Neural Processing Letters **10:** 7–15, 1999.

A Pattern Adaptive Technique to Handle Data Quality Variation

P. M. WONG[1,⋆] and T. D. GEDEON[2]
[1] *School of Petroleum Engineering, The University of New South Wales, Sydney, NSW 2052, Australia, e-mail: pm.wong@unsw.edu.au;* [2] *School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia*

Abstract. This study proposes a new technique, namely the Pattern Adaptive Neural Network (PANN), for simplifying existing noise detection and removal methods. This technique is developed based on a modified backpropagation algorithm using a fuzzy membership function on the error term. It is able to make use of noisy data in a single step, with an automatic adjustment of data contribution to network training. It is demonstrated via an application on an oil well data set. The results show that the predictions from PANN matched well with the expert interpretations on the data set regarding the data quality.

Key words: well logging, outliers, permeability

1. Introduction

Neural networks are useful in solving complex, nonlinear classification and mapping problems. These networks, particular supervised networks, rely strongly on the quality of the training data (which are input data paired with target data). In practice, data corruption is unavoidable, and noisy data is usually present in the sampled data set. Therefore, a large quantity of data for network training does not necessary result in better generalisation. What is important is the quality of the data. Hence, there is a need to filter out the noisy data and make neural networks a more viable tool to solve practical problems.

The term ‘noisy data’ is commonly used to mean situations similar to taking a clean input and output set and superimposing a Gaussian white noise which is readily learnt by neural networks. In the context of this paper, the input and output values all have the same degree of correctness prior to addition of noise. In the petroleum engineering domain, there are inherent measurement errors which are due to the limitations of the measurement tools and the discontinuous nature of the real properties being measured. That is, the input and output properties can only be measured at different scales, producing measurement discontinuities. Then, the

⋆ Author for correspondence

usual sampling errors provide additional white noise. This low correlation between input and output spaces is difficult for neural networks to learn.

Inclusion of noisy data for conventional neural network training has two effects on prediction performance: (1) to slow down the learning of the clean or representative data; (2) to over-fit the noisy or unrepresentative data. Thus, selection of clean data set for training is an important step in the use of neural networks. From our experience, this step requires a significant amount of man-hours to examine each and every single data point in order to construct such a clean data set. This is not economically viable in most engineering applications.

In the past years, we developed various techniques to detect and remove noisy data embedded in the data set using backpropagation neural networks [1, 2]. The techniques were also successfully applied to a petroleum engineering problem [3]. The techniques developed were based on a two-step process: (1) to use a crisp statistical property to define (or detect) noisy data; (2) to completely discard the noisy data and use the remaining (clean) data for training.

The objective of this paper is to simplify the two-step process to a single, one-step process. We will first briefly review the two previous techniques in noisy data detection and removal, namely Bimodal Distribution Removal and Error Sign Testing. In the later sections, we will introduce the concept of our technique in pattern adaptive networks using a modified backpropagation algorithm, together with the concept of fuzzy logic. Our technique does not require a crisp definition of noisy data and no data is discarded during training. The proposed technique is also demonstrated with an application to an oil well data set in the Asian region.

2. Assumptions

In this paper, we will assume a multi-layer feed-forward network trained using backpropagation and will use the general expression ‘neural network’ to mean such a network. All connections are from neurons in one level to neurons in the next level, with no lateral, backward or multi-layer connections. Each neuron is connected to each neuron in the preceding layer by a simple weighted link. The network is trained using a training set of input patterns with desired outputs, using backpropagation of error measures. By backpropagation we mean the general concept of developing the error gradient with respect to the weight, and not restricted to the original gradient method. In the examples we used here, we have used the basic sigmoid logistic function, $y = 1/1 + \mathrm{e}^{-x}$, though this is not essential to the substance of our results.

3. Detection and Removal of Noisy Data Revisited

Slade and Gedeon [1] introduced the Bimodal Distribution Removal (BDR) method to detect and remove noisy data. It is based on the frequency distribution of the errors for all patterns in the training set at intervals of a certain number of epochs.

