Neural networks Optimization using Bimodal distribution removal technique and genetic algorithms

Jiawei Wang[1]
Australian National University, Canberra, Australia
u6460277@anu.edu.au

Abstract. Bimodal Distribution Removal (BMR) technique and Genetic Algorithm (GA) was implemented in the training of a simple 2-layer neural network with the oil well dataset collected from the North West Shelf in Western Australian. The hyper-parameters of the neural network model, the BMR and the GA were tuned using trail and error. The tuned model was trained on 3 separate dataset of different oil wells and the confusion matrix of testing results were generated. By comparing the testing result from training with and without BMR, we learned that BMR implementation does not have effects on the accuracy of the model. The tuned model was also used to exam the performance GA feature selection, the results indicate GA can improve model accuracy and reduce training time by removing redundant features in the dataset.

Keywords: Bimodal distribution removal, Neural Network, Genetic Algorithm

1 Introduction

Neural networks have the potential to outperform any other parametric estimators given the condition that it has infinity of data for training [1]. However, it is usually not the case when analyzing data in the real world where the availability of training data is limited. The shortage in the training data usually lead to the trained network model to be very sensitive to the actual realization of the training samples. In another word, the network could struggle to learn the patterns that are not present in the given training data. This fact increases the variance in the network training. One of solution to this problem is to introduce some bias into the network training procedure. Given introducing bias cause its own problem such as network training to converge to the incorrect direction, it is sometimes necessary when the availability of training data is limited.

Popular techniques used to introduce bias into neural network includes neural network pruning techniques [2] and the outlier removal techniques. Neural network pruning removes hidden units in the exist network and reduces network complicity. This procedure simplifies the neural network structure and reduces the calculations that the neural network can perform. The Outlier removal is to remove the outliers in the training dataset. The removal procedure can take place before the training or during the training process. Both methods mentioned above have the potential to improve the bias and variance trade off and we will look at the outlier removal methods closely in the next part.

Popular techniques in outlier removal includes Absolute Criterion Method (ACM), Least Median Squares (LMS). Least Trimmed Squares (LTS) [3] and bimodal distribution removal (BDR). The idea of ACM and LMS is to minimize the absolute or medium error, but both methods are slow to converge. The LTS can increase the rate of convergence, but it has its own limitations such as it ignores the fact that the testing data may contain noise and the number of outliers is unknown in the real-world dataset [4]. The BDR techniques has the potential to address all the limitations mentioned in the above techniques and it is used in the training procedure of this paper.

Bimodal distribution removal (BDR) attempts to remove outliers during the neural network training process. It starts to remove outliers when the calculated the error variance of the training samples by the network is reduced to below certain threshold (normally 0.1). The outlier removal is based on how far the error of the targeted sample is away from another threshold which is calculated in the BDR algorithm. The BDR is designed to address the weakness of other outlier removal methods [5]. The idea of implementing BMR in the training process allows the neural network to identify the outlier dynamically. Also, it removes the outliers slowly so that the network can still have enough time to learn from the data. It also can prevent overfitting and reduce training time.

Another problem which could occur with limited training data is the overfitting on the redundant features in the dataset. When these redundant features outweigh the effective features, the performance of the network can be adversely affected. To ensure that only the effective features in the training dataset are to be learned by the model, feature selection is required. One of the widely used technique for feature selection is the Genetic Algorithm (GA). The GA is a global search algorithm inspired by the natural selection [6]. The algorithm contains operators such as mutation, cross over and selection [7]. A hybrid version GA can also be implemented to select not only the features but the hyper parameters of the neural network model [8].

2 Method

2.1 Oil well dataset

The oil well dataset used in this paper is obtained from the oil wells in the North West Shelf in Western Australian [9]. The data in the dataset was collected from 3 different oil wells. For each oil well, the data contains 10 feature columns and 1 label column. The label column marking the quality of the rock samples are created by experienced geologists. The 10 features are GR (Gamma Ray), RDEV (Deep Resistivity), RMEV (Shallow Resistivity), RXO (Flushed Zone Resistivity), RHOB (Bulk Density), NPHI (Neutron Porosity), PEF (Photoelectric), DT (Sonic Travel Time), PHI (porosity) and logK (permeability). The labels contain 3 different classes. They are Frac, Ok and Good. The labels were used as truth values for training and testing our classifier in this paper.

