Distinctiveness Measurement and Network Reduction – Implementation and Investigation

Hengjia Li
Research School of Engineering, Australian National University
01/05/2019

Abstract. For any neural network architecture, it is most difficult to determine the number of hidden units, although the numbers of input and output unit can be determined straightforwardly. To obtain an optimal structure of hidden layer, it has been proposed that the number of hidden units can be manipulated according to their "distinctiveness". In this research, this criterion of distinctiveness is implemented on a simple two-layer neural network. The feasibility of this method is proved by a comparison between the network performance before and after the implementation. Three parameters of the network model, which are number of input features, hidden units and value of learning rate, are proved to have some effects on the reduction performance by a set of experiments.

Keywords: neural network, hidden units, distinctiveness, network reduction

1 Introduction

To design a neural network architecture, it is usually easy to determine the number of input and output units straightforwardly, but it is most difficult to find out the optimal number of hidden units. This research focuses on investigating the method of distinctiveness measurement (Gedeon & Haris, 1991) to remove excess hidden units and to decide an optimal network size. A simple two-layer network is adapted to perform prediction tasks and distinctiveness measurement on the Jae-asi eye-tracking data set (Kim, et al., 2014). Meanwhile, some other experiments are conducted to investigate three parameters (number of input features, hidden units and value of learning rate), which may potentially affect the reduction performance.

1.1 Jae-asi eye-tracking data set

Jae-asi eye-tracking data set is collected to exam the user's search performance and behaviors on different size of screens (Kim, et al., 2014). It consists of gaze data from 640 patterns, including 320 patterns collected from large screens and 320 patterns from small screens. There are in total 25 features representing the user's search behavior. Comparing to those small data set with few input features, Jae-asi eye-tracking data set provides enough data for implementing and analyzing the distinctiveness reduction method. This is the primary reason that Jae-asi eye-tracking data set is selected as the target data source. Another reason of selecting this data set is that it mainly consists of numerical data and there is no need to pre-code the input features.

Since there are same amount of large screen and small screen patterns, naturally the screen size is chosen to be the prediction target. Therefore, it forms a binary classification problem (classify large or small screen) by 25 input features, which is solvable by a relatively simple network architecture. This simple classification problem and simple network architecture saves a lot of time while performing iterative experiments.

1.2 Network Reduction Methods

Existing network reduction methods includ measurements of relevance (Mozer & Smolenski, 1989), contributions (sanger, 1989), sensitivity (Karnin, 1990) and badness (Hagiwara, 1990) of hidden units. (Gedeon & Haris, 1991) demonstrates analysis on the disadvantages of above methods and they propose a superior solution for network reduction, named as disctiveness measurement.

The concept of distictiveness refers to similarity between hidden neuron's performance throughout the entire data set. For each hidden unit, a vector of same dimensioality as the number of training patterns is constructed and therefore, the activations of hidden units are vectorised in pattern space. From the paper, there are three scenarios that a neuron may be removed:

- when the magnitude of pattern space vector is relatively small. It imlies that the unit is always off during training process. Figure 1 demostrates an example of this scenerio. There are 25 hidden units in total and it can be observed that the 18th and 19th units are deactivated during training, since their corresponding vector magnitudes are almost zero.

- when there are a pair of pattern space vectors that has a angular seperation smaller than 15 degrees. It implies that the pair of hidden units are producing silimar outputs for each pattern. To remove the redundant unit without further training, the weight vector of the removed unit is added to the weight vector of the remained unit. Similarly, a pair of hidden units are effectively complementary if their angular seperation is larger than 165 degrees. In this case, the removed unit's weight vector is substracted from the remained unit's weight vector.

- Groups pf units together having no function or producing a constant effect. According to (Gedeon & Haris, 1991), this scenerio is uncommon and it is ignored in our experiment.

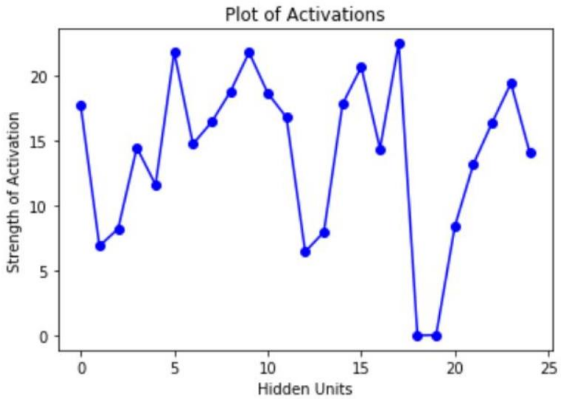

Figure 1: Magnitude of pattern space vectors of hidden units

The following parts of this paper will firstly prove that the above reduction methods can efficiently remove the excess units without performance drop. Then, a set of experiments will be conducted to reveal the parameters' (number of input features, hidden units and learning rates) effect on reduction performance.

