Forecasting **Products**

Research and Development

Forecasting Research
Technical Report No. 314

Estimating forecast error with an artificial neural network

By

J. Green

October 2000

The Met.Office

Excelling *in weather services*

Forecasting Research
Technical Report No. 314

Estimating forecast error with an artificial neural network

By

J. Green

October 2000

The Met. Office
Forecasting Products
London Road
Bracknell
Berkshire
RG12 2SZ
United Kingdom

Estimating forecast error with an artificial neural network

J. Green

Technical Report 314

Contents

1 Introduction

1.1 Background

Numerical weather prediction models solve a set of governing equations to approximate the state of the atmosphere at some future time. The hypothetical perfect model of the atmosphere would be a multi-dimensional non-linear function, mapping an atmospheric state from one time to another. Given enough hidden nodes an artificial neural network can approximate any continuous mapping from input to output domain. A neural network could therefore be used to approximate the evolution of the atmosphere. The size of network and required training data needed make this infeasible, although some studies have shown success in predicting complicated single meteorological events, such as El Nino, [Hsieh].

Useful results may be achieved by using the current model output as a starting point and using an artificial neural network to produce a correction, in a similar way to a Kalman-Bucy filter. The artificial neural network could also be trained to estimate the absolute value of error which could be used to predict possible forecast error.

Linear regression methods have been used to estimate the size of forecast errors, [McNair]. This has included model corrections, in the form of a constant wind speed multiplier, and absolute error estimation, in the form of increasing uncertainty with lead time. The application of a multi layer perceptron to predict forecast error is the next step from using these linear regression techniques.

The ability to predict forecast skill has many applications outside the intended aims of this project. Error estimation is very important in the application of variational assimilation [Grooms] and ensemble computation. Indeed the process of neural network training is very similar to the minimisation of the cost function which occurs within variational assimilation.

1.2 Aims

This work was conducted in an attempt to provide aviation customers with skill forecasts. Aviation operators use the output from the numerical models to make routing and fuel loading decisions. Currently contingency fuel is loaded to account for forecast wind errors. The ability to estimate the confidence of the forecasts would allow contingency fuel loadings to be varied, producing considerable cost savings and safety benefits. Previous work has shown some benefit in using linear regression to model forecast errors [McNair]. Improvements should be gained by extending the number of predictors and using a non-linear regression method, such as an artificial neural network. The eventual aim is to provide prior warning of possible forecast errors so that extra fuel or routing precautions can be taken.

1.3 Simple example

Consider a very simple chaotic system, the Lorenz attractor, described by the following 3-dimensional differential equations.

