A Neural-Fuzzy Technique for Interpolating Spatial Data via the Use of Learning Curve

P.M. Wong[1], K.W. Wong[2], C.C. Fung[2] and T.D. Gedeon[3]

[1]Centre for Petroleum Engineering, The University of New South Wales, Sydney 2052, Australia

[2]School of Electrical and Computer Engineering, Curtin University of Technology, GPO Box U1987, Perth 6001, Australia

[3]School of Computer Science and Engineering, The University of New South Wales, Sydney 2052, Australia

ABSTRACT

In this paper, we present a new and simple method for function interpolation based on the use of neural networks and fuzzy logic. We particularly discuss the application of the technique in spatial data analysis. In this application domain, conventional early-stopping criteria to avoid over-training in neural networks based on the use of minimum error on validation set, may not be suitable. In the proposed method, we use "interpolated error" to stop training. The trained networks are used as fuzzy rules, and these rules are interpolated to the location of interest. We demonstrate the methodology in petroleum reservoir modelling in which properties are estimated between two oil wells. Data from a third well, which are withheld from the training process, is used to evaluate different prediction models. We also compare our method with the recent data-splitting approach using self-organising map (SOM) with the use of early-stopping in neural training. The results of this study show that the SOM approach is only applicable to wells in which their locations are half-way between the two given wells. The proposed methodology, however, provides the best results in the test well and is also suitable for any location of interest. The end result is a simple and computationally-cheap method in engineering studies.

Keywords: Neural, Fuzzy, Interpolation, Spatial Data, Well Logs.

INTRODUCTION

Function approximation and interpolation are important in most engineering disciplines. The former involves the fitting a curve or a surface through existing data points. These data points are usually clustered. One example of function approximation method is linear regression in which the total error of predictions is minimised. The later is similar to function approximation, however, the available data points are more scattered and separated in space (e.g. distance). One example is distance-weighted interpolation method in which the weighting factors of samples are calculated based on the location of the point-of-interest.

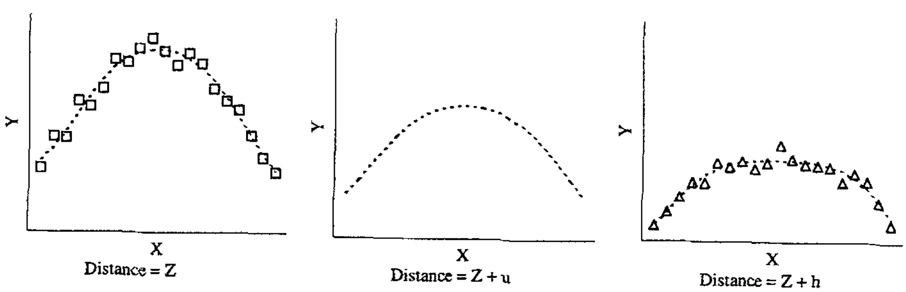


Figure 1. Example for function approximation and interpolation.

Figure 1 shows an example for the need to approximate and interpolate functions. In this figure, we have two sets of noisy data measured at two sample locations, **Z** and **Z+h**, where **Z** is the distance between the first sample location and a reference location, and **h** is the separation distance between the two sample locations. The notion of function approximation is to fit two noise-free curves through the data points. This is done to ensure that we can obtain function values, **Y** (the dependent variable), for any **X** values (the independent variable).

In some application domains, there is also a need to predict function values at location **Z+u** where $0<\mathbf{u}<\mathbf{h}$ if the measurable **X** values are available at that location. This is the purpose of function interpolation. If, however, $\mathbf{u}<0$ or $\mathbf{u}>\mathbf{h}$, this becomes an extrapolation problem. There is usually a larger uncertainty in extrapolating functions than interpolating functions, especially when $\mathbf{u}\ll 0$ or $\mathbf{u}\gg\mathbf{h}$.

In this paper, we are proposing a methodology to achieve these tasks by the use of neural-fuzzy technique. The next section will revisit pattern selection criteria in neural networks, followed by the introduction of our new methodology. In the later sections, we will demonstrate the use of this technique in petroleum reservoir modelling.

