NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[AI6121]
Computer Vision

Project: MNIST Handwritten Digit Recognition

Li Hantao, G2101725H, MSAI, hli038@e.ntu.edu.sg
Other Team Members

The *Handwritten Digit Recognition* is the ability of computers to recognize human handwritten digits. This project will work on the dataset MNIST, which is probably one of the most popular datasets among machine learning and deep learning enthusiasts.

This project consists of the following tasks:

1) Design and develop your handwritten digit recognition network. Train your network over the MNIST training set, evaluate over the test set, and report your evaluation results.

2) Investigate different hyper-parameters and design options such as learning rate, optimizers, loss functions, etc. to study how they affect the network performance.

3) Benchmark with the state-of-the-art and discuss the constraint of your designed network and possible improvements.

CONTRIBUTIONS

Li Hantao: Implementation of the Resnet-18; Investigating of different hyper-parameters in Resnet-18 design; Introduction Part; Formatting the report. (Section I & III)

Other Team Members

I. INTRODUCTION

Handwritten digit recognition is the ability of computers to recognize human handwritten digits. The traditional machine learning method, such as *Linear Classifiers* and *K-Nearest Neighbors*, typically cannot obtain satisfying outcomes in classification. Consequently, we will implement methods of deep learning, including multi-layer perceptron (MLP) and convolutional networks. Firstly, we will introduce the dataset we use and the network we implemented.

A. MNIST

MNIST database (**M**odified-**N**ational-**I**nstitute-of-**S**tandards -and-**T**echnology-database) is a large database in which the handwritten digits are usually used to train image processing and machine learning systems, for instance, the digits recognition system we will implement [1]. It was founded by remixing the samples from the NIST dataset. The authors believe that since NIST's training-set is from U.S. Census Bureau employees while the test-set is from U.S. high school students, it is not suitable for machine learning experiments. In addition, the black-and-white image from NIST is normalized and anti-aliased to fit the boundary box of 28x28 pixels.

MNIST contains 60,000 training images and 10,000 test images. Dataset uses a two-dimensional array picture, which means [28,28], to represent each handwritten numeral. Each element in the array corresponds to a pixel; thus, the size of each image is fixed at 28x28 pixels, shown in Fig. 1.

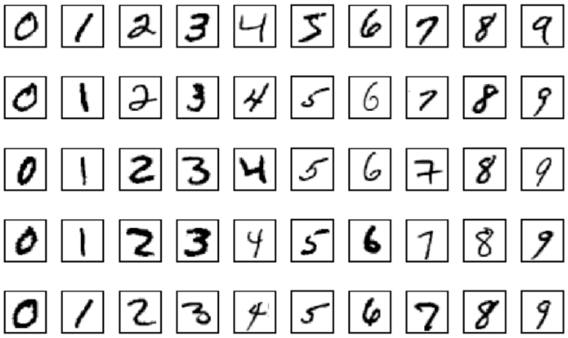

Fig. 1. Examples of MNIST [1].

The images in MNIST are 256-level gray images, where the value 0 represents white (background) .and 255 represents black (foreground). The uint-8 data type within the value of [0,255] is used to represent the image. In order to speed up the training, we usually need to standardize the data and scale the gray value to the float-32 data type within the value of [0,1].

Before implementation, we will analyze some characteristics of the image in MNIST at the beginning, which may be valuable in the following recognition system design.

Firstly, almost all handwritten digits are located in the central part of the image, with no severe imbalance in image proportion or incomplete digitals, which can be observed in Fig. 1. Secondly, the background of the images is extremely clear. The parts except the digits are pure black, which helps the model capture valuable features. Thirdly, scrawl fonts are difficult for even humans to distinguish in MNIST, shown in Fig. 2. These characters are illegible for the computer models we design.

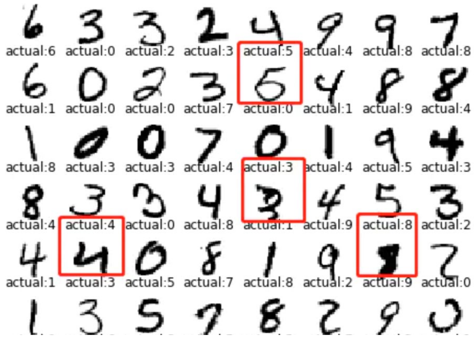

Fig. 2. Examples of illegible digits in MNIST

B. Multi-Layer Perceptron

MLP (Multi-Layer Perceptron) can be said to be the simplest artificial neural network structure. MLP is a fully connected network, which can be regarded as a network model composed of multiple hidden layers. Each node of the hidden layer is connected to all nodes of the previous layer, so it is called fully connected. After the input of MLP passes through several hidden layers, it enters the output layer to get the output.