An almost bimodal error distribution, with the low error peak containing patterns the network has learnt well and the high error peak containing the noisy data, is expected. The noisy data are defined according to the mean and standard deviation of the error distribution. The detected noisy data are then discarded from the training set and the whole process repeats until the variance of the error distribution is below a tolerance value.

Wong and Gedeon [2] subsequently introduced another noisy data detection and removal technique. The method is known as Error Sign Testing (EST). It is based on the change of error during training. In successful learning of a good pattern, the error in nth epoch, say E_n, should be smaller than the previous one, E_{n-1}. Therefore, counting the number of negative signs of the expression $(E_n - E_{n-1})$ can be used to define a noisy pattern. The noisy patterns are removed and further training is performed using a reduced pattern set size.

In both the BDR and EST methods, a crisp definition of noisy data is used, and further training is performed by discarding the identified noisy data.

4. Pattern Adaptive Neural Networks

In conventional neural networks, we generally assume that all the training data contribute equally to the learning algorithm (backpropagation). In BDR and EST, we assume that the detected noisy data should contribute nothing to the learning algorithm and these data are discarded for further training. All these methods define data using a crisp approach, that is, either 'clean' or 'noisy'. In reality, however, it is difficult to define the quality of data with certainty. Hence, fuzzy logic was employed in our proposed method.

Using fuzzy logic, each data can be both 'clean' and 'noisy' with various degrees of membership. The new technique, Pattern Adaptive Neural Network (PANN), is based on the use of a modified backpropagation (BP) algorithm with a fuzzy membership function for the error term (i.e., target value minus calculated output). Conventional BP makes use of the following expression to propagate the error from the output layer to the hidden layer:

$$W_{jk}(t+1) = W_{jk}(t) + \beta \sum_{p}^{n} (T_{kp} - Y_{kp}) Y_{kp} (1 - Y_{kp}) Z_{jp},$$

where $W_{jk}(t+1)$ and $W_{jk}(t)$ are the new and old weights connecting hidden neuron j and output neuron k, respectively, β is the learning rate, T_{kp} and Y_{kp} are the target value and the calculated output at output neuron k for pattern p (n of them), respectively, and Z_{jp} is the output value at hidden neuron j for pattern p.

In the proposed PANN method, the above equation is modified with the inclusion of a fuzzy membership function for the (absolute) error:

$$W_{jk}(t+1) = W_{jk}(t) + \beta \sum_{p}^{n} (T_{kp} - Y_{kp}) Y_{kp} (1 - Y_{kp}) Z_{jp} F(|T_{kp} - Y_{kp}|)$$

and,

$$F(X) = f_x = \exp\left(-\frac{X}{C}\right),$$

where F is a fuzzy membership function of Gaussian type (f ranges from 0 to 1) for the absolute error value X and C is a user-definable control constant for the exponential function.

Note that other membership function can also be used. The f value is basically a scaled error value. It is important to note that when C is large (say, greater than 10), the f value will be close to one. Hence, the inclusion of membership function has no effect on the learning algorithm. This is equivalent to the conventional BP algorithm.

The fuzzy membership function F calculates the degree of quality for each pattern. If the error of a pattern is large, its membership of the quality will be low, and vice versa. The membership function value of each pattern varies during training. In this way, there is no hard boundary to separate 'clean' and 'noisy' data, except at the extreme cases.

From the definition of the membership function, a measure of average contribution R of each pattern can also be defined:

$$R = \frac{\sum_p^n \exp\left(-\frac{X_p}{C}\right)}{n}.$$

We could compare two data sets with the same value of C and use R as a measure of cleanness: if all patterns are mostly clean (small X_p), the R value will be high, and vice versa. Note that in the conventional BP algorithm (i.e., large C), the R value will be equal to one (see later sections).

5. Benchmarking Tests

In order to understand the proposed methodology, a benchmark test was carried out using a cancer data set obtained from the Proben1 benchmark collection at CMU [4]. The training data consisted of 525 data points with 9 inputs and two outputs (two classes). There are another 174 test patterns. We first trained a neural net with a large C value (100) for 5,000 epochs, and the trained network was applied to the test patterns. From this analysis, we obtained 99.2% and 98.9% classification accuracy in the training and test set, respectively. When training with a smaller C value (1), the corresponding results were 97.7% and 98.3%. Since we obtained essentially identical results irrespectively of the C value used, we concluded that the data set was quite clean, in the sense of not including discontinuities such as encountered in geological data sets.

