IMA 7221 - Signal Processing and Statistical Data Analysis
Report for Convolutional Neural Network

Xuefeng Wei
Institut Polytechnique de Paris
xuefeng.wei@ip-paris.fr

Abstract

The MINST dataset (with 60,000 28x28 black and white images) was used in this lab project.The dataset was separated into 50,000 training images and 10,000 test images. Convolutional Neural Network (CNN) was trained on this dataset. Four CNN architectures were designed and trained on this dataset with 50 epochs, producing 90%, 79%, 92%, 95% accuracy respectively. Convergence and Confusion Matrix were discussed. Finally, The advantages and disadvantages of four different CNN models are summarized and future directions for improvement and work are proposed.

Index Terms——Convolution, Neural Network

1. Introduction

The MNIST dataset was chosen for this lab project, which consistes of 60,000 28x28 black and white handwritten digits images in 10 different classes. The 10 classes are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. These are common digits in handwritten in daily-life,it is meaningful to create and test deep learning algorithm on this dataset for classification.

One of the most striking features of the MNIST database is the relatively small size of each image (28x28) and the black and white colors, which means that we can use convolutional neural networks with different structures to train quickly on this dataset to compare and test their performance capabilities.

In this lab, the main algorithm used is a convolutional neural network. Because convolutional neural networks are widely used in applications targeting images. Testing the performance of CNNs of different architecture by accuracy as a metric, since each class has a similar number of training and testing example. Four different structures of convolutional neural networks were designed and applied to this dataset to compare their performance. The dataset was seperated to 50,000 training data and 10,000 testing data.

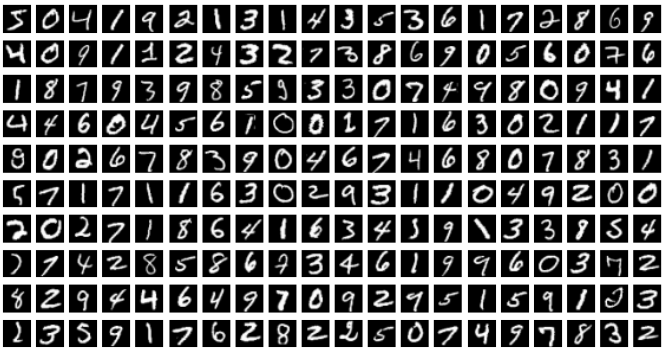

Figure 1. MNIST Sample Images

1.1. Data Exploration

The data can be found from the URL: http://yann.lecun.com/exdb/mnist/. Each class is considered to be completely independent. Figure 1 shows the first 200 items of MNIST which include all the classes:

This dataset has widely used in machine and deep learning research. The similar datasets of MNIST involve Fasion MNIST and Kuzushiji MNIST.

1.2. Definition of variables

Since the keras datasets library already contains MNIST datasets, it can be imported directly. 4 variables are defined to store the data, they are x_train, y_train, x_test, y_test.Here is what each of them means.

x_train: data from the training set. y_train:label from the training set. x_test: data from the testing set. y_test: label from the training set.

2. Classification Experiments

Since the main goal of this experiment is CNN, four different structures of CNNs were designed and trained on the same dataset to measure their performance. The MNIST training dataset was imported from the keras library and had a shape of (60000, 28, 28). The 28 x 28 black and white im-

ages were then reshaped to have the shape(60000, 28, 28, 1). Since the function of Conv2D need a 4-dim parameter.

2.1. Loss Criterion and Optimizer

Cross entropy loss was used to be the loss function for all models in multi-class classification task.

$$L = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{1}$$

Cross entropy loss was wildly used in classification task since it avoids learning slow-down and helps with the vanishing gradient problem from which deep neural networks suffer, also because softmax was choose as the output layer, and it is a probability distribution.

The optimizer that used in the experiments was Adam optimizer, The Adam optimization algorithm is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications in computer vision and natural language processing. Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems and is relatively easy to configure where the default configuration parameters do well on most problems. In this experiment default parameters were used.

