Network Pruning Technique – Implementation and Analysis

Jinshuai Ma,

Research School of Computer Science, Australian National University
u5870682@anu.edu.au

Abstract. In this research, network pruning technique has been implemented and applied on different neural network models, namely simple two-layer neural network and convolutional neural network (CNN). Cross validation method is used to test the performance of the network models before and after pruning. Then, some experiments are conducted to investigate the aspects that may affect pruning performence and then improve them. After improvment, the new network models have been validated again and the results are compared to those of a published paper, which uses other classification methods on the same dataset. The preliminary results of validation show that both network models has a very high accuracy without pruning, but the CNN break down after pruning while simple two-layer network remains high accuracy. The result of experiments shows that the training process, particularly batch size have important efftect on pruning performance. After improvement, both network models have robust performance and high accuracy which is better than the result of the published paper.

Keywords: Neural Network, Convolutional Neural Network, CNN, Pruning, Hidden Neuron, Batch Training

1 Introduction

The main focus of this research are implementation and analysis of network pruning technique which measures and removes similar and complementary hidden neurons using distinctiveness. To achieve this, a dataset *Optical Recognition of Handwritten Digits* [1] is chosen, and two neural network models, a simple two-layer neural network and a CNN model are designed to perform classification on the chosen dataset. Based on these network models, the network pruning technique is implemented and tested. Then some experiments are conducted to investigate the aspects that affect its performance. All the model and technique are implemented using Python programming and Pytorch deep learning library [2].

1.1 Distinctiveness and Network pruning

Distinctiveness is a measurement of similarity and complementarity of hidden neurons of a neural network. It is determined by the hidden neuron activation vector [3]. Each hidden neuron has an output activation vector over a certain number of input data. If the angle between two output activation vectors is less than 15 degrees, then the two hidden neurons are regarded as similar and one of them can be removed. If the angle between two output activation vectors is greater than 165 degrees, then the two hidden neurons are regarded as complementary, and both of them can be removed without significant effect of the network functionality and accuracy [3].

This process that remove similar hidden neurons is called network pruning. Network pruning can remove redundant hidden neurons thus reduce the size of the network and computational load [4]. Pruning is applicable on fully connected layers, particularly, the hidden layer of simple two-layer neural network and fully connected layer of convolution neural networks.

1.2 Dataset information

The chosen dataset is *Optical Recognition of Handwritten Digits* [1], which contains written digit, 0-9, from 43 people. These data are stored both in original format and in pre-processed format. The original format is normalized bitmaps with 32 pixels in height and 32 pixels in width. The value of pixel is either 0 or 1. A class (label) attribute is attached to each bitmap, indicating the actual digit. The pre-processed data format has 64 features and 1 class attribute. All the 64 features are integers ranging from 0 to 16, and the class attribute is in range from 0 to 9. The dataset has been divided into training set and testing set originally. The training set has 3823 instances and the testing set has 1797 instances [1]. Also, the number of each class are equally distributed as shown in figure 1.

There are many benefits for using this dataset. Firstly, the input data has already been pre-processed, so that no further efforts required for data processing. This not only save time on processing data but also ensures that every research that using this dataset always have exactly the same input, which can remove unnecessary variance when comparing result

with different other researches. Secondly, each class of this dataset have similar number of instances which can reduce bias when training the network [5]. Finally, a non-technical reason is that the topic of this dataset, written digit recognition is a very common and useful application of neural network, which has positive influence of solving real world problems.

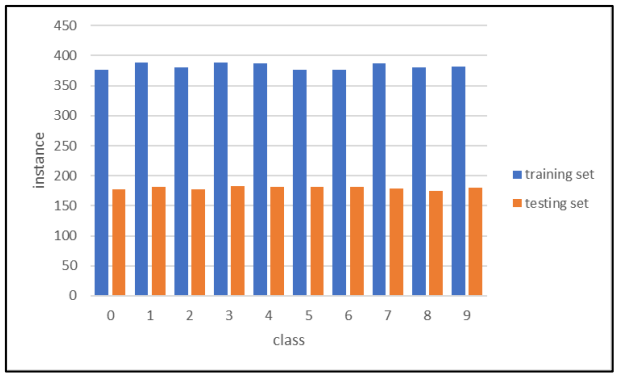

Fig. 1. Class distribution of the dataset

To make it easy for the CNN model to load data, the original bitmap data has been reformatted. Originally, the bitmap is stored as a 32 by 32 matrix, followed by an extra row of the label. The new format flattens the matrix and make it only one row with 1025 elements (1024 pixel and 1 label). This only change the format of the data storage but do not apply any processing of the data.

