N89-15595

CONTROLLING BASINS OF ATTRACTION IN A NEURAL NETWORK-BASED TELEMETRY MONITOR

Benjamin Bell - Electromagnetics Institute, Technical University of Denmark
James L. Eilbert - Grumman Corp., A02-026, Bethpage, NY 11714

Abstract

The size of the basins of attraction around fixed points in recurrent NNs can be modified by a training process. Controlling these attractive regions by presenting training data with various amount of noise added to the prototype signal vectors is discussed. Application of this technique to signal processing results in a classification system whose sensitivity can be controlled. This new technique is applied to the classication of temporal sequences in telemetry data.

1- Introduction

The ability to do associative retrieval and classify in the presence of of noise, plus their parallel nature makes NNs attractive pattern classification tools [3,5]. For the recurrent NNs used in this paper, pattern categories are defined by their fixed point attractors, and all patterns lying in the basin of attraction are classified as members of that category [1]. A region of attraction may be interpreted geometrically as a subspace of the input space containing one prototype vector, i.e. the fixed point. Researchers have been able to design fixed points into NNs [4,6], and to predict the minimum size of these basins for binary networks [7]. However, researchers have not been able to control the size of the basins through training.

For a NN pattern classifier, the size of its basins of attraction determine its sensitivity. In some problems, it may be necessary to classify every input as member of some category. In other cases, it may be more appropriate to classify a fraction of the inputs. Thus, a good pattern classifier must learn not only the patterns, but the desired sensitivity associated with each category. (In fact, the ability to place decision surfaces through learning has lead to the preeminence of feedforward networks as pattern classifiers among NNs [8].) This paper investigates the qualitative relationships between the learning parameters selected and the resulting sizes of the basins of attraction.

2- Sample Problem

The pattern pattern classification technique is applied to monitoring the temporal behavior of a satellite telemetry point. A short sequence of consecutive telemetry points are measured, and the difference between adjacent points is the data given to the NN. The role of the NN is to decide if the telemetry sequence should be identified as a member of one of six predefined categories shown in Fig. 1 or not. The sensitivity of the system determines whether an input resembling a prototype pattern will be identified as as instance of that category.

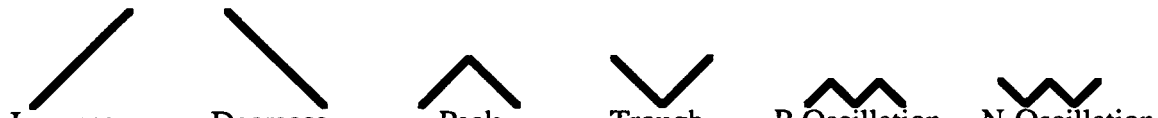


Figure 1. Predefined Prototype Patterns

3- The Representation of Patterns in NNs

3.a- The Network Model

The network model considered here is a recurrent network of discrete value elements. The network is fully connected, with its connection strengths maintained in an n x n matrix *A*, where A_{ij} identifies the strength of the synaptic connection from neuron *j* to *i*. The network is updated synchronously, i.e. every neuron is updated on each cycle. Initially, the activity of the nodes is a real number obtained from the telemetry data or synthetic training data.

The subsequent activities of each node are computed in two steps. First, a weighted sum of nodes inputs is computed to give a 'post-synaptic potential' (PSP), $s_i = \sum_j A_{ij} y_i$. The PSP can be either a positive or a negative number, whose magnitude is compared to the neuron's threshold, Q_i. The activity, x_i, is given by

$$x_i = \begin{cases} 1 * \mathrm{sgn}(s_i) & \text{if } |s_i| \geq Q_i \\ 0 & \text{otherwise} \end{cases}$$

These three-valued neurons allow a much broader set of outputs than binary neurons.

3.b- Placement of Category Prototypes

To create a connectivity matrix with particular fixed points, the outer product of each prototype vector with itself is computed, and the resulting matrices are summed over all prototypes. This process guaranties that the prototype vectors correspond to stable states, if they are mutually orthogonal [6]. For the telemetry problem six prototype vectors are placed in an eight dimensional space. The prototype vectors are made up of ± 1 values to place them on the outer boundary of activity space. In general, the outer product procedure leads to large basins of attraction around each of the prototypes. There also tend to be a few spurious fixed points with this approach.

3.c- The Learning Mechanism

The behavior of the NN may be modified by altering its connection strength matrix. The performance of the system can be improved by modifying connections so that the difference between the observed and the desired output are reduced. A variety of learning algorithm which achieves this type of improvement is the Widrow-Hoff or Delta rule algorithm [9].