PATTERN SELECTION

Choosing proper training and validation patterns is crucial in neural learning. Wong et al. [1] has recently presented a systematic way to select training and validation patterns using self-organising map (SOM). SOM is first applied to classify the available data. Then the training and validation data are selected from each class. This is to ensure that the training and validation sets are generalised to cover the entire data range. The results showed that the SOM method in data-splitting consistently provided good generalised networks and the amount of training time was dramatically reduced.

While the above technique is applicable in most situations, it may not be appropriate in spatial function interpolation, especially when there is a continuity between the functions approximated at locations **Z** and **Z+h**. Under this situation, if **u** is close to 0, one may expect that its function relation should be similar to that at location **Z**. One solution to this problem is to construct a training set based on the relative proportions of patterns at locations **Z** and **Z+h**. The disadvantage of this method is that it is a computationally-expensive process as we need to run these analyses for a large number of times in order to avoid bias introduced to the estimates.

NEURAL-FUZZY TECHNIQUE

Some related works have been done using neural nets to interpolate spatial data in both mining and petroleum industries [2,3], however only areal data maps are produced and there is no indications on how to extend the methodology to vertical data interpolation. The proposed methodology is appropriate for interpolating vertical measurements. It is developed based on the use of learning curve in neural networks and fuzzy rules interpolation [4,5]. It consists of three steps: 1) pattern selection for neural learning; 2) neural learning to extract fuzzy rules; 3) interpolating the results of fuzzy rules to obtain final estimate.

1) Pattern Selection

In the proposed technique, the selection of training and validation sets is straightforward. All the patterns at location **Z** are used for the training set, and all the patterns at location **Z+h** are used for the validation set. The statistics of these two sets may not necessarily the same, but they need to be correlated.

2) Neural Learning

The purpose of this step is to approximate functions at the sample locations. Standard BPNN procedure is utilised. Most neural nets avoid over-training by stopping iteration at minimum error on the validation set (see Figure 2a). The proposed technique uses minimum "interpolated error" or IErr to stop training. This error is defined as:

$$\mathrm{IErr} = \frac{\sum_{j}^{n} W_j E_j}{\sum_{j}^{n} W_j} \qquad \ldots(1)$$

and,

$$W_j = \frac{1}{D_j} \qquad \ldots(2)$$

where W_j is the weighting factor of the j^{th} data set, D_j is the separation distance between the point-of-interest and j^{th} sample location. In the case shown in Figure 1 (n=2), D_1 and D_2 are simply **u** and (**h**-**u**) respectively. E_j represents the error on the learning curve using j^{th} data set for training (and the rest for validation). Thus, when **u** equals 0, this is equivalent to over-learning of the training set, and when **u** equals **h**, this is equivalent to the standard early-stopping method in BPNN (i.e. stop training when minimum validation error is reached). In cases where $0<\mathbf{u}<\mathbf{h}$, the interpolated error curve is between the validation and training errors as displayed in Figure 2b. When the minimum interpolated error is reached, training is stopped.

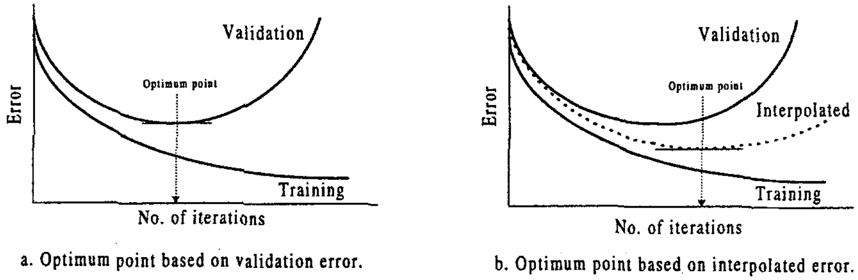


Figure 2. Stopping criteria for BPNN learning.

