Communicated by Michael Jordan

FIR and IIR Synapses, a New Neural Network Architecture for Time Series Modeling

A. D. Back
A. C. Tsoi
Department of Electrical Engineering,
University of Queensland, Queensland 4072, Australia

A new neural network architecture involving either local feedforward global feedforward, and/or local recurrent global feedforward structure is proposed. A learning rule minimizing a mean square error criterion is derived. The performance of this algorithm (local recurrent global feedforward architecture) is compared with a local-feedforward global-feedforward architecture. It is shown that the local-recurrent global-feedforward model performs better than the local-feedforward global-feedforward model.

1 Introduction

A popular class of neural network architecture, in particular, a multilayer perceptron (MLP) may be considered as providing a nonlinear mapping between an input vector, and a corresponding output vector (Lippman 1987). From a set of input and output vectors, an MLP with a given number of hidden layer neurons may be trained by minimizing a least mean square (LMS) cost criterion.

Most work in this area has been devoted to obtaining this nonlinear mapping in a static setting, that is, the input–output pairs are independent of one another. Many practical problems may be modeled by such static models, for example, the XOR problem and handwritten character recognition.

On the other hand, many practical problems such as time series forecasting and control plant modeling require a dynamic setting, that is, the current output depends on previous inputs and outputs. There have been a number of attempts to extend the MLP architecture to encompass this class of problems. For example, Lapedes and Farber (1987) used an MLP architecture with linear output units, rather than nonlinear output units. The linear output units allow the output values to be real rather than discrete as in classification problems. Waibel *et al.* (1989) used a time delay neural network architecture that involves successive delayed inputs to each neuron. All these attempts use only a feedforward architecture, that is, no feedback from later layers to previous layers. There are other approaches that involve feedback from either the hidden layer or from the output layer to the input layer (Jordan 1988). This class of network is known broadly as recurrent networks. In one way or the other, all these approaches attempt to incorporate some kind of contextual information (in our case, the dynamic nature of the problem is the context required) in a neural network structure. However, these are not the only neural network architectures that can incorporate contextual information.

In this paper we will consider a class of network that may be considered as intermediate between a (global) feedforward architecture and a (global) recurrent architecture. We introduce architectures that may have local recurrent nature, but have an overall global feedforward construction. Our contribution is the derivation of a training algorithm that is based on a linear adaptive filtering theory. The work presented here is similar to Robinson's (1989), except that in his network the feedback occurs globally, whereas in ours the feedback is local to each synapse. It is shown by simulation that networks employing this local-feedback architecture perform better than those with only local feedforward characteristics.

The structure of the paper is as follows: in Section 2, a network architecture is introduced. In Sections 3 and 4 training algorithms for the FIR synapse case and IIR synapse case, respectively, are derived (the nomenclature will be clarified in Section 2). In Section 5, the performance of an IIR synapse case is compared against an FIR synapse.

2 A Network Architecture

In a traditional MLP architecture, each synapse is considered as having a constant weight. Using the same methods as introduced by Lapedes and Farber (1987), the dependency of current outputs on previous inputs may be modeled using the following synaptic model.

$$\hat{y}(t) = \sum_{j=0}^{M} b_j z(t-j) \tag{2.1}$$

where $\hat{y}(t)$ is the synapse output at time t, $z(t-j)$ is a delayed input to the synapse, and b_i, $i = 0, 1, 2, \ldots, M$ are constants.

This synaptic weight is the same as a finite impulse response filter (FIR) filter in digital filter theory. As a result, we will denote this synapse an FIR synapse (Fig. 1).

On the other hand, the output may be dependent on both the previous inputs and outputs. In this case, we have the following model. Let q^{-j} $x(t) \equiv x(t-j)$. Then

$$\hat{y}(t) = \frac{\sum_{j=0}^{m} b_j q^{-j}}{1 - \sum_{j=1}^{n} a_j q^{-j}} z(t) \tag{2.2}$$

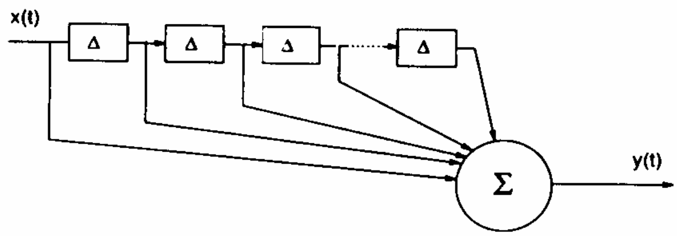

Figure 1: An FIR synapse.

