Rainfall Prediction Using Neural Fuzzy Technique

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

[1]*School of Information Technology*
Murdoch University, Western Australia
Email: { k.wong | t.gedeon} @murdoch.edu.au

[2]*School of Petroleum Engineering*
University of New South Wales, Sydney, Australia
Email: pm.wong@unsw.edu.au

[3]*School of Electrical and Computer Engineering*
Curtin University of Technology, Western Australia
Email: tfungcc@cc.curtin.edu.au

Abstract: This paper constructs numeral fuzzy rule bases with the aid of Self-organising Map (SOM) and Backpropagation Neural Networks (BPNNs). These fuzzy rule bases are then used to perform the spatial interpolation on the 367 rainfall data in Switzerland based on the information found in the nearby 100 locations. The SOM is first used to classify the data. After classification, BPNNs are then use to learn the generalization characteristics from the data within each cluster. Fuzzy rules for each cluster are then extracted. The fuzzy rules bases are then used for rainfall prediction.

Keywords: Spatial interpolation, artificial neural networks, self-organising map, backpropagation neural networks, fuzzy rule extraction.

1. INTRODUCTION

Artificial Neural Networks (ANNs) have emerged as an option for spatial data analysis (Friedman, 1994; Lee et. al, 1998). The observation sample that is used to derive the predictive model is known as training data in an ANN development. The independent variables, or the predictor variables, are known as the input variables and the dependent variables, or the responses, are known as the output variables.

In supervised learning (Kartalopulos, 1996), an ANN makes use of the input variables and their corresponding output variables to learn the relationship between them. Once found, the learned ANN is then used to predict values for the output variables given some new input data set. For unsupervised learning, an ANN will only make use of the input variables and attempts to arrange them in a way that is meaningful to the analyst.

ANN analysis is quite similar to statistical approaches in that both have learning algorithm to help them realise the data analysis model. However, an ANN has the advantages of being robust with the ability to handle large amounts of data. Novice users can also easily understand the use of an ANN. An ANN also has the ability to handle very complex functions (Cherkassky et. al, 1994). The main limitation of using ANN is that the data analysis model built may not be able to be interpreted.

Fuzzy logic is also becoming popular in dealing with data analysis problems that are normally handled by statistical approaches or ANNs (Kosko, 1997). However, conventional fuzzy system systems do not have any learning algorithm to build the analysis model. Rather, they make use of human knowledge, past experience or detailed analysis of the available data by other means in order to build the fuzzy rules for the data analysis. The advantages of using fuzzy system are the ability to interpret the analysis model built and to handle fuzziness in the data. The data analysis model can also be changed easily by modifying the fuzzy rule base. The major limitation is the difficulty in building the fuzzy rules due to lack of learning capability.

ANNs and fuzzy logic are complementary technologies in designing an intelligent data analysis approach (Williams, 1994). That suggests combining the two (Nauck, 1995). For example, fuzzy logic could be used to enhance the learning capabilities or performance of the neural network. In another approach, a neural network and fuzzy system could be integrated into a single architecture. However, a human analyst may still have difficulties understanding the analysis model computed. Analysis of the prediction model is also very time consuming. Therefore, it was one of the prime objectives of the paper presented to find a better way of combining the advantages of the ANN and fuzzy logic such that these particular problems could be overcome.

2. NEURAL FUZZY SPATIAL INTERPOLATION

ANN and fuzzy logic are complementary technologies for the designing of spatial interpolation tools. However, there are many ways that the combination can be implemented (Nauck, 1995). Table 1 shows the different ways that ANN and FL can work together. It is important to observe the characteristics under each class so as to search for a better technique that the analyst will be comfortable with.

Table 1: Different ways to combine ANN and fuzzy logic

Techniques	Description
Fuzzy Neural Networks	Use fuzzy methods to enhance the learning capabilities or performance of ANN
Concurrent Neuro-Fuzzy	ANN and Fuzzy systems work together on the same task without any influence on each other
Cooperative Neuro-Fuzzy	Use ANN to extract rules and then it is not used any more
Hybrid Neuro-Fuzzy	ANN and Fuzzy are combined into one homogeneous architecture

The Cooperative Neuro-Fuzzy technique is selected as the more appropriate technique to be used in this application. The reasons are as follow. As the BPNN can generalise from the data through some learning algorithm, the spatial interpolation function could be realised automatically. This will also enable the fuzzy rules to cover the whole universal of discourse, so that they can be used to approximate data that are not present in the training set. As fuzzy rules are closer to human reasoning, the analyst could understand how the interpolation model performs prediction. If necessary, the analyst could also make use of his/her knowledge to modify the interpolation model.