The oil well dataset was selected to study the effectiveness of bimodal distribution removal technique (BDR) and genetic algorithms (GA) on the training of our neural network classifier. For the study of BDR, this dataset can be an ideal choice as it contains limited number of samples (less than 200 samples for each well). In such a small dataset, Outliers are likely to have a larger impact on the neural network training and therefore their removals can be obvious to detect. For the study of GA, the dataset provides 10 features that can be turned on and off during the training to show the effectiveness of the algorithm.

2.2 Neural network

The design of a neural network plays an important role on its performance. For each problem to solve, the structure of the neural network should be designed to fit the nature of the problem [10]. Hyper parameters of the neural network should also be tuned to address overfitting. In our problem, we used a simple neural network, which consisted of 2 fully connected layers. The number of neurons in the hidden layer was one of the hyper-parameters to be tuned. The output layer had 3 neurons, each of them was corresponding to one of the three classes of rock quality to be classified. Sigmoid activation function was applied on the outputs of the first layer neurons. The cross-entropy was selected as the loss function. During the training, 10-Fold Cross validation technique was used to prevent the model from overfitting. The provided train and test dataset were first combined and then randomly separated into two groups at a ratio of 4:1. The bigger group was used for cross validation training and the smaller group was used for additional testing.

Our strategy to determine the neural network hyper parameters was by trial and error. First, we determined the type of hyper parameters to be tuned. These parameters include the number of hidden neutrons, number of epochs and learning rate and the choice of optimizers between SGD (Stochastic Gradient Descent) and ADAM (Adaptive Moment Estimation). In our strategy, multiple training and testing were conducted with different values of hyper – parameters. The results of the experiment were summarised in Table 1.

Table 1. Test and Validation Accuracies from tuning the neural network Hyper Parameters

		SGD Optimizer			**ADAM Optimizer**		
Num. Epoch	**Num. Hidden Neuron**	**Learning Rate** 0.05	**Learning Rate** 0.1	**Learning Rate** 0.2	**Learning Rate** 0.01	**Learning Rate** 0.05	**Learning Rate** 0.1
		Test and Validation Accuracy (Validation Accuracy/Test Accuracy)					
100	5	0.42/0.48	0.55/0.48	0.66/0.48	0.62/0.78	0.71/0.58	0.68/0.56
	30	0.48/0.45	0.56/0.52	0.66/0.70	0.68/0.69	0.64/0.81	0.67/0.71
	50	0.49/0.48	0.52/0.49	0.64/0.65	0.70/0.61	0.66/0.69	0.66/0.64
300	5	0.57/0.47	0.66/0.55	0.67/0.75	0.7/0.72	0.73/0.61	0.74/0.57
	30	0.57/0.59	0.7/0.7	0.65/0.76	0.70/0.73	0.67/0.70	0.73/0.58
	50	0.69/0.52	0.72/0.7	0.63/0.72	0.68/0.64	0.70/0.68	0.7/0.67
500	5	0.66/0.64	0.63/0.82	0.66/0.70	0.68/0.67	0.6/0.8	0.63/0.64
	30	0.71/0.69	0.69/0.60	0.76/0.63	0.68/0.67	0.7/0.64	0.66/0.72
	50	0.69/0.65	0.65/0.77	0.69/0.62	0.68/0.75	0.67/0.76	0.70/0.68

The result in the table indicated that the best test and validation accuracy can be achieved by training the model using an ADAM optimizer, a learning rate of 0.01, 30 hidden neurons and 300 total training epochs. These obtained hyper-parameters were used in the paper to determine other parameters in the later study.