1.3 Objectives

The objectives of this research are:

- Investigate whether network pruning technique works on hidden layer of simple two-layer neural network and fully connected layer of convolutional neural network. This need to show that whether pruning technique can actually remove certain number of hidden neurons without affect the accuracy of the network.
- Investigate the aspects that may affect the performance pruning and improve it.

2 Method

To achieve the goal of this research, a simple two-layer neural network and a convolutional neural network are designed and implemented with network pruning technique. Also, some experiments are designed to investigate the aspects that affect the performance of pruning technique.

2.1 Model Design and Implementation

The configurations of the models below are based on the assumption that these parameter values are sufficient for the chosen dataset, and some preliminary test shows an actually positive performance with these configurations.

The simple two-layer neural network model accepts 64 inputs and contains one hidden layer with 50 neurons as default and one output layer with 10 neurons as there are totally 10 classes.

The convolutional neural network model contains two convolutional layers with max pooling and two fully connected layer. The first convolutional layer has 5 filters and the second layer has 10 filters. All filter has a kernel size of 5 and stride of 1. All max pooling layers have kernel size of 2 and stride of 2. The first convolutional layer is similar to the hidden layer of two-layer network and has 200 neurons in default. The second convolutional layer is similar to the output layer of two-layer network which has 10 neurons. Since the pruning technique calculate output activation of hidden neurons, dropout is not implemented in this CNN model to avoid unnecessary effect.

Both models have a learning rate of 0.01. The models are trained using batch training method and initially the batch size is the same as instance number of training set. That is, for each epoch, all the data in training set is feed into the model and then perform back-propagation. The number of epoch for training is 200.

2.2 Pruning Technique Implementation

The core of pruning technique is the calculation of angle between output activation vectors. When perform pruning, the network model will firstly be trained, and then all the data from training set will be fed into the network and return the output activation vectors of target layer.

Then the angle between any two of the vectors will be calculated iteratively and stored in a matrix. To simplify the algorithm, in this implementation, the hidden neurons are not actually removed. Instead, all the weight of that neuron will be set to 0. With the weight set to 0, the hidden neuron will lose functionality, and thus behave the same as removed.

If the angle is less than 15 degree, then all the weight values of second hidden neuron will be added to the first hidden neuron and the weights of second neuron will be set to 0. On the contrary, if the angle is greater than 165 degree, then both hidden neurons will have their weights set to 0.

2.3 Validation and Experiments

A 10-fold cross validation is used to test the performance of the models and pruning technique. Some experiments are also designed to investigate the aspects that may affect the performance of the pruning technique. Based on the results of investigation, the CNN model has been improved and another 10-fold cross validation has been conducted to test the performance of new model and pruning technique.

10-fold cross validation

The validation method used in this research is 10-fold cross validation. Both models run the validation twice, the first time without pruning and second time with pruning, as shown in Table 1. For each validation, the result contains an overall accuracy, a matrix of class accuracy and a confusion matrix. With these result, not only the overall performance but also the accuracy of each class will be examined to ensure the model is robust.

Table 1. 10-fold cross validation design

Validation	Model	Pruning
1	two-layer	No
2	CNN	No
3	two-layer	Yes
4	CNN	Yes

Experiments

The result of CNN with pruning is negative, which will be demonstrated in section 3.1. To investigate the possible reasons that lead to a low performance, some experiments are designed based on some hypotheses made in section 3.1. Basically, the experiments focus on two aspects, namely the training process and network size. Specifically, the training process focus on the number of epoch and batch size while the network size focus on number of hidden neurons. There are totally 3 experiments according to the three hypotheses introduced in section 3.1 as shown in Table 2. All the experiment uses 5-fold cross validation which can show a general result and reduce the computational load compare to 10-fold.

Table 2. Parameters and configurations for each experiment

experiment	model	number of epoch	number of hidden neuron	batch size
1	two-layer	200	10, 20, …, 100, 200, …, 1000, 1100, …, 2000	all
2	two-layer	1500	10, 20, …, 100, 200, …, 1000, 1100, …, 2000	all
	CNN	10, 20, …, 100, 200, …, 1000	200	
3	two-layer	100	10, 20, …, 100, 200, …, 1000, 1100, …, 2000	10
	CNN	20		

Experiment 1 uses two-layer net. It changes the number of hidden neurons and keeps the number of epoch unchanged. The batch size value ‘all’ means using all the data as 1 batch as introduced in section 2.1. This experiment aims to investigate whether the two-layer neural network can have same good performance with bigger size.

