THE USE OF RECURRENT NEURAL NETWORKS FOR CLASSIFICATION

T. L. Burrows M. Niranjan
Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, England

Abstract–Recurrent neural networks are widely used for context dependent pattern classification tasks such as speech recognition. The feedback in these networks is generally claimed to contribute to integrating the context of the input feature vector to be classified. This paper analyses the use of recurrent neural networks for such applications. We show that the contribution of the feedback connections is primarily a smoothing mechanism and that this is achieved by moving the class boundary of an equivalent feedforward network classifier. We also show that when the sigmoidal hidden nodes of the network operate close to saturation, switching from one class to the next is delayed, and within a class the network decisions are insensitive to the order of presentation of the input vectors.

INTRODUCTION

Many classification problems depend on the context in which class data is received, *ie.* the history of previous classes. Human perception of speech is a typical example, in which coarticulation effects between adjacent phonemes are important contextual factors for correct recognition, especially in noise. The performance of a classifier can be enhanced by providing past and future context. Future context can be provided by a delay between input window and output decision. Past context can be presented within an input window which contains a fixed number of previous frames [1], and by including delayed feedback paths (recurrent connections), which provide information about previous local decisions [2]. For a fixed input window, the depth of the context *ie.* the number of frames spanned by the input, is fixed. The classifier may miss dynamic features of the class with a longer duration than that of the input window and cause smoothing of features that change rapidly within this window. For a recurrent network, the depth of the context is potentially infinite, but in practice is determined by the relative size of the recurrent connection weights.

Much experimental work *eg.* [2], has reported improved performance of recurrent networks over feed-forward networks. In a previous paper [3], we looked at how this is achieved for the system identification of time-varying patterns. In this paper, we proceed by studying how recurrent networks operate for classification of time-varying patterns. We concentrate specifically on how the recurrent connections make use of previous context during 2-class classification problems such as classification of phoneme pairs from the TIMIT database.

EFFECT OF FEEDBACK ON DECISION BOUNDARY POSITION

Consider the unit delay recurrent connection around a single hidden node, with a nonlinearity $f(x) = \tanh(x)$, shown in Fig. 1. The output node is linear and the classification decision is determined by an output threshold at zero.

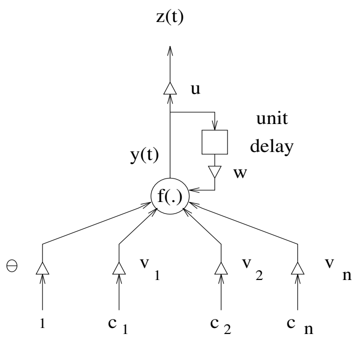

Figure 1: Single Hidden Node With Recurrent Connection

For such a network, the equations for the output of the hidden node, $y(t)$ and network output, $z(t)$, are :

$$y(t) = f\left(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}(t) + wy(t-1) + \theta\right) \quad (1)$$

$$z(t) = uy(t) \quad (2)$$

where $\boldsymbol{v}$ is a vector of input weights, $\boldsymbol{x}(t)$ is a vector of input parameters, $[c_1, c_2, \ldots, c_n]^{\mathrm{T}}$, θ is a bias term and $(.)^{\mathrm{T}}$ denotes transpose. We used cepstral coefficeints derived from phoneme segments from the TIMIT database as an example input. The networks were trained as 2-class classifiers using back-propagation through time to minimise the mean squared-error, with class targets of -1 and 1. The decision boundary is defined by :

$$\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}(t) + wy(t-1) + \theta = 0 \quad (3)$$

