CANADIAN JOURNAL OF EXPLORATION GEOPHYSICS
VOL. 30, NO. 1 (JUNE 1994), P. 60-68

GEOLOGICAL PATTERN RECOGNITION AND MODELLING WITH A GENERAL REGRESSION NEURAL NETWORK

ZEHUI HUANG[1] AND MARK A. WILLIAMSON[1]

ABSTRACT

The computer neural network (CNN) technique continues to draw more attention from geoscientists. The back-propagation neural network (BP-CNN), for example, has been used for such geological applications as lithology identification from well logs and seismic data processing. In this paper, we highlight the geological utility of a different type of CNN called general regression neural network (GRNN). In contrast to BP-CNNs, GRNNs do not converge to a local minimum, are more easily implemented, can be trained more efficiently and have the capacity to handle incomplete patterns. We demonstrate the usefulness and potential of the GRNN algorithm with reference to an example that examines relationships between the total organic carbon content of rock intervals and their well-log responses. In another example, we show GRNN's ability to recognize automatically the carbon isotopic profiles of crude oils, after being trained with a small data set.

INTRODUCTION

The past two decades have seen the development and application of numerous quantitative approaches in the geosciences. Ranging from spectral to cluster analysis, these approaches have found wide application throughout geosciences and have been used for process simulation, modelling, mapping, stratigraphic analysis and correlation, classification and prediction (Agterberg and Bonhan-Carter, 1990; Agterberg and Griffiths, 1991). Most of the quantitative approaches applied in the past two decades have been statistical in nature. Their application has permitted systematic, rapid and objective analysis and processing of geological data. The usefulness of any of these methods depends in large part on the nature of the problem to be solved as they often try to simplify processes that are acting in a complicated, dynamic and often nonlinear way. Geoscientists therefore are constantly searching for, and embracing, new quantitative techniques to provide solutions for both new and old problems. The geological application of artificial intelligence is one of the techniques introduced during the last five years and has seen rapid development (Simann and Aminadeh, 1989).

Artificial intelligence broadly defines such computing techniques as expert systems, fuzzy set logic and computer neural networks (CNNs). These have been developed as a result of the cooperative efforts made by scientists from many disciplines such as mathematics (traditional statistics), computer science, biology and psychology. Of these new techniques, CNNs (Stephen, 1990; Hertz et al., 1991) are appealing to geologists due to several advantages they possess over traditional statistical and deterministic approaches. There is, for example, less need to determine relevant factors a priori. The algorithms are more tolerant to noisy data because of the abundance of input factors and have an ability to model a problem directly and in a more sophisticated way. As a nonlinear dynamic system, a CNN can learn to recognize patterns or model complicated relationships through the generation of its own rules after being presented with a series of input values (patterns) and corresponding output values (definition of the patterns).

The types of problems that lend themselves to a CNN approach are: (1) where the problem area is rich in historical data or examples; (2) where data are numeric and are intrinsically noisy; (3) where the discriminator or function to determine solutions is unknown or expensive to discover; (4) where the problem is dependent upon multiple interacting parameters whose interdependence is too realistically difficult to describe; and (5) where conventional approaches are inadequate because the problem is too complex for a programmer to find procedural solutions.

The application of CNNs to geoscientific problems was initiated just a few years ago and appears to hold much promise. Examples include the recognition of mineral/lithology from well logs (Baldwin et al., 1990; Rogers et al., 1992); seismic data processing (Pulli and Dysart, 1990; Wang, 1992); rock permeability estimation from porosity and spatial position (Osborne, 1992); and mineral reserve estimation (Wu and Zhou, 1992). Most of these attempts use the most common type of CNN algorithm, called back propagation (BP-CNN) (Rumelhart et al., 1986), or a variation. A BP-CNN, however, has the following limitations: (1) slow

Manuscript received by the Editor May 5, 1994 ; revised manuscript received August 30, 1994; manuscript accepted September 12, 1994.

[1]Atlantic Geoscience Centre, P.O. Box 1006, Dartmouth, Nova Scotia B2Y 4A2

Financial support from GSC, OERD and Unocal Corporation is acknowledged. An earlier version of the manuscript was read by K. Coflin and A. Agrawal of Atlantic Geoscience Centre and J. Shimeld of Dalhousie University. Critique and suggestions from three anonymous reviewers are greatly appreciated.

learning convergence rate; (2) an inefficient approach to determine network architecture; (3) propensity to converge to local minima instead of finding the global minimum of the error surface (Caudill, 1988); and (4) sensitivity to the initially randomized weights. These limitations are only partially addressed in applications seen to date. For example, the quickprop algorithm (Fahlman, 1988) and dynamic node creation scheme (Ash, 1989) were adopted in recent work by Wu and Zhou (1992). It is of use therefore to experiment with other CNN algorithms for solving geological problems.

In this paper, we present and demonstrate the ability of a different computer neural network algorithm, the General Regression Neural Network (GRNN) (Specht, 1991), to perform geological pattern recognition and modelling. We outline the algorithm and then present two examples of its application. The first models the relationship between a suite of well-log responses and measured total organic carbon content as the basis for predicting organic carbon from well logs. The second classifies crude oil derived from different types of source rock according to their geochemical characteristics. Data for these applications are from the published literature (Passey et al., 1990; Chung et al., 1994).

The Algorithm and Neural Network Implementation

The GRNN algorithm has little resemblance to the more widely used BP-CNN but it is similar to the probabilistic neural network (Specht, 1990). No iteration is involved in computing with the GRNN algorithm. It has, at its root, one of the most commonly used statistical techniques – regression analysis. To perform regression, one has to assume (or guess) a certain functional form to analyze the data points and to determine the parameters of the function. The limitation of this approach is in the choice of the function. If the wrong functional form is assumed, the value of the established function for prediction is reduced. The GRNN algorithm proposed by Specht (1991) successfully overcomes this shortcoming by not requiring an assumption in the form of the regression function. Instead, it approaches the problem on the basis of the probability density function of the observed data (or the training examples). Thus, there is no need to determine parameters for the function. Another attractive feature is that, unlike BP-CNNs, a GRNN does not converge to local minima (Specht, 1991). Also, the training process with a GRNN-type algorithm is much more efficient than with the BP-CNN algorithm (Specht and Shapiro, 1990).