Experiment 2 aims to investigate whether more epoch number will improve the performance of models with pruning. Different variable configurations have been applied on two-layer net and CNN. For two-layer net, an epoch number of 1500 is applied over a range of hidden neuron numbers. For CNN, the number of hidden neuron is fixed to 200, while the number of epoch varies.

Experiment 3 investigate the effect of batch size. The batch size is set to be 10, which is a very small number compare to previous experiments. The epoch number of both models are also set to be a relatively small number, namely 100 for two-layer net and 20 for CNN.

3 Results and Discussion

3.1 Preliminary Cross validation result and analysis

Figure 2 and 3 show the preliminary result of 10-fold cross validation. The x-axis represents each fold of the validation, while the y-axis is the overall accuracy. The two-layer network has almost the same accuracy with and without pruning, which is about 97.5% on average. Also, each fold has almost the same accuracy, which means the performance is stable and robust. The original CNN has a stable and higher accuracy than two-layer network, however, the CNN with pruning has an unpredictable performance. The highest accuracy is about 92%, while the lowest is 28%, which means the model's performance is very unstable.

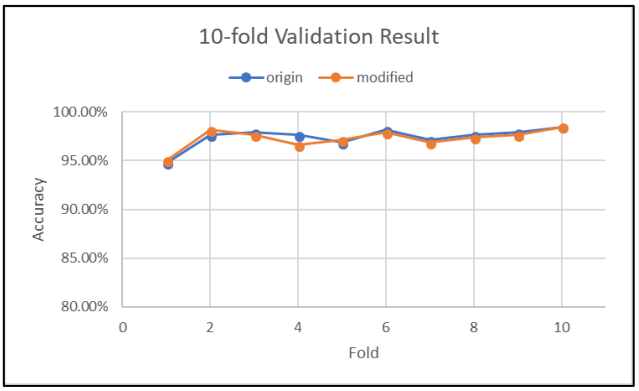

Fig. 2. 10-fold Cross Validation Result of two-layer network

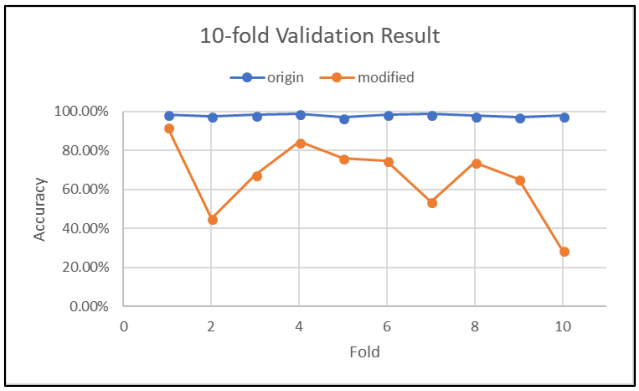

Fig. 3. 10-fold Cross Validation Result of CNN

The validation result shows a significant difference between the performance of two-layer network and CNN, even though they have very similar structure, on which the pruning applied. The only difference between the two models is the network size. The two-layer network has 64 input and 50 hidden neurons, while the CNN's fully connected layer has 200 hidden neurons and 250 input, which is determined by the kernel size and stride of filters.

Since the pruning focus on the output activation of hidden neurons, which has a very close relation to the training process, particularly the number of epoch and batch size, it is possible that the CNN does not learn sufficient information originally. Based on this, three hypotheses are suggested:

1. The two-layer network with pruning may also perform unpredictably if have much bigger size.
2. The models with pruning will have a better performance with more training epoch.
3. The models with pruning will have a better performance with smaller batch size.

To prove the hypotheses, three experiments are designed as introduced in section 2.3, to further investigate the aspects that affect the performance of network with pruning.

3.2 Experiments result and analysis

Before discuss experiment result, a concept, removal rate, should be introduced. The removal rate of pruning technique can be expressed using equation (1).

$$removal\ rate = \frac{removed\ neurons\ number}{initial\ neurons\ number} \times 100\% \quad (1)$$

Basically, removal rate indicates how many hidden neurons have been removed. This can be used as a measurement of the pruning performance. If a pruning has a high removal rate and the network remains high accuracy, then it has a positive performance, because it removed as many redundant neurons as possible without losing functionality.