$$\frac{dx}{dt} = -10x + 10y \tag{1}$$

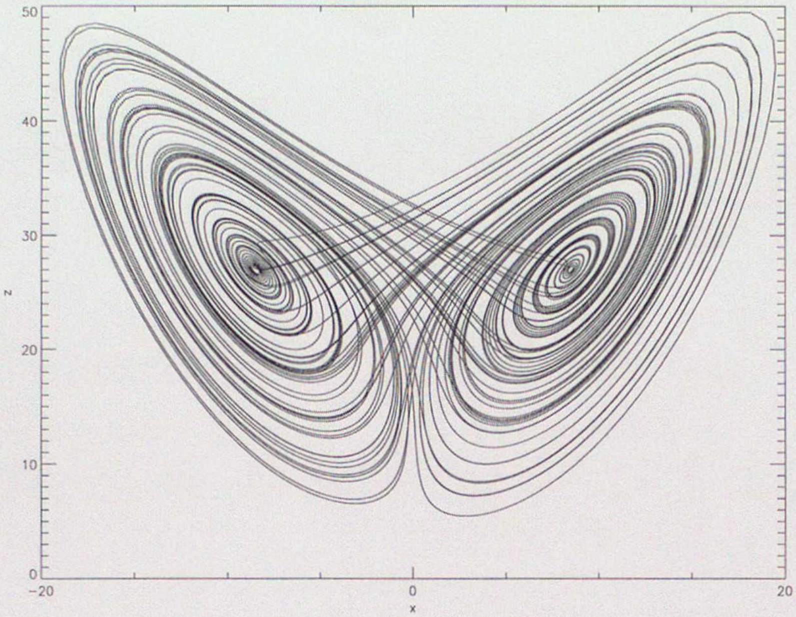

Figure 1: x-z plane of the Lorenz attractor

$$\frac{dy}{dt} = 28x - y - xz \tag{2}$$

$$\frac{dz}{dt} = -\frac{8}{3}z + xy \tag{3}$$

Solving eqns 1, 2 and 3 numerically gives rise to the famous butterfly shape shown in Figure 1. The system consists of two orbital planes, identified here by A for $x < 0$ and B for $x > 0$. Consider a point $\mathbf{x}(t)$ and we want to predict if $\mathbf{x}(t + t')$ is in A or B. Unlike numerical weather prediction the equations are explicitly defined so this is simple to determine given any t', subject to the numerical accuracy of the system used to integrate the above equations. If instead we consider a point $\mathbf{x}'(t)$ such that $0 < |\mathbf{x}'(t) - \mathbf{x}(t)| < r$ for some small $r > 0$, where r represents the observational accuracy. Now if we try and predict $\mathbf{x}(t + t')$ using $\mathbf{x}'(t)$ as a starting point the solution $\mathbf{x}'(t + t')$ will diverge from $\mathbf{x}(t+t')$ as $t' \to \infty$. The accuracy of predicting which plane $\mathbf{x}'(t+t')$ will reside will gradually reach 50%, i.e no skill. The value of t' at which this happens depends greatly on the value of $\mathbf{x}'(t)$. If $\mathbf{x}(t)$ is just beginning an orbit of plane B (i.e $x > 0$ and $\frac{dx}{dt} > 0$) then we know that $\mathbf{x}'(t + t')$ will stay in plane B for longer that for a point close to the cross over point ($x > 0$ and $\frac{dx}{dt} < 0$). Therefore by analysing the behaviour of the model we can predict the accuracy of the system at time $t + t'$ from its state at time t.

Although numerical weather prediction is a more complicated chaotic system it is possible to envisage that certain characteristics of the system are harder to model than others. Being able to recognise these characteristics would allow estimation of the confidence in the model from a given state.

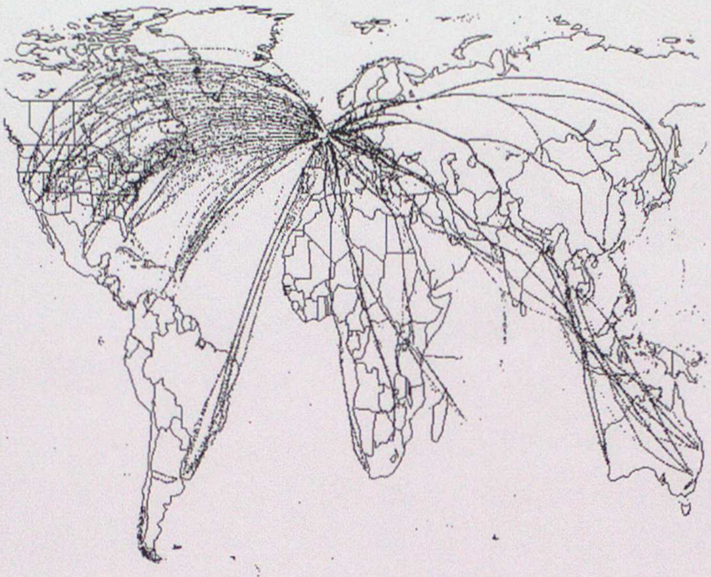

Figure 2: Global distribution of Mode C observational data

2 Method

2.1 Data

Actual observations or model analyses are required to train the network. Observations have the advantage of representing measured values, although some preprocessing must be carried out to protect the neural network from erroneous data. Using analyses allow the forecast and 'truth' domains to be matched exactly, providing 'truth' values for areas of the globe with relatively few observations. If the difference between the forecast and the analysis is the same order of magnitude as the difference between analysis and observations, then the use of analysed fields as "truth" would be ill advised. A predictor such as data density might help the modelling of analysis error, although this has not been carried out in this particular study.

The observations used in this study are GADS[1] Mode C data, high density observations of wind vector retrieved from British Airways Boeing 747-400 series flight data recorders. This data is not used within the assimilation process and is collected post flight, removing many of the problems inherent with real time aircraft data. The dataset includes measurements of wind speed and direction at 128 second intervals. The data is also organised into individual flights so the take off time can be used to more accurately determine the numerical forecasts likely to be used in flight planning.

These forecast fields were then interpolated onto these observations. The fields were chosen to represent the fields used when flight planning. Selection criteria were:

- data time at least 6 hours before take off time (00Z or 12Z).

