Analysis of the Effectiveness of Pruning Methods on Neural Networks of Varied Depth

Prateek Aranha

Research School of Computer Science, Australian National University, Canberra, Australia. u6872485@anu.edu.au

Abstract. Pruning can be defined as cutting down on excess. In the context of neural networks pruning can be thought as the process of eliminating excess neurons or neurons that don't contribute significantly in a neural network. This paper is aimed at comparing how different Pruning Methods perform on a Two Layered Neural Network against a Deep Neural Network. In this paper we explore the performance of Pruning by Relevance, Sensitivity, Badness and Distinctiveness. Due to the simple nature of its implementation Pruning by Relevance gave us significant results in each of the two neural networks. We see that while the more complicated techniques like Distinctiveness, Sensitivity and Badness gave comparable results when applied to both neural networks the accuracy wasn't satisfactory.

Keywords: Pruning · Deep Neural Networks · Network Reduction · Classification · Network Performance.

1 Introduction

Neural Networks of some kind are being used in every aspect of life today. A constant challenge being faced with the broader implementation of neural networks is to attain the ability to run these networks on devices with limited memory and processing power. Most neural networks utilise more hidden neurons than required in order to be able to train the network under a certain time constraint. These excess units at times may not have significant contributions to the final network. We could intuitively come up with a bunch of methods through which we attempt to understand the contribution of each neuron in the network and thus prune the network by eliminating the ineffective neurons. Thus, ending up with a smaller and faster neural network. This paper attempts to analyse the effect on accuracy of a model after applying multiple pre-existing pruning techniques[2] on a geographical dataset[1].

How accurate algorithms or methods are is a question that is usually context specific. In most cases it is hard to analyse the result of techniques over models of varied complexity with datasets from diverse domains and with distinct characteristics as resources available are constrained. We thus try and predict how accurate an algorithm is under certain situations and then try to generalise the prediction over other situations. The same holds for this paper where the results are based on certain assumptions in terms of the model specification and type of dataset used.

1.1 Background

Both neural networks used are feed-forward neural networks without lateral, backward or multilayer connections. The back-propagation algorithm has been used to update the weight values in both networks because of the ease in implementation and theoretical simplicity. The learning rate of both models was set to 0.01. In order to check for the error or loss a k-fold cross validation algorithm was used with k set to 10 as it is shown to be the best value for balancing between generalisation and learning sufficient features in Payam Rafaeilzadeh et al[3]. In order to increase the reliability of the findings each method was applied on 3 two layered and 3 deep neural networks all initialised with different weights. The results for each network type are averaged across the 3 neural networks.

The double layered neural network consisted of the hidden and output layers. The hidden layer of the model consisted of 13 neurons based on the general practice of choosing two thirds of the number of feature (19) values. On further analysis the network

also seemed to perform better with 13 neurons. The output layer consisted of 2 neurons. We have used a Sigmoid activation function in the hidden layer given by

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

In order to come up with the ideal number of epochs, the accuracy with respect to the epochs was analysed and 501 was selected as that seemed to be the number where the curve began to flatten as shown in Fig.1. The Stochastic Gradient Decent Optimiser was used in the neural network as it seemed to be the optimiser that performed best on the neural network.