2.1 *Self-organising Map (SOM)*

In most spatial analysis, the first step is to classify the available data into different classes so that the data are split into homogeneous sub-populations (Lee et. al, 1998; Huang et. al, 1998; Fung et. al, 1997). The objective in this step is to make use of an unsupervised learning algorithm to sub-divide the whole population. Self-organising Map (SOM) is selected for this purpose mainly because it is a fast, easy and reliable unsupervised clustering technique.

SOM is designed with the intention to closely simulate the various organisations found in various brain structures and has a close relationship to brain maps (Kohonen, 1990; Kohonen, 1995). Its main feature is the ability to visualise high dimensional input spaces onto a smaller dimensional display, usually two-dimensional. In this discussion, only two-dimensional arrays will be of interest. Let the input data space $\Re^n$ be mapped by the SOM onto a two-dimensional array with i nodes. Associated with each i node is a parametric reference vector $m_i=[\mu_{i1},\mu_{i2},\dots\ ,\mu_{i2}]^T \in \Re^n$, where μ_{ij} is the connection weights between node i and input j. Therefore, the input data space $\Re^n$ consisting of input vector $X=[x_1,x_2,..,x_n]^T$, ie $X \in \Re^n$, can be visualised as being connected to all nodes in parallel via a scalar weights μ_{ij}. The aim of the learning is to map all the n input vectors X_n onto m_i by adjusting weights μ_{ij} such that the SOM gives the best match response locations.

SOM can also be said to be a nonlinear projection of the probability density function $p(X)$ of the high dimensional input vector space onto the two-dimensional display map. Normally, to find the best matching node i, the input vector X is compared to all reference vector m_i by searching the smallest Euclidean distances $\| X - m_i \|$, signified by c. Therefore,

$$c = \arg\min_i \{\| X - m_i \|\} \qquad (1)$$

or

$$\| X - m_c \| = \min_i \{\| X - m_i \|\} \qquad (2)$$

During the learning process, beside the node that best matches the input vector X is allowed to learn, those nodes that are close to the node up to a certain distance will also be allowed to learn. The learning process is expressed as:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[X(t) - m_i(t)] \qquad (3)$$

where t is discrete time coordinate
and $h_{ci}(t)$ is the neighbourhood function

After the learning process has converged, the map will display the probability density function $p(X)$ that best describes all the input vectors space. At the end of the learning process, an average quantisation error of the map will be generated to indicate how well the map matches the entire input vectors X_n. The average quantisation error is defined as:

$$E = \int \| X - m_c \|^2 \; p(X)dX \qquad (4)$$

After the 2-dimensional map has been trained, the reference vectors that were used in the nodes of the map can also obtain. In spatial interpolation, the output value is the interested feature from the neighbouring location, we proposed here to construct the clustering boundaries based on the output reference vector of the nodes. The rule of thumb for deciding on the clustering boundaries is mainly by performing the distance measure between the neighbouring reference values. If the distance measure between the present reference node and the neighbouring nodes is high, that suggests another cluster.

2.2 *Backpropagation Neural Network (BPNN)*

After the set of available training data has been sub-divided, BPNN are trained in each cluster to predict only data within the cluster. Therefore, if SOM identified c clusters, then c number of BPNNs need to be trained. When a BPNN (Rumelhart et. al, 1986) is used in spatial analysis, the observations obtained from the neighbouring are used as the training data, thus it is a supervised learning technique. The input neurons of the BPNN in this case correspond to the x and y position coordinates, and the output neuron is assigned to z, the rainfall measurement. The BPNN has a number of layers. The input layer consists of all the input neurons and the output layer just the output neuron. There are also one or more hidden layers. All the neurons in each layer are connected to all the neurons in next layer with the connection between two neurons in different layers represented by a weight factor.