[1] Global Aircraft Data Set

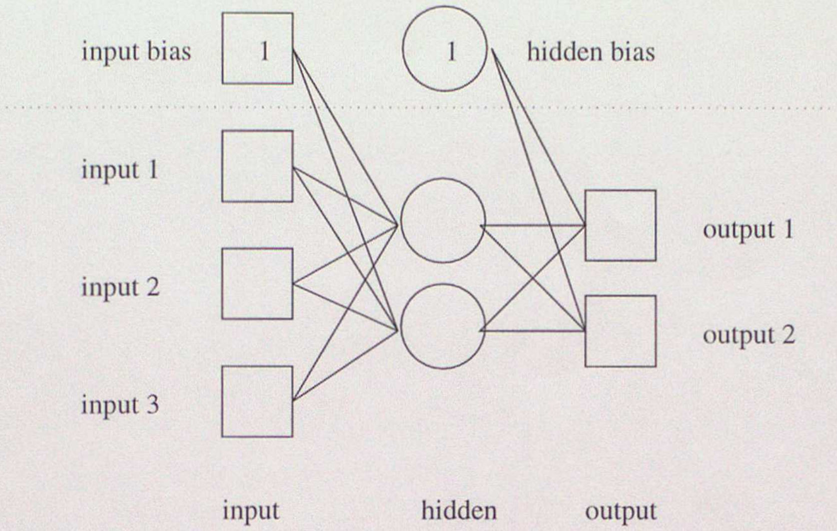

Figure 3: A example of the neural network topology used (in this example 3 predictors, 2 hidden nodes and 2 predictands).

- 4D linear interpolation between 200, 250, 300 and 400hPa pressure levels, and 6,12,18,24,30 and 36 hour forecasts. Observations outside these ranges were discarded.
- exclusion of all data from units with flight data that was clearly unphysical. This was checked by comparing recorded airspeed with airspeed derived from groundspeed and observed winds.

At each data point further information was derived from the forecast fields from which the predictors would be selected. This included u, v and the first partial derivatives in each dimension (derivatives in the horizontal plane were combined to form 2D divergence and vorticity). Lead time, longitude, latitude, pressure and orographic height were also stored. The error values could then be derived from the absolute value of forecast minus observation.

A similar method was used to produce the data from analyses. A regular grid of pseudo observations were created over the area of interest at a given pressure level (in this case 300hPa). These were then interpolated from the 00Z and 12Z analysis fields for January and February 1999. The predictors could then be derived from the forecast fields as described above, but without a specific take off time only 12 hour forecasts were used.

2.2 Neural Network

A multi layer perceptron was used, a detailed description of which can be found in any good book on artificial neural networks, [Bishop]. A single hidden layer was used between the input and output layers.

All inputs and outputs were normalised to [0,1]. Preprocessing by nonlinear scaling was not carried out on inputs or outputs. Each active layer had a constant bias input attached, set to 1. The sigmoid function $\left(f(x) = \frac{1}{1+e^{-\alpha x}}\right)$ was used as the activation function on both active layers. The weights were incrementally adjusted to minimise the training error function using the back-propagation method. Momentum and learning rate were set at 0.5. The spread factor of the sigmoid function was set to 1.

As the purpose of this work was to prove a concept these values were not tuned although they are consistent with the choices made in a similar study using surface data, [Grooms]. The other network parameters needed to be selected with reference to the training data set sizes.

2.3 Training

The data was split up into training, validation and test sets. The training data had to be kept small when testing, due to the length of time the network takes to train, at least when assessing different configurations. The training data consisted of a random selection of 50,000 points from February 1999. Another 50,000 points were used for validation as the network trained. The rest of this month was used for final testing before using the data from the whole of January 1999 to produce the final results. Ideally more data would be used for the training stage, but the associated increase in computation time made this infeasible.

It has been suggested that for noisy data a network with a larger number of hidden nodes and training epochs performs better, [Lawrence]. In fact the optimal training may be achieved just before the data becomes over fitted (although without care the data will be over fitted to both training and validation data without being able to generalise well to other data). By testing the network on a range of different data sets the possibility of over fitting should be reduced. A range of hidden nodes and epochs lengths were tested. One problem is that the optimal network configuration varies depending on the size of training data. Generally the more training and validation data used, for a given number of hidden nodes, the greater the number of training epochs can be iterated before over fitting occurs.

Rather than binning the data into latitudual bands, as in [Grooms], the data was selected by geographic region. A box of 20W to 130E and 0N to 85N was chosen as this represented a region where forecast errors have serious implications to aviation customers. This also had the effect of excluding regions, such as the data poor Southern Oceans, where the forecast error characteristics were likely to be different. Overall this gave a complete data set for February of 300,000 points usisng analyses (from a 100x50 grid), and 172,912 points using observations.

2.4 Choice of predictors

A selection of predictors to model forecast errors needs to be made from all the values derived from the model.

