Pruning Methods based on Hidden Neuron Functionality

Idris Mustafa
Australian National University
Acton, ACT, Australia
u6733671@anu.edu.au

Abstract: Pruning has always been an important part of fine tuning neural networks. The complex architecture of the neural network and various parameters used may cause problems in generalisation performance. Pruning process reduces the size of neural network, making it more efficient. In this paper we compare the similarity measures, introduced in Weight Matrix versus Neuron Behaviour (T.D. Gedeon)[1], used for the pruning process in a back propagation neural network designed for a classification problem. The paper focuses on the results of pruning performed based on two different similarity measures, first using the *distinctiveness* computed by the hidden neuron output and second, using the *distinctiveness* computed by the weight matrix.

Keywords: Pruning, Neuron Behaviour, Network Reduction

1. Introduction

Neural network is a popular tool used to solve problems ranging from simple classification, regression to natural language processing and computer vision tasks. Massive amount of data gathered in the last decade has contributed to the increasing popularity of deep neural networks used in numerous fields of study. As the application domain is constantly growing, the architectures of various neural network models have become more complex. Therefore, problems such as generalisation, computation costs, storage and efficiency require more attention.

Pruning is a popular method used to compress the pre trained network, which helps reduce the computation cost and increase generalisation performance. J. Sietsma and RJF. Dow have described how generalisation is improved in multilayered neural network by taking only few neurons in first layer[4]. Rules extraction from pruned networks for breast cancer diagnosis has been described in detail by R. Setiono[6]. In this paper we use pruning of hidden neurons based on the *distinctiveness* property introduced by Gedeon and Harris[2] and compare the results with pruning based on the *distinctiveness* property defined by the weight matrix[1]. Another successful research describes the *sensitivity* property that can be used for pruning process[5]. In our experiment we use a two layer feed forward neural network designed on the Static Facial Expression dataset[3]. The dataset contains 675 images labelled for 7 expressions *angry, disgust, fear, happy, sad, neutral* and *surprise.*

2. Methodology

We use a feed forward neural network with one layer of hidden neurons and the weights are updated by the backpropagation algorithm. It is a common practice of taking the number of hidden neurons somewhere between the number of input neurons and the output neurons to train a model efficiently, and increase the number of hidden neurons further if there is no improvement in the model. As the number of hidden neurons are increased, the model is likely to produce some hidden neurons that are too similar or complimentary and unnecessary. To easily depict the scenario we choose 15 hidden neurons for our model, the number being larger than the input and the output size.

We define a similarity measure for the hidden neurons equivalent to the *distinctiveness* property defined by Gedeon and Harris[2]. In our experiment, we take the hidden neurons activation output vector and the weight vectors connected to the output neurons to measure the similarity between neurons and compare the accuracy after pruning using these two similarity measures.

2.1 Similarity Measure

The similarity measure is computed by taking any two vector and finding the angle of separation between them. We then implement a threshold system to detect the similar pairs and complementing pairs. The pair is similar if the angle separation between them is less than 30 degrees and complementary if the angle of separation exceeds 150 degrees. The computation that follows after detecting the similar or complementary pair is discussed in the subsequent sections.

3. Network Model

The input to the model is the 10 features extracted from the dataset that includes 5 features belonging to Local Phase Quantization and 5 features belonging to Pyramid of Histogram of Gradients. The accuracy got from using Local Phase Quantization features and Pyramid of Histogram of Gradients is slightly varying as discussed by Gedeon in his experiment[3]. As the motive of our experiment is focussed mainly on the outcome of pruning from two different similarity measures, we care less about the accuracy of LPQ and PHOG separately, hence both are chosen as the input features to our model.

The model is trained on 70% of the original dataset for 1000 epochs at a learning rate of 0.01. Sigmoid activation is used for the hidden neurons and the model is optimised using the Adam optimiser. The testing set accuracy got is 22%.

Emotions	**Angry**	**Disgust**	**Fear**	**Happy**	**Neutral**	**Sad**	**Surprise**
Precision	0.23	0.20	0.31	0.21	0.03	0.43	0.15
Recall	0.18	0.13	0.27	0.24	0.07	0.31	0.28
F1-score	0.20	0.16	0.29	0.23	0.04	0.36	0.20

Table1. Various evaluation measures implemented on the test set

We extract the hidden neurons output after the model is trained to begin the pruning process. We find the angle separation between neurons and split them into similar and complementary pairs using the threshold system discussed in section 2.1. For the similar pairs, we remove one of the neurons and the weight vector of the removed neuron is added to the one that remains. For the complementary pairs, both the neurons are unnecessary and removed. The new weights are then replaced in the model that is pre trained.

Similar procedure is repeated but we now use the weight vectors from hidden neurons to the output neurons to measure similarity.

Pairs		Angles	Similarity
0	1	24.331404861484447	Similar
0	2	146.1164337441419	
0	3	132.25963282396518	
0	4	122.56606442934299	
0	5	142.69377130953515	
0	6	131.88621272166222	
0	7	34.664166264298515	
0	8	145.00200641878428	
0	9	27.853400444737986	Similar
0	10	151.02292745517724	Complementary
0	10	151.02292745517724	Complementary
0	11	146.94683448305372	
0	12	145.25858930514497	
0	13	77.83848868724445	
0	14	141.41475008555855	

Table2. The angle separation calculated from activation output of the hidden neurons.