We will utilize a fundamental design of MLP to introduce the basic idea of full connected neural network, as shown in Fig. 3

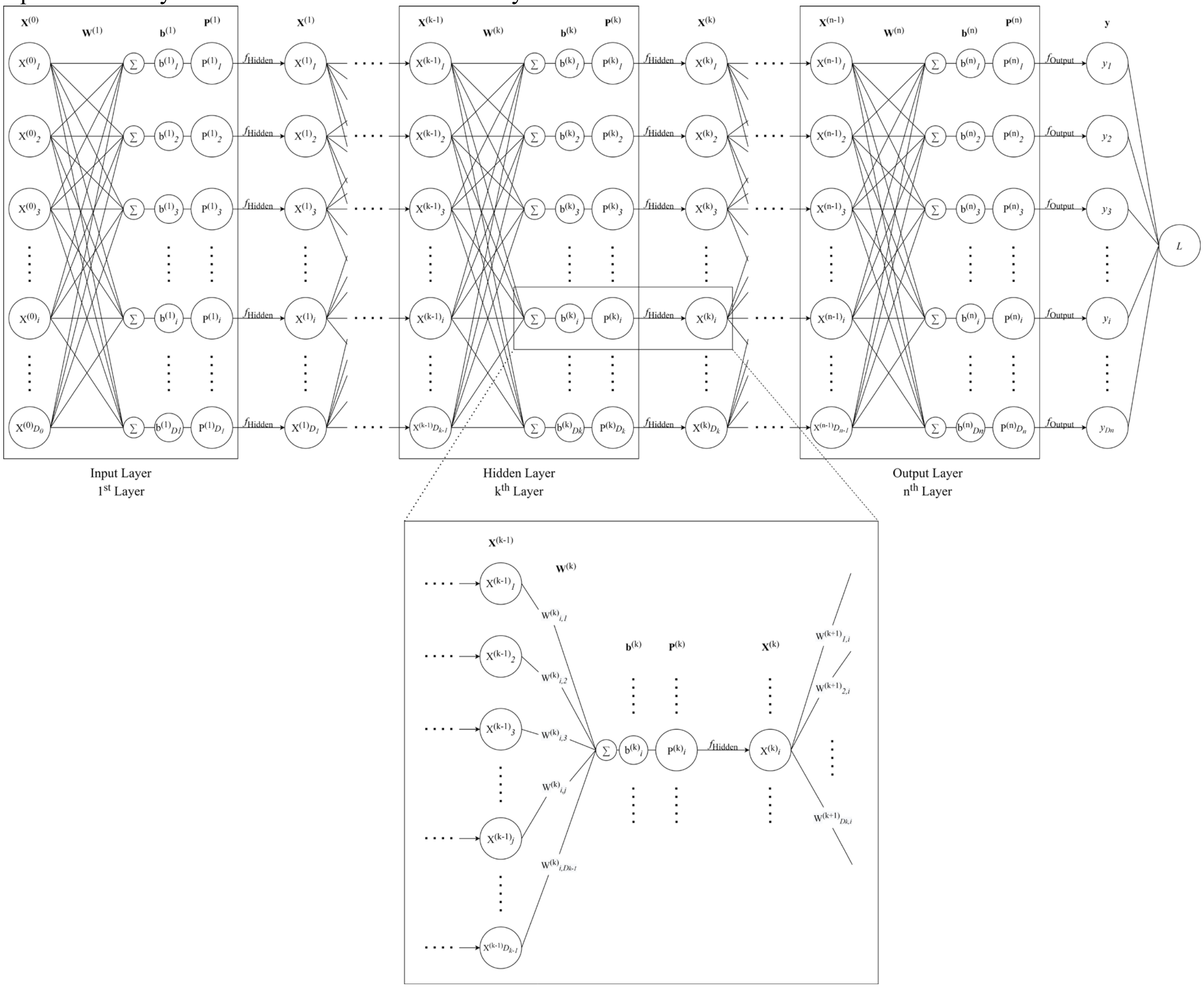

Fig. 3. Structure of *n* layers fully connected neural network (MLP).

