An Implement of Network Reduction Technique Using Distinctiveness of Hidden Neurons

Shaohua Zuo

Research School of Computer Science, Australian National University
u6074583@anu.edu.au

Abstract. A network reduction technique, which uses the distinctiveness to calculate input pattern space vector of hidden units, for the artificial neural network is implemented in this reporting. In a trained artificial neural network model, many hidden units are not necessary to have a good prediction precision. More hidden units will cost more time and space in training and predicting process, which is not favourable. This report is building a model of neural network to solve binary classification problem that whether a student can pass the course on a data set from the real world. The network reduction technique using distinctiveness can effectively remove undesirable hidden units without reducing the predictive precision of the model. The reduction technique is especially good for complex model trained with a large data set.

Keywords: Artificial Neural Network, ANN, Conventional Neural Network, CNN, Hidden Units, Distinctiveness, Network Reduction, Pruning

1 Instruction

There has been lots of research on neuron pruning techniques (i.e. network reduction techniques) of the artificial neural network that uses feed-forward algorithm to get output and back-propagation algorithms to train models. To do the prediction for complex problems, more hidden units may be required depending on the complexity of the problem. To train a network with a good precision of prediction, more hidden units are preferred when been initiated. However, many hidden units in a network could be unnecessary with these units significantly adding computational cost to the training process, especially when we need to train a model for classification on a large data set which has a large number of samples and each sample has many features. Thus, how to detect the redundant units and remove them from the network is the main problem to be solved in this reporting. Distinctiveness could be the property of units to be evaluated to determine which should be removed from the network.

The aim of this report is to do a verification of feasibility for the network reduction techniques suggested in the published conference paper [3]. Firstly, a simple neural network based on PyTorch and an automatically detecting method of undesirable units based on the distinctiveness property presented by the vector angles between each pair of hidden neurons have been implemented. The data set used to train the network is derived from real world, which was used by the research paper [2] and contains 649 samples for Portuguese course and 395 samples for Mathematics course with 33 dimensions including the target, are used to form a binary classification network to predict whether a student can pass the course. And secondly, the same network reduction technique is applied to a more complicated conventional neural network which is trained by the famous MNIST dataset to recognise digits the from handwritten digit images. The MNIST dataset have 60000 samples for the training dataset and 10000 for the testing set.

This report will firstly describe the methods used to detect undesirable units, and then discuss the effect of removing hidden units marked on the above detecting method. After that, this report will analyse the result of the network reduction to obtain some recommendations on the application of the technique.

2 Detecting Undesirable Hidden Units

2.1 Theoretical Basis

According to [3], there are several properties of hidden units, such as relevance, contribution, sensitivity, badness and distinctiveness, that could be used to analyse which units to be removed. The former 4 properties mentioned above are the analysis of the relation between hidden units and the whole network to determine whether each unit is important to the current network. But instead, the final one, distinctiveness, are measuring the similarity and constancy of every pair

of hidden units. If there are two units similar, or complementary to each other, or that produce together to a constant output, those constant or complementary pairs of unit should be all removed, and one of each similar pair of unit should be removed. Because the similar pair of unit tends to produce similar or even same output for the same input pattern, one of them could be removed without impact the output of the network. And similarly, the complementary pair of unit tends to together produce zero-sum output which does not affect the output of the network, thus all of them could be removed.

2.2 The Training and Evaluating of The Simple Network

2.2.1 The Simple Neural Network

The simple neural network is a two layers network that has only 1 hidden layer and 1 output layer. The amount of units of the hidden layer is initiated large, where there are 50 hidden units while the inputs are just 32.
The data set is divided according to the courses into two parts, which are Mathematics and Portuguese. The two courses may have different learning pattern thus should be modelled separately. The binary classification problem is, by input the student features like sex, age, school, address type and so on (the first 32 dimensions of the data sets) to predict the value of G3, which is the target column and in the data set are reset to 0 if less than 10 (i.e. fail the course), or 1 if greater than 10 (i.e. pass the course). All the binary values of yes or no in any column are reset to 1 or 0. And all nominal values are aligned with arithmetic progression as 0, 1, … , k, where k is the number of the nominal values. [2]