Delta-rule learning calculates a delta from the difference between the final state reached and the final state desired. Each connection coming into a neuron then has its strength modified by an amount equal to the product of delta and the presynaptic activity. Note that the final result of learning depends on both the learning rate and the order in which the input/desired-output pairs are presented.

4- Training Design

Our purpose in applying a training sequence to the telemetry network is to change the shape of its basins of attraction so that the correct classification is achieved.

4.a- Approach

Two subsets patterns makeup a full training set associated with each prototype vector or category: a set of 'good' patterns that lies on the desired boundary of attraction, and a set of 'bad' patterns that lies a little beyond the the boundary. For simplicity, both the good and bad patterns were placed on a hypersphere. The range of radii used for the bad pattern hypersphere was 1-2 times the radius of the desired basin. To obtain a point on a hypersphere around a prototype, a random number is added to each component of the prototype vector, such that the Euclidean distance between the prototype and the constructed point is equal to the desired radius. The random numbers for each component are chosen to lie between 0 and the portion of the radius not yet accounted for. An example of applying this procedure is given below for a desired basin radius of 0.7:

Prototype Vector ---> Training Vector with total noise applied = 0.7

(1 1 -1 -1 1 1 -1 -1) ---> (1.475 1.006 -.897 -.858 1.018 0.792 -.569 -.935)

In order to apply the Delta rule, there must be a desired final state associated with every training vector. For the good training vectors, the desired final state is the prototype. However, for the bad training vectors, a reasonable desired state must be chosen. The method we chose for selecting the desired final state for the bad vectors was to use the corner closest to the input state. This approach worked better than the other two approaches tested: all unclassified vectors were sent to the origin, or to the compliment of the prototype nearest to the input (If the nearest prototype is (-1,1,1,-1), the desired output would be (1,-1,-1,1)). Training on bad vectors tended to destabilize the network when either the origin or the compliment were used as the desired state. This seemed to to occur, because to reach these desired states the trajectory often had to cross the basin of attraction of other category.

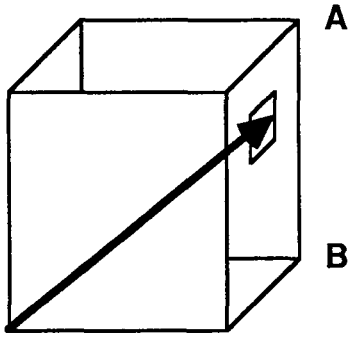
The desired final state of a bad pattern is calculated in the following way:

Find k such that $(|t_k| - |x_{0k}|) = \max(|t_i| - |x_{0i}|)$.

Reverse the sign of bit k of the closest prototype to obtain a desired vector that is Hamming distance one from the prototype.

Fig. 2 shows a bad input vector in R^3 and its respective prototype and desired final states.

C-5



The input vector C to be rejected is derived from the vector to corner A, and has desired final state of the next closest corner B.

Figure 2. Selecting the Desired Final State for Rejected Vectors

5- Testing Network Performance

To study the effects of training, training data for three different basins of attraction were studied using several different learning rates. For each set of parameters, the final matrix of connection strengths was stored for performance testing.

The connection matrices from the various learning sessions were tested by applying a set of test input vectors. The test data were chosen at random, and were constrained to lie at particular distances from the prototype vectors. Each trained network is scored on the number of correct classifications it achieves on the test data. The inputs are created to provide two different methods of estimating the attractive regions of a network. The first method uses test vectors whose distance from the prototype is normally distributed. One can then count the number of correct classification of a large number input points. The data shown in Fig. 3 agrees with the intuitive notion that training data lying on a larger hypercube around a prototype vector leads to a larger basin of attraction. It illustrates that the fraction of points converging to the prototype does increase as the size of the trained radius increases.

The second method measures the number of classifications made at randomly selected point lying on a sphere at a particular distance (eg. from 0.3 to 0.9 at intervals of .2), and demonstrates the fall off of attractive strength with distance. If a spherical basin of attraction is created by the training procedure then any points outside of the desired radius would not be classified and all of those within the radius would be classified. Fig.4 clearly shows that this does not happen. Instead one finds a significant drop in the percentage of test patterns classified as the radius of the test patterns falls below the desired radius. This implies that the resulting basin of attraction is not spherical, but that on the average it is approximately the right size.

The rates for both connections and thresholds have a significant effect on the performance of these networks. The threshold learning rate seemed to have a destabilizing effect as shown in Fig. 5. The majority of the patterns end up converging on the origin. Learning rates for connections have an optimal range in which learning can take place.

