PERGAMON

Engineering Applications of Artificial Intelligence 14 (2001) 15–21

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

An integrated neural-fuzzy-genetic-algorithm using hyper-surface membership functions to predict permeability in petroleum reservoirs

Yuantu Huang[a], Tom D. Gedeon[b,*], Patrick M. Wong[c]

[a] TechComm Simulation Pty Limited, Level 1, 53 Balfour Street, Chippendale, Sydney, NSW 2008, Australia
[b] School of Information Technology, Murdoch University, Perth, WA 6150, Australia
[c] School of Petroleum Engineering, The University of New South Wales, Sydney, NSW 2052, Australia

Abstract

This paper introduces a new neural-fuzzy technique combined with genetic algorithms in the prediction of permeability in petroleum reservoirs. The methodology involves the use of neural networks to generate membership functions and to approximate permeability automatically from digitized data (well logs) obtained from oil wells. The trained networks are used as fuzzy rules and hyper-surface membership functions. The results of these rules are interpolated based on the membership grades and the parameters in the defuzzification operators which are optimized by genetic algorithms. The use of the integrated methodology is demonstrated via a case study in a petroleum reservoir in offshore Western Australia. The results show that the integrated neural-fuzzy-genetic-algorithm (INFUGA) gives the smallest error on the unseen data when compared to similar algorithms. The INFUGA algorithm is expected to provide a significant improvement when the unseen data come from a mixed or complex distribution.

Keywords: Neural networks; Fuzzy logic; Genetic algorithms; Petroleum reservoirs; Permeability; Well logs

1. Introduction

A petroleum reservoir is a volume of porous sedimentary rock which has been filled with a substantial amount of hydrocarbons, such as crude oil and natural gas. Reservoir properties are a set of parameters which are used to characterize the spatially varied geological information. Permeability, porosity, and fluid saturation are the three major reservoir properties in petroleum engineering which are used to determine the economic reserves and production rates of such hydrocarbons.

The determination of reservoir properties is a complex problem because laboratory-measured properties on rock samples ("cores") are only available in limited and isolated well locations and/or intervals. There is a great demand to develop correlation models to relate the properties to other measures which are relatively abundant. One example of such kinds of measures is "well logs" which are a series of multi-type digital measurements along the vertical depth of drilled wells. These models are used to transform the well log data to the reservoir properties at locations where no cores are obtained.

In more general terms, the problem is to predict the system output responding to a new input vector based on past observations (or training data) of the system. In this paper, the system output is a scaler. The solution to the problem uses three basic domains of information: input data (e.g. well logs), output data (e.g. permeability) and a transfer function. The domain of the problem can be expressed in its most generalized form by

$$y = f(\boldsymbol{X}) + \varepsilon, \tag{1}$$

where $\boldsymbol{X}$ and y represent the vector of inputs to the system and the desired output, respectively, $f(\cdot)$ is a function of the domain and $\varepsilon = y - \hat{y}$ is the estimation error where $\hat{y} = f(\boldsymbol{X})$. The goal of the function design is to minimize the total $|\varepsilon|$or ε^2 for all training data.

There are many algorithms for solving this problem. Techniques such as multiple linear statistical regression (Jian et al., 1994), neural networks (Wong et al., 1995;

*Corresponding author. Fax: +61-8-9360 2941.

E-mail addresses: yuantuh@techsim.com.au (Y. Huang), tom@it.murdoch.edu.au (T.D. Gedeon), pm.wong@unsw.edu.au (P.M. Wong).

0952-1976/01/$ - see front matter
PII: S0952-1976(00)00048-8

Mohaghegh et al., 1996a), and fuzzy-neural networks (Huang et al., 1996) are common. These algorithms, however, do not make use of the uncertain knowledge generally available within the problem domain.

