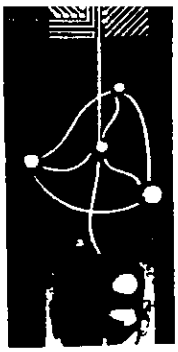

1° Congresso Brasileiro de Redes Neurais

Escola Federal de Engenharia de Itajuba
Itajuba, 24 a 27 de outubro de 1994

Hybrid Networks: A Selective Committee Architecture

Antonio G. Thomé
Instituto Militar de Engenharia
Departamento de Engenharia Eletrica
Rio de Janeiro - Brasil
e-mail: s3thome@imerj.bitnet

Manoel F. Tenório
Purdue University
School of Electrical Engineering
Indiana - USA
tenorio@ecn.purdue.edu

Abstract. In this report we describe a hybrid network technique to generate a committee architecture for a time series prediction case study. The algorithm, here named Selective Multiple Prediction Network - SMP, consists of three steps: a systematic partition of the input hyperspace, a selective training of many agents and a flexible combining strategy. This algorithm generates potentially uncorrelated agents which may improve the performance of the combination process. The proposed architecture is easily extended to the class of pattern classification problems.

Key words: Committe Architecture, Team Prediction, Hybrid Architecture

1. Introduction

System Identification and Linear Prediction are two very important topics in the field of System Theory, and are widely applied to many diverse areas such as Signal Processing, Control, and Forecasting. System Identification is the process of estimating an unknown structure by the knowledge of only its input / output pairs. Linear Prediction, on the other hand, is the process of estimating a future system response based only on the knowledge of its present and past responses.

The emphasis on creating a predictor relies on the identification of underlying patterns, and on the estimation of the model parameters. Historical data analysis and pattern consistency are respectively the major resource and the major underlying assumption for the estimation of such parameters. It may be intuitive the understanding that more complex the problem the more difficult is the underlying patterns identification, and even more difficult is the estimation of a single model that satisfactorily covers the entire problem.

The main idea here is that a large class of prediction problems can be better solved by decomposing the original problem into several subproblems and then combining the multiple sub-solutions, something like divide-and-conquer. This approach generally leads to simpler individual networks and also to a higher accuracy than solving the problem with a single and global predictor. Our committee approach has a simple architecture (fig. 1) composed of three distinct modules: the selector, that performs a pattern classification; the predictor, that is a set of simple networks working in parallel; and the combiner, that generates the system output. The SMP training algorithm consists basically of three steps as follows

- decomposition of the original problem into several and ideally disjoint subproblems;
- parallel estimation of the parameters for each model (agent); and
- combination strategies.

The first step, partition of the original problem, relies on the application of unsupervised methods such as K-means and fuzzy locally sensitive [Tho93a] clustering algorithms. For time series problems, for example, it turns out to be necessary a previous transformation to the original time dependent sequence of points generating a set of state-space vectors and then, spatial similarities of these vectors are exploited by the use of an unsupervised clustering procedure.

In the second step, as many agents as the numbers of partitions are trained, in parallel, on the subsets created by the clustering procedure. Linked to each cluster there is a corresponding agent that can be seen as an expert on a particular view of the underlying structure. Uncorrelated agents are expected to result from this training scheme. Each agent provides its own prediction, and the network or committee final prediction is then obtained through the combination of the individual contributions. We propose here three different combining strategies.

2. The Combining Paradigm

In a seminal paper [Bat69], Bates and Granger showed that a simple linear combination of distinct predictions generally outperforms the individual predictions. A stream of papers followed this initial work. Clemen and Winkler [Cle86] and Clemen [Cle89] provide excellent summaries and extensive bibliographic references.

The field has so far been dominated by works in statistical decision theory, with particular emphasis

on optimal linear combination and on Bayesian inference. However, connectionist researchers have recently begun to show a strong interest in the subject in the form of network committees, agent teams, stacked generalization and others [Lit91, Bey93, Wol92, Sch89, Mac93, Zha92].