The contribution $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}(t) + \theta$, represents a static linear decision boundary which can be interpreted as the decision boundary for a feed-forward classifier which has the same weights u, $\boldsymbol{v}$ and θ. The term $wy(t-1)$ represents a variable bias which causes the decision boundary to move parallel to $\boldsymbol{v}$. We consider a trajectory of points in class 1, Fig. 2 a), for which some of the points are incorrectly classified by the static boundary. For these errors to be corrected, the decision boundary must move away from class 1, biasing the current decision towards that of the previous classification. This occurs for positive w, which also gives stable feedback around the node. Hence for maximum classifier performance, we require a training algorithm which develops positive w. The limits of the boundary movement lie at $\pm\, w$ on either side of the static boundary, and with this they divide the input space into 4 regions, A–D, as shown in Fig 2 b). Classification of points in A and D is unaffected by the position of the decision boundary and is independent of their context. A and D define a region of the input space in which the number of classification errors made by the recurrent net is predetermined. B and C define an indeterminate region of the input space, of width $2|w|$, in which the classification of points requires knowledge of their context, since movement of the decision boundary in this region causes both correction of and addition to errors made by the feed-forward net. The sensitivity of the output to the context of the input data implies that the order of presentation of the training classes is important. Different orders of presentation of the classes will not converge to the same solution, when starting from the same weight initialization.

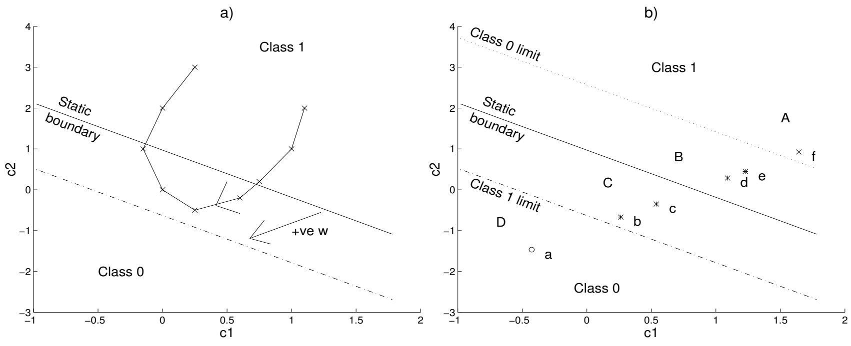

Figure 2: a) Movement of decision boundary by recurrent connection. b) Decision boundary limits and classification regions

EFFECT OF DECISION BOUNDARY MOVEMENT ON OUTPUT SWITCHING

The decision boundary movement, which biases the current decision towards that of the previous decision, gives a classifier output which exhibits a switching delay and is trajectory sensitive *ie.* is dependent on the order of presentation of input data within the current class. The magnitude of the delays and the extent of the trajectory sensitivity is determined by the relative range of the indeterminate region, B and C, and the *approximately linear* region of the node function, Fig. 3. All points in regions A and D are trajectory insensitive, since they cannot move the decision boundary. Only indeterminate points which lie within the linear region of the node function, shown hatched in Fig. 3, can cause boundary movement and are therefore trajectory sensitive. The entire indeterminate region will be trajectory sensitive if it is completely spanned by the linear region of the node function.

The variation in switching delay for different points within the indeterminate region is shown in Fig. 4. The decision boundaries and test data points for this classifier

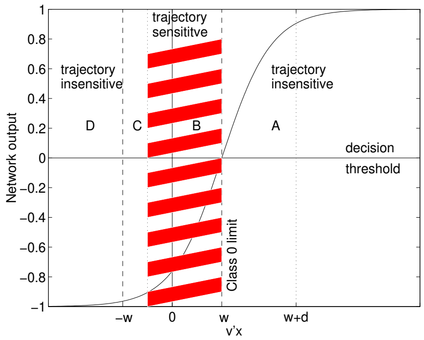

Figure 3: Trajectory sensitive region of input space when decision boundary lies at class 0 limit, for a nonlinear function which is linear over the range 2d and $\theta = 0$