Experiment 1

Figure 4 shows the accuracy of two-layer net with epoch number of 200 under different hidden neuron numbers. The x-axis is the number of hidden neuron while the y-axis is the accuracy. The blue boxes represent origin network, i.e. network without pruning, while the orange boxes represent 'modified' network i.e. network with pruning. Figure 5 shows the removal rate of two-layer net. The x-axis is the epoch number while the y-axis is the removal rate.

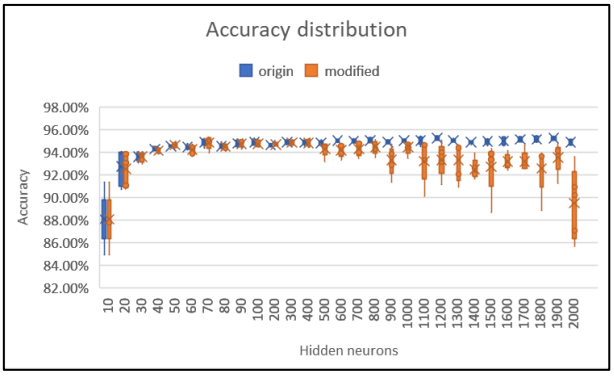

Fig. 4. Accuracy distribution of Test 1 (CNN)

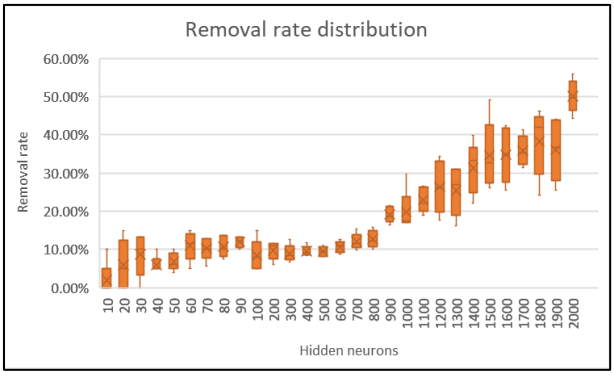

Fig. 5. Accuracy distribution of Test 2 (two-layer net)

It is clear that with the increasing of hidden neuron numbers, the accuracy increases, until hidden neuron reaches 50. The accuracy of origin network then keeps at a stable state with the further increasing of hidden neurons. However, the accuracy of 'modified' network decreases after the hidden neuron researches 500. The accuracy of 'removed' network also become unstable and discrete when have more than 500 hidden neurons.
Figure 5 shows the removal rate of hidden neurons. As can be seen in the figure, from 10 to 50 hidden neurons, the removal rate increases, and then from 50 to 500 neurons, the removal rate reaches a stable state and remains at about 10%. However, after 500 neurons, the removal rate suddenly increases rapidly, and reaches 40% at 1700 neurons, and still have a trend of increasing.

This shows that when having more than 500 hidden neurons, the pruning technique removes too many hidden neurons, as a result, the network loses information so that the accuracy decreased. This experiment proved hypothesis 1 that with very big number of hidden neurons, the performance of two-layer network also become unpredictable. This proved that the pruning technique actually have same performance on both two-layer network and CNN.

Experiment 2

Experiment 2 aim to investigate the effect of epoch number. Figure 6 and 7 show the accuracy and removal rate of two-layer network with 1500 epochs under different number of hidden neurons. Compare to experiment 1, experiment 2 has a much bigger number of epoch. However, the result shows same trend as experiment 1 that when having too many hidden neurons, the pruning technique removes too many hidden neurons so that the accuracy of network decreased and become unstable. This means, the increasing of epoch number does not increase the performance of two-layer network with pruning.

