DECRYPTING NEURAL NETWORK DATA: A GIS CASE STUDY

R.A. Bustos and **T.D. Gedeon**

School of Computer Science Engineering
University of New South Wales, Sydney NSW 2052, Australia

ABSTRACT

The problem of data encoding for training back-propagation neural networks is well known. The basic principle is to avoid encrypting the underlying structure of the data. This is not easy in the real world, where we often receive data which has been processed by at least one previous user. The data may contain too many instances of some class, and too few instances of other classes. Then topology, and parameters settings designed. Finally, the network produces some results which need to be explained, or decrypted.

We present our experience and results on some satellite data augmented by a terrain model. The task was to predict the forest supra-type based on the available information. In this process we were forced to invent some methods to deal with very large amounts of erratically reliable data, and to produce meaningful predictions at the end.

INTRODUCTION

The choice of a suitable pattern representation is often crucial for good learning and generalisation in machine learning in general, and also for neural networks. In this case study we focus initially on the Engineering aspects of this area and then on the Scientific. Thus, we first present our experience on a set of raw data as provided to us by a Geographer, and the decisions we made based on analysis of the data. This is essentially an engineering approach. Then we present the new method we needed to discover to improve the performance of the neural network on the data.

The raw data for this study comes from a forest in New South Wales, Australia. The available information is from a rectangular grid of 244,494 points, and is a vector of 16 values, 7 from satellite images, augmented by 9 values from a terrain model derived from soil maps and aerial photography and so on. The last 5 are the forest supra-type.

The data as provided by the Geographer had already been somehow encoded. There was some available information on the encoding, as shown below:

◊ Aspect: 0: flat, 10: North, 20: NE,
30: East, ..., 70: West, 80: NE.

◊ Sin & Cos Aspect: some unknown encoding has been applied.

◊ Altitude: metres above sea level.

◊ Topographic position: 32: gully, 48: lower slope,
64: mid-slope, 80: upper slope, 96: ridge.

◊ Slope: 10: < 1%, 20: < 2.15%, 30: < 4.64%,
40: < 10%, 50: < 21.5%, 60: < 46.4%,
70: < 100%, 80: > 100%

◊ Geology descriptor: unknown encoding.

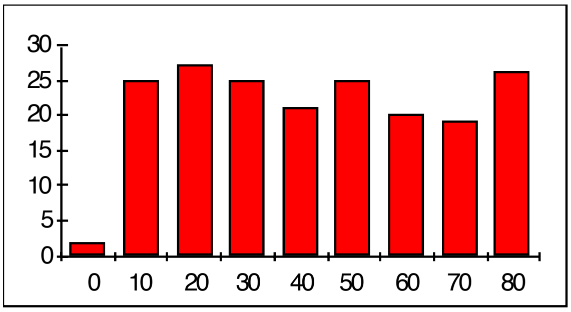
◊ Rainfall: (mm - 801)/5.

◊ Temperature: (degrees - 11)*30.

◊ Landsat tm bands 1 to 7: values in range 0 to 256, rescaled to 0 to 99.

DATA ANALYSIS (EXAMPLES)

Aspect:



◊ This is a ‘circular’ value, so AS=80, AS=10 may cause a net to generalise to 45!

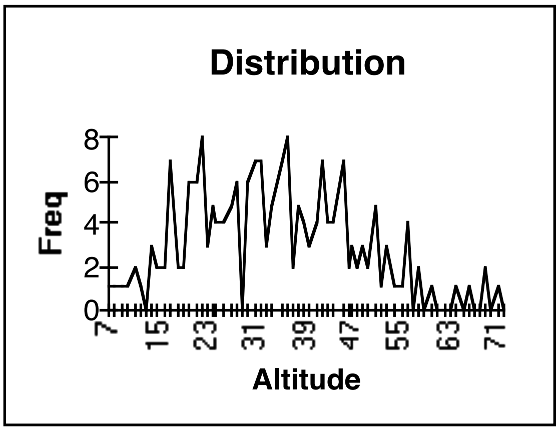
◊ The 0 value for aspect is redundant since the slope degree also encodes this information.

◊ The sin and cos of the aspect are also redundant.

