

A NEURAL NETWORK MODEL FOR PREDICTION ERROR IDENTIFICATION.

N. Z. Hakim*, J.J. Kaufman*, R.S. Siffert* and H.E. Meadows*

*Department of Electrical Engineering, Columbia University, New York, NY 10027.

*Department of Orthopaedics, Mount Sinai School Of Medicine, New York, NY 10029.

ABSTRACT

Neural networks can be a powerful tool for non-linear signal processing and systems modelling. We present a class of neural network based non-linear models suitable for application in the prediction error systems identification method. The parameters of this class of models include a local interconnection neighborhood size, a time constant which characterizes the neurons, a weight matrix, and input/output interconnection matrices. We also present some examples of the system behavior for different values of these parameters to evaluate the suitability of the neural network prediction error model to fit a wide class of systems. The method is then applied to train the net to perform three classical problems: generation of Van der Pol oscillations, estimation of a sum of cosines in additive white gaussian noise, and prediction of a chaotic time series generated by the logistic function.

INTRODUCTION

Recently neural networks have found applications in such areas as optimization, pattern recognition and control, often with encouraging results. However, their appearance in signal processing has been slower and very little has been published on this particular application area. Until now, the problem of system modelling and prediction with neural nets has been treated using feed-forward neural networks and a classical learning algorithm, the back-propagation algorithm [1]. This algorithm performs a steepest descent on the prediction error over a set of discrete time samples from a segment of a known signal. However this network lacks the properties of a dynamical system because of the absence of feedback between its elements. And although it achieves good performance on a class of ergodic time series it cannot be used as a generic class of models of dynamical systems, especially non-stationary ones.

System identification provides a powerful framework for modelling these systems and predicting their behaviour. In [2] Ljung defines system identification as "building mathematical models of dynamical systems based on observed data from the systems." To achieve this goal three entities are needed:

- 1- A sequence of data points from an experiment on the system.
- 2- A set of candidate models.
- 3- A criterion by which to evaluate the best model of the system.

With these basic entities the identification process can be carried out to select the model best fitting the data sequence. Many models have been suggested for linear time-invariant systems.

These models however, cannot represent all systems occurring in real life, for which non-linear models are needed. The problems become more complicated with non-stationary systems, especially when we have very little information about the system we are trying to model. To identify the parameters of the model, we follow the approach called "Prediction Error Method" introduced by Ljung and which provides a methodology for the actual identification process.

In section II this approach is presented while in section III a detailed description of the neural network model is carried out. After that we discuss the preliminary results we obtained and we postulate on the usefulness of these models to be used in system identification.

THE PREDICTION ERROR METHOD

To describe this approach, let Z^N be a sequence of data comprising N inputs and outputs generated by the real system at times $t=1,2,\dots,N$. For time series analysis, the set of inputs in this sequence corresponds to the empty set. The prediction error of a given model $M(\theta)$ is defined as

$$e(t, \theta) = y(t) - y^M(t; \theta)$$

where θ is a set of parameters describing the model, y^M is the output of the model and y is the output of the real system. Identifying the system is the process of minimizing a function of the prediction error over the whole data sequence. The prediction error cost function can be computed using a norm of the error sequence $e(t)$, $t=1,\dots,N$, which can be written as

$$V_N(\theta, Z^N) = 1/N \sum_{t=1}^N f(e(t, \theta))$$

where f is a scalar valued function. The optimal set of parameters, θ^* , is the one that minimizes the function V_N :

$$\theta^* = \arg \min V_N(\theta, Z^N).$$

The minimization of V_N can be carried out using many classical techniques. Among these we consider simulated annealing [3] or a fast version of it which has recently been proposed [4]. The Alopex algorithm [5] has also been applied as a stochastic descent for neural networks.

The problem we would like to solve is, given a time series, to find a model which is capable of predicting one or more future points in this time series, based on a set of past observations.

These observations may consist of the time series past itself but may also include observations from other time series (the multi-channel case). For this we define a model based on neural networks which we describe in the next section. The global approach we envision can be summarized by the following steps:

- Step 1: Initialize the model parameters with initial values.
- Step 2: Estimate the prediction error sequence based on the data set and the current model parameters.
- Step 3: Calculate the error cost function.
- Step 4: Update the parameters and repeat step 2 until convergence.

