ELSEVIER

International Journal of Forecasting 13 (1997) 21–31

international journal of forecasting

# Short-term inter-urban traffic forecasts using neural networks

Mark S. Dougherty*, Mark R. Cobbett[1]

*Institute for Transport Studies, University of Leeds, Leeds, UK*

**Abstract**

Back-propagation neural networks were trained to make short-term forecasts of traffic flow, speed and occupancy in the Utrecht/Rotterdam/Hague region of The Netherlands. A problem which had to be faced when designing the system was the vast number of possible input parameters. Whilst neural networks which utilised all available inputs performed well, their size made them impractical for implementation. A technique of stepwise reduction of network size was developed by elasticity testing the large neural networks, showing a way of overcoming this difficulty. Results for occupancy and flow forecasts by this method show some promise, but do not out-perform naive predictors. Forecasts of vehicle speed were much less successful, perhaps because of the distorting effect of slow moving vehicles, particularly in low flow conditions. The elasticity tests were found to be useful, not only as a means of enabling network size reduction, but as a means of interpreting the neural network model.

*Keywords:* Neural networks; Back-propagation; Traffic; Elasticity

## 1. Introduction

Motorways instrumented with appropriate sensors provide streams of data (on, for example, flow and occupancy) that are potentially very appropriate for use with neural networks, because of the richness of the information on 'outputs' (e.g. flows a short time ahead) and on inputs. Previous studies of the application of neural networks to analyse traffic data have demonstrated the value of neural networks for several tasks: pattern recognition (Dougherty et al., 1993), modelling (Dougherty et al., 1994) and forecasting (Smith and Demetsky, 1994; Dochy et al., 1995). Other related work involving applications of neural networks in transport is reviewed in Dougherty (1995).

The above-mentioned studies have shown that the problem of short-term forecasting is not necessarily straightforward: the traffic forecasting problem is a multi-dimensional one, involving relationships between measurements made at different times and geographical sites; the data are also noisy; and any system devised also needs to be robust enough to cope with incorrect or missing values.

* Corresponding author at: Centre for Research on Transportation and Society, Högskolan Dalarna, S78188 Borlänge, Sweden. Tel.: +46 23 77 85 35; fax: +46 23 77 85 01.

[1] Present address: Ordnance Survey, Southampton, UK.


*PII* S0169-2070(96)00697-8

This article describes the development of a systematic approach to neural network forecasting methodology. An initial study, described in Section 4, was undertaken with data from an inter-urban motorway site near Utrecht, The Netherlands. The methodology was further developed and applied to another site near Rotterdam. The use of data from both these sites arose in the context of a project (called GERDIEN) under the DRIVE-II programme of the European Commission (EC).

The data sources used are first described in Section 2; and a brief introduction to neural networks is given in Section 3. Readers already familiar with neural networks are advised to pass over this latter section as it is intended for those with no experience of the technique.

During the initial study, it was found that the greatest problem was selection of a suitable set of input parameters to the neural networks. Whilst neural networks which utilised all possible inputs performed well, their size made real-time implementation impractical. A technique of step-wise reduction of network size by performing elasticity tests on the large neural networks produced a novel way of overcoming this difficulty; this is described in Section 4. Section 5 discusses the evaluation of the adequacy of the forecasts obtained, and Section 6 the traffic engineering interpretation of the models developed.

## 2. The data

All analysis reported here was conducted with data for inter-urban motorways in The Netherlands. It consisted of flow, speed and occupancy values collected from pairs of inductive loops (1 m apart) placed at various points (average separation 3 km) along the motorways in the Rotterdam/The Hague/Gouda triangle (Fig. 1).

Initial trials were conducted with a data set for the A2 motorway near Utrecht. The methodology was more fully developed for the A12 motorway near Rotterdam, which was the main test site for the GERDIEN pilot. For that test area, there were 16 or 17 geographical sites on each carriageway (Fig. 2) for which it was considered appropriate to include measurements when making a prediction for a given point.

