Can pruned neural networks trained with backpropagation compete with genetic algorithms?

Alexander Horvat,

Research School of Computer Science, Australian National University.
Canberra, Australia.

u6048805@anu.edu.au

Abstract: Creating detailed maps of tree species in forests is a useful, albeit prohibitively expensive process. Using features about forests and satellite data, neural networks were trained using backpropagation to classify forest plot species as dry sclerophyll. These neural networks were pruned by 4 to 20% and performed within 1% of each other on the testing data set. Other neural networks were trained using genetic algorithms which outperformed the backpropagation neural networks by only 1.94%. This means that backpropagation neural networks can compete with genetic algorithms in accuracy and speed for future training.

Keywords: feedforward neural network, network reduction, dry sclerophyll forest classification, genetic algorithms

1 Introduction

Detailed maps of forests are useful for forest management purposes. However, collecting data for such detailed maps is typically a prohibitively expensive process [1]. The areas being studied can also change significantly over short periods of time, making previous surveys inaccurate. It is desirable to instead determine forest attributes automatically as these can be frequently updated for much less cost. Automatically gathered information such as altitude, rainfall and satellite imagery data can be used as features for neural networks to classify the species of trees in a forest.

In recent years, neural networks have become a popular method for solving classification problems and often outperform other methods such as decision trees and maximum likelihood. This is because, when given enough training data, they can more accurately define nonlinear boundaries between the classification classes.

Backpropagation is one commonly used method to train neural networks as they are easy to understand and implement. Backpropagation involves calculating the gradient of the loss function with respect to the weights of a neuron and adjust the weights of the network to reduce the loss function. The major disadvantage for using backpropagation is that it can be slow to train networks and the optimal network architecture cannot be identified prior to training. In practice, this leads to neural networks with a larger number of hidden nodes than required, which in turn slows the training process as well as classification [1].

Some of the disadvantages of backpropagation can be reduced by performing pruning on the network after it has finished training. This process identifies neurons which produce similar or opposite output and thus can be removed as they perform the same function in the neural network. Removing neurons that are not necessary for the network to function can improve computational speeds as less calculations are necessary when classifying new data points or when doing further training [4].

Genetic Algorithms are another commonly used method to train neural networks for the same reasons outlined above and address the speed shortcomings of backpropagation. Genetic algorithms work by mimicking the natural process of evolution. They generate a random initial population which is then evolved through the breeding process of crossover and mutation which aims to breed the best characteristics of the top performers each generation for the next. Genetic algorithms are typically faster and more efficient than backpropagation as they quickly search the solution space for a result that is often good enough. This is especially useful in most real-world scenarios when the solution space is complex with many local optima [3].

In this paper, we will be comparing the classification accuracy of 3-layer feed forward neural networks, before and after pruning, that have been trained using backpropagation against unpruned neural networks trained with genetic algorithms. The goal is to compare the performance of the backpropagation neural network before and after pruning against the neural network trained with the faster genetic algorithm method.

2 Method

2.1 Data

Geographical data (GIS) from the Nullica State Forest on the south cost of New South Wales was used. The area was broken up into a grid of 179,831 pixels each of which were 30 by 30m in [3]. The database contained 190 samples with the following features: aspect information, altitude, slope, geology, topographic position, rainfall, temperature and seven TM satellite bands.

The input data was encoded as described by Milne et al. (1995) except for topographic position and geology. The topographic position feature was encoded as a discrete variable using the following function $enc = \frac{tp-16}{80}$ as the only possible values were 16, 32, 48, 64, 80 and 96. The meaning of the geology feature was unknown although as can be seen from Figure 1 below, each pixel most commonly corresponded to one of the values 50, 70 or 90. This feature was encoded using the following discrete mapping: $50 = 0.33$, $70 = 0.66$, $90 = 0.99$, $everything\ else = 0$.