For example, forecast errors can occur,

- Over high orography, due to sub-optimal model parameterisations.
- Around jet cores, due to their small relative size and steep wind gradients.
- Near rapidly changing features, such as fronts.

The predictors need to represent as many of these kinds of problems as possible. The number of input nodes has a large impact on the number of training epochs required to train the network. The selection of appropriate predictors is therefore of prime importance.

A rough estimation of the usefulness of a range of predictors can be gained by working out the correlation coefficient between the inputs and outputs. For simplicity the vector components were combined to form magnitudes and everything squared to make inputs and outputs all positive. The following table shows how each compared for all the Mode C observational data, ranked in order of linear regression correlation coefficient against absolute wind speed error. The overall correlation coefficient using multiple linear regression is also given.

Ranking	Predictor Squared
1st	Wind speed
2nd	dwdp
3rd	Divergence
4th	dwdt
5th	Vorticity
6th	Orography
7th	Latitude
8th	Lead time
9th	Pressure
10th	Longitude
Overall R	0.205

The following shows a similar table for the subset of observational data falling between 20W to 130E and 0N to 85N, again ranked in order of correlation coefficient.

Ranking	Predictor Squared
1st	Divergence
2nd	Wind speed
3rd	dwdt
4th	dwdp
5th	Orography
6th	Vorticity
7th	Longitude
8th	Latitude
9th	Lead time
10th	Pressure
Overall R	0.210

Synoptic predictors appear to outperform spatial predictors in both cases. Interestingly lead time appeared quite far down the list and although it is where a relationship with error would be expected it is not as good as the other predictors. This is probably due to the distribution of flight data, in particular with the middle of routes typically over more data sparse areas that the beginning and end. Many of the long lead time observations were over, the relatively well forecast, Europe. The distribution of the observations was such that the use of spatial predictors, such as longitude, increased the possibility of poor network performance due to extrapolation between flight paths.

The following, mainly synoptic predictors were chosen, which all rate highly in the above rankings.

- $\frac{\partial u}{\partial t}$and $\frac{\partial v}{\partial t}$
- $\frac{\partial u}{\partial p}$and $\frac{\partial v}{\partial p}$
- *divergence* and *vorticity*
- u and v
- Orographic Height

Similar results were obtained from ranking the predictors from the analysis dataset, though the lead time and pressure were not included when using analyses as these were kept constant. The same choice of predictors were used for both analyses and observation based training, mainly for simplicity but this also allowed the results of the two networks to be compared.

More complicated predictors, including different fundamental quantities such as humidity, temperature or potential vorticity were not included. The first derivative predictors should be able to model timing and position errors. Orographic height gives a measure of the influence orographic parametrisations upon the forecast.

2.5 Network configuration

A range of hidden nodes needed to be tested, as well as various combinations of analyses and observations in predicting errors. The performance of the network can be measured as the training progressed. The normal way of measuring performance is by using the coefficient of determination (the R-Squared value), and measures the proportion of the total variance explained by the model. It takes values between 0 and 1, with 1 indicating a perfect fit.

The actual metric minimised during the training is the mean square error between the training network output and the actual outputs. Therefore the validation R-Squared value does not necessarily increase through training as the network can become over fit to the training data. Comparing training and validation R-Squared values is a good method of assessing if over fitting has occurred. It was desirable to be able to separate the errors in the u and v direction. This would enable flight planners to resolve possible errors in head wind and cross wind components, and therefore assess more fully the likely forecast flight time errors. Having more than one output does however complicate the training as we need to assess more than one R-squared value. It may therefore be desirable to train a different network for each output component, although this would again add to the computation time and has not been carried out at this stage.

A plot of validation u error R-Squared value as training progressed, for a range of number of hidden nodes is shown in Figure 3. There is some variability of final R-Squared value after 500 epochs due to the randomness introduced from the initial weights and data split.

To illustrate the effect of this randomness the training needs to be performed a number of times with the same network configuration. The results were similar