Data were available averaged over 1-min intervals, and each lane was measured independently. Data collected over a period of a month were used, with off-peak, peak and incident conditions (unusual flow patterns caused by disturbances such as accidents) all represented. Initial experiments with neural networks using data for 1-min periods were not very successful because of the large minute-by-minute stochastic variation (presumably caused by individual platoons passing the detectors). This type of short-term variation is exacerbated by considering the data in each individual lane; in certain flow conditions a large amount of lane changing takes place. The data were therefore combined across the carriageway, and averaged over a 5-min period before being presented to forecasting algorithms. This pre-processing greatly reduced the amount of noise in the data and improved the quality of the results.

## 3. Neural networks

A brief introduction to neural networks follows. All of the work described was done using feed-forward networks, trained using a back-propagation rule and hence the introduction only covers this paradigm.

### 3.1. *Elements of a back-propagation neural network*

A back-propagation neural network consists of nodes, connection weights, a transfer function and a topological structure relating them. These are described below.

- Nodes: The basic building block is the neuron, also known as a node or processing element. A node takes in a set of inputs (normalised between zero and one) and computes a normalised output according to transfer function.

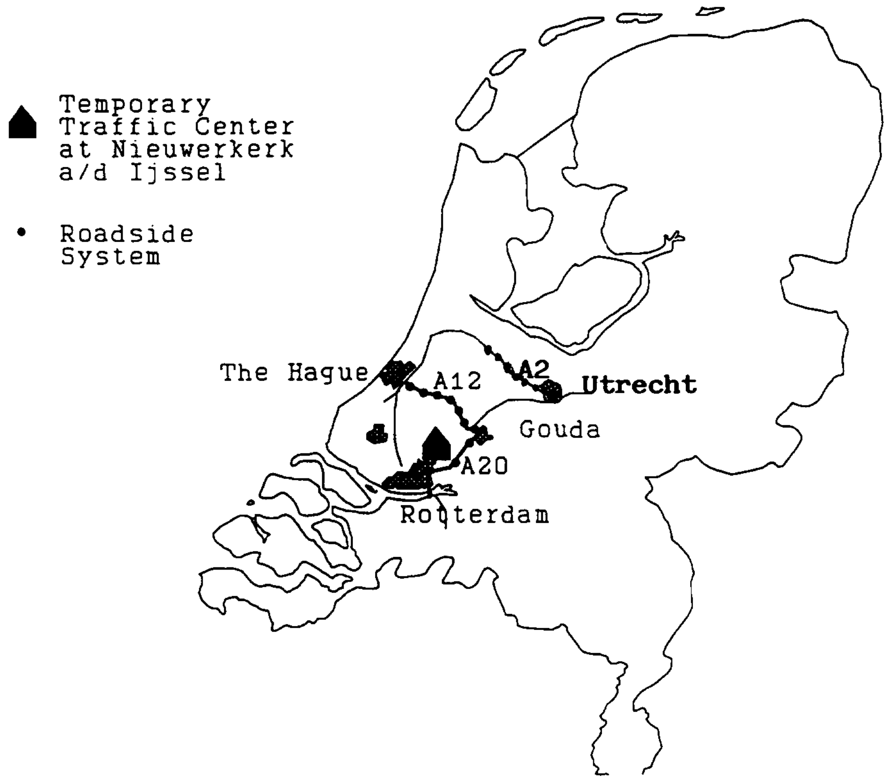


Fig. 1. The Rotterdam/The Hague/Gouda triangle.

Connection weights: A neural network is composed of many nodes joined together by connections; thus outputs of some nodes are the inputs to others. These connections are of varying strength; each connection having a real weight associated with it.

Transfer function: Typically the output state of a single neuron can be characterised as either 'on' or 'off'. A change from one state to the other is triggered when the sum of the inputs (weighted by the strength of their respective connections) passes a threshold, usually represented by a sigmoidal transfer function.

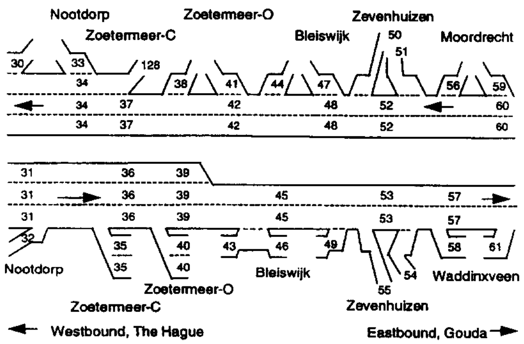


Fig. 2. Geographical sites.