2.2. Convolutional Neural Networks

In order to compare the performance of neural networks with different architectures, four neural models with different structures are applied, CNN1, CNN2, CNN3, CNN4, where CNN1 is the base CNN architecture, so it is used as the baseline

2.3. Hyper-parameters Setting

To measure the performance of different models, the hyperparameters of the convolutional neural network were fixed in this experiment.

Here is how they are set up: epoch is 50, batch-size is 128, metric is accuracy, cross entropy loss and adam optimizer as loss function and optimizer function.

2.4. CNN1

The exact architecture used of the first CNN can be found in Figure 2. The chosen kernel for convlutional layer were 3x3 and the activation function used in this study was the ReLU, which has been widely adopted in other studies. Also before the dense layer of 128 neurons, a flatten layer was set and a dropout layer was set after the Max-Pooling layer and fully connected layer with 128 neurons. The total training parameters for this CNN1 is 1,199,882.

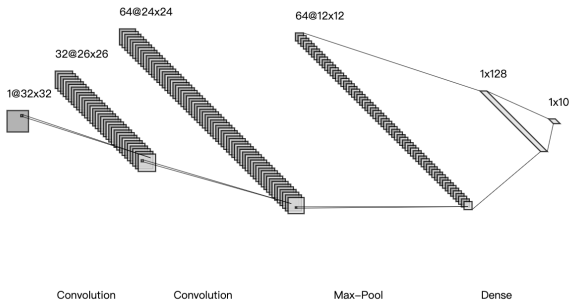

Figure 2. Architecture of CNN1

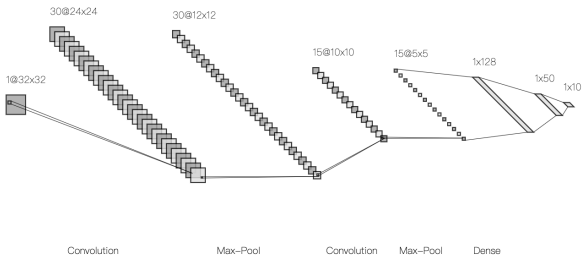

Figure 3. Architecture of CNN2

2.5. CNN2

To explore the effect of different kernal numbers of convolutional layers on convolutional neural networks, and the effect of architecture, CNN2 was proposed and applied to the MNIST dataset. The exact architecture of CNN2 can be found on Figure 3.

In this case, we can see there are 2 convolutional module which has one convolutional layer and pooling layer,and the number of convolutional sampler is 30 and 15 respectively, the kernel size for convolutional layer is 5x5 and 3x3 respectively. The total training parameters is 59,933 which is much lesser than CNN1, we can expect that its training speed will quicker than CNN1.

2.6. CNN3

Based on CNN1 and CNN2 architectures, we consider adding Batch-normalization layer already more convolutional layers as CNN3, The exact architecture of CNN3 can be found on Figure 4. For CNN3, 3 convolutional module has been considered. First convolutional layer with 32 samplers and 5x5 kernel size, the second and third module with max pooling layer and same kernel size. One batch-normalization layer added before the output. The total number of parameters are 160,842. it is more than CNN2 and less than CNN1 but expected to have better performance