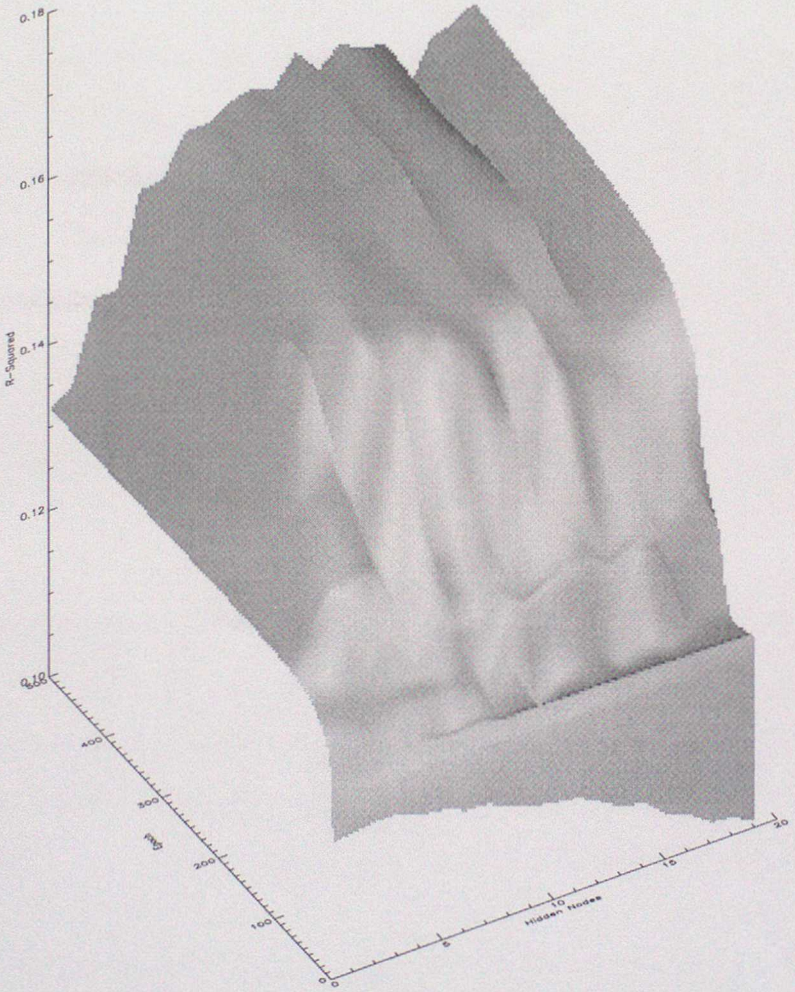

Figure 4: A comparison of the final R-Squared values achieved for different numbers of hidden nodes

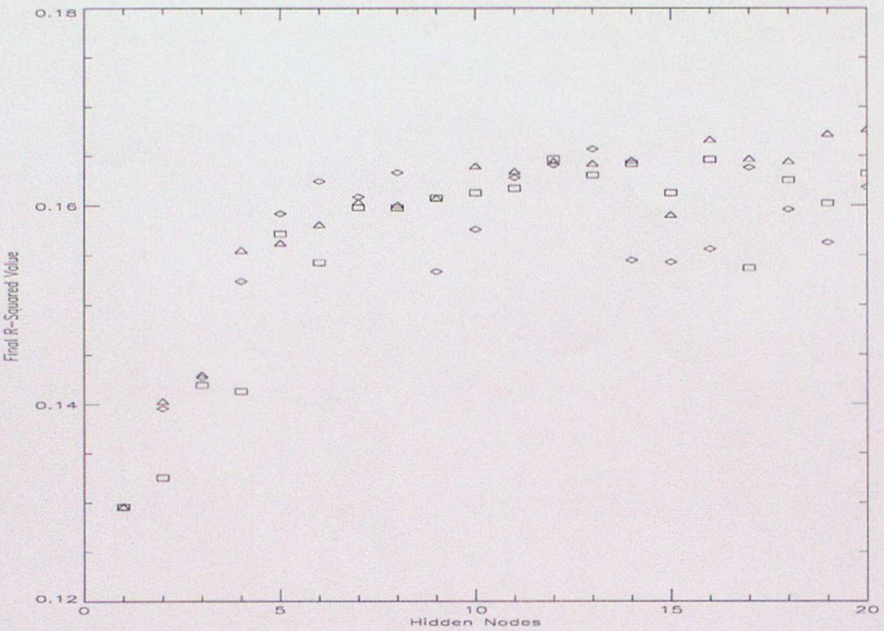

Figure 5: Final validation R-Squared values using network trained for 500 epochs.

for each case. The final test R-squared values for 3 different training experiments, using different data splits and initial weights, are shown in Figure 4.

Over fitting did not occur, mainly due to the large amount of data used relative to the number of epochs. The best validation error was therefore achieved on the last epoch. This inferred that benefit could still be gained by increasing the number of training epochs even further. By reducing the training and validation data sets from 50,000 observations to just 1,000 the effect of over fitting can be illustrated. A plot of R-Squared value as training progressed for a network with 10 hidden nodes is shown in Figure 5.