From the above analysis, we cannot demonstrate the usefulness of the proposed method because it is not aimed to apply to clean data sets or data with simple

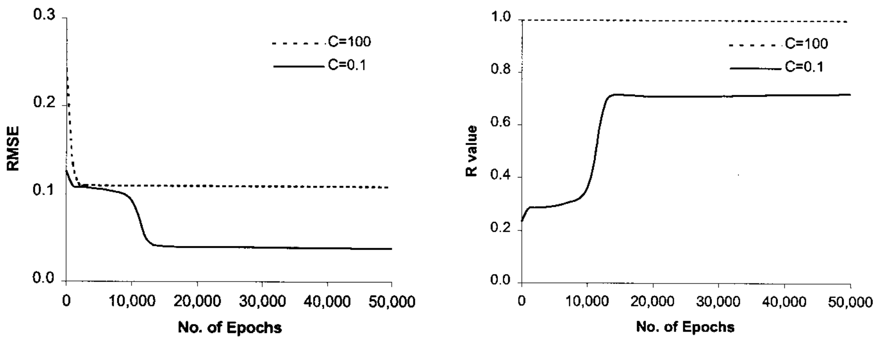


Figure 1. The learning performance of the synthetic data set.

normally distributed noise. In a typical geological data set, it is highly corrupted by complex discontinuities.

We now present a synthetic example to demonstrate the functionality of the method. In the synthetic example, we used an exponential function $y = \exp(-2x)$ to generate 10 sample points starting from $x = 0.1$ to 1.0 at a 0.1 interval. We also created two additional points at $x = 0.35$ and 0.85, with a noise component of 0.3 added together with the corresponding y values. These were equivalent to 0.80 and 0.48, respectively. We used the proposed neural network to learn the 12 data points (10 clean plus 2 noisy data). Two cases were run using two function constant C values: 100 (similar to conventional BP algorithm) and 0.1.

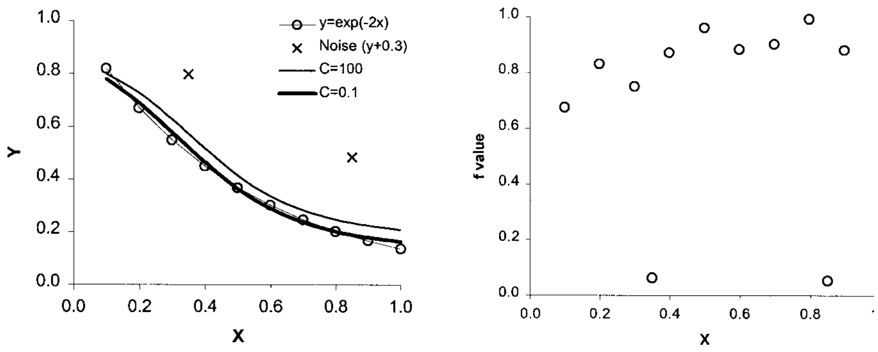
In both cases, the same network configuration was used. The learning and momentum constants were both set to 0.1. One input, two hidden and one output neurons were used. Both networks were trained for 50,000 epochs. Note that this study was aimed to incorporate data with various quality, and hence no issues of model validation and testing are discussed.

Figure 1 displays the learning profiles for the cases with the two different values of the control constant for the fuzzy membership function. In Figure 1(a), the Y-axis shows the root mean square error (RMSE) of the system which is defined as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_p^n (T_p - Y_p)^2 F(|T_p - Y_p|)}{n}},$$

where the function F has the same definition as before.

Figure 1(a) shows that the $C = 0.1$ case gave smaller error compared to the $C = 100$ case. The minimum error values for the $C = 0.1$ and $C = 100$ cases were 0.038 and 0.109, respectively. In the context of this paper, a smaller C value means to put a smaller weight on a pattern compared to the case with a larger C value. This means, a smaller C value will give rise to a smaller R value. Figure 1(b) displays a plot of the R values versus the number of epochs. In the $C = 100$ case, it was a flat line with a R value of almost one. This was because all the patterns contributed



a. Approximated functions. b. The f value for each training data.

