NEURON DYNAMICAL MODELS AND OPTIMAL PREDICTION

Victoria Boghanastjuk, Nicolae Kobileatzky ,Călin Marian, Olesea Berdaga, Alexei Tonu, Alexandru Larchencov, Eugen Alexeienko, Liliana Erhan

Technical University of Moldova, Department of Automation,
Stefan cel Mare av., 168, MD 2004, Chisinau, Republic of Moldova
E-mail: boghan1@yahoo.com

Abstract. This paper describes a new approach to informational technological system the mathematical Hopfield model is derived for the dynamic terminal expansion of the content addressable memory in dynamic systems. The model structure has been specifically designed to facilitate control studies. This a real – time temporal supervised learning algorithm leads to a system.

Keywords: Neuron Dynamical Model, Additive Model, Hopfield models, optimal prediction, dynamic of the network.

INTRODUCTION

The interest system for possess four general characteristics:

1. A large number of degrees of freedom. The human cortex is a highly parallel distributed system that is estimated to possess about 10 billion neurons, with each neuron modeled by one or more state variables. It is generally believed that both the computational power and the fault – tolerant compatibility of such a neuron dynamical system are the result of the collective dynamics of the system.
2. Dissipation. A neuron dynamical system is therefore chracterized by the convergence of the phase – space volume auto a manifold of lower dimensionality as time goes on Noise.

Consider the noiseless dynamical model of a neuron shown [1, 2]. The inputs applied to a output resistance. The main objective of this paper is to supply the solution of the prediction problem for neuron dynamical models with periodic coefficients. The notion of minimum – phase neuron dynamical models is introduced. It is shown that the prediction rule for this model can be given a simple input/output from which generalizes the well known time – invariant prediction formulas [2, 3]. In the no minimum – phase case, the solution of the prediction problem call for a suitable Nation of canonical representation of the cycle stationary process associated with the original neuron dynamical models [3].

The error signal (fig.1.) at the output of neuron j at iteration n is defined by

$$E_j(n)=d_j(n)-y_j(n). \qquad (1)$$

Neuron j is an output node.

NEURON DYNAMICAL MODELS

The instantaneous sum of squared errors of the network is written as

$$e(n) = \frac{1}{2}\sum_{j\in C} e_j^2(n), \tag{2}$$

where the set C includes all the neurons in the output layer of the network. The average squared error is obtained by n summing E(n) over all n and then normalizing with respect to the set size N, as show by

$$e_n = \frac{1}{N}\sum_{n=1}^{N} E(n). \tag{3}$$

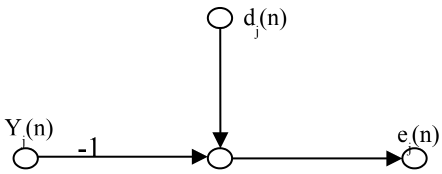

Fig.1.The details of output neuron j

The net interval activity level $v_j(n)$ produced at the input of the nonlinearity associated this neuron j is therefore (fig.2.)

$$v_j(n) = \sum_{i=0}^{p} W_{ji}(n) y_i(n), \tag{4}$$

where p is the total number of input, w_{ij} - the synaptic weight and

$$y_j(n)=\wp_j(v_j(n)). \tag{5}$$

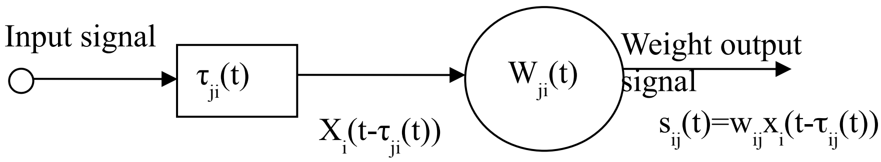

Fig. 2. A dynamic model of a neuron, with each synapse consisting of a time – varying delay followed a time – varying weight

The back-propagation algorithm applies a correlation $\Delta w_{ji}(n)$ to the synaptic Weight $w_{ji}(n)$ which is proportional to the instantaneous gradient $\partial\varepsilon(n)/\partial w_{ji}(n)$. The gradient $\partial\varepsilon(n)/\partial w_{ji}(n)$ represents a sensitivity factor, determining the direction of search in weight space for the synaptic weight w_{ji}. Differentiating (2), (1), (4) with respect t_0 $e_j(n), y_j(n), V_j(n)$ it get

$$\frac{\partial\varepsilon(n)}{\partial e_j(n)} = e_j(n); \qquad \frac{\partial e_j(n)}{\partial y_j(n)} = -1; \quad \frac{\partial y_j(n)}{\partial v_j(n)} = \varphi'(v_j(n)), \tag{6}$$

where the use of prime signifies differentiation with respect to the argument. Finally differentiating (3) with respect to $w_{ji}(n)$ yields

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_j(n); \ \frac{\partial\varepsilon(n)}{\partial w_{ji}(n)} = -e_j(n)\varphi_j'(v_j(n))y_j(n).. \tag{7}$$

The correction $\Delta w_{ji}(n)$ applied to $w_{ji}(n)$ is defined by the delta rule

$$\Delta w_{ji}(n) = -\eta\frac{\partial\varepsilon(n)}{\partial w_{ji}(n)}; \tag{8}$$

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n), \tag{9}$$

where η is a constant that determines the rate of learning; it is called the learning-rate parameter of the back-propagation algorithm, $\delta_j(n) = e_j(n)\varphi_j'(v_j(n))$ is local gradient.