Altitude

Statistics

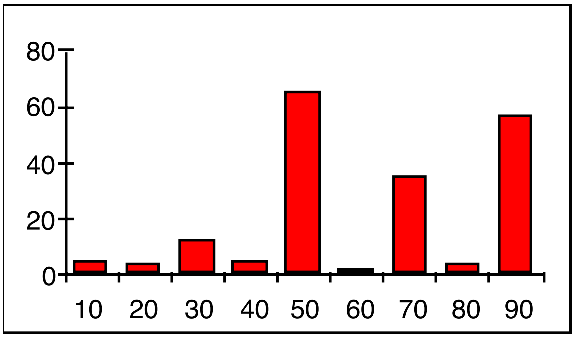
Maximum Value	71
Minimum Value	7
Average Value	34
Standard Deviation	13
Average Absolute Deviation	11



◊ Need to normalise data over the range 0 - 1 for the network, for the logistic function.

◊ Consider using statistical Z function to remove bias of lowest and highest values.

Geology descriptor:



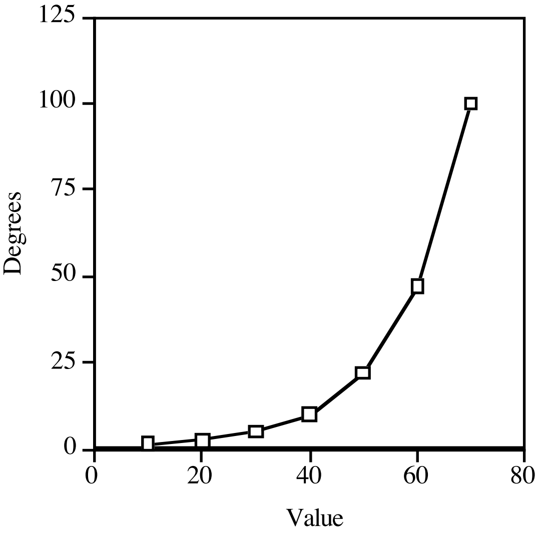
◊ 3 significant groups, not distributed normally.

◊ Patterns with ‘50’ output similar to those with ‘70’ output may cause spurious ‘60’ results.

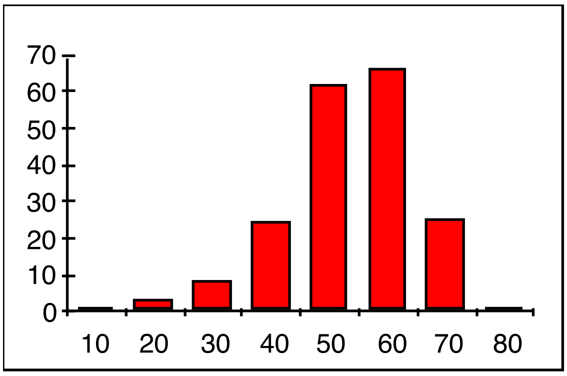
◊ No information was available on the encoding used by the geographer on this field.

Slope degree

Encoding format:



◊ The degrees increase exponentially with the values. This logarithmic encoding is unlikely to be coincidental, and we assume incorporates some domain knowledge of the Geographer.



◊ Note that the slope degree categories are not linear in their variation with the percentage of slope.

◊ Consider using a single continuously valued unit or a number of category units with soft boundaries.

<u>ENCODING DECISIONS (EXAMPLES)</u>

The encoding decisions described are the result of trialing a number of alternative representations.

Where there were no significant differences, the simplest is presented, otherwise the best result is described.

Aspect

◊ The best input coding for generalisation would be a category for each direction, with a redundant gaussian activating the input and the adjacent inputs to a lesser amount. This would require 8 inputs (too many).

◊ The raw encoding of *aspect* does not reflect the circular nature of the information.

◊ As a compromise, code the *aspect* into 4 units, using one input to represent each point of the compass.

As	Dir.	A1	A2	A3	A4	Σ activ
0	Flat	0	0	0	0	0
10	N	1	0.5	0	0.5	2
20	NE	1	1	0	0	2
30	E	0.5	1	0.5	0	2
40	SE	0	1	1	0	2
50	S	0	0.5	1	0.5	2
60	SW	0	0	1	1	2
70	W	0.5	0	0.5	1	2
80	NW	1	0	0	1	2

◊ Total activation for valid aspects is 2. The activation is apportioned in a way that should aid generalisation as there is a gradual change between directions.

Altitude:

◊ Distribution of values in the training set is fairly normal, so use a simple linear squashing function.

Slope Degree:

◊ Retain existing logarithmic encoding. Normalise to range 0 to 1 using a single continuous valued output.