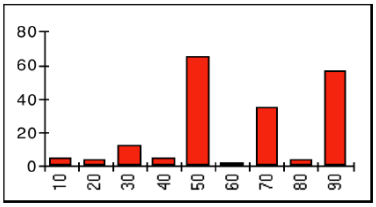

Figure 1 Geology feature (from lecture slide COMP4660 NN05 pp. 11)

The output data had values of 90 and 10 which labelled the datapoint as being dry sclerophyll or not and were scaled to 1 and 0, respectively. The training set was a random 80% selection of the GIS dataset and the remaining 20% was used for the testing set.

2.2 Neural Network Architecture

The neural network consisted of a 3-layer feed forward architecture with 17 input neurons, a varying number of hidden neurons and one output neuron. The middle, hidden layer was tested with starting node amounts of 50, 40, 20 and 14. These numbers were chosen arbitrarily except for 14 which was the number of nodes used in the original GIS classification paper by Milne et al (1995).

All the input features from the dataset were used so that the performance of these neural networks could be compared to the performance of the networks in the GIS classification paper. This aims to serve as a base performance level which can be improved upon. The output layer is a single node as the classification problem is only to determine whether the forest species is dry sclerophyll or not. If the output produces a value above a certain threshold, then the neural network believes that feature pattern belongs to a dry sclerophyll species otherwise it does not.

The neural network was trained and verified using the same methods of back propagation on the training set until the error on the test set was minimal (ibid). This was confirmed by a 5k cross-validation performed on the training set to help prevent overfitting. A k value of 5 was chosen as it would result in a training test data set of approximately 30 which is close to the size of the validation test data set.

Binary cross-entropy, a sigmoid activation function, was used as the loss function because it is suitable for class classification when there are two classes. The optimisation function used was stochastic gradient descent as that is what is being compared in the paper.

2.2.1 Pruning

The pruning step was completed after training finished and followed the similarity methodology described by Gedeon and Harris, 1991. The output of each neuron in the hidden layer was compared pairwise and the vector angles of the neuron's output were calculated using dot product. If the angles were above 165 (complementary) or below 15 (redundant) for two neurons, then one could be removed from the network.

2.3 Genetic Algorithm Architecture

The genetic algorithm was used to train a 3-layer feed forward architecture with 17 input, 14 hidden and 1 output neuron. Only 14 hidden neurons were used as the performance of the neural network did not vary by much when using other sizes. The genetic algorithm optimised the weights of the neural network and thus, each chromosome was a list of 252 numerical

values in the range [-1, 1]. The population size, number of generations and mutation rate were 100, 50 and 1% respectively as larger values did not have a big impact on performance.

2.3.1 Fitness

The fitness metric for the genetic algorithm was just the accuracy of the neural network on the training data set. This ranged from 0 to 100.

2.3.2 Breeding

Each generation, the top 10% of the population was chosen as the potential pool of parents used for breeding. 50 sets of parents were chosen from this pool at random and bred together using 2-point crossover to create two children. Each gene in the chromosome had a 1% chance to mutate to another value in the range [-1, 1]. These operations helped to prevent the genetic algorithm from stagnating in a local minimum and constantly introduce diversity into the population.

2.4 Evaluating performance

The performance of the neural network was evaluated by creating a confusion matrix to see how often the dry sclerophyll forest species was correct classified. The performance of the network was calculated before and after pruning for 100 neural networks to avoid initial starting weight biases. The averages were compared to see general trends for pruning between the different starting neuron amounts. The same process was used to evaluate the genetic algorithms without pruning.

3 Results

3.1.1 Backpropagation

Several threshold values were tested as done in Milne et al, 1995 on the backpropagation neural networks to see if a difference in performance could be seen. The results in Table 1 and Table 2 are consistent with the results found by Milne et al. (1995). As the threshold value is increased, false negative classifications occur more frequently instead of false positive classifications. This is desirable as when the model says the classification is a dry sclerophyll, this is very likely to be true. This relationship however, is not as strong as the original paper as can be seen if the false positive values are compared when the threshold is set to 0.5.