2.2.2 The Evaluating Method

The published research paper [2] illustrated the classification precision results derived by using 10-folds cross validation method on the data set. Therefore, I also implemented a n-folds cross validation method for the neural network and set n to 10.
N-folds cross validation evaluation means that to fully use the data set while the data volume is limited, divided data set into n equal size groups, iteratively choose one of them as the testing data set and choose the rest as the training data set and use the two data sets to train and test the model, finally calculate the overall accuracy.

2.2.3 The Model Performance Before Pruning

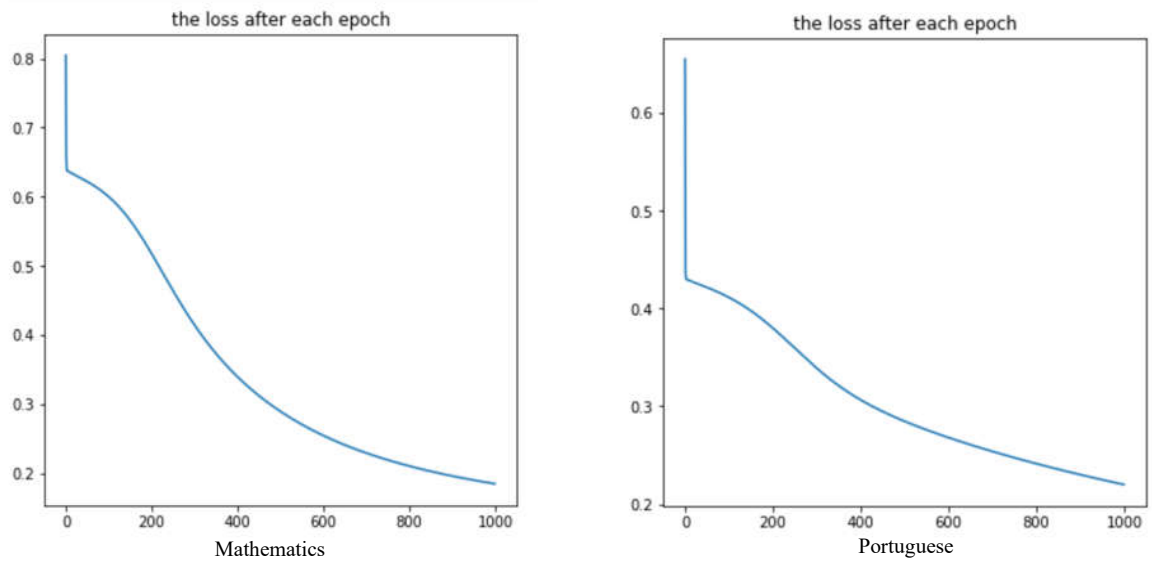

Fig. 1. The loss in the training process on the different data sets

In Fig. 1, the left diagram is that the loss changes along the training process on the Mathematics data set, and the right diagram is on the Portuguese, where the x-axis is the number of epoch while the y-axis is the total loss over all the training data. As shown in Fig. 1, the cross-entropy loss of the network over the two data sets are reduced in the training process, which proves that the training process has built an effective model. As the Mathematics data set (whose size is 395 × 33) has fewer samples than the Portuguese (whose size is 649 × 33), the final loss of the network of Portuguese is more than that of Mathematics.

Table 1. The prediction accuracy comparation between my model and the results from the research paper

	Mathematics (%)	Portuguese (%)
1	92.31	81.25
2	85.00	93.85
3	94.87	90.77
4	80.00	83.08
5	89.74	92.31
6	82.50	87.69
7	94.87	86.15
8	92.50	87.69
9	87.18	89.23
10	92.50	90.77
Overall	89.15	88.28
The accuracy of classification in the research paper	88.3±0.7	90.7±0.5

Table 1 shows the predict precision of 10-folds cross validation evaluation and the results published in the research paper. The model I built has obtained about 89.15% prediction accuracy on the Mathematics data set while the result on the same data set published in the research paper is about 88.3%. And the accuracy of my model on the Portuguese data set is about 88.28% while the result published is about 90.7%. The results derived from my model are sufficiently close to the result published, which is good enough to be used as the baseline to apply the network reduction technique.

