Analysis of MNIST dataset using various models

2022056262 Ju Ri Gwak

ABSTRACT

This report compares various algorithms for classifying the MNIST dataset. Performance is judged through accuracy, prediction time, and learning time. Considering the experimental results, accuracy, and time required, the most suitable model for our purpose is SVM (polynomial deg=3).

1. Problem setting

a. Dataset Introduction

Datasets are 1797 black and white images of handwritten numbers from 0 to 9. Each image is 8x8 pixels in size, and the value of each pixel is between 0 and 16. The pixel value represents contrast.

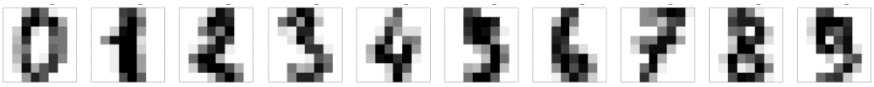

Figure 1: MNIST dataset

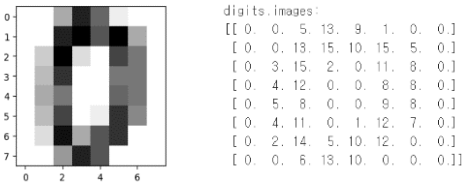

Figure 2: 0 as a matrix

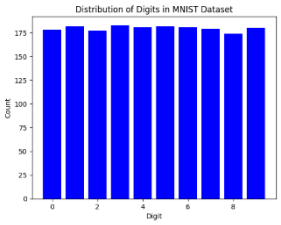

Figure 3: Distribution of Digits

Figure 2 is the result of expressing sample data 0 as a matrix. You can see that each pixel value is between 0 and 16 and is an 8x8 array. Figure 3 is a graph showing how many of each target class there are in the dataset. You can see that it is not biased toward a specific class and is evenly distributed.

b. Purpose

Using the MNIST dataset, we find the algorithm that most accurately classifies handwritten numbers. Predict which model will perform best and confirm through experiment. The performance of the model is judged based on accuracy, and the time spent on prediction and learning are additionally measured to compare in more detail. For accurate comparison, each model is trained under the same conditions and performance is evaluated through a test set.

2. Experiments

a. Preprocessing

Before training, you need to segment the data and resize the images. Errors that may occur when each characteristic of data has a different range can be resolved through data standardization. Since our datasets all have the same range from 0 to 16, there is no need to standardize data. First, because the input data is expressed as a column vector, the dataset was reshaped to create a 1D array. Second, split the data. To train and test a model, you need to divide the data into training data and test data. After shuffling the entire dataset, the training data and test data were divided 8:2. Since CNN is a special form, it was converted into a 3D tensor to make it suitable for CNN.

b. Introduction to Algorithms

i) Logistic Regression

$$L(\mathbf{w}, b) = \sum_{i=1}^{N} \log\left(\exp\left(-y_i\left(\mathbf{x}_i^\top \mathbf{w} + b\right)\right) + 1\right) + \frac{1}{2}\mathbf{w}^\top \mathbf{w}$$

Logistic regression is mainly used when solving binary classification. Although the name is regression, it is actually a model used for classification. Because it is classified as 0 or 1, it is not suitable for our dataset. Therefore, we need to split the multi-class dataset into several binary classification problems.

ii) Linear Discriminant Analysis (LDA, FDA)

LDA is a method of finding linear combinations representing features. FDA calculates the between-class and within-class variance to find the axis that optimizes the variance ratio. Both LDA and FDA are capable of multi-class classification.

iii) Multi-Layer Perceptron

It consists of an input layer, a hidden layer, and an output layer. At this time, each layer consists of several neurons. Each neuron exhibits one characteristic. The hidden layer may have one or more hidden layers. The number of neurons in the output layer varies depending on the problem. Since our dataset is a multi-class classification problem, a total of 10 neurons corresponding to 0 to 9 are used. Backpropagation is used to adjust the weights and find the optimal result.

iv) CNN (ReLU, Softmax)

