MNIST Classification Using CNNs and Feed-Forward Neural Networks

Mahtab Faraji.[1] and Fatemeh Taghvaei.[2]

[1] *University of Illinois at Chicago, Department of Biomedical Engineering, Illinois, USA*
[2] *University of Illinois at Chicago, Department of Electrical and Computer Engineering, Illinois, USA*

Reception date of the manuscript:
Acceptance date of the manuscript:
Publication date:

Abstract— Handwritten digit recognition is a challenging task that has been extensively studied using the MNIST dataset. In this project, we discuss various techniques used for MNIST classification, including Convolutional Neural Networks (CNNs) and Feed Forward Neural Network (FNN). We also provided the accuracy per class and compared the results.

Keywords— MNIST, Classification, Convolutional neural networks

I. Introduction

Handwritten digit recognition is a classical problem in the field of machine learning, and the MNIST dataset has become a widely-used benchmark for evaluating the performance of various models in this domain. The dataset consists of 70,000 grayscale images of handwritten digits from 0 to 9, each of which is 28x28 pixels in size. In recent years, neural network models have shown remarkable success in achieving high accuracy in classifying the MNIST dataset. In this paper, we explore two popular neural network models, namely a feedforward neural network and a convolutional neural network (CNN), and compare their performance on the MNIST dataset.

The feedforward neural network is a classical neural network model that consists of one or more hidden layers of neurons between the input and output layers. In our model, the input layer consists of the flattened 28x28 pixel image, and the output layer consists of 10 neurons, each representing one of the possible digit classes (0-9). We use activation functions such as ReLU and Softmax to introduce nonlinearity and produce class probabilities.

The CNN model, on the other hand, is a type of neural network that is particularly well-suited for image classification tasks. It uses convolutional layers to extract features from the input image and pooling layers to reduce the dimensionality of the feature maps. These features are then passed through one or more fully connected layers to produce the final output. CNN models can achieve higher accuracy compared to feedforward models for image classification tasks, especially for complex datasets like MNIST.

In this paper, we present a comparative analysis of the performance of these two models on the MNIST dataset [1]. We pre-processed the dataset, define and train both models, and evaluate their performance on the test set. Our results demonstrate that both models achieve high accuracy in classifying the MNIST dataset, but the CNN model outperforms the feedforward model.

The rest of the paper is organized as follows. In section 2, we describe the preprocessing steps for the dataset. In section 3, we provide a detailed description of the feedforward neural network and CNN models used in this study. In section 4, we present the experimental results and performance analysis of both models. Finally, in section 5, we conclude the paper and discuss future research directions.

II. Method Description

Data Preprocessing: The MNIST dataset was downloaded from the official website and preprocessed before training the models. The images were converted to grayscale, and pixel values were normalized to lie in the range [0,1]. The dataset was then split into training, validation, and test sets, with 50,000, 10,000, and 10,000 images respectively.

Feedforward Neural Network: The feedforward neural network model consisted of an input layer with 784 nodes, two hidden layers with 512 and 256 nodes respectively, and an output layer with 10 nodes representing the digit classes. The ReLU activation function was used for the hidden layers, and the Softmax activation function was used for the output layer to produce class probabilities. Dropout with a rate of 0.5 was applied to the hidden layers to prevent overfitting. The model was trained using cross-entropy loss and the Adam optimizer with a learning rate of 0.001. CNN: The CNN model consisted of an input layer with grayscale images of dimensions 28 x 28, followed by a convolutional layer with 16 filters of size 3 x 3, stride of 1, and padding of 1. A ReLU activation function was applied to the output of the convolutional layer, followed by a max pooling layer with a 2