The objective of training the BPNN is to adjust the weights so that the application of a set of inputs interpolates the output. When BPNN performs learning, a calculation is done to obtain the actual output set by proceeding in order from the input layer to the output layer. At the output, the total error on each output neuron, which is the sum of squares of the differences between the desired output and the computed output is calculated. This value is used in a learning algorithm to update the weights and the process is back propagated through the network. In order to avoid the BPNN from memorising, cross validation is used to ensure its generalization capability.

Once the modification of all the connection weights is done, a new set of outputs can be computed and subsequently a new total error will be obtained. This back-propagated process repeats until the value of the total error is below some particular threshold. At this stage, the BPNN is considered to have learned the function. After the BPNN has learned and generalised from the training data, it is then used to construct the fuzzy rules bases.

2.3 *Fuzzy Rule Extraction*

As all the BPNNs have generalized from the training data, the next step is to extract the knowledge learned by the BPNNs. In this case, it is the same as the previous section, we will have to extract c number of fuzzy rules bases.

The following algorithm outlines the steps in extracting the fuzzy linguistic rules for one BPNN.

As we have to extract fuzzy rules that can cover the whole universal of discourse in order to cover the whole sample space as seen by the BPNN, for T membership functions or linguistics terms, we would have T^2 fuzzy rules as we have only two variables (x, y) in this case.

Randomly generate input variables that could cover all the possible input space as seen by the BPNN and fed it into the BPNN to obtain the rainfall measurements predicted by the BPNN.

For the two inputs (x, y), the BPNN generated input (x, y)-output (z) data pairs with n patterns are:

$$
\begin{gathered}
(x^1, y^1; z^1) \\
(x^2, y^2; z^2) \\
\vdots \\
(x^n, y^n; z^n)
\end{gathered}
$$

The number of linguistics terms T used in this fuzzy rule extraction has to be the same as the predetermined one when generating output from the BPNN. The distribution of the membership functions in each dimension of the domain in this case is evenly distributed. For ease of interpolation and computational simplicity, the shape of the membership functions used in this rule extraction technique is triangular. In this case, we will obtain for every $x \in X$,

$$A_t : X \rightarrow [0,1] \qquad (5)$$

After the fuzzy regions and membership functions have been distributed, the available input-output pairs will be mapped. If the value cuts on more than one membership function, the one with the maximum membership grade will be assigned to the value:

$$R_n \Rightarrow [x^n (A_x, \max), y^n (A_{y,} \max) : z^n (B_z, \max)] \qquad (6)$$

After all the input-output values have been assigned a fuzzy linguistic label, Mamdani type fuzzy rules are then formed (Mamdani and Assilian,1975).

After the fuzzy rules base corresponding to the BPNN for a class has been constructed, the BPNN is not used anymore when performing spatial interpolation. With these set of fuzzy rules, human analyst can now examine the behaviour of the interpolation. Changes and modification can then be performed if necessary. The fuzzy rules extracted can also handle fuzziness in the data and thus may improve the performance of the spatial interpolation. Figure 1 shows the block diagram of establishing the spatial interpolation model and Figure 2 shows the block diagram of the performing the spatial interpolation.

3. RAINFALL PREDICTION

In this case study, the data available from the AI-GEOSTATS mailing list in Italy (Dubois, 1998) is used. The data is collected on 8th May 1996 in Switzerland. 100 data locations are used as the training data and the other 367 locations data are then used to verify the prediction accuracy of the established spatial interpolation model. The two input variables used in this case is the 2D coordinate position (x, y); and the output used is the rainfall measurements (z). The digital elevation model (DEM) (v) is also available but was not used in the case study. The 100 training data points are fed into the SOM for unsupervised clustering. After clustering, the vector map for the output (z) is as shown in Figure 3. In this case study, we have use 10 by 10 two-dimensional map. 10 by 10 map was selected by examining the average quantisation error. After performing the cluster boundaries determination, the classes are formed as shown in Figure 4.

