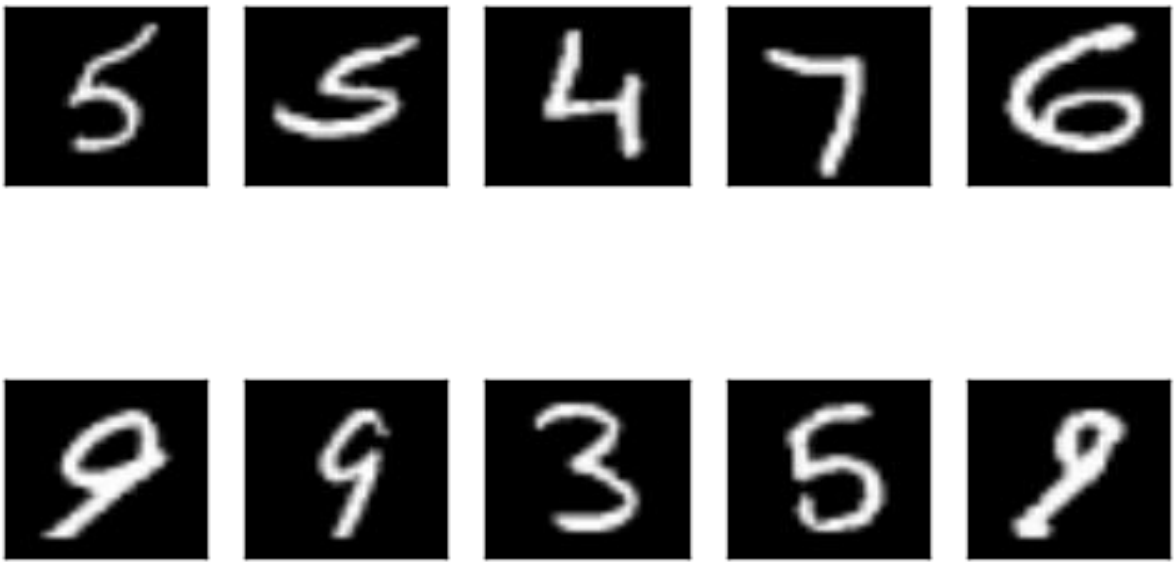
Middlesex University London

Faculty of Science and Technology

Department of Computer Science



Optical Recognition of Handwritten Digits

CST3170 (Artificial Intelligence)

Student Name- Manjot Kaur

Module Leader- Chris Huyck

Campus- Hendon

Table of Contents

1 Introduction-

This project aims at creating a machine learning algorithm that categorizes and predicts handwritten digits. This project employed two techniques-

1. Euclidean Distance
2. Support Vector Machines

2 Problem-

Create a training architecture capable of learning from one of the two datasets provided and then classifying and predicting handwritten digits from the second dataset.

3 Solutions-

To solve this problem, two solutions have been implemented-

3.1 Euclidean Distance-

Based on the Euclidean distance formula for 64 inputs, the test data is classified as the closest training data point. Afterward, the classification is compared with the expected classification of the test data.

3.2 Support Vector Machine-

The most descriptive solution used for this problem is support vector machines.

Each class is first split into several perceptron. The best perceptron is then selected. In the following step, each perceptron is used to classify the test data, and the perceptron that classifies it correctly is selected. Data is classified according to the perceptron that has the correct classification. Next, the predicted classification is compared to the expected classification based on the test data.

3.2.1 Perceptron Learning Algorithm-

To calculate the dot product, each data point is first weighted. Based on the dot product, the (forecast) function determines whether each data point is above or below the hyperplane. Then, the data points are compared against each other and if they belong above or below the hyperplane. Data points misclassified are put into an array, in which a random point is selected. To correctly classify the randomly chosen misclassified data point, the weights are either increased or decreased, and the error rate is also increased. The error rate corresponding to misclassified data points is eventually raised to their highest value, which effectively removes them from the training data points since they will cause an endless loop of adjustment of the weights. In support vector machines, the above process must be repeated until no misclassifications remain.

3.2.1.1 Dot Product-

Dot product is when two vectors are multiplied which results in a scalar quantity.

$$a \cdot b = \sum_{i=1}^{n} a_i b_i$$

Where a and b are the inputs and n is the number of inputs to the perceptron.

3.2.1.2 Equation of the Step Function-

The equation of the step function is calculated by the sum of dot product and bias.

$$\mathbf{w} \cdot \mathbf{x} + b$$

Where w is the randomly assigned weights and x is the inputs array and b is the bias.

3.2.1.3 Hypothesis/Step Function Constraint-

The hypothesis function classifies the value of inputs as either positive or negative.

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b \geq 0, \\ -1 & \text{otherwise} \end{cases}$$

Weights w are represented as vectors of real values, $w \cdot x$ is the dot product and b is the bias. Based on the bias, the decision boundary is shifted away from the origin, regardless of the input value.

4 Results-

```
--------------------------------------------------------------------------------
                                FOLD 1
--------------------------------------------------------------------------------
ALGORITHM NAME:                 SUPPORT VECTOR MACHINES
CORRECT CLASSIFICATIONS:        2368
INCORRECT CLASSIFICATIONS:      442
ACCURACY:                       84.27046263345196%
ERROR RATE:                     15.729537366548044
--------------------------------------------------------------------------------
                                FOLD 2
--------------------------------------------------------------------------------
ALGORITHM NAME:                 SUPPORT VECTOR MACHINES
CORRECT CLASSIFICATIONS:        2461
INCORRECT CLASSIFICATIONS:      349
ACCURACY:                       87.58007117437722%
ERROR RATE:                     12.419928825622776
Average of 2 fold test: 85.92526690391459
```

5 References-

Kowalczyk, A. (2017) Support Vector Machines Succinctly. (SVM).