Proc. 1996 Australian New Zealand Conf. on Intelligent Information Systems, 18-20 November 1996, Adelaide, Australia. Editors, Narasimhan and Jain

Indicators of Input Contributions: Analysing The Weight Matrix

Tamás D. Gedeon
School of Computer Science and Engineering
The University of New South Wales
Sydney 2052 Australia

ABSTRACT

The problem of data encoding and feature selection for training back-propagation neural networks is well known. The basic principles are to avoid encrypting the underlying structure of the data, and to avoid using irrelevant inputs. 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. Real data sets often include many irrelevant or redundant fields.

This paper examines the use of the weight matrix of the trained neural network itself to determine which inputs are significant. A novel technique is introduced, and compared with two other techniques from the literature. 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. A brute force technique eliminating randomly selected inputs was used to validate our approach.

INTRODUCTION

The initial network topology was 16-10-5, being sixteen inputs, ten hidden neurons, and five output neurons. The raw data for this study comes from an area in the Nullica State Forest on the south coast of New South Wales, Australia. The available information is from a rectangular grid of 179,831 pixels 30 m by 30 m, and is a vector of 16 values [1]. Each pixel has a value for altitude, aspect, slope, geology, topographic position, rainfall, temperature (from a terrain model derived from soil maps, aerial photography and so on), and Landsat TM bands 1 to 7. The outputs are the forest supra-type, being *scrub*, *dry sclerophyll*, *wet-dry sclerophyll*, *wet sclerophyll*, and *rainforest*. For the purpose of training 190 detailed sample plots have been surveyed [4]. This data gives us classifications for 190 of the pixels in the field area, and have used 150 for training the neural network and retained 40 for testing.

The network was trained using error-backpropagation [5]. All connections are from units in one level to units in the next level, with no lateral, backward or multi-layer connections. Each unit is connected to each unit in the preceding layer by a simple weighted link. The network is trained using a training set of input patterns with desired outputs, using the back-propagation of error measures. The network is tested using a validation set of patterns which are never seen by the network during training and thus can provide a good measure of the generalisation capabilities of the network. Thus all results quoted in this paper are for the test set. We have used the basic sigmoid logistic activation function, $y = \left(1 + e^{-x}\right)^{-1}$, though this is not essential to the substance of our results.

ANALYSIS TECHNIQUES

Garson [2] proposed the following measure for the proportional contribution of an input to a particular output:

$$G_{ik} = \frac{\sum_{j=1}^{nh} \frac{w_{ij}}{\sum_{p=1}^{ni} w_{pj}} \cdot W_{jk}}{\sum_{q=1}^{ni} \left(\sum_{j=1}^{nh} \frac{w_{qj}}{\sum_{p=1}^{ni} w_{pj}} \cdot W_{qj} \right)}$$

A disadvantage of this approach is that during the summation process, positive and negative weights can cancel their contribution which leads to inconsistent results.

Wong, Gedeon and Taggart [6] used the following measure for the contribution of an input to a neuron in the hidden layer:

Proc. 1996 Australian New Zealand Conf. on Intelligent Information Systems, 18-20 November 1996, Adelaide, Australia. Editors, Narasimhan and Jain

$$P_{ij} = \frac{|w_{ij}|}{\sum_{p=1}^{ni} |w_{pj}|}$$

Milne [3] commented that the sign of the contribution is lost, and proposed the following measure:

$$M_{ik} = \frac{\sum_{j=1}^{nh} \frac{w_{ij}}{\sum_{p=1}^{ni} |w_{pj}|} \cdot w_{jk}}{\sum_{q=1}^{ni} \left(\left| \sum_{j=1}^{nh} \frac{w_{qj}}{\sum_{p=1}^{ni} |w_{pj}|} \cdot w_{qj} \right| \right)}$$