Mathematically, general regression obtains the conditional mean of y, a scalar random variable y, when given X (a particular measured value of a vector random variable x) using nonparametric estimator of the joint probability density function $f(x, y)$. In the algorithm Specht (1991) employs a class of consistent estimators proposed by Parzen (1962) for this purpose. These estimators are based upon sample values X_i and Y_i and are proven to be also applicable to the multidimensional case (Cacoullos, 1966). With different estimators, two forms of the general regression are proposed:

$$\hat{Y}(X) = \frac{\sum_{i=1}^{n} Y_i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} \exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \tag{1}$$

$$D_i^2 = \sum_{j=1}^{p} \left(X_j - X_{ij}\right)^2 \tag{2}$$

$$\hat{Y}(X) = \frac{\sum_{i=1}^{n} Y_i \exp\left(-\frac{C_i}{\sigma}\right)}{\sum_{i=1}^{n} \exp\left(-\frac{C_i}{\sigma}\right)} \tag{3}$$

$$C_i = \sum_{j=1}^{p} \left|X_j - X_{ij}\right| , \tag{4}$$

where n is number of sample values X_i and Y_i, D_i^2 is the Euclidean distance between X and X_i with vector size p, while C_i is their city block distance, and σ is a smoothing factor. From equations (1) and (3), we can see that general regression is actually a weighted average of all of the observed values, Y_i, where each observed value is weighted exponentially according to its Euclidean or city block distance from X. When the smoothing factor σ becomes very large, $\hat{Y}(X)$ becomes the mean of the observed Y_i. When σ approaches zero, $\hat{Y}(X)$ assumes the value of the Y_i associated with the X_i which approaches X. With intermediate values of σ, all values of Y_i contribute to $\hat{Y}(X)$, with those corresponding to points closer to X being given heavier weight. In general regression, σ is the only important computing parameter to be determined.

When n is very large, for efficiency of general regression it is necessary to group the observations into a smaller number of clusters. Many clustering techniques can be used. The cluster version of general regression can be written as:

$$\hat{Y}(X) = \frac{\sum_{i=1}^{m} A_i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{m} B_i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \tag{5}$$

$$\hat{Y}(X) = \frac{\sum_{i=1}^{m} A_i \exp\left(-\frac{C_i}{\sigma}\right)}{\sum_{i=1}^{m} B_i \exp\left(-\frac{C_i}{\sigma}\right)} \tag{6}$$

$$A_i(k) = A_i(k-1) + Y_j \tag{7}$$

$$B_i(k) = B_i(k-1) + 1 , \tag{8}$$

where m is the number of the cluster, A_i and B_i are the value of the coefficients for cluster i after k observations. A_i is actually the sum of the Y values and B_i the number of observations grouped into cluster i. Those coefficients can be

determined in one pass through the observed samples. A very simple clustering method is to set a single radius, r, with certain information from the data set. The first data point is arbitrarily taken as the first cluster centre. If the second data point has a distance from this cluster centre less than r, the first cluster centre is updated with equations (7) and (8); otherwise, the second data point becomes a new (second) cluster centre. Each future data point from the data set is processed in this way.

In terms of computer neural network technique, a GRNN constructed on the basis of equations (5), (7) and (8) or (6), (7) and (8) consists of four layers (Figure 1). The input layer provides all of the scaled input variables (X) and is fully connected to the pattern layer. The weights of each pattern unit that connect to the input layer actually represent a cluster centre in a multidimensional space. The A and B coefficients associated with these clusters serve as the weights connecting the pattern layer and the summation layer. The unit(s) in output layer receive results from the summation units and then calculate the output.

For training examples (X_1, Y_1, ... X_n, Y_n) with the X vector size being p and the Y vector size q, the number of input units is equal to p, the number of output units q and the number of summation units q+1. It should be noted that only one summation unit uses the B coefficients as the connecting weight, which can be called the B unit, while all other summation units use the A coefficient as the weight, which can be called the A unit. The number of pattern units depends on the structure of the training data set and the clustering technique employed in the training.

As we mentioned earlier, the smoothing factor, σ, is the most important computing parameter for the GRNN's performance. Theoretically, it is not possible to determine the true σ because the underlying parent distribution is not known. However, the optimum σ for a GRNN built from a training data set can be automatically approximated using the "holdout method" (Specht, 1991). This approach starts from an estimated (or guessed) σ and builds the network with all but one example from the training data set. The network is then tested by the remaining example. Each training example in the data set is held once for testing with a certain σ and a mean square error (MSE) is obtained after one round of holding. With another σ the process is repeated. After using a series of trial σ, the one which is associated with the smallest MSE is taken as the optimum σ for the training data set. Then all of the examples in the data set are used to build the final GRNN. The "holdout method" tends to increase the computing time; however, it is more objective.

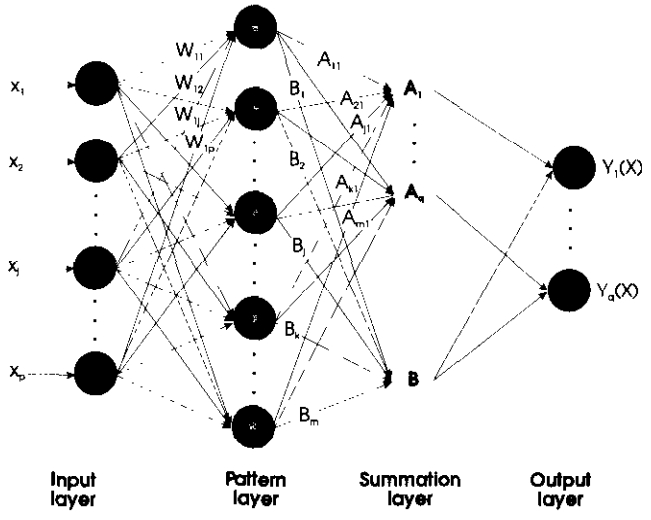

Fig. 1. The general regression neural network. The size of input layer is p, pattern layer m, summation layer q+1 and output layer q. Note the partial connection between the summation and output layers.