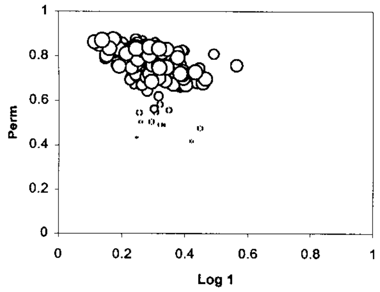
Figure 2. Approximated functions and the f values.

nearly the same amount (about 1.0) to the network. As discussed previously, this is equivalent to the conventional BP algorithm. In the $C = 0.1$ case, however, the average contribution of each pattern was increased from about 0.3 to 0.7, especially from 10,000 to 13,000 epochs. This was because the neural network has learnt most patterns and their f values changed from low to high.

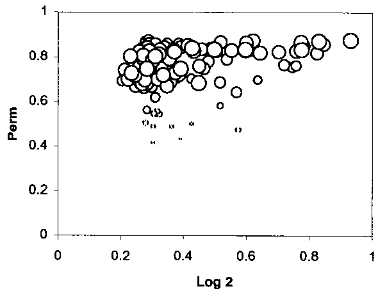
Figure 2 displays the resulting functions from the two networks. In Figure 2(a), it shows that the approximated function from the $C = 0.1$ case fitted well with the original function, even when the two noisy data were included in the training set. The $C = 100$ case, however, was affected by the noisy data and the approximated function swayed towards the noise. Figure 2(b) displays the contribution of the training data in a form of the fuzzy membership function f values for the $C = 0.1$ case results. Note that the f values for the $C = 100$ case were not shown because all the f values were close to one. In this figure, the high quality data are shown by the high f values, and vice versa. This plot was, in fact, consistent with the quality of the training data we knew beforehand. The two noisy data were shown with low f values and our PANN model automatically filtered out their contributions to the final approximated function.

6. An Oil Field Example

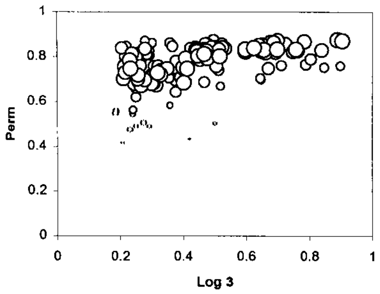
The proposed technique was applied to a data set from an oil well located in the Asian region. The data set consisted of a number of data points at different well depths. At each depth, seven different types of information or ‘well logs’ about the electrical, acoustic, nuclear and other physical properties of the formation were recorded. For simplicity purposes, we named the logs as ‘Log 1’, ‘Log 2’ to ‘Log 7’. Rock samples were retrieved at the corresponding depths and a petrophysical property, namely permeability (or simply ‘Perm’) was measured for each rock sample in the laboratory. Permeability is a crucial parameter in reservoir evaluation as it measures the ability of fluid conductance in sedimentary rocks.



