# Classification of Pictures Using Different Algorithms

Zurui Chen

Shenzhen College of International Education, Shenzhen,China

s20664.chen@stu.scie.com.cn

**Abstract.** This study can be seen as an attempt of handwritten text recognition and analysis, because the dataset can be extended to a dataset including alphabets with or without the colour, and when the model can successfully learn the patterns, it can be applied to real-life applications such as scanning a handwritten document and save its text into a digital document.

**Keywords:** machine learning; algorithms; accurate; MNIST.

## 1. Introduction

Artificial intelligence (AI) and machine learning technology is becoming more and more common in modern industries, including retail, banking, education and more. This is because AI and machine learning are able to predict data using trained models, which can help people make better decisions, such as a trend in interest rate, what university students go to in relation with student's grades, or even in image recognizing, such as showing whether a person is wearing a mask or not, etc. Therefore, a good model to predict data is very important, since a good model will give a more accurate result of prediction, which will reduce possible errors and increase efficiency.

This study aims to design various algorithms for the task of classifying different types of images of the digital numbers from 0 to 9, and to compare the performance of these algorithms. Three algorithms were considered: K-means clustering, logistic regression, and multilayer perceptron (MLP). The hyperparameters of the algorithms were adjusted to optimize their performance.

In this study MNIST dataset [1] is used to test and train the models. MNIST dataset contains 70000 images of handwritten numbers from 0 to 9 in the shape of 28x28 resolution.

## 2. Methods

### 2.1 K-Means Clustering

K-means clustering is a method of classifying data into K groups, such that the mean distance from the data points in the group to the center in each group is at a minimum. The algorithm first chooses K points randomly and then calculates the distance of each data point from the assigned means; then, it groups the data by the minimum distance to each point and calculates the central point in each group. This process is repeated until the points no longer change, which signifies that the grouping task is complete.

Fig. 1 is a representation of K-means clustering for 2D data. The data is clustered into three clusters using the K means clustering method.

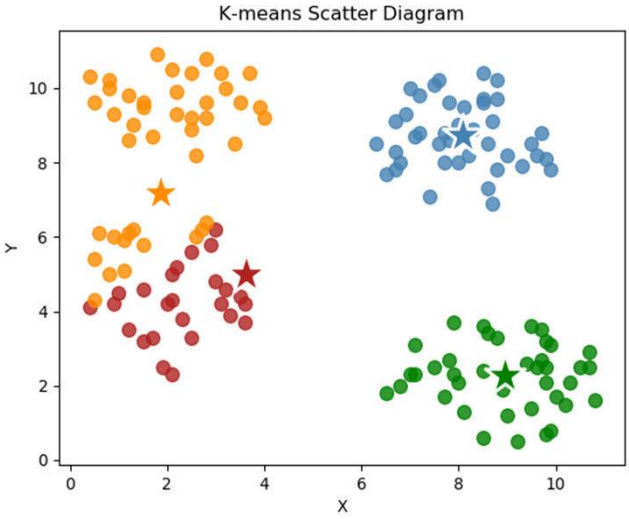


Fig. 1. The data is classified into four groups using K means clustering. The respective mid points are labelled with the star.

Although this is a clustering algorithm, it can also be used to classify data. Data are classified by first assigning 10 points as means (since the pictures contain digits of 0–9 which have a total of 10 different types) randomly on the board of data for the data points and then clustering the data using the normal K-means clustering algorithm along with a special function that can be used to calculate the distances between the pictures of 0–9 to calculate the mean. Subsequently, the algorithm calculates and determines the type (0–9) of picture in each cluster and makes the type of picture the label of the new center.

### 2.2 Logistic Regression

Logistic regression is used to predict data using the training data given. This technique uses a linear or polynomial function to fit a curve to the given test data and to predict future values.

It does so by attempting to fit the data into a polynomial equation,

$$y = \omega_0 + \omega_1 x + \omega_2 x^2 + \cdots + \omega_n x^n \quad . \tag{1}$$

where the value “n” indicates the degree of the model. For example, when n is 0, the model will only attempt to fit the data points to a straight line that is parallel to the x-axis, and when n is 2, the model will attempt to fit the data points to a parabolic shape.

Fig. 2 shows a model with an n value of 1 which is a linear line of best fit of the data set.

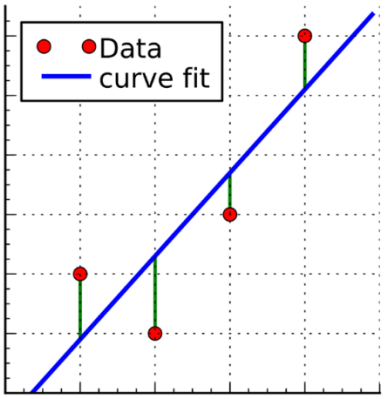


Fig. 2. A representation of a linear logistic regression (line of best fit). The green lines are the losses of the prediction.