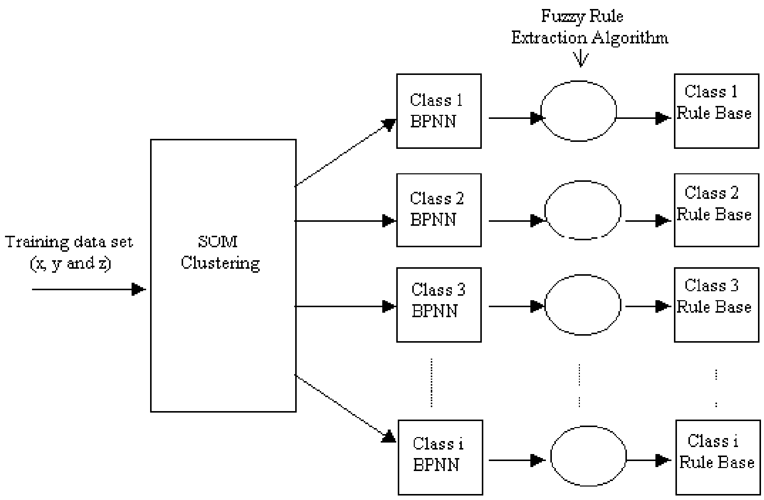

Figure 1: Establishing the spatial interpolation model

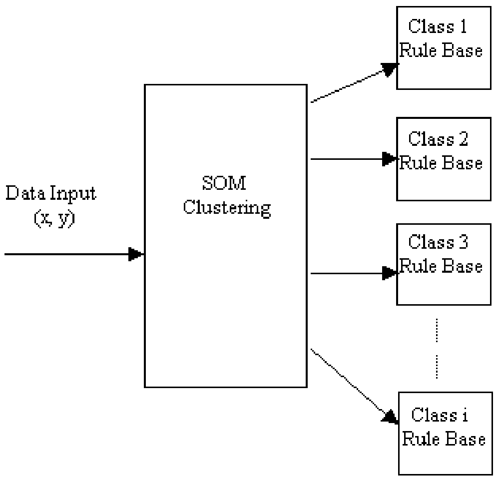

Figure 2: Performing spatial interpolation from the fuzzy rules bases

	0	1	2	3	4	5	6	7	8	9
0	49.07	68.49	101.39	146.15	195.26	251.58	278.66	346.97	305.80	248.81
1	68.98	119.63	127.22	132.80	202.07	216.55	313.67	325.59	285.68	216.18
2	170.97	147.21	176.29	198.65	147.32	101.23	264.75	303.90	231.52	146.79
3	125.29	155.71	156.49	138.93	145.27	221.58	351.26	352.78	242.52	117.42
4	126.69	125.32	137.97	110.82	122.38	250.33	321.66	348.32	166.78	120.25
5	130.86	119.28	118.82	77.96	111.27	164.85	282.00	365.54	340.93	171.91
6	124.37	131.55	80.91	84.45	61.10	143.53	207.36	356.25	265.77	197.01
7	184.45	288.04	179.71	64.20	90.60	142.81	274.69	452.64	302.09	165.12
8	271.93	311.74	283.21	161.43	81.58	127.68	116.04	346.14	291.78	157.07
9	205.36	270.88	226.25	56.87	67.34	90.67	107.63	262.06	318.77	268.53