– Topological structure: In theory any topological arrangement of nodes and connections could be used. To ease the problems of analysis and visualisation it is usually to arrange the neurons in layers, with all nodes in adjacent layers connected to each other (Fig. 3). A neural network thus has an input layer and possibly one or more hidden layers, so named because all of their connections are internal to the network.

### 3.2. Learning algorithm

The back-propagation paradigm is known as a supervised learning technique. To carry out the

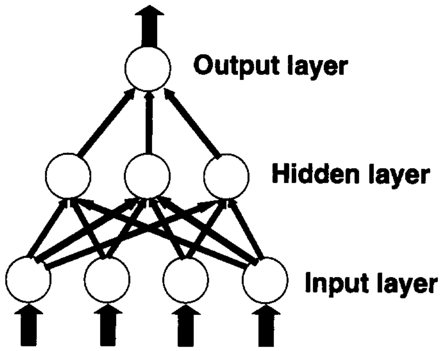

Fig. 3. Topological structure.

training a large number of examples of the desired behaviour are required. A summary of the steps involved in training the network with the data follows:

(1) Present an example of input data, normalised to between 0 and 1 to the input layer of the network.
(2) Using the current connection weights, compute a set of outputs which emerges from the input layer via the hidden layer.
(3) Compare this set of outputs with the values of output that was observed for this example by computing a root mean square difference, or global error.
(4) Update the connection weights by a small amount to displace the outputs towards the desired outputs. This involves 'sharing' the global error between all the different connections. This is done using a partial derivative function which propagates the errors back down to the input layers.

If the training is successful, one finds that the global error reduces, as examples are repeatedly presented. This convergence can be displayed graphically by plotting the error against time. The rate of convergence varies greatly, and there are various methods of increasing it, such as the inclusion of a variable momentum term. (Introductory texts are Beale and Jackson, 1990, and Hecht-Nielsen, 1990. A more advanced text is Hassoun, 1995, which covers the subject with considerable mathematical rigour.)

## 4. Input selection

There were three quantities (speed, flow, occupancy) available at each of the prediction points in our test area. As expected, the neural network predictions proved to be more effective if data from previous times slices were available too, as temporal patterns are often present. This gives hundreds of possible input data points, and although neural networks trained using this neural network converged, a crucial requirement was to keep the neural networks as small as possible. The system was expected to run in real time, and the computational effort required rises approximately with the square of the number of inputs to the neural networks. Furthermore, as the number of inputs increases, ever larger data sets are required to train them. This is known as the *curse of dimensionality* (Bellman, 1961) and leads to a paradoxical situation whereby *feature extraction* of certain key input variabled results in better performance, despite the apparent loss of information (Bishop, 1995). We therefore needed to specify a subset of the available inputs to act as inputs for each prediction point.

### 4.1. The reduction methodology

Specifying the inputs to the neural networks was found to be by no means an easy task. Initial trials showed that, in order to develop neural networks that could readily be implemented for all sites in the GERDIEN pilot, we had to consider two important questions:

(1) What neural network topology is needed (and hence the number of pieces of input data we were going to use to make the predictions)?
(2) Which particular pieces of input data would give the best overall forecasting result for a particular prediction?

Question 1 (the number of inputs) had, for simplicity of implementation, to be the same for all the neural networks. This is because only a single neural network was implemented within the code. This was then modified for each prediction by loading in a set of weights associated with that prediction point.

Implementation of many different networks of different size would have been too cumbersome and costly. Size of network therefore became a straightforward trade-off between the accuracy of the model and the computational effort required. After some scoping studies we decided that 40 inputs (and a single hidden layer of 10 units) was a reasonable compromise.

Question 2 was much more complicated. One carriageway of our test area produces hundreds of possible input data points; many more than the 40 inputs the implemented neural networks had. We therefore had to select a subset of points which gave optimum forecasting performance under a range of conditions. This selection would obviously be different for each prediction point. There were obviously many different possible combinations. Initial experiments, which compared the performance of a large (but impractical) neural network using all the input data points with smaller neural networks with the parameters chosen by hand showed a marked drop in forecasting performance. We either had insufficient understanding of the system to make a judgement as to which parameters to use, or our value of 40 parameters (chosen for reasons of computing performance) would have to be increased at the cost of a reduction in the response time of the system.

