Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/18/20230956

Applying multi-layer perceptron and ResNet for handwritten digits recognition

Zehui He

Department of software engineering, Zhongyuan University of Technology, Zhengzhou, 450007, China

202028034232@zut.edu.cn

Abstract. The task of handwritten digit recognition is to recognizing the handwritten digits from pictures. Applying machine learning based models to automatically perform handwritten digit recognition task can significantly improve efficiency. This paper applies two machine learning based models, including multi-layer perceptron and residual neural network, for such a task. Firstly, this paper introduces the basic concept of the simple multi-layer perceptron model and then presents the structure of the residual neural network model. Subsequently, such two models are trained on the MNIST corpus, one of the classical dataset for the handwritten digit recognition task. The data pre-processing, like the splitting of training and test set, is described. Also, the processes of testing and training of the two models are presented. According to the experiments on the test set of MNIST, it is observed that the residual neural network can achieve better performance where the accuracy score is 99.240%, while the accuracy score of the multi-layer perceptron model is 97.260%.

Keywords: multi-layer perceptron, ResNet, handwritten digits recognition.

1. Introduction

Deep learning, one branch of the machine learning, is one of the most advanced and cutting-edge disciplines in the realm of artificial intelligence, and continually changes many aspects of human life, such as housework, study mode, work style, healthcare [1] and weather forecasting [2]. Furthermore, machine learning can integrate knowledge and methods from many different fields and areas, such as probability theory, mathematical statistics, linear algebra and data analysis, to address various practical problems, like recognizing handwritten digits. Machine learning methods in a supervised manner are based on annotated training data and is able to learn underlying patterns from training samples. This work applies supervised machine learning methods, including multi-layer perceptron (i.e., MLP) and residual neural network (i.e., ResNet) models, to address the task of handwritten digits recognition.

To be specific, machine learning based models can be trained on the training data with manually annotated labels. For example, the MNIST corpus, one of the typical datasets for handwritten digits recognition task, is composed of pictures of 10 different types of handwritten digits and corresponding labels, including the number of 0~9, respectively. There are 60,000 pictures for training, and the remaining 10,0000 pictures are used for testing. The above two models can be trained over training data and evaluated on the test set. According to experiments, the performance of ResNet is better than the performance of MLP, which are 99.240% and 97.260% individually.

DOI: 10.54254/2755-2721/18/20230956

2. Multi-layer perceptron

A multi-layer perceptron model usually consists of an input layer, a hidden layer and an output layer [11]. The MLP can learn and store a huge number of input-output mappings. Moreover, such a model does not require the mathematical equations that describe the relationships between the mappings to be revealed in advance, which allows the model to achieve better generalization ability. The weights and thresholds of the model are continuously adjusted by using the fastest descent approach through back propagation, which can reduce the summation of total squared errors to the minimal extend. The input signal is propagated in the forward direction. The error spreads in the opposite direction from the way that the signal spreads. This is the main feature of the MLP model.

The neuron is the most basic component of the MLP model. Each neuron accepts the input signals that are from the input source or other neurons. Such input signals are transmitted through the connection with the weights, so that the neurons can accumulate all these signals and determine the input value overall. Additionally, the aggregated input values are evaluated against the neuron's threshold value, then the output values of the neuron are produced by an activation function. The results of the activation function are then sequentially passed to the subsequent neurons. In this way, the output are considered as the input signals of the next neurons. In this paper, the Sigmoid function is adopted as the activation function. The following equation (1) is the calculation of the Sigmoid function:

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{1}$$

3. ResNet

CNN-base models usually consist of a input layer, convolutional layers, activation functions, pooling layers, fully-connected layers and the output layer [3,4]. Before the advent of the residual neural network, the depth of a convolutional neural network is not very large due to poor performance. One of the main reasons is that the training error could gradually increase when the number of network layers is increased [2,5]. Until 2015, the residual neural network was designed, which has a great significance in the development of deep convolutional neural networks [2,5,6]. A residual network is made up of a series of residual blocks. Figure 1 demonstrated the architecture of a residual block. The emergence of ResNet effectively alleviates the problems about gradient vanishing and network degradation caused by the increased depth of the CNNs. Moreover, ResNet can also increases the training speed of the neural network as well as dramatically improve the generalization ability.