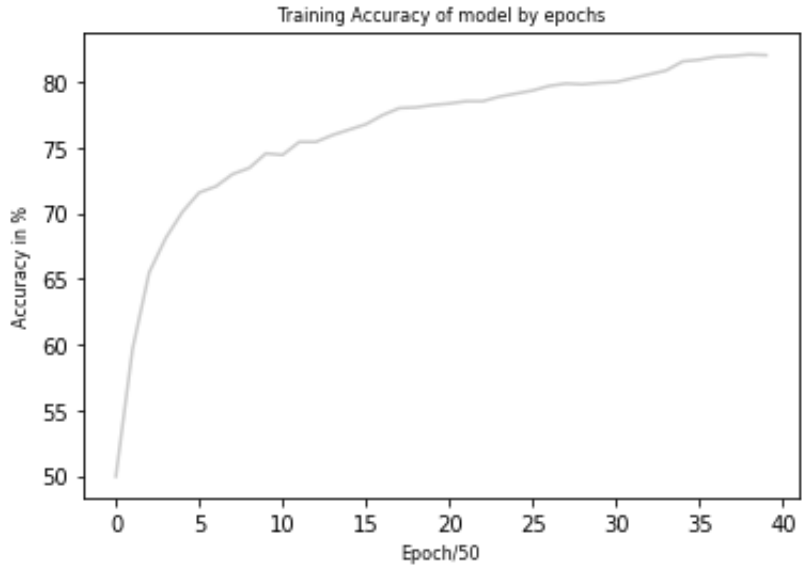

Fig. 1. This figure plots the value of training accuracy that the model generates with respect to the number of epochs in the Two Layered Neural Network.

The Deep neural network consisted of 5 hidden layers and one output layer. The number of neurons in each hidden layer are given in Table 1. The number of neurons in each layer was decided through a trial and error method and the mentioned number of neurons seemed to give the best accuracy. To avoid the loss of information due to gradient loss during back-propagation we have used the Leaky ReLU activation function in the hidden layers given by

$$f(x) = \begin{cases} \alpha x & x < 0, \alpha < 1 \\ x & x \geq 0 \end{cases} \tag{2}$$

The activation function on the output layer was the Sigmoid activation function given by (1). In order to come up with the ideal number of epochs, the loss with increase in the epochs was analysed and 190 was selected as in the epochs after that the loss seemed to become volatile as shown in Fig.2. The Adam optimiser [7] was selected on this neural network as it seemed to be perform the best.

The average test accuracy for the Double Layered Neural Network and the Deep Neural Network was shown to be 69.8% and 72.1% respectively. The performance of the Deep Neural Network was not significantly greater than the Two Layered Neural

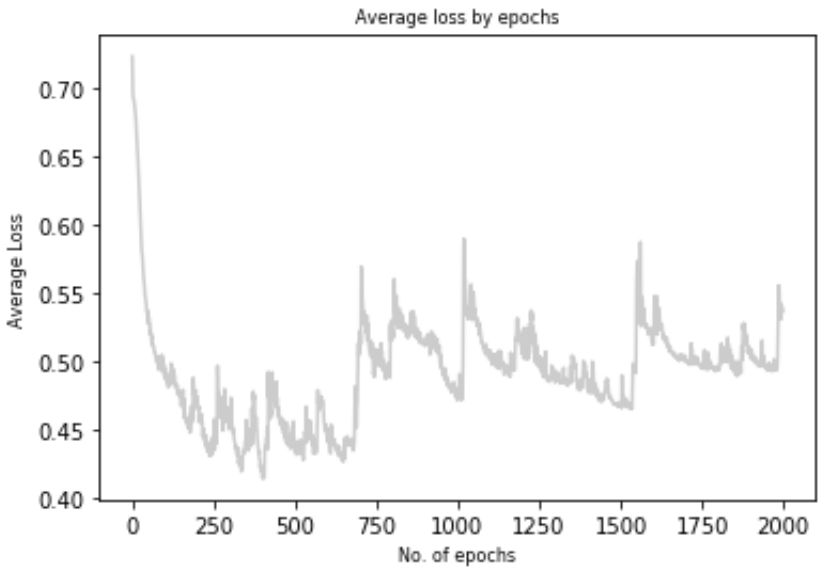

Fig. 2. This figure plots the value of training loss that the model generates with respect to the number of epochs in the Deep Neural Network.

Hidden Layer	Hidden Layer 1	Hidden Layer 2	Hidden Layer 3	Hidden Layer 4	Hidden Layer 5	Output Layer
Number of Neurons	10	7	5	7	10	2

Table 1. This table shows the number of neurons in each layer of the Deep Neural Network

Network. This could be attributed to the limited availability of training data samples (171). Table 2. gives a comparison of the parameters in the two neural networks.