Figure 3: The SOM vector map for rainfall measurement

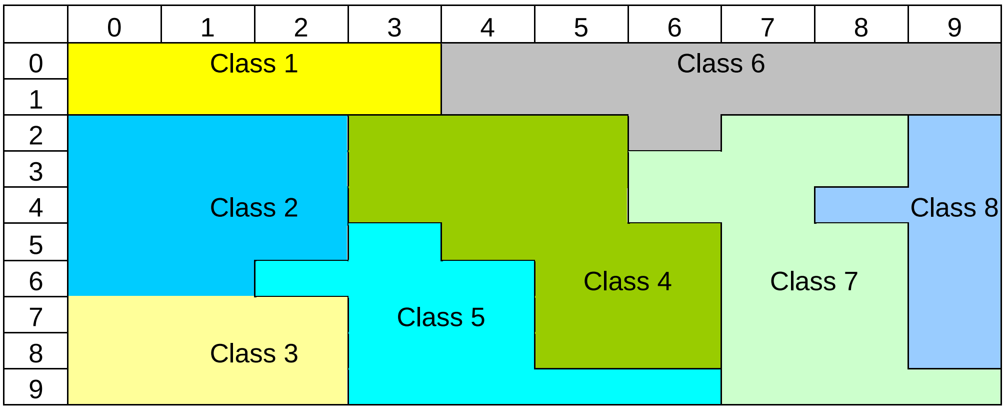

Figure 4: The cluster boundaries on the SOM 2-dimensional map

Before feeding into individual BPNN, the data need to be normalized between 0 and 1. Linear normalization is used with maximum and minimum vales unique to the class. In this case, the SOM identified a total of 8 classes. After the data has been normalized, 8 BPNNs are trained to handle their own sub-population.

After examining the maximum and minimum value of each class, the appropriate number of membership used is determined to be 7. In this case, the number of fuzzy rules extracted for each BPNN (each class) is 49, i.e. 7^2. Part of the fuzzy rules used in class 1 are shown in Figure 5, where EL is extreme low, VL is very low, L is low, M is middle, H is high, VH is very high, and EH is extreme high. With the distribution information for each linguistics term, the user can easily understand the set of fuzzy rules and understand how the prediction is performed.

```
If x= VVL and y= EL then z= VL
If x= VVL and y= VL then z= VL
If x= VVL and y= L then z= VL
If x= VVL and y= M then z= VH
If x= VVL and y= H then z= VH
If x= VVL and y= VH then z= M
If x= VVL and y= EH then z= EL
If x= VL and y= EL then z= H
If x= VL and y= VL then z= H
If x= VL and y= L then z= L
If x= VL and y= M then z= M
If x= VL and y= H then z= VH
If x= VL and y= VH then z= H
If x= VL and y= EH then z= VL
If x= L and y= EL then z= VH
```

Figure 5: Part of the fuzzy rules used to predict class 1 rainfall.

After all the 8 fuzzy rules bases have been constructed, the rainfall for the 367 locations in the testing set can then be interpolated. The minimum, maximum, mean, median and standard deviation of the 367 observed data and the interpolated data are tabulated in Table 2. The ten highest and lowest values of the predicted Z are shown in Table 3. The relative mean absolute error (MAE), root mean square error (RMSE), the correlation measure, and the relative error between the predicted and observed rainfall are shown in Table 4. Figure 6 gives a cross plot of the predicted rainfall and the observed rainfall.

Table 2: Comparison between observed and predicted rainfall

	Observed Z	Predicted Z
Min	0	12
Max	517	467
Mean	185	194
Median	162	163
Standard deviation	111	110

Table 3: The ten highest and lowest values.

10 lowest value		10 highest value	
Observed	Predicted	Observed	Predicted
0	12	517	467
0	29	503	446
0	31	493	375
0	45	445	360
0	72	444	395
1	51	434	393
5	13	434	341
6	13	432	375
8	12	429	378
13	13	415	405

Table 4: Error measures between the predicted and observed rainfall.