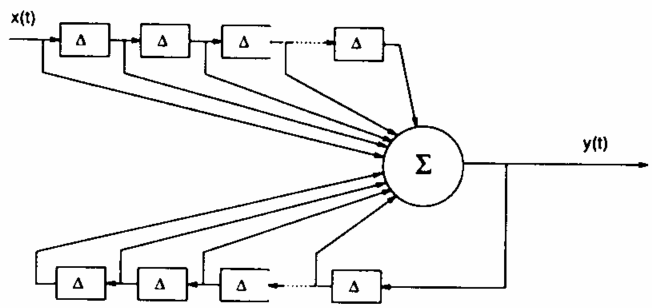

Figure 2: An IIR synapse.

This is called an infinite inpulse response (IIR) synapse (Fig. 2).

An MLP may use FIR synapses, IIR synapses, or both. Note that this type of network is still globally feedforward in nature, in that it has a global feedforward structure, with possibly local recurrent features (for IIR synapses). Thus, in the FIR synapse case, we will have a local-feedforward global feedforward architecture, while in the IIR synapse case, we will have a local-recurrent global feedforward architecture. It is obvious that a more complicated structure will be one involving both FIR synapses and IIR synapses. Figure 3 shows the neuron structure.

Consider an $L+1$ layer network. Each layer consists of N_l neurons. Each neuron i has an output at time t as $z_i^l(t)$, where l is the index for the layer, $l = 0$ denotes the input layer, and $l = L$ denotes the output layer.

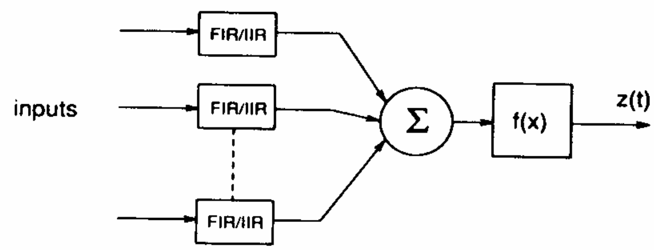

Figure 3: A neuron showing IIR synapses.

An MLP with FIR synapses can be modeled as follows:

$$z_k^{l-1}(t) = f\left[\hat{x}_k^{l+1}(t)\right] \tag{2.3}$$

$$\hat{x}_k^{l-1}(t) = \sum_{i=1}^{N_l} B_{ik}^{l+1}(q^{-1}) z_i^l(t) \tag{2.4}$$

where

$$B_{ik}^{l+1}(q^{-1}) = \sum_{j=0}^{M} b_{ikj}^{l+1}(q^{-j}) \tag{2.5}$$

$$f(\alpha) = \frac{e^{\alpha/2} - e^{-\alpha/2}}{e^{\alpha/2} + e^{-\alpha/2}} \tag{2.6}$$

$$k = 1, 2, \ldots, N_{l+1} \text{ (output layer index)} \tag{2.7}$$

$$l = 0, 1, 2, \ldots, L \tag{2.8}$$

$$M = \text{number of delayed inputs to a neuron} \tag{2.9}$$

$$\hat{x}_k^l|_{k=N_l} = \text{bias} \tag{2.10}$$

Note that we have made the simplifying assumption that each neuron receives the same number M, delayed inputs from the previous layer. This can be made to vary for each neuron. It is not used here since it would add unnecessary burden to the notation.

An MLP with IIR synapses can be modeled as follows:

$$z_k^{l+1}(t) = f\left[\hat{x}_k^{l+1}(t)\right] \tag{2.11}$$

$$\hat{x}_k^{l+1}(t) = \sum_{i=1}^{N_l} \frac{B_{ik}^{l+1}(q^{-1})}{A_{ik}^{l+1}(q^{-1})} z_i^l(t) \tag{2.12}$$

where

$$A_{ik}^{l+1}(q^{-1}) = 1 - \sum_{j=1}^{N} a_{ikj}^{l+1}(q^{-j}) \tag{2.13}$$

denotes the local feedback in each synapse. All the other notation is the same as for the FIR case.

3 Derivation of a Training Algorithm for an FIR MLP

Let the instantaneous error be

$$E(t) = \frac{1}{2}\sum_{k=1}^{N_L} e_k^2(t) = \frac{1}{2}\sum_{k=1}^{N_L}\left[y_k(t) - z_k^L(t)\right]^2 \tag{3.1}$$

where $y_k(t)$ is the desired output at time t.