It is a neural network specialized for images and is efficient because the number of learning parameters is very small. Unlike MLP, it determines the presence or absence of specific information in the image. I extract the features I want from the image through a kernel (a.k.a filter, mask). In the pooling layer, the size is sometimes reduced to reduce the number of network parameters or computation amount.

v) Support Vector Machine (Linear, Non-linear)

First, in the case of linear SVM, a decision boundary that divides the data linearly is found. This crystal boundary is expressed as a hyperplane. Non-linear SVM is used when data is not linearly separated, and in this case, a kernel trick must be used to introduce nonlinearity. Nonlinear kernels usually use polynomial kernels or RBF kernels. In this experiment, we will use a polynomial kernel.

vi) Naïve bayes (Gaussian, Multinomial)

It is learned by using training data to calculate the prior probability and the conditional probability when each feature is a given class. Given new data, calculate the posterior probability of each class and classify the data into the class with the highest probability. These include

Multinomial Naïve Baye, which is mainly used for text classification, Gaussian Naïve Bayes, which is used when dealing with continuous data, and Bernoulli Naïve Bayes, which is used when dealing with binary data. In this experiment, Gaussian and Multinomial will be used.

c. Result Prediction

CNN, which specializes in image processing, is expected to have the highest accuracy and performance because it has the highest accuracy and the fewest number of parameters.

d. Result Output Type

For each class, there is precision, which determines how accurately the model predicted, and recall, which determines how much the model actually detected. These two factors are useful when the class distribution in the dataset is unbalanced. As the dataset we used has an even distribution, as shown in Figure 3, we only examine the accuracy. Additionally, the prediction time and actual learning time are output to compare performance.

3. Experimental results

ALGORITHM	ACCURACY	PREDICTION TIME (SEC)	TRAINING TIME (SEC)
LOGISTIC REGRESSION	**0.9778**	**0.0006**	**4.2336**
LDA	**0.9694**	**0.0005**	**0.0324**
MLP	**0.9853**	**0.0017**	**1.8909**
SVM(LINEAR)	**0.9750**	**0.0070**	**0.0298**
SVM(POLY-2)	**0.9833**	**0.0085**	**0.0299**
SVM(POLY-3)	**0.9889**	**0.0077**	**0.0302**
NAÏVE BAYES (GAUSSIAN)	**0.8861**	**0.0013**	**0.0021**
NAÏVE BAYES (MULTINOMIAL)	**0.9111**	**0.0003**	**0.0028**
CNN	**0.9869**	**0.9427**	**2.0220**

Table 1: Experimental results

a. Logistic Regression

Because the calculation is repeated until the function converges, it takes a long time to learn, but the accuracy is above average and the prediction time is fast. Since it is a simple algorithm, it is easy to implement, but it takes a long time if it does not converge properly.

b. LDA

It is more suitable for general cases because it compares the covariance of the data, but in this case, since the data is assumed to follow a normal distribution, accuracy may decrease in some cases.

c. MLP

Because it has a multi-layer structure, complex models can be created and accuracy increases accordingly, but the learning time is long because there are many parameters.

d. SVM

Both linear and nonlinear methods have high accuracy, showing that they perform well even on small datasets. In the case of nonlinear SVM, linear separation is converted into a feasible problem by mapping the data into a high-dimensional space. Therefore, nonlinear shows higher accuracy than linear. Performance is good even with small datasets, but the higher the dimensionality of the dataset, the longer the mapping time becomes.

e. Naïve Bayes

Both linear Bayes and nonlinear Bayes have low accuracy because they are learned under the assumption that they are probabilistically independent. Input data can be classified quickly, but if the data is not independent, accuracy is very low.

f. CNN

For similar reasons to MLP, it takes a long time to learn. In particular, it takes more time because it goes through pooling and convolution processes. However, since it is a model specialized for image classification, the accuracy was very high. If the size of the data is large, it may take too long to use.