1.2 Data Set

It would be more beneficial to implement our techniques on a dataset that is rich and diverse in information so that we know that the network ensures that the complexity of the data is captured. This is my motivation for basing my analysis on the geographical data obtained from an area in the Nullica State Forest on the south coast of New South Wales, Australia[1]. This dataset comprises of various geographical features of a region and the corresponding forest types. In our analysis we will be trying to predict if a particular region has the Dry Sclerophyll forest type.

We looked at various methods for preprocessing the data in order to improve our predictions. We adopted a few methods from L.K. Milne et al. [1] as these methods improved the performance of the networks when applied. In addition to the methods mentioned in the paper we converted our target variable to a binary value (0,1) from the existing representation where 10 and 90 were used to depict if the forest type was or was not Sclerophyll respectively. We also grouped the values in the feature SL (Slope) into 5 buckets as shown in Table 3. The values for TE (Temperature) were changed from 0, 30, 60 and 90 to 0, 1, 2 and 3 repectively. An in depth description of the feature values after preprocessing can be found in Table 7.

The dataset consists of 190 samples. The target values were almost evenly split with 96 samples where the type was Sclerophyll and 94 where the type was not Sclerophyll. On eliminating unnecessary features like index number and features describing other

Parameter	Two Layered Neural Network	Deep Neural Network
Number of Layers	2	6
Total Number of Neurons	15	41
Activation Function used	Sigmoid	Leaky ReLU
Optimiser	SGD	Adam
No. of training epochs	501	190
Average Accuracy	69.8%	72.1%

Table 2. This table gives information on the specifications of the two neural networks used

Original Slope Value	≤ 30	$= 40$	$= 50$	$= 60$	≥ 70
New Slope Value	1	2	3	4	5

Table 3. This table shows how slope values were grouped intp different baskets

forest types we ended up with 19 features. Of these there were 4 features for Aspect Ratio, Slope, Geology, Topographic position, rainfall, temperature and Landsat TM bands T1 to T7.

2 Method

2.1 Relevance

Relevance is based on the idea of understanding how well the network works with a neuron in place, versus when the neuron is removed from the network [4]. In this method we take the pre-existing trained network and remove a certain neuron from the hidden layer. This is done by switching all the corresponding incoming weights of the neuron to zero, thus not allowing the neuron to contribute to the model prediction. If on eliminating a neuron the accuracy of the network did not fall more than 3% then that neuron was accepted.

In our analysis we implemented two versions of Relevance. The first version is as described in the previous paragraph. In the second version we removed two neurons from the neural network and analysed the performance of the network without them. For the sake of ease of reference the first version would be referred to as Relevance1 and the second version as Relevance2 henceforth in this paper.

2.2 Sensitivity

Sensitivity or Karnin Sensitivity is an approach for pruning neural network algorithms where difference in accuracy given the presence or absence of weights is measured[5]. In this method we set the value of a particular weight on a pre-trained model to zero and then calculate the error in the absence of this weight. We do the same for all weights connected to a particular neuron. If on eliminating all weights to a neuron individually in a model, the accuracy does not drop by more than 1% of the actual accuracy of the model then we can safely conclude that the the error won't decrease on eliminating that neuron from the network.

The sensitivity S_{ij} of weight i and neuron j is defined by

$$S_{ij} = E(w_{ij} = 0) - E(w_{ij} = w_{ij}^{f}) \tag{3}$$

where w_{ij} is the weight i to neuron j.

If A is the neurons obtained from Sensitivity, B is the the neurons obtained from Relevance1 and $len(X)$ is the number of neurons present in X. Accuracy of Distinctiveness is calculated by:

$$accuracy(S) = 100 * \frac{len(A \cap B)}{len(A)} \tag{4}$$

2.3 Badness

Badness factor of a neuron is defined as the amount of change the corresponding weights from that neuron to the neurons in the next hidden layer have had to go over the training period[6]. The neuron which has the highest value of badness factor at the end of training is considered to be the neuron with least effect on the model and it is thus considered safe to eliminate that particular neuron.

In this process, during training we keep adding the change in each weight from the current to the next layer of each neuron and thus come up with the badness factor.