This is an *n* layer's fully connected neural network. Single-layer contains an input column vector **X**, a weight matrix **W**, a bias column vector **b**, and a hidden layer column vector **P**, whose element is used as an independent variable passing through the function f_{Hidden} to obtain the **X** of the next layer. In vector form, we can represent the input component in a more comprehensible way:

$$\mathbf{X}^k = \begin{bmatrix} \mathrm{X}_1^k \\ \mathrm{X}_2^k \\ \vdots \\ \mathrm{X}_{D_k}^k \end{bmatrix}$$

For the task of handwriting digital recognition with MNIST, the input vector is an image in the training-set. In a more particular representation, the input vector is the column arrangement of 784 pixels, as shown in Fig. 4.

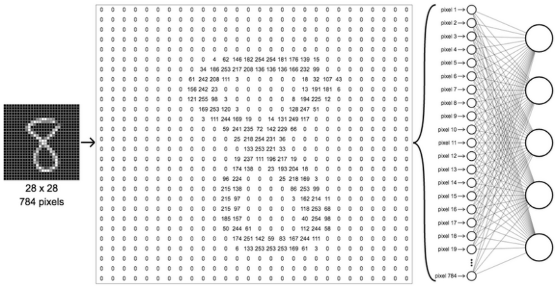

Fig. 4. Examples of the image as an input vector.

When using the *Gradient Descent* method to update the parameters, we need to update the weight matrix **W** and the bias coefficient vector **b** in each layer of the neural network.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^k} = \frac{\partial \mathcal{L}}{\partial \mathbf{P}^k} \cdot \frac{\partial \mathbf{P}^k}{\partial \mathbf{W}^k} \qquad \frac{\partial \mathcal{L}}{\partial \mathbf{b}^k} = \frac{\partial \mathcal{L}}{\partial \mathbf{P}^k} \cdot \frac{\partial \mathbf{P}^k}{\partial \mathbf{b}^k}$$

With the Gradient descent updating method, we can obtain the expression of training equations. α is the learning rate.

$$\mathbf{W}_{(s+1)}^{k} = \mathbf{W}_{(s)}^{k} - \alpha \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{W}_{(s)}^{k}}$$

$$\mathbf{b}_{(s+1)}^{k} = \mathbf{b}_{(s)}^{k} - \alpha \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{b}_{(s)}^{k}}$$

In this way, the weight matrix and basis vector in each state will update through the cost value from the loss-function in the epoch. Eventually, we can obtain the convergence value in each neuron, where the final model of MLP is.

II. Digits Recognition with MLP

…

III. Improvements

A. *Defects of MLP*

From the theoretical analysis and experimental results in the previous two sections, we can find that the training of an MLP network needs to store a considerable number of parameters. As the number of layers increases, the requirements for the model training platform are pretty costly. The optimal NN model displayed on the MNIST database website is a 6-layer NN, where the total number of neurons reaches 8294, which is too large for a common computing platform to realize training [2]. However, there is an apparent correlation between the accuracy of model recognition and the number of neurons. Table V can observe the selected NN part of the model and its accuracy to prove the above suppose. Therefore, we need to find a new idea, that is, a convolution network. From Table V, we can notice that the classification accuracy of the CNN model is significantly higher than that of the MLP model, and the state-of-art model on the website is also a CNN model [3]. Therefore, it is meaningful to implement a CNN-based handwritten digital recognition model to improve the properties of the model.

TABLE V
Training Result of Different Learning Rate

Classifier Type	Classifier Details	Accuracy (%)
MLP	2-layer NN, 300 hidden units	95.30
	3-layer NN, 300+100 hidden units	96.95
	3-layer NN, 500+300 hidden units	98.47
	6-layer NN 784-2500-2000-1500-1000-500-10 hidden units [2]	99.65
CNN	Convolutional net LeNet-1	98.30
	Convolutional net LeNet-4	98.90
	Large convolutional net	99.40
	Large/deep convolutional net, 1-20-40-60-80-100-120-120-10	99.65
	Committee of 35 convolutional net, 1-20-P-40-P-150-10 [3]	**99.77**

B. *ResNet-18*

In CNN, we chose an uncomplicated and easy-to-implement network, Residual neural network (ResNet), proposed by He Kaiming et al. of Microsoft Research Institute [4], which won the championship in ILSVRC in 2015.