2 Method

A simple two-layer neural network is adapted for solving this binary classification problem based on Jae-asi eye-tracking data set. The reason of establishing a simple network architecture is that it will save computational resources while implementing iterative experiments later.

2.1 Pre-process of dataset and network architecture

The binary classification problem aims to apply up to 25 input features about user's search behavior to predict the screen size (large or small). These 25 input features are all numerical, and therefore no pre-coding is needed. However, the values of these inputs are recorded in different units and Table 1 demonstrates some example data.

Table 1: Example data from Jae-asi eye-tracking data set

Search Performance			*Search Behavior*		
Time to first click (sec)	Task Completion Duration(sec)		Mean Fixation	Skip	Search Traceback
13.235	74.97	……	5.328	0	0
18.958	48.582		4.798	4	2
45.679	114.027		21.364	1	1

According to Table 1, it can be observed that the value of input data varies in a relatively large range and normalization of input features is necessary in this scenario. Without normalization, some large-value inputs may produce lager impact to the network than the other small-value inputs during training. Another reason for normalization is that later experiment will investigate how the number of input features will affect reduction performance. Therefore, the values of

input features are expected to be in the same range (e.g. from -1 to 1) to ensure that they have the same importance to the network.

We applied normalization by standard deviation (Abdi & Williams, 2010) in this experiment. Assume x stands for an arbitrary column of input feature data. Then, its value is scaled into the range of 0 and 1 by

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

Then, the scaled values are normalized by their standard deviation and the normalization result can be obtained as

$$x_{normlaized} = \frac{x' - \min(x')}{sd(x')} \quad (2)$$

Therefore, the target two-layer network model will receive 25 normalized input features. It also contains 15 hidden neurons by default and two output neurons for binary classification. 80 % of Jae-asi eye-tracking data set is randomly selected as training set and the rest are grouped as testing set.

2.2 Implementation of network reduction method

After certain number of epochs, the entire training data set is fed into the target two-layer neural network and activations of hidden layer are recorded. For each hidden unit, an activation vector can be formed in pattern space and distinctiveness measurements are performed to locate excess units. According to the magnitude and vector angles of each activation vector, deactivated or cloned units can be removed by setting its outgoing weight to be as 0. Thus, the hidden unit will behave just as removed.

2.3 Experiments

There are in total 4 training experiments in this research and their corresponding network parameters are shown in Table 2.

Table 2: Experiment Parameters

Experiment	*Number of hidden units*	*Number of input features*	*Learning rate*	*Number of Epochs*
0	20	25	0.01	1000
1	20 - 700	25	0.01	1000
2	20	3 - 25	0.01	1000
3	20	25	0.0001 - 0.1	1000

Experiment 0 primarily replicates the method of distinctiveness measurement from (Gedeon & Haris, 1991) to prove its feasibility. The prediction accuracies of the target network will be compared before and after unit reduction. Therefore, the reduction performance can be visualized by observing differences of these two accuracies.

Experiment 1 investigates whether the number of hidden units will affect the reduction performance. The prediction accuracies before and after reduction are again recorded and compared to each other, when different numbers of hidden unit (from 20 to 700) are employed in the network.

Experiment 2 investigates the relationship between the number of input features and the reduction performance. The 25 input features are fed into the target network one by one and each prediction accuracy is recorded for visualizing reduction.

Experiment 3 aims to reveal whether the value of learning rate will affect the reduction performance. The learning rate of training is set to be from 0.0001 to 0.1 and similarly, the corresponding training performance is recorded for visualization of reduction effect.

3 Results and Discussion

Experiment 0

Figure 2 demonstrates the training log of the proposed two-layer network. According to the training result, the training accuracy reaches 95.77% at the last epoch. The hidden neuron behaviors are projected onto the pattern space and the following modifications are made:

1) Small magnitude of activation vector implies that the 16th unit is always off during training. To remove this unit, its outgoing weight is adjusted to be 0. Figure 2 shows an overview of the magnitude of hidden units.

2) Small vector separation implies that the 1st unit and the 20th unit are performing the same functions during training. To remove clone units, the outgoing weight of 20th unit is adjusted to be 0.

As a result, there are two hidden units "removed" from the two-layer network. Then the entire training set is fed into the network again and the training accuracy after reduction reaches 95.96%. Surprisingly, the network even performs a slightly better prediction on the training set than itself before reduction, and this result agrees with the proposed concepts from (Gedeon & Haris, 1991).

