Multiple Methodologies for Enhanced Handwritten Digit Recognition

EES4408 Final Project

U2301048 Shi Jianrui
U2301061 Tang Longbin

March 26, 2024

Abstract

In this study, we explore the efficacy of three distinct machine learning algorithms–Nearest Neighbours (KNN), Logistic Regression, and Support Vector Machine (SVM)–in recognising handwritten digits from the MNIST dataset. The source code can be obtained from https://github.com/TangLongbin/mnist-classification.

1 Introduction

The MNIST dataset is a large collection of handwritten digits commonly used in the field of machine learning and deep learning for training and testing algorithms. It consists of 70,000 images, split into a training set of 60,000 and a test set of 10,000 grayscale images. Each image is 28x28 pixels in size and represents a single digit from 0 to 9. This dataset is widely used as a benchmark for evaluating the performance of various image processing and machine learning models.

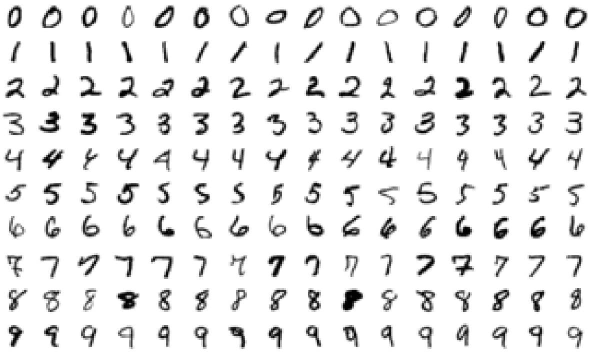

Figure 1: MNIST Dataset

2 Methodology

2.1 Task Definition

Let $X = \{x_1, x_2, ...\}$ be a set of feature vector of training examples and $Y = \{y_1, y_2, ...\}$ be the label of training examples. Each x_i is a vector of D dimension. In MNIST classification task, $D = 28 * 28 = 784$. Our goal is to provide a y_i given a new example of x_i based on its feature.

2.2 Logistic Regression

When deep learning is not popular, logistic regression is the most common method for classification. Logistic regression is a linear method which tries to learn a simple decision boundary for the samples. for every class c, it has a weight vector W_c of which the number of dimensions is the same as the feature vector of a sample. we multiply x with the weight vector W_c to obtain a score for the class c. Then we use softmax to get a normalised score for each class c. Finally we choose the class with the highest score as the predicted label. The equation 2 shows the above process of classification.

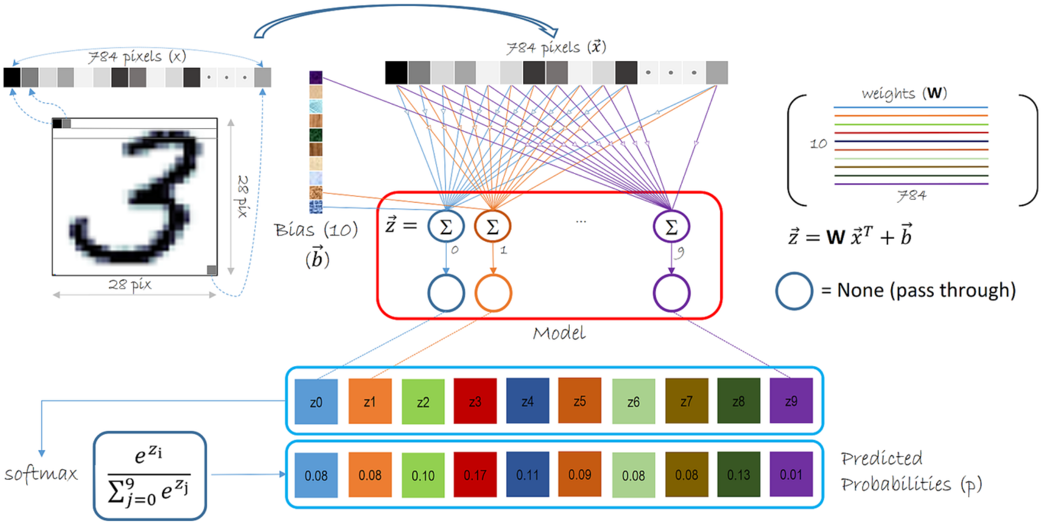

Figure 2: Logistic Regression

$$p(y|x) = \frac{exp(W_y \cdot x)}{\sum_{c=1}^{C} exp(W_c \cdot x)} \tag{1}$$

During training, we minimise the cross entropy loss function over the full dataset, which is the negative log probability of the right class.