6- Discussion

There are several directions in which this work should be expanded. First, one needs to try more sophisticated learning algorithms than the Delta rule. The Delta rule involves the difference of two terms. The first is essentially an outer product of the input

and the desired output. The second is an outer product of the input and the actual output. However, the distance to where a trajectory actually ends up has more to do with where the nearby prototypes are rather than how large a change in the connection matrix must be made. One alternative is to scale the product of input and desired output by the following two factors: the initial rate of change away from the correct prototype, and divide by the difference between the input and the desired output.

Another direction of improvement involves making use of context information. The desired sensitivity of the telemetry classifier varies with time of day, season, and spacecraft activity. Thus, a more sophisticated NN pattern classifier would be able to select the desired sensitivity from input data. One approach is to construct a pair of NNs where the first NN identifies the sensitivity category based on context information, and use this information to set the parameters of the second NN which would actually classify the telemetry data.

In its current form, the network can provide useful information to an expert system. A satellite diagnostic system, for example, could identify a telemetry point as showing a steady increase if the network identifies this behavior several times in a row. Thus, repetition could partially overcome the fuzziness of the basin of attraction boundaries.

Although some ability to control basins of attraction has been demonstrated, the refinement in constructing decision surfaces through learning found in some feedforward networks [8] is a long way off. However, the pursuit of classification in recurrent networks remains an important goal. The brain makes use of processes running on several different time scales. For example, edge detection, allocation of attention, and learning are examples of processes whose temporal scale are at least an order of magnitude apart [2]. If NNs are to be used in producing cognitive capabilities, then the ability to use processes at different temporal scales is critical. At present, fixed points provide the only way of this type of communication.

Acknowledgements

This work was done in part at the Mathematics and Computer Science Department, Drexel University as an independent study project.

References

1- Anderson, J.A., Silverstein, J.W., Ritz, S.A., Jones, R.S.: Distinctive features, categorical perception, and probability learning: Some applications of a neural model. Psycho. Rev. 84:413-51 (1977).
2- Eilbert, J.L., Guez, A.: Attentional states and behavior modes in a hierarchical neural network. Proceedings First IEEE Conference on Neural Networks June 20, 1987.
3- Feldman, J.A., Ballard, D..H.: Connectionist models and their properties, Cognitive Science, 6:205-254.
4- Guez, A., Protopopescu, V., Barhen, J.: On the stability, storage capacity and design of nonlinear continuous neural networks, (to appear in IEEE Man, Systems, & Cybernetics).
5- Hinton, G.E., Anderson, J.A.: Parallel Models of Associative Memory. Hillsdale,N.J.: Lawerence Erlbaum Ass. 1981.
6- Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl.Acad. Sci. (USA) 79:2554-2558 (1982)
7- Peronnaz, L., Guyon, I., Dreyfus, G.: Collective computatinal properties of neural networks: New learning mechanisms. Phy. Rev. A. 34(5):4217-4229 (1986).

8- Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Parallel Distributed Processing v.1. Rumelhart, D.E., McCelland, J.L. (eds.). Cambrdige, MA: MIT Press. 1986.
9- Stone, G.O.: An analysis of of the Delta rule and the learning of statistical associations. In: Parallel Distributed Processing v.1. Rumelhart, D.E., McCelland, J.L. (eds.). Cambrdige, MA: MIT Press. 1986.