3) Rules Interpolation

Once each data set has been used as the training set, we will have **n** trained neural nets. These networks are used as fuzzy rules. These are disjoint rules in a large sparse rule-base. Each of these rules can be expressed as:

$$\text{Rule}_j\text{: } \mathbf{IF}\ X_j = (x_1,\ldots, x_m) \text{ is } A_j\ \mathbf{THEN}\ Y_j = NN_j(x_1,\ldots,x_m) \qquad \ldots(3)$$

where **m** is the dimension of the input vector **X**, A_j is the fuzzy set of the j^{th} partitioned rule space, and Y_j is the output from j^{th} trained neural net or NN_j.

The conceptualisation of the use of neural nets as fuzzy rules allows us to use the developed formal machinery of fuzzy set theory such as fuzzy interpolation and defuzzification. We can then interpolate the relevant rules to location **u**, and only recombine the effects of the interpolated rules subsequently. The interpolated estimates can be expressed as:

$$Y_u = \frac{\sum_{j}^{n} W_j Y_j}{\sum_{j}^{n} W_j} \qquad \ldots(4)$$

where Y_u is the final estimate at location **u**.

In the case displayed in Figure 1, if **u** equals 0, $\mathbf{Y_u}$ is the result of the fuzzy rule using data at location **Z** as the training patterns. Similarly, if **u** equals **h**, $\mathbf{Y_u}$ is the result of the fuzzy rule using data at location **Z+h** as the training patterns. Since data set is trained independently at each location and if more data become available at new sample locations, it is straightforward to incorporate the new information in the existing system and there is no need to re-learn all the data.

CASE EXAMPLE

1) Background of petroleum reservoir modelling

The proposed technique was applied to petroleum reservoir modelling. In reservoir modelling, we often drill only a few holes at different locations. Reservoir properties are measured at the laboratory from the rock samples (or cores) retrieved at different depths. One of these properties is permeability, which is a measure of fluid conductivity of the rock sample. Unlike "well-logs", a series of digital measurements at different depths, which are available in every hole, retrieving rock samples (or "coring") is not a routine process because it is an expensive activity, and hence permeability values will be available at every hole. Thus, estimation of permeability has to be relied on well-logs alone [6].

2) Data description

In this example, we had data from two oil wells, namely Well 1 and Well 2. They are separated by approximately 3 km. The data was obtained from the North West Shelf, offshore Australia. Wells 1 and 2 recorded 222 and 166 data respectively at various depths, together with the corresponding permeability measurements. Each data point consisted 5 well-log measurements, namely gamma ray (GR), neutron porosity (NPHI), bulk density (RHOB), photoelectric adsorption index (PEF) and sonic travel times (DT).

3) Objective

The objective of this example is to predict the permeability values of another well, namely Well 3, using only well-logs. This well is located between Well 1 and Well 2. It is 2 km away from Well 1 and 1 km away from Well 2. It had 174 measurements of the same five well-logs as in Well 1 and Well 2. We will first split the data according to SOM method described in Wong et al. [1]. Then we will use the standard BPNN to predict the permeability values at Well 3 from its well-log data. We will compare the predictions with those obtained from the proposed neural-fuzzy technique. Actual permeability values are available at this well for performance evaluation.

4) Network setup

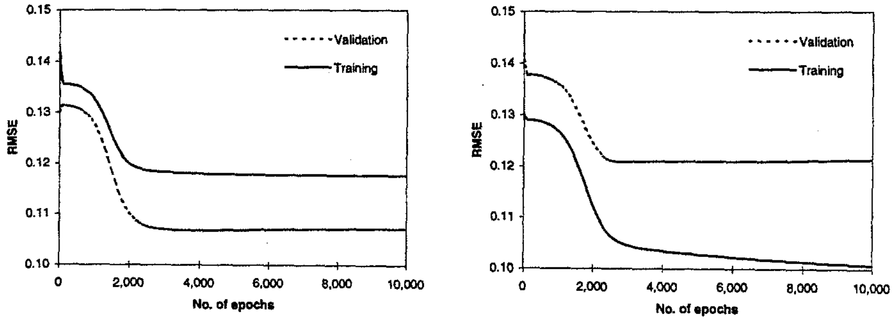
Data from Well 1 and 2 are first combined together. These data are then classified by the SOM approach. The SOM matrix is a two-dimensional 14-by-14 network that gives a total of 196 classes. Splitting of data is made in such a way that the two sets of data are selected from each class. The number of data in each sets, namely Set A and Set B, were 226 and 162, respectively. We will first use the data from Set A for training and the data from Set B for validation. We will then swap the use of the data (i.e. Set B for training, Set A for validation). Final estimates are obtained by averaging the results from the two trained networks.