Zeng and Singh (1996) used fuzzy logic to emulate the flexibility of human reasoning processes and to draw conclusions from imprecise and incomplete information, thus "capturing the richness of natural language". This method of reasoning is known as fuzzy approximate reasoning, which is a rule-based system of inference in which a fuzzy conclusion is deduced from a collection of fuzzy premises. It is very much suitable for representing uncertain knowledge (Kasabov, 1996). Fuzzy modeling can model highly complex nonlinear systems, such as multi-input and multi-output problems. It has also been applied successfully in geosciences. One example is the work done by Fang and Chen (1997). They used fuzzy modeling to improve the capability of making both linguistic and numeric predictions based on qualitative knowledge and semi-quantitative data in geology. The other example is from Gedeon et al. (1997). They incorporated fuzzy IF−THEN rules into neural networks to interpolate the reservoir properties.

The major difficulty in fuzzy modeling is how to make the decision on the parameters for the fuzzy membership functions. The parameters are often specified by users, either from their own experience or trial and error exercises. In order to automate the process, an optimization technique can be employed to minimize the total error for all the given data. Genetic algorithms (GAs) are popular (Micinney and Lin, 1994; Wang and Elbuluk, 1996; Karr and Freeman, 1997) and become increasingly important in geosciences (Fang et al., 1996; Huang et al., 1998; Mohaghegh et al., 1996b).

GAs were first introduced in the field of artificial intelligence by Holland (1975). These algorithms mimic processes from the Darwinian theories of natural evolution in which winners survive to reproduce and pass along "better" genes to the next generation, and ultimately, a "perfectly adapted" species is evolved. Hence, the term "genetic" was adopted as the name of the optimization. More details about GAs can be found in Lucasius and Kateman (1993, 1994) and Goldberg (1989).

The objective of this paper is to propose a new methodology for predicting permeability from well logs using our integrated neural-fuzzy-genetic-algorithm (INFUGA). It is an assumption-free, model-free, and adaptive estimator and is suitable for handling multi-dimensional inputs and outputs. The methodology will first be described step by step. The workings of the method will be demonstrated via a case study in Australia. The study uses well logs (multiple inputs) to predict permeability (single output) in an oil well where actual permeability values are available for performance evaluation. Results and conclusions follow.

2. The methodology: INFUGA

The proposed methodology, INFUGA, consists of five steps: (1) select appropriate well data sets; (2) generate fuzzy rules by neural networks; (3) generate hyper-surface membership functions by neural networks; (4) optimize defuzzification operator parameters by genetic algorithms; and (5) interpolate fuzzy rules to provide estimates. Fig. 1 displays the corresponding flow chart for INFUGA. Our technique is novel in a number of ways. These are, in the use of hyper-surface membership functions, in the use of GAs to determine defuzzification parameters in the way we have done, and the subsequent use of interpolation. Of course, many components of our technique are based on modifications of techniques developed by others which are not novel in their own right, for example fuzzy rule interpolation is a well-known technique. Overall, our technique can also be seen as an extension of that of Takagi and Hayashi (1991) so that we use a GA-based learning method of the defuzzification parameters.

2.1. Select appropriate well data sets

The decision on selecting appropriate data sets is based on the spatial locations of the available data and the spatial location of the single datum to be estimated (we will henceforth refer to this as a "point"). This, in turn, depends on whether the geological setting of the unknown point is similar to those of the available data. The basic idea is to select training data with similar distribution (statistical and geological) as the unknown point. Expert knowledge is often required to make such a decision because the data distribution of the unknown point is generally not known before a core sample is made.

The estimation of the permeability in an un-cored well (which is a well with no core samples available) relies on the input−output relations from the "nearby" cored wells. Depending upon the complexity of the geology,

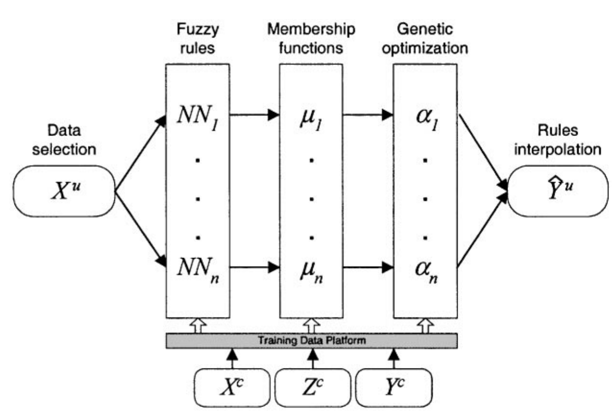


Fig. 1. INFUGA flow chart.