When a trained GRNN such as the one in Figure 1 is used for prediction, its response to an input pattern (X) can be explained as follows. The Euclidean or city block distances of the input pattern from each pattern unit are calculated through subtracting each input element from the corresponding weight. The distances from each pattern unit are fed with the smoothing factor into a nonlinear activation function (e.g., an exponential function), in a manner specified by either equation (5) or (6). Then the summation units perform dot product between the weights [A and B coefficients in Equations (7) and (8)] and the output signals from the pattern units (i.e., the activation results). The output from each A summation unit divided by that from the B unit then go to the corresponding output unit as the final GRNN response.

Applications

Major attractions of the GRNN algorithm are that it will not converge to a local minimum (Specht, 1991) and, furthermore, it is not necessary to determine the network architecture. The single important parameter, σ, can be automatically determined during training. In addition, GRNNs need fewer examples than BP-CNNs to be properly trained. Therefore, GRNNs are an attractive and promising alternative to BP-CNNs for use in geosciences. The following two examples illustrate the usefulness of GRNN.

Further explanation is warranted on how the radius of influence is determined. The program first calculates all of the distances among the data points in the training set and finds the difference between the maximum and minimum interdata distance. If the training data set is small ($N \leq 50$), the minimum distance plus 5% of the difference is taken as the radius of influence for clustering; if the training data set is large ($N > 50$), the minimum distance plus 10% of the difference is taken as the radius. We also experimented with different types of distance (Euclidean versus city block). In the later discussion, the use of equation (5) in training and testing is referred to as mode 1 and the use of equation (6) as mode 2.

Although the GRNN technique is able to handle cases when Y is a vector, both of our examples have only one value to be predicted. In both cases, the training process has employed the most simple form of clustering methods as explained earlier.

Example 1: Predicting rock's total organic carbon from well logs

There have been many attempts to identify source rock intervals and estimate their total organic carbon (TOC) from well logs (e.g., Schmoker and Hester, 1983, 1989; Meyer and Nederlof, 1984; Herron and Le Tendre, 1990). Estimating the geochemical nature of rocks from well-log information in the absence of cores can provide greater stratigraphic resolution than laboratory measurements. This information helps in understanding the distribution of organic matter and the processes controlling its deposition and preservation.

Of numerous attempts, the ΔlogR method proposed by Passey et al. (1990) seems to be more successful. The ΔlogR method is a simple model of the relationship between the responses of resistivity log, a porosity log (sonic, density or neutron porosity), and rock's TOC. This method is compatible with Archie's law (Archie, 1942) and has been calibrated by drilling sample data. The usefulness of the ΔlogR method was recently demonstrated in its application to the Egret Member source rock in the Jeanne d'Arc Basin, offshore Eastern Canada (Huang et al., 1994). Drawbacks in using this method are the need to determine baseline values on the well logs, which is a subjective process, and the need to know levels of maturity (LOM) for the interval of interest. LOM can be difficult to determine, especially during the early stage of exploration.

Using a published data set (Passey et al., 1990), we modelled the relationship between well-log responses and TOC content with a GRNN. The data set consists of measured TOC for several exploration wells (locations were not indicated in the original source of data) and their corresponding gamma ray, resistivity and sonic log response (see Appendix 1). The total number of data points is 112. As these data are mostly from source rock intervals, patterns from nonsource rock intervals are also necessary in order to provide a set of comprehensive training examples. Resistivity and sonic responses for sandstone, limestone, dolomite and claystone containing negligible TOC (0 - 0.8 wt.%) at different depth ranges (associated with varying porosity values) were estimated. This was done with a combination of sediment compaction curves (Sclater and Christie, 1980) for estimating porosity at depth, the Wyllie equation (Wyllie et al., 1958) for sonic traveltime, and the Archie equation (Archie, 1942) and the Waxman-Smith equation (Waxman and Smith, 1968) for resistivity.

We extracted 23 data points from the observational data (Passey et al., 1990) for use in the testing data set. The remaining of the observational data were combined with the theoretical well-log responses of nonsource rocks to make the training data set.

After being trained in mode 1, the GRNN found an optimal σ of 0.03 from a starting σ of 0.5 (Figure 2a). The resultant GRNN has 43 pattern units. After being trained in mode 2, the GRNN found an optimal σ (0.1) from a starting σ (0.5) (Figure 2a) and the GRNN has 39 pattern units. Both of the GRNNs trained in different modes give excellent prediction of TOC in response to the 23 testing data points (Figure 3a, b). In terms of the slope of fitting line and the correlation coefficient, it seems that the choice of distance calculation methods has little effect on the performance of the trained GRNN. One of the major benefits of using the trained GRNN for obtaining continuous TOC curves in areas where these training examples are derived is that there is no need to determine the rocks' LOM and well-log baseline values.

For comparison, we used the more common BP-CNN technique. Through several trial runs, we decided that the network architecture with a 4-unit input layer, a 4-unit hidden layer and a 1-unit output layer achieved the best generalization. The prediction (Figure 3c) is slightly better than the prediction by GRNN, as the slope of the fitting line is nearer to 45° and visually the points between 2 and 4 wt.% are more correlatable than those in Figure 3a and 3b. This occurs because for