2.3 The Network Pruning Method of The Simple Network

For a two layers network, for every input pattern, each hidden unit produce an output to be the input of the output layer. And then the output of every hidden unit for each pattern forms a vector, which represents the effect of the hidden unit in the input pattern space. As the hidden unit will produce the output after a sigmoid function, which takes (0.5, 0.5) as the centre of the space rather than (0, 0), the vectors need to remove the bias by minus (0.5, 0.5).
Then we can use these input pattern space vectors to calculate the vector angles to measure the similarity and constancy of unit pairs:

$$Angle(\boldsymbol{v}_1, \boldsymbol{v}_2) = \arg cos \frac{\boldsymbol{v}_1 \boldsymbol{v}_2}{|\boldsymbol{v}_1||\boldsymbol{v}_2|} \quad (1)$$

Where the $\boldsymbol{v}_1, \boldsymbol{v}_2$ are the two vectors to calculate the angle. After this calculation, the angle should be in (0, 360), which need to be normalised into (0, 180) by:

$$Angle(\boldsymbol{v}_1, \boldsymbol{v}_2) = \begin{cases} Angle(\boldsymbol{v}_1, \boldsymbol{v}_2) & , \quad Otherwise \\ 360 - Angle(\boldsymbol{v}_1, \boldsymbol{v}_2) & , \quad Angle(\boldsymbol{v}_1, \boldsymbol{v}_2) > 180 \end{cases} \quad (2)$$

Because the situation of two vectors being perfectly identical or complementary are rare, we may assume that if two vectors have the angle less than 15° between them, we deem them to be similar. On the contrast, if two vectors have the angle more than 165° between them, we deem them to be complementary.
For each pair of the hidden units, we calculate the angle and determine whether they are similar or complementary. The pairs of similarity would be recorded in a list and the pairs of constancy in another list. With the two list, we can determine which are the undesirable units, in other words, the units should be removed.

2.4 The Training and Evaluating of The Complicated CNN

2.4.1 The CNN

The CNN consists of two conventional layers and a full connection layer. The conventional layers are used to extract abstract features from the images and the lower level features, the full connection layer is used to compute the probabilities that the sample belong to each class. As the model is complicated and the training dataset is very large, the CNN is trained in a mini-batch method, and the batch size is 50.

2.4.2 The Evaluating Method

There is an official test dataset provided for precision testing of the model. The testing dataset, which has 10000 samples in it, is large enough to obtain a convincing precision testing result.

2.4.3 The Model Performance

Because the training is processed in the mini-batch method, and the batch size is set to 50, the training process is divided into 1200 steps. The training result is shown as:

Table 2. The performance of my CNN model along training process

STEP	LOSS	TEST ACCURACY (%)
1	2.2941	19.98
100	0.2333	91.37
200	0.2419	93.61
300	0.0629	95.92
400	0.2469	96.84
500	0.1124	96.70
600	0.0354	97.52
700	0.1091	97.51
800	0.1150	97.18
900	0.1809	98.14
1000	0.0442	97.77
1100	0.1027	98.20
1200	0.0849	98.25

In Table 2, the step column is the number of steps having been take, the loss column is the total loss over the batch of training data in this step, and the test accuracy column is the test accuracy of my CNN model after trained in so many steps, which is evaluated by the testing dataset thus can be seen as the predicting accuracy. Finally, the predicting accuracy of my CNN is 98.25%, which is good enough. The newest research [1] shows the best predicting accuracy is about 99.77%. And the result of my CNN is really close to it. The CNN model trained above is an effective recognition model to classify handwritten digits.

