

Handwritten digit recognition based on the MNIST dataset under PyTorch

Ruwei Wang

School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, Shandong Province, 250014, China

rrw1227805945@163.com

Abstract. Thanks to advancements in machine learning and artificial intelligence techniques, computers can now practice on data and learn from it in a manner that is similar to how the human brain works. Handwritten character and number identification has been one of the most pressing and fascinating subjects in pattern recognition and image processing. One of the most urgent and intriguing topics in pattern recognition and picture processing has been the identification of handwritten characters and numbers. As a crucial part of artificial intelligence, handwritten digit identification technology provides a vast array of application possibilities. The data demonstrates that, even though handwritten numbers are simply created with a few straightforward strokes, the appearance of numbers is more variable due to the various writing styles of each person. In this study, a deep learning framework-based upgraded LeNet-5 convolutional neural network model is used to build a handwritten number recognition model in Python. Automatic recognition of handwritten numbers will become the standard recognition technique if it can be applied to a wide range of industries, including banking and accounting, and hence save human costs.

Keywords: MNIST, convolutional neural network, PyTorch, handwritten digit recognition, offline recognition, artificial intelligence.

1. Introduction

Recent years have seen a lot of research and advancement in deep learning. Handwritten digit recognition is one of these branches that is currently widely used for AI recognition and has become a big challenge to focus on and overcome. Prior attempts in offline handwritten digit recognition using conventional classifiers based on SVM [1] failed to provide sufficient results. In this study, a convolutional neural network in Python and the MNIST classical dataset are used to train a handwritten digit identification model. A modified LeNet-5 model served as the model's foundation. The studies produced positive results, with the model's accuracy reaching 97% after only three training phases. The accuracy of handwritten digit identification in the final model may potentially reach 98%. Homemade handwritten digit examples were examined in addition to the test set of the MNIST dataset, and these samples may serve as a useful benchmark for developing handwritten digit recognition algorithms.

2. Materials and methods

Please follow these instructions as carefully as possible so all articles within a conference have the same style to the title page. This paragraph follows a section title so it should not be indented.

2.1. MNIST dataset

One of the most essential models in the study of computer vision is called the MNIST dataset (Mixed National Institute of Standards and Technology database). Both a training set and a test set of the MNIST dataset are imported and supplied as input to the training model. The feature extraction process is one of the most critical steps in effective picture categorization. The variety of figures offered has led to a wide range of ways being thought up for individuals who have previously come up with them because of their distinct writing styles. The ability to detect handwritten numerals is being developed over time using deep learning methods.

2.2. The PyTorch deep learning framework

With the help of this article, the PyTorch framework can take use of significant GPU acceleration and can support dynamic neural networks, which is something that many popular frameworks, like TensorFlow, do not. In order to accomplish processing that may speed up the model runs and give a solid foundation for further research, the grayscale pictures from the MNIST dataset are used in this study instead of color images from other datasets.

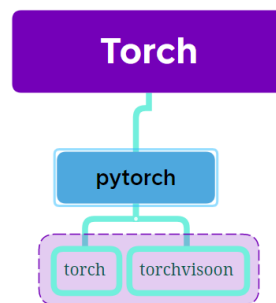


Figure 1. Architecture diagram of PyTorch.

2.3. Convolutional Neural Network (CNN)

Convolutional Neural Networks have reached a certain level of accuracy in data research with the development of the times, but there are still some shortcomings for some algorithms: The training set of data has a limited variety or fewer samples; the problem of overfitting occurs during the training process; a supervisory link is missing in the training process of the data [2]. For the purpose of resolving any potential issues, this article is built using the PyTorch deep learning platform. The two are merged to apply the model to their handwritten figures in addition to training and testing the model on the MNIST dataset.

3. Results and discussions

3.1. Experimental background

Train a convolutional neural network to recognize a handwritten test set in the MNIST dataset using the training set from the MNIST dataset.

A convolutional neural network is a feed-forward neural network, typically a deep neural network. The reason why [3] was first conceptualized was that often for networks of high complexity, the generalization of samples is limited, so the concept of a convolutional neural network (CNN) was proposed to solve the problem of high recognition and accuracy models. In this paper, a CNN neural

network is built using two 2D convolutional layers and two fully connected (or linear) layers for handwritten digit recognition.