The model developed using logistic regression is trained using the training data and tested using the test data. A value (K, the hyperparameter) is assigned to the model to indicate the degree of the polynomial. The higher K is, the more accurately the model fits to the test data. Although a higher K value results in a lower training loss, it may cause a greater testing loss, because at a higher K value, the model pays more attention to the specific identity of the test value set and does not consider general patterns; consequently, the model only fits a curve to the test value set (and fits it almost perfectly), which is called "overfitting."

Overfitting can be reduced by increasing the number of training data points and using regularization.

Increasing the data points means the training data now gives greater detail about the general pattern, which means the model will fit better to the test data, because the test data has the same general pattern as the training data. However, there is still the possibility of overfitting when the degree of the polynomial is quite large.

The regularization algorithm is an algorithm that limits a model's complexity by limiting the weight of the polynomial; it does so by changing the loss function.

The following is the original loss function of the model, which the model attempts to minimize:

$$Loss = \sum_{i=1}^{N} (y_i - \omega_0 - \sum_{j=1}^{n} w_j * x_{i,j})^2 \tag{2}$$

Regularization adds the sum of all the weights squared to the end of the loss function, so that it looks like this:

$$Loss = \sum_{i=1}^{N} (y_i - \omega_0 - \sum_{j=1}^{n} w_j * x_{i,j})^2 + \sum_{j=1}^{n} w_j^2 \tag{3}$$

Therefore, the regularization makes the model decreases the weight of the polynomial, which reduces the complexity of the model and thus reduces overfitting.

The algorithm is implemented in PyTorch; because the result of the algorithm is a probability distribution, it uses a cross-entropy loss function. Cross entropy indicates the difference between two probability distributions. The purpose of cross entropy is to estimate the output probabilities and then determine the difference between the output probabilities and the actual corresponding values.

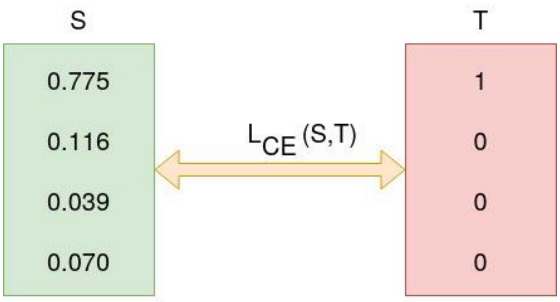


Fig. 3. The relationship between S, the probability, and T, the predicted results, using function L.

The smaller the cross-entropy value is, the better the model performance is.

### 2.3 Multilayer Perceptron

Multilayer perceptron (MLP) is used to achieve a deeper analysis of the trends in the data than logistic regression; for example, it can be used for some trends that cannot be found using logistic regression.

MLP transforms the data so that the features of the data are easier to represent. The transformation is obtained via training.

A single-layer perceptron is equivalent to logistic regression.

Fig. 4 explains the uses of layers in the MLP.

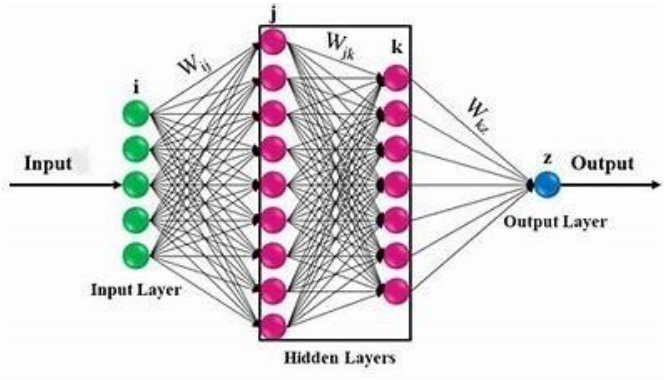


Fig. 4. A representation of the layers in the Multilayer perceptron.

An MLP algorithm receives input signals and assigns weights to them. The inputs and weights are provided to a transformation function (the perceptron function) and the obtained result is provided to an activation function that determines whether the perceptron function produced the correct result. If the result is incorrect, the error result is used as input in the next iteration. This process is repeated until the correct result is obtained from the perceptron function.