are shown in Fig. 2 b). For a narrow nonlinear function, only a few indeterminate points are within the linear region and can cause switching. This switching is rapid since the boundary moves quickly across the indeterminate region to the other class limit. Most points cause saturation and no switching, Fig. 4 a). Hence most of the indeterminate region is trajectory insensitive and smoothing of the classifier output occurs *ie.* previous decisions are favoured. This is obvious in the limiting case of a step function, in which the linear width is zero. For this nonlinearity, the boundary can only lie at a class limit and only points in A or D cause switching. In this case, all indeterminate points are also trajectory insensitive and the previous decision is always chosen, giving maximum output smoothing. For a wide nonlinear function, Fig. 4 b), more of the indeterminate points fall within the linear region and are therefore trajectory sensitive, as shown in Fig. 3. The wider the linear region, the more slowly the decision boundary crosses the indeterminate region, giving longer switching delays and greater output smoothing. The actual boundary movement caused by a trajectory of points, **f** to **a**, which span the 'indeterminate region', is shown in Fig. 5 a), for $f(x) = \tanh(x)$. The boundary and data trajectory move in opposite directions, and due to the finite switching delay illustrated in Fig. 4 b), the recurrent decision lags the feed-forward decision, Fig. 5 b).

The limited trajectory sensitivity of the recurrent network is illustrated in Fig. 6, for a network with 10, 5 and 1 units in the input, hidden and output layer respectively. The network was trained as a classifier of voiced and unvoiced phonemes on sentences from the TIMIT database. In testing, adjacent input frames within a class segment were swapped and the classifier output compared with that for the normal input order. For most segments, the hidden nodes saturate, giving similar recurrent network outputs in Fig. c) and d). Only segments in which a node operates in the linear region (node output < 0.5 in Fig. 6 b).) are the network outputs very different. Smoothing of the recurrent network output within a class segment is seen in the unvoiced segments around frames 100 and 145 and a switching delay in the unvoiced/voiced class transition at frame 151.

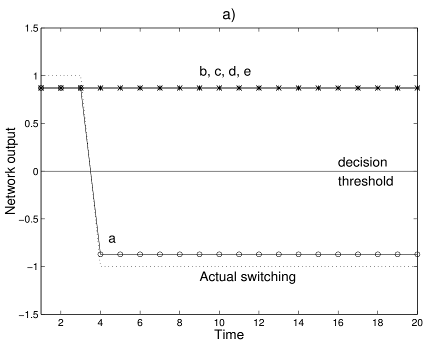

Figure 4: Delay in 1–0 switching, **f** to **a**: a) Narrow node function f(x)= tanh(10x). b) Wide node function f(x)= tanh(x)

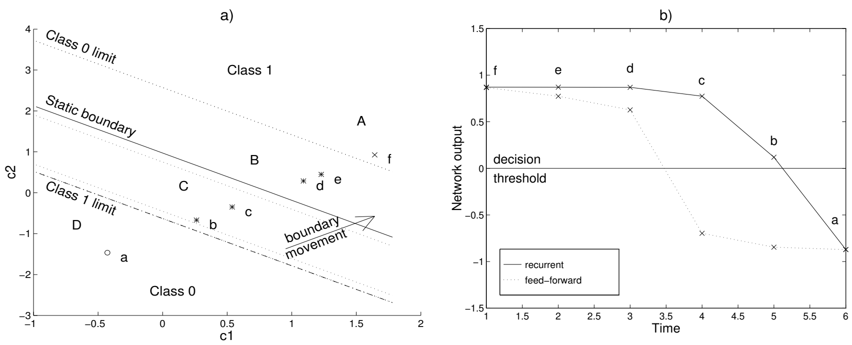

Figure 5: Movement of decision boundary due to a recurrent connection: a) Decision boundaries for 1–0 trajectory, b) Classifier output for 1–0 trajectory