the definition of "nearby", in practice, can range from hundreds to thousands of meters. By taking into account the spatial variability of the data sets from different cored wells, each well data set can be used to produce its own functional relation (fuzzy inference rule) between well logs and permeability. All the rules (data from different cored wells) can then be interpolated to the un-cored well. This will be described in the later sections.

2.2. Generate fuzzy rules by neural networks

This step first generates a standard backpropagation neural network (Gedeon et al., 1997) using the training set from each of the cored wells (assuming n of them). The error function is defined as $\sum_j^{s_i} (y_{ij}^{\rm c} - \hat{y}_{ij}^{\rm c})/s_i$, where $\hat{y}_{ij}^{\rm c} = \mathrm{NN}_i(\boldsymbol{X}_{ij}^{\rm c})$ is the neural network permeability estimate using $\boldsymbol{X}_{ij}^{\rm c}$ (m dimensions), the jth input vector (well logs) at the ith cored well and $y_{ij}^{\rm c}$ is the core permeability data (target output). The trained neural network is then used to produce a fuzzy inference rule

$$\text{Rule } i: \ \mathrm{IF} \boldsymbol{X}^{\rm u} \in A_i \ \mathrm{THEN} \ \hat{y}_i^{\rm u} = \mathrm{NN}_i(\boldsymbol{X}^{\rm u}), \quad i = 1, \ldots, n, \tag{2}$$

where $\boldsymbol{X}^{\rm u}$ is the un-cored input vector (test or unseen data) and $A_i = \{\boldsymbol{X}_{ij}^{\rm c} : j = 1, \ldots, s_i\}$ is the crisp universe of discourse of the inputs at the ith cored well. This is equivalent to applying only neural networks to predict properties from different relations obtained in different wells (rules).

Our new algorithm uses A_i as a fuzzy set. This means that all the rules could be fired depending upon the degree of the hyper-surface membership of $\boldsymbol{X}^{\rm u}$ belonging to each set. The next step will show how we obtain the membership values.

2.3. Generate hyper-surface membership functions by neural networks

The value of the hyper-surface membership function can be defined as the output of the trained neural network (Takagi and Hayashi, 1991). The network is essentially a standard structure for pattern classification with m input and n output units. The training patterns become $(\boldsymbol{X}_{ij}^{\rm c}, \boldsymbol{Z}_j^{\rm c})$, where $\boldsymbol{Z}_j^{\rm c} = (z_{1j}^{\rm c}, \ldots, z_{nj}^{\rm c})$ is the target vector. The elements of the target vector are

$$z_{kj}^{\rm c} = \begin{cases} 1 & \text{if } \boldsymbol{X}_{kj}^{\rm c} \in A_1 \cup \cdots \cup A_n, \\ 0 & \text{otherwise,} \end{cases} \quad k = 1, \ldots, n, \ j = 1, \ldots, s_k. \tag{3}$$

Thus, the network outputs are used as the hyper-surface membership function values. For prediction purposes, these values are

$$\mu_k(\boldsymbol{X}^{\rm u}) = \hat{z}_k^{\rm c}(\boldsymbol{X}^{\rm u}), \quad k = 1, \ldots, n, \tag{4}$$

where $\mu_k \in [0, 1]$ values are the fuzzy membership grades which express the degree of membership of $\boldsymbol{X}$ belonging to the kth cored well. These values are to be used in the defuzzification operators.