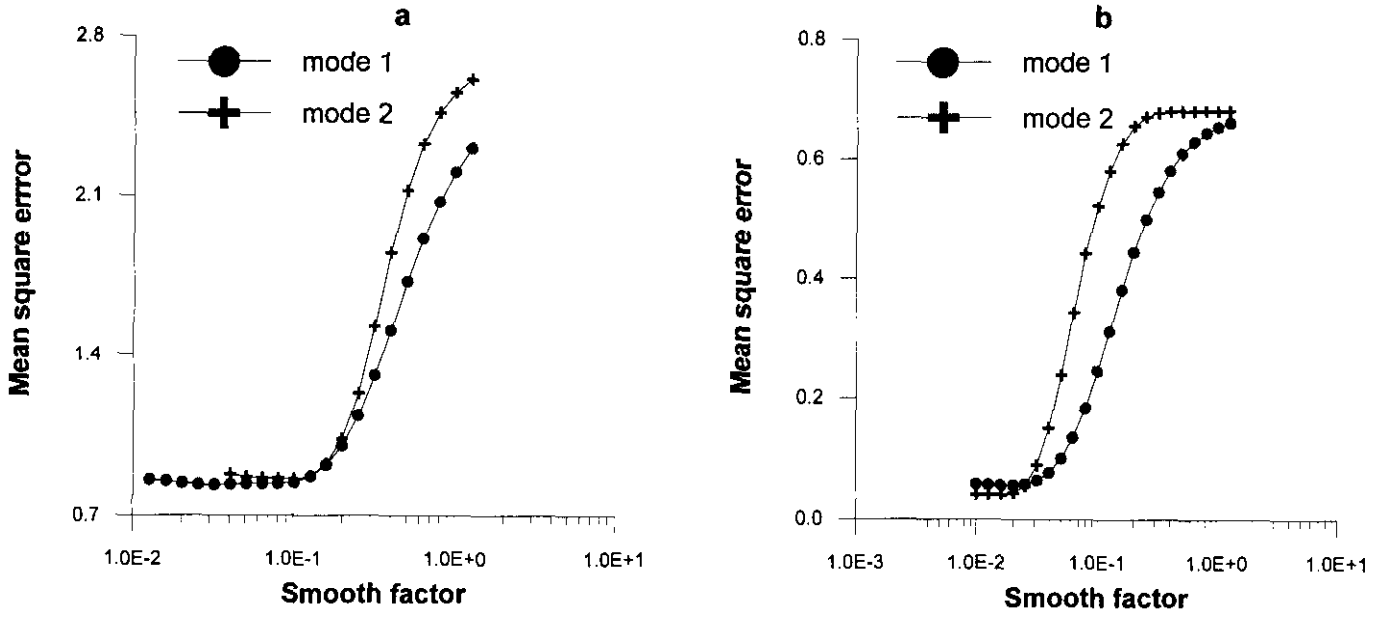

Fig. 2. σ-MSE curves: (**a**) example 1; (**b**) example 2.

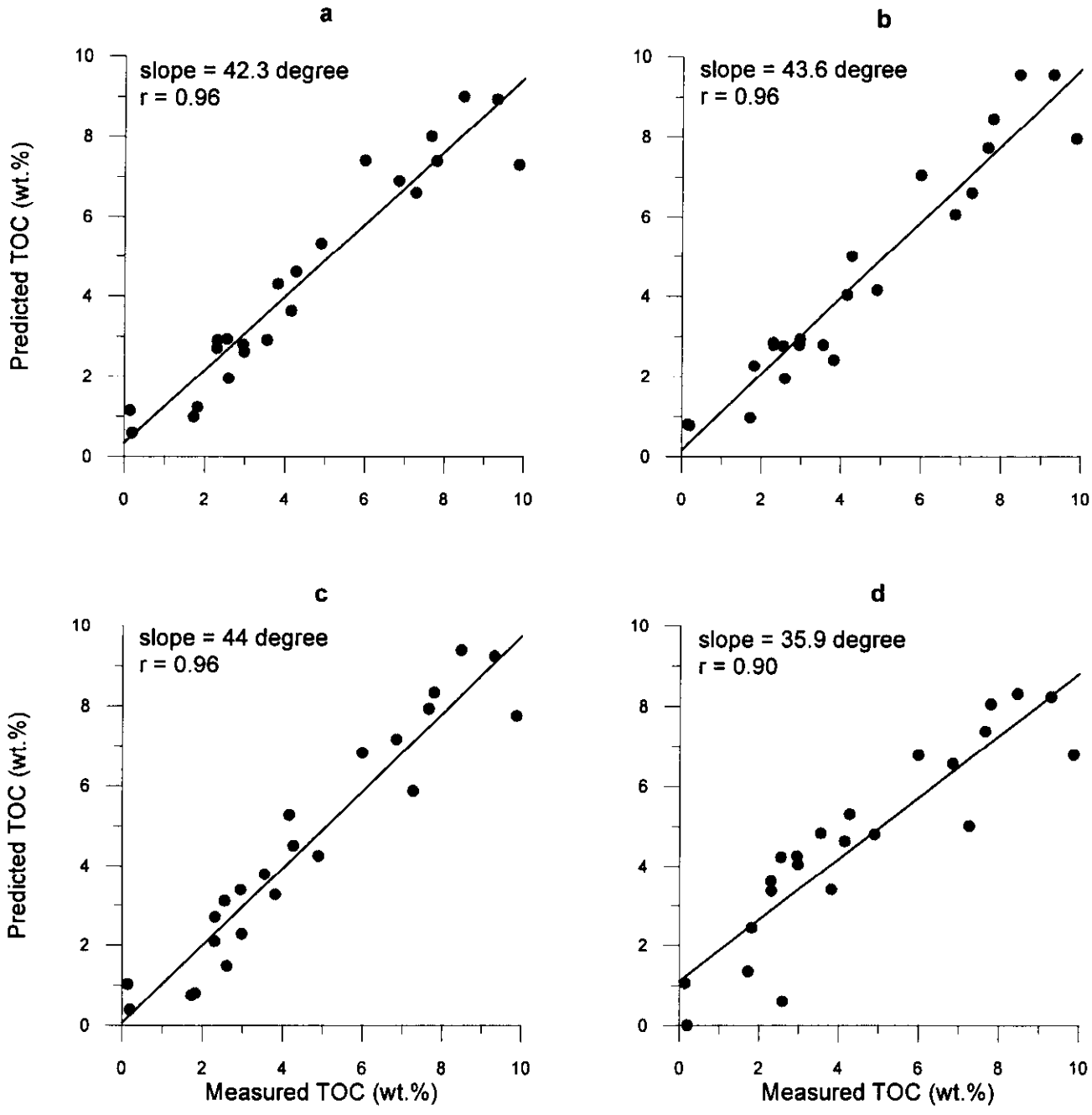

Fig. 3. Crossplots of measured TOC and predicted TOC: (**a**) prediction with the GRNN trained in mode 1; (**b**) prediction with the GRNN trained in mode 2; (**c**) prediction with a BP-CNN (1 hidden layer of four units); (**d**) prediction with the equation from multiple linear regression.