Table 1 Neural network performance on the training set

Threshold	Correct	False +ve	False -ve	Accuracy
0.4	95	48	10	0.62
0.45	106	33	14	0.69
0.5	108	23	22	0.71
0.55	106	14	33	0.69
0.6	106	7	40	0.69
0.65	96	5	52	0.63
0.7	80	1	72	0.52

Table 2 Neural network performance on the testing set

Threshold	Correct	False +ve	False -ve	Accuracy
0.4	23	11	3	0.62
0.45	26	8	3	0.70
0.5	26	8	3	0.70
0.55	27	5	6	0.70
0.6	28	1	8	0.76

0.65	22	1	14	0.59
0.7	19	0	18	0.51

The overall accuracy of the backpropagation trained neural network is 71% on the training data and 70% on the testing data. This is a considerable 18.4% and 4.3% increase over the 52.6% and 65.7% accuracies found by Milne et al (1995). This increase in performance may be due to the way cross-validation was set up in the original paper as it was not described and thus could not be replicated.

Table 3 below, shows the average number of neurons pruned and the network's accuracy for both training and testing before and after pruning for 100 neural networks. Although pruning the neural network decreases its accuracy, the number of nodes removed from the hidden layer is substantial, ranging from 4 to 20%. This suggests that there are many hidden neurons that are not contributing to the performance of the neural network. The percentage of neurons removed increases as the number of hidden neurons increase. This may be due to the small training data set as more complex or varied data points are not present and thus cannot be captured in larger neural networks. This does however, show that pruning neural networks can improve the computational performance of a network, as neurons are removed, while only slightly reducing its accuracy on this data set.

Table 3 Average accuracy for 100 neural networks before and after pruning

hidden neurons	neurons pruned	pre-prune accuracy		post-prune accuracy		accuracy difference	
		training	testing	training	testing	training	testing
14	0.53	67.91	71.29	68.19	70.42	0.28	-0.87
20	1.77	69.52	73.29	69.40	72.67	-0.12	-0.62
40	5.34	70.30	69.92	71.01	71.05	0.71	1.13
50	9.67	71.16	68.76	72.88	70.86	1.72	2.10

If we compare the performance of the networks that started with 40 and 50 hidden neurons, we can see that after pruning the 50 neuron network we have a similar accuracy to the pre-pruned 40 neuron network. This suggests that you can often build neural networks smaller than they currently are. A reasonable starting size for a neural network might be when approximately 5% of its neurons are removed when pruning. If too many neurons are removed, it may make it difficult to train the neural network on fresh data.

The performance of the genetic algorithm training the neural network can be seen below in Figure 2 and Figure 3. Although the training performance of the neural network is consistently increasing, it does plateau at around 76% accuracy. In Figure 3 below, the legend refers to the testing accuracy of the neural networks that placed first, second and third in the training phase. As can be seen, the testing accuracy of the neural networks vary up to 10% between each generation, however it too generally increases in accuracy across the generations.

It is worth noting, that although a neural network may have the highest training accuracy, that does not mean that it will have the highest testing accuracy. The blue line (first place in training) is constantly overtaken by the grey line (third place in training) in Figure 3. This is due to the training and testing data sets being different, which can result in neural networks capturing different patterns. The fitness function for the last generation should be changed to test the accuracy of the neural network on the testing data set and sorted accordingly to pick the best performers.

