# Comparison of recognition on hand written digits between MLP network and CNNs

Chao Li

College of Engineering and Computer Science, Australian National University, Canberra, 2601, Australia
u6458350@anu.edu.au

**Abstract.** Hand written digits recognition has been researched for years, and plenty of methods used to implement it, such as CNNs, which is focused in this paper. The paper not only managed a recognition with good accuracy, but also implemented it with modified parameters and structure as well as another algorithm. Experiments shows CNNs do make an improvement beyond simple recognition algorithm, as well as various parameters and structure will influence the performance of CNNs.

**Keywords:** digit recognition, CNNs, parameter, structure.

## 1 Introduction

The neural networks are widely used for image recognition. Based on neural network, various of hand written digits recognition algorithms was achieved, such as such as K-Nearest-Neighbors (KNNs), Support Vector Machines (SVMs), Convolutional Neural Networks (CNNs), etc. According to these algorithms, hand digits can be recognized with relatively high accuracy, which used to be an industry problem [2].

In this article, we deploy only CNN to implement digits recognition. Contrasting with a simpler Multilayer Perceptron (MLP) network, as well as CNNs with modified parameters and structure, readers will have a better understanding about it.

The dataset used to train networks and test is MNIST [3], which is developed for evaluating machine learning models on the handwritten digit classification problem. The dataset was constructed from a number of scanned document dataset available from the National Institute of Standards and Technology (NIST). Images of digits were taken from a variety of scanned documents, normalized in size and centered. This makes it an excellent dataset for evaluating models, allowing the developer to focus on research with very little data cleaning or preparation required. Each image is a 28 by 28-pixel square and there are 10 digits (0 to 9) or 10 classes to predict.

The structure of paper is organized as following. Methods used in this paper, main of which are the procedures of implementing recognition, is in section 2. In section 3, the result of comparison among MLP and CNNs and discussion on it are spread. In the end, conclusion and future work are introduced.

## 2 Method

### 2.1 Implementation of Sample CNNs

At the beginning, we built a CNN as standard sample. With the powerful library PyTorch, we firstly defined a network by creating a new class which extends an encapsulated class with helpful functions like activation functions. As mentioned above, PyTorch is powerful. Within PyTorch, Dataloader class is used to load datasets. It can automatically divide the data into matches as well as shuffle it among other things. What's more, the MNIST data is accessible with torchversion, and the process of creating train and test data tend to be fairly easy. To run several epochs, a train method is defined to run our training loop. To compute the loss, Negative Log Likelihood loss is used to estimate how far away the output is from the target. The update rule we used to model is the Stochastic Gradient Descent (SGD). Similarly, the test function is constructed, but inside which, net.eval() is called to disable some functions should only active in train function. With these procedures, framework and main functions of a CNN is constructed. After adding a repeat loop, the model can be trained for several epochs.

The sample CNNs in this experiment is constructed with 2 convolutional layers, a dropout, 2 max pooling layers. The kernel sizes of convolutional layer and pooling layer are 5 and 2. The forward pass is defined with a ReLU activation function for each hidden layer and dropout.

### 2.2 Modification on Experimental Algorithms

To research the accuracy of CNNs, we implemented another algorithm with MLP [6]. In this algorithm, the net was made up with three simple layers, between which ReLU is used to connect each other. As images from the dataset are of 28 by 28-pixel squares, the fist parameters of first layer have to be 784.

To research performance of CNNs with padding, we modified padding of convolutional layer as 2 and keep other parameters the same (thus kernel size should be 9), to compare with sample CNNs with padding of 0.
To research performance of CNNs with different structure, we rewrite the Net class in which one convolutional layer, one ReLU and one max pooling layer are combined within a sequential, and two of such sequential are used to replace former structure in sample CNNs [5].

# 3 Result and Discussion

With specific part and parameters modified and other parts remained, the differences will be performed are just differences between experimental algorithms networks with sample CNNs.

## 3.1 Comparison among Sample and Experimental Algorithms

According to Fig1, the trends of change on average loss are smaller, and apparently, CNNs performs much better than MLP. Average loss of CNNs is approximately 40% smaller than MLP network during the whole process, and the CNN which combined convolutional layer, ReLU and pooling layer performs best, which loss only 0.038 when all data are executed. While, CNNs with padding performs excellent as well.
According to Fig2, all 4 networks arrive at an approximate 98% recognition accuracy, which improves a lot compared with their initial accuracy of epoch1. CNNs still perform much better than MLP, lead the whole process with nearly 1 more percent. There are differences among CNNs as well. Sample CNN is still the moderate one, gradually increasing on accuracy with epoch grows, but performed worse than modified CNNs. CNN with combined sequential looks more stable than CNN with padding, former of which gradually increase at a bigger initial accuracy while latter of which has a minor sudden drop at epoch9.