The weight changes can be adjusted using a simple gradient method

$$\Delta b_{ikj}^{l}(t) = -\eta \frac{\partial E(t)}{\partial b_{ikj}^{l}(t)} \tag{3.2}$$

$$b_{ikj}^{l}(t+1) = b_{ikj}^{l}(t) + \Delta b_{ikj}^{l}(t) \tag{3.3}$$

The learning rule for the output layer weights is

$$\Delta b_{ikj}^{L}(t) = -\eta \frac{\partial E(t)}{\partial b_{ikj}^{L}(t)} \tag{3.4}$$

$$= \eta f'\left[\hat{x}_k^L(t)\right] z_i^{L-1}(t-j) e_k(t) \qquad 1 \le j \le M \tag{3.5}$$

The learning rule for the hidden layers can be obtained using a chain rule as

$$\Delta b_{ikj(t)}^{l} = -\eta \frac{\partial E(t)}{\partial b_{ikj}^{l}(t)} \tag{3.6}$$

$$= \eta \delta_k^l z_i^l(t-j) \tag{3.7}$$

where

$$\delta^L k(t) = e_{k(t)} f'\left[\hat{x}_{k(t)}^L\right] \tag{3.8}$$

$$\delta_k^l(t) = \sum_{p=1}^{N_{l+1}} \delta_p^{l+1}(t) b_{kp0}^{l+1} f'\left[\hat{x}_k^L(t)\right] \tag{3.9}$$

These equations define an LMS weight adjusting algorithm. It is quite easy to modify the gradient learning rule to incorporate a momentum term.

Notice that while the FIR MLP model is nonlinear, the weight updating rules are linear in the unknown parameters. This property implies that the weight updating rules will converge to a minimum, not necessarily the global minimum, in the mean square error surface of the weight space.

The derived updating rules for the FIR case are not new, but are given for completeness, and serve as a background for the derivation of the IIR synapse to be considered in the next section.

4 Derivation of a Training Algorithm for an IIR MLP

A training algorithm for an MLP consisting of IIR synapses can be obtained by minimizing the cost criterion 3.1.

For the output layer, we have

$$\Delta b_{ikj}^{L}(t) = -\eta \frac{\partial E(t)}{\partial b_{ikj}^{L}(t)} \tag{4.1}$$

$$= \eta e_k(t) f'\left[\hat{x}_k^L(t)\right] \frac{z_i^{L-1}(t-j)}{A_{ik}^{L}(q^{-1})} \qquad 1 \le j \le M \tag{4.2}$$

For the $\Delta a_{ikj}^{L}(t)$ parameter,

$$\Delta a_{ikj}^{L}(t) = -\eta \frac{\partial E(t)}{\partial a_{ikj}^{L}(t)} \tag{4.3}$$

$$= -\eta \frac{\partial E(t)}{\partial \hat{x}_k^L(t)} \frac{\partial \hat{x}_k^L(t)}{\partial a_{ikj}^{L}(t)} \tag{4.4}$$

$$= \eta e_k(t) f'\left[\hat{x}_k^L(t)\right] \frac{B_{ik}^{L}(q^{-1})|_{t-j}}{A_{ik}^{L}(q^{-1}) A_{ik}^{L}(q^{-1})|_{t-j}} z_i^{L-1}(t-j) \tag{4.5}$$

where

$$B_{ik}^{l}(q^{-1})|_{t-p} \equiv \sum_{j=0}^{M} b_{ikj}^{l}(q^{-j-p}) \tag{4.6}$$

$$A_{ik}^{l}(q^{-1})|_{t-p} \equiv q^{-p} - \sum_{j=1}^{N} a_{ikj}^{l}(q^{-j-p}) \tag{4.7}$$

For the hidden layers we have

$$\Delta b_{ikj(t)}^{l} = -\eta \frac{\partial E(t)}{\partial b_{ikj}^{l}(t)} \tag{4.8}$$

$$= \eta \delta_k^l \frac{z_i^{l-1}(t-j)}{A_{ik}^{l}(q^{-1})} \tag{4.9}$$

where

$$\delta_k^l(t) = \sum_{p=1}^{N_{l+1}} \delta_p^{l+1}(t) \frac{b_{kp0}^{l+1}}{A_{kp}^{l+1}(q^{-1})} f'\left[\hat{x}_k^l(t)\right] \tag{4.10}$$

The updating rule for $a_{ikj}^l(t)$ is given by

$$\Delta a_{ikj}^l(t) = -\eta \frac{\partial E(t)}{\partial a_{ikj}^l(t)} \tag{4.11}$$

$$= \eta \delta_k^l(t) \frac{B_{ik}^l(q^{-1})|_{t-j}}{A_{ik}^l(q^{-1}) A_{ik}^l(q^{-1})|_{t-j}} z_i^{l-1}(t-j) \tag{4.12}$$

where δ_k^l is defined in 4.10.