2.4. Optimize defuzzification operator parameters by genetic algorithms

Let $(\alpha_1, \ldots, \alpha_n)$ be a family of parameters to be optimized. The permeability (defuzzified value) can be obtained as the expected value of the parameterized family of defuzzification operators

$$\hat{y}_{ij}^{\rm c} = \frac{\sum_k^n \mu_k^{\alpha_k}(\boldsymbol{X}_{ij}^{\rm c}) \mathrm{NN}_i(\boldsymbol{X}_{ij}^{\rm c})}{\sum_k^n \mu_k^{\alpha_k}(\boldsymbol{X}_{ij}^{\rm c})}, \quad i = 1, \ldots, n. \tag{5}$$

In conventional defuzzification, $\alpha_1 = \cdots = \alpha_n = 1$. Filev and Yager (1991) proposed the *basic defuzzification distributions* (BADD) and assumed that $\alpha_1 = \cdots = \alpha_n$ but not necessarily equals one. Note that when $\alpha_1 = \cdots = \alpha_n = 0$, the estimate becomes the simple average $\sum_i^n \mathrm{NN}_i(\boldsymbol{X}^{\rm u})/n$. This is analogous to the neural network approach pulling all the well data (independent of well locations) into a single rule to produce an estimate (Wong et al., 1995).

In this paper, the $\alpha_1 = \cdots = \alpha_n$ assumption is removed and different α values are allowed in the defuzzification operators. The corresponding values are optimized by GAs. The following fitness function is used in this study

$$F(\alpha) = \frac{1}{1 + \lambda \sum_i^n \varepsilon_i + (1 - \lambda) \sum_i^n \sum_{j=i+1}^n |\varepsilon_i - \varepsilon_j|}, \tag{6}$$

where $\varepsilon_i = \sum_j^{s_i} (y_{ij}^{\rm c} - \hat{y}_{ij}^{\rm c})^2/s_i$, s_i is the number of validation patterns available at the ith well and $\lambda \in [0, 1]$ is a user-definable constant. If $\lambda = 1$, the algorithm prefers to get the smallest error irrespective of the significance of each fuzzy rule. If $\lambda = 0$, the algorithm prefers to balance the errors from different rules. The parameters α are optimized by maximizing $F(\alpha)$.

Note that the α_i values in Eq. (5) indicate the contribution of the fuzzy rule i to the estimate. Since $\mu_i \in [0, 1]$, small α_i will give high $\mu_i^{\alpha_i}$. This means, a smaller α_i value will give a higher weighting to the estimate from fuzzy rule i. In other words, the input−output relation in rule i is strong.

2.5. Interpolate fuzzy rules to provide estimates

Using the well logs at the un-cored or test well, the final permeability estimate is

$$\hat{y}^{\mathrm{u}} = \frac{\sum_k^n \mu_k^{\alpha_k}(\boldsymbol{X}^{\mathrm{u}})\mathrm{NN}_i(\boldsymbol{X}^{\mathrm{u}})}{\sum_k^n \mu_k^{\alpha_k}(\boldsymbol{X}^{\mathrm{u}})}. \tag{7}$$

3. Case study

3.1. Data descriptions

The data sets used in this study came from four oil wells, namely 1, 2, 3 and 4 which were drilled in a complex reservoir located at offshore Western Australia. According to geological experts, all the wells were drilled in a similar depositional environment. Hence, the functional relation from one well can be used to infer the relation in another well.

In each well, six well logs ($n = 6$) and the corresponding core permeability values are available. There are 152, 156, 115, and 140 measurements in each well, respectively, at various depths. The well logs used for the analyses were gamma-ray (GR), deep resistivity (LLD), sonic travel time (DT), bulk density (RHOB), neutron porosity (NPHI) and a rock-type (FACIES) log. The first five recorded numerical measurements while the last one represents the discrete groupings of the rock. All the data used were normalized and became dimensionless.

The objective of the case study is to demonstrate the workings of the proposed methodology for predicting permeability from the six well logs. In order to evaluate the performance of the methodology, well 4 was selected as the test well where predictions were made through the use of data from the other wells (1−3). The predictions were then compared with the actual values. The proposed methodology was also compared with other prediction methods: neural networks only, conventional defuzzification and BADD (Filev and Yager, 1991).