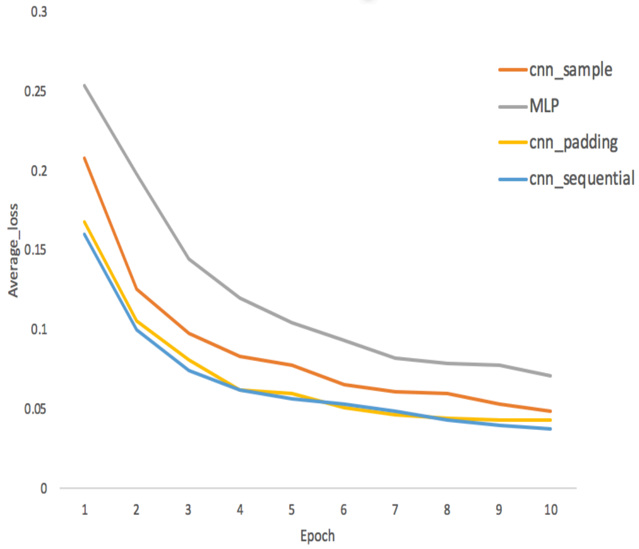


Fig 1. Average loss of 4 implementation

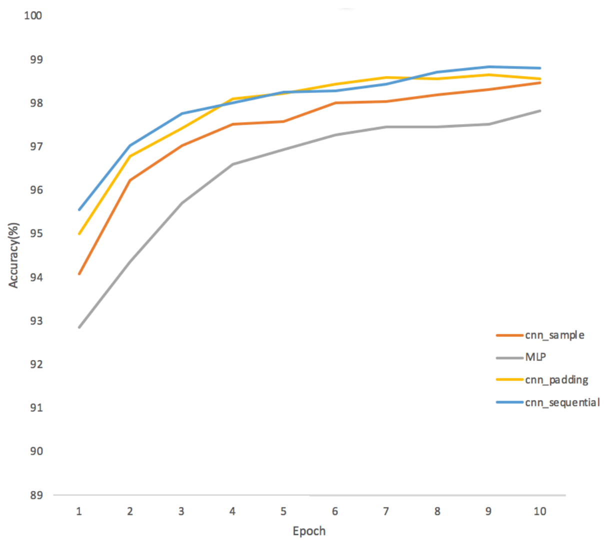


Fig 2. Accuracy of 4 implementation

Comprehensively, CNNs is a relatively efficient algorithm on hand written digit recognition. It can obtain an accuracy of nearly 99% with enough training. While, there are various modification can be exerted on CNNs. As analyzed above, convolutional layer with padding can decrease the average loss and increase the accuracy of CNNs with an extent cannot be ignored. The structure of layers inside CNNs is also of much importance. With same numbers of convolutional layers and max pooling layers, the CNNs with sequential performs much better than sample CNNs though they are likely with same layer sequence.

## 3.2 Comparison with Other's Research

CNNs is a moderate structure, as contrast nomination layer can be added and other algorithms can be modified to combine with CNNs, like a hybrid CNN-SVM algorithm managed in 2012. Niu and Suen have CNNs works as a trainable feature extractor and SVM performs as a recognizer[1]. Simply speaking, they replace the last output layer of CNNs with SVM with a Radial Basis Function (RBF) kernel, the SVM has output of CNNs as input for training to make further classification, and in this way, they managed a recognition accuracy on the same dataset MNIST with 99.81%, which is 1 percent bigger than mine and nearly perfect.

# 4 Conclusion and Future Work

## 4.1 Conclusion

We can implement hand written digits recognition with network as simple as Multilayer Perceptron network, but it is relatively inaccurate. To improve the performance, CNNs is considered and which also managed on both bigger accuracy and less information loss. Further, CNNs can be modified aiming better performance. According to this paper, adding padding and optimize the internal structure are all good choices. While, CNNs can also be modified on other aspects to achieve a better result, various alternatives like KNNs will do a different job as well.

## 4.2 Future Work

For CNNs, there are plenty of places to improve. The processing spends much time, which is about three times over that of MLP network, so pruning on the neurons should be considered [5]. As there are many activation and update methods available, and the paper just used ReLU and SGD, various methods will be used within CNNs in the future. What's more, accuracy of other algorithms should be researched, like hybrid CNN-SVM algorithm. Such combinations of algorithms look interesting and are of much potential.

# References


1. Niu X.X., Suen C.Y. :A novel hybrid CNN–SVM classifier for recognizing handwritten digits. In Pattern Recognition, pp. 1318–1325, vol. 45 (2012).
2. Deng L.: The MNIST Database of Handwritten Digit Images for Machine Learning Research. In: IEEE Signal Processing Magazine pp: 141 – 142, Volume: 29, Issue: 6, (2012).
3. LeCun Y., Cortes C. & Burges C.J.C.: The MNIST Database of handwritten digits, http://yann.lecun.com/exdb/mnist/.
4. Han S., Mao H.Z. & Dally W.J.: Deep Compression: Compression Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In ICLR (oral) (2016).
5. Zhou M.: Convolutional Neural Network, https://morvanzhou.github.io/tutorials/machine-learning/torch/4-01-CNN/ (2017).
6. Brownlee J.: Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras. In: Machine Learning Mastery, https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/ (2016).