The methodology developed was to allow the neural networks to select their input parameters themselves as part of the training and testing regime. This built on some earlier work (Dougherty and Joint, 1992), which used elasticity analysis to understand trained neural networks. The trained networks were analysed by perturbing each of the input nodes by a fixed percentage, and observing the movement of the output. In this way the importance of each node can be assessed. Items can then be removed from the input data set and the network reduced in size (in a similar manner to eliminating factors in regression analysis). Thus the data inputs associated with the neural networks become a result of the tuning/testing process, described in more detail below. This idea has an obvious relationship with neural network pruning algorithms (Reed, 1993), but differs from previous work in that the algorithm presented here is a 'one-shot' process, whereas pruning is usually carried out as a gradual process of refinement.

### 4.2. Testing and tuning

The task of testing and tuning the networks was subdivided into several sequential tasks:

(1) Preliminary training with superset: Initially, large networks were trained using all the available inputs; the 'superset'. This stage took quite a considerable amount of computing time, as might be expected. The 'superset' contained all data measurements from all geographical sites from four time steps spanning the present $(t-0)$ to $t-15$. Data further into the past were not used, as elasticity testing (see below) showed that data from further into the past were of little influence.

(2) Elasticity testing and network reduction: The elasticity tests on the 'superset' neural networks, once trained here done by exposing the neural network to examples of data, and then observing how the output changed as each input in turn was altered by a small percentage (typically 5%). This is equivalent to finding an estimate of the partial derivatives of the output with respect to the various inputs. However, because the model is non-linear, these partial derivatives are themselves dependent on the values of the inputs. It is therefore necessary to repeat this test many times, so as to examine a distribution of elasticity values, rather than a single example which might be unrepresentative. The distribution can be displayed as a histogram of bands of elasticity values, with the height of the bars indicating the relative frequency of a particular range of elasticities occurring (taken over a large test set). Such a histogram is shown in Fig. 4. Note that this figure is a composite histogram, containing elasticity distributions pertaining to several different inputs. What was observed was that

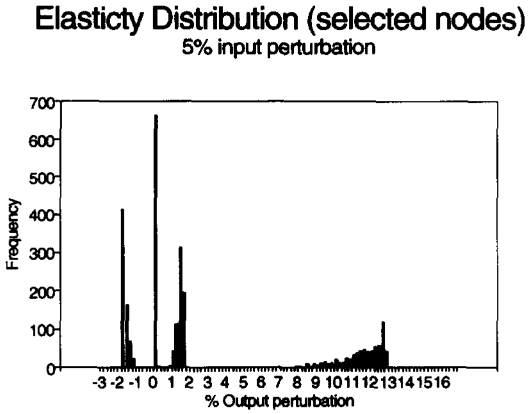


Fig. 4. Elasticity distribution (selected nodes), 5% input perturbation.

the distributions varied widely and could be placed into five classes. A brief description of the classes follows:

Class 1: Highly positive elasticity (output increases on average by more than 5%). Spread of elasticity values is large.
Class 2: Mildly positive elasticity (output increases on average, but by less than 5%). Spread of elasticity values is quite small.
Class 3: Almost no elasticity (output does not vary as input changes). Spread is very small.
Class 4: Mildly negative elasticity (output decreases on average, but by less than 5%).
Class 5: Highly negative elasticity (output decreases by more than 5%). This class was very rare and so does not appear in the figure.

We hypothesised that input points of Classes 1 and 5 are the most important. They represent the highly active parts of the model, and therefore all points of this type should be included. Those of Classes 2 and 4 represent less active parts of the model; it was further hypothesised that these have a large amount of redundancy between them. Therefore the model should incorporate only a representative sample of these points. Those of Class 3 are unimportant; they represent parts of the neural network which have 'died off'; the weights of the connections to these inputs are, in the main, very low. Therefore these inputs can all be excluded. There were no cases observed where a large spread of elasticity values was centred closed to the origin. Greater spread seems directly related to a higher absolute value of average elasticity values.

(3) Final training: Final training was carried out using a very large data set comprising many thousands of examples. Efforts were made to ensure that this data set was representative of the entire range of conditions that the neural networks were likely to come across.