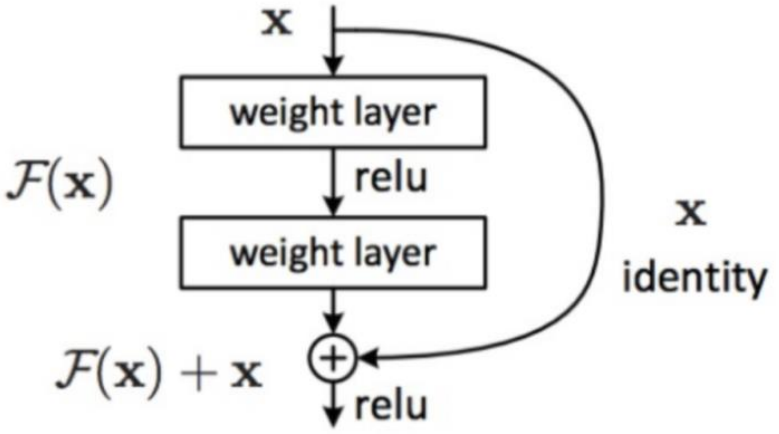

Figure 1. The architecture of a residual block.

3.1. Convolutional layer

For an input image, it is first converted into a matrix whose elements are the corresponding pixel values of the input image [7]. The convolution layer consists of a set of filters, which are used to extract features from the input matrix. A convolution layer can be calculated as:

DOI: 10.54254/2755-2721/18/20230956

$$u_k = \frac{1}{n \times n} \sum_{j \in N_k} x_j^l \tag{2}$$

In the equation (2), N_k represents the k-th n × n image block, and x_j^l refers to the j element contained in the N_k image block. The batch normalization is then adopted after each convolution layer to alleviate the covariate shift issues during training [8].

3.2. Activation function

In the convolution neural networks, the Rectified Linear Unit (in short, ReLU) function is utilized to alleviate the gradient vanishing problem. The ReLU function is defined as:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{3}$$

The curve of ReLU function is illustrated in figure 2.

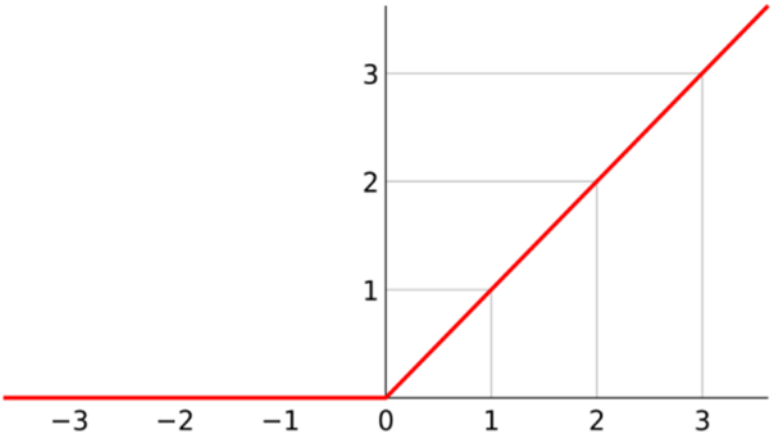

Figure 2. The curve of ReLU function.

Additionally, utilising the ReLU function performs better in gradient descent and back propagation than other activation functions in neural networks, such as the Sigmoid and the Tanh functions. Therefore, gradient explosion and gradient disappearance can both be avoided by utilising the ReLU function. Moreover, compared to the Sigmoid and Tanh, the calculation of ReLU function is much simpler, which could decrease the overall computing cost of neural network models [9].

3.3. Fully-connected layer

A fully-connected layer can perform matrix multiplication and bias addition operations on the input features and the connection weights between each neuron to obtain the output result. Each input feature has a specific connection weight with each neuron in the fully-connected layers, in which every neuron is connected to the other neurons in the next layer. During training, the neural network utilises a back-propagation algorithm to modify both weights and bias of each neuron in order to achieve an optimal model performance [10]. Since the fully-connected layer can be converted to the convolutional layer with a convolutional kernel of 1x1, which is entirely connected to the front layer, the fully connected layer can be obtained in the practical application by the convolution operation. The output results of a fully connected layer can be considered as a nonlinear transformation of the input features. Such a layer can transfer the input feature space into the output result space, which also allows models to achieve more complexity and nonlinear fitting capability. A fully-connected layer with activation function f can be calculated as:

DOI: 10.54254/2755-2721/18/20230956

$$y^k = f(w^k x^{k-1} + b^k) \tag{4}$$

where w^k represents the weight coefficient in the fully-connected layer, and x^{k-1} represents the features extracted by convolution layer and pooling layer, and b^k represents the bias of the fully-connected layer.

4. Results

The pixels of each handwritten digital picture are 28×28. During model training, the pixels in the each image are converted into vectors, which have the length of 784. Therefore, the training set for MNIST corpus can be regarded as a tensor of [60000, 784], where the first dimension corresponds to the picture index and the second dimension corresponds to each image's pixel.

For MLP model, the dimension of the input layer is [batch size, 784]. The hidden layer consists of two layers, which have the dimensions of [batch size, 400] and [batch size, 200], respectively. The dimension of the output layer is [batch size, 10]. The optimizer is Adam with learning rate of 0.01. The loss function is cross entropy loss function.