The measure introduced here is an extension of our technique [6]. We can define a measure P_{jk} for the contribution of a hidden neuron to an output neuron similar to the measure P_{ij} used above:

$$P_{jk} = \frac{|w_{jk}|}{\sum_{r=1}^{nh} |w_{rk}|}$$

The contribution of an input neuron to an output neuron is then:

$$Q_{ik} = \sum_{r=1}^{nh} \left(P_{ir} \times P_{rk} \right)$$

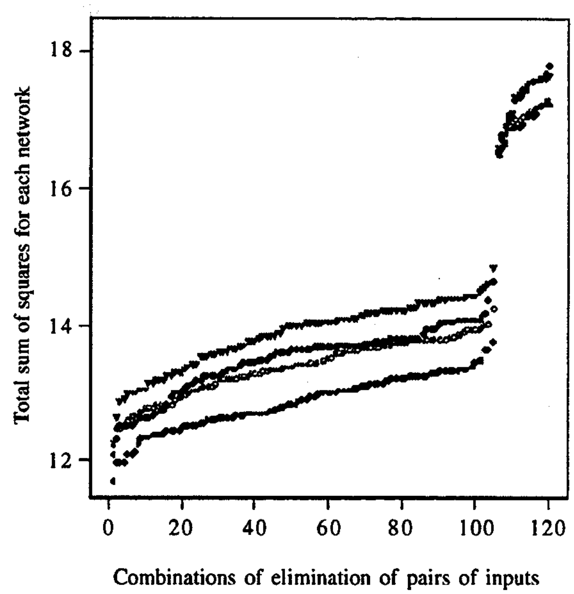
The benefit of this approach is that the magnitude of the contribution is disentangled from the sign of the contribution.

The magnitude of contributions is significant in indicating whether an input is important, while the sign of contribution is largely irrelevant in the decision to remove or retain an input, and is recoverable in any case from the raw data by simple statistical methods.

Each of the above techniques could be extended to networks with larger numbers of hidden layers than the topology used in this experiment.

BRUTE FORCE ANALYSIS

The brute force approach is to eliminate inputs and to compare the results with the predictions. Eliminating only 1 input produced inconsistent results, hence 2 inputs were eliminated. With 16 inputs, there are 120 ways to chose 2 inputs to remove. Four networks with the same topology (14-10-5) were trained for each of the 120 possibilities.



The above graph shows the results on the best total sum of squares (tss) value on the test set for each of the 480 networks run.

The initial weights for each of the 4 runs was generated for the full 16 input network, and the appropriate weights excluded when 2 inputs were eliminated.

Thus, each run had largely the same initial (random) starting weights. This has some small effect on all of the descendant networks, as shown by the consistent (minor) overall differences between the curves.

The total sum of squares (tss) values are sorted into increasing order, as there is no meaningful one dimensional scale on which to represent the removal of pairs of inputs. The discontinuity in the values plotted demonstrates the point at which there was some significant degradation of the neural network prediction, which correlated well with the tss values.

The right hand part of the graph and the leftmost part of the longer curve were used to determine the most and least significant inputs, respectively, by calculating the average rank for the tss values for each combination of inputs.

COMPARISON OF RESULTS

The contribution of inputs to each of the 5 outputs were averaged to determine the significance of inputs to the entire task the network is solving.

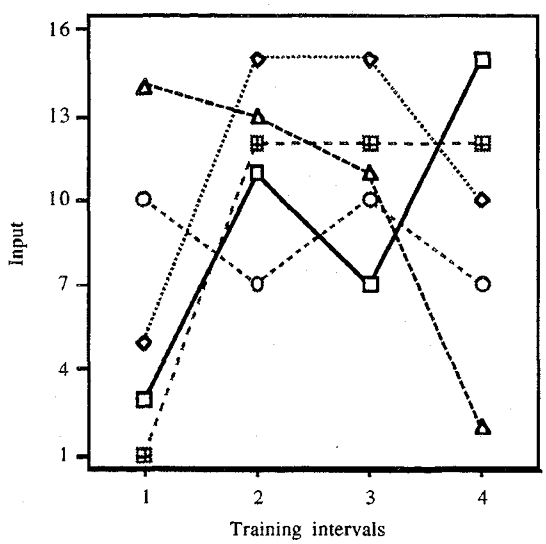
For clarity, the comparisons are made between the brute force technique and the calculated measures on the top and bottom thirds of the ordering.