MAE	53.86
RMSE	72.95
Correlation Measure	0.784
Relative Error	0.31

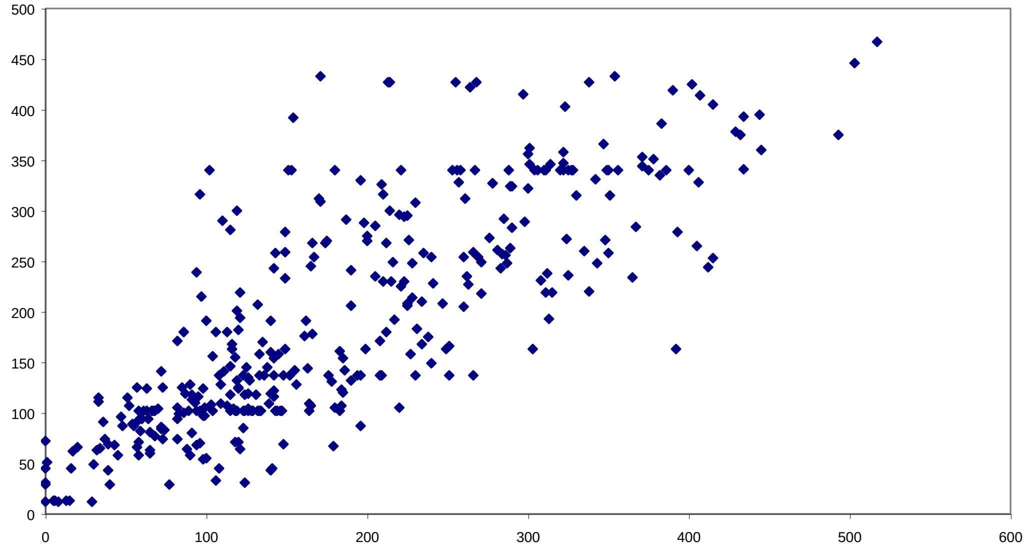

Figure 6: Plot of the 367 predicted rainfall and the observed rainfall.
(Note: The x-axis is the observed rainfall and the y-axis is the predicted rainfall)

4. CONCLUSION

In this paper, a spatial interpolation technique has been used to predict rainfall in Switzerland. This technique uses SOM to perform clustering so as to sub-divide the whole sample space into homogenous sub-population. After the classification boundaries have been identified, the whole training data set is then sub-divided into the respective classes. BPNN corresponding to each individual class is then trained using cross-validation approach. After all the BPNN has been trained, fuzzy rules bases for each class are then constructed. The case study used has shown that this method can produce reasonable rainfall prediction. The advantages of using this technique are as follow. First it makes use of the robustness and learning ability of the ANN to sub-divide and generalized from the training data. After which, the learned underlying function is then translated into fuzzy rules. With the use of fuzzy rules, the interpretability and the ability of handling fuzziness has enhanced the interpolation model. Most important of all, this technique put forward a self-learning and self-explanation spatial interpolation technique. The next phase of this research can emphasis on examining the human understandable fuzzy rules in improving the prediction results.

REFERENCES

1. Cherkassky, V., Friedman, J.H., and Wechsler, H. (1994) *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, Springer-Verlag.

2. Dubois, G. (1998) Spatial Interpolation Comparison 97, ftp://ftp.geog.uwo.ca/SIC97/intro.html

3. Friedman, J.H. (1994) "An Overview of Predictive Learning and Function Approximation", in *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, Springer-Verlag, pp. 1-61.

4. Fung, C.C., Wong, K.W., Eren, H., Charlebois, R., and Crocker, H. (1997) "Modular Artificial Neural Network for Prediction of Petrophysical Properties from Well Log Data," in *IEEE Transactions on Instrumentation & Measurement*, vol. 46(6), December, pp. 1259-1263.

5. Huang, Y., Gedeon, T.D. and Wong, P.M. (1998) "Spatial interpolation using fuzzy reasoning and genetic algorithms", *Journal of Geographic Information and Decision Analysis*, vol. 2(2), pp. 223-233.

6. Kartalopulos, S.V. (1996) *Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications*, IEEE Press.

7. Kohonen, T. (1990) "The Self-Organising Map", *Proceedings of the IEEE*, Vol. 78, No. 9, September, pp. 1464-1480.

8. Kohonen, T. (1995) *Self-Organising Map*, Springer-Verlag.

9. Kosko, B. (1997) *Fuzzy Engineering*, Prentice-Hall

10. Lee, S., Cho, S. and Wong, P.M. (1998) "Rainfall prediction using artificial neural network", *Journal of Geographic Information and Decision Analysis*, Vol. 2(2), pp. 254-264.

11. Mamdani, E.H. and Assilian, S (1975) “An experiment in linguistic synthesis with a fuzzy logic contoller,” *International Journal of Man-Machine Studies*, pp.1-13.

12. Nauck, D. (1995) “Beyond Neuro-Fuzzy: Perspectives and Directions”, *Proceedings of the Third European Congress on Intelligent Techniques and Soft Computing*, pp. 1159-1164.

13. Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) "Learning Internal Representation by Error Propagation" in *Parallel Distributed Processing*, vol. 1, Cambridge MA: MIT Press, pp. 318-362.

14. Williams, T. (1994) “Special Report: Bringing Fuzzy Logic & Neural Computing Together”, *Computer Design*, July, pp.69-84.