Exploring the impact of Bimodal Distribution Removal and feature selection using Evolutionary Algorithm in Neural Networks

Abhishek Chetri
u667717@anu.edu.au

Research School of Computer Science, Australian National University, Australia

Abstract. The objective of the paper is to analyze the impact of Bimodal Distribution Removal and Feature Selection using Evolutionary Algorithms. Using this method, I was able to remove the patterns which were contributing to high error rates and remove the irrelevant features from the data set. These methods were applied to Oil Well data this resulted in a decrease in accuracy compared to Slade, P., and Tamás D. Gedeon (1993). On the other hand, there was a reduction in training time compared to neural networks with the same structure.

Keywords: Neural Network, Outlier Detection, Outlier Removal, Bimodal Distribution Removal, Evolutionary Algorithm, Feature Selection

1 Introduction

The Oil and Natural Gas sector has huge implications in our day to day lives. Even small changes in the prices of Crude Oil can have impacts on the growth of economies. Tilak Abeysinghe (2001) mentions that oil prices can "play a critical role in small open economies". One of the reasons for increase in oil prices is the production cost. So, one of the ways to keep the oil prices constant will be to use data analysis and neural networks in order to predict indicators for the quality of crude oil.

In this paper, I will be using the data obtained from an oil reservoir in the North West Shelf off shore from Western Australia. The features available are GR (Gamma Ray), RDEV (Deep Resistivity), RMEV (Shallow Resistivity), RXO (Flushed Zone Resistivity), RHOB (Bulk Density), NPHI (Neutron Porosity), PEF (Photoelectric Factor) and DT (Sonic Travel Time). These features will be then use to predict PHI (Porosity) and logK (Permeability) which is a regression task and also be used to predict FLAG (Frac/ Good/ OK) which is a classification task. These predictions impact the quality of Crude Oil as the FLAG value will be able to point if the sample is fractured, or OK or good.

Some work has already been done on this classification task. H Kuo, T.D. Gedeon and P.M. Wong (1999) used Fuzzy Clustering Classification Method (FCC) to predict the FLAG attribute. In the paper, they improved on the previously used Fuzzy Classification Method used by Abe and Lan. The FCC method uses the DOM based on the distance from the vector x from the center of the hyper box. This had resulted in an increase in accuracy by 12% on the same data set. This paper also mentions that neural network need network optimization and large computation time for practical applications.

Many outlier detection methods have been proposed in the past. Joines and White (1992) identified a few of the methods like Absolute Criterion Method, Least Median Squares, Least Trimmed Squares. All these have problems as mentioned by Slade, P., and Tamás D. Gedeon (1993). However, in this report I will be discussing an outlier detection method called Bimodal Distribution Removal and how it can help speed up the training time. These outliers slow down the learning and this increased time can also result in over fitting. The best practice is to remove these patterns from the training dataset after the network has learned after a few epochs. Usually the outlier removal process is run after every 50 epochs which gives time for the network to learn from the data set. I will also compare the results with a neural network without any outlier detection.

Feature selection is the process in which we select only the relevant features and using it to train and test the model. This can be of huge help when there is a very large data set and not all the features are useful for predicting the output..

As described by Oh, Il-Seok, Jin-Seon Lee, and Byung-Ro Moon (2004), there are multiple ways of doing this. But in this paper, I will be focusing on using evolutionary algorithm of select a subset of the data set.

2 The Data Set

The data set contains the details from 3 Oil Wells from the North West Shelf. The data set has been split into training and testing for each of the 3 Oil Wells.

These data sets have the following attributes –

- GR (Gamma Ray)
- RDEV (Deep Resistivity)
- RMEV (Shallow Resistivity)
- RXO (Flushed Zone Resistivity)
- RHOB (Bulk Density)
- NPHI (Neutron Porosity)
- PEF (Photoelectric Factor)
- DT (Sonic Travel Time)
- PHI (Porosity)
- logK (Permeability)
- FLAG (Frac/ Good/ OK)

The data set contains no mission values and the values for all the features in normalized from 0 to 1. Here is a figure showing the normalized data.

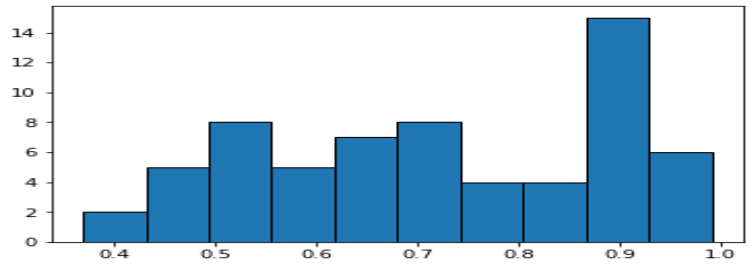

Fig. 1. GR vs Frequency from Oil Well 1

The data set also contains a label column FLAG which contains if the core sample is Fractured or Good or Ok. This will need to be processed later.