The network configuration of 10 hidden nodes with 50,000 observations for both training and validation was then tested to see how much improvement could be gained by increasing the training epochs to 5,000. The results as training progressed are shown in Figure 6. In this run the validation R-Squared value began to decrease after around 4,500 epochs, indicating that the network had become over fit.

The trial results indicate that enough data is available to train for a large number of epochs. This would also indicate that benefit can be gained from increasing the number of hidden nodes. Due to the large amount of time to train these networks the use of these larger networks was not tested. The network configuration explored above consisting of 10 hidden nodes was used for the rest of this study. The training shown in Figure 6. took just under 3 days to compute, so the amount of possible experimentation was limited.

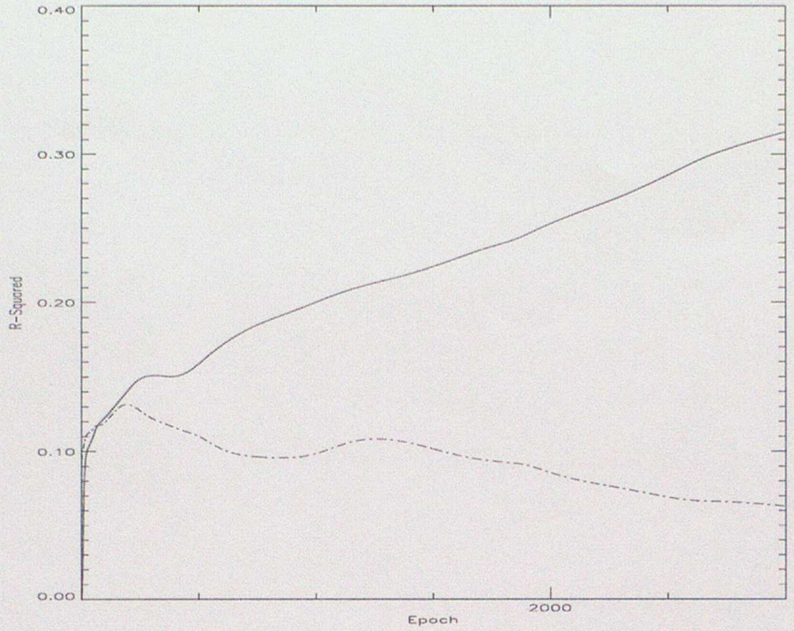

Figure 6: Example of over fitting with small training (solid) and validation (dashed) dataset size.

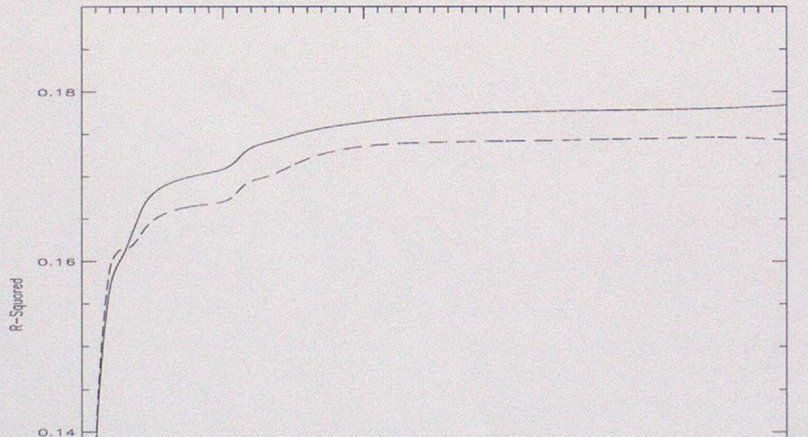

Figure 7: A long run of 5,000 epochs shows sign of over fitting after 4,500 epochs, using 50,000 data points for both training (solid) and validation (dashed).

3 Results

3.1 Using Analyses

The network was trained using the configuration described above. The weights were obtained for a network of 9 input nodes (u, v, $\frac{\partial u}{\partial p}$, $\frac{\partial v}{\partial p}$, $\frac{\partial u}{\partial p}$, $\frac{\partial v}{\partial p}$, div, $curl$ and Orography), 10 hidden nodes and 2 output nodes ($|u_{fore} - u_{anal}|$, $|v_{fore} - v_{anal}|$). A randomly selected 50,000 points from the February data was used to train the network whilst another 50,000 points were used for validation whilst training. The weights which achieved the greatest validation R-Squared value were taken to be the best trained weights. The final R-Squared value achieved for all the training and validation data using these weights was 0.1785 whilst the R-Squared value using these weights on the remaining two thirds of February was 0.1723. This occurred after around 4,500 epochs.

