Mark Prediction with Back-Propagation Neural Network Using Mask Pruning to Reduce Network Weights

Zhijie Huang[1]

Research School of Computer Science
Australian National University
Canberra ACT 0200
u6151827@anu.edu.au

Abstract. In the university, the final assessment is a kind of way to evaluate the academic performance. Also, we use every assignment mark to predict the final results about the course through a back-propagation neural network. For this prediction, we adopt the dataset about the marks of every assessed task in COMP1111 of the University of New south Wales. Because we do not know which features are the principal components for the final assessment, we use all those data to train the network. The results provided by the model is considered to be accurate in the case of prediction below five points and as a kind of warming for poor performance students. For this experiment, we use back propagation neural network to predict the students' final mark and use mask to prune the weights which are close to zero to improve the efficiency. As for the experimental gain, we found that the sample size is not enough to build a high-accuracy shallow network, conclusion the pruning rate that can still keep closed accuracy than the original network and discover that the accuracy may increase prune partial weights.

Keywords: Mark prediction · Back-propagation neural network · Mask pruning.

1 Introduction

The reason adopting this mark dataset is that we expect to know the model perform and it is meaningful for students to obtain more suggestions from prediction results. Specifically, the dataset about undergraduate students in Computer Science College of University of New South Wales who enroll the COMP1111 has 153 raw samples which record their achievement performance[4]. As shown in Figure 1. Since the dataset has string, integer and missing values, we should do preprocessing such as removing empty data and converting data type, before training those data.

```
(154, 15)
```

Regno	Crse/Prog	S	ES	Tutgroup	lab2	tutass	lab4	h1	h2	lab7	p1	f1	mid	lab10	final
FullMark	-	-	-	-	3	5	3	20	20	3	20	20	45	3	100
0168826	3971	2	FS	T1-yh	2	3	2.5	19.5	.	2.5	11	.	33	2.5	71
0168883	3971	2	F	T9-ko	3	5	2	20	17	.	8	.	27	2.4	67
0168907	3400	1	F	T6-no	3	3	3	10	.	2	.	.	9.5	2.4	30
0169379	3970/1061	2	F	T3-ku	2	3	2	20	19.5	2	15.5	.	21	.	62

Fig. 1. Partial Raw Data

Before choosing a new technique that can perform better than the previous one, we research this new technique in the Google Scholar. we find some useful essays that provide us more backgrounds of new techniques. Among those new techniques, we adopt the mask reduction technique to reduce the network which can remove the unimportant weights rather than units in the shallow back-propagation neural network, and we will compare this technique to the previous one.

This research focuses on predicting the total grade of those students and compare the new technique to the distinctiveness of hidden units. Firstly, we do data preprocessing, such as deleting empty samples and transforming string-type data[5]. During the training, we reconstruct a three-layer shallow mask neural network to fit the training data and calculate the corresponding mask values in the network. Moreover, we decide the mask pruning proportion and use those mask values to change the weights to zero[1]. Through the continuously experiment, we can gain the well performed pruned network. Finally, this experiment can compare the performance with the previous results and verify the better technique.

2 Method

In this section, we state the details of mask pruning to improve the efficiency of mark prediction. In the Initial step, we process the dataset and how to rebuild a neural network to calculate the mask values. During the latter step, we present that we use those values to prune the network and evaluate the performance.

2.1 Data Preprocessing

In the stage of data preprocessing, we need to wash the data firstly before training themwhich enables model to be stronger[7]. Initially, by observing the original data, we should solve the problems that missing values, data-type transformation, deleting empty data, and dividing the data set. Concretely, the neural network needs to input the float data and input cannot empty, so we change the null value to zero and wash the dataset, such as converting data type, which ensures input to enter the network[9]. Especially, there are different ways to solve the missing values and empty samples. Missing values are partial null values but empty samples mean most of data in the sample are empty. Furthermore, in order to mine more information behind the data, we use all the features except the attribute 'Reno'. After completing these processing operations, we save the processed data to a csv-format file, shown as Figure 2.