3.2. Prevention of problems that may arise

To prevent the problem of overfitting, we usually address it in terms of insufficient training data, too many iterations, dropout regularization, etc. In this paper, we use reducing the number of unnecessary iterations and dropouts to solve the overfitting problem. Dropout can deactivate any node in the convolutional neural network at random to achieve a solution to the situation where the convolutional neural network will place too much weight on a certain node during the training process. As a result, the convolutional neural network does not rely on the weights and features of any node and finally compresses the weights to achieve regularization.

When backpropagation is used, the ReLU function has many of the desirable properties of a linear activation function when compared to a traditional sigmoid function, and the model's recognition rate is greatly improved, as well as the gradient disappearance problem being solved.

m , the z component of the nuclear spin, is italic because it can have different values whereas n is Roman because it is a label meaning nuclear.

$$f(x) = \text{relu}(x) = \max(x, 0) \quad (1)$$

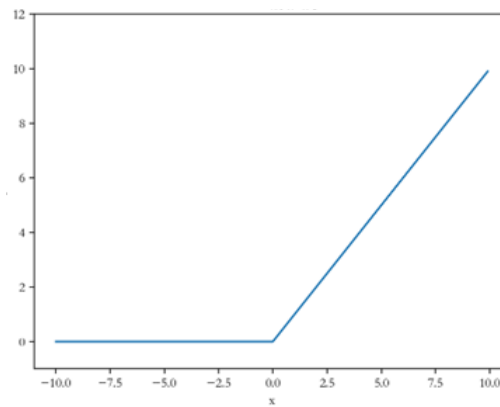


Figure 2. ReLU function.

3.3. Experimental implementation

3.3.1. Test dataset. This article uses the DataLoader under the PyTorch dataset, which is where TorchVision comes into play. TorchVision is a graphics library for PyTorch that is primarily used to build computer vision, a model that serves but is independent of the PyTorch deep learning framework. Multiple threads may be provided by DataLoader, which combines a dataset with a sampler, to process the dataset, beginning with initialization of the MNIST data. using Matplotlib to plot the sample numbers from the MNIST dataset.

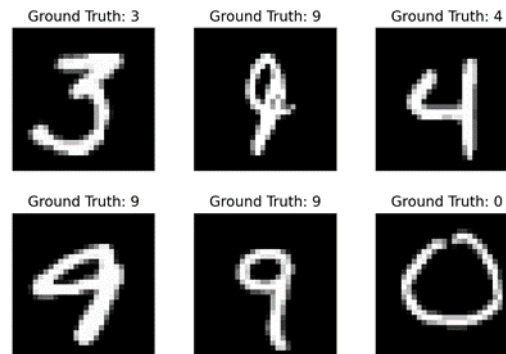


Figure 3. Sample MNIST training set.

3.3.2. Building the network. Building a convolutional neural network (CNN) based on an improved LeNet-5 model, using a ReLU function to solve the gradient vanishing problem, and making the deep network trainable. The addition of the dropout layer can improve the generalization ability of the neural network and achieve better recognition results. It is evident from experimental comparison that, after the dropout layer was added, the recognition rate of the model in [4] is 3.4% more accurate than the prior standard neural network model.

3.3.3. Model training. A training loop is created to iteratively train the 60,000 training sets in the MNIST dataset, and the output is printed to track progress. And use the `optimizer.state_dict()` to save and load the internal state of the neural network.

Table 1. Iterations and Accuracy.

| Iterations | Accuracy(%) |
|------------|-------------|
| 1 | 96 |
| 2 | 97 |
| 3 | 97 |
| 4 | 98 |
| 5 | 98 |
| 6 | 98 |
| 7 | 98 |
| 8 | 98 |

The training results demonstrate that the constructed convolutional neural network has good feasibility for handwritten digit recognition with high accuracy and good processing power for data sets with a large number of digits

3.3.4. Test set testing. A call to `load_state_dict(state_dict)` tests Test loss and accuracy the test set with the newly trained model against the MNIST dataset and prints out the numbers and predictions for a portion of the test set for viewing and validation. It can be found that the model has a test loss of 0.0527 for the test set and an accuracy of 98% for the test set.

```
test.py::test PASSED
Test set: Avg. loss: 0.0527, Accuracy: 9821/10000 (98%)
```

Figure 4. Test loss and accuracy.