2.3 Bimodal distribution removal

Bimodal distribution removal (BDR) technique is used to remove possible outliers in the dataset during training. Same as the neural network, the BDR technique also have parameters to be tuned to improve its effectiveness. In our study, trial and error approach was used to optimize the parameters. The parameters of BDR to be investigated are the loss variances range where the BDR starts and finishes, the BDR constant and the BDR frequency (number of epochs to wait before another BDR removal occurs). The results of the experiment were summarised in Table 2.

Table 2. Test and Validation Accuracies from tuning the BDR Hyper Parameters

BDR Frequency	**BDR Constant**	**Loss Variance Range (Start, Finish)**		
		(0.1, 0.01)	(0.6, 0.1)	(0.7, 0.5)
		Test and Validation Accuracy (Validation Accuracy/Test Accuracy)		
10	0.4	0.69/0.65	0.70/0.63	0.67/0.61
	0.8	0.60/0.79	0.72/0.69	0.62/0.72
	0.9	0.65/0.63	0.68/0.69	0.69/0.66
20	0.4	0.70/0.66	0.68/0.69	0.66/0.78
	0.8	0.65/0.69	0.71/0.71	0.68/0.58
	0.9	0.70/0.69	0.66/0.66	0.68/0.62
50	0.4	0.69/0.62	0.69/0.74	0.67/0.65
	0.8	0.63/0.72	0.71/0.75	0.69/0.67
	0.9	0.67/0.65	0.71/0.67	0.72/0.64

The result in the table indicated that the best test and validation accuracy can be achieved by implementing the BDR using a loss variance range of 0.6 and 0.1, a BDR constant of 0.8 and a BDR frequency of 20.

2.4 Genetic Algorithm (GA) for Feature Selection

Genetic algorithm (GA) is search algorithm inspired by natural selection. It is widely used in the feature selection [11]. Training a neural network with many features can be time consuming. In this paper, the effectiveness of using GA on the training of the oil well dataset was studied. The oil well dataset contains 10 feature columns. Each feature column can be turned on and off by the GA.

To implement the GA, we first encoded the 10 feature columns in the dataset as binary strings. A 1 in a position in the binary string means the corresponding feature in that position is turned on and a 0 means the feature is turn off. For example, a string of 1111111111 means all 10 features are turned on and used in the training. Then we defined the fitness function to be a function of testing accuracy and training time. In our case, the feature combination which lead to a higher testing accuracy and a lower training time yields a higher fitness score. The trial and error approach were also used to optimize other parameters for GA. The parameters of GA to be tuned were cross rate, the rate at which the existing feature combinations can exchange their feature representations, the mutation rate, the chance that a certain feature representation can flip to the opposite side and the number of generations. The results of the tuning experiment were summarised in Table 3.

Table 3. Fitness Score from tuning the GA Hyper Parameters

Cross Rate	**Mutation**	**Num. Generations**		
		2	4	6
		Fitness Score		
0.8	0.01	67.6	67.71	65.38
	0.1	67.7	67.72	67.53
	0.5	65.8	69.77	67.68
0.9	0.01	67.44	67.72	65.65
	0.1	67.55	67.81	63
	0.5	67.57	65.3	63.47

The result in the table 3 indicated that the fitness score can be achieved by implementing GA using a cross rate of 0.8, a mutation of 0.5 and 4 generations.

3 Results and Discussion

3.1 Hypothesis 1: BMR can improve the accuracy of the model

Once the optimal hyper parameters of neural network and BDR were determined. The effectiveness of BMR on the neural network performance was examined. The study was carried out on three oil well datasets. The hyper- parameters used in the study is summarized in Table 4 and the test confusion matrix were plotted in Table 5, 6, 7 respectively in for each oil well.