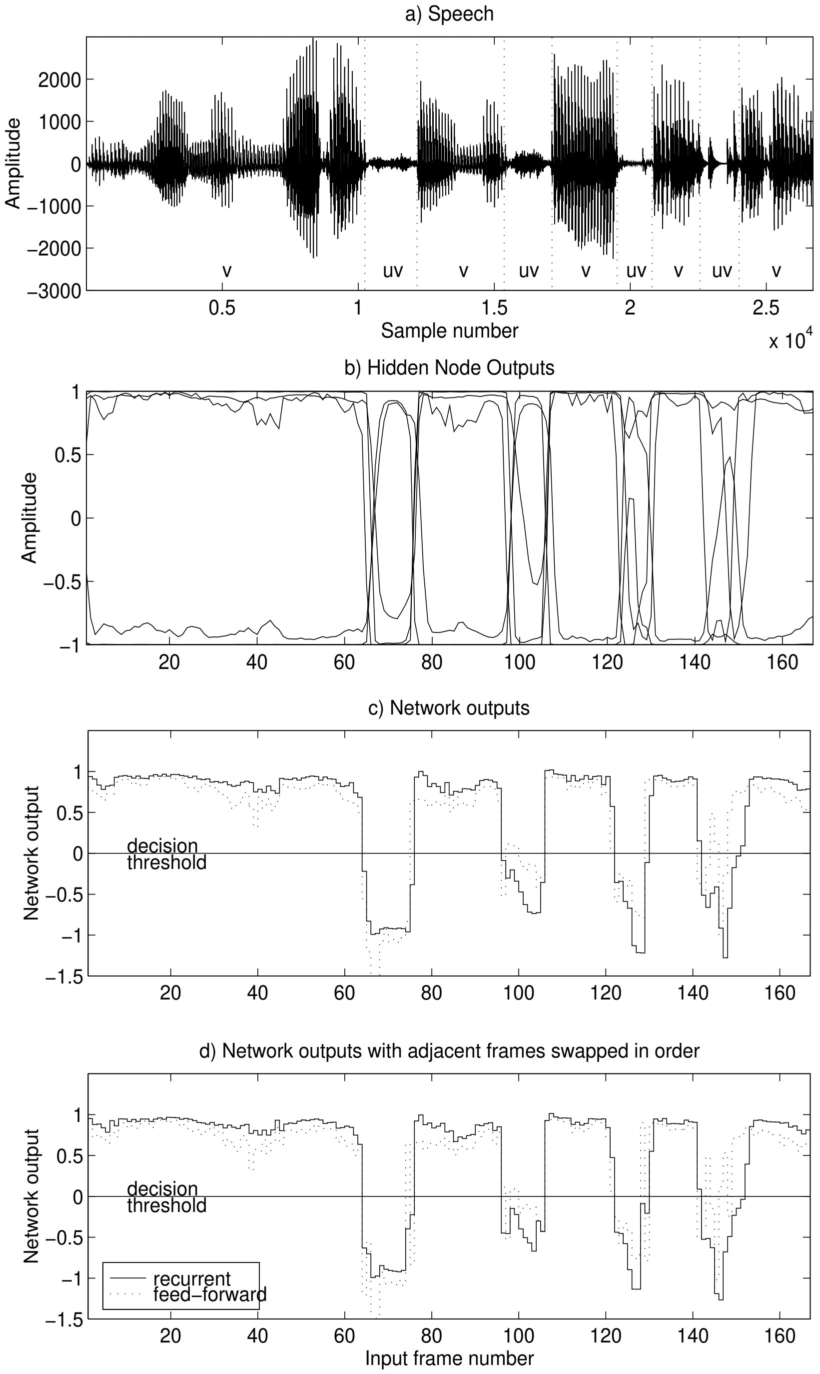

Figure 6: Voiced-unvoiced classification of phrase "a woman met a famous author"

EFFECT OF DECISION BOUNDARY MOVEMENT ON CLASSIFIER PERFORMANCE

The movement of the decision boundary has the potential to both improve and impede the performance of a recurrent net over that of a feed-forward net due to the variable classification of points in the indeterminate region, B and C. The recurrent net cannot correct errors in regions A and D. The combined effect of the trajectory sensitivity and switching delay caused by boundary movement, is to smooth output decisions of the recurrent net causing them to lag those of the feed-forward net. If the 'indeterminate region' is too narrow, Fig. 7 a), the feed-forward and recurrent outputs are almost identical and there is little difference in performance, Fig. 7 c). Conversely, if the 'indeterminate region' is too wide, Fig. 7 b), most classifications are dependent on previous decisions and over-smoothing of the output occurs, causing the performance of the recurrent net to fall below that of the feed-forward net, Fig. 7 d). Hence to minimise the additional errors of the recurrent network caused by switching delays, we require the indeterminate region to bind, as tightly as possible, any region of data overlap surrounding the static boundary.