Combining is theoretically no worse than any of the individual agents, which can be shown as follows:

Let A_α and A_β be two distinct agents working on information sets I_α and I_β, and let f_α^n and f_β^n be their corresponding predictions for time step n.

If the predictions are optimal with respect to their respective information sets they can be written in terms of posterior expected values, i.e.

$$f_\alpha^n = E\{X_n / I_\alpha\}, \quad (1)$$

and

$$f_\beta^n = E\{X_n / I_\beta\}. \quad (2)$$

The optimal prediction, based on all possible information is then known to be

$$f^n = E\{X_n / I_\Gamma\}, \quad I_\Gamma = I_\alpha \cup I_\beta. \quad (3)$$

This complete estimation problem is normally very complex and computationally expensive. A particular subset of $\{I_\Gamma\}$ that can be considered for example, is a linear combination of the individual predictions

$$C^n = \alpha_1 f_\alpha^n + \alpha_2 f_\beta^n . \quad (4)$$

It is expected that α_1 or α_2 should go to zero whenever f_α^n or f_β^n is optimal with respect to the global information set $\{I_\Gamma\}$. If neither one is optimal then α_1 and α_2 are expected to be different from zero. In general, C^n and f^n are not equal, which clearly indicates that the combination will not be optimal too, although a superior result to each of the original predictions is expected.

Although showing potential for performance improvement, combining techniques present some weak points. The combined performance is highly dependent on the estimation error cross-correlation, serial correlation, and bias. The most effective combinations are achieved with no positive cross-correlation between individual model errors. When negative correlation occurs, which is quite rare, the gains can be spectacular. However, with high positive cross-correlation it is often difficult to achieve even a small improvement. Moreover, if an unstable optimization technique is used, the results may be even worse than those of using equal weights or of selecting the apparently best model. Therefore, the keystone for any combining scheme relies on the generation of as less correlated agents as possible.

3. The Selective Multiple Prediction Network

Training agents over distinct subsets of the full training set is not a new idea. Wolpert [Wol92] uses arbitrarily selected partitions to train the first layer of generalizers; Schapire [Sch89] adopts a residual scheme in which every new agent is trained only on those vectors which previous agents have disagreed on. Ersoy's parallel, self-organizing, hierarchical neural networks [Ers89, Hong91], can also be seen as a kind of residual partition where new agents are trained on transformed versions of those samples rejected by previous agents. In SMP a different scheme to partition the input data set and to perform the prediction task is used. First of all a clustering algorithm is adopted to subdivide the original problem into sets of more homogeneous and easier subproblems, which may eventually lead to learning and prediction improvements, and later, many agents are used in parallel to provide the network output.

The Selective Multiple Prediction Network (fig. 1) involves three distinct processing steps and a number of distinct agents working in parallel. These agents can form a hybrid or a homogeneous structure depending on how they differ from one another. In our studies we only considered homogeneous systems in which each agent is a neural back-propagation network.

3.1 Processing Steps

Pattern matching, function approximation, and a combining strategy are the most important components of the selective multiple prediction task. Pattern matching involves feature selection and unsupervised learning; function approximation involves selective supervised learning, where each neural network is trained to become an expert on specific views of the entire environment; and the combining strategy generates the final prediction. Figure 2 shows a block diagram of these steps.

The selection of relevant features is the first and one of the most important steps. In univariate time series problems this selection process can be thought of in terms of defining different embedding dimensions, i.e., the number of past values to be used in the model. Feature selection [The89, Hsu93, Lap86] is generally a very time consuming and complex task. Here we favored the use of a spread ratio measure r_s (eqn 6) to select those possible embeddings leading to a more consistent unsupervised partition of the input space. The model for a time series is generally expressed as

$$Y = f(X) + \varepsilon, \quad (5)$$

where

$X = [x(t)\ x(t-\tau)\ x(t-2\tau)\ \ldots\ x(t-(m-1)\tau)]^T$ is a vector of mx1,
$Y = x(t+T)$ is a scalar value,
τ is the sampling period,
m is the embedding size,
T is the prediction horizon (lead time).