The convolutional layer of ResNet includes 3×3 convolution kernels. A batch normalization layer and a ReLU activation function are placed after each convolutional layer. The optimizer is Adam with learning rate of 0.01. The cross entropy loss function is adopted as the loss function.

4.1. Results for MLP

Figure 3 demonstrates the training and test loss curves for the MLP model. Figure 4 illustrates the change of the accuracy scores of the MLP model during training and test processing, respectively. It can be clearly seen in the following two graphs that the loss decreases sharply with the increasing of the number of iterations, where the final training loss value reaches 0.109004. The accuracy score of the handwritten digit recognition first increases dramatically and then gradually stabilizes. The final accuracy score of MLP on test set is 97.260%.

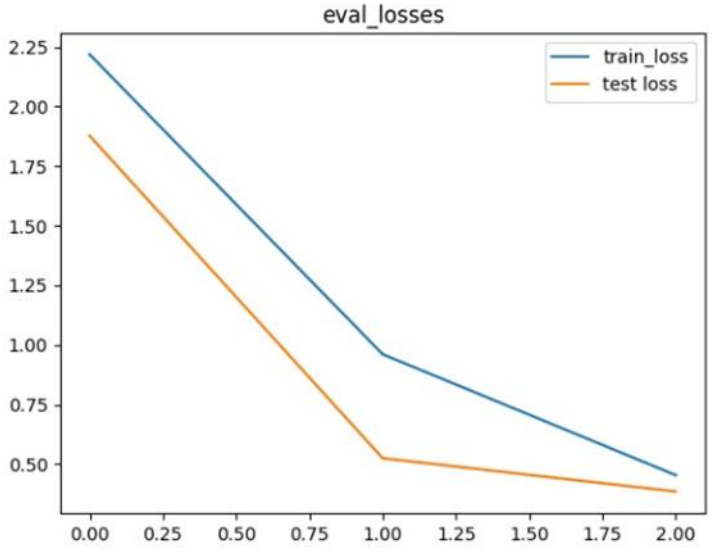

Figure 3. The loss curve of MLP.

DOI: 10.54254/2755-2721/18/20230956

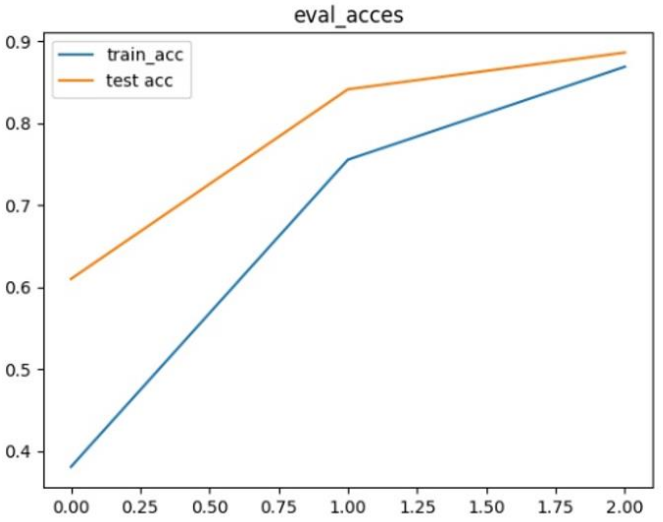

Figure 4. The accuracy score of MLP.

4.2. Results for ResNet

Figure 5 demonstrates the training and test loss curves for the ResNet model. Figure 6 illustrates the change of the accuracy scores of the ResNet model during training and test processing, respectively. It is clearly illustrated that the loss converges rapidly with the increasing of the number of iterations. The lowest loss value during training is 0.000796. At the same time, the accuracy score of the model also increases and then tends to stabilize. The accuracy score of test set is 99.240%.

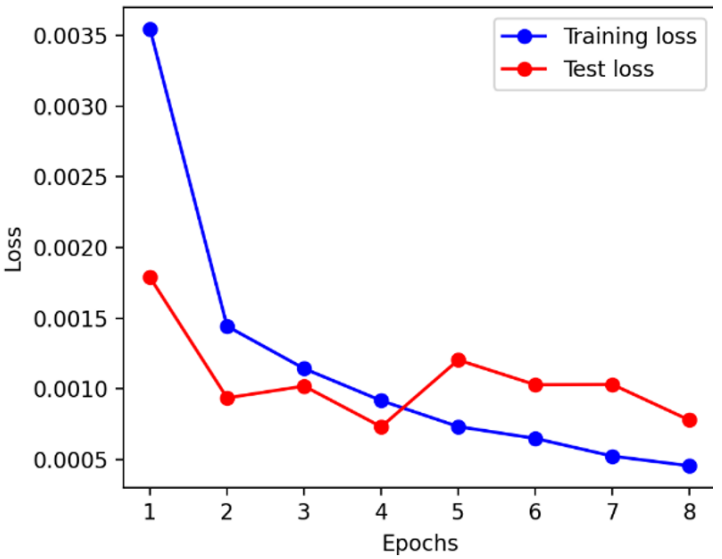

Figure 5. The loss curve of ResNet.