THE NEURAL NETWORK MODEL

Our neural network class of models is shown in fig. 1. It is built around a set of n neurons arranged in a linear or two-dimensional fashion. Each neuron is connected to the neurons belonging to a certain neighborhood defined by its size and topology. The connectivity between the neurons is expected to provide the dynamical behavior of the model. One or more channels are connected to an input layer of linear non-interconnected elements which maps these channels onto the neural net (middle layer) via an input matrix called M1. Similarly, an output layer of linear summation elements connects the neurons' outputs to a number of output channels via a connection matrix M2. While the elements of the input and output layers are linear, a continuous non-linear output function for each neuron is used in the middle layer and is illustrated in fig. 2. The dynamics of this system can be described by a set of differential equations

$$C_i du_i/dt = -u_i/R_i + \sum_{j=1}^n W_{ij} y_j + I_i$$

$$y_i = s(u_i)$$

where C_i and R_i are the input capacitance and resistance of neuron i , as described in [6]. W is the $n \times n$ weight matrix of the middle layer, and I_i is the input current to unit i . u_i is the output of the amplifier before the non-linearity s , and y_i is the output of neuron i .

The set of parameters on which the identification is carried out is the following:

1. The weight matrix of the middle layer of neurons. This matrix comprises n^2 elements where n is the number of neurons. To insure stability, W is restricted to being symmetric with zero diagonal elements.
2. The size and topology of the neighborhood. This parameter defines the connectivity of the neural net.
3. The average time constant of the neurons. Each time constant can be defined as the product $R_i C_i$ for neuron i where R_i depends on the weight matrix.
4. The input and output connection matrices. These matrices define a mapping of the input and output channels onto the neurons in the middle layer.

This set of parameters completely defines our class of models for the prediction error identification method. We simulated the system for several different values of these parameters on a deterministic cosine input function to investigate its behavior. The simulation was carried out by means of an adaptive step second order Runge Kutta method. Fig. 3 shows the dramatic effect of the size of the neighborhood on the output of the system.

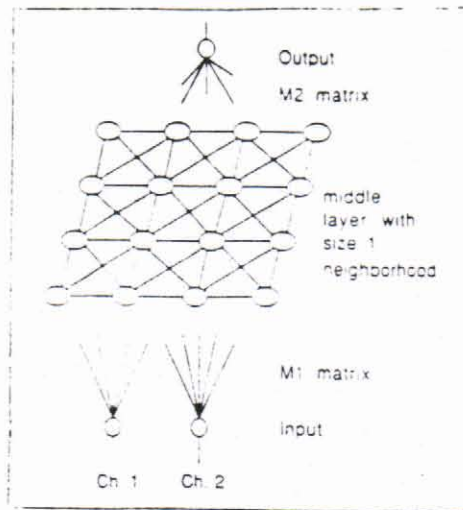


Fig. 1 Structure of the neural network class of models for prediction error identification method.

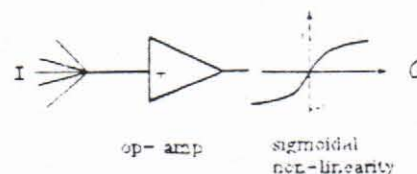


Fig. 2 Basic structure of the neuron with a sigmoidal non-linearity.

In a similar way, keeping the size of the neighborhood constant and varying the average time constant of the neurons produces a wide range of dynamical behavior. The outputs range from almost a clipping behaviour for a small time response to a delayed low pass filtered version of the input and between these two extremes, lies a rich set of intermediates which may prove useful for identification. In fig. 3, 9 neurons were used. The other parameters were chosen $M1 = [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1]$, $M2 = [0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0]$, time constant = 0.3, neighborhood size = 3, and W was generated randomly with a uniform distribution, always with the same seed.