Table 4. Parameters used in the neural network training

Parameters		**Parameters**	
Input size	10	**Learning rate**	0.01
hidden size	30	**Validation split ratio**	0.9
Number of classes	3	**BMR constant** $\propto$	0.8
Number of epochs	300	**BMR variance range**	$0.1 < v < 0.6$
Batch size	*10*		

Table 5. Confusion matrix for the test results on oil well dataset 1

Confusion matrix	**Neural network without BMR**			**Neural network with BMR**		
Rock Quality	**Frac**	**Good**	**OK**	**Frac**	**Good**	**OK**
Frac	14	0	1	14	0	1
Good	0	7	8	2	5	7
OK	1	5	14	1	5	14
Test accuracy	70%			66%		

Table 6. Confusion matrix for the test results on oil well dataset 2

Confusion matrix	**Neural network without BMR**			**Neural network with BMR**		
Rock Quality	**Frac**	**Good**	**OK**	**Frac**	**Good**	**OK**
Frac	10	1	4	10	2	3
Good	0	30	5	0	28	7
OK	2	12	11	2	12	11
Test accuracy	68%			65%		

Table 7. Confusion matrix for the test results on oil well dataset 3

Confusion matrix	**Neural network without BMR**			**Neural network with BMR**		
Rock Quality	**Frac**	**Good**	**OK**	**Frac**	**Good**	**OK**
Frac	6	0	2	6	0	2
Good	2	9	20	2	11	18
OK	0	16	30	0	18	28
Test accuracy	54%			53%		

From Table 5, 6 and 7, we can notice that our trained classifier's performance varies on the rock samples in different oil wells. In oil well 1, the classifier had achieved relatively good accuracy in classifying test samples with OK and Frac qualities but achieved relatively low accuracy in classifying test samples with Good qualities. In oil well 2, the classifier had achieved relatively good accuracy in classifying test samples with Frac and Good qualities but achieved relatively low accuracy in classifying test samples with OK qualities. In oil well 3, the classifier had achieved relatively good accuracy in classifying test samples with Frac qualities but achieved relatively low accuracy in classifying test samples with Good and OK qualities. This may be explained as a result from variance in the training data. As the availability of the training data was very limited, outliers can exist in the training dataset and the training set may not represent all the features from all rock samples but only from the samples that are accounted to generate the dataset.

Compare the results between the network performance after the trainings with and without BMR, we can notice that implementation of BMR had negatively impact the accuracy of the trained model. This phenomenon may be explained by the fact that the samples removed by the BMR from the training set are not the outliers and contain important features that the neural network should learn to improve its accuracy.

3.2 Compare the result with Fuzzy Clustering Classification

Table 8. Comparing the result with Fuzzy Clustering Classification (FCC)

Oil well	FCC	without BDR	BDR
1	70%	70%	66%
2	75%	68%	65%
3	60%	54%	53%

Compare the results between the network performance with Fuzzy Clustering Classification (FCC) [8], we can notice that performance of FCC is better than our results in all wells

3.3 Hypothesis 2: feature selection by GA can improve training time but reduce testing accuracy

The idea of genetic algorithm is to select a subset of important features from all the features in the dataset, it is not hard to imagine that using less features for training will lead to a decrease of training time. However, the saving of training time comes at a cost of model testing accuracy due to loss of information. To test how well the GA can balance this trade off, we measured the test accuracy and training time of genetic sequences that produced by GA while training the oil well data. The training time is time for the neural network to reach an average error of 0.75. The results are summarized in Table 9.

Table 9. Confusion matrix for the test results on oil well dataset 3

Case	Genetic sequence	Test accuracy	Training time (seconds)
1	[1,1,1,1,1,1,1,1,1,1]	68%	1.64
2	[0,1,1,1,1,1,1,1,1,1]	70%	1.62
3	[1 1 0 1 1 1 1 1 1 1]	68%	1.71
4	[1 0 1 1 1 1 1 1 1 1]	70%	1.6
5	[0 1 1 0 1 1 1 1 1 1]	74.0%	2.66