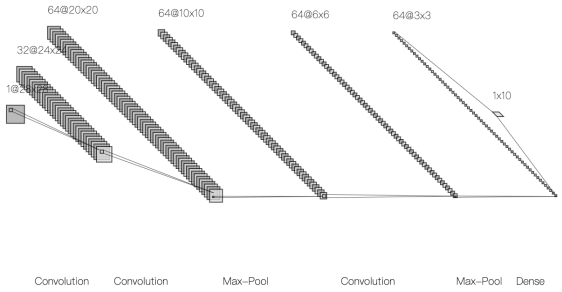

Figure 4. Architecture of CNN3

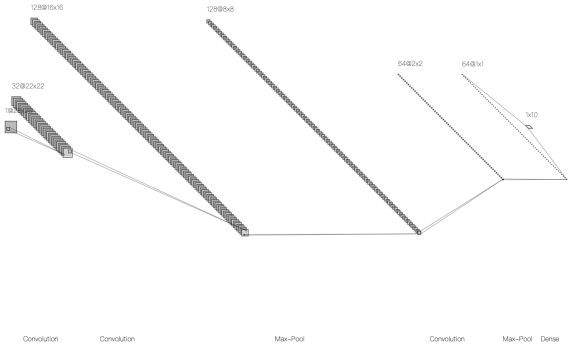

Figure 5. Architecture of CNN4

Model	Accuracy
CNN1	0.7807
CNN2	0.2208
CNN3	0.7703
CNN4	**0.8573**

Table 1. Accuracy when epoch is 10

than CNN1.

2.7. CNN4

To improve the accuracy on the test set, increase the kernel size value of the convolutional layers and more batch-normalization layers, and set the padding to valid in order to expect a high accuracy model. The exact architecutre of CNN4 can be found on Figure 5.

In this case, we can see it is similar with CNN3 but with more convolutional samplers at each layer, also extra normalization layer added between the module1 and module2 and add a hyperparameter padding with valid. The total numbber of parameters are 605,322 which has the most number of parameters among the CNNs proposed before, however, it would also bring the best performance among these models.

Model	Accuracy
CNN1	0.8534
CNN2	0.3498
CNN3	0.8661
CNN4	**0.9064**

Table 2. Accuracy when epoch is 20

Model	Accuracy
CNN1	0.8778
CNN2	0.5427
CNN3	0.8987
CNN4	**0.9278**

Table 3. Accuracy when epoch is 30

Model	Accuracy
CNN1	0.8924
CNN2	0.7113
CNN3	0.9175
CNN4	**0.9416**

Table 4. Accuracy when epoch is 40

Model	Accuracy
CNN1	0.9018
CNN2	0.7978
CNN3	0.9297
CNN4	**0.9503**

Table 5. Accuracy when epoch is 50

3. Performance Comparison

First, the test accuracy of the model is compared, which is the only metric in this experiment, and the closer it is to 1, the better the classification ability of the model.5 different epoch values are set for the capability of the model at different epochs.The performances can be found in Table 1, 2, 3 and 4 when epoch is 10, 20, 30, 40, 50. The accuracy of CNN4 maintains the highest value at different epochs, it has the best performance among these models.

Next, we consider the complexity of different models, we can evaluate it by there number of parameters and its running time under same software and hardware environment.The number of parameters has mentioned before, from the number, we can know the most complex model is CNN4, then CNN1, then CNN3 then CNN2.From the running time, the exact data can be found on Table 6. From the data, we can see that CNN2 runs the fastest because it has

Model	Running Time(s)
CNN1	170
CNN2	**101**
CNN3	212
CNN4	261

Table 6. Running time when epoch is 50

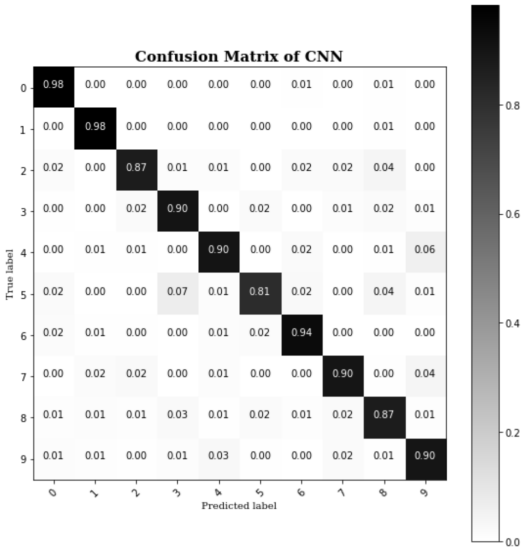

Figure 6. Confusion Matrix of CNN1

the least number of parameters to calculate, but it has the worst performance.