APPLICATIONS

In order to investigate the potential of our non-linear class of models we investigated its performance on a few classical problems. The first one consists of emulating a non-linear system typified by the Van der Pol oscillator. This system behaves according to the differential equation

$$d^2x/dt^2 + \mu(1-x^2) dx/dt + x = 0$$

It admits a limit cycle, toward which all trajectories tend. For this problem, 9 fully connected neurons were used with time constants between 0.1 and 10. The input to the net was an impulse and the output consisted of 2 channels corresponding to x and its derivative dx/dt . The training data was generated using Van der Pol's equation with $\mu=1$. After training the net on 30 points, the neural net admitted the same limit cycle as the one of the training set, thus successfully emulating the oscillator. Fig. 4 shows a trajectory and the limit cycle respectively of the oscillator and the trained neural net.

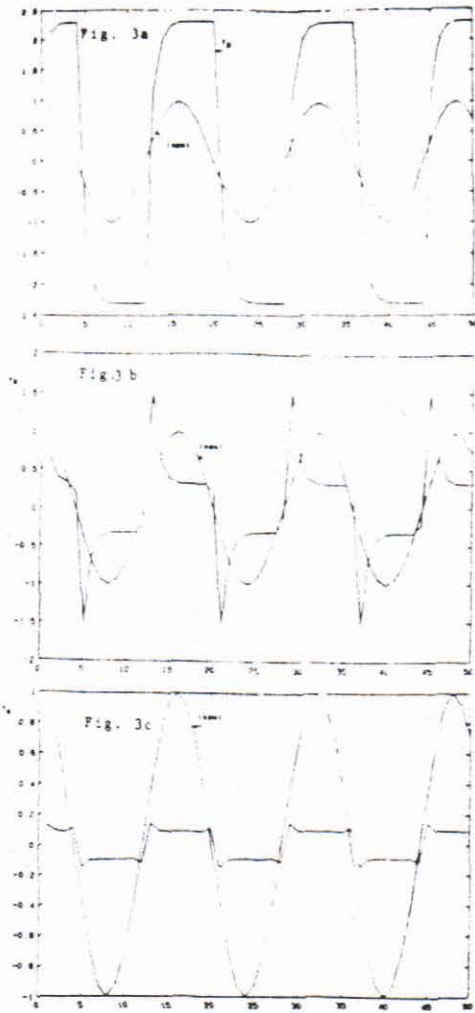


Fig.3 Response of the model to a cosine function input and different neighborhood sizes. All parameters are constant : $n=9$ neurons connected in a 1-D pattern. the sizes of the neighborhoods were 1 in 3a, 3 in 3b and 8 in 3c.

Another application was considered, which consists of estimating a sum of cosine signals in noise. We chose three cosines of frequencies 1, 3 and 10 respectively in additive gaussian noise with zero mean and unit variance. Training was performed using the noisy signal as input to the net and the clean sum of cosines as a desired output. Minimizing the error between the dynamical response of the neural net and the desired function allowed us to train the net. The performance of the neural net on the training set and on 25 subsequent points is illustrated in Figs. 5. Although the error on the last 25 points in Fig 6b is considerably larger than the one in the previous 25 points, which belong to the training set, the network is often able to extract the desired signal, even when the noise component is large. For this problem, 9 fully connected neurons were also used with time constants between 0.1 and 10.

The last application is the prediction of the logistic function, which in its discrete version can be described by

$$X_{n+1} = k X_n(1-X_n).$$

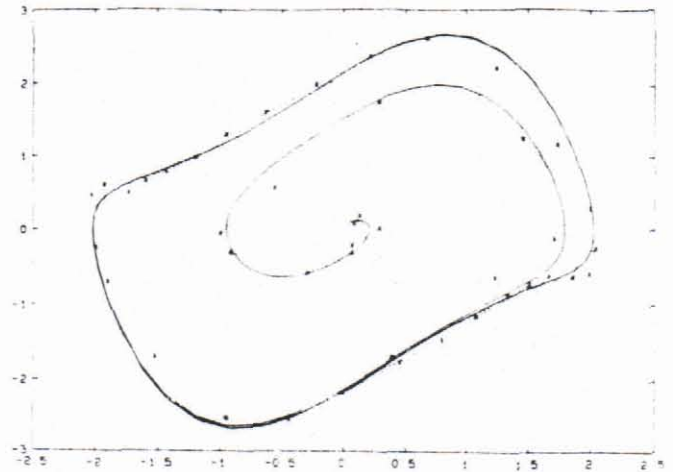


Fig.4 Neural net emulation of the Van der Pol oscillator. The solid line shows a trajectory and the limit cycle of the oscillator. The dotted line shows the behavior of the neural net after training.