The spread ratio measure is defined as

$$r_s = mean\left(r_\sigma^i r_f^i\right) \qquad (6)$$

where

$$r_\sigma^i = \frac{mean\left(\sigma_i^2\right)}{\sigma^2},$$

measures the data consistency,

σ_i^2 is the outcome variance for the input vectors belonging to cluster i

σ^2 is the outcome variance for the whole training vectors, and

$$r_f^i = \frac{mean\left(d_w^i\right)}{mean\left(d_B^i\right)}, \qquad (7)$$

is the Fisher descriminator term, which measures the ratio within (d_w) x between (d_B) cluster distances

$$d_w^i = \frac{1}{n_i}\sum_{j=1}^{n_i}\left\|X_j - V_i\right\|^2, \quad \left\{\forall j / X_j \in Cluster\ i\right\}$$

$$d_B^i = \frac{1}{c-1}\sum_{j\neq i}\left\|V_j - V_i\right\|^2,$$

where V_i represents the center of mass of the i^{th} cluster.
Once the embedding m is determined, the time series can be rewritten as a collection of input vectors X, also known as state vectors, and their corresponding outcome Y. This performs a transformation from time to spatial domain where the time dependence is respected within each vector but ignored among different vectors; the regression problem can then be viewed as a case of pattern association of pairs of vectors as follows

x_{11}	x_{12}	.....	x_{1n}
x_{21}	x_{22}	.....	x_{2n}
:	:	:	:
:	:	:	:
x_{m1}	x_{m2}	.....	x_{mn}
y_1	y_2	.....	y_n

This transformation is the key to instance-based methods [Aka91, Hsu93] in which the outcome prediction for the current state vector is based on the outcomes of a number of past state vectors found using a look-up search and a kind of k nearest neighbor approach

Here, in the SMP algorithm, a similar transformation is applied to the time series with the objective to subdivide the original problem into a set of more homogeneous subproblems. An embedding size is selected and the original time series is then transformed into a set of state vectors of that chosen dimension. To select the optimal embedding we used the fuzzy locally sensitive clustering, described in [Tho93a], and also a K-means algorithm. The previously mentioned performance criterion (eqn 6) was then applied to identify those embeddings resulting in more consistent and better shaped partitions. The cluster centroids of the best partition were then assumed to be the class representatives for the pattern matching and selective function approximation phases of the SMP algorithm.

Each cluster defines an associated agent that is trained only on those samples that are classified to the corresponding partition. Three strategies for learning and combining the individual predictions to form the committee prediction were evaluated. The first and simplest one, named *winner-take-all*, selects a single agent at every time step to perform the prediction. The selected agent is the one whose corresponding cluster centroid is closest to the current input vector. The second approach, called *full committee*, is at the other extreme, where all agents are taken into consideration. Each agent contributes and is trained on a percentage of the final prediction error. The percentages or weights add up to one and direct correspond to the degree of membership of the current input vector with respect to each cluster. The third approach, called *windowed-committee*, is in between the two others, since it takes into consideration a subset of the available agents. A temporal window is used as a selection criterion to induce time continuity or time similarity as well as spatial similarity. The combination of spatial and temporal similarities has special appeal in time series applications.

4.2 - Learning Procedures

The Quickprop algorithm [Fah88] with adaptive region of nonlinearity, as described in [Tho93b], was used in all experiments. All training data sets were 1500 or more samples long, and the agents (backpropagation networks) were trained in parallel accordingly to each combining strategy. Batch training with a fixed number of epochs upper bounded at 1000, was used. All procedures assume a previous unsupervised partition step where clusters representing the underlying patterns are generated.

case a) Winner-take-all procedure

1 For every state vector in the training set

step 1- classify current input vector with respect to the existing clusters;

step 2- select winning agent (closest one to the current input vector);

step 3- estimate desired outcome; and

step 4- train selected agent through error backpropagation.