3.1. Confusion Matrix

In order to visualize the differences between the models and to visualize the data, confusion matrices were introduced to measure the differences in the models.

From the confusion matrix, it is clear that CNN4 is darker on the diagonal and the rest of the squares are whiter, which means that the model performs better on the classification task, CNN3 performs better than CNN1 because CNN1 contains more ”impurity” squares, and CNN2 performs the worst, with some squares on the diagonal not close to black and many ”impurity” squares around it.

3.2. Convergence

In order to discuss the convergence of different models, graphs are introduced to visually measure the convergence ability, and convergence of accuracy and loss are considered simultaneously. The exact curve can be found on Figure 10 to Figure 13.

In terms of convergence, CNN1, 3, and 4 converge relatively quickly, while CNN2 converges more slowly, converging near epoch=50 and its value is not very high in comparison.It is clear that CNN4 performs best in terms of convergence of accuracy and loss, converging at epoch=4 or 5, with a very significant rapid increase or decrease observed, while CNN1 and CNN3 are relatively flat. Faster decreases in the curve can be observed.

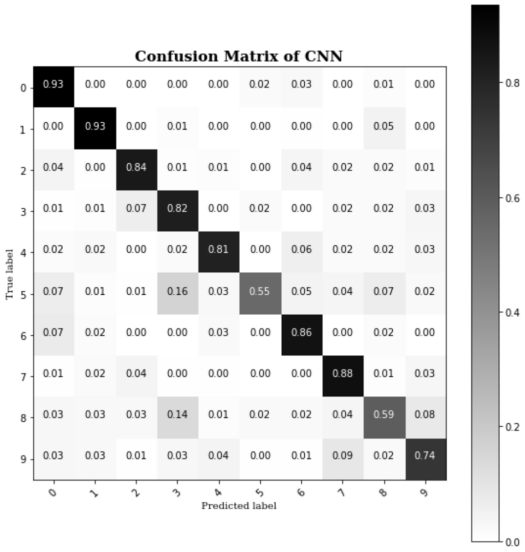

Figure 7. Confusion Matrix of CNN2

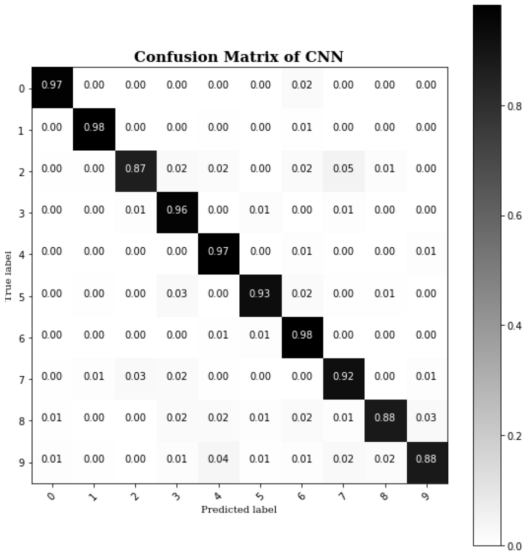

Figure 8. Confusion Matrix of CNN3

4. Final Conclusion and Future Work

On the whole, CNN4 has the highest accuracy and the fastest convergence while taking the longest time, CNN1 and CNN3 are more moderate, and CNN2 is not as accurate but has fast operation speed and low computational bur-