For the proposed technique, we will use the data from Well 1 for training and the data from Well 2 for validation. We then swapped the use of the data as before. We will predict the permeability values by first assuming that the location of Well 3 is half-way between Well 1 and Well 2 (1.5 km away from each well). Then we will re-run the analyses using the actual separation distances.

All the well-logs (the independent variables) were all normalised in the range of (0,1), while the permeability values (the dependent variable) were scaled in the range of (0.1,0.9). All the BPNN models were based on the same configuration. It had 1 hidden layer with 5 hidden neurons, the learning rate and the momentum term were both 0.01. Sigmoid functions were used as the transfer functions. The maximum allowable epoch was 10,000.

5) Results

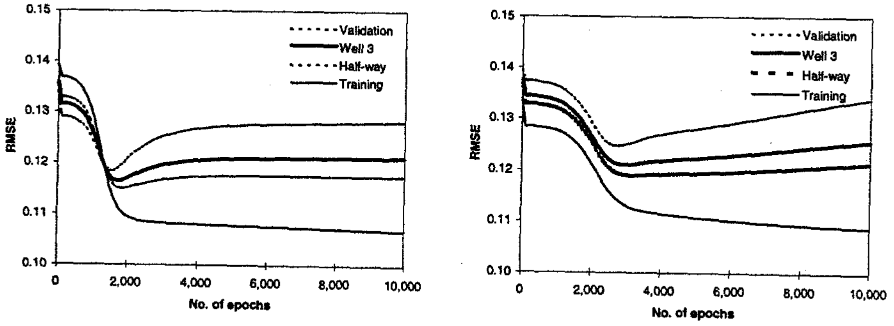
Figure 3 shows the BPNN learning profiles for the SOM data-splitting approach. We stopped training based on the minimum validation errors. Note that the shapes of the learning curves are similar in both training schemes. This is because the statistics of training and validation sets are similar. The final estimates were averaged. Figure 4 shows the learning profiles for the proposed neural-fuzzy method. We stopped training based on minimum interpolated errors. The interpolated error curves for different well locations are also shown in Figure 4. The shapes of the curves are dissimilar because the statistics of the training and validation sets are different. The final estimates were calculated based on the separation distances as in Equation (4). Figure 5 displays the predicted permeability profiles at Well 3 together with the actual data. This figure shows only range of the predicted values (0.35-0.65). Total sum-of-squares were used to evaluate the performance of different models, and the results of the comparisons are summarised in Figure 6.



a) Learning profile using Set A for training, Set B for validation.

b) Learning profile using Set B for training, Set A for validation.

Figure 3. BPNN learning profiles for the SOM data-splitting approach with root-mean-squares error (RMSE) versus number of epochs.



a) Learning profile using Well 1 for training, Well 2 for validation.

b) Learning profile using Well 2 for training, Well 1 for validation.

Figure 4. BPNN learning profiles for the neural-fuzzy approach with root-mean-squares error (RMSE) versus number of epochs.

The results showed that the SOM method (based on minimum validation errors) gave approximately the same performance as the neural-fuzzy method (based on minimum interpolated errors) when we assumed the location of Well 3 is half-way between Well 1 and Well 2. This shows that the SOM method is only applicable if the well location is "averaged" among the locations of the well-log databases. When the actual location of Well 3 is used, the neural-fuzzy method provided the best results (lowest error). This concludes that the neural-fuzzy method is applicable in all well locations.