2. If desired accuracy is achieved stop else go back to 1.

case b) Full-committee procedure

1. For every state vector in the training set:

step 1- compute degree of membership for current input vector

$$\mu_i = \frac{\exp\left(-\sqrt{(X-V_i)^T(X-V_i)}\right)}{\sum_{i=1}^{c}\exp\left(-\sqrt{(X-V_i)^T(X-V_i)}\right)};$$

step 2- estimate the outcome for all agents in parallel

$$\hat{y}_i = f(X, W_i); \quad i = 1,\ldots,c$$

step 3- generate the committee prediction by combining the individual outcomes

$$\hat{y} = \sum_{i=1}^{c} \mu_i \hat{y}_i;$$

step 4- train each agent by backpropagating its contribution to the overall error

$$e_T = y_d - \hat{y}, \quad and$$

$$e_i = \mu_i e_T, \quad i = 1,\ldots,c.$$

2. If desired accuracy is achieved stop else go back to 1.

case c) windowed-committee procedure

1. For every state vector in the training set:

step 1- classify current input vector

step 2- select winning agent (closest one to the current vector)

step 3- insert the winner vector at the head of the time-window queue (FIFO) and eliminate the oldest entry

$$WD = \left[X_t\, X_{t-1}, \cdots X_{t-k+1}\right],$$

window of size k

where X_t means the winner vector at time instant t.

step 4- estimate the output for each agent belonging to the time-window queue

$$\hat{y}_i = f(WD_i, W_i), \quad i = 1,\ldots,k$$

where WD_i is the i^{th} column vector and W_i is the corresponding set of weight parameters.

step 5- combine individual outcomes

$$\hat{y} = \sum_{i=1}^{k} \lambda_i \hat{y}_i,$$

where

$$\lambda_i = \frac{\beta^i}{\sum_{i=1}^{k}\beta^i}, \quad 0 < \beta \le 1$$

and

$$\sum_{i=1}^{k} \lambda_i = 1, \qquad \text{is the time weight decay}$$

step 6- train each agent by backpropagating its contribution to the overall error

$$e_T = y_d - \hat{y} \quad and \quad e_i = \lambda_i e_T.$$

2. If desired accuracy is achieved stop else go back to 1.

4.3 - SMP Properties and Drawbacks

SMP provides a powerful architecture to deal with complex real world problems. A set of specialized networks are used, rather than a single one which must accommodate all aspects and underlying dynamics of the problem. Specialization, team cooperation, and truly parallel operation are the key issues in SMP. Robustness, complexity and learning effort reduction, and prediction accuracy improvement are the major goals.

Major properties:

- transforms complex problems into a set of more homogeneous and easily treated subproblems;
- uses smaller individual networks which reduces dimensionality problems and improves learning time;
- exploits spatial and temporal similarity of the input vector which is intuitive and appealing for many real world time series applications;
- combines instance based with parametric approaches without the memory and recall time overhead of the former;
- adopts either hybrid or homogeneous structure, with a flexible combining strategy;
- generates potentially distinct agents by training them on different partitions of the training set.

Drawbacks:

- requires large training sets to avoid situations where an agent is trained on a very small number of patterns;
- requires frequent full retraining and input space partitioning if applied to non-stationary time series;
- since each agent has its own distinct training set, which may have different sizes and degrees of complexity, the algorithm may present overfitting problems if the number of training epochs is set equal for all agents and

5. Empirical Results

The Mackey-Glass chaotic time series was chosen for this benchmark due to its common use among connectionist researchers. The purpose behind the use of Mackey-Glass time series was not to show improvements of current estimates that are already at practical limits. Further improvement is of little practical value. Rather, our purpose was to use a chaotic system defined by a continuous orbit, which by nature does not have clearly definable clusters in the state space. If a reasonable prediction can be attained with this thecnique, functions that are clearly decomposable into multiple mappings can therefore, more easily be dealt with.