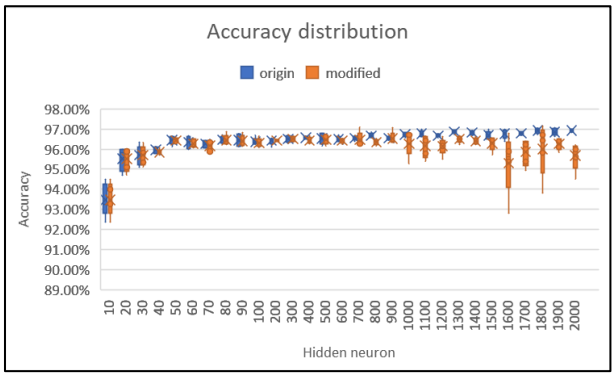

Fig. 6. Accuracy distribution of two-layer net

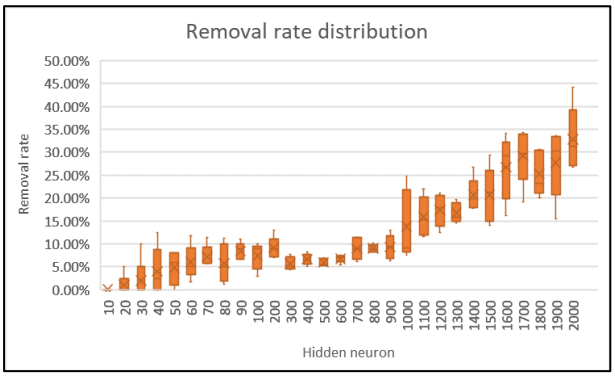

Fig. 7. Removal rate distribution of two-layer net

Figure 8 and 9 show the accuracy and removal rate of CNN with 200 hidden neurons and different epoch number. With very little epoch, the pruning has very high removal rate because the weights are still very similar to the initial random numbers, but the removal rate remains at about 20% with more epoch. However, the accuracy of CNN with pruning still has an unpredictable performance, and the accuracy is much lower than original CNN.

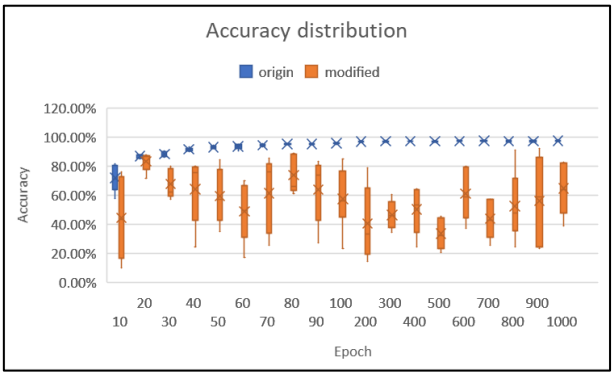

Fig. 8. Accuracy distribution of CNN

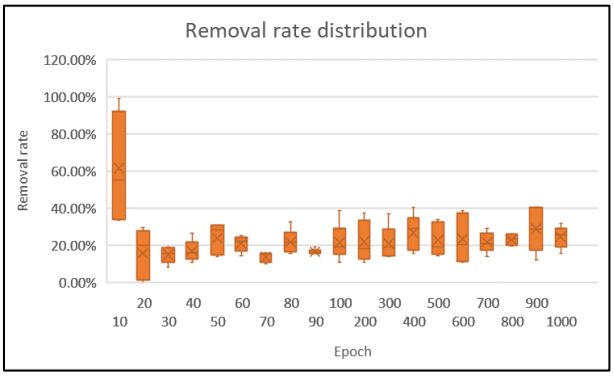

Fig. 9. Removal rate distribution of CNN

The result of two-layer network and CNN shows that even if the network model have a very large number of epoch, it still has a low performance if the number of hidden neuron is big. This also disprove the hypothesis 2.

Experiment 3

Experiment 3 changed batch size for both two-layer network and CNN. The batch size is set to be 10, which is a very small number compare to previous experiments. The epoch number of both models are also set to be a relatively small number, namely 100 for two-layer net and 20 for CNN.

Figure 10 and 11 shows accuracy and removal rate of two-layer network with small batch size. As can be seen in the figure, the removal rate tends to be stable with the increasing of hidden neuron number, and the accuracy of network with pruning remains at a high value, which is almost the same as original network.