Equations 4.2–4.12 form the complete set of updating rules for the IIR MLP. It is quite simple to incorporate a momentum term in the gradient update rules. Note that in contradistinction with the FIR MLP case, 3.7–4.8 are nonlinear in the parameters. Hence, there is no guarantee that the training algorithm will converge. Indeed, from our own experience, for unsuitably large chosen gain η, the algorithm may explode. The problem of instability that is normally present with linear IIR filters does not arise in the same way with the model presented here. The maximum output from each neuron is limited by the sigmoidal function, thus giving a bounded output (the weights should also be bounded). The usual stability monitoring devices such as pole reflection and weight freezing used in the linear case are therefore not necessary for this model.

5 Simulations

We tested the performance of the FIR MLP and IIR MLP on the following plant

$$y(t) = \sin\left\{\pi\left[\frac{\beta_1(q^{-1})}{1-\alpha_1(q^{-1})-\alpha_2(q^{-2})}x(t)\right]\right\} \tag{5.1}$$

where $x(t)$ is a zero mean white noise source, low-pass filtered with a cut-off frequency of 7 rad/sec, with $\alpha_1 = 0.8227$, $\alpha_2 = -0.9025$, and $\beta_1 = 0.99$. These parameters are chosen to highlight the dynamics of the system and its nonlinearity.

For the FIR MLP, we have chosen $L = 2$, $N_l = 6$, $(l = 1)$, $N_L = 1$. At the hidden layer we selected $M = 11$ and $\eta = 0.0001$, and for the output layer $M = 1$ and $\eta = 0.005$. Zero bias was used throughout.

The simulation was run for 5×10^6 data points. After training we tested the learned weights on a new data set of 1000 points. The mean square prediction error for the test set was 0.0664 and the variance was 0.0082. The results of the simulation are shown in Figures 4 and 5. It is

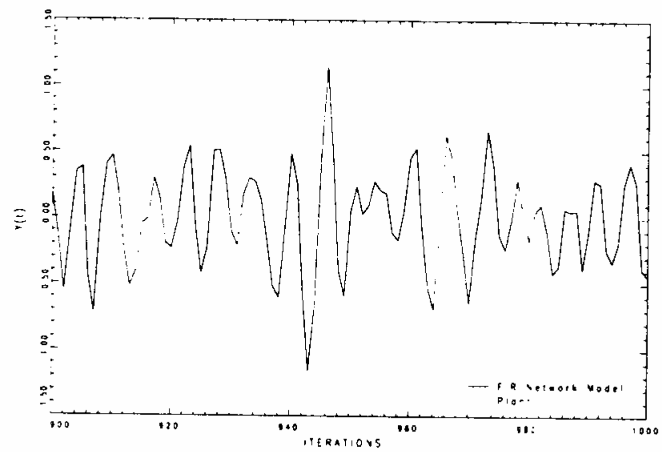

Figure 4: The plant output, and the response from an FIR MLP with architecture described in the text.

observed that the plant and model output, while following one another, appear to have significant differences at points; this is indicated more clearly in Figure 5.

We have also used the IIR network to model the plant given by 5.1. The architecture of the model is the same as for the FIR case, except that $M = 5$ and $N = 6$ in the hidden layer, and in the output layer $N = 1$ with $a^l_{ikj} = 0 \ \forall\ i, k, j$ $(l = 2)$. In this case the mean square error over the test set was 0.0038 and the variance was 0.000012. The results of the simulation are shown in Figures 6 and 7. In Figure 6, it is observed that the response of the IIR MLP is much closer to the plant. This is revealed in the error plot of Figure 7.

6 Conclusions

We have investigated a class of neural networks which has a globally feedforward architecture with locally recurrent nature. A training algorithm has been derived which can be seen as an extension of the FIR MLP and the more widely used static MLP. It is shown, by simulation, that the IIR MLP is a better model than the FIR MLP for modeling a nonlinear plant.