	lab2	tutass	lab4	h1	h2	lab7	p1	f1	mid	lab10	...	Tutgroup_T1-yh	Tutgroup_T10-yh	Tutgroup_T2-no	Tutgroup_T3-ku	Tutgroup_T4-ko
1	2.0	3.0	2.5	19.5	0.0	2.5	11.0	0.0	33.0	2.5	...	1	0	0	0	0
2	3.0	5.0	2.0	20.0	17.0	0.0	8.0	0.0	27.0	2.4	...	0	0	0	0	0
3	3.0	3.0	3.0	10.0	0.0	2.0	0.0	0.0	9.5	2.4	...	0	0	0	0	0
4	2.0	3.0	2.0	20.0	19.5	2.0	15.5	0.0	21.0	0.0	...	0	0	0	1	0
5	2.0	3.5	1.5	19.0	15.5	2.0	17.5	0.0	13.0	2.5	...	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
141	3.0	4.0	3.0	20.0	19.0	3.0	15.0	18.0	40.0	2.4	...	0	0	0	0	0
142	0.0	2.0	2.0	8.0	5.0	2.0	3.0	9.0	6.5	2.4	...	0	0	1	0	0
143	2.5	0.0	1.5	19.5	5.5	1.5	12.5	0.0	29.0	2.4	...	0	0	0	0	0
144	3.0	5.0	3.0	18.0	20.0	3.0	20.0	18.0	24.0	2.4	...	0	0	0	0	0
145	3.0	2.0	2.5	18.5	18.5	0.0	13.0	15.0	8.2	2.4	...	0	0	1	0	0

145 rows × 50 columns

Fig. 2. Processed Data

For training the network, we read the processed data files and divided these data into train set and test set, proportions are 80% and 20% relatively. The train process obtains a well-performed network with regression equation, and the test set proves the model with high generalization ability rather than overfitting, because overfitting enable network to provide poor prediction results.

2.2 Constructing Neural Network

After cleaning the data, we started to train experiment to obtain masked neural network. First, as shown in Figure 3, we used pytorch to build a three-layer shallow mask network: input layer, hidden layer and output layer. During neuronal transmission, we calculate the value using regression equation and spare space to record the mask values in the neurons[1]. Also, the hidden layer adopts the ReLU function as the activation function to scale the data, as shown in figure . This network back-propagates to revise the weights and bias in the equation continuously revised and fit the training data. Moreover, we use Adam optimizer to speed up convergence by adjusting parameters. Finally, the test set is used to test the reliability of the model.

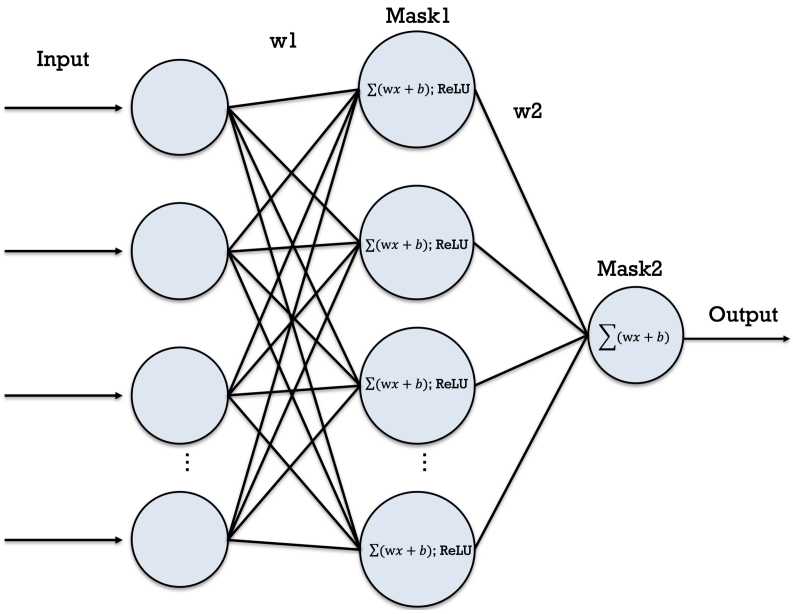


Fig. 3. Constructed Unpruned Network

2.3 Optimising the Network Using Mask Pruning

If we optimising the network using mask pruning, we should know the basic principle that improve generalisation through deleting the weights which are close to zero[1]. As shown in Figure 4(a) and 4(b) initially, we obtain all weights and calculate mask values. Specifically, we calculate the absolute value of weights and sort them. According the pruning rate, we set the binary mask according the pruning rate, which means that the pruned mask value is set by zero and the unpruned is one[8]. From the results of mask, we can obtain new network with edited weights as Figure 4(c).

