Applications of Neural Networks to Telecommunications Systems

R J Frank, S P Hunt and N Davey
Department of Computer Science,
University of Hertfordshire,
Hatfield, Herts., UK. AL10 9AB.
Email:{R.J.Frank, S.P.Hunt, N.Davey}@herts.ac.uk

ABSTRACT: This paper gives an overview of a project involving the application of neural networks to Telecommunications Systems. Five application areas are discussed, including cloned software identification and the detection of fraudulent use of cellular phones. The systems are summarised and the general results are presented. The conclusions highlight the difficulties involved in using this technology as well as the potential benefits.

KEYWORDS: Neural Networks, Telecommunications, Applications

1 INTRODUCTION

In this paper we report on a variety of neural computational systems that have been applied in the telecommunications industry. All the systems described here were developed in a collaboration between NORTEL UK and the University of Hertfordshire, UK. In all, five application areas were investigated, resulting in two fully functioning systems, that are incorporated in NORTEL products, two successful prototypes and one application area for which we did not find suitable for a neural computational solution. In the paper we briefly describe the five applications, evaluate our resulting solutions and conclude by reflecting upon the lessons learnt.

2 SOFTWARE ANALYSIS TOOLS

Our first two applications were both designed to help with the analysis of large software systems, that are typically found in telecommunication systems - today a digital telephone exchange incorporates tens of millions of lines of code. Such systems have evolved over the years, with the current systems incorporating code from the very first system. The telecommunications industry, average error rate, is about 25 errors per 1,000 lines of code, Wayt Gibbs (1994). It is apparent, therefore, that maintaining these systems presents a problem of considerable difficulty.

2.1 COMPLEXITY ANALYSIS

This project investigated the possibility that units of software fall into natural clusters when represented by a set of complexity measures. Two data sets were examined. The first data set was a set of 2,236 procedures drawn from a single software product written in the proprietary NORTEL language PROTEL, a block structured language designed for the control of telecommunication systems. The second set was of 4,456 PROTEL procedures drawn form a variety of software products. Twelve standard measures of software complexity were used, so that each procedure was represented as a 12-ary real valued vector. Each data set was then presented to three neural network clusterers: a simple competitive network, a self organising feature map (SOM) and Fuzzy ART. The principle clustering that we found, using these networks, suggested that the procedures could be grouped primarily according to size, with a group of very small procedures , with less than 10 lines of code and another group with very large procedures. Another, major grouping, clustered procedures with a large language content compared to their size. The results are given in more detail in Field (1996).

2.2 CLONE DETECTION

After our first general investigation of complexity we were presented with a more concrete problem, namely to produce a system that could identify copied and modified software units. Such “cloned” software is very common in large software

systems that have evolved over several years, and it presents a problem in software maintenance. For example if it is necessary to change a faulty block of code it will probably be important to modify direct and modified copies of that code as well. To use a neural network on this problem it was first necessary to find a representation of a block of code as a fixed length, numeric feature vector. Such an encoding is central to the success of an application such as this, since it is imperative that similar blocks of code have similar representational vectors. It is important to note here that the notion of 'similar' is different in either case; for the source code similarity relates to the probability of the code being cloned, and for the corresponding vectors that they are close in Euclidean space. The representation we use is based on three measures. Firstly the frequencies of keywords in the unit are accumulated, secondly the length of each line is recorded and lastly the indentation pattern is represented. The latter parameter is important, as each unit of code is automatically indented in a manner that captures its syntactic structure. To map indentation (and line length) onto a fixed length vector, we first took the raw indentation values and viewed them as ordinates on a graph; we then sampled one hundred points from this graph, using linear extrapolation where necessary. This coding method is relatively stable against minor modifications to the source code, such as the addition or removal of a line. The final vector representation of a unit of code is made up of 100 frequency samples, 100 corresponding line length samples, and 96 keyword frequencies, to give a 296-ary feature vector, see Figure 1.