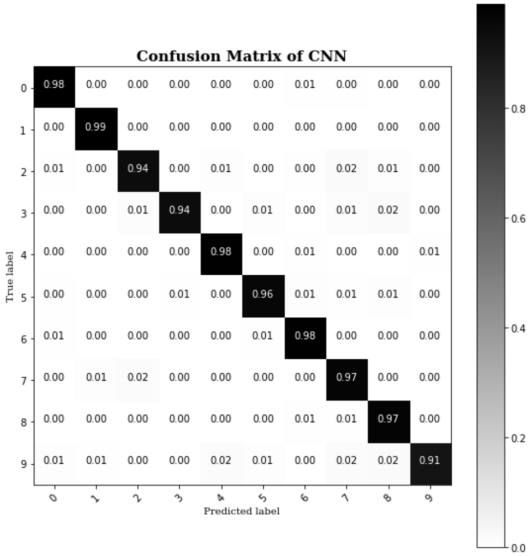

Figure 9. Confusion Matrix of CNN4

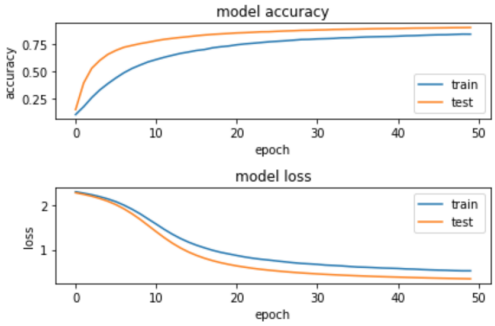

Figure 10. Convergence on Loss and Accuracy of CNN1

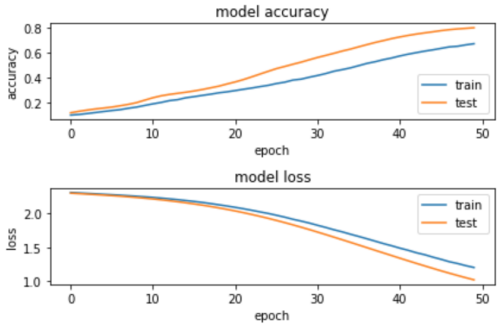

Figure 11. Convergence on Loss and Accuracy of CNN2

den.However, with today's increasing computing resources, this runtime difference decreases and can be ignored on better devices.For reality, CNN4 is the desired model and is worth sacrificing computation time for better performance capabilities.

In image classification, CNN plays an important role so

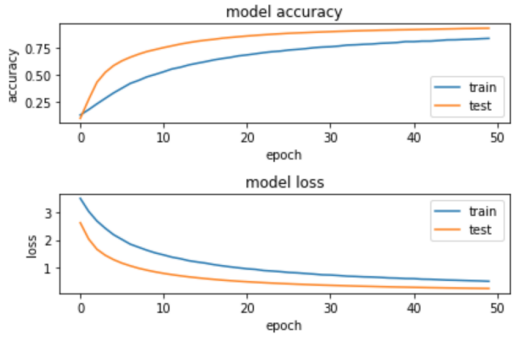

Figure 12. Convergence on Loss and Accuracy of CNN3

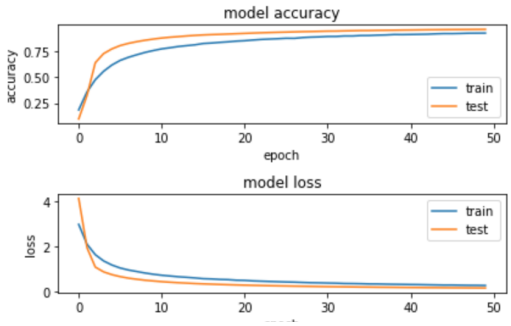

Figure 13. Convergence on Loss and Accuracy of CNN4

far, although more and more advanced models are proposed, such as transformer Despite the different architectures discussed in this report, there is much more to explore with CNNs.In image classification, CNN plays an important role so far, although more and more advanced models are proposed, such as Transformer, Swin Transformer and MLP-Mixer in Computer Vision area.

In future work, CNNs can be compared with more traditional machine learning models Some traditional machine learning algorithms, such as SVM or KNN, can be fused into CNNs to improve performance and thus invent hybrid models

More, discuss the impact of CNN hyperparameters on the model and the implementation of a more SOTA models compared to CNN and integrated with it.