The Mackey-Glass equation was first proposed as a model of white blood cell production [Mac77] and subsequently popularized in the nonlinear field due to its richness in structure [Far82]. It is a time-delayed differential equation stated as follows:

$$\frac{dx}{dt} = \frac{ax(t-\Delta)}{\left[1+x^c(t-\Delta)\right]} - bx(t) \qquad (7)$$

Which in discrete time domain can be rewritten as:

$$x(t+1) = \frac{ax(t-\Delta)}{\left[1+x^c(t-\Delta)\right]} - (b-1)x(t). \quad (8)$$

In Mackey-Glass benchmarks, it is commonly avoided to draw conclusions based solely on direct numerical comparisons with other published results. This is because of the differences that can arise from the use of different integrators, initial conditions, sampling rate, and transient elimination. In our study, all results are reported in terms of *Nrmse*.

According to Takens, a chaotic time series $x(t)$ can be predicted T time steps in the future by using only m number of equally spaced past samples of the time series itself. The prediction value is then obtained as follows:

$$x(t+T) = F\left\{x(t), x(t-\tau), \ldots, x(t-(m-1)\tau)\right\} \qquad (9)$$

where F, under suitable assumptions, is a nonlinear continuous function. The choice of an embedding scheme for a benchmark means the determination of the three parameters T, m and τ for the time series. In our experiments we adopted the most widely used values, i.e. m= 6, τ= *6 and T= 6 and 85.*

Using the above parameters, many distinct partitions of a training set with 700 samples were evaluated. A K-means clustering algorithm was used for several values of c (number of clusters), and the performance criterion r_s (eqn 6) was evaluated for each resulting partition. The results indicated a systematic partition improvement as the number of clusters increase. Other observation was that the quality of the partition deteriorates as the lead time T goes further in the future. This is because of the chaotic nature of the series.

Winner-take-all, full-committee and windowed-committee schemes were evaluated on different partition sizes for lead times of 6 and 85. Each model (neural network structure) was defined with a single hidden layer (5 units for the T=6 case and 7 units for the T=85 case) and one output unit, hyperbolic tangent with ARON was adopted for all units. Tables 1 and 2 show some of the obtained results.

The architecture of SMP provides the flexibility to customize and individually tune each Agent. Therefore, in this experiment for example, Agents with poor training performance could, in the WTA scheme, be selected for individual retraining and final accuracy may eventually improve.

Table 3 and figure 3 show the committee prediction results for a partition size of 23 clusters. Observe that the prediction provided by the WTA scheme shows very good performance on turning points, with almost no lag, which may be of great interest for some real world applications.

6. Conclusion

The Selective Multiple Prediction Network provides a very flexible and powerful architecture to handle those more complex problems, where a single and global model is very unlikely to exist. Decomposing the original problem into more homogeneous subproblems leads to potentialy uncorrelated and simpler Agents. Less demanding training effort, and customized tuning according to the requirements of each subproblem are some of the characteristics of this approach. This proposed architecture can also be seen as a structure to combine neural networks (prediction module) with more sophisticated schemes of expert systems (selection module)

7. References

[Bat69] Bates, J. M. and Granger, C.W.J., 1969, "The combination of forecasts", Opl Res Q., vol 20, pp. 451-468.

[Bey93] Beyer, U. and Smieja, F., 1993, "Learning from examples, agent teams and the concept of reflection"

[Cle86] Clemen, R. T., 1986, "Linear constraints and the efficiency of combined forecasts", Journal of Forecasting, vol 5, pp. 31-38.

[Cle89] Clemen, R. T., 1989, "Combining forecasts: a review and annotated bibliography", International Journal of Forecasting, vol 5, pp. 559-583.

[Ers89] Ersoy, O. K. and Hong, D., 1989, "Parallel, self-organizing hierarchical neural networks",