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} -log(\frac{e^{f_{y_i}}}{\sum_{c=1}^{C} e^{f_c}}) \tag{2}$$

2.3 Support Vector Machine

Like logistic regression, SVM also tries to find a decision boundary to classify the samples. There are many decision boundaries to split the samples, while SVM tries to find the one that has the maximum margin. The margin means the sum of the distances of two support vectors to the boundaries. The support vector is a sample which is close to the decision boundary.

2.4 k–Nearest Neighbour

The k-nearest neighbours algorithm (kNN) is a non-parametric method used for classification and regression. Unlike the above methods, it does not need to learn a large number of parameters to build a model. It remembers all the training samples into its memory. During prediction, It calculates the distance of the predicted sample and all the training samples. Then the model will choose k-Nearest neighbour of the predicted sample as the candidates. These neighbours will choose a label as the predicted label by voting.

3 Experiments

3.1 Dataset

We use the official MNIST dataset, which has a training set of 60000 images, and a test set of 10000 images. Every image contains a handwritten number ranging from 0 to 9, and every image's size is $28 * 28$, the number locates in the centre of the image.

3.2 Models

We implement 3 models. The details of each model are as follows.

Logistic Regression: We use a linear layer contains a weight matrix W_c and a bias matrix b_c to calculate the confidence of every class, then we use a softmax layer to normalise the score.

Table 1: Logistic Regression Performance

LR	**epoch 3**	**epoch 5**	**epoch 10**	**epoch 20**	**epoch 50**	**epoch 100**
1e-4	0.8938	0.9064	0.9146	0.9217	0.9257	0.9288
1e-3	0.9234	0.9251	0.9261	0.9281	0.9284	0.9279
1e-2	0.9166	0.9110	0.8959	0.9188	0.9143	0.9111

Support Vector Machine: We use the Gaussian radial basis function kernel as the kernel function in our implementation.

Table 2: SVM Performance

epoch 3	**epoch 5**	**epoch 10**	**epoch 15**	**epoch 20**	**epoch 50**
0.4475	0.5778	0.5162	0.7126	0.7525	0.8818
epoch 100	**epoch 200**	**epoch 500**	**epoch 1e3**	**epoch 2e3**	**epoch 5e3**
0.9271	0.9697	0.9797	0.979	0.9795	0.9792

k-Nearest Neighbour: We choose the odd number ranging from 3 to 49 to observe the effect of different k for our model, we pick some typical value for our conclusion.

Table 3: kNN Performance

k	**3**	**5**	**7**	**9**	**15**	**27**	**49**
accuracy	0.9720	0.9720	0.9708	0.9705	0.9664	0.9611	0.9526