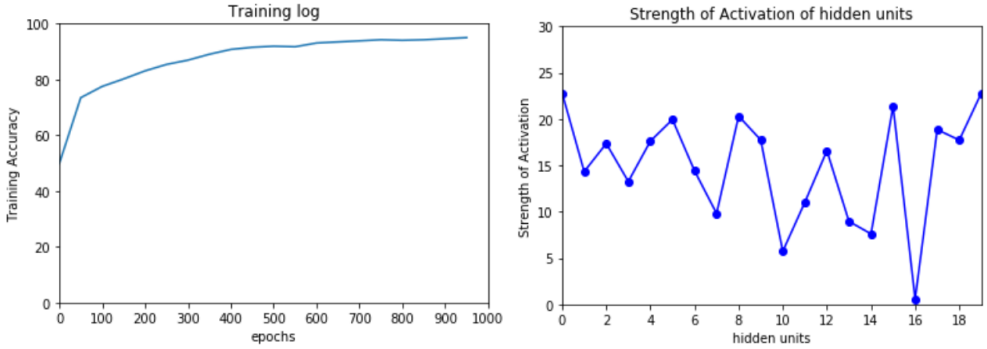

Figure 2: Training log of accuracy (left) and magnitude of activation vectors(right)

Experiment 1

Figure 3 (left) demonstrates the prediction accuracy of the target two-layer network, with different number of hidden units (from 20 to 700). Figure 3(right) shows the reduction rate and difference of training accuracies before and after unit reduction, both presented in percentage. The number of training input is fixed to be 25 and the learning rate of training is 0.01. According to (Ma, 2018), the reduction rate here is calculated by

$$Reduction\,rate = \frac{number\ of\ removed\ units}{number\ of\ initial\ units} * 100\% \qquad (3)$$

and it basically implies how many units are removed.

According to the experiment result, it can be observed that:

1. The training and testing accuracies remain stable while increasing the number of hidden units in the network. It implies that the very large number of hidden units doesn't make much effect on the network performance.

2. The reduction rate increases significantly while larger number of hidden units are employed. Meanwhile, the network prediction accuracies remain stable before and after unit reduction. It reveals that the distinctiveness measurement becomes more efficient when there are more hidden units in the network. It also further proves that the unit reduction will not cause significant performance drop, no metter how many units are removed.

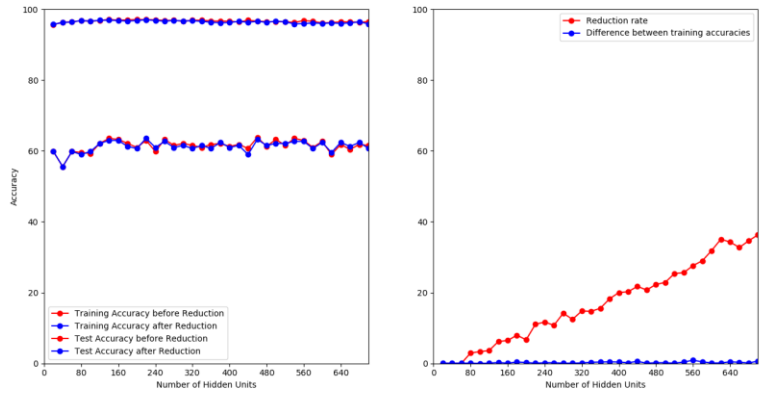

Figure 3: Network prediction accuracies (left) and Reduction ratio (right) with different numbers of hidden units.

Experiment 2

In this experiment, the number of input features are changed from 3 to 25 and the corresponding training accuracies and reduction ratio are recorded as in Figure 4. The number of hidden units is fixed to be 20 and learning rate is 0.01. According to the experiment result, it can be observed that:

1. The prediction accuracy improves with more input features fed into the network.

2. With few training inputs, a significant gap is observed between prediction accuracies before and after removing excess units. It implies that reduction is unstable at this stage and it is sometimes removing useful units. Meanwhile, a high removal rate is observed. It is because that the weights are not fully trained with too few input features and the pattern-space vectors remain small angular separations between each other until the end of training.

3. The unit reduction starts to be stabilized after more than 12 input features are fed into the network. Correspondingly, the post-reduction classification accuracy starts to converge to the accuracy level before reduction. It can be observed that the distinctiveness reduction technique favors a properly trained network with enough input features as reduction target.