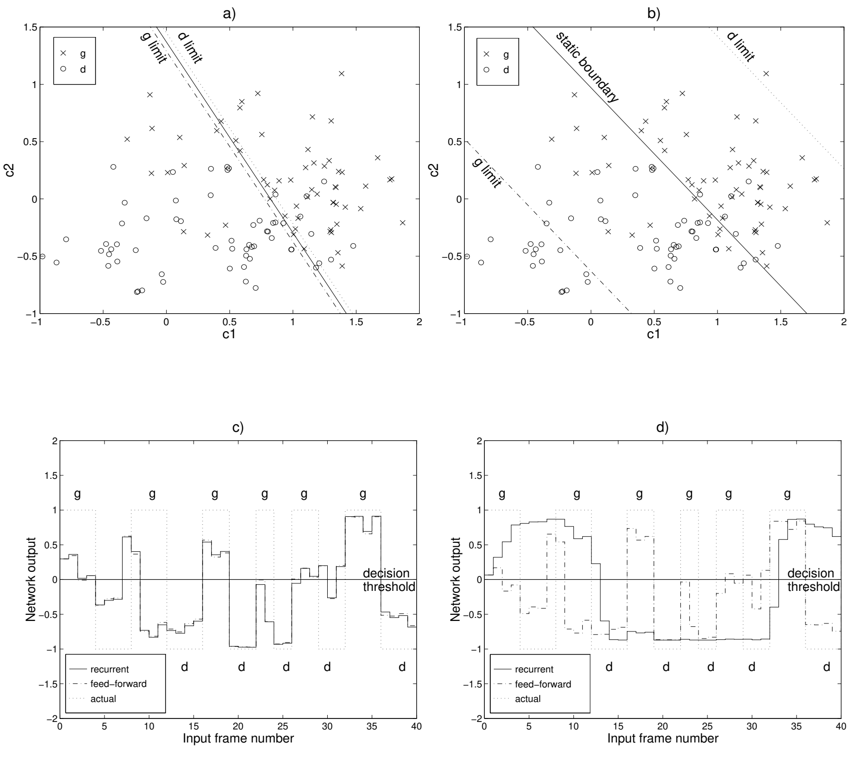

Figure 7: Test patterns and decision boundary limits for a 'g–d' classifier: a) w too small, b) w too large. Network output: c) w too small, d) w too large.

The smoothing of the recurrent network output can explain the change in relative performance of recurrent and feed-forward classifiers at different frame-rates (resolutions) [4], where recurrent networks are reported to perform better at lower frame-rates. At a higher frame-rate, there are more frames for a given phoneme duration but the parameters vary much less on a frame-to-frame basis than at a lower rate, causing saturation of the recurrent network. At a phoneme boundary, the small changes in parameter values at each frame cause the recurrent net to switch slowly, causing smoothing of the output decisions and a fall in performance below that of a feed-forward net.

DISCUSSION

Recurrent neural networks are widely used for context dependent pattern recognition. In speech recognition, for example, their application is motivated by the need to integrate acoustic cues that are distributed over time. It is generally claimed that this ability to model the temporal correlation in the data vectors gives recurrent neural network classifiers greater power than state-of-the-art acoustic models based on hidden Markov modelling. The observations reported in this paper suggest this may not be the case in practice. We have shown that the contribution of the feedback is primarily a smoothing operation. This can improve performance over a static classifier in regions of the input space where the class data may overlap, by moving the class boundary of the static classifier. The smoothing can also cause a delay in switching from one class to the next.