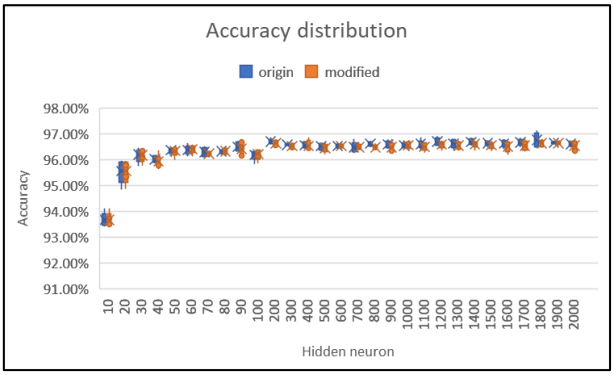

Fig. 10. Accuracy distribution of two-layer net

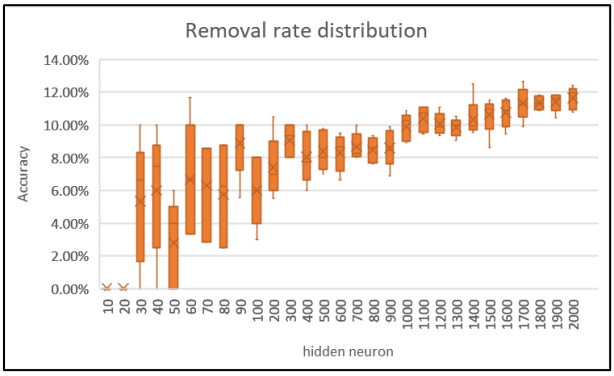

Fig. 11. Removal rate distribution of two-layer net

Figure 12 and 13 shows accuracy and removal rate of CNN with small batch size. As can be seen in the figure, the removal rate tends to be stable with the increasing of hidden neuron number, and the accuracy of network with pruning remains at a high value, which is almost the same as original network.

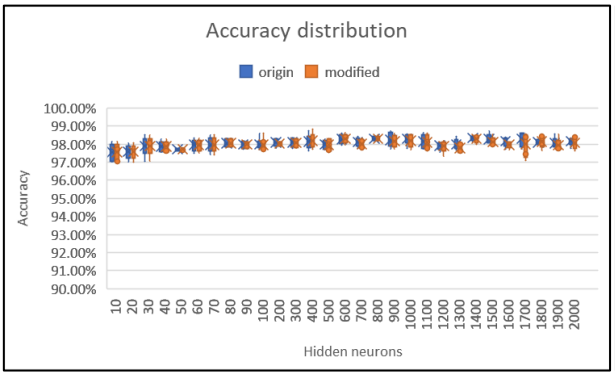

Fig. 12. Accuracy distribution of CNN

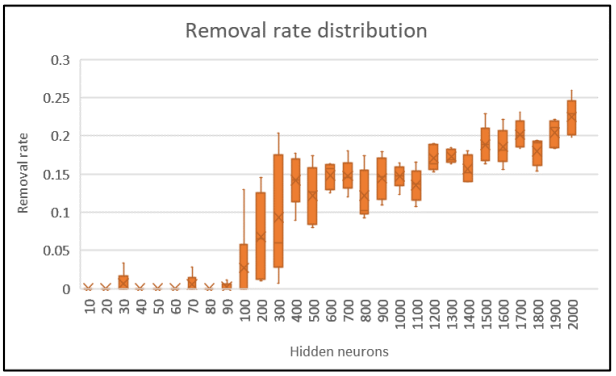

Fig. 13. Removal rate distribution of CNN

The result of experiment 3 shows that both two-layer network and CNN can have a very stable performance and high accuracy if trained using a small batch size. This proves the hypothesis 3.

In conclusion, two of the three hypotheses are proved through the experiments. The network with pruning can have same performance as original network, only if it is trained using small batch size. Although the number of epoch also related to learning process, it has no significant effect on network with pruning.

3.3 Validation result of improved CNN model and comparison to published paper

As introduced in section 2.3, the CNN model is improved based on the result of experiments. The new CNN model is tested using 10-fold cross validation.

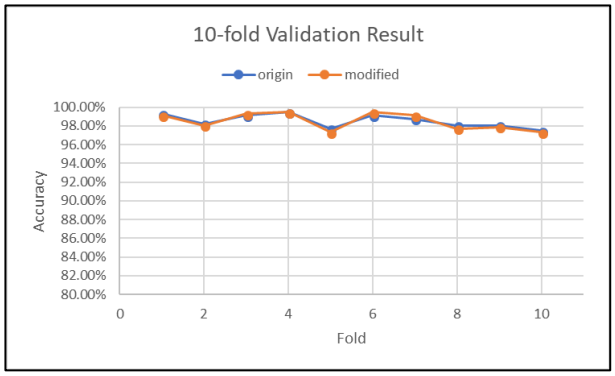

Fig. 14. 10-fold Cross Validation Result of CNN

class	0	1	2	3	4	5	6	7	8	9	accuracy
0	52	0	0	0	0	0	0	0	0	0	100.00%
1	0	56	0	0	0	0	0	0	0	0	100.00%
2	0	0	60	1	0	0	0	0	0	0	98.36%
3	0	0	0	56	0	0	0	1	1	0	96.55%
4	0	0	0	0	65	0	0	0	0	1	98.48%
5	0	0	0	0	0	57	0	0	0	0	100.00%
6	0	0	0	0	0	0	50	0	0	0	100.00%
7	0	0	0	0	0	0	0	55	0	0	100.00%
8	0	0	0	0	0	0	0	0	55	1	98.21%
9	0	0	0	0	0	0	0	0	0	51	100.00%