2.4.4 The pruning of the CNN

The pruning of the CNN happens in the full connection layer, where the 1568 features extracted by the conventional layers are used to compute the probabilities that a sample belongs to each class. The CNN uses ReLU activation function which return exactly the value of the input for the input more than 0, and 0 other wise. Therefore, the centre of the output should be (0,0). The same vector angle analysis is performed and this time we get 1568 input pattern space vectors for all hidden units in this layer.

3 Network Reduction Results and Discussion

3.1 The Result of The Simple Neural Network

3.1.1 The Impact of Removing Hidden Units

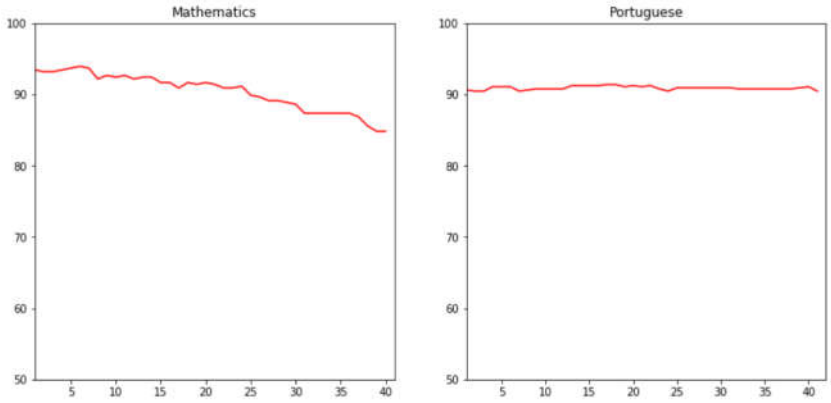

Fig. 2. The impact on model precision of removing hidden units

The Fig. 2 shows that the precision of my model is changed with the removing of hidden units. In the figure, the left diagram shows the precision changes on the Mathematics data set and the right diagram show that on the Portuguese data set. The x-axis of each diagram is the number of hidden units that have been removed from the current model, and the y-axis is the precision. From the figure above, we can see that while the model on the Portuguese data set is not affected significantly by the pruning, the precision of the model on the Mathematics data set is decreasing along the reduction process.

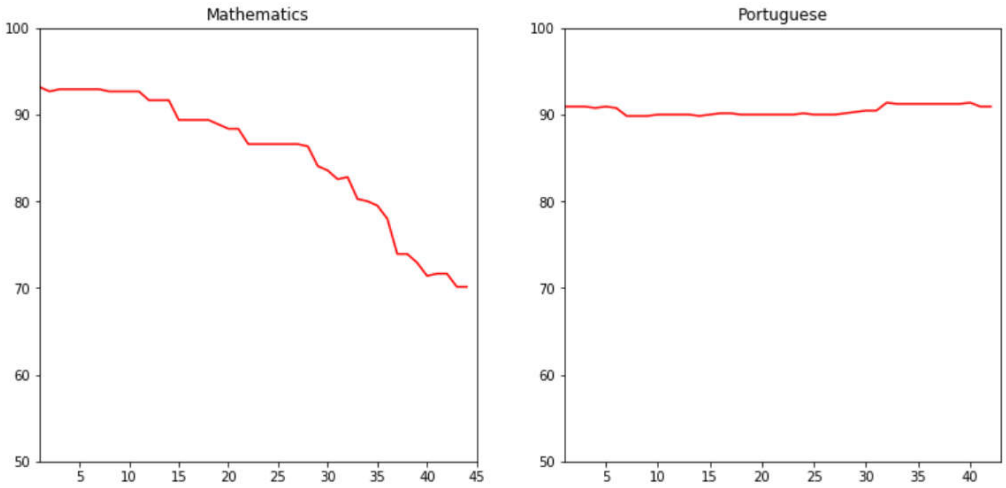

Fig. 3. The worst result of the pruning

After I did more times of experiment I found the best result is that both of the two models are not affected by the pruning, but that is rare in many runs. And the worst result is as shown in Fig. 3, while the precision on the Portuguese model is not impacted, it on the Mathematics model is decreasing significantly by more than 20%.