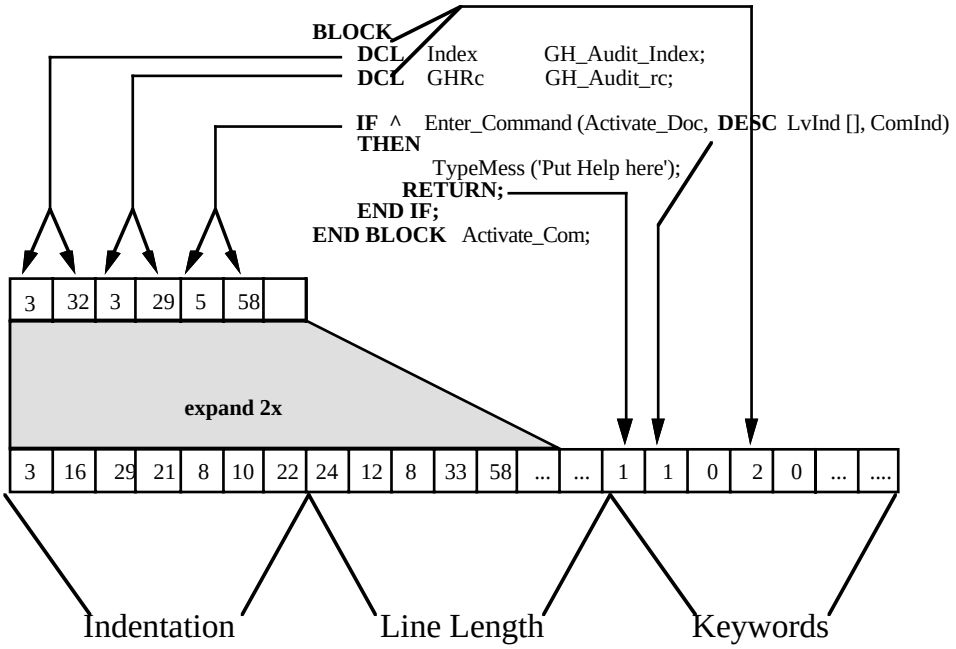

Figure 1: The representation of a block of code, as used in the clone detector

A 55 by 55 (3025 neurons) SOM was trained on 10,257 software unit feature vectors, and the resulting map gave a view of the data representing relative similarity of the input vectors. The SOM was then integrated into a larger system so that the most similar procedures of a given procedure could be identified. This then formed the basis of a completed clone detection tool, which proved useful in identifying many examples of cloned software in the PROTEL systems examined. A fuller description of this system and its development can be found in Davey (1995).

4 CALL ROUTING

The aim of this project was to investigate the possibility of using Hopfield nets to find optimal call routes. Essentially this is equivalent to using a Hopfield network to solve the travelling salesman, optimisation problem. To use the technique on a particular problem it is necessary to find an appropriate set of parameters for the network. Despite considerable investigation we were unable to find an acceptable set, and this project was abandoned.

5 TRAFFIC TRENDS PREDICTION

To make maximum usage of network capacity it is useful to be able to predict traffic demand. In this study we investigated how a neural network forecaster could be used to predict voice traffic demand, over an ATM network. The network architecture employed was the standard sliding window, feed-forward network, trained with the conjugate gradient algorithm. In addition to the sliding window input, chronological context was explicitly represented, as shown in figure 2. This information was added as much telecoms traffic is periodic, with weekly, daily and hourly trends superimposed. A critical parameter of this type of model is the size of the sliding window. We use a technique from dynamic systems theory, the false nearest neighbour heuristic, Frank (1999). This method suggested that a window of four steps back would give best performance. Good predictive performance was observed, but this system was not taken any further. More detail can be found in Edwards (1997).