3 Data Prepossessing

The data set is very clean with no missing values and all the values are already normalized from 0 to 1. However, the FLAG column was needed to be represented in numerical data. So, for this Frac was converted to 0, Good was converted to 2 and Ok was converted to 1. This will make the prediction easier in the later stages. For predicting logK and Phi no other data manipulation is necessary.

4 Implementation of neural network

There are two tasks at hand. Predicting logK and Phi using regression and prediction FLAG using classification. The neural network contains three layers – input layer, hidden layer and output layer. The input layer contains nodes for

each of the input variable. So, for our dataset it will be the data in the columns 'GR', 'RDEV', 'RMEV', 'RXO', 'RHOB', 'NPHI', 'PEF', 'DT'. This input layer will be connected to the hidden layer which then will be connected to the output layer via a sigmoid / relu function. In this network I will be using the back propagation method to reduce the error. As pointed out by Goh, Anthony TC (1995) it can help achieve low error rates and can be applied to large, real world tasks.

I used the following parameters –

- input_size = 8
- num_classes = 1
- hidden = 50
- learning_rate = 0.01
- num_epochs = 500

The input size is fixed as 8 as there are only 8 attributes which are used to train the network. The num_classes is 1 as there is only 1 output. And the rest of the parameters were fixed after testing. Next is the classification task. In this case, I used similar parameters. But the difference was that the output (num_classes) will be 3 because we will need to predict the FLAG and the number of hidden neurons was 50.

5 Improvement to the neural network using Bimodal Distribution Removal

The main essence of using Bimodal Distribution Removal is to decrease the training time by removing outliers. This process has been discussed in Slade, P., and Tamás D. Gedeon (1993). Outliers are the patterns which have unusually high error rate even after learning from that pattern. Here is an example -

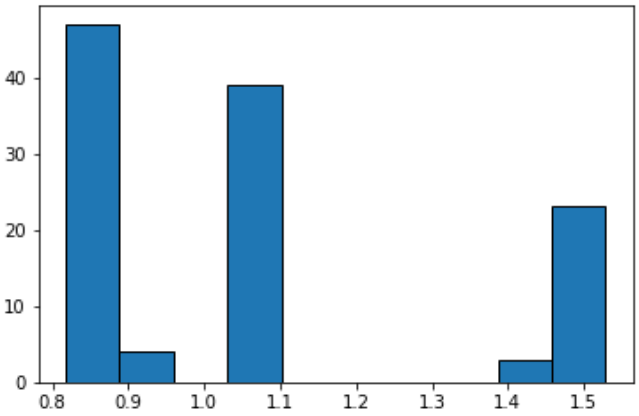

Fig. 2. Error vs Frequency at Epoch 10

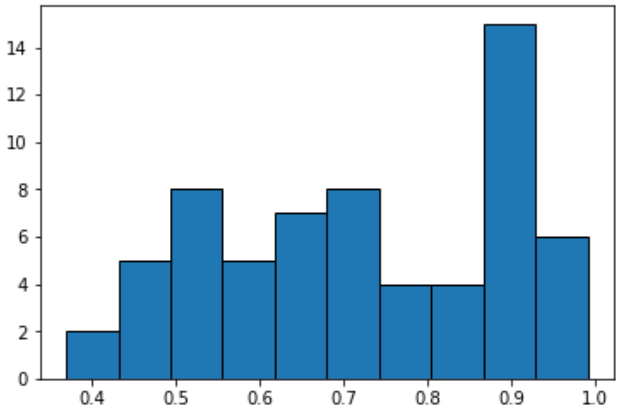

Fig. 3. Error vs Frequency at Epoch 100

Here we can see that even after learning after 100 epochs there are still a few patterns which are producing high error rate. At this epoch, we can see that most of the patterns have been learnt well however the patterns with high errors have not been learnt even after 100 epochs. Due to this the variance of the error is also very high. So, these are considered as outlier and can be removed in order to reduce training time. In order to find the outliers, the first step is to calculate the mean error of all the patterns in certain epoch. The next step is to calculate the error of all the individual pattern in the training set and create a subset of all the patterns with error greater than the mean error of all the patterns.