the BP-CNN training we used a rather small value to update the weights. That small value and the iterative process in BP-CNN training resulted in slightly smaller errors in predicting TOC in the 2 - 4 wt.% range. Several runs of training indicate that the architecture and the training parameters we used do make the network converge to the global minimum. However, much more time and effort were involved to determine the architecture of the BP-CNN and the training parameters. With our example 1, one training run with a GRNN using the "holdout" method in order to automatically find the optimal σ takes 1 minute and 26 seconds on a SPARC 10 SUN workstation, and only one training run is needed. If the user provides the σ value, it only takes 1 second to finish the training with a GRNN. With a BP-CNN, one training run takes 2 minutes and 42 seconds on the same machine, and usually several runs are needed to ensure satisfactory results.

Using the same training data set, we established a predictive equation through multiple linear regression analysis (Davis, 1986). When the linear predictive equation was applied to the testing data set, the prediction (Figure 3d) is poorer than those made by the GRNNs (Figure 3a, b) and the BP-CNN (Figure 3c). This can be explained by the fact that the relationship between rock's TOC and well-log responses is nonlinear.

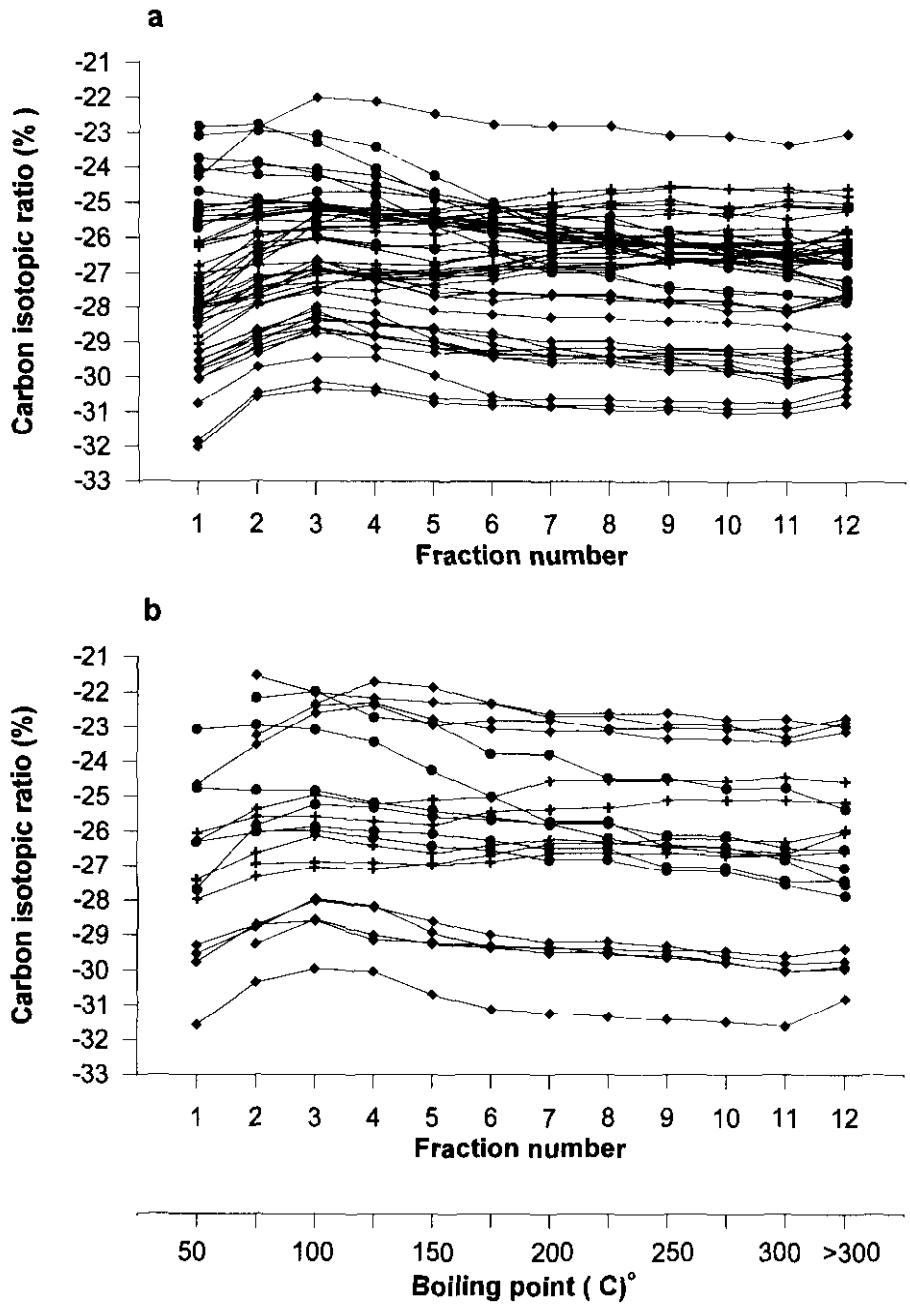

Fig. 4. Carbon isotopic profiles of three different types of crude oil from 16 basins around the world (data compiled by Chung et al., 1990). Cross stands for carbonate oil, circle for deltaic oil and diamond for marine shale oil. (*a*) *Profiles for training the GRNN (N = 49).* (*b*) *Profiles for testing the* training of the trained GRNN (N = 20). Note some profiles for this purpose are incomplete patterns.