Fig. 15. One Confusion matrix of CNN with pruning

Figure 14 shows the validation result of improved CNN model. As can be seen, the model with pruning has almost the same accuracy as the original one. Both model has a very stable performance and high accuracy, which is about 98.5% on average.

Figure 15 shows one confusion matrix of CNN with pruning. The accuracy in the figure is the accuracy of each class. It is clear that all the class have very high accuracy with a maximum of 100% and a minimum of 96.5%.

In paper *Pareto-optimality of oblique decision trees from evolutionary algorithms*, decision trees are used to classify the same optical-digits dataset [6]. The best accuracy of that paper is 88.69%. Compare to this published paper, the CNN model with pruning implemented in this research has a relatively higher performance, which shows that CNN is possibly more capable for image classification.

4 Conclusion and Future Work

This research shows that the network pruning technique can effectively reduce network size without losing accuracy. It works well on fully connected layer of both simple neural network and convolutional neural network. The network models with hidden neurons removed have same performance of the original network. A comparison between the result of this research and the result of a published paper shows that CNN with pruning technique has a much better performance of image classification on Optical-digits dataset than decision trees.

According to the result of further investigation, batch size has a significant effect on network model with pruning. Some conclusions are:

- When trained with a big batch size, the network model can have a high accuracy and stable performance regardless of the number of hidden neurons.
- Apply pruning technique on a network model which has been trained using a big batch size will result in an unpredictable network performance.
- If the network model is trained using a small batch size, it will remain high accuracy and stable performance after pruning applied.
- The number of epoch have limit effect on network performance. The network models require a certain number of epoch, but when have more epochs, the performance will not be further improved.

There are still some works of this research needs to be done in the future.

First of all, many parameters, like learning rate, number of filters etc. of the network models in this research is configured based on assumption. Some further research could be conducted to investigate the most appropriate configuration.

Secondly, this research only shows that the batch size affects the performance of network model with pruning. However, the relation between batch size and network performance remains unknown. Also, it is not clear that why batch size only affect network with pruning, but not the original network.

Finally, from the result of this research, it is obviously that the network may still have too many hidden neurons after pruning. For example, the CNN model in this research can work well with less than 100 hidden neurons, however, if initialize to 2000 hidden neurons, the pruning technique can only remove about 200 hidden neurons and left 1800, which is still too many compare to 100. Therefore, it is important to investigate the method to reduce the hidden neuron to its minimum.

References

[1] E. Alpaydin and C. Kaynak, Department of Computer Engineering, [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits. [Accessed 4 2018].

[2] Torch Contributors, "PyTorch documentation," 2018. [Online]. Available: http://pytorch.org/docs/stable/index.html. [Accessed 4 2018].

[3] T. Gedeon and D. Harries, "Network Reduction Techniques," *Proceedings International Conference on Neural Networks Methodologies and Applications,* vol. 1, pp. 119-126, 1991.

[4] A. Sajid , H. Kyuyeon and S. Wonyong, "Structured Pruning of Deep Convolutional Neural Networks," *ACM Journal on Emerging Technologies in Computing Systems,* vol. 13, no. 3, 2017.

[5] A. M. Maciej, A. H. Piotr, M. Z. Jacek, Y. L. Joseph, A. B. Jay and D. T. Georgia, "Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance," *Neural Networks,* vol. 21, no. 2, pp. 427-436, 2008.

[6] J. M. Pangilinan and G. K. Janssens, "Pareto-optimality of oblique decision trees from evolutionary algorithms," *Journal of Global Optimization,* vol. 51, no. 2, pp. 301-311, 01 10 2011.

[7] L. Xu, A. Krzyzak, Y. Ching and Suen, "Methods of Combing Multiple Classifiers and Their Applications to Handwriting Recognition," *IEEE Transactions on Systems,* vol. 22, no. 3, 1992.