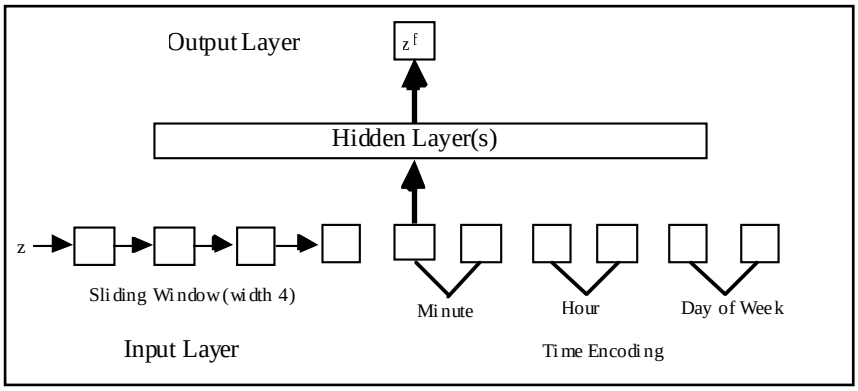

Figure 2: The modified sliding window network configuration used for predicting traffic on an ATM network.

6 FRAUDULENT USE OF CELLULAR PHONE DETECTION

Fraudulent use of cellular phones is a huge problem, for example in 1994 the estimated cost to the US industry was $482 million, representing 3.7% of revenue. In this project we investigated whether a neural network could be trained to give an indication of whether a pattern of phone usage was indicative of fraud.

Whenever a completed phone call is made a call detail record (CDR) is created. Depending on the operation currently being performed the structure of these will vary, however for our investigations we have produced a generic record which encapsulates all the salient features which are required. These include: account number, telephone number, date and time of call, the duration, the originating area and receiving area, as well as a number of other fields. These records therefore constitute an enormous database within which anomalous use must be detected.

The type of problem here is unusual and difficult, as it mixes both static classification and temporal prediction. Anomalous use has to be classified as such, but only in relation to an emerging temporal pattern. Over a period of time an individual phone will generate a macroscopic pattern of use, in which, for example, intercontinental calls may be rare; however within this overall pattern there will inevitably be violations: on a particular day the phone may be used for several intercontinental calls.

Against this background anomalous use can be identified as belonging to one of two types:

- The pattern is intrinsically fraudulent - it will almost never occur in normal use. This type is relatively easy to detect.
- The pattern is anomalous only relative to the historical pattern established for that phone

In order to detect fraud of the second type it is necessary for a neural network to have knowledge of both the historical, macro, behaviour of the phone and the recent micro behaviour. We have chosen to present both of these pieces of information as input vectors to the net. The output then is a two bit representation of the credibility of these two inputs

when taken together. Note that this method also copes quite adequately with fraud of the first type since this should be evident regardless of the historical behaviour.

For each user within the network the details gathered from the CDR's are recorded as a statistical representation; as a user profile. This includes: the proportion of local, national and international calls, number of units used, number of calls and aver duration for that user, as well as other details. These are collated over two time periods; historical and recent. The historical profile must be periodically updated to reflect gradual change in the use pattern of the unit. Several of the fields are subjected to differing normalisation techniques, in order to maintain no a-priori preference for one field over another, before the profiles are presented to the network. A Multi-Layered Perceptron network, with 18 input units, various hidden unit configurations and two output units representing, one to represent valid use and one for fraudulent use was trained using conjugate gradient training.

Figure 3 shows the networks performance for unseen fraudulent profiles. A similar graph is produced which represents normal behaviour. The confidence figure represents the value of the output node representing valid use subtracted from the value of the node representing fraudulent use.

The accounts in the lower part of the graph are divided between border-line cases and miss-classified results. For example the misclassifications on the left of the figure represent low usage accounts, where it is difficult to detect anomalies due to the paucity of data. Some of the misclassifications on the right are made up accounts where a high use of intercontinental calls are made; again it is hard for the net to pick some forms of fraud against this historical background. Both of these scenarios correspond with common sense expectations.