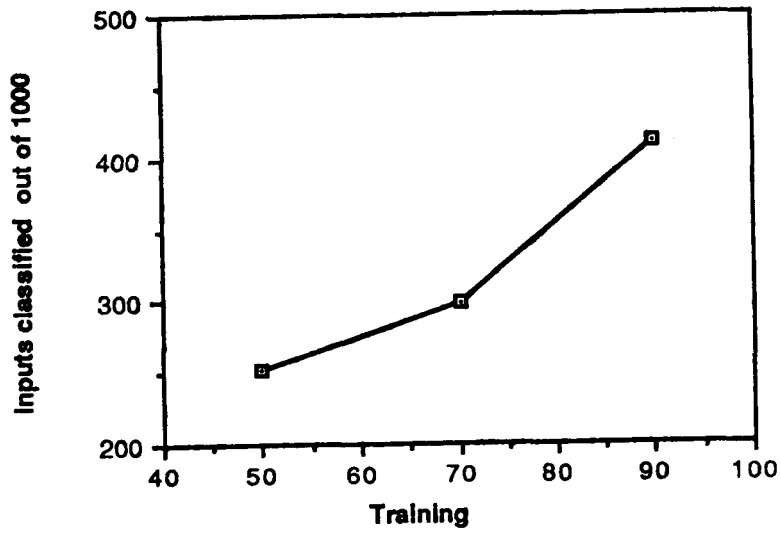


Figure 2. Effects of Training on Basin Size

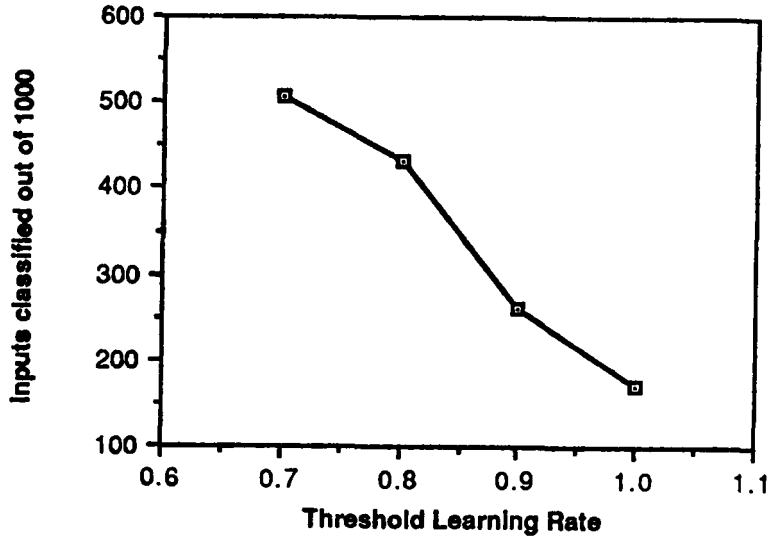


Figure 5. Effect of Threshold Learning Rate

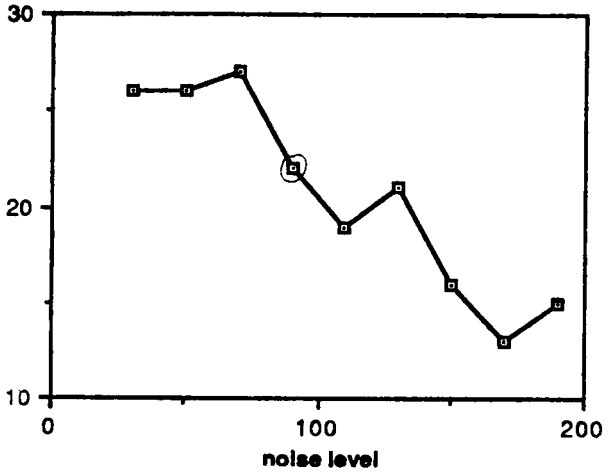


Figure 4a.
Basin Size

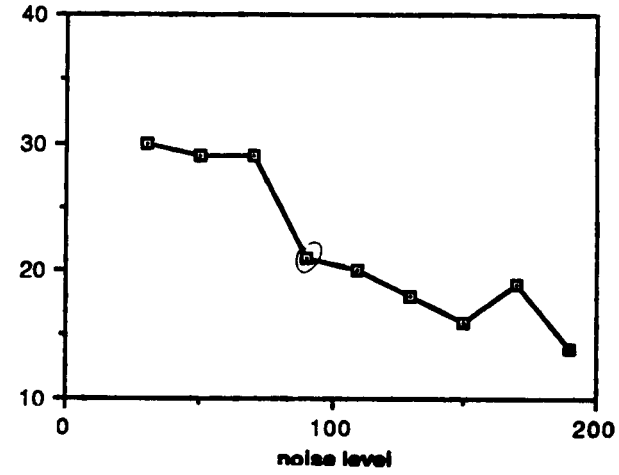


Figure 4b.

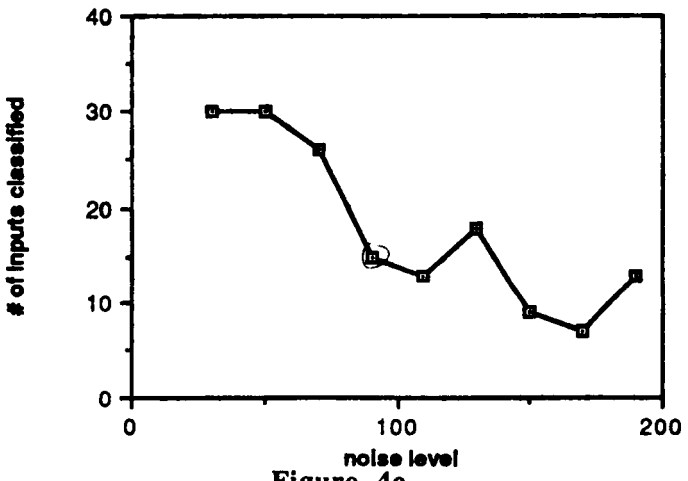


Figure 4c.

Figure 4. Basin Behavior of 3 Networks with Input on Increasing Spherical Boundaries