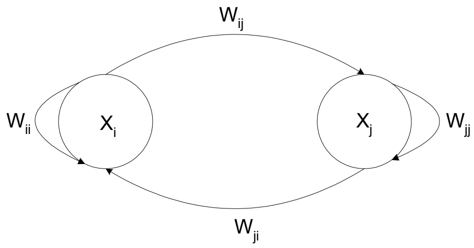

Fig.3. Architectural graph of a two- neuron recurrent network

REAL- TIME RECURRENT NETWORKS

The back propagation through time for training a recurrent network is an extension of the standard back propagation algorithm. It may be derived by unfolding the temporal operation of the network into a multilayer feed forward network, the topology of which grows by one layer at every time a step (fig.3).. Let no denote the start time of an epoch and n_1 denote its end time. Given this epoch, it may define the cost function

$$\varepsilon_{total}(n_0, n_1) = \frac{1}{2}\sum_{n=n_0}^{n_1}\sum_{j\in A}\varepsilon_j^2(n), \tag{10}$$

where A is the set of indices j pertaining to those neurons in the network for which desired responses are specified , and $e_j(n)$ is the error signal at output of such a neuron measured It wish to compute the partial derivates of the cost function $\varepsilon_{total}(n_0, n_1)$ with respect to synaptic weights of the network. The epoch wise back-propagation-through-time algorithm is described as follows:

1. A signal forward passes of the data though the network for the interval $[n_0, n_1]$ is performed. The complete record of input data, network state and desired responses over this interval is saved.
2. A signal backward passes over this past record is performed to compute the values of the local gradients

$$\delta_j(n) = -\frac{\partial \varepsilon_{total}(n_0, n_1)}{\partial v_i(n)}, \tag{11}$$

for all $j\in A$ and $n_0<n\leq n_1$ by using the equations :

$$\delta_j(n) = \begin{cases} \varphi'(v_j(n))e_j(n), & if \quad n = n_1; \\ \varphi'(v_j(n))[e_j(n) + \sum w_{kj}\delta_k(n+1)], & if \quad n_0 < n < n_1, \end{cases} \tag{12}$$

where $\varphi'(.)$ is the derivative of an activation with respect to its argument. The number of steps involved here is equal to the number of time steps contained in the epoch.

3. Once the computation of back propagation has been performed back to time n_0+1 the following adjustment is applied to the synaptic weight w_{ji} of neuron j:

$$\Delta w_{ji} = \frac{-\eta \partial \varepsilon_{total}(n_0, n_1)}{\partial w_{ji}} = \eta \sum_{n=n_0+1}^{n_1} \delta_j(n) \chi(n-1), \tag{13}$$

where η is the learning rate parameter and x_i(n-1) is the i-th input of neuron j at time n-1.

Consider a network consisting of a total of N neuron with M external input connections.

Let W denote the N-by-(M+N) recurrent weight matrix of the network. In order to make provision for a threshold for the operation of each neuron it simply include among the M inputs lines one input whose value is constrained to be always -1 . The net interval activity of neuron j at time n for j$\in$B is given by

$$v_i(n) = \sum_{i \in A \cup B} w_{ji}(n) u_i(n), \tag{14}$$

where A$\cup$B is the union of sets A an B . At the next time step n+1, the output of neurons j is computed bu passing v_j(n) through the nonlinearity φ (i), obtaining

$$y_i(n+1) = \varphi(v_j(n)). \tag{15}$$

The system of equation (14),(15) where the index j ranges over the set B and where u_i(n) is defined in terms of the external inputs and neuron outputs by (14) , constitutes the entire dynamics of the net work .

CONCLUSION

Note in this work time t approaches infinity so as to permit the recursive network to relax to a stable condition. The asymptotic storage capacity of the network has to be maintained small for the fundamental memories to be recoverable. This is indeed a major limitation of the Hopfield network. The temporal back – propagation algorithm to be described by two distinct characteristics: discrete – time operation and fixed time delays. The obtained algorithm a based on the on the real time temporal supervised learning proceeds as follows: for every time step n

$$\Delta W_{ke}(n) = \eta \sum_{j \in Z} e_j(n) \pi_{ke}^{j}(n) \quad \text{and} \quad W_{ke}(n+1) = W_{ke}(n) + \Delta W_{ke}(n). \tag{16}$$

REFERENCES

1. Hopfield I. I. (1990) *The effectiveness of analogue "neural network" hardware. Network 1*, 27-40.
2. Palmieri F.,(1993) *Linear self –association for universal memory and approximation .World Congress on Neural Network* .Vol.2, pp.339-343, Portland,012.
3. Bohanastiuc V.,Coroban Iu., Anestiadi V. (1999) *Modelele Retelelor cu Partajarea deplasarilor in sisteme energetice* . Chisinau -/Preprints of A doua Conferinta Internationala de sisteme electronice . V.III. – p. 293-397.