Fig. 4. Mask Pruning Principle

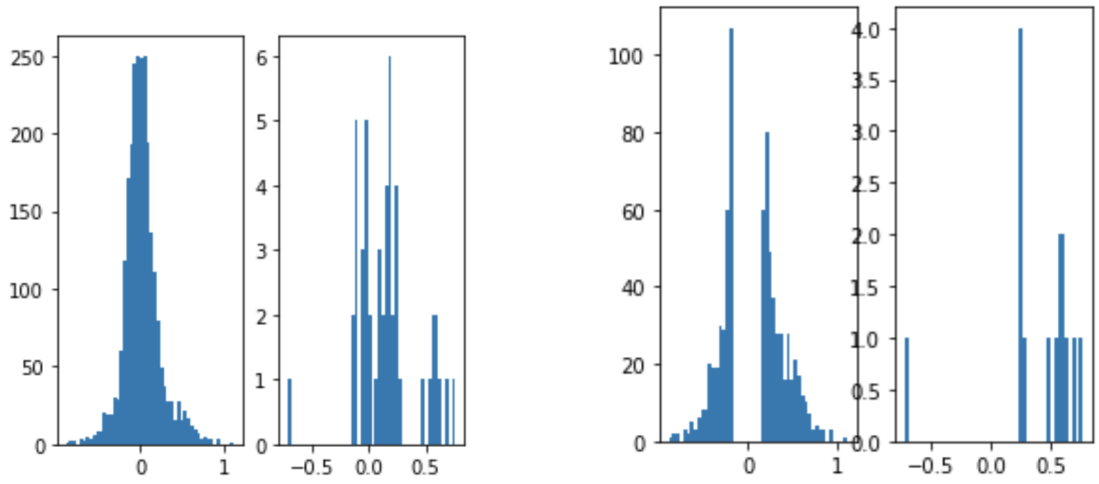
After pruning, we expect the network becomes sparse and computing efficiency increases, and keep high performance. Hence, we should find the best pruning proportion that has highest accuracy than previous techniques. This means that we need to experiment the various pruning percentages and obtain well-performed results. As for evaluation the performance of the network, we determine a reliable range 0 to 5, which means that the prediction results can be considered as correct if the difference between predicted results and actual results is less than 5. According to this evaluation method, the network with distinctiveness has only 5 correct results.

3 Results

In this section, we compare it to the original network and the network with distinctiveness which removes neurons[3]. Also, we find the range of pruning percentages can perform closed or better to the unpruned network. For clarifying the those three kinds of network, we determint that the network with distinctiveness as previous network, new unpruned network as unpruned network and mask pruning network as pruned network.

3.1 Pruning Rate Exploration

After mask pruning, we compare the unpruned network and mask pruning network. Specifically, there are two pairs figures which represent that there are two layers to save the mask values such as hidden and output layers. Due to gain the parameters in the unpruned and pruned model such as weights and biases, We can conclude that the weights closed to zero have been changed to zero, as shown in Figure 5.



(a) Unpruned Network (b) Mask Pruning Network

Fig. 5. Comparison of Weights

In this section, we expect to explore the best pruning percentages to sparse the network and still maintain a well performed network. Hence, we research the different accuracy in various pruning percentage[6]. From the Table 1 , we statistics the correct number in the test set varying pruning rate, and find that the mask pruning causes an effect on the unpruned network. Accuracy may improve or decrease, but mask pruning does not have a huge impact on the unpruned network. As for pruned network, if the pruning rate is less than 30%, the network can still maintain a closed accuracy comparing the unpruned network, but the accuracy will decrease for a high pruning rate. Also, the pruning rate is 20%, it will improve the unpruned network accuracy.