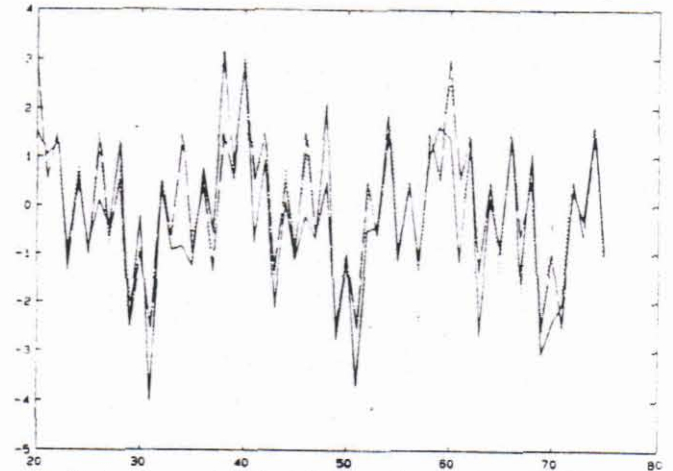


Fig.5 Neural net estimation of a sum of cosines in noise. The solid and broken lines represent the input and desired output of the net, while the dotted line represents the actual output of the estimator. The first 30 points in this graph are from the training set.

For the purpose of the prediction we chose a value of $k=4$, for which the series is quite chaotic. The structure of the net also consisted of 9 fully connected neurons with time constants ranging between 0.1 and 10. The training was performed by presenting the net with 50 consecutive points in the time series, and a desired output consisting of the one-step ahead prediction. The net learned to emulate the logistic function and was able to predict it quite accurately both on the training set and on the whole time series, including points it never saw before. Figs. 6a and 6b show the neural net's performance on the training data and on the following 50 points respectively.

DISCUSSION AND CONCLUSION

The choice of error criterion, mentioned earlier, is critical for the performance of a given model. For the particular case of a least mean square error criterion, the optimal predictor is known to be the conditional mean estimator. For gaussian statistics, it is known that the optimal predictor is a linear system. Linear system

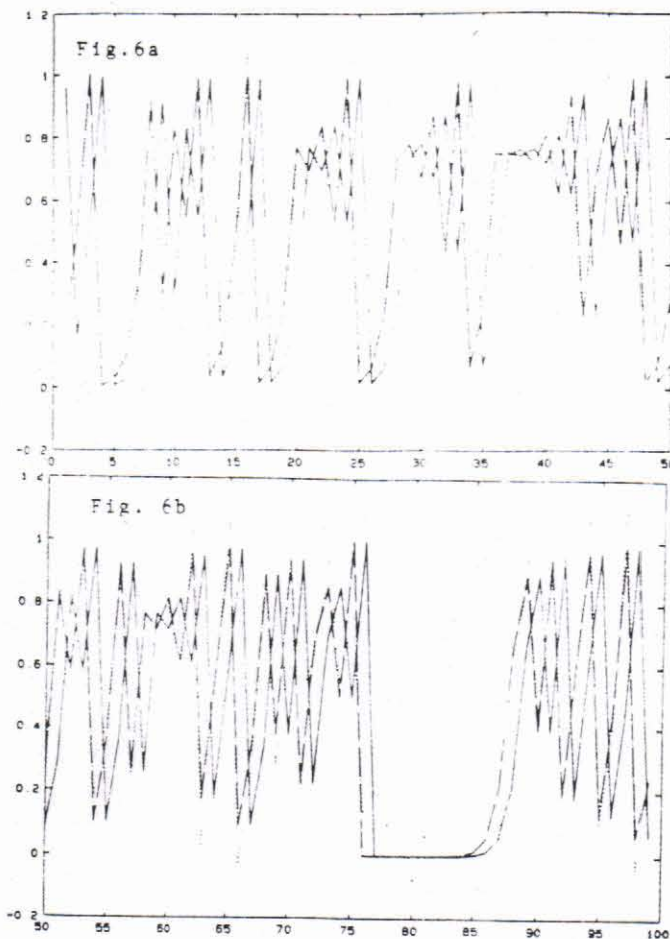


Fig.6 Neural network prediction of the logistic function. Fig 6a shows the performance of the net on the training data, while Fig.6b shows its response to data not seen before. Solid and broken lines represent the input and desired output of the net while the dotted line represents the actual output of the predictor.