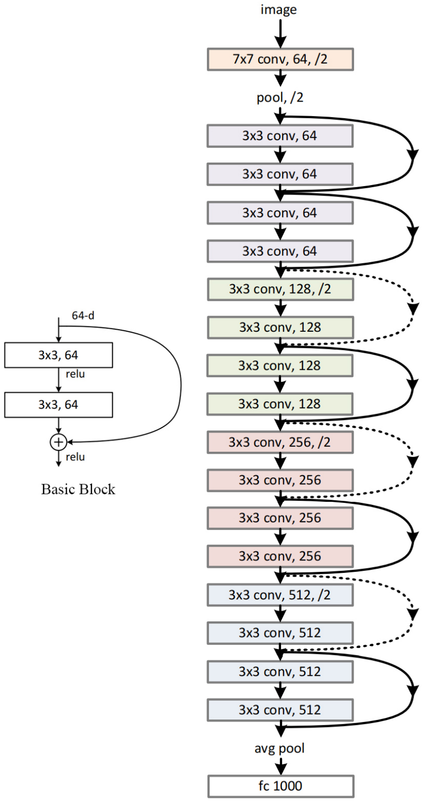

Fig. 8. Architecture of ResNet-18 [4]

ResNet proposes two block models; one is 'basic', the other is 'bottleneck'. ResNet-18 utilizes the shallow basic block, where the two layers have equal output channels, as shown in Fig. 8. After successful implementation, we find that the MNIST can get a high convergence within 15 epochs: thus, we apply comparative experiments to test the influence of hyper-parameters of ResNet-18 training.

C. *Learning Rate*

Firstly, we design a series of experiments to examine the influence of the learning rate. For each learning rate, we conducted ten repeated trials and took the mean of the final results. The network is trained with three different η (0.001, 0.01, and 0.1), with 15 as the epoch. The results and curves are shown in Table VI and Fig. 9.

We can observe from the data that the best results, whether loss or accuracy, are the training conducted by the learning rate of 0.01. It can be easily understood by considering the convergence velocity, where the step size in the descending direction of the loss function gradient, under η=0.001, is small, while the bias of each approximation will be enlarged so that the downward trend of the loss function will be impaired under η=0.1.

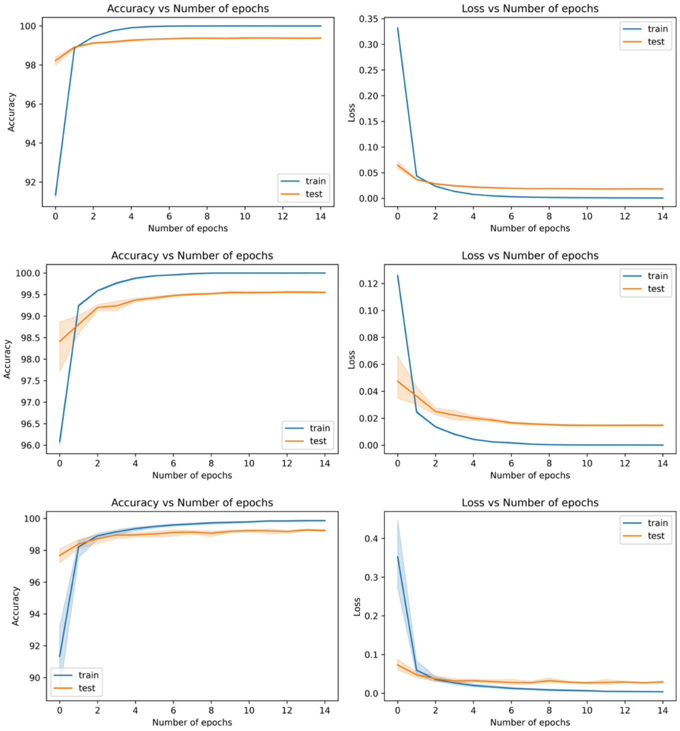

Fig. 9. Curve of training result with different learning rate (from up to down is η=0.001, 0.01, and 0.1).