3.2. Neural networks and fuzzy rules

In this study, a standard three-layer feedforward neural network was used to provide the fuzzy rules and the hyper-surface membership functions. A separate network was used in each of the three cored wells. The network consisted of six inputs and one output. The same number of hidden units was used for all the three networks. The number of hidden units and the number of learning iterations were determined by trial and error based on the minimum error on the validation set, which was formed by randomly picking 50% of the training data. The remaining 50% of the training data were used to train the network. Both the learning and momentum constants were set at 0.5. In this example, three hidden units produced the smallest error.

After training, three fuzzy rules were obtained. These rules can be used separately to produce the permeability values at the test well 4 (neural networks only). In this case, different rules will provide different permeability estimates. The rules can also be interpolated to the test well, using the hyper-surface membership function values $\mu_k(\boldsymbol{X}^{\mathrm{u}})$, using the conventional defuzzification, the BADD approach or the proposed INFUGA.

A similar analysis was done to optimize the network configuration for predicting the $\mu_k(\boldsymbol{X}^{\mathrm{u}})$ values. Fig. 2 shows the membership function values (scaled to a total of 100%) obtained in the three wells using the well 4 data (140 points). The vertical axis represents the scaled membership values and the horizontal axis is the sample number (sorted by reservoir depths). Note that 52% of the well 4 data produced the highest membership function values from the second fuzzy rule 2. This indicates that well 4 data set has a similar distribution to the well 2 data set.

3.3. Genetic algorithms

In this example, there were three parameters $(\alpha_1, \alpha_2, \alpha_3)$ which were required to be optimized by

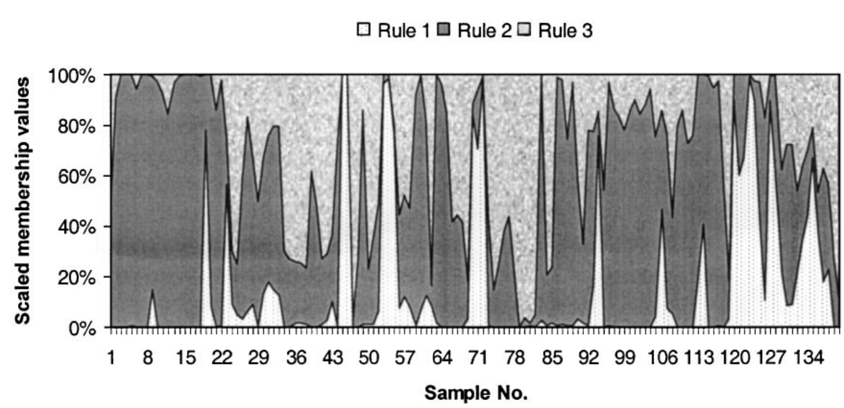


Fig. 2. Hyper-surface membership function values (scaled) using well 4 data.

genetic algorithms. Binary encoding was used. Each individual parameter was linearly generated in [0,10] and the bit-string length was set to $\iota = 32$. Therefore, when concatenating the three parameters, each of the individuals has a total bit string of 96 bits. The Karr and Freeman (1997) relation was used for the parameter mapping. In this study, λ was set to 0.2. The population size was 30 and the number of generations was 10,000. The probability values for crossover and mutation were 0.8 and 0.004 respectively. The final values for $(\alpha_1, \alpha_2, \alpha_n)$ were (8.31, 0.16, 9.80) at maximum $F(\alpha)$. Since α_2 is the smallest, a higher weighting will be assigned to the fuzzy rule 2. This may indicate that the input−output relations in well 2 are much stronger than those embedded in the other wells.

For BADD, a similar GA optimization was done by setting $\alpha_1 = \alpha_2 = \alpha_3 = 1$. The optimum α value was 1.65. Note that $\alpha_1 = \alpha_2 = \alpha_3 = 1$ for the conventional defuzzification approach.