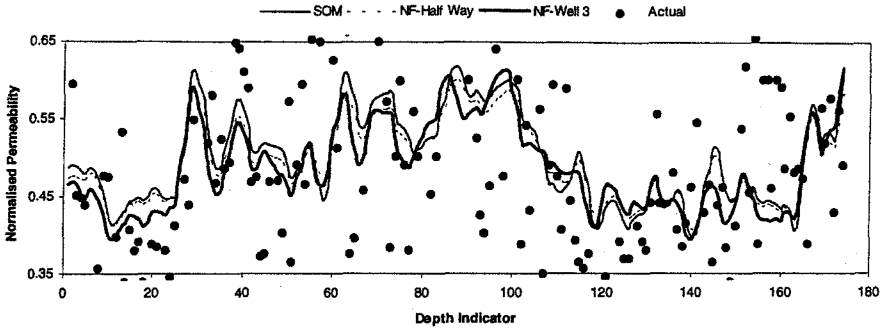


Figure 5. Permeability profiles at Well 3. "SOM" denotes results by data splitting by self-organising map, "NF-Half Way" and "NF-Well 3" mean neural-fuzzy estimates at half way between Well 1 and Well 2, and actual Well 3 location, respectively.

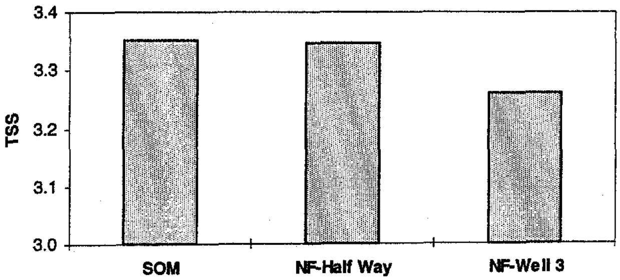


Figure 6. Summary of analyses.

CONCLUSIONS

The paper uses a new technique in function approximation and interpolation based on neural networks and fuzzy rules interpolation. The technique was applied to petroleum well-log analysis in which data is spatially correlated. Based on the results obtained from this paper, the major findings are:

1) Splitting a combined well-log database into training and validation patterns is not inappropriate in spatial data analysis, unless the location of interest is averaged among all the locations of the database.
2) The use of minimum validation error to stop neural training is not applicable in spatial data analysis, unless, again, the location of interest is averaged among all the locations of the database.
3) The proposed neural-fuzzy interpolation method, based on the use of minimum interpolated error, provides reliable estimates at any location of interest.
4) The current methodology assumes that the spatial functions are fully correlated which is not always true in geological modelling (e.g. in the presence of faulting). Hence, further work is required to extend the methodology for interpolating partial correlated functions.

REFERENCES

1. Wong, K.W., Fung, C.C. and Eren, H., 1996. "A Study of the Use of Self-Organising Map for Splitting Training and Validation Sets for Backpropagation Neural Network", *IEEE Region Ten Conference (TENCON) on Digital Signal Processing Applications*, November 27-29, Perth, Australia, vol. 1, pp. 157-162.

2. Shibli, S.A.R., Wong, P.M. and Maignan, M., 1996. "Interpolation Neural Networks: A New Methods for Interpolation", *5th International Workshop on New Computing Techniques in Physics Research*, September 2-6, Lausanne, Switzerland, 4 pages.

3. Wu, X. and Zhou, Y., 1993. "Reserve Estimation using Neural Network Techniques", *Computer and Geosciences*, vol. 19, no. 4, pp. 567-575.

4. Baranyi, P., Gedeon, T.D. and Kóczy, L.T., 1995. "A General Method for Fuzzy Interpolation: Specialised for Crisp Triangular and Trapezoidal Rules", *Proceedings of the 3rd European Congress on Intelligent Techniques and Soft Computing*, Aachen, 5 pages.

5. Kóczy, L.T. and Hirota, K., 1993. "Approximate Reasoning by Linear Rule Interpolation and General Approximation", *International Journal of Approximate Reasoning*, no. 9, pp. 197-225.

6. Wong, P.M., Taggart, I.J. and Gedeon, T.D., 1995. "Use of Neural Network Methods to Predict Porosity and Permeability of a Petroleum Reservoir", *AI Applications*, vol. 9, no. 2, pp. 27-37.