The trained network could then be used on the data from January, of which the trained network has no experience. It is the performance of the network in predicting error for this month which is important in terms of network evaluation. The final correlation coefficients for the trained network prediction of absolute u error and v error in January was 0.418 (R-Squared=0.174) and 0.335 (R-Squared=0.112), respectively.

3.2 Using Observations

The same network configuration as described above was used to test the network performance using observational data, i.e 9 inputs, 10 hidden nodes, 2 outputs. A training and validation dataset size of 50,000 observations was used, as with the analyses. The maximum validation R-Squared value occurred after 2,500 epochs. This shorter training time was probably due to the distribution of the observational data (along flight paths) requiring a smaller set of error features to be learnt. Multiple runs using data from different period would be required to confirm this.

The final R-Squared values for the training and validation data was 0.152, with the remaining February dataset giving a R-Squared value of 0.135. A number of reasons could cause this poorer performance, principly the error surface using observational data is more complicated due to errors introduced from using the observations. Further complications may be due to the minimal quality control performed on the observational data. In similar studies, [Grooms] and [Parrett], all errors over a predefined level were removed. In this study however only clearly unphysical measurements were removed as it was desirable to include many of the large forecast errors in the training dataset, as these errors were the most desirable to predict.

The observation trained network was then tested on the data from January. The correlations coefficients for the trained network prediction of absolute u error and v error in January was 0.318 (R-Squared=0.108) and 0.280 (R-Squared=0.078), respectively. These were again lower than the results achieved using forecast minus analysis errors.

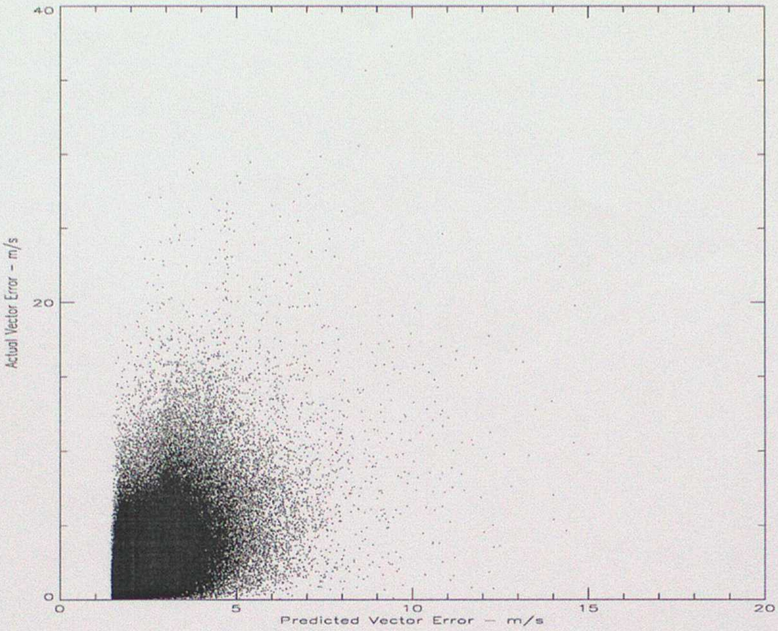

Figure 8: Plot of predicted vector error against actual vector error for January 1999 analysis dataset.

3.3 Uses of the network output

The skill of the network is heavily dependent on the forecast errors being generated by the features it has previously learnt, i.e those which are characterised by the chosen predictors and have occurred in the training data. There will be errors caused by factors which are not identifiable from the predictors, and therefore will not be forecast by the network. The distribution of the data, with a large proportion of the errors being small, also causes the network to favour underestimating the forecast error. This is shown more clearly by plotting the results for whole of January, combined to form vector errors. The performance of the network can then be studied by comparing individual points, see Figure 7.

Geographic plots of network output are more encouraging. The network seems quite successful at identifying regions of error, even if underestimating the actual size. Figure 8 shows one example forecast from January where the analyses network seems to have performed very well compared to the actual vector wind error derived from the forecast and analysis. It is also clear from this plot that the network sometimes predicts regions of error which are close, but do not coincide exactly with the regions of actual forecast error. In these cases any point based verification system would indicate a failure, where if used for flight planning the aircraft would most likely fly through the whole region of interest and the information would still be useful. This could be due to the lack of information about the model leading up to the forecast, with errors advecting along the flow. For example you would expect larger errors downwind of an area of high orography, even though the actual orography downwind is small.