The next step is to find the mean of the individual patterns of the subset (δ) and the standard deviation (σ) . The total error of all the patterns (E) is also calculated. Now the patterns with

$$E \geq \delta + \sigma\alpha \text{ , where } \alpha \geq 0 \text{ and } \alpha \leq 1$$

are removed from the subset and the rest of the patterns are put back in the training data set.

This process is repeated every 50 epochs, but this could also be a problem if the BDR process runs multiple times it can remove all the patterns from the training set. Every time the BDR method is used, it results in a decrease in δ and also a sharp decrease in σ . So, the variance as pointed out by Slade, P., and Tamás D. Gedeon (1993) can be used as a stopping condition. So, the variance of the data set is calculated after the removal of outliers. If the variance is less than a constant value, the training is stopped. In my model, I have chosen the constant variance as 0.001.

This method has a lot of advantages the patterns removed are data driven, which reduces the need to hard code the outlier removal, patterns are removed slowly which gives the network time to learn from those patterns as well, the stopping condition prevents over fitting and patterns are not removed until the data has outliers.

On applying this method to the neural network, the time spent of the training data decreased as will be discussed in the next sections.

6 Improvement to the neural network by feature selection using Evolutionary Algorithm

The purpose of using feature selection (FS) is to obtain a subset of the data set in order to improve the performance, faster training and to provide a better understanding of the data as pointed out by Isabelle Guyon, and André Elisseeff (2003) . This process can be hugely beneficial in situations where there are a huge number of features to consider. As pointed out by Majid Almaraashi (2018), feature selection helped improve the Root Mean Squared Error (RMSE) loss by 19.9% to 52.1% thus having a positive impact on the evaluation metrics of neural networks.

Evolutionary algorithms as the name suggests is based on the evolution process observed in nature. As mentioned in [6], this process has the following components -

1. Encoding of the chromosome and the fitness function
2. Initialization of initial population
3. Selection of fittest chromosome
4. Reproduction of chromosomes

This algorithm has also been described in Leardi, Riccardo, R. Boggia, and M. Terrile (1992). For the data set used in this paper, there are 8 features which is used to train the neural network, so the size of the chromosome will be 8. The chromosome will be encoded in 0s and 1s. Eg. 00110011, this chromosome will contain the 3^{rd}, 4^{th}, 7^{th} and 8^{th} features will be used for training the neural network. These chromosomes are initialized randomly and then later evolved.

These chromosomes are then used as inputs for the fitness function. The fitness function will assign a fitness to each chromosome which will be used to select for the next generation. If the chromosome is selected, it will be used to create new offsprings which will then be mutated and used as new chromosomes in the environment.

The mutation step is one of the most crucial steps in this algorithm as it helps the new chromosomes develop some new features. This is done by flipping the bits of the selected chromosome. This process will go for for a few

generations and finally the best chromosome at the end will be selected as the best individual.

In my model, after multiple testing, I found out that one point crossover function and tournament selection produced the best results. One point crossover function is modeled after nature where two parents contribute to half of the DNA of the offspring.

7 Results and Discussion

One of the few parameters to be checked was the number of hidden neurons. As the number of data points in the data set was low. Setting the hidden neurons very high would lead to over fitting and very low number would lead to the network not learning properly. Here is a table showing the accuracy with different number of hidden neurons.

Number of Neurons	Accuracy
10	54
50	52
100	46

As 10 hidden gave the best accuracy, I will be using the parameter for the subsequent experiments.

The next parameter that I needed to check is the impact of α to the error percentage and the number of outliers removed. The value for α should remain between 0 and 1 and is used to permanently remove the high error patterns.

Table 1. Table showing the impact of α on the test loss.

α	Test Loss (MSE)	Test Loss (RMSE)
0	0.0272	0.1648
0.1	0.0244	0.1561
0.2	0.0259	0.1609
0.3	0.0283	0.1681
0.4	0.0288	0.1697
0.5	0.0251	0.1584
0.6	0.0243	0.1558
0.7	0.0240	0.1548
0.8	0.0247	0.1571
0.9	0.0242	0.1557
1	0.0245	0.1566

In this case, the best RMS Error was achieved at 0.7. So, I will be using this parameter in the subsequent experiments.

The variance parameter is also very important as it can help in stopping the training process. Slade, P., and Tamás D. Gedeon (1993) mentions that this parameter is typically around 0.01. I tested the optimal value for my network in the following table.

Table 2. Table showing the impact of variance on the removal of outliers

Variance	Outliers Removed
0.1	0
0.01	0

0.001	13

So, for 0.1 and 0.01 there were no outliers removed so the best option will be to consider 0.001 as the fixed variance.