DOI: 10.54254/2755-2721/18/20230956

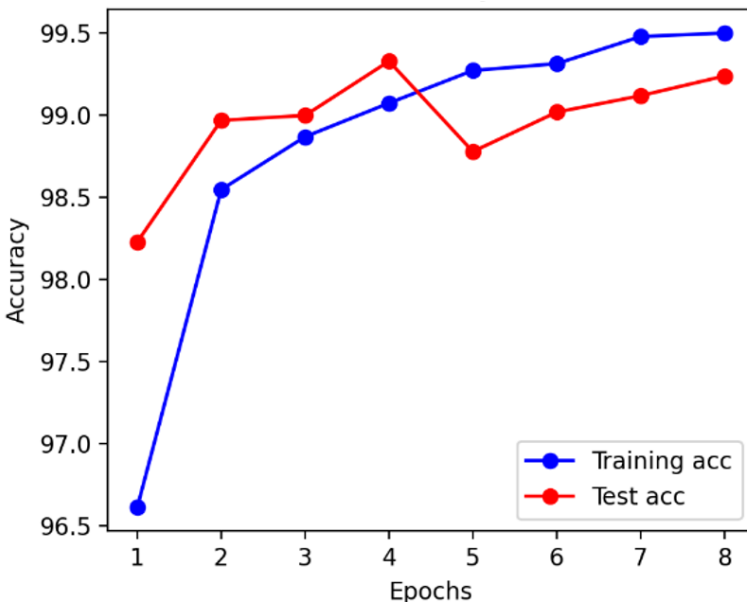

Figure 6. The accuracy score of ResNet.

4.3. Discussion

Compared such two models, ResNet model can achieve lower loss value for both training and testing set. Also, the performance of ResNet is better than the MLP model. However, the running speed of MLP is much faster than ResNet due to the simple structure of MLP.

5. Conclusion

This paper applies MLP and ResNet models to address the handwritten digits recognition task. According to experiments, the accuracy scores of MLP and ResNet models on the testing set are 97.260% and 99.240%, respectively. Therefore, it is concluded that, for the task of handwritten digit recognition, the performance of ResNet is generally better than MLP. In the future, hyper-parameter tuning and designing better architectures of CNN can further improve the performance on handwritten digit recognition task.

References

[1] Rahman M S, Khomh F, Hamidi A, Cheng J, Antoniol G, and Washizaki H 2023 Machine learning application development: practitioners' insights Software Quality Journal p 1-55

[2] Huang J C, Ko K M, Shu M H and Hsu B M 2020 Application and comparison of several machine learning algorithms and their integration models in regression problems Neural Computing and Applications vol 32 p 5461-69

[3] Ali S, Shaukat Z, Azeem M, Sakhawat Z, Mahmood T and ur Rehman K 2019 An efficient and improved scheme for handwritten digit recognition based on convolutional neural network SN Applied Sciences vol 1 p 1-9

[4] Lakshmi K L, Muthulakshmi P, Nithya A A, Jeyavathana R B, Usharani R, Das N S and Devi G N 2023 Recognition of emotions in speech using deep CNN and ResNet Soft Computing p 1-7

[5] Jogin M, Madhulika M S, Divya G D, Meghana R K and Apoorva S 2018 Feature extraction using convolution neural networks (CNN) and deep learning In IEEE Int. Conf. on RTEICT p 2319-23

[6] Ibrahim A O, Shamsuddin S M, Abraham A and Qasem S N 2019 Adaptive memetic method of multi-objective genetic evolutionary algorithm for backpropagation neural network Neural

DOI: 10.54254/2755-2721/18/20230956

Computing and Applications vol 31 p 4945-62
[7] Ghosh M M and Maghari A Y 2017 A comparative study on handwriting digit recognition using neural networks In Int. Conf. on Promising Electronic Technologies p 77-81
[8] Jafar A and Lee M 2021 High-speed hyperparameter optimization for deep ResNet models in image recognition Cluster Computing p 1-9
[9] Rössig A and Petkovic M 2021 Advances in verification of ReLU neural networks Journal of Global Optimization vol 81 p 109-52
[10] Nayef B H, Abdullah S N H S, Sulaiman R and Alyasseri Z A A 2022 Optimized leaky ReLU for handwritten Arabic character recognition using convolution neural networks Multimedia Tools and Applications p 1-30
[11] Sarkar S, Agrawal S, Baker T, Maddikunta P K and Gadekallu T R 2022 Catalysis of neural activation functions: adaptive feed-forward training for big data applications Applied Intelligence vol 52.12 p 13364-83