3.4. Model application

The newly trained convolutional neural network model was used to complete the task of recognizing handwritten digits in addition to those in the MNIST dataset. To compare the recognition accuracy, the model was verified to be more accurate not only for the given dataset but also on homemade handwritten

samples based on the previous ones. The samples were also found to have good recognition accuracy. The experiments ultimately improve the reliability of the model and can be a better pointer to digital recognition in areas such as finance and auditing.

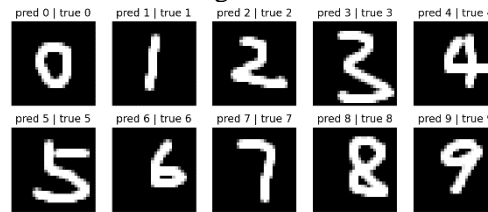


Figure 5. Home-made handwritten samples and predicted results.

3.5. Assessment models

To see the changes in the test set more clearly, a training curve was set up, and it can be seen that after just 3 stages of training, it was able to achieve an accuracy of 97% in the test set and eventually up to 98% with no overfitting problems.

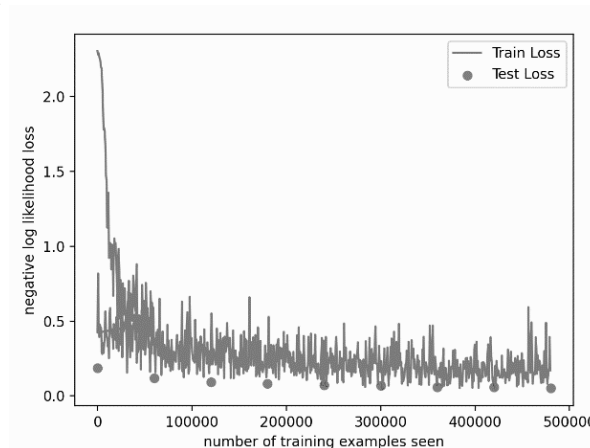


Figure 6. Test curve.

3.6. Experimental results

With the help of the PyTorch framework and Torchvision, the convolutional neural network CNN model built can accurately identify the test set of the MNSIT dataset and improve on the LeNet-5 model, which in turn has no overfitting problems. The accuracy of the model is as high as 98%, and the generalization ability for new samples is also strong, which can achieve more accurate identification.

4. Conclusion

This article proposes an improvement based on the LeNet-5 model. The improved convolutional neural network model has better applicability in practical areas, and the application area of handwritten numeral recognition using computers is gradually becoming more widespread. With the development of information technology, electronic computers can automatically recognize handwritten Arabic numerals and give high accuracy, or even a zero-error rate. These tasks previously required a lot of manual entry and solved the problem of requiring a lot of manual processing of complicated data. The input of human and material resources is greater, and the intensity of labor is a larger problem. In addition, the study of handwritten digital recognition adapts to the needs of a paperless office, reduces the cost of input, and can greatly improve work efficiency. But there are still two major problems. Firstly, the current accuracy of handwritten digit recognition needs to reach a higher level, and there is still room to improve the accuracy of the model when this experiment is conducted for model testing. In the field of digital finance, where every digit is extraordinarily important, further research is needed to achieve an error rate of zero.

Secondly, there is still much room for improvement in the speed of the model for handwritten digit recognition. Often, areas requiring digital recognition where the input of numbers is large, even in succession, place higher demands on the algorithm, and precision and speed are at odds with each other, requiring a deeper study of the algorithm, for example.

References

- [1] Zhang, H. M. & Zhang, Kyauk J.. (2021). A review of research on offline handwritten digit recognition based on artificial intelligence. *Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition)* (05), 83-91. doi:10.14132/j.cnki.1673-5439.2021.05.012.
- [2] Zong, Chunmei, Zhang, Yueqin & Shi, Ding. (2021). CNN-based handwritten digit recognition under PyTorch and application research. *Computer and Digital Engineering* (06), 1107-1112.
- [3] Tang, J.B., Li, W.J., Zhao, B. & Xi, L.P.. (2022). Research on handwritten digit recognition method based on convolutional neural network. *Electronic Design Engineering* (21), 189-193. doi:10.14022/j.issn1674-6236.2022.21.040.
- [4] Song Xiaoru, Wu Xue, Gao Song & Chen Chaobo. (2019). A study of handwritten digit recognition simulation based on deep neural networks. *Science, Technology & Engineering* (05), 193-196.