## 5. Evaluation of the performance of neural networks

Evaluation was done at several levels. Statistical analysis was used to provide a numerical measure of forecasting accuracy. Some qualitative conclusions were drawn during the training testing processes. Since GERDIEN is real-time system the time taken to produce forecasts is highly significant and was also considered; this and other questions of implementation are covered in Appendix A.

### 5.1. Statistical analysis

A statistical measure was used to measure prediction accuracy; a root mean square error proportional indicator, known as RMSEP. It is defined as:

$$\mathrm{RMSEP} = \frac{\sqrt{n \sum_{i=1}^{n} (X_i^{\text{predicted}} - x_i^{\text{observed}})^2}}{\sum_{i=1}^{n} x_i^{\text{observed}}}$$

The RMSEP was calculated for a sample of 820 test examples (approximately 06:00 in the morning to 19:30 at night at a typical weekday), and the results are displayed in Table 1. The

Table 1
Analysis of forecast results

| Horizon | Site 36 | Site 39 | Site 45 | Site 53 |
|---|---|---|---|---|
| *RMSEP for eastbound flow predictions* | | | | |
| $t+5$ (one step) | 0.20 (0.18) | 0.23 (0.24) | 0.23 (0.21) | 0.19 (0.19) |
| $t+15$ (two steps) | 0.22 (0.23) | 0.22 (0.27) | 0.24 (0.25) | 0.23 (0.25) |
| $t+30$ (six steps) | 0.27 (0.26) | 0.26 (0.28) | 0.28 (0.28) | 0.27 (0.28) |
| *RMSEP for eastbound occupancy predictions* | | | | |
| $t+5$ (one step) | 0.20 (0.15) | 0.22 (0.17) | 0.22 (0.17) | 0.18 (0.16) |
| $t+15$ (two steps) | 0.22 (0.22) | 0.21 (0.25) | 0.23 (0.22) | 0.21 (0.23) |
| $t+30$ (six steps) | 0.26 (0.25) | 0.25 (0.26) | 0.27 (0.26) | 0.26 (0.26) |
| *RMSEP for eastbound speed predictions* | | | | |
| $t+5$ | 0.32 (0.17) | 0.33 (0.19) | 0.31 (0.15) | 0.39 (0.16) |
| $t+15$ | 0.39 (0.28) | 0.40 (0.30) | 0.36 (0.28) | 0.41 (0.31) |
| $t+30$ | 0.41 (0.35) | 0.39 (0.34) | 0.35 (0.31) | 0.42 (0.33) |

figures in brackets refer to a baseline or naive forecast where the current value is taken as the forecast. The absolute forecasting performance generally deteriorates as the forecasting horizon extends further into the future. It is also clear that the speed forecasts appear very unsuccessful. When looking at the flow and occupancy predictions, it is clear that the 15-min forecasts are encouraging; in general the RMSEP is significantly lower. No clear pattern emerges for either the 30-min or 5-min forecasts, although it is surprising how little the baseline figures deteriorate for the 30-min forecasts. We conclude that between 5 and 30 min probably represents the look-ahead window for which the forecasting method is worthwhile.

### *5.2. Qualitative analysis*

The results of the speed predictions were disappointing at the 'superset' training stage; the table of results in this case refers to these networks. Because of the poor performance, smaller networks have not been produced; there is little point implementing a neural network which shows inadequate performance. Work is still continuing to extract greater performance. Note that the use of speed data in any forecasting model incurs two problems, particularly in low flow conditions. Firstly, occasional vehicles travelling very fast or slowly distort the speed estimate, and secondly, if no vehicles pass in a given interval, what value does one adopt for speed? Such problems are particularly acute when modelling speed itself.

The neural networks never learn to cope with these cases. The first problem can be partly overcome by building training sets of data which are more representative. This is done by searching through the data bases for extra examples of unusual behaviour which are added to the training set. Large amounts of data are needed to do this, and it is a time-consuming process.