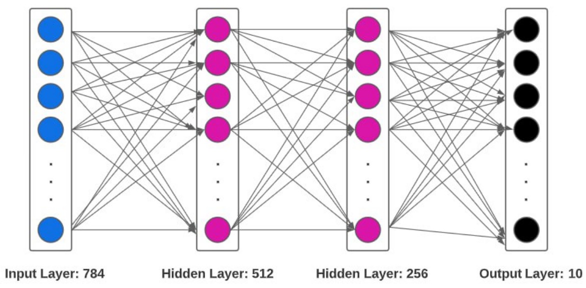

Fig. 1: FNN model

x 2 kernel and a stride of 2. The process was repeated with a second convolutional layer with 32 filters of size 3 x 3, stride of 1, and padding of 1, and a second max pooling layer with a 2 x 2 kernel and a stride of 2. The output of the second max pooling layer was flattened and passed through three fully connected layers with 1568 nodes at the input layer, 512 and 256 nodes in the first and second hidden layers respectively, and an output layer with 10 nodes representing the digit classes. The ReLU activation function was used for the fully connected layers, and the Softmax activation function was used for the output layer to produce class probabilities. The model was trained using cross-entropy loss and the Adam optimizer with a learning rate of 0.001.

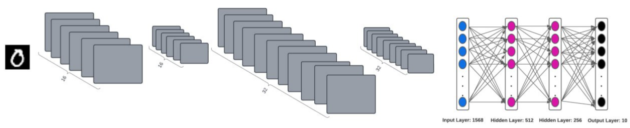

Fig. 2: CNN model

Training and Evaluation: Both models were trained using mini-batch gradient descent on the training set for 20 epochs. The batch size was set to 64 for the feedforward neural network and 128 for the CNN. The models were evaluated on the test set using accuracy as the performance metric. All experiments were conducted using Python programming language and PyTorch deep learning framework.

III. Experimental results

The performance of both models was evaluated using accuracy on the test set. The feedforward neural network achieved an accuracy of 98.14%, while the CNN achieved an accuracy of 99.12%.

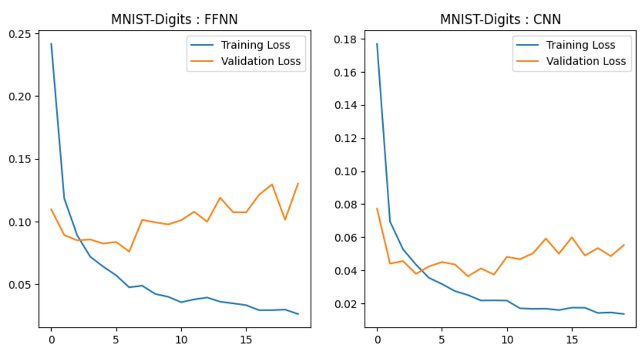

Fig. 3: Training and validation loss curves

TABLE 1: Accuracy of FNN and CNN per class

Class Name	Accuracy (FNN) (%)	Accuracy (CNN) (%)
Class "0"	98.46	99.79
Class "1"	99.11	99.73
Class "2"	97.57	99.03
Class "3"	98.61	99.80
Class "4"	97.96	99.28
Class "5"	99.10	98.99
Class "6"	98.01	99.37
Class "7"	97.66	99.51
Class "8"	96.91	98.66
Class "9"	97.72	97.02

TABLE 2: The overall accuracy of the FNN and CNN

Method	FNN	CNN
Accuracy(%)	98.12	99.13

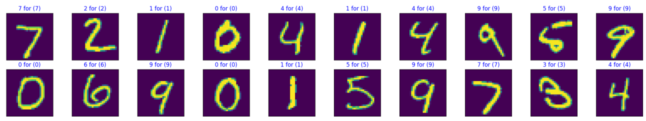

Fig. 4: test results

IV. Conclusion

In conclusion, MNIST classification is a well-studied task that has seen the use of various techniques. Convolutional Neural Networks (CNNs) have emerged as the most popular technique due to their high accuracy and ability to extract features from the input images. The choice of the technique depends on the specific requirements of the task, such as accuracy, speed, and interpretability. Future research in MNIST classification can explore new techniques that can improve the accuracy of the classification task while also reducing training time. Additionally, research can focus on exploring techniques that can work with noisy or incomplete MNIST datasets. Overall, MNIST classification remains an important benchmark dataset for testing and comparing image classification algorithms, and its continued use will drive advancements in the field of computer vision.

V. Contribution of each student to the project

Mahtab Faraji and Fatemeh Taghvaei contributed half and half to the project. We both worked on the Python code together and also wrote the report together.

References

[1] NVS Yashwanth. Mnist handwritten digits recognition using pytorch. *GitHub repository*, 2020.