## 3. Results

Table I. Results of experiments on divided datasets.

| Models | Data Split Acc (25%) | Data Split Acc (50%) | Data Split Acc (75%) | Data Split Acc (100%) |
|---|---|---|---|---|
| MLP | 99.6 | 99.3 | 98.7 | 99.1 |
| Logistic | 98.1 | 95.9 | 95.7 | 95.3 |
| K-means | 87.9 | 87.3 | 88.9 | 89.5 |

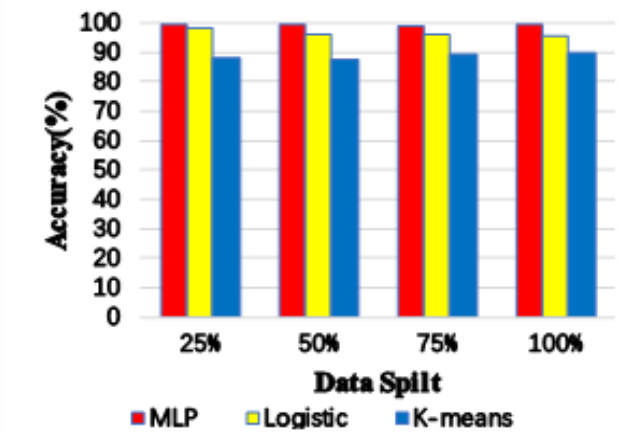


Fig. 5. The results of the four models across all divided datasets.

The dataset is divided randomly into 4 parts, which contain 25% to 100% of the original dataset respectively. The divided datasets are used to train and test the models, where the model computes its parameters by the data and the hyperparameters, which is set manually. The hyperparameters of the models are optimized by testing and comparing results from different parameters and hyperparameters. Fig. 5 shows accuracy of different machine learning models.

According to the figure, it can be clearly seen that MLP generates/obtains the best performance in all sizes of data sets, where the Logistic Regression being the second, and K-means clustering being the most inaccurate, due to the fact that MLP and Logistic Regression uses supervised learning which is suitable for this task. The reason is that MLP, in comparison to the Logistic Regression, has more

parameters, which increase the capacity to capture non-linear relationships between the pictures in the datasets and their respective numbers. Hence, MLP outperform LR.

K-means clustering performs the worst, because K-means clustering is an unsupervised learning algorithm, which means it learns by using un-labelled data, which is unsuitable for label-prepared classification tasks. Although it can work on MNIST task after some adjustments are made to the algorithm, its performance is worse than supervised learning algorithms Logistic Regression or MLP.

In Fig. 6, we show several predicted digital numbers with true label and predicted label using trained MLP model. We can see that all four random examples are predicted correctly, which indicates MLP is able to distinguish MNIST classification task.[2]

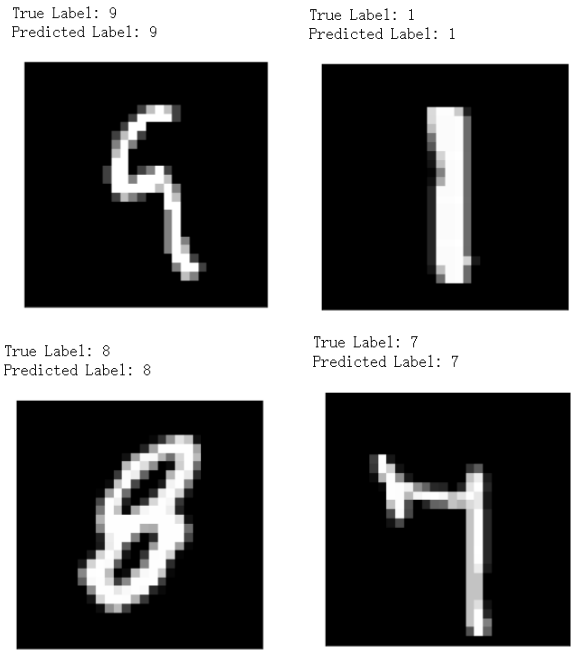


Fig. 6. predicted digital numbers with true label and predicted label

## 4. Conclusion

In conclusion, after I divided the dataset[3] into four different sizes, trained and test the models using those datasets, and finally done the comparisons of the different models in the graph, the MLP is the best model for this task in all size of datasets, because it is able to capture non-linear relationship between the pictures with their corresponding labels.

This comparison can be further extended by comparing the effectiveness of different optimizers for the MLP, and showing which optimizer performs the best in different sizes of datasets, or bringing in different advanced supervised task models in the comparison, and compare their performances in different sizes of datasets.

## References

[1] Birjit Gope, Sagar Pande,Nikhil Karale,et al. Handwritten Digits Identification Using Mnist Database Via Machine Learning Models. IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012108

[2] Majid Vafadar, A Convolutional Neural Network solution for MNIST dataset,February 2018 , Information on: https://www.researchgate.net/publication/339439927

[3] Meshaal mouawad,Pattern Recognition of Handwritten Digits MNIST Dataset,2021,Information on: https://www.academia.edu/48976711