TABLE VI
TRAINING RESULT OF DIFFERENT LEARNING RATE

Learning Rate η	Training Loss	Training Accuracy	Test Loss	Test Accuracy
0.001	0.0008	100.00	0.0185	99.39
0.01	**0.0002**	**100.00**	**0.0147**	**99.46**
0.1	0.0039	99.87	0.0292	99.25

D. *Learning Rate Schedule*

We choose *Cosine Annealing* as the learning rate schedule method in the experiment. The results and curves generated are shown in Fig. 10 and Table VII.

TABLE VII
TRAINING RESULT OF DIFFERENT LEARNING RATE SCHEDULE METHOD

Schedule Method	Training Loss	Training Accuracy	Test Loss	Test Accuracy
Constant η	0.0002	100.00	0.0147	99.46
Cosine Annealing	**0.0001**	**100.00**	**0.0139**	**99.55**

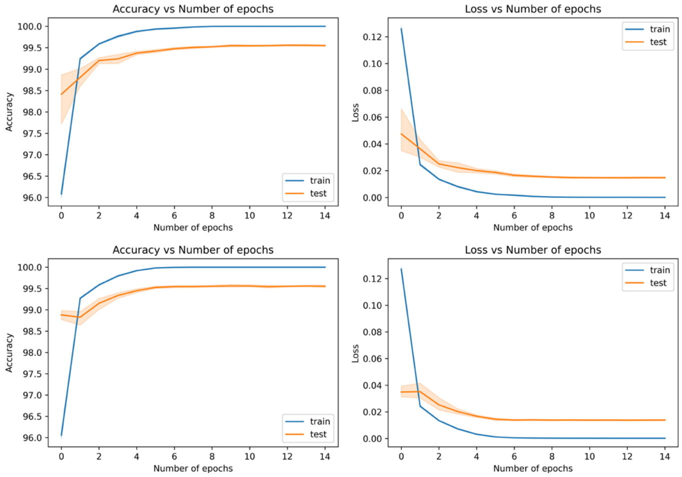

Fig. 10. Curve of training result with different learning rate schedule method (from up to down is constant and cosine annealing).

E. *Weight Decay*

Now we will discuss the influence of regularization on the model. We modify coefficient λ on the weight decay regularization process to set up a series of experiments, and the results are shown in Fig. 11 and table VIII.

It can be seen that the addition of weight decay effectively reduces the overfitting phenomenon of the model to the training set, that is, the increase of training loss and the improvements in the final classification accuracy.

We can notice that when the regularization coefficient is too large, there is a drastic underfitting phenomenon in the early stage of training. Although the model finally shows the best accuracy, this is due to cosine annealing restricting it to a reasonable range in the final stage. This cannot indicate that this coefficient is optimal. Therefore, we believe that the $\lambda=5\times10^{-4}$ is the most appropriate. Of course, to prove our argumentation, we will attach the model with the $\lambda=1\times10^{-2}$ in subsequent experiments, used as the control group.

TABLE VIII
TRAINING RESULT OF DIFFERENT WEIGHT DECAY PARAMETER

Weight Decay λ	Training Loss	Training Accuracy	Test Loss	Test Accuracy
1×10^{-2}	0.0228	99.97	0.0269	99.70
2×10^{-3}	0.0014	100.00	0.0128	99.57
5×10^{-4}	**0.0003**	**100.00**	**0.0127**	**99.58**

