Preliminary Results on Adaptively Trained Neural Networks

C. E. Pedreira and N. M. Roehl
Catholic University, PUC-Rio, DEE
C.P. 38063, 22453-900 Rio de Janeiro Brazil

Real world applications often involve time varying models. There is an intrinsic difficulty in dealing with this sort of models, specially when one is concerned with nonlinear systems. Layered neural networks have been successfully used in a variety of relevant problems when invariance assumptions can be properly made. On the other hand, very little can be found in the literature concerning time varying systems. We propose a new procedure to adjust weights in Neural Networks suitable for this type of models with no necessity of retraining. One of the main features of our approach concerns the designer flexibility to control a trade off problem between fitting new incoming data and causing minimum damage to the information related to the original data set. We desire to keep the error related to the late incoming data within a pre established tolerance, while maximizing the information incorporated by original training data. In this way, the designer is able to judge the relevance he or she wants to attribute to the latest data. Given a trained network and a new data pair, our goal is to determine a new weight set such that the following associated energy function is minimized, subject to the new data constraint.

$$\text{Minimize } E = \frac{1}{2} \sum_{i=1}^N \|d_i - y_i\|^2 \quad (1)$$

(P_ε)

$$\text{subject to } -\epsilon^1 \leq (d_{N+1}^1 - y_{N+1}^1) \leq \epsilon^1, i=1...0 \quad (2)$$

where $\epsilon^1 \in (0, \text{Min}(d_{N+1}^1, (1 - d_{N+1}^1)))$, $i = 1...0$ is the associated tolerance, y and d are the output and the target, respectively. Problem P_ε represents a trade off between the old system information and the new data, that is presumably reflecting a variance in the original plant. In order to improve analytic and implementation tractability, we linearize constraint (2), and define the feasible region as the intersection between a linearity constraint and the hyperspace defined by (2). If parameter ϵ is set null this constraint becomes identical to the one imposed in [1]. In this way, our problem is in fact a generalization of the one proposed in [1], where a rigid constraint is used in order to oblige the network output to exactly match the latest income data. By using a flexible constraint, the designer should choose the larger ϵ the less slow the process is. This seems to be quite reasonable since for rapid varying processes one does not expect an exact forecast for the last income data. Nonlinear programming techniques are used in order to properly calculate the new weight set. We can prove that the proposed algorithm does not drift. Finally, we point out, that our approach allows the designer to choose how much he or she is prepared to pay in order to match the new data. The smaller the chosen parameter ϵ , the better the last data will be matched, but more damage will be imposed to the previously trained network.

REFERENCES:

- [1] Park D.C., El-Sharkawi M.A. and Marks II R.J., An Adaptively Trained Neural Network, IEEE Trans. N.Nets, Vol 2, N.3, May 1991.