We also observed that when the hidden nodes operate in the saturated regions of the sigmoid, the network outputs are not sensitive to the order of presentation of the input examples within a class. When this happens, the network is not modelling the trajectory of the input vectors and is effectively treating each data vector within a class independently, similar to a hidden Markov model state. We suggest some of the above problems can be overcome by setting the network targets (or weighting the error function) in a similar manner to Etemad [5] and Watrous [6]. These authors use a ramp-like target function over the duration of a class, say a phoneme in speech recognition, to reflect the increasing confidence of class membership as more and more data is received. Such training will force the hidden units to stay out of saturation, avoiding some of the problems we have pointed out.

For the single hidden node recurrent net, a linear decision boundary, $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}$, is defined, with a bias of $\theta + wy(t-1)$. We have shown that the effect of w is to bias the current decision towards that of the previous decision, in a similar way to which the log prior ratio biases the decision boundary of a Bayes optimal descriminant function towards the most probable class [7]. We can interpret $wy(t-1)$ as acting like a variable prior ratio, since $wy(t-1)$ determines which class is favoured. The recurrent connection thus updates our estimate of the priors, depending on the previous context, $y(t-1)$. In [8], variation in the class priors between training and test data is accounted for by scaling the network outputs. Recent work on feed-forward nets by Gish [9] has shown that adjustment of the output biases is sufficient to adapt the classifier to the new data. For a recurrent net, this suggests that modifications to both the recurrent weights and the biases are necessary.

For a feed-forward network (multi-layer perceptron or MLP) with a single output node, training by back-propagation is known to yield a minimum mean squared-error estimate of the Bayes optimal descriminant function [10], in which the outputs are treated as posterior probabilities. The MLP approximation is only accurate if there are sufficient hidden nodes to capture the complexity of the function. With multiple hidden nodes, the decision boundaries become nonlinear and result as a combination of local decisions by each node. For the recurrent network case, cross terms in the feed-back matrix, $\boldsymbol{w}$ determine how previous decisions in other local regions of the input space affect the current local decision. We are now studying the multiple hidden node case more closely and expect the indeterminate regions for each local decision to overlap resulting in more of the input space being context sensitive.

REFERENCES

[1] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," Tech. Rep. TR-1-0006, ATR, October 1987.

[2] A. J. Robinson and F. Fallside, "Phoneme recognition from the TIMIT database using recurrent error propagation networks," Tech. Rep. CUED/F-INFENG/TR.42., Cambridge University, England, 1990.

[3] T. L. Burrows and M. Niranjan, "The use of feed-forward and recurrent neural networks for system identification," Tech. Rep. CUED/F-INFENG/TR158., Cambridge University, England, 1993.

[4] S. Renals, M. Hochberg, and A. J. Robinson, "Learning temporal dependencies in connectionist speech recognition," in *Advances in Neural Information Processing Systems 6*, Morgan Kaufmann, 1994.

[5] K. Etemad, "Phoneme recognition based on multi-resolution and non-causal context," in *Proc. 1993 IEEE Workshop on Neural Networks for Signal Processing* (C. A. Kamm, G. M. Kuhn, B. Yoon, R. Chellappa, and S. Y. Kung, eds.), pp. 343–352, 1993.

[6] R. L. Watrous and L. Shastri, "Learning phonetic features using connectionist networks: An experiment in speech recognition," in *Proc. 1987 1st International Conference on Neural Networks*, pp. 318–388, 1987.

[7] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[8] M. D. Richard and R. P. Lippman, "Neural classifiers estimate bayesian a posteriori probabilities," *Neural Computation*, vol. 3, no. 4, pp. 461–483, 1991.

[9] H. Gish and M. Siu, "An invariance property of neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (Adelaide)*, pp. 541–544, 1994.

[10] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a Bayes optimal descriminant function," *IEEE Trans. on Neural Networks*, vol. 1, no. 4, pp. 296–298, 1990.