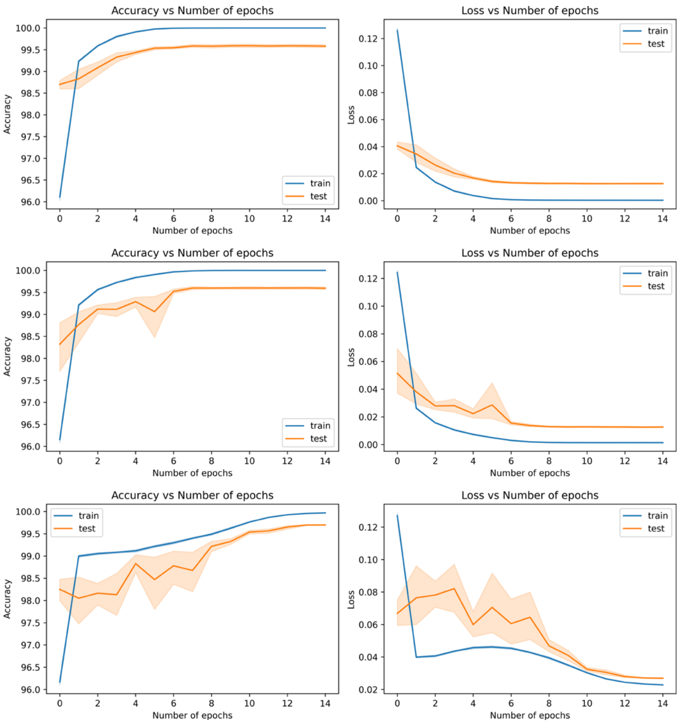

Fig. 11. Curve of training result with different weight decay coefficient

F. *Data Augmentation*

Data augmentation can make the model learn more effective features and ameliorate the overfitting phenomenon. We generally select the RandomErasing function in the *PyTorch* library as the data augmentation. We randomly mask the image with the parameter of size=0.02-0.2 and probability=0.5. Because the model needs more epochs to train after regularization, in this experiment, we increase the number of epochs to 300 and only train once. The experimental results are shown in Table IX and Fig. 12.

TABLE IX
TRAINING RESULT OF DATA AUGMENTATION

Weight Decay λ	Training Loss	Training Accuracy	Test Loss	Test Accuracy
1×10^{-2}	0.0478	99.09	0.0295	99.54
5×10^{-4}	**0.0033**	**99.91**	**0.0156**	**99.60**

Firstly, we can observe that the addition of data augmentation further reduces the overfitting of the model, which is reflected in the improvement of classification accuracy. Secondly, the experimental results prove our point of view above; that is, the weight decay coefficient of 0.01 is too large to be suitable for model training.

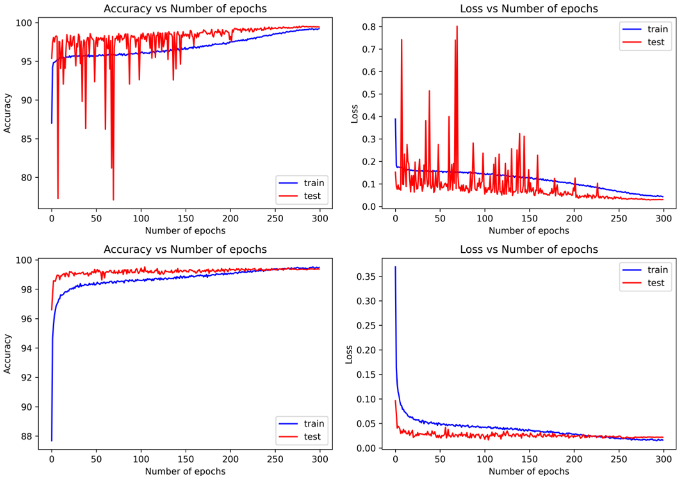

Fig. 12. Curve of training result with different RandomErasing parameters

G. *Discussion*

After a series of experiments based on CNN, we can see that after changing the MLP model to the CNN model, the model's classification accuracy has been significantly improved without a significant increase in the amount of calculation. This shows that CNN has advantages over MLP for image processing.

Secondly, we notice that under the most basic model, that is, only the learning rate schedule, the model has achieved excellent performance. Our subsequent regularization did not bring significant performance improvement to the model (99.55-99.58-99.60). There are two reasons for this. First, for OCR tasks such as handwritten numeral recognition, the regularization method is generally distortion rather than masking; therefore, our regularization does not achieve satisfying results. Secondly, the MNIST data set is a simple and clear dataset in which the images have no noise. The model is not easy to learn useless features; thus, the improvement of regularization is limited.

REFERENCES

[1] http://yann.lecun.com/exdb/mnist/

[2] Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep big simple neural nets excel on hand-written digit recognition. CoRR abs, 1003.

[3] Cireşan, D., & Meier, U. (2015, July). Multi-column deep neural networks for offline handwritten Chinese character classification. In 2015 international joint conference on neural networks (IJCNN) (pp. 1-6). IEEE.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).