Technical Report - TR-EE-89-56, School of Electrical Engineering, Purdue University,IN.
[Fah88] Fahlman, S. E., 1988. "An empirical study of learning speed in back-propagation networks. TR, CMU-CS-88-162.
[Far82] Farmer, D., 1982. "Chaotic attractors of an infinite-dimensional dynamical system", Physica, vol 4D, pp. 366-393.
[Hon91] Hong, D. and Ersoy,-O. K., 1991. "Parallel, self-organizing neural networks", Technical Report - TR-EE-91-13, School of Electrical Engineering, Purdue University,IN
[Hsu93] Hsu, W., 1993. "Nonlinear and self-adapting methods for prediction", Ph.D. Thesis, School of Electrical Engineering, Purdue University, IN.
[Lap87] Lapedes, A. and Farber, R., 1987, "How neural nets work". Proc of IEEE. Denver Conference on Neural Nets.
[Lit91] Littlestone, N. and Warmuth, M.K., 1991, "The weighted majority algorithm", TR UCSC-CRL-91-28. University of California, Santa Cruz, CA.
[Mac77] Mackey, M.C. and Glass, L., 1977, "Oscillation and chaos in physiological control systems", Science, pp. 197-287.
[Mac93] Mackay, D., 1993. "Bayesian non-linear modeling of the energy prediction competition", University of Cambridge, Cambridge, United Kingdon.
[The89] Theilen, C., 1989. Decision Estimate and Classification, John Willey & Sons, NY.
[Tho93a] Thorne, A.G. and Tenorio, M.F., 1993, "A fuzzy locally sensitive method for cluster analysis", Submited to IEEE Transactions on Fuzzy Systems.
[Tho93b]Thorne, A.G. and Tenorio, M.F., 1993, "Accelerated Learning through a Dynamic Adaptation of the Error Surface", Submited to NN magasine.
[Win83] Winkler, R.L. and Makridakis, S., 1983, "The combination of forecasts", Journal of the Royal Statistical Society, Series A, 146, pp. 150-157.
[Wol92] Wolpert, D.H., 1992, "Stacked generalization", Neural Networks, vol 5, pp. 241-259.
[Zha92] Zhang, X., Mesirov, J.P and Waltz, D.L., 1992, "Hybrid system for protein secondary structure prediction", Journal of Molecular Biology.

Committee	Num. Clusters	T=o	T=85
WTA	09	.1962	.4309
WTA	23	.0773	.3403
Windowed	09	.1904	.4233
Windowed	23	.0728	.3194
Full-Committee	23	.1221	.3752

Table 2 - Mackey-Glass Committee Training Performance in Nrmse

Cls	No	Ns	Cls	No	Ns	Cls	No	Ns
01	.1182	.1117	09	.2285	.2742	17	.1551	.2677
02	.5304	.3842	10	.6292	.6807	18	.3656	.3961
03	.2167	.2912	11	.0660	.0712	19	.5488	.6979
04	.7403	.1614	12	.7220	.6555	20	.3548	.4264
05	.1727	.1478	13	.2011	.2296	21	.5299	.4234
06	.3123	.1949	14	.2516	.1834	22	.2752	.2823
07	.3374	.3378	15	.3698	.3582	23	.5284	.3587
08	.2821	.2876	16	.1447	.1548	WTA	.3403	.3694

Table 3 - Mackey-Glass WTA-Committee Training/Prediction for T=85 (Cls - cluster number, No - cluster training Nrmse, Ns - cluster testing Nrmse; partion size of 23 clusters)

Committee	T=o	T=85
WTA	.0825	3092
Windowed	.0835	3053
Full	.1123	

Table 4 - Mackey-Glass Committee Prediction Comparison [illegible]