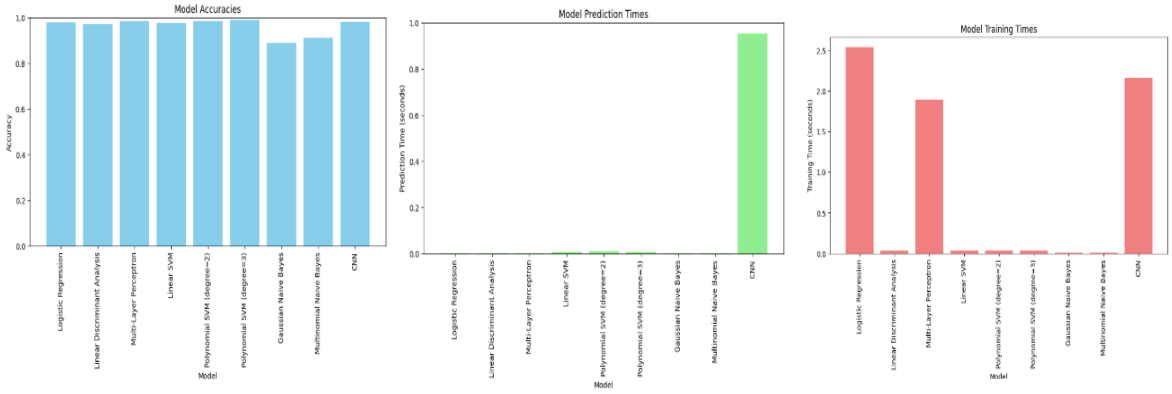

Figure 4: Accuracy, Training time, Prediction time per model

4. Discussion

Naïve Bayes models performed best in terms of prediction time and learning time. However, in the case of the Naïve Bayes model, as can be seen from the Figure 5, the accuracy is significantly lower than other models, so it is not the most suitable algorithm. For reference, the element value of the confusion matrix refers to the rate of predicting the corresponding class when learning was performed 10 times. The accuracy seems to be relatively low because it is predicted on the assumption that all data will be independent. In the case of CNN, the accuracy is high, but the prediction time and learning time take too long, so this model is also not the most suitable. Although Cnn has fewer parameters than MLP, it still has more parameters than other models and includes pooling layers and convolution layers, so the learning time seems to increase. CNN predicts classes by hierarchically extracting features, but in the case of the MNIST dataset, because the numerical positions in each image are similar, it takes a long time to extract features, which seems to increase the prediction time. Unusually, it took a long time to learn in Logistic Regression and MLP, which is due to repeated calculations until the model converges and parameter adjustment using backpropagation due to the multi-layer structure.

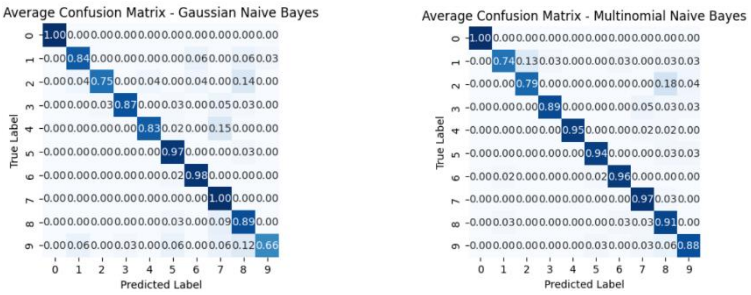

Figure 5: Confusion Matrix for Naive Bayes

5. Conclusion

In terms of accuracy, the two models classified most accurately: CNN (98.6%) and SVM (polynomial deg=3) (98.8%). However, contrary to expectations, in the case of CNN, the prediction time (0.9427) and learning time (2.0220) take a very long time due to the multi-layer structure and the characteristics of the data are not very different, making it difficult to use when the data is large. If the characteristics of each data are significantly different and the amount of data is not too large, it would be better to use CNN when classifying images. In the case of Naïve Bayes, which showed good performance in terms of execution time, the loss is likely to increase if the data are correlated. Therefore, the most suitable model to classify the MNIST dataset is SVM (polynomial deg=3).