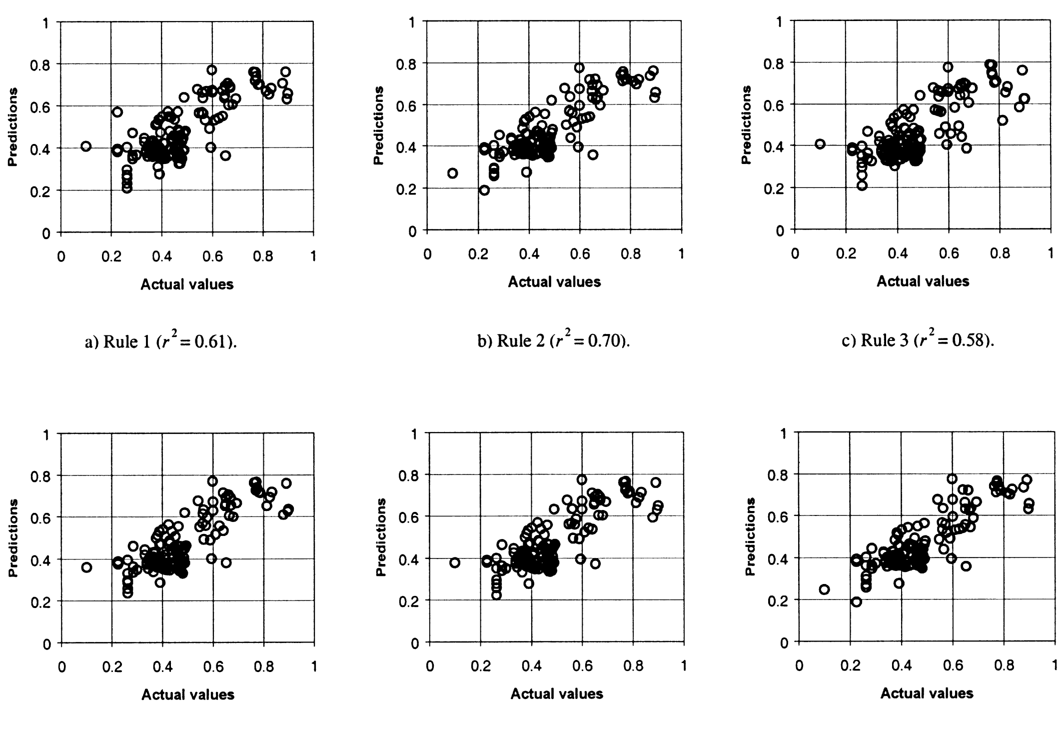
4. Results and discussion

Table 1 shows the comparison of the total sum of error squares (TSS), for 140 data points at well 4 using different algorithms: neural network approach using Eq. (2), conventional defuzzification, BADD and INFUGA. According to the minimum criteria in Eq. (1), the INFUGA estimates were the best overall. The percentage improvement and the ranking of the algorithms are also shown. The worst results were from the neural network techniques using rules 1 and 3. The INFUGA predictions were 25 and 31% better, respectively.

It is important to note that the fuzzy rule 2 gave the second best results. INFUGA only showed slightly better results (3%). This was because the data distributions from wells 2 and 4 were similar as displayed in Fig. 2. That means neural networks give good performance to the unseen data if they were trained by data coming from the same or similar distribution. The INFUGA technique is expected to provide a significant

Table 1
Comparison of the total sum of error squares (TSS) at well 4 using different algorithms. The percent improvement was referenced at INFUGA

Algorithms	TSS	Percent improvement	Rank
Neural network using rule 1	1.34	25	5
Neural network using rule 2	1.04	3	2
Neural network using rule 3	1.47	31	6
Conventional defuzzification	1.17	14	3
BADD	1.22	17	4
INFUGA	1.01	n/a	1



a) Rule 1 ($r^2 = 0.61$). b) Rule 2 ($r^2 = 0.70$). c) Rule 3 ($r^2 = 0.58$).

d) Conventional defuzzification ($r^2 = 0.66$). e) BADD ($r^2 = 0.64$). f) INFUGA ($r^2 = 0.71$).

Fig. 3. Scatter-plot of the actual values and predictions at well 4.