Geology descriptor

◊ From the data it appears to be a nominal value. There is no particular distribution, three of the types are quite common and the others are rare.

◊ A single continuously value input is not appropriate. The categories will be represented by 4 inputs, distinguishing between the popular types and the rare ones, thus losing some information. In any case, the input vector over these inputs would be quite sparse.

VALUE	G1	G2	G3	G4
10	0.9			
20	0.9			
30	0.9			
40	0.9			
50		0.9		
60	0.9			
70			0.9	
80	0.9			
90				0.9

◊ All unmarked activations are 0.1. Cases activating G1 are similar in total to the other three categories.

OUTPUT ENCODING

The network is a classifier, and the output vectors are very sparse. This can lead to difficult learning, since the tss can improve by pushing output vectors to all 0's.

This can be avoided by equilateral coding [1], so the 5 possibilities are represented by 4 units. All units have some activation on each pattern, and the maximum distance between vectors is maintained.

Category	Unit 1	Unit 2	Unit 3	Unit 4
Scrub	0.1838	0.3174	0.3709	0.4
Dry scler.	0.8162	0.3174	0.3709	0.4
Wet-dry scler.	0.5	0.8651	0.3709	0.4
Wet scler.	0.5	0.5	0.8872	0.4
Rain Forest	0.5	0.5	0.5	0.9

◊ To retrieve the category, calculate Euclidean distance between the output vector and the above values:

$$\text{Output category} = \text{MIN}(\text{distance }(U_i))$$

where

$$\text{distance }(U_i) = \sqrt{\sum_{j=1}^{4} (y_j - u_{j_i})^2}$$

where

$$U_i = (u_{j_S},\ u_{j_{DS}},\ u_{j_{WDS}},\ u_{j_{WS}},\ u_{j_{RF}})$$

PATTERN REDUCTION

We have done some work in speeding up learning by removing outliers [2, 3]. On very large sets of patterns, such methods can be too slow to be useful.

We have also done some work on a simple heuristic method of reducing training sets to half or quarter the size and retaining and sometimes improving performance [4]. Such reduction in size of a training set speeds up training by a commensurate ratio.

RESULTS

◊ 12 hidden unit network

epochs	tss train	tss test	actual train	perform. test
750	12.9	3.9	77%	58%
1050	11.4	4.4	82%	55%
1350	9.7	4.8	84%	55%
1650	9.1	5.0	84%	51%

Twelve hidden units provided the best performance.

Note that generalisation suffers as more epochs are run.

Adding Random noise

◊ Trained with normal, and randomly distorted data in 50 epoch alternation.

Note the oscillation of the test set results as the training continues. This effect was more pronounced with higher levels of added noise (or *randomness*), this observation lead to our decreasing amplitude noisy training.

epoch	tss train	tss test	actual train	actual test	randomness
200	19.2	3.8	58%	55%	0.4
300	17.9	3.5	69%	58%	0.3
400	16.7	3.1	70%	62%	0.2
500	16.3	3.6	67%	44%	0.15
600	18.9	3.6	72%	58%	0.1
700	16.9	3.4	71%	55%	0.05
800	13.2	3.3	77%	68%	0.0

CONCLUSION

We have shown using our example of Geographical Information System (GIS) data pre-encoded in a fashion appropriate to the use of the data in Geography.

To produce a neural network which performed as well as the standard statistical results of 57% accuracy on the test set, some effort was required on encoding of the data suitably for the network, as well as using pattern reduction techniques to reduce the large number of points without losing significant information, and hence again degrading network performance.

To outperform the statistical result, we developed a method for alternating noisy training with normal training.

REFERENCES

1. Masters, T, *Practical Neural Network Recipes in C*, Academic Press, Boston, 1993.

2. Slade, P and Gedeon, TD “Bimodal Distribution Removal,” in Mira, J, Cabestany, J and Prieto, A, *New Trends in Neural Computation*, pp. 249-254, Springer Verlag, Lecture Notes in Computer Science, vol. 686, 1993.

3. Wong, PW and Gedeon, TD “The Error Sign Testing Method in Pattern Reduction,” *International Journal of Systems Research and Information Science*, (in press), 1994.

4. Gedeon, TD and Bowden, TG, “Heuristic Pattern Reduction II,” *Proceedings ICCS*, Invited Position Paper, pp. 3.43-3.45, Beijing, 1993.