Table 1. Correct Number in Different Pruning Rate

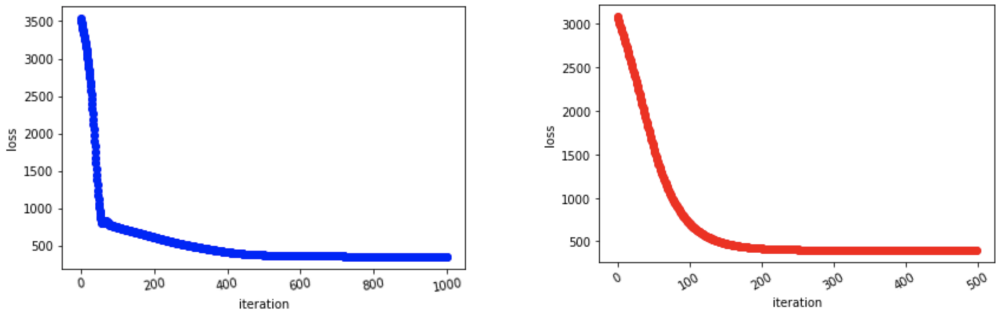
Pruning Rate(%)	5	10	15	20	25	30	35	40	45	50	55	60	65	70
Unpruned model	7	8	8	9	8	8	7	8	7	8	8	9	7	9
Mask pruning model	7	7	8	7	7	7	3	5	4	2	2	3	2	1

3.2 Performance Comparison

To begin with, we compare the network before mask pruning to the reduction network through the distinctiveness of hidden units. As shown in Figure 6, we can find that the unpruned model has faster convergence speed and lower loss than the previous network. Furthermore, from the results of previous network, we can know that the correct number is 5 which is less than the unpruned and pruned network. Also, we can prove it through the Table 2 , and hence, the new neural network has better performance than the network with distinctiveness[2].

Table 2. Comparison with Previous Network

-	Loss after 1000 epoch	Range of epoch	Range of loss
Unpruned network model	361	1-100	3085-708
Model with distinctiveness	399	1-100	3487-747



(a) Unpruned Network (b) Network with Distinctiveness

Fig. 6. Comparison of Loss Function

In conclusion, according to the results of network with distinctiveness and Table 1, we can find that this new model has a better performance in different perspectives such as accuracy, sparsity and convergence speed.

4 Discussion and Future Work

In this experiment, we build a new model through the mask pruning method. Combining the previous one (network with distinctiveness), we have trained three models and compared the performance among those models. We found that the new models improve the accuracy and efficiency but decrease the loss and calculation, which means that the new models' prediction results are more reliable and precise. However, since the samples of dataset are not adequate to train the model, the accuracy is still low although higher than previous network and adjusting the hyperparameters to optimize the model. If we need to improve the models in the next step, we may consider a multiple-hidden-layer neural network or collect more reliable data for training and testing.

In the future work, we should not just focus on the shallow neural network and try a complicated network. For instance, we can use a two-hidden-layer neural network and mask pruning them with different pruning percentages. Since the output is one result in this mark prediction context, mask pruning may cause a huge influence on the final results. Hence, if we further research this problem, we will distinguish the pruning rate with different kinds of layers.

References

[1] S. Anwar, K. Hwang, and W. Sung. "Structured Pruning of Deep Convolutional Neural Networks". In: *Emerging Technologies in Computing Systems* 13 (3 2017). DOI: https://doi.org/10.1145/3005348.

[2] J. Choi, M. Bouchard, and T.H. Yeap. "Decision feedback recurrent neural equalization with fast convergence rate". In: *IEEE Transactions on Neural Networks* 16.3 (2005), pp. 699–708.

[3] T.D. Gedeon and D. Harris. "Network Reduction Techniques". In: vol. 2. 1991, pp. 25–34.

[4] T.D. Gedeon and H.S. Turner. "Explaining student grades predicted by a neural network". In: 1991, pp. 609–612.

[5] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas. "Data processing for supervised learning". In: *International Journal Of Computer Science* 1 (1 2006). ISSN: 1306-4428.

[6] A. Mallya and S. Lazebnik. "PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[7] O.E.D. Noord. "The influence of data preprocessing on the robustness and parsimony of multivariate calibration models". In: *Chemometrics and intelligent laboratory systems* (1994), pp. 65–70.

[8] SunY., X. Wang, and X. Tang. "Sparsifying Neural Network Connections for Face Recognition". In: *CoRR* abs/1512.01891 (2015). arXiv: 1512.01891. URL: http://arxiv.org/abs/1512.01891.

[9] S. Wu et al. "Training and Inference with Integers in Deep Neural Network". In: *A conference paper at ICLR* (2018).