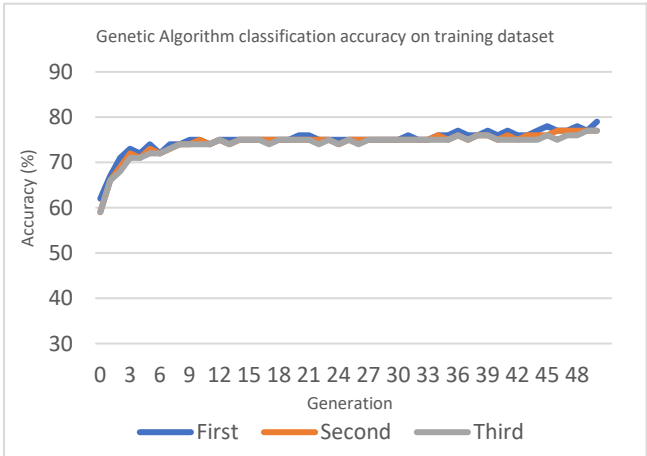

Figure 2 GA classification accuracy on training dataset

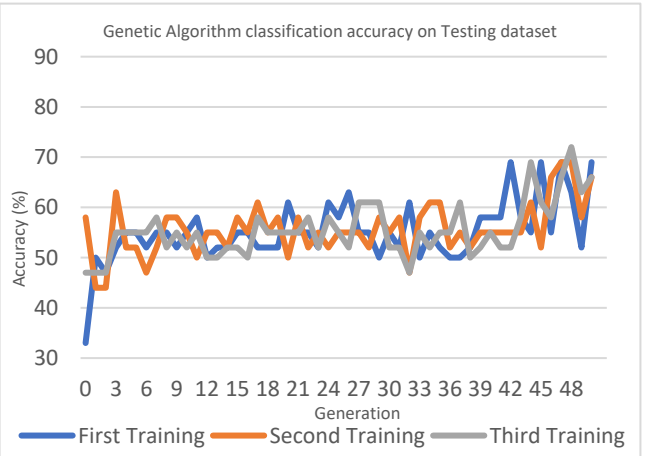

Figure 3 GA classification accuracy on testing dataset

The genetic algorithm was used to train 100 neural networks and the average of the training and testing accuracies were taken. The results of this can be seen in Table 4 below and compared to the backpropagation method pre and post pruning. The genetic algorithm outperformed backpropagation training, both pre and post pruning by more than 7.13% in training and 1.94% in testing. Although a significant difference exists in training, the difference on the testing data set is small. If the GIS dataset has more complex patterns but the sample size is not high enough to capture it, then the genetic algorithm may be able to more clearly out perform the backpropagation with more data points. This result means that when training a neural network model, developers should choose genetic algorithms if they prefer speed otherwise they should be able to choose whichever model is easiest to implement and maintain on this GIS dataset.

Table 4 Comparing accuracy of backpropagation (pre and post pruning) against the genetic algorithm over 100 neural networks

	Accuracy (%)	
Training method	Training	Testing
Backpropagation Pre-Prune	67.91	71.29
Backpropagation Post-Prune	68.19	70.42
Genetic Algorithm	75.32	73.23

4 Conclusion and Future Work

We have found that neural networks trained with backpropagation are, on average, able to perform similarly on the GIS dataset before and after pruning. These neural networks however, could be pruned by as much as 4 to 20% which can lead to considerable computational gains if the size of the neural network is large enough. The genetic algorithm also outperformed the backpropagation algorithm before and after pruning by 7.13% on the training data set and 1.94% on the testing data set. Although significant on the testing data set, this is a small difference on the testing data set.

Working from the findings of this paper, future work could include comparing backpropagation against genetic algorithms on more diverse data sets. This may lend the results to being more generalized and applicable in more problem domains. Pruning the neural networks generated by genetic algorithms and comparing their performance before and after pruning may be able to make them even better. Retraining these neural networks after pruning may also help to define the most important neurons and make them more efficient.

References

1. Gedeon, TD & Harris, D 1991, 'Network Reduction Techniques', *Proceedings International Conference on Neural Networks Methodologies and Applications*, AMSE, vol. 1, pp. 119-126, San Diego.
2. Milne, LK, Gedeon, TD & Skidmore, AK 1995, 'Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood training', *Proceedings 6th Australian Conference on Neural Networks*, pp. 160-163, Sydney, 1995.
3. Erenturk, S & Erenturk K 2007, 'Comparison of genetic algorithm and neural network approaches for the drying process of carrot', Journal of Food Engineering, vol. 78, no. 3, pp. 905-912. doi: 10.1016/j.jfoodeng.2005.11.031
4. Qiao, JF, Zhang, Y & Han, HG 2008, 'Fast unit pruning algorithm for feedforward neural network design', *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 622-627. doi: 10.1016/j.amc.2008.05.049