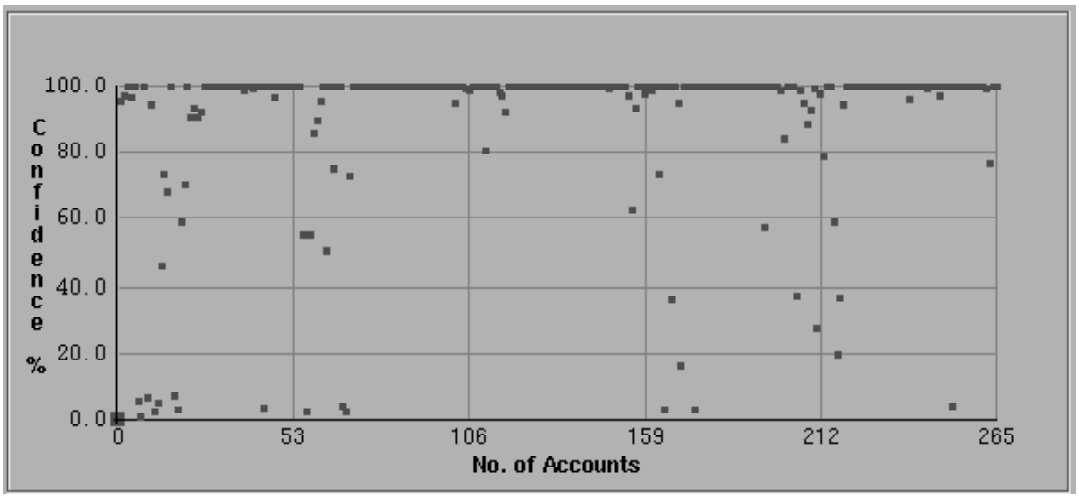

Figure 3: A visual representation of unseen fraudulent profiles. The confidence value measures the degree to which the network identifies the profile as fraudulent.

This prototype system has now been developed and integrated into a full scale fraud detector. Further details can be found in Barson (1996).

7 CONCLUSIONS

Over the period of the collaboration we were able to investigate five application areas for neural computational methods, and we achieved some success in four areas, with one failure. Four prototypes were developed and two were further developed into products.

Since neural network are data driven models, one key requirement of using them is that real world data is needed early in the project. Difficulties may arise for a number of reasons, such as data confidentiality and sensitivity, reluctance of customers to release data, and the size of the data sets involved.

It has also become apparent that the task of moving from a successful, neural net based prototype to a full system should not be underestimated. All neural network applications depend heavily on the appropriate pre-processing of the input data and post-processing of the output data. A major part of our work has been concerned with the pre-processor, the user interface and the overall quality of the system.

However the two products developed are successful and show that neural networks provide a powerful method for data driven computation, in the telecommunications industry.

8 REFERENCES

Barson, P., Davey, N. Field, S.D.H., Frank, R. J., McAskey, G. (1996)
"The Detection of Fraud in Mobile Phone Networks"
Neural Network World Vol 6, No 4. 477-484.

Davey, N., Barson, P., Field, S.D.H., Frank, R. J. & Tansley, D.S.W. (1995).
"The Development of a Software Clone Detector".
The International Journal of Applied Software Technology,
Volume 1 Number 3/4 pages 219-236

Davey N, Field S.D.H, Frank R, Barson P & McAskey G. (1996)
"The Detection of Fraud in Mobile Phone Networks, Neural Network World", Volume 6, no 4, 477-484, 1996

Edwards T, Tansley D.S.W, Frank R.J. & Davey N. (1997)
"Traffic Trends Analysis Using Neural Networks".
Proceedings. International Workshop on Applications of Neural Networks to Telecommunications 3 (IWANNT'3), 157-164, 1997

Field, S.D.H., Davey, N., Frank, R. J.(1996)
"Using Neural Networks to Analyse Software Complexity"
Australian Journal of Intelligent Information Processing Systems, Vol 3, No. 3, p14-32

Frank R.J, Hunt S.P, Davey N (1999)
"Times Series Prediction and Neural Networks"
To be published in: Proceedings of Engineering Applications of Neural Networks, EANN 99.

Wayt Gibbs, W. (1994),
Software's Chronic Crisis, *Scientific American* pp 72-81, September 1994.

Acknowledgements

We would like to acknowledge the financial assistance of the Department of Trade and Industry of the UK Government, who have partly funded a Teaching Company Scheme at NORTEL UK, through the Teaching Company Directorate organisation, under grant no. TCS–1326.

The authors of this paper would also like to thank the S&SE department of NORTEL UK Harlow, for the invaluable support given throughout the work reported here.