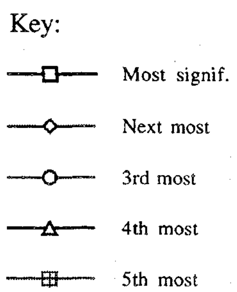
model:	B	Q	G	M
Most significant	5	11	2	11
	1	10	4	15
	10	2	6	7
	14	12	7	13
	11	14	11	12
	...	...	...	...
	7	13	9	8
	8	5	13	10
	13	4	14	2
Least significant	9	8	1	4
	6	7	15	5

From the table it is clear that the model Q introduced in this paper is $^3/_5$ths in accord with the brute force method B for both the most and least significant inputs, while both of the other methods are only $^1/_5$ths in accord on either end of the significance scale.

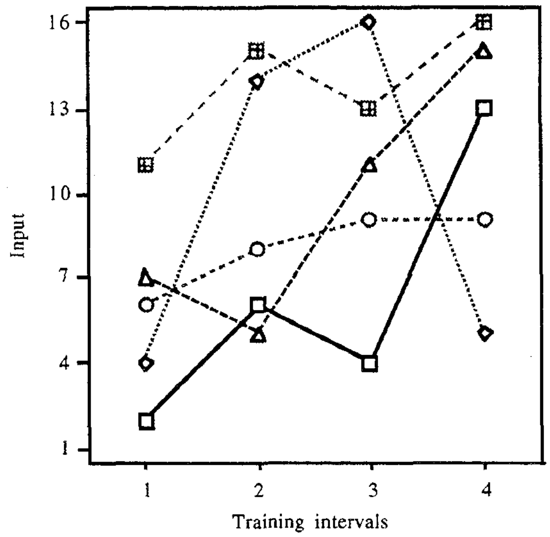
The following diagrams show the changes in the 5 most significant inputs on overtraining.

M

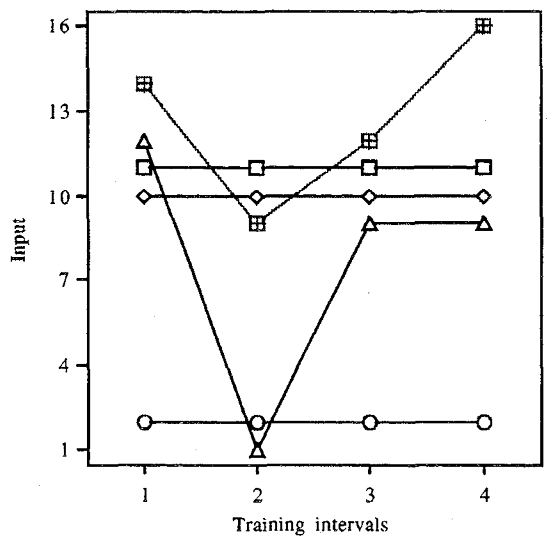




G



Q



CONCLUSION

A measure for determining the contributions different inputs make to the outputs was introduced, and validated by a brute force input ranking technique eliminating all combinations of pairs of inputs.

The measure was 60% in accord with the brute force ranking, while the comparison measures were only 20% in accord.

The measure introduced here is also more stable during training, which suggests that there is closer coupling to the network behaviour over time than with the other measures.

Note that this stability is not due to the use of a different network, but is a result of the choice of formula solely.

REFERENCES

[1] Bustos, RA and Gedeon, TD "Decrypting Neural Network Data: A GIS Case Study," *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA), 4 pages,* Alès, 1995.

[2] Garson, GD "Interpreting Neural Network Connection Weights," *AI Expert*, pp. 47-51, April, 1991.

[3] Milne, LK "Feature Selection Using Neural Networks with Contribution Measures," *Proceedings of the Australian Conference on Artificial Intelligence (AI'95),* Canberra, 1995.

[4] Milne, LK, Gedeon, TD and Skidmore, AK "Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood," *Proceedings of the Australian Conference on Neural Networks*, pp. 160-163, Sydney, 1995.

[5] Rumelhart, DE, Hinton, GE, Williams, RJ, "Learning internal representations by error propagation," in Rumelhart, DE, McClelland, *Parallel distributed processing*, vol. 1, MIT Press, 1986.

[6] Wong, PM, Gedeon, TD and Taggart, IJ "An Improved Technique in Porosity Prediction: A Neural Network Approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, n. 4, pp. 971-980, 1995.