3.1.2 The Result Analysis

Although in the theory mentioned above, the removing of undesirable units will not affect the precision of the model, the situation of applying this to real data may be more complicated. The first cause of this should be that the number of samples in the Mathematics data set is just about half of the number of sample in the Portuguese data set, which leads the weights and biases of the hidden units to be not very precise to solve the classification problem. Therefore, the input pattern space vectors of the hidden units could be not precise to analyse. Moreover, we have made a less strict assumption. When calculating the similar pairs and complementary pairs of unit, we assume that if the two units have the input pattern space vector angle within 15°, they are similar to each other and if they have the angle more than 165°, they are complementary to each other. This assumption may lead us to mistakenly believe more unit pairs to be similar than it actually is.

3.2 The Result of The Complicated CNN

From the vector angle analysis I found that there are only similar pair of unit in the CNN but no constancy pair, which may be because the output of ReLU cannot be opposite to others as they are all positive.

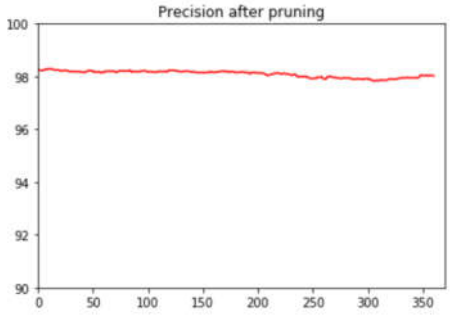

Fig. 4. The precision of my CNN model after pruning several undesirable units

Fig. 4 shows that the predicting precision would not be reduced by the pruning. The x axis is the number of the hidden units that pruned by the network reduction technique, and the y axis is the precision of the model after pruning that number of undesirable units. The predicting precision before pruning is 98.25% and turns to 98.02% after pruning all undesirable units detected above. It can be seen from the figure that the predicting precision is approximately flat, which means it would not be impacted in the pruning. The input pattern space vectors are derived in the same way as in the simple neural network. The difference is the complicated CNN model is trained with a very large dataset which contains 60000 samples while the simple neural network is just trained with a dataset with not more than 1000 samples. The result shows that the more complicated the model and the more the data used to train the model, the less the precision of the model reduced by the pruning.

4 Conclusion and Future Work

For a simple neural network trained with few of data, the network reduction technique has potential risks that the predicting precision might be significantly reduced. But on the other hand, if the network model is simple, there are no necessity to apply such network reduction technique. The network reduction technique is mainly used to improve the calculating speed when processing new data. The computation of simple models will not take much time thus the network reduction is not necessary. For a complicated model trained with a large dataset, the network reduction technique will not reduce the precision, therefore the technique is good to apply to it.

As the conclusion, although the pruning of undesirable units may have the potential risk of decreasing the precision of model, the network reduction technique using distinctiveness is practicable to apply to the real-world classification problems. The network reduction technique is only necessary when the network is complicated. However, more research is needed in order to figure out when should the model after pruning be restrained, how to use other properties like relevance, contribution and so on to do the network reduction, and which of the different techniques should be applied in the training work.

5 References

[1] Cireşan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column Deep Neural Networks for Image Classification. *Computer Vision and Pattern Recognition* (arXiv:1202.2745).

[2] Cortez, P., & Silva, A. (2008). Using Data Mining to Predict Secondary School Student Performance. *Proceedings of 5th Future Business Technology Conference*, (pp. 5-12). Porto.

[3] Gedeon, T. D., & Harris, D. (1991). Network Reduction Techniques. In *Proceedings International Conference on Neural Networks Methodologies and Applications* (Vol. 1, pp. 119-126).

[4] Gedeon, T. D. (1995). Indicators of Hidden Neuron Functionality: The Weight Matrix versus Neuron Behaviour. *Artificial Neural Networks and Expert Systems IN Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*. (pp. 20-23).

[5] Hagiwara, M. (1990). Novel back propagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection. IJCNN. (vol. 1, pp. 625-630).