The *distinctiveness* property defined by Gedeon and Harris[2] uses a threshold of 15 and 165 to measure similarity. In our experiment, the model rarely produces any vector pair with an angle separation less than 15 degrees when it is repeatedly trained from the scratch. Hence a threshold of 30 and 150 is used only to better understand the pruning effect by two different similarity measure computation.

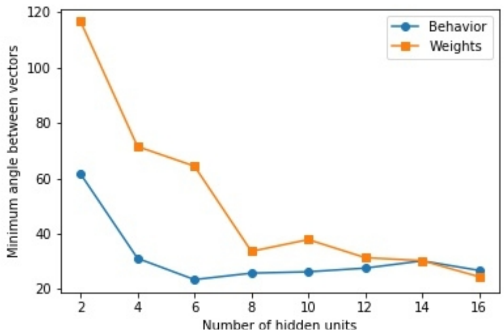

Figure1. Minimum angle between vectors during pruning process

3.1 Pruning Results

We compare the results of pruning from similarity measure computed by neuron behaviour and output weight matrix. We add more neurons in the hidden layer to test our pruning process. The testing is done in 3 different settings and the model is run for different number of epochs in each of the settings.

Setting 1 – 50 neurons in the hidden layer.

Setting 2 – 75 neurons in the hidden layer.

Setting 3 – Varying number of hidden neurons trained for fixed number of epochs(1000) repeatedly.

In settings 1 and 2, the result comparison focuses mainly on the test set accuracy and in settings 3 we focus on the degree of compression after the pruning process.

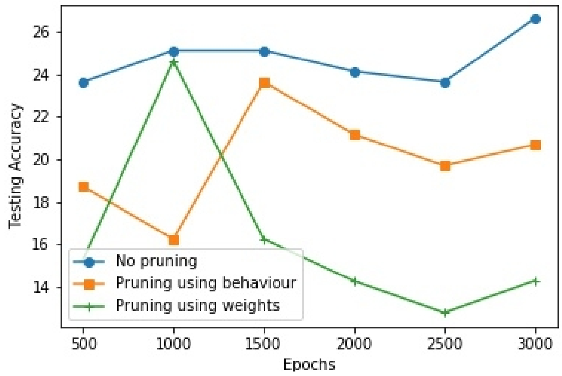

Figure 2. Test set accuracy in setting 1

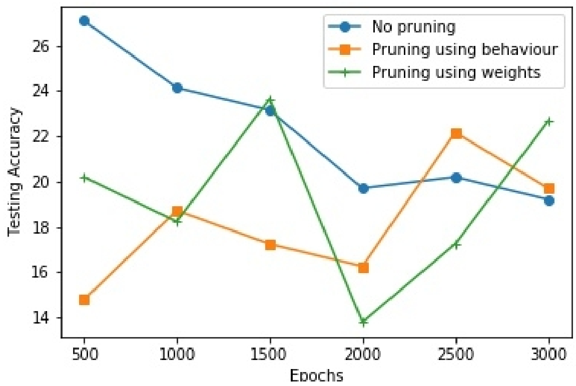

Figure 3. Test set accuracy in setting 2

Hidden Neurons	**Test accuracy (no pruning)**	**Neurons pruned**	**Test accuracy**
75	26.11%	21	21.67
100	26.11%	44	18.23
125	24.14%	49	16.26
150	23.65%	68	19.7

Table 3. Pruning in setting 3 using neuron behaviour to compute the similarity.

Hidden Neurons	**Test accuracy (no pruning)**	**Neurons pruned**	**Test accuracy**
75	26.11%	24	20.20
100	26.11%	51	15.76
125	24.14%	66	18.23
150	23.65%	89	13.79

Table 4. Pruning in setting 3 using output weight matrix to compute the similarity.

4. Conclusion and Future Work

We have shown the significance of using hidden neuron activation output as a tool to measure similarity against the similarity measure computed from static weight matrix. Figure 2 and 3 shows how testing accuracy is highly deviant when pruning is done using static weight matrices. The hypothesis made by Gedeon[1] holds true in our experiment. The static property of the network is not a reliable information to carry out the pruning process. The behaviour is indeed the right tool to use for differentiating the functionality of neurons. Our experiment uses a simple feed forward network and describes the effect of pruning using two different similarity measures and the outcome of the experiment only explains the generalisation performance. The research can be extended to deep neural networks to investigate better the impact on other properties like computation cost and storage.

5. Reference

[1] T. D. Gedeon, "Indicators of hidden neuron functionality: the weight matrix versus neuron behaviour," Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, Dunedin, New Zealand, 1995, pp. 26-29.

[2] Gedeon, T. D., and D. Harris. "Network reduction techniques." *Proceedings International Conference on Neural Networks Methodologies and Applications*. Vol. 1. 1991.

[3] Dhall, Abhinav, et al. "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark." *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2011.

[4] Sietsma, Jocelyn, and Robert JF Dow. "Creating artificial neural networks that generalize." *Neural networks* 4.1 (1991): 67-79.

[5] E. D. Karnin, "A simple procedure for pruning back-propagation trained neural networks," in *IEEE Transactions on Neural Networks*, vol. 1, no. 2, pp. 239-242, June 1990.

[6] Setiono, Rudy. "Extracting rules from pruned neural networks for breast cancer diagnosis." *Artificial intelligence in medicine* 8.1 (1996): 37-51.