models however, have a limited scope of applications especially in time series where often a chaotic behavior is observed, a characteristic of non-linear dynamical systems. The application of linear predictors to non-gaussian stochastic processes leads in general to non-optimal performance. This prompted the development of new methods such as higher order spectral estimation for non-Gaussian processes. In [7] a framework for defining a criterion is presented to allow a good generalization performance in layered neural networks. Our approach is different from this one but aims to achieve the same goal. We are currently investigating the usefulness of the above neural network model in the identification scheme. The minimization of the prediction error cost function is initially to be performed by the fast simulated annealing algorithm [4]. This is a stochastic descent technique similar to conventional simulated annealing but which uses Cauchy distributions (instead of Gaussian) to allow a more efficient search in the parameter space. Together with this numerical approach, we are investigating the derivation of an analytical updating rule for the system by expressing the partial derivative of the cost function with respect to each of the parameters [8]. Even if the latter approach can be solved only for a subset of the parameters, the weights in the

weight matrix for example, we could envisage a combination of simulated annealing and the analytical solution to provide more rapid convergence.

Another theoretical issue that could lead to more effective computation of the optimal parameter vector is neural network canonical forms. We are exploring various neural network parameterizations which have a minimal number of parameters. This should improve the efficiency and accuracy of the identification scheme. We are also investigating the development of a cost function which accounts for neural network complexity, similar to the Akaike information criterion [2].

The success of the identification scheme depends on the correct choice of the cost function. In the same line of thought as the mean square cost function, one could envision a higher exponent affecting the prediction error instances, possibly fractional and coupled to a forgetting function. This function would put emphasis on the most recent occurrences in the time series and yet not neglecting totally the previous ones, as happens in the back-propagation models previously proposed, where the number of inputs in the neural network correspond to a window in the time series rather than to a number of channels.

Neural network prediction error identification is a new approach which may have significant applications in time series analysis, prediction and modeling. Further research, both analytical and computer simulation studies, is necessary to determine if this approach can provide performance of sufficient quality to make it a practical alternative to current system identification methods. The work presented here should provide the basis for further studies.

ACKNOWLEDGEMENT

This research was supported by NSF grant #BCS - 8914242.

REFERENCES

- [1] A. Lapedes, R. Farber, "Non-linear Signal Processing Using Neural Networks: Prediction and System Modeling," Los Alamos National Laboratory, Technical Report LA-UR-87-2662, 1987
- [2] L. Ljung, "System Identification for the User," Prentice Hall Inc., New Jersey, 1987.
- [3] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "Optimization by Simulated Annealing," Science, Vol. 220, pp. 671-680, 1983
- [4] H. Szu, "Fast simulated annealing," Proc of the AIP Conference on Neural Networks, Snowbird Utah, pp. 420-425, 1986
- [5] E. Harth, "A Universal Optimization Network," Proc. of a Special Symposium on Maturing Technologies and Emerging Horizons in Biomedical Engineering, Myklebust, J.B., Harris, G.F., eds., IEEE, pp. 97-107, 1988
- [6] Hopfield, J.J., "Neurons with graded response have computational properties like those of 2-state neurons", Proc. Natl. Acad. Sci., Vol 81, pp. 3058-3092
- [7] N. Tishby, E. Levin, S.A. Solla, "Consistent Inference of Probabilities in Layered Networks: Predictions and Generalization," Proc. of the 1989 Joint Conference on Neural Networks, Washington D.C., Vol. 2, pp. 403-409, June 1989.
- [8] F.J. Pineda, "Generalization of Backpropagation to recurrent and higher order neural networks," Proc. of the IEEE Conf. on Neural Information Processing Systems. New York: IEEE.