It is almost trivial to modify an algorithm to a recursive second-order gradient algorithm (Kalman type filter) used in traditional adaptive

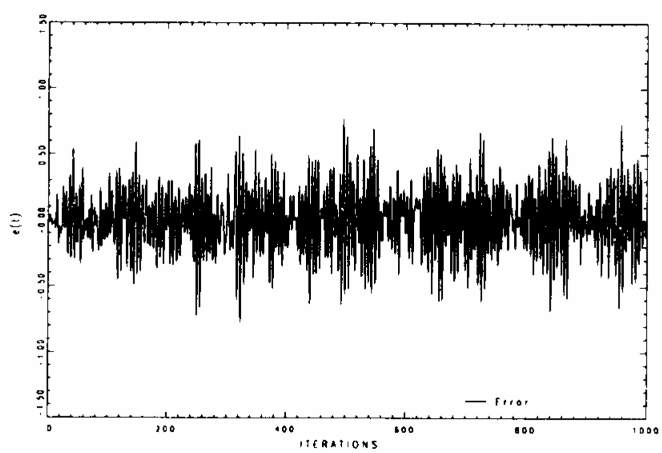

Figure 5: The error between the plant output, and the response of the FIR MLP.

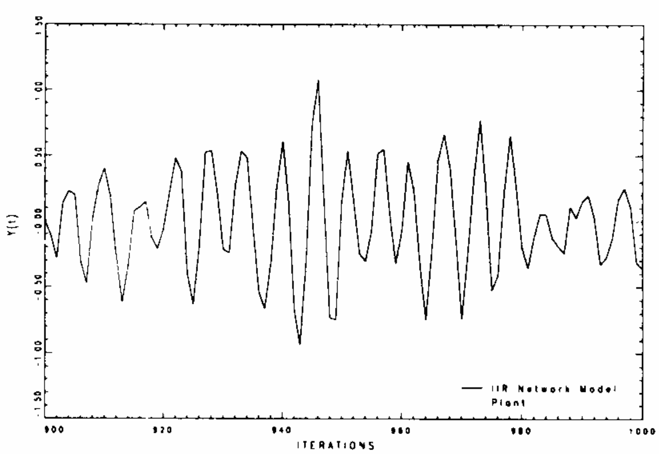

Figure 6: The plant output, and the response from an IIR MLP with architecture described in the text.

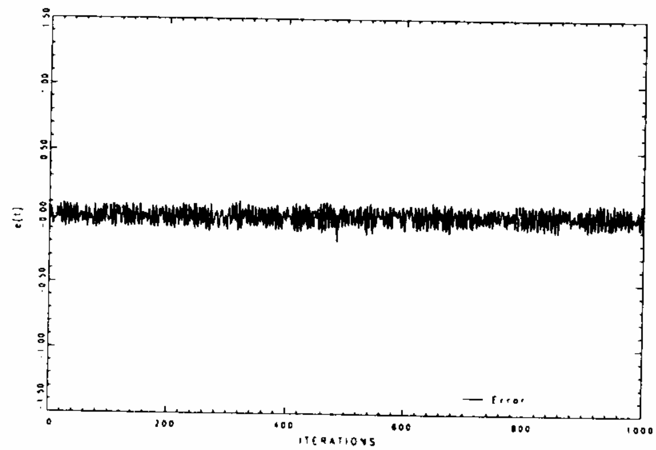

Figure 7: The error between the plant output, and the response of the IIR MLP.

identification or control literature. As indicated, our algorithm is a recursive first-order gradient algorithm. While there is a certain advantage to use a Kalman filter type second-order gradient algorithm, the added computational complexity slows the computation considerably. Hence, in the work reported here we only show the performance of the first-order gradient method.

It would be interesting to compare the performance of the IIR MLP model with a fully recurrent model. This will be presented in future work.

Acknowledgments

The first author is supported by a Research Fellowship with the Electronics Research Laboratory, DSTO, Australia.

References

Jordan, M. I. 1988. *Supervised learning and systems with excess degrees of freedom.* COINS Tech. Rep. 88-27, MIT.

Lapedes, A., and Farber, R. 1987. *Nonlinear signal processing using neural networks: Prediction and system modelling.* Tech. Rep. LA-UR87-2662, Los Alamos National Laboratory.

Lippmann, R. P. 1987. An introduction to computing with neural networks. *IEEE ASSP Mag.* **4**, 4–22.
Robinson, A. J. 1989. Dynamic error propagation networks. Ph.D. dissertation, Cambridge University Engineering Department.
Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. 1989. Phoneme recognition using time-delay neural networks. *IEEE ASSP* **37**, 328–339.

Received 3 December 1990; accepted 28 January 1991.