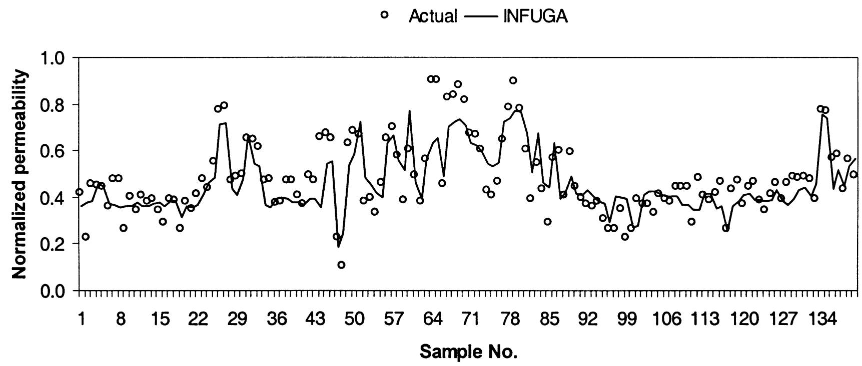


Fig. 4. Permeability profiles at well 4.

improvement when the unseen data come from a mixed or complex distribution. More studies will be conducted in the future.

Comparing the fuzzy methods, the INFUGA technique was 14 and 17% better than the conventional defuzzification and the BADD methods, respectively. Setting $\alpha_1 = \alpha_2 = \alpha_3$ decreases the performance of the estimator. Clearly, by removing this assumption, the INFUGA estimator can approximate more complex relations.

Fig. 3 shows the scatter-plots of the 140 predictions versus the actual values for different methods. The INFUGA technique gave the maximum correlation coefficient (r^2) of 0.71. Fig. 4 shows the results of the INFUGA predictions and the actual (normalized) permeability values in well 4. The predictions matched very well with the actual values.

The performance of the INFUGA was excellent compared with other neural-fuzzy methods. The proposed method not only has the intrinsic advantages of the neural-fuzzy techniques, but the parameters are also optimized by genetic algorithms. The integrated approach improved the predictions by 3−31%. This adaptive technique is not limited to petroleum engineering problems, but can also be used in other engineering areas. This is particularly useful when it is desirable to incorporate interpretive knowledge based on a more complex understanding of the data. The cost for improving the performance is the CPU time for the numerical optimization.

5. Conclusions

This paper introduces an integrated neural-fuzzy-genetic-algorithm (INFUGA) to predict permeability from well logs in a petroleum reservoir in Australia. INFUGA is an improved version of our previous neural-fuzzy techniques which optimized the parameters in the defuzzification operators via genetic algorithms. It is an assumption-free, model-free, and adaptive estimator and is suitable for handling multi-dimensional inputs and outputs. It does not require a structured knowledge base. The only disadvantage is the additional CPU time for the performance.

In the case study, the performance of the INFUGA was excellent compared to the other neural-fuzzy methods we tried. The comparison shows that the integrated neural-fuzzy-genetic-algorithm (INFUGA) provided the smallest error on the unseen data for our data. The INFUGA technique is expected to provide a significant improvement when the unseen data come from a mixed or complex distribution. Our research in this area is on-going.

References

Fang, J.H., Chen, H.C., 1997. Fuzzy modelling and the prediction of porosity and permeability from compositional and textural attributes of sandstone. Journal of Petroleum Geology 20 (2), 185−204.

Fang, J.H., Karr, C.L., Stanly, D.A., 1996. Transformation of geochemical log data to mineralogy using genetic algorithm. The Log Analyst 37 (2), 26−31.

Filev, D.P., Yager, R.R., 1991. A generalized defuzzification method under BAD distribution. International Journal of Intelligent System 6, 687−697.

Gedeon, T.D., Wong, P.M., Huang, Y., Chan, C., 1997. Two dimensional neural-fuzzy interpolation for spatial data. Proceedings of GIS AM/FM ASIA '97 & Geoinformatics '97, Vol. 1, Taipei, Taiwan, pp. 159−166.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA.

Holland, J., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Harbor.