The Table 8 indicated that reducing features did not necessarily guarantee a reduction in training time. In fact, in case 3 and 5, it increased the training time. This can be explained as neural network become harder to converge due to missing important features. But in case 4, missing feature 2 indeed decrease the training time, which means feature 2 was not an important feature for training and therefore was redundant. On the other hand, in case 2, 4 and 5, the testing accuracy was higher than case 1, which indicated reducing features did not necessarily guarantee a reduction in testing accuracy. However, this may be due to fluctuation in the testing accuracy caused by the random shuffles of dataset at beginning of each training

4 Conclusion and Future work

The performance of the trained neural network model is depended on the quality and availability of training data. With unlimited supply of data, the neural network can theatrically outperform any other parametric estimators. However, in the real-world problems, the availability of dataset is limited. Parametric methods should be used to introduce bias to improve the variance and bias trade off. The outlier removal method (BMR) introduced in this paper has the potential to remove the outliers in the training data. It also has the potential to address the limitations in other outlier removal methods such as ACM, LMS, LTS.

Given the factor that the outcomes of the implantation of BMR in this paper has not yield the expected increase in the performance of the neural networks. Some future work is suggested to refine the approach. This first suggestion would be to treat the problem as a regression problem. The rock quality in the training data may contain relationships. For example, we can set the value for frac quality as 0 and set the value for ok as 2. This may correct the way that BMR performs. The second suggestion would be to collect more training data from the same wells. The training data used in this paper is very limited (less than 150 samples per well). This has resulted in large fluctuations in accuracy after trainings. The fluctuation could have offset the effect of BMR.

The genetic algorithm (GA) is an effective tool to for features selection. Our experiment shown that adequate feature selections could lead to a reduction in the training time while not compromise too much on the model accuracy, while an inadequate one would lead to longer training time and lower model accuracy.

References

1. Geman, S, Bienenstock, E and Doursat, R, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, pp. 1-58, 1992.

2. Gedeon, TD and Harris, D, "Network Reduction Techniques," *Proc. Int. Conf. on Neural Networks Methodologies and Applications*, San Diego, vol. 2, pp. 25-34, 1991.

3. Joines, M and White, M, "Improving generalisation by using robust cost functions," *IJCNN*, vol. 3, pp.911-918, Baltimore, 1992.

4. Beliakov, G., Kelarev, A & Yearwood J. (2011). Robust artificial neural networks and outlier detection Technical report. Optimization 61.12 (2012): 1467–1490. Crossref. Web.

5. Slade, P., & Gedeon, T. D. (1993, June). Bimodal distribution removal. In *International Workshop on Artificial Neural Networks* (pp. 249-254). Springer, Berlin, Heidelberg.

6. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison-wesley (1989)

7. Rojas, I., Gonzlez, J., Pomares, H., Merelo, J.J., Castillo, P.A., Romero, G. Statistical analysis of the main parameters involved in the design of a genetic algorithm. In: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 32(1), pp.31-37. (2002)

8. Sharma, N., Gedeon, T. Hybrid Genetic Algorithms for Stress Recognition in Reading. EvoBIO 2013, LNCS 7833, pp. 117–128, 2013.

9. Kuo, H., Gedeon, T. D. & Wong, P. M. (1999). A Clustering Assisted Method for Fuzzy Rule Extraction and Pattern Classification. *ICONIP'99. ANZIIS'99 & ANNES'99 & ACNN'99. 6th International Conference on Neural Information Processing. Proceedings (Cat. No.99EX378)*, Perth, WA, Australia, 1999, pp. 679-684 vol.2, doi: 10.1109/ICONIP.1999.845677.

10. Khoo, S., Gedeon, T.: Generalisation Performance vs. Architecture Variations in Constructive Cascade Networks. In: International Conference on Neural Information Processing (pp. 236-243). Springer, Berlin, Heidelberg. (2008)

11. Leardi, R. (2000), Application of genetic algorithm–PLS for feature selection in spectral data sets. J. Chemometrics, 14: 643-655. doi:10.1002/1099-128X(200009/12)14:5/6<643::AID-CEM621>3.0.CO;2-E