The second problem is a question of interpretation. Should the normal default (zero) be accepted, despite its absurdity (given that zero speed implies infinite travel time); or should one choose an average speed? It was found that, whatever value is chosen, results are very poor, because the results produced by the neural network have a wide spread in these cases; it seems that these events are very difficult to predict using this method. One can of course ignore all such cases when computing the results, but is this realistic? It seems rather extraordinary to exclude a significant proportion (about 5%) of the results purely because this situation is hard to predict and does not fit in well with the methodology! Incidentally, this problem probably explains why the RMSEP figures for the speed predictions are so poor for the neural network forecast. A large proportion of the RMSEP came from these cases, because the actual value had been set to zero by default. Since it is quite common for a value of zero to appear either consecutively or near another value of zero, the baseline forecast sometimes gets these results 'right', whereas the neural network never does; its behaviour tends to make it 'ride over' such data and given a result approximating to an average speed. This effect becomes less noticeable as the forecasting horizon increases, as the chance of a coincidental overlap become less.

## 6. Traffic engineering interpretation

A classic problem of neural networks is their 'black box' nature and the difficulty of gaining any kind of explanation from them (Ripley, 1992). An interesting exercise is therefore to examine the sets of inputs chosen by the neural networks in order to gain some interpretation of the model.

An example table of chosen inputs for a prediction network for the eastbound carriageway is presented in Table 2, to illustrate the sort of information that can be extracted from the elasticity analysis technique. The furthest upstream sites are at the top of the table, the further downstream sites at the bottom. The reader will probably need to refer back to the schematic map of the test area (Fig. 2) to interpret the information.

General conclusions which can be reached from examining these sets of tables are listed below.

- Surprisingly, data collected downstream of the prediction point proves as important as data collected upstream. For example, Site 45 is roughly in the middle of our test zone, but it uses as many inputs from downstream as upstream.
- A common phenomenon is selection of a measurement quantity at several different time periods, indicating that the neural network is making use of temporal patterns in the data. A good example of this can be seen with the data from Site 36, where occupancy is chosen at three consecutive time slices.
- Some measurement sites proved to have little influence on predictions. It is not clear why this is so, particularly as some of these sites would probably be chosen by an engineer asked to make a judgment of which sites would provide the most useful data.
- It is comparatively rare for two parameters to be selected from an identical measurement point and time slice and selection of all three parameters is extremely rare, occurring on

Table 2
Parameters selected for $t+5$ occupancy prediction at Site 45

| Detector site | Data from $t-0$ | Data from $t-5$ | Data from $t-10$ | Data from $t-15$ |
|---|---|---|---|---|
| Site 31 | Flow, occ | Speed | | Speed |
| Site 32 | | | | |
| Site 35 | Flow, speed, occ | | | |
| Site 36 | Occ | Occ | Occ | Flow |
| Site 40 | Occ | Flow | | Flow |
| Site 39 | Flow, occ | Flow, occ | Occ | |
| Site 43 | | Flow | Flow, occ | Occ |
| Site 46 | | | | |
| Site 45 | | Flow, occ | Flow | |
| Site 49 | | | Speed | |
| Site 55 | Occ | | Flow | |
| Site 53 | Flow | Occ | Flow | Flow, occ |
| Site 54 | | | | |
| Site 58 | | | | |
| Site 57 | Flow, occ | Flow | Flow | Occ |
| Site 61 | | | Speed | |

only a few occasions at one particular site and time. It is possible there is some unusual behaviour occurring at this site.

- The parameters selected for flow and occupancy predictions for a particular forecasting horizon and measurement point are very similar. This is not surprising, because the curves are very similar in most traffic conditions. There are also quite strong similarities between the geographical sites. This is interesting from a transferability perspective, but is must be remembered that our test site was quite small, and this will tend to exaggerate such similarities as the number of patterns and relationships observed will be quite small.

It is also useful to examine some tables summarising the parameters selected by the neural networks. The summaries in Table 3 were compiled from more detailed tabulations for the individual models. From Table 3 it can be seen quite clearly that the number of parameters selected reduces as one looks further back into the past. This phenomenon becomes more marked as the forecasting horizon increases. This suggests that temporal patterns become less useful the further ahead one tried to forecast.

## 7. Conclusion

The conclusions arising from the various studies on the development of neural networks for short-term traffic forecasts, based on 1-min traffic data from inter-urban motorways, are as follows.