The next parameter, I will be testing is the mutation rate and population size. This is the probability of when will the chromosome will be flipped while the chromosome evolves. A high mutation rate can prevent the population from converging. The population size will be used to create random chromosomes with 0s and 1s for each feature and then use them in the evaluation function. In the next sections, I will be using the population 50 as this provides a broad search area for the size of the data set. A high population means that there is a big area to explore once this has been done the mutation helps in exploiting the best chromosomes from the previous generation.

Table 3. Table showing the impact of mutation rate on accuracy

Mutation rate	Accuracy (%)
0.1	40
0.2	48
0.3	46
0.4	50
0.5	40
0.6	45
0.7	44
0.8	48

In the data set, the best mutation rate would be 0.4 as this produced the best accuracy.

In order to compare time taken to train the network. I will be concentrating on one data set i.e. Oil Well 1 and show the results in predicting logK and FLAG. This will be done for the neural network, neural network with bimodal distribution removal and neural network with bimodal distribution removal and feature selection will compare the time taken to train.

Table 4. Table showing the time elapsed during training

	Regression	Regression	Regression	Classification	Classification	Classification
Runs	Neural Network (in sec.)	Neural Network w/ BDR (in sec.)	Neural Network w/ BDR + FS	Neural Network (in sec.)	Neural Network w/ BDR (in sec.)	Neural Network w/ BDR + FS
Run 1	0.9461	0.3471	0.0400	0.4527	0.1407	0.1196
Run 2	0.7256	0.2912	0.0423	0.4934	0.1319	0.1094
Run 3	0.6658	0.3112	0.0295	0.4876	0.1296	0.1206
Run 4	0.6695	0.2453	0.0299	0.4624	0.1346	0.1220
Run 5	0.6686	0.2075	0.0276	0.4441	0.1236	0.1088
Run 6	0.5247	0.1639	0.0301	0.4518	0.1272	0.1175
Run 7	0.4438	0.1947	0.0432	0.4676	0.1336	0.1149
Run 8	1.0237	0.2054	0.0415	0.5500	0.1678	0.1143
Run 9	0.4288	0.2179	0.0406	0.4678	0.1296	0.1180
Run 10	0.4338	0.3516	0.0311	0.4578	0.1287	0.1250
Average Time	0.6530	0.2536	0.0355	0.4735	0.1347	0.1170

In the table given above we can see that due to the application of Bimodal Distribution Removal there is a decrease of an average time by 0.3995 seconds. This was mostly due to the stopping criterion placed in the training process. Once the error variance was below 0.001 the training process had to be stopped. Even though the training was stopped there was very little impact to the loss rate. On the other hand using BDR along with feature selection had a further decrease in the training time. The time got reduces by 0.6175 on average. The application of feature selection removed the columns which do not impact the output of the result. In fact the application of feature selection improved the accuracy of the classifier which will be shown later. In addition to this decrease in training time, the error rate also decreased dramatically when the Bimodal Distribution Removal and Feature Selection was performed. This is due to the fact the at every 50^{th} epoch. The outliers responsible for high error rates were removed from the training data set and only the features which are relevant. This resulted in a much faster convergence. This can be better represented by the following graphs.

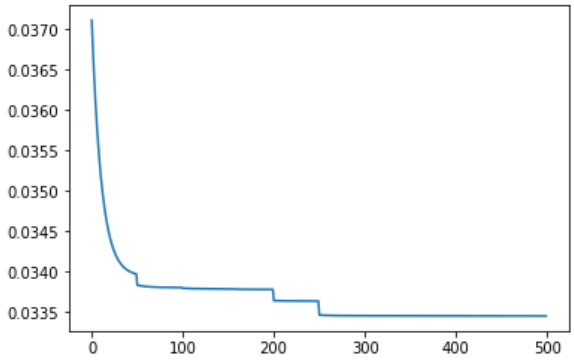

Fig. 4. Error vs Epoch for Oil Well 3 predicting PHI.

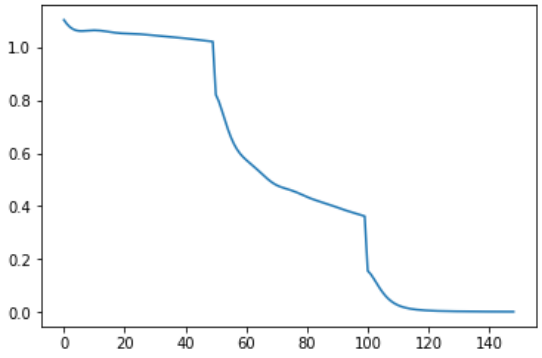

Fig. 5. Error vs Epoch for Oil Well 1 predicting FLAG.