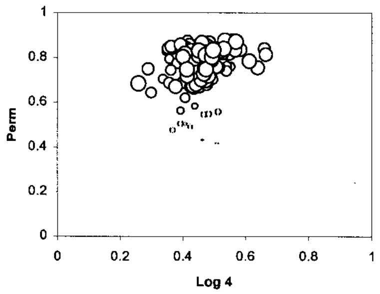
a. Perm vs Log 1



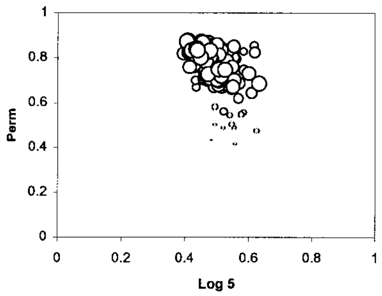
b. Perm vs Log 2



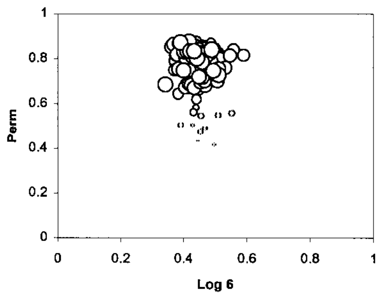
c. Perm vs Log 3



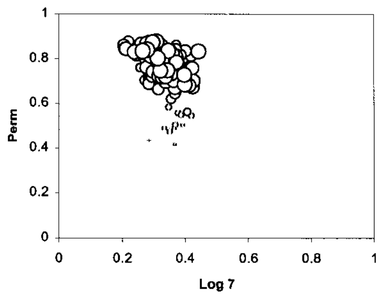
d. Perm vs Log 4



e. Perm vs Log 5



f. Perm vs Log 6



g. Perm vs Log 7

Figure 3. Input and output data cross-plots. The size of the circles is directly proportional to the f values.

In petroleum reservoir engineering, rock extraction and laboratory measurement are expensive, and hence permeability data is relatively few compared to well logs. It is therefore important to develop functional relations to convert well logs (independent variables) to permeability (dependent variable). In most cases, the correlation between well logs and permeability (at the same depth) are complex and highly corrupted by noise. Neural networks are popular in this problem domain and some recent applications can be found in Wong et al. [3], Huang and others [5] and Mohaghegh et al. [6].

In the example study, the seven logs were used as input data and the permeability values were used as the target data. A total of 138 pairs of data was available as the training data. The C value was 0.1. The learning and momentum constants were both set to 0.1. Seven input neurons, four hidden neurons and one output neuron were used. The network was trained for 10,000 epochs. The final R value was 0.6.

After training, an expert geologist closely examined the results and it was confirmed that the points with low f values were relatively unreliable and should not be considered equally in the network. Geologically speaking, data located in homogeneous rocks are considered as high quality (high f) and those located in heterogeneous rocks are of low quality (low f). The f values obtained were consistent with the geological interpretation of the reservoir.

Figure 3 displays the cross-plots of the seven input logs and permeability. For visualisation purpose, the size of the circles is directly proportional to the f values, that is, the larger the circle, the larger the f value and the better the quality. The plots clearly show that low quality points tend to form a separate cluster away from the main population. Hence, the PANN model successfully trained a network to automatically discount the contribution of unreliable data in a single, one step process.

7. Conclusions

A new technique, namely Pattern Adaptive Neural Network (PANN), can be used to train noisy data in a fast and simple manner. The proposed method was applied to both a synthetic data set and an oil well data set in the Asian region. Based on the results obtained from this study, the following conclusions can be drawn:

(1) Sampled data in a noisy environment should not be treated equally.
(2) PANN gives a measure of average contribution of pattern during the training stage.
(3) PANN provides a single, one step method to train data with various quality.

Based on the results from the present study, we will examine the sensitivity to the initial weights of our PANN technique and selection of the values of the control constant for the fuzzy membership function. We also plan to apply similar methodology to estimate the quality (uncertainty) of predictions where actual data are not known.

References

1. Slade, P. and Gedeon, T.D.: Bimodal distribution removal, In: J. Mira, J. Cabestany and A. Priesto (eds), *A New Trend in Neural Computation, Lecture Notes in Computer Science*, Vol. 686, Springer-Verlag, Germany, 1993, pp. 249–254.
2. Wong, P.M. and Gedeon, T.D.: A new method to detect and remove outliers in noisy data using neural networks: error sign testing, *International Journal of Systems Research and Information Science* **7** (1995), 55–65.
3. Wong, P.M., Henderson, D. and Brooks, L.: Permeability determination using neural networks in the Ravva Field, offshore India, *SPE Reservoir Evaluation and Engineering* **1** (1998), 99–104.
4. Prechelt, L.: PROBEN1 – A set of benchmarks and benchmarking rules for neural network training algorithms, Technical Report 21/94, University of Karlsruhe, 1994.
5. Huang, Z., Shimeld, J., Williamson, M. and Katsube, J.: Permeability prediction with artificial neural network modelling in the Venture Gas Field, offshore Eastern Canada, *Geophysics* **61** (1996), 422–436.
6. Mohaghegh, S., Ameri, S. and Arefi, R.: Virtual measurement of heterogeneous formation permeability using geological well log responses, *Log Analyst* (March–April) (1996), 32–39.