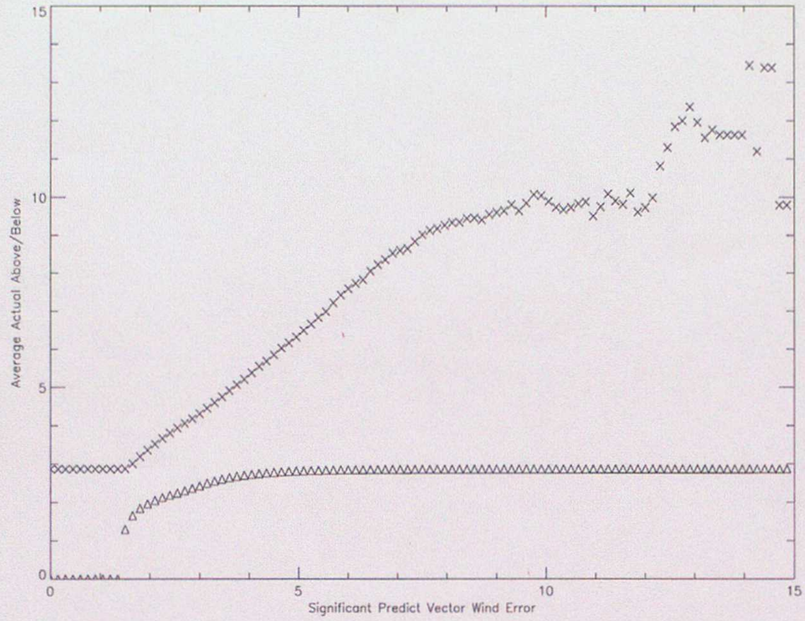

Figure 10: Average actual vector wind error above (X) and below (Δ) significant value

The network can be tested as a method of predicting error by using a contingency table. The average predicted vector wind speed error for January is 2.47 ms^{-1} whilst the average actual vector wind speed error is 2.86 ms^{-1}. Using this as our warning level gives the following results.

	Actual above mean	Actual below mean	Total
Predicted above mean	69351	48448	117799
Predicted below mean	47853	134348	182201
Total	117204	182796	300000

Using the above threshold gives a hit rate of 58.9% and a false alarm rate of 26.3% of predicting above average vector wind error. Different results will be gained from using different significance levels. This can be shown by varying the significant value of predicted vector wind error and calculating the average actual vector wind error above and below this level. This is shown in Figure 9.

4 Conclusions and Recommendations

Compared to multiple linear regression, improvements can be gained from using a multi layer perceptron to model forecast errors . There is certainly scope for further improvements, both to the system and in investigating how the output could be used.

Adding more predictors would allow the network to identify errors caused by different types of features, but this would also add more complexity to the network. It would also be necessary to train and test the system on a wider range of data, in particular data from different seasons. Using the network to predict error on a continuous training cycle could also be tested, assuming that forecast error features are consistent over a number of weeks. In this case including geographic predictors such as longitude and latitude could be of use. An estimate of the errors leading up to the forecast would also be useful, as many cases of poor forecast performance can be traced back to a poor original analysis.

The stability of the trained networks also needs to be investigated. Some testing was carried out by adding small perturbations to the inputs but this needs to be tested further. With such noisy data it is often beneficial to train a network many times, using different data splits and configurations. Using this ensemble of networks to produce the final output reduces the risk of over fitting, as well as indicating the confidence of each network output by producing a probability distribution of predicted error.

Forecasting numerical model error is a complicated and challenging task. The non-linear statistical estimation performed by using an artificial neural network goes some of the way to highlighting areas of the forecast where errors are likely to occur. Further work needs to be carried out in the training, tuning and validation of the neural network, in particular assessing how the results may be utilitised.

References

[Bishop] Bishop, C. M., Neural Networks for Pattern Recognition, 1995.

[Daley] Daley, R. Atmospheric Data Analysis, 1991.

[Grooms] Grooms, S., The Application of Artificial Neural Networks to Predict Short Range Numerical Weather Forecasting Errors, 1998, Internal Report.

[Hsieh] Hseih, W. W. and Tang, B., Applying Neural Networks to prediction and data analysis in meterology and oceanography, Bull Am Met Soc., 79, 1855-1870.

[Lawrence] Lawrence, S., Giles, C., Tsoi, A., What Size of Neural Network gives Optimal Generalisation?, NEC Research Institute.

[McNair] I.J. McNair and D.A. Forrester, Wind Forecast Accuracy Quantisation, 1997, Internal Report.

[Parrett] Parrett, C. A., Background Errors for Quality Control and Assimilation of Atmospheric Observations in the Unified Model - the situation in July 1992, 1992 Short-range Forecasting Division Technical Report No. 22.

[Sarle] Sarle, S., Neural Networks and Statistical Models, 1994, Proceedings of the Nineteenth Annual SAS Users Group International Conference.