Theoretical Foundations of
Machine Learning Project Report
MNIST Dataset Handwritten Digit Recognizer
Winter 2021-2022

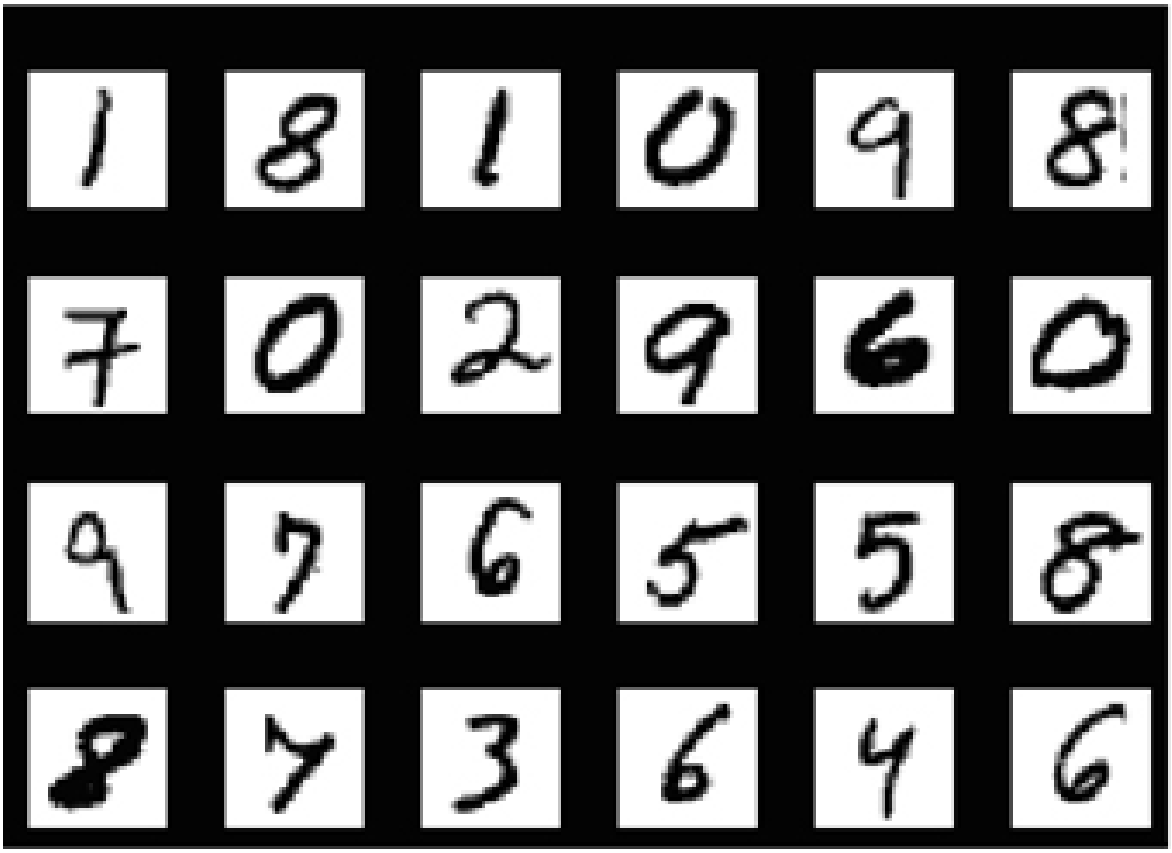
Dataset used:

We used the Modified National Institute of Standards and Technology (MNIST) dataset that was built in the TensorFlow library, which is composed of images of numbers in different shapes written by a large population of volunteers,

It has a training set of 60,000 examples, and a test set of 10,000 examples, it is a subset of the larger set available from National Institute of Standards and Technology of the USA.

The digits have been size-normalized and centered in a fixed-size image.

Each image is a 28x28 pixel image.



Preprocessing:

The dataset was load from tensorflow's keras as

```
from tensorflow import keras
from keras.datasets import mnist
```

The dataset was already divided into X_train, X_test, Y_train, and Y_test, where X_train and X_test are the features matrices and the Y_train and Y_test are the labels.