Example 2: Pattern recognition of source characteristics of marine oils

Identification of the relative contributions of a number of source rocks to oil found in giant hydrocarbon accumulations can be generalized as a pattern recognition problem. Identification of source rock type is important for correlating oils and for understanding the history of the source rocks' maturation, hydrocarbon expulsion and migration. Usually crossplots between API gravity, sulfur content, pristane/phytane (Pr/Ph) ratio and ratio of the saturated-to-aromatic nonvolatile hydrocarbon (SAT/ARO) can be used for classification of oil genesis (i.e., marine, deltaic or carbonate) (Schoell, 1984; Sofer, 1984; Chung et al., 1994). Alternatively, a new procedure using carbon isotopic profiles can be employed (Chung et al., 1994). A carbon isotopic profile is a plot of carbon isotopic ratios of eleven crude oil distillation fractions and the residuum against the fraction number. Each fraction represents a 25°C increment in the temperature range of 50 to 300°C (Figure 4).

Chung et al. (1994) studied 69 carbon isotopic profiles of crude oil of different types (carbonate oil, marine share oil and deltaic oil) from 16 basins around the world (Figure 4).

They summarized the characteristics of each profile type and offered an explanation of the differences. As shown in Figure 4, there are differences among the three types of carbon isotopic profile, but overlaps among them are also obvious. If a profile from a crude oil sample of unknown source rock is plotted on top of this diagram, it may not be easy to determine its type. To classify crude oil in a consistent and automatic manner, we used 49 complete profiles from the data set shown in Figure 4a to train a GRNN. Another 20 profiles shown in Figure 4b, including seven incomplete ones with the ratios of the first and second fractions missing, are used to test the trained GRNN. After being trained in mode 1, the GRNN found an optimal σ (0.02) from a starting estimated σ (0.5) (Figure 2b) with 17 pattern units. After being trained in mode 2, the GRNN found an optimal σ (0.013) from a starting estimated σ (0.5) (Figure 2b) with 19 pattern units. When the trained GRNN are fed with testing profiles, they give very good classification for these profiles (Table 1), compared to human classification which is based on such independent geochemical data as API gravity, sulfur content, Pr/Ph and SAT/ARO ratios (Chung et al., 1994). Both of the GRNNs trained in different modes misclassified only one testing profile out of the 20 cases and this one is incomplete. Of the seven incomplete patterns, the GRNN is able to give correct answers for six. Considering the small size of the training data set, the GRNN's pattern recognition ability for the application is very powerful. Also, its ability to recognize incomplete patterns is particularly useful for solving some geological problems which may not be handled with conventional pattern recognition approaches or the BP-CNN.

Table 1. Comparison between human and GRNN classifications of the testing carbon isotopic profiles *.

Profile No.**	Human	GRNN (mode 1)	GRNN (mode 2)
1	1	1.001435	1.000000
2	1	1.000526	1.000000
7	1	1.000131	1.000000
12	1	1.000038	1.000000
17	1	1.000001	1.000000
22	3	3.000000	3.000000
25	3	1.966177	2.000001
28	3	2.887807	2.999974
33	3	3.000000	3.000000
38	3	2.998791	3.000000
40	3	2.998169	3.000000
44	2	2.000000	2.000000
46	2	2.000000	2.000000
50	2	2.000000	2.000000
55	2	2.000000	2.000000
60	2	2.000000	2.000000
65	2	2.000001	2.000001
66	2	2.000001	2.000001
68	2	2.000001	2.000001
69	2	2.000001	2.000001

* 1 = carbonate oil; 2 = marine shale oil; 3 = deltaic oil.
** same as numbered in Table 1 of Chung et al. (1994).

Concluding Remarks

The simple GRNN algorithm permits a more efficient CNN than the BP-CNN. With GRNN, it is not necessary to determine the architecture of the network and several parameters through tedious trial runs. If automatic searching of the smoothing factor (σ) is not needed, a GRNN can be trained in just one pass and in a much reduced time. Even if automatic searching for optimal σ is activated, a GRNN is still more computationally efficient than a BP-CNN. More importantly, GRNN will not converge to local minima. One of our examples shows that, with increasing computational time and by not converging to a local minimum, the BP-CNN made a slightly better prediction than the GRNN. With more complex problems, it would be safer and more efficient to use a GRNN.

We have applied the GRNN to two geological problems. It performs well with a small number of training data sets (example 2) and it appears that the choice of distance calculation methods is not important to the performance of the trained GRNN. Our trained GRNN is able to recognize incomplete patterns successfully. Although limited, our applications indicate that GRNN is a good alternative to the more widely used BP-CNN and promises vast potential as a tool for solving geological problems of variety of types.

References

Agterberg, F.P. and Bonhan-Carter, G.F., Eds., 1990, Statistical applications in the earth sciences: Geol. Surv. Can., Paper 8909, 588p.

_____and Griffiths, C.M., 1991, Computer applications in stratigraphy 1989/1990: a review: Computer and Geosciences **17**, 1105-1118.

Archie, G.E., 1942, The electrical resistivity log as aid in determining some reservoir characteristics: Trans. Soc. Petr. Eng., Am. Inst. Min., Metall. Petr. Eng. **146**, 54-62.

Ash, T., 1989, Dynamic node creation in backpropagation networks: ICS Rep. 8901, Institute of Cognitive Science, Univ. of California, San Diego, La Jolla, California, 11p.

Baldwin, J.L., Otte, D.N. and Wheatley, C.L., 1990, Application of neural network to problem of mineral identification from well logs: The Log Analyst **3**, 279-293.

Cacoullos, T., 1966, Estimation of a multivariate density: Ann. Inst. Sattist. Math. (Tokyo) **18**, 179-189.

Caudill, M., 1988, Neural networks primer: Part III: AI Expert, June 1988, 53-59.

Chung, H.M., Claypool, G.E., Rooney, M.A. and Squires, R.M., 1994, Source characteristics of marine oils as indicated by carbon isotopic ratios of volatile hydrocarbons: Bull. Am. Assn. Petr. Geol. **78**, 396- 408.