Badness factor of the neuron i at epoch k B_{ik} can be calculated as

$$B_{ik} = B_{ik-1} + b_{ijk} \tag{5}$$

where b is the summation of change from neuron i in the current layer to neuron j in the next layer at epoch k and is given by

$$b_{ijk} = \sum_{j=0} |w_{ijk} - w_{ijk-1}| \tag{6}$$

This method was extended by calculating the rank of the all neurons based on the B value. If b_i is the rank of the i^{th} neuron returned by Badness, r_i is the rank of the i^{th} neuron returned by Relevance1, S is the number of neurons present in the network and $isTrue(x)$ returns 1 if x is $True$ and 0 otherwise. The accuracy of Badness for the Neural network can be calculated as

$$accuracy(D) = \begin{cases} 100 * \frac{\sum_{i=0} isTrue(|b_i - r_i| \leq 1)}{S} & TwoLayeredNeuralNetwork \\ 100 * \frac{\sum_{i=0} isTrue(|b_i - r_i| \leq 3)}{S} & DeepNeuralNetwork \end{cases} \tag{7}$$

2.4 Distinctiveness

Distinctiveness is measured by understanding the similarity between the hidden neurons[2]. Neurons that are similar to each other thus not contributing to the network and neurons that are complementary to each other thus cancelling out each others contribution are found. After this one of the neurons in a pair of similar or complementary neurons is kept and the other is dropped.

In this model we first get the final activation of each neuron for each of the sample inputs. We then scale each input over a range of -0.5 and 0.5. After this we use cosine similarity and effectively get the angle of similarity between the activations of each of the neurons. If the similarity is less than 15 degrees we deduce that the two neurons are similar and thus we can eliminate one of the neurons without affecting the accuracy too much. Similarly, if the similarity is greater that 165 degrees we understand that the two neurons are cancelling each other out and thus we drop one of them without affecting the accuracy.

If A is the number of unique values of the neurons obtained from the multiple combinations of tuples, B is the the neuron values obtained from Relevance1 and $len(X)$ is the number of neurons present in X. Accuracy of Distinctiveness is calculated by

$$accuracy(D) = 100 * \frac{len(A \cap B)}{len(A)} \tag{8}$$

Network number	Two Layered Neural Network	Deep Neural Network
First	44.4%	35%
Second	44.4%	60.8%
Third	41.6%	32.4%
Average	43.6%	42.7%

Table 4. This table gives information on the accuracy of the Pruning method Sensitivity

3 Results and Discussion

Note that none of the pruning methods were implemented on the output layer with 2 neurons.

3.1 Relevance

On implementing Relevance1 on the Two Layered Neural Network out of the 13 neurons we were on able to identify 7, 7 and 5 neurons in each of the three neural networks on removing which the accuracy of the network stayed reliable. When Relevance2 was implemented on the same networks 17, 18 and 18 pairs of neurons out of the 85 $((13 * 13)/2 + 1)$ possible combinations respectively were identified as neuron pairs on eliminating which the accuracy of the network stayed reliable.

On implementing Relevance1 on the Deep Neural Network out of the 39 neurons we were on able to identify 17, 15 and 12 neurons in each of the three neural networks on removing which the accuracy of the network stayed reliable. When Relevance2 was implemented on the same networks 86, 54 and 65 pairs of neurons out of the 761 $((39 * 39)/2 + 1)$ possible pairs respectively were identified as neuron pairs on eliminating which the accuracy of the network stayed reliable.

3.2 Sensitivity

From Table 4. we see that the average accuracy when Pruning by Sensitivity is applied to the Two layered Neural Network is 43.6% and the average accuracy when applied to the Deep Neural network is 42.7%. The average accuracies in both network types is almost the same however the result seemed to be more consistent when applied to the Single Layered Neural Network. In the Deep Neural Network the accuracy seemed to be around 33% except for one instance in the second Deep Neural Network where the accuracy went up to 60% thus raising the average. From this we can conclude that Pruning by Sensitivity can be more reliably applied on a Two Layered Neural Network over a Deep Neural Network. However, the accuracy rate of around 43% is still very low, thus inhibiting us to confidently apply the method

3.3 Badness

When looking at the accuracy levels when applying the Pruning method of Badness in Table 5. we see that there is a lot of volatility in the accuracy levels when applied to the Two Layered Neural Network while the accuracy level when applied on a Deep Neural Network is relatively consistent. In either case however the accuracy level is too low for it to be implemented confidently in real life. This is is line with the discussion in T.D. Gedeon & D. Harris [2] where the authors state that the neurons which have had significant change in values over the training period need not be the most insignificant neuron in the network as due to the excessive change during training the neuron might have learnt far more than other neurons and may have a significant contribution to the network.