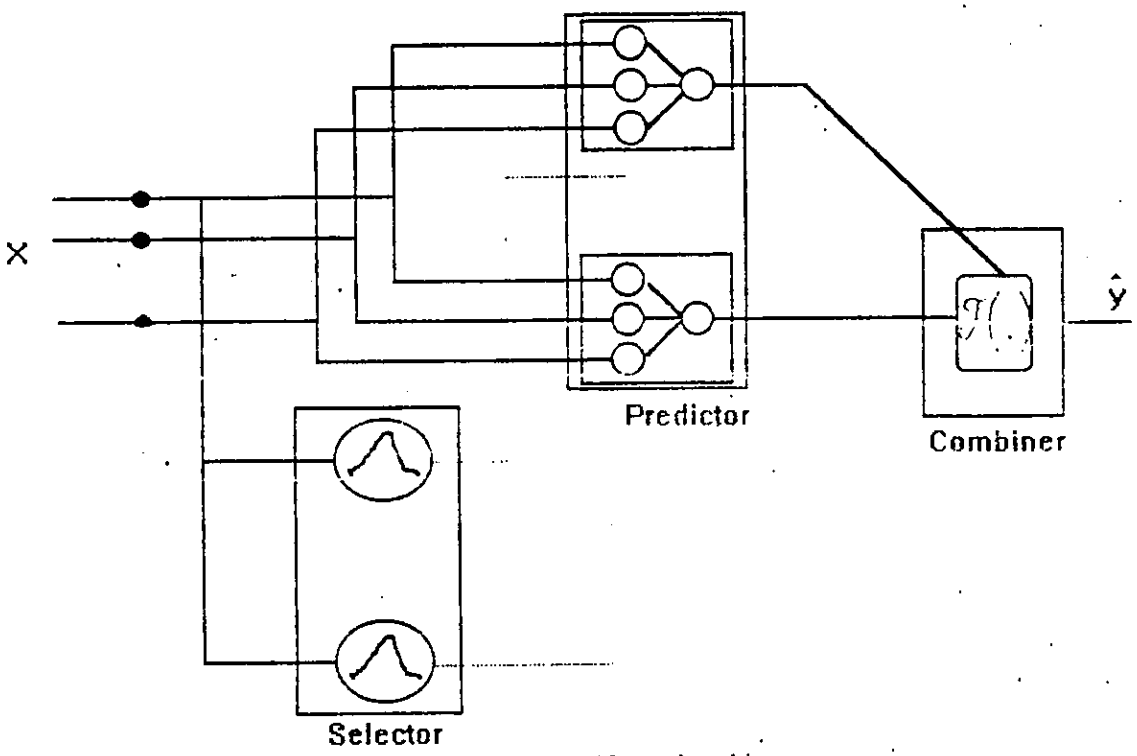

Figure 1 - SMP Network architecture

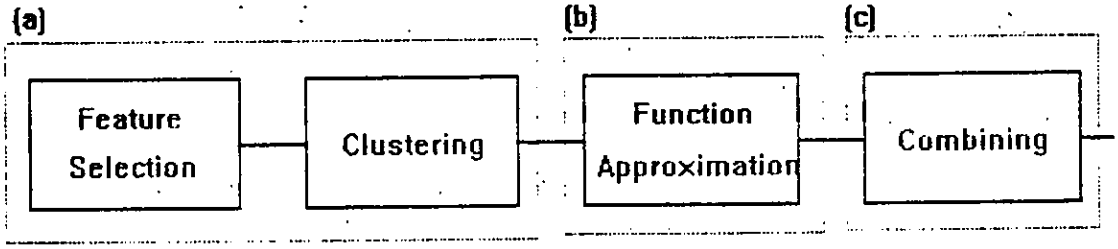

Figure 2 - SMP Processing Steps block diagram. (a) pattern matching, (b) function approximation, (c) combining strategy.

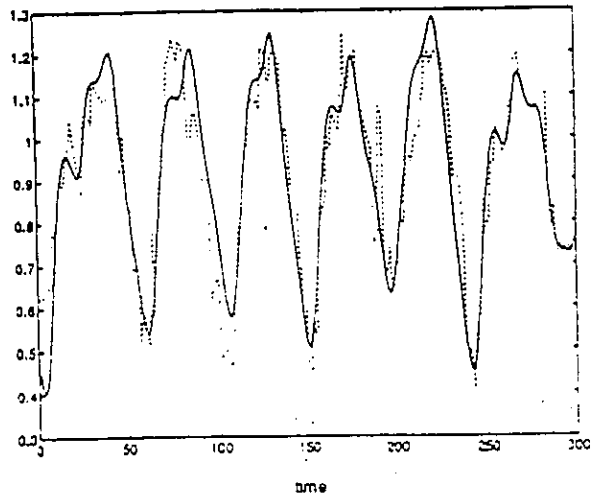

Figure 3 - WTA Committee Prediction on Mackey-Glass Time Series [illegible] with 7 units in the hidden layer and one output unit.