- The results of using neural networks to predict

Table 3
Selection of inputs viewed in time domain

| Prediction | Data from $t-0$ | Data from $t-5$ | Data from $t-15$ | Data from $t-30$ |
|---|---|---|---|---|
| *$t+5$ predictions; total number of parameters selected* | | | | |
| Site 36 flow | 12 | 10 | 11 | 7 |
| Site 36 occ | 12 | 10 | 11 | 7 |
| Site 39 flow | 13 | 11 | 10 | 6 |
| Site 39 occ | 13 | 11 | 10 | 6 |
| Site 45 flow | 13 | 10 | 10 | 7 |
| Site 45 occ | 13 | 10 | 10 | 7 |
| Site 53 flow | 14 | 10 | 8 | 8 |
| Site 53 occ | 13 | 10 | 10 | 7 |
| *$t+15$ predictions; total number of parameters selected* | | | | |
| Site 36 flow | 13 | 14 | 8 | 5 |
| Site 36 occ | 12 | 14 | 8 | 6 |
| Site 39 flow | 11 | 14 | 8 | 7 |
| Site 39 occ | 12 | 13 | 8 | 7 |
| Site 45 flow | 13 | 12 | 9 | 6 |
| Site 45 occ | 9 | 10 | 9 | 12 |
| Site 53 flow | 11 | 12 | 11 | 6 |
| Site 53 occ | 12 | 12 | 11 | 5 |
| *$t+30$ predictions; total number of parameters selected* | | | | |
| Site 36 flow | 14 | 13 | 8 | 5 |
| Site 36 occ | 15 | 13 | 8 | 4 |
| Site 39 flow | 13 | 13 | 9 | 5 |
| Site 39 occ | 13 | 13 | 9 | 5 |
| Site 45 flow | 15 | 13 | 7 | 5 |
| Site 45 occ | 14 | 13 | 7 | 6 |
| Site 53 flow | 15 | 15 | 7 | 3 |
| Site 53 occ | 15 | 15 | 7 | 3 |

flow and occupancy whilst not outstanding show some promise, although certain simplifications to the problem are needed in order to first reduce the noise inherent in this type of data. It is suggested that future work should investigate adaptive neural networks such as recurrent back-propagation in order to improve performance.
- Speed predictions are problematic, and performance achieved with neural networks was poor. Part of this can be attributed to the difficulties experienced in very low flow conditions, which suggests that an alternative forecasting methodology should be sought.
- Analysis of the elasticities is helpful in reducing the problems to manageable size, and enabling a configuration of neural network to be engineered that is common to all sites.

It was found that individual vehicles travelling very slowly or quickly distort the data and make training the networks difficult, particularly in conditions of low flow. This problem is exacerbated if, during a forecasting period, no vehicles pass. One way of resolving the difficulty would be to remove all such cases from the training set and simply not calculate forecasts during conditions of low flow (when forecasts are not as necessary in any case because drivers' journey times are less influenced by the behaviour of the rest of the traffic). This simplistic solution could prove effective, but work would need to be done to quantify the optimum cut-off threshold. An alternative solution would involve devising more complex measures of forecasting error and providing a real-time confidence limit in tandem with the forecast.

The results from the elasticity tests have a useful function apart from enabling network size reduction, as various useful traffic engineering inferences can be drawn from them. Temporal and spatial patterns can be seen emerging from the data. Therefore, although neural networks may not prove to be the optimum forecasting solution, they may very well have a role to play in optimising alternative techniques. Of particular interest is the possibility of interpreting the patterns within the selected data points.

**Acknowledgements**

The development of the methodology would not have been possible without the support of both the European Commission, through the DRIVE-II project GERDIEN (ref. V2044), and the UK Engineering and Physical Sciences Research Council (EPSRC), whose predecessor body, the Science and Engineering Research Council, funded complementary work (grant GR/H/82549). Particular thanks is due to TNO–INRO in The Netherlands and the Dutch 'Rijkswaterstaat' for allowing us to continue working with the Utrecht data set on the SERC project.

## Appendix A. Implementation

For implementation in the on-street trials, it was necessary to transform the neural networks into C code, in order to integrate the short-term forecasting modules into the GERDIEN application. The software used to develop the neural nets, NeuralWorks Professional, included a Designer Pack, which was used to convert the networks created by NeuralWorks into C source code modules.

Designer Pack uses a neural network definition file created by NeuralWorks to produce C data structures that describe the topology of the network. Additionally, it produces a set of routines to implement the control of data flows and operational aspects of the network such as the transfer function. Designer Pack allows different options for describing the topology of the network, with varying amounts of the necessary information being either 'hard-wired' into the code, or dynamically loaded in at run-time from ASCII files. There is a trade-off between speed and flexibility, which affects the choice of option.