Network number	Two Layered Neural Network	Deep Neural Network
First	30.7%	15.3%
Second	15.3%	10.2%
Third	7.6%	10.2%
Average	17.8%	11.9%

Table 5. This table gives information on the accuracy of the Pruning method Badness

Network number	Two Layered Neural Network	Deep Neural Network
First	NA	59%
Second	NA	56.5%
Third	NA	38.2%
Average	NA	51.2%

Table 6. This table gives information on the accuracy of the Pruning method Distinctiveness

3.4 Distinctiveness

From Table 6. we observe that on implementing Pruning by Distinctiveness all neurons in the 3 Two Layered Neural networks were identified as significant contributors and thus the method did not return any neuron that can be pruned. This could be a result of having a reasonable number of neurons modelling a very complex data set. In the Deep Neural Network the Pruning method performs with an average accuracy of 51% thus proving to be the best performing pruning method out of the discussed methods.

4 Conclusion and Future Work

We find Pruning by Sensitivity to be more reliable when performed on a Two Layered Neural Network whereas Pruning by Badness was more reliable when performed on the Deep Neural Network. However, the accuracy of both these networks were very low, which stops us from implementing these techniques confidently. Distinctiveness seemed to be the best performer with an average accuracy rate above 50%. However, this is still not satisfactory as to be able to implement these methods in real life the accuracy would need to be much higher.

To explore this domain further, given that Pruning by Distinctiveness already gives us decent results in comparison to other methods we need to start by analysing Distinctiveness further and understanding how it performs when implemented by using different similarity functions and parameters. Another possible way to approach this task would be to try and apply multiple methods on the Neural Network and analyse the result that it achieves.

References

1. L.K. Milne, T.D. Gedeon and A.K.Skidmore: Classifying Dry Sclerophyll Forest from Aumented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood.
2. T.D. Gedeon & D. Harris: Network Reduction Techniques.
3. Payam Rafaeilzadeh, Lei Tanh, Huan Liu: Cross-Validation, 2008 http://www.springer.com/lncs.
4. Mozer, MC, Smolenski, P, “Using relevance to reduce network size automatically,”, Connection Science, vol. 1, pp. 3-16, 1989.
5. Matthew Robert Wilson: Comparison of Karnin Sensitivity and Principal Component Analysis in Reducing Input Dimensionality, TigerPrints(2016).
6. Masafumi Hagiwara:Novel Back Propagation Algorithm for Reduction of Hidden Units and Acceleration of Convergence using Artificial Selection, IEEE Explore, May 06, 2020.
7. Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization, 2014

Feature Name	Minimum Value	Maximum Value	Mean Value	Median Value
SA	0	99	56.6	50
CA	0	99	56.6	50
AL	7	71	33.9	33.5
TP (Topographic position)	16	96	65.7	64
SL (Slope)	1	5	3.3	3
GE (Geology)	1	99	62.4	50
RA (Rainfall)	19	79	39.9	39
TE (Temperature)	0	3	1.8	2
T1	53	91	58.2	58
T2	16	46	19.9	20
T3	14	66	19.8	19
T4	25	80	53.9	54
T5	13	92	34.8	33
T6	15	63	38.9	39
T7	16	90	30.5	28
A1 (Aspect Ratio North)	0	50	26.3	25
A2 (Aspect Ratio East)	0	50	25.7	25
A3 (Aspect Ratio South)	0	50	23.1	25
A4 (Aspect Ratio West)	0	50	23.6	25

Table 7. This table gives a description of the feature values in the GIS dataset