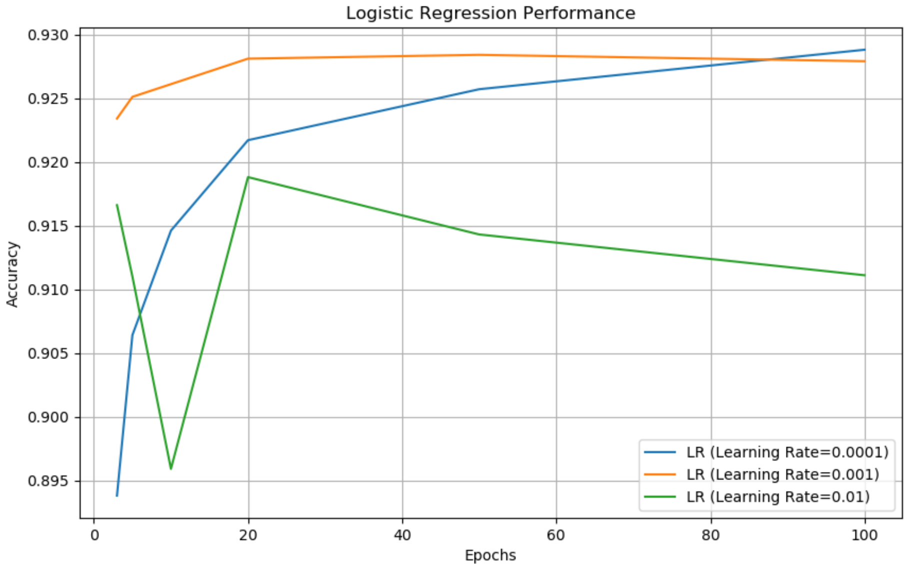

Figure 3: Logistic Regression Performance

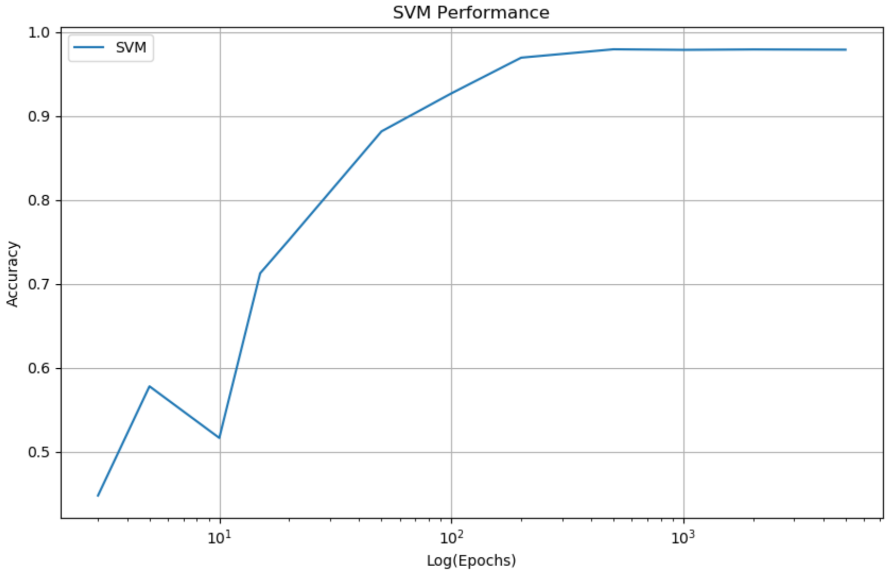

Figure 4: SVM Performance

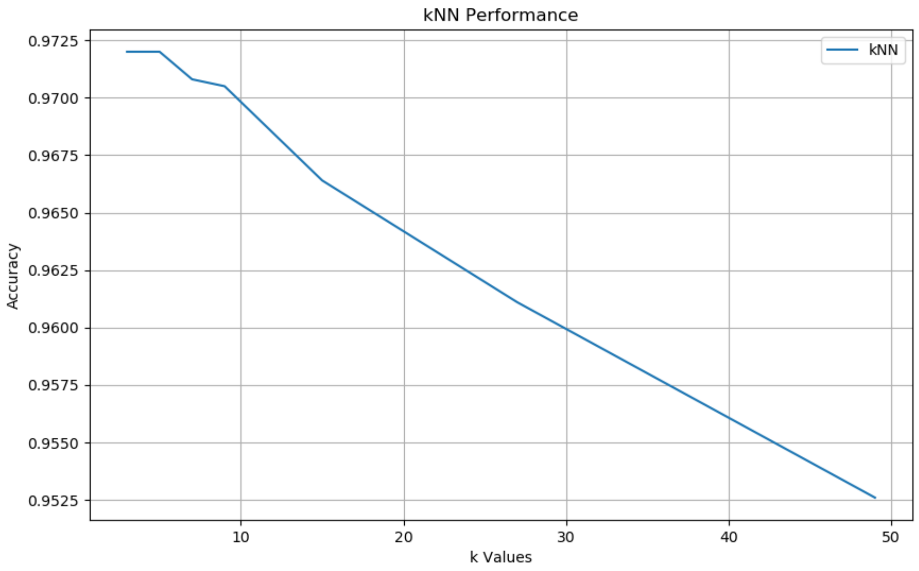

Figure 5: kNN Performance

4 Conclusion

In this study, we explored the efficacy of three distinct machine learning algorithms—Nearest Neighbours (kNN), Logistic Regression, and Support Vector Machine (SVM)—in recognising handwritten digits from the MNIST dataset. Our experiments revealed interesting insights into the performance of these algorithms. Firstly, the k-Nearest Neighbours (kNN) algorithm exhibited the highest accuracy when k=3, achieving an accuracy rate of 0.972. This highlights the effectiveness of the kNN method in this classification task, particularly for smaller values of k. On the other hand, Logistic Regression and Support Vector Machine (SVM) showed varying performance depending on the hyperparameter settings. Further optimisation of these models is necessary to improve their overall performance.