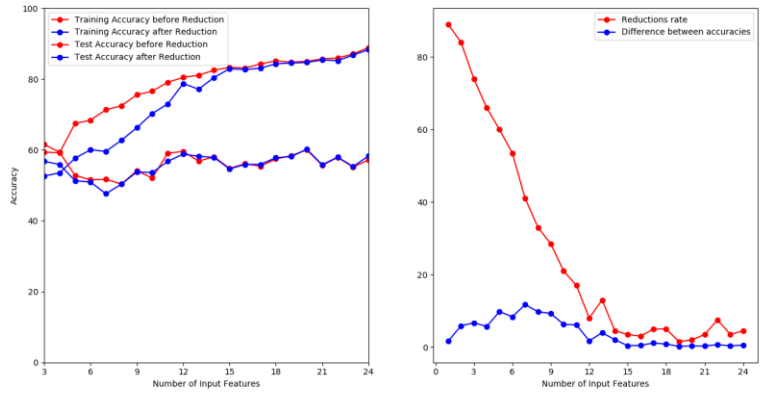

Figure 4: Network prediction accuracies (left) and Reduction ratio (right) with different numbers of input features

Experiment 3

The learning rate is adjusted from 0.0001 to 0.1 during training. Meanwhile, the number of hidden units is fixed to be 20 and all the 25 input features are fed into the target network. The corresponding classification accuracies and reduction ratio are recorded as in Figure 5. According to the figure, it can be observed that:

1. The training accuracy becomes unpredictable while increasing learning rate. A small learning rate limits the step size of back propagation and a large learning rate results in over-shoot. Therefore, the network provides

low training accuracies at the two ends of the accuracy plot. By observation, the network produces a relatively stable classification performance when the learning rate is between 0.01 and 0.045.

2. The reduction ratio is slightly increasing but the prediction accuracy is unpredictable while increasing the learning rate. It indicates that distinctiveness measurement is biased by the defective network. In other words, the network becomes unstable and the reduction fails to maintain the classification accuracy after removing excess units, when improper learning rate is applied.

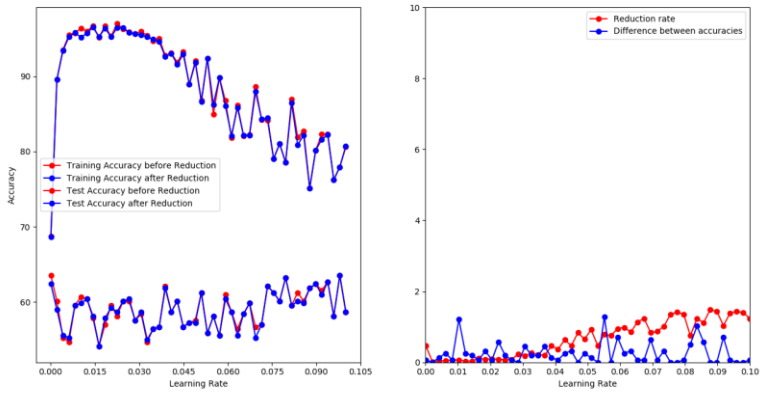

Figure 5: Network prediction accuracies (left) and Reduction ratio (right) with different values of learning rates

4 Conclusion

This research proves that the distinctiveness measurement technique can effectively remove excess units in the target network, without significant drop on prediction accuracy. This experiment result is compared to other published works and the following conclusions are drawn based on investigation:

- The method of distinctiveness measurement can remove more excess units with no performance drop, when there are more hidden units employed in the network
- This method performs much more stable and efficient with enough input features.
- If the target network is not fully trained (e.g. no enough input features or extreme learning rate), the network performance after unit reduction will become unstable or unpredictable. So, network reduction is unreliable in this case.

In this research, only three parameters (number of hidden units, input features and learning rates) of the network model is studied and discussed. The other parameters, such as batch size, number of epochs etc. of the target network are assumed to be constant in our experiment. The future research can be established to investigate how these parameters will affect the reduction performance. Meanwhile, only one data set is used here, and other future works may include using more input data to improve the generality of our conclusions.

References

[1] Abdi, H. & Williams, L., 2010. Normalizaing Data. *Encyclopedia of research design,* Volume 1.

[2] Gedeon, T. & Haris, D., 1991. NETWORK REDUCTION TECHNIQUES. *Proceedings International Conference on Neural Networks Methodologies and Applications,* Volume 1, pp. 119-126.

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

[4] Karnin, E., 1990. A simple procedure for pruning back-propagation trained neural networks. *IEEE Transactions on Neural Networks,* Volume 1, pp. 239-242.

[5] Kim, J. et al., 2015. Eye-tracking analysis of user behavior and performance in web search on large and small screens. *Journal of the Association for Information Science and Technology,* 11 June, 66(3), pp. 526-544.

[6] Ma, J., 2018. *Network Pruning Technique – Implementation and Analysi,* Canberra: Research School of Computer Science, Australian National University.

[7] Mozer, M. & Smolenski, P., 1989. Using relevance to reduce network size automatically. *Connection Science,* Volume 1, pp. 3-16.

[8] Sanger, D., 1989. Contribution analysis: a technique for assigning responsibilities to hidden units in connectionist networks. *Connection Science,* Volume 1, pp. 115-138.