The preprocessing done was that the RGB values of each pixel of X_train and X_test each was divided by 255 to transform the RGB values of the images to the gray scale, and to limit the range of values to {0,1}.

```
x_train = x_train.astype('float32') / 255
x_test = x_test.astype('float32') / 255
```

such preprocessing allows the classifiers to learn faster.

Libraries used:

Tensorflow: To import the dataset and the keras sequential classifier for the artificial neural network.

Numpy: To manipulate tensors as variables in the form of numpy arrays

Sci-kit learn: To import the K-nearest neighbour classifier, the Support Vector Machine classifier, the classification_report, the confusion_matrix, and GridSearchCV, and we used it extensively to train, test and validate the classifiers.

SkImage: was used to import the HOG feature extraction function.

Matplotlib: was used to plot the data resulting from the heat map data visualization.

Feature extraction:

We used the Histogram of Oriented Gradients (HOG) function to extract the features for the classifiers,

The HOG descriptor is a feature extractor that is used in computer vision and image processing for the purpose of object detection, The HOG descriptor focuses on the structure or the shape of an object and is considered to be the best.

Classifiers:

The used Classifiers were:

- K-Nearest Neighbors
- Support Vector Machine
- Artificial Neural Network

The K-Nearest Neighbor Classifier:

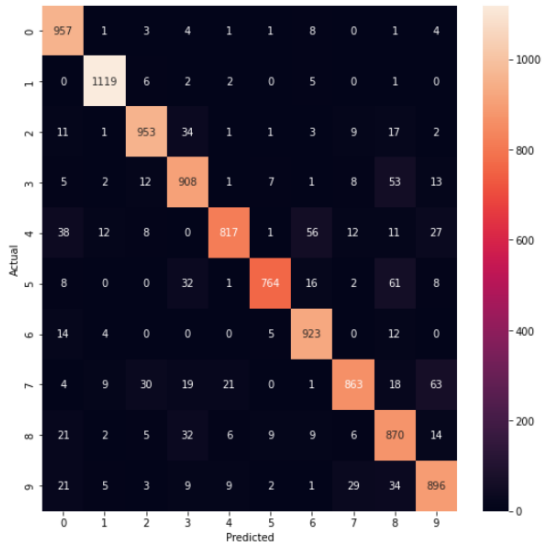
Regarding the KNN classifier, we utilized the GridSearchCV() method

for the parameter hyper tuning and found out that

The best results were at hyper-parameter $K = 8$, which yielded a best score of 90.7%.

Generally speaking, the KNN classifier and its family (Parzen windows classifier) run very poorly on image type data.

The KNN Heat Map:



The KNN Classifier report:

	precision	recall	f1-score	support
0	0.89	0.98	0.93	980
1	0.97	0.99	0.98	1135
2	0.93	0.92	0.93	1032
3	0.87	0.90	0.89	1010
4	0.95	0.83	0.89	982
5	0.97	0.86	0.91	892
6	0.90	0.96	0.93	958
7	0.93	0.84	0.88	1028
8	0.81	0.89	0.85	974
9	0.87	0.89	0.88	1009
accuracy			0.91	10000
macro avg	0.91	0.91	0.91	10000
weighted avg	0.91	0.91	0.91	10000

The SVM Classifier:

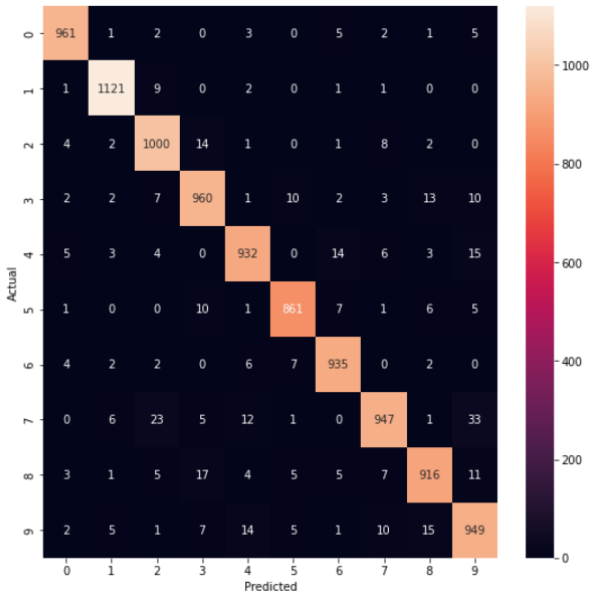
In the case of the SVM classifier, we also utilized the `GridSearchCV()` method

for the parameter hyper tuning and found out that

The best results were hyper-parameters c = 5 and kernel= 'rbf' which yielded a best score of 96%.

the SVM classifier performs better on image type data but runs a lot slower.

The SVM Heat Map:



The SVM report:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	980
1	0.98	0.99	0.98	1135
2	0.95	0.97	0.96	1032
3	0.95	0.95	0.95	1010
4	0.95	0.95	0.95	982
5	0.97	0.97	0.97	892
6	0.96	0.98	0.97	958
7	0.96	0.92	0.94	1028
8	0.96	0.94	0.95	974
9	0.92	0.94	0.93	1009
accuracy			0.96	10000
macro avg	0.96	0.96	0.96	10000
weighted avg	0.96	0.96	0.96	10000

The keras sequential (neural network) Classifier:

Regarding the ANN classifier, we were forced to brute force search for the optimal parameters

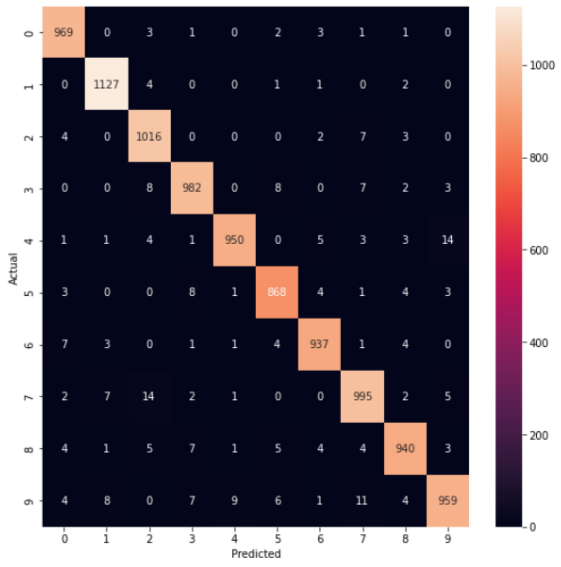
as the `GridSearchCV()` method does not work with keras sequential model when searching for the activation parameter

for the parameter hyper tuning and found out that

The best results was at hyper-parameters optimizer = Nadam, first activation function = tanh and the second activation function = sigmoid which yielded a best score of 97%.

Generally speaking, the ANN classifier and its family run extremely well on image type data.

The ANN Heat Map:



The ANN report:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	980
1	0.98	0.99	0.99	1135
2	0.96	0.98	0.97	1032
3	0.97	0.97	0.97	1010
4	0.99	0.97	0.98	982
5	0.97	0.97	0.97	892
6	0.98	0.98	0.98	958
7	0.97	0.97	0.97	1028
8	0.97	0.97	0.97	974
9	0.97	0.95	0.96	1009
accuracy			0.97	10000
macro avg	0.97	0.97	0.97	10000
weighted avg	0.97	0.97	0.97	10000

Final analysis and conclusion:

After comparing results from the three classifiers (KNN, SVM, ANN) and found out that the artificial neural network classifier
was the best as it achieved the highest score among them across all score measures

However, the ANN classifier is a double-edged sword,
as it allows maximum flexibility with the many types of
hyper-parameters to choose from, consisting of number of layers, number of neurons, optimizers, activation function and initializers, but that comes at the cost of severe computational load to choose the optimal hyper parameters using brute force search.