Huang, Y., Wong, P.M., Gedeon, T.D., 1996. An improved fuzzy neural network for permeability estimation from wireline logs in a petroleum reservoir. Proceedings of IEEE Region Ten Conference (TENCON) on Digital Signal Processing Applications, Vol. 2, Perth, Australia, pp. 912−917.

Huang, Y., Wong, P.M., Gedeon, T.D., 1998. Prediction of reservoir permeability using genetic algorithms. AI Applications 12 (1−3), 67−75.

Jian, F.X., Chork, C.Y., Taggart, I.J., McKay, D.M., Bartlett, R.M., 1994. A genetic approach to the prediction of petrophysical properties. Journal of Petroleum Geology 17 (1), 71−88.

Karr, C.L., Freeman, L.M., 1997. Genetic-algorithm-based fuzzy control of spacecroft autonomous rendezvous. Engineering Applications of Artificial Intelligence 10 (3), 293–300.

Kasabov, N., 1996. Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. MIT Press, Cambridge, MA.

Lucasius, C.B., Kateman, G., 1993. Understanding and using genetic algorithms: part 1. Concepts, properties and context. Chemometrics and Intelligent Laboratory System 19, 1–31.

Lucasius, C.B., Kateman, G., 1994. Understanding and using genetic algorithms: part 2. Representation, configuration and hybridisation. Chemometrics and Intelligent Laboratory System 20, 9–145.

Micinney, D.L., Lin, M.S., 1994. Genetic solution of groundwater management models. Water Resources Research 30, 1897–1906.

Mohaghegh, S., Arefi, R., Ameri, S., Aminiand, K., Nutter, R., 1996a. Petroleum reservoir characterization with the aim of artificial neural networks. Journal of Petroleum Science and Engineering 16, 263–274.

Mohaghegh, S., Balan, B., Ameri, S., McVey, D.S., 1996b. A hybrid, neural–genetic approach to hydraulic treatment design and optimization. SPE 36602, SPE Annual Technical Conference and Exhibition, Denver, 6 pp.

Takagi, H., Hayashi, I., 1991. NN-driven fuzzy reasoning. International Journal of Approximate Reasoning 5, 91–212.

Wang, X., Elbuluk, M., 1996. Neural network control of induction machines using genetic algorithm training. 31st IAS Annual Meeting, Vol. 3. IEEE, New York, USA, pp. 1733–1740.

Wong, P.M., Taggart, I.J., Gedeon, T.D., 1995. Use of neural network methods to predict porosity and permeability of a petroleum reservoir. AI Applications 9 (2), 27–38.

Zeng, X., Singh, M.G., 1996. Approximation accuracy analysis of fuzzy system as function approximators. IEEE Transactions on Fuzzy Systems 4, 44–63.

Yuantu Huang received a B.Sc. in mathematics in 1982 from the Zhejiang Normal University and M.Sc. in mathematical geology and computer applications in geology in 1989 from the Research Institute of Petroleum Exploration and Development in China. He received his Ph.D. in Computer Science and Engineering in 1999 from the University of New South Wales in Australia. His research interests are the integration of neural networks, fuzzy logic and genetic algorithms in petroleum engineering. He is currently a Senior Software Engineer at TechComm Simulation Pty. Ltd. in Australia.

Tom D. Gedeon received a B.Sc. (Hons.) and a Ph.D. in computer science in 1981 and 1989, respectively, from the University of Western Australia. He is currently Head of School of Information Technology at Murdoch University and a Visiting Professor at the Department of Telecommunications and Telematics, Technical University of Budapest, Hungary. His research interests are information retrieval and index generation, extracting knowledge (data mining) from trained neural networks, and AI applications.

Patrick M. Wong received a B.E. (Hons.) and a Ph.D. in petroleum engineering in 1993 and 1996, respectively, from the University of New South Wales. He is currently a Senior Lecturer in Petroleum Engineering at the same university. His research interests are in the areas of petrophysics, geostatistics and AI applications in reservoir characterisation. He was formerly with Petroconsultants Australasia Pty. Ltd. in Sydney.