Because we required the software to run in real time, we chose a fairly 'hard-wired' option, in order to maximise performance. This decision was made following some tests with different prototypes, which showed that completely dynamically loaded networks suffer a considerable performance penalty. Loading in and initialising the network structure is computationally costly. The number of networks we would be able to implement would therefore be very small.

By specifying the same topology and transfer function for each neural network, a compromise is possible. This is done by implementing a single 'skeleton' module for the neural networks, with just the weights loaded in for each network. This would allow on-line retraining without re-compilation, and would be a feasible option with a high performance workstation. The methodology developed therefore allows progression to dynamically loaded networks; this offers a much flatter performance curve when dealing with very large numbers of networks.

In a trial of system performance, the neural network system was implemented on a Sun SPARC station IPC (not a particularly fast machine at the date of writing). The system could cope with loading approximately 100 neural networks and computing their outputs per minute. This was more than adequate for the system trial.

## References

Beale, R. and T. Jackson, 1990, *Neural Computing: An Introduction* (Adam Hilger, Bristol).

Bellman, R., 1961, *Adaptive Control Processes: A Guided Tour* (Princeton University Press).

Bishop, C.M., 1995, *Neural Networks for Pattern Recognition* (Oxford University Press).

Dochy, T., M. Danech-Pajouh and Y. Lechevallier, 1995, Short term road forecasting using neural network, *Recherche Transports Sécurité*, English Issue 11, 73–82.

Dougherty, M.S., 1995, A review of neural networks applied to transport, *Transportation Research Part C*, 3(4), 247–260.

Dougherty, M.S. and M. Joint, 1992, A behavioural model of driver route choice using neural networks, in: S.G. Ritchie and C.T. Hendrickson, eds., *Conference Preprints, International Conference on Artificial Intelligence Applications in Transportation Engineering*, June 1992, San Buenaventura, California, USA (Inst. of Transp. Studies, Irvine, CA).

Dougherty, M.S., H.R. Kirby and R.D. Boyle, 1993, The use of neural networks to recognise and predict traffic congestion, *Traffic Engineering and Control*, 34 (6), 311–318.

Dougherty, M.S., H.R. Kirby and R.D. Boyle, 1994, Using neural networks to recognise, predict and model traffic, in: M. Bielli, G. Ambrosino and M. Boero, eds., *Artificial Intelligence Applications to Traffic Engineering* (VSP, Utrecht, The Netherlands) 233–250.

Hassoun, M.H., 1995, *Fundamentals of Artificial Neural Networks* (MIT Press, MA).

Hecht-Nielsen, R., 1990, *Neurocomputing* (Addison-Wesley).

Reed, R., 1993, Pruning algorithms – a survey, *IEEE Transactions on Neural Networks*, 4(5), 740–747.

Ripley, 1992, *Statistical aspects of neural networks*, in: *Proceedings of Semstat, Denmark* (Chapman and Hall, London).

Smith, B.L. and M.J. Demetsky, 1994, Short term traffic flow prediction: neural network approach, *Transportation Research Record*, 1453, 98–104.

**Biographies:** Mark S. DOUGHERTY graduated in Computer Science from Cambridge University. Since his appointment as research fellow at the Institute for Transport Studies, University of Leeds, he has worked on a variety of projects applying artificial intelligence to the transport field. He has applied logic programming to constraint satisfaction problems such as traffic signal design. He was involved in the development of the knowledge-based intelligent traffic supervisor CLAIRE. Through several projects he has helped pioneer the use of neural networks in the transport community, with emphasis on the problems of short-term forecasting, parameter estimation and behavioural modelling. Recently he took up an appointment as a senior research fellow in the Centre for Research on Transportation and Society at Borlänge, Sweden. He will continue an association with ITS through an appointment as an honorary research fellow.

Mark R. COBBET currently works as a systems developer for the Ordnance Survey, Britain's National Mapping Agency. Previously he was a research fellow at the Institute for Transport Studies, University of Leeds, where he developed software for a wide range of transport based applications over a period of 5 years. He has a degree in Physics from the University of Liverpool and a masters degree in Software Engineering from the University of Sheffield.