In figure 5, we can see that the training ran for 150 epochs and at 50 and 100 epochs, the error reduced dramatically which is the result of removal of outliers. After the 150^{th} epoch, the variance of the error subset was below the threshold of 0.001 so the training stopped at 150 epochs. So, the network with BDR converged at less epochs. Even though the network converged quicker, the mean squared error rate remained almost the same 0.16136 (with bimodal) and 0.15693 (without bimodal). In addition to this, using feature selection did slightly decrease the loss to 0.15502 (with bimodal and FS). This maybe because some of the columns contribution to the total error may have been removed.

In the next phase, I will be comparing the results obtained from with my own results. The previous paper used Fuzzy Clustering Classification which produced much better results than the results from my model. For Oil Well 1, the best classification accuracy was 70% however for BDR method described in this paper the best accuracy was only 54% and for BDR and feature selection the highest classification accuracy was 58%. For Oil well 2, FCC gave the best accuracy of 75% but BDR produced only an accuracy of 53.33%. Where as BDR and feature selection produced highest accuracy of 58.66%. Again, on Oil Well3, FCC had an accuracy of 60% which is close to the accuracy predicted by BDR and neural network and neural network with BDR and feature selection. On an average, the neural networks on an average did perform worse than FCC. But feature selection did improve the performance by around 5%.

Table 5. Table showing the accuracy the network with, without BDR and BDR+FS

Data set	Neural Network	Neural Network w/ BDR	Neural Network w/ BDR and FS
1	52	50	52
1	52	54	58
2	50.67	53.33	58.66
2	56	53.33	56
3	55.29	54.12	54
3	62.35	57.65	56

On average, I can see that out of the 8 features only five features are selected. After considering five runs, I can see that the columns GR, RDEV, RHOB, PEF and DT are able to produce a higher accuracy rate compared to the simple neural network. Whereas RMEV, RXO and NPHI are being excluded in most runs. This means that these three columns are not essential to the neural network and are hampering the output.

Compared to the FCC, the neural network is not able to produce similar or better results. One of the reasons for this could be that the structure of the neural network is not optimum. One could try adding more neurons and hidden layers in order to get better results. One more way of getting better results is trying to use the BDR and feature selection on a larger data set, this could help the neurons learn better and in turn perform better.

8 Conclusion and Future Work

Although fuzzy based classifiers are fast to train. This paper has demonstrated that a simple neural network can be made faster using the Bimodal Distribution Removal and Feature Selection Evolutionary Algorithm. Here we can see that there was a 0.6 sec improvement in a data set with 116 records. This shows that given a large enough data set BDR and feature selection method can train a neural network much faster and can provide a natural stopping criterion for the training process.

In addition to this, the error rate or the accuracy did have an improvement when feature selection and BDR was applied. This can to be improved by experimenting using numbers of hidden layers or by implementing cascade neural networks. This could result in better error rate and accurate. Furthermore, there was a noticeable decrease in accuracy compared to FCC. This could be since the previous paper used 11 inputs instead of 8 which was used here. This in turn resulted in lower quantity of patterns for the network to learn from.

References

Abeysinghe, Tilak. "Estimation of direct and indirect impact of oil price on growth." *Economics letters* 73.2 (2001): 147-153.

"What Causes Oil Prices To Fluctuate?". *Investopedia*, 2019, https://www.investopedia.com/ask/answers/012715/what-causes-oil-prices-fluctuate.asp. Accessed 3 June 2019.

Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." *Journal of machine learning research* 3.Mar (2003): 1157-1182.

Almaraashi, Majid. "Investigating the impact of feature selection on the prediction of solar radiation in different locations in Saudi Arabia." *Applied Soft Computing* 66 (2018): 250-263.

Engelbrecht, Andries P. *Computational Intelligence: An Introduction*. John Wiley, 2007.

Leardi, Riccardo, R. Boggia, and M. Terrile. "Genetic algorithms as a strategy for feature selection." *Journal of chemometrics* 6.5 (1992): 267-281.

Oh, Il-Seok, Jin-Seon Lee, and Byung-Ro Moon. "Hybrid genetic algorithms for feature selection." *IEEE Transactions on pattern analysis and machine intelligence* 26.11 (2004): 1424-1437.

Kuo, H., T. D. Gedeon, and P. M. Wong. "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). Vol. 2. IEEE, 1999.

Slade, P., and Tamás D. Gedeon. "Bimodal distribution removal." International Workshop on Artificial Neural Networks. Springer, Berlin, Heidelberg, 1993.

Goh, Anthony TC. "Back-propagation neural networks for modeling complex systems." Artificial Intelligence in Engineering 9.3 pp. 143-151, 1995

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