Davis, J.C., 1986, Statistic and data analysis in geology, 2nd ed.: John Wiley & Sons, Inc., 646p.

Fahlman, S.E., 1988, Fast learning variations on backpropagation: an empirical study, *in* Touretzky, D.S., Hinton, G. and Sejnowski, T., Eds., Proc. 1988 connectionist models summer school: Morgan Kaufmann Publ., 38-51.

Herron, S. L. and Le Tendre, L., 1990, Wireline source-rock evaluation in the Paris Basin, *in* Huc, A.Y., Ed., Organic facies: Am. Assn. Petr. Geol., Studies in Geology No. 30, 57-71.

Hertz, J., Krogh, A. and Palmer, R.G., 1991, Introduction to the theory of neural computation: Addison-Wesley Publ. Co., 327p.

Huang, Z., Williamson, M.A., Fowler, M. and McAlpine, K.D., 1994, Predicted and measured petrophysical and geochemical characteristics of the Egret Member oil source rock, Jeanne d'Arc Basin, offshore eastern Canada: Marine and Petr. Geol. **11**, 294-306.

Meyer, B.L. and Nederlof, M.H., 1984, Identification of source rocks on wireline logs by density/resistivity and sonic transit time/resistivity crossplots: Bull. Am. Assn. Petr. Geol. **68**, 121-129.

Osborne, D.A., 1992, Neural networks provide more accurate reservoir permeability: Oil and Gas J., Sept. 28, 80-83.

Parzen, E., 1962, On estimation of a probability density function and mode: Ann. Math. Stat. **33**, 1065-1076.

Passey, Q.R., Creaney, S., Kulla, J.B., Moretti, F.J. and Stroud, J.D., 1990, A practical model for organic richness from porosity and resistivity logs: Bull. Am. Assn. Petr. Geol. **74**, 1777-1794.

Pulli, J.J. and Dysart, P.S., 1990, An experiment in the use of trained neural networks for regional seismic event classification: Geophys. Res. Lett. **17**, 977-980.

Rogers, S., Fang, J.H., Karr, C.L. and Stanley, D.A., 1992, Determination of lithology from well logs using a neural network: Bull. Am. Assn. Petr. Geol. **76**, 731-739.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986, Learning internal representations by error propagation, *in* Rumelhart, D.E. and McClelland, J.L., Eds., Parallel distributed processing: v. 1, foundations, M.I.T. Press, 318-362.

Schmoker, J.W. and Hester, T.C., 1983, Organic carbon in Bakken Formation, United States portion of Williston basin: Bull. Am. Assn. Petr. Geol. **67**, 2165-2174.

_____and _____, 1989, Oil generation inferred from formation resistivity – Bakken Formation, Williston basin, North Dakota: Trans., 30th Ann. Logging Symp., Soc. Prof. Well Log Analysts, paper H.

Schoell, M., 1984, Stable isotopes in petroleum research: Advances Petr. Geochem. **1**, 215-245.

Sclater, J.G. and Christie, P.A.F., 1980, Continental stretching: an explanation of the post-mid-Cretaceous subsidence of the central North Sea Basin: J. Geophys. Res. **85**, 3711-3739.

Simaan, M. and Aminzadeh, F., Eds., 1989, Artificial intelligence and expert systems in petroleum exploration: Advances in geophysical data processing, v. 3: JAI Press Inc.

Sofer, Z., 1984, Stable carbon isotopic compositions of crude oils: application to source depositional environments and petroleum alteration: Bull. Am. Assn. Petr. Geol. **68**, 31-49.

Specht, D.F., 1990, Probabilistic neural networks: Neural Networks **3**, 109-118.

_____, 1991, A general regression neural network: Inst. Electr. Electron. Eng., Trans. on Neural Networks **2**, 568-576.

_____and Shapiro, P.D., 1990, Training speed comparison of probabilistic neural networks with back-propagation networks: Proc., Internat. Neural Networks Conf. (Paris, France), v. 1, 440-443.

Stephen, J.J., 1990, Neural network design and the complexity of learning: M.I.T. Press, 150p.

Wang, L.X., 1992, A neural detector for seismic reflectivity sequences: Inst. Electr. Electron. Eng., Trans. on Neural Networks **3**, 338-340.

Waxman, M.H. and Smith, L.J.M., 1968, Electrical conductivities in oil-bearing shaly sands: J. Soc. Petr. Eng., June 1968, 107-122.

Wu, X. and Zhou, Y., 1992, Reserve estimation using neural network techniques: Computer and Geosciences **19**, 567-575.

Wyllie, M.R.J., Gregory, A.R. and Gardner, G.H.F., 1958, An experimental investigation of factors affecting elastic wave velocities in porous media: Geophysics **23**, 459-493.

APPENDIX 1

Data for modelling relationship between well-log responses and TOC (from Passey et al., 1990).

Depth (m)	Gamma (API)	Resistivity (ohm-m)	Sonic (us/ft)	TOC (wt.%)	
Well A (LOM 6-7)					
2249	171	0.93	133	9.31	*
2252	175	0.97	131	10.24	
2255	195	0.95	132	9.59	
2262	183	0.79	135	8.2	
2268	187	0.78	138	8.78	
2274	181	0.85	132	8.47	*
2278	185	0.59	140	9.3	
2297	146	0.49	134	6.16	
2307	153	0.55	133	6.32	
2313	145	0.77	123	5.94	
2321	141	0.64	129	6	*
2324	150	0.6	130	7.04	
2329	154	0.65	132	9.5	
2336	160	0.73	117	5.82	
2339	159	0.67	130	9.49	
2342	154	0.68	131	7.66	*
2347	157	0.81	131	9.8	
2353	156	1.02	136	11.3	
2362	151	0.77	130	7.76	
2385	120	1.08	124	6.04	
2394	116	1.19	116	2.42	
2414	72	0.97	103	1.82	*
2426	103	0.83	106	2.02	
2435	99	0.73	104	2.8	
Well B (LOM 7-8)					
2950	145	6.12	126	9.3	
2950	150	3.82	125.6	7.8	*
2952	172	3.78	122.6	8.75	
2953	189	2.9	109.6	6.15	
2954	156	2.76	105.6	6.35	
2955	168	2.78	110.6	6.85	*
2956	178	3.51	121.6	10.1	
2957	166	3.34	123.6	8.75	
2958	159	3.19	114.6	3.25	
2964	109	3.49	108.6	5.55	
2967	113	3.49	112.6	5.6	
2967	107	3.92	110.6	4.28	*
2967	96	5.34	82.6	3.8	
2968	67	6.93	73.6	3.12	
2970	107	2.67	101.6	1.38	
2973	75	1.7	88.6	0.84	
2973	72	1.73	88.6	1.73	*
2974	63	2.07	79.6	1.29	
2975	65	2.37	90.6	1.69	
2984	57	1.65	85.6	1.15	
Well C (LOM 7)					
2688	166	4.68	110	9.88	*
2709	158	3.42	109	9.1	
2729	148	4.57	99	8.62	
2747	169	2.33	114	8.54	
2753	179	2.03	115	8.18	
3771	187	2.96	114	9.18	
2792	119	2.68	108	7.5	
2817	126	2.1	94	3.82	*
2827	109	2.58	84	3.54	
2831	157	3.09	102	5.4	
Well D (LOM 9)					
2090	84	5.77	82	2.13	
2120	60	9.6	68	0.36	

Depth (m)	Gamma (API)	Resistivity (ohm-m)	Sonic (us/ft)	TOC (wt.%)	
2150	100	8.58	89	1.51	
2155	74	6.94	74	2.6	*
2160	78	7.32	75	2.6	
2180	99	4.46	87	2.57	
2190	91	5.48	80	1.51	
2200	98	4.93	83	1.51	
2240	105	10.2	100	4.16	*
2245	102	12.64	98	3.64	
2250	114	11.41	104	4.42	
2264	96	39.22	101	5.19	
2270	92	48.51	96	7.27	*
2280	88	54.07	99	8.05	
Well E (LOM 8-9)					
3322	121	1.59	109	3.42	
3325	114	1.76	106	2.42	
3337	98	1.77	106	2.98	*
3355	127	2.25	103	3.04	
3365	159	3.03	110	5.32	
3367	163	3.66	106	5.4	
3380	120	4.8	100	3.86	
3385	133	3.8	100	4.9	*
3410	148	3.45	105	4.5	
3418	128	9.37	100	9.45	
3422	105	4.32	92	3.74	
3429	112	4.84	90	1.99	
3445	94	1.86	80	0.14	*
3452	94	1.8	83	0.06	
3456	101	1.92	82	0.38	
Well F (LOM 10-11)					
4023	60	1.89	82	0.26	
4035	66	3.35	82	0.2	
4038	58	2.67	70	0.2	*
4044	67	1.69	83	0.22	
4048	78	2.49	74	2.46	
4051	118	5.83	108	1.64	
4052	104	4.16	98	2.84	
4054	99	3.65	96	2.3	*
4060	115	7.97	86	1.62	
4061	130	10.21	93	3.58	
4063	121	7.92	82	2.26	
4066	118	9.16	92	2.54	
4067	117	9.42	85	4.12	
4069	112	13.54	90	2.56	*
4078	140	15.15	84	2.14	
4080	143	15.48	82	2.96	
4081	142	14.23	81	3.28	
4083	169	15.27	77	2.96	*
4090	141	15.43	84	2.76	
4092	143	19.09	84	3.54	
4093	135	18.05	81	2.64	
4096	109	12.91	68	3.04	
4098	149	11.86	88	3.56	*
4099	141	13.58	82	2.18	
4101	135	8.48	79	2.96	
4102	131	8.36	79	2.85	
4104	136	8.73	77	2.48	
4105	138	8.12	80	2.32	*
4107	121	11.33	71	2.06	
Theoretical values					
1400	30	0.8729307	96.66521	0	
1800	30	1.083396	92.45103	0	
2200	30	1.344606	88.66827	0	
2600	30	1.668793	85.27275	0	
3000	30	2.071144	82.22485	0	
3400	30	2.570501	79.48896	0	
3800	30	3.190255	77.03315	0	
4200	30	3.959434	74.82875	0	
4600	30	4.914063	72.85002	0	
5000	30	6.098855	71.07385	0	
1000	15	0.4313075	115.6857	0	
1400	15	0.5143076	109.9502	0	
1800	15	0.6132799	104.6979	0	
2200	15	0.7312983	99.88802	0	
2600	15	0.8720279	95.48331	0	
3000	15	1.039839	91.44967	0	
3400	15	1.239944	87.75581	0	
3800	15	1.478557	84.37312	0	
4200	15	1.763087	81.27538	0	
4600	15	2.102372	78.4386	0	
5000	15	2.506948	75.84078	0	
1000	15	0.4313075	113.5599	0	
1400	15	0.5143076	107.6581	0	
1800	15	0.6132799	102.2535	0	
2200	15	0.7312983	97.30415	0	
2600	15	0.8720279	92.77173	0	
3000	15	1.039839	88.62112	0	
3400	15	1.239944	84.82016	0	
3800	15	1.478557	81.33939	0	
4200	15	1.763087	78.15182	0	
4600	15	2.102372	75.23279	0	
5000	15	2.506948	72.55965	0	
1000	90	0.1229447	109.6769	0.8	
1400	90	0.3145346	101.4323	0.8	
1800	90	0.509715	94.70918	0.8	
2200	90	0.7094869	89.22672	0.8	
2600	90	0.9151946	84.75597	0.8	
3000	90	1.128684	81.11025	0.8	
3400	90	1.352562	78.13731	0.8	
3800	90	1.590658	75.71298	0.8	
4200	90	1.848883	73.73603	0.8	
4600	90	2.137022	72.1239	0.8	
5000	90	2.473009	70.80927	0.8	

* Testing data.