International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 5, May - 2014

Techniques For Correlation in Neural Networks: A Survey

Manoj. T. R
M.Tech,
CMRIT, Bangalore

Sreedevi
Asst.Professor
CMRIT,Bangalore

***Abstract*: In this paper, we are comparing two techniques for correlation in neural networks: Back Propagation Algorithm and Pair-wise Correlation. The back-propagation algorithm operates in two distinct phases: (1) the forward pass or recall phase and (2) the backward pass or learning phase. But in Pair-wise correlation, hardware design using hierarchical systolic arrays are used. From the investigation, we came to a conclusion that the computational delay is less for pair-wise correlation as compared to Back Propagation Algorithm.**

***Keywords: Systolic Network, Pair-wise Correlation, Back Propagation Algorithm, Neural Network*.**

I. INTRODUCTION

Artificial neural networks (neural nets) have emerged as a promising alternative for solving real world problems such as speech and patter recognition, radar signal tracking, and sonar target detection and bio-medical application. They are able to satisfy the basic requirement of real world problems, i.e., high execution speed. But, for solving any problem, first a neural net has to be trained or the network weights have to be adjusted to correctly classify a set of example patterns, an operation that is highly computation intensive. The correlation map, which represents the correlations between all pairs of recorded units, has become an effective modelling method of biological neural circuits and brain disease biomarker. For example, correlation maps have shown specific deviations in the neural network organizations in Alzheimer's and epilepsy patients. Real-time tracking of underlying neural network properties is important not only for monitoring these nervous system related diseases but also for improving our understanding of their biological bases. Correlation maps facilitate network analysis and monitoring, the computational cost required to construct correlation maps exhibits quadratic growth with the number of input channels. Thus, correlation maps of spike trains recorded by multielectrode arrays (MEAs) are mostly constructed offline. With the rapid advance of MEAs, the drastic increase in the number of channels would further increase the computational cost required to construct the correlation map.A number of systolic algorithms are available for matrix_vector multiplication, the basic computation involved in the operation of a neural net. Using these, many systolic algorithms have been formulated for the implementation of neural nets. Kung et al. have proposed a unified systolic architecture for the implementation of neural net models. It has been shown that the proper ordering of the elements of the weight matrix makes it possible to design a cascaded DG (dependency graph) for consecutive matrix vector multiplication, which requires the directions of data movement at both the input and the output of the DG to be identical. Using this cascaded DG, the computations in both the recall and the learning iterations of a back-propagation algorithm have been mapped onto a ring systolic array. The same mapping strategy has been used in for mapping the hidden Markov model (HMM) and the recursive back-propagation network (RBP) onto the ring systolic array. The main drawback of the above implementations is the presence of spiral (global) communication links. Thus, an important advantage of the systolic architecture, i.e., use of a locally communicative interconnection structure, is lost. By placing side by side the arrays corresponding to adjacent weight layers, both the recall and the learning phases of the back-propagation algorithm can be executed efficiently. But, as the directions of data movement at the output and the input of each array are different this leads to a very non uniform design. Again, a particular layout can only implement neural nets having identical structures. For neural nets that are structurally different, another layout would be necessary. In this paper, we are comparing two techniques for correlation in neural networks: Back Propagation Algorithm and Pairwise Correlation**.**

II. BACK-PROPAGATION ALGORITHM

The back-propagation algorithm operates in two distinct phases: (1) the forward pass or recall phase and (2) the backward pass or learning phase. The recall phase is used to compute the state values of the hidden and output layer neurons. In the learning phase, the error values computed for the output layer neurons are propagated backward to compute the error values of all the hidden layer neurons and to adjust their input weights.

The computations involved in the recall phase can be represented in the matrix form as follows:

$$U^l = W^l A^{l-1} + \theta^l$$
$$A^l = f(U^l).$$

In the above equation, A^l and θ^l are vectors representing the state values and bias inputs of layer l neurons and W^l is a matrix representing the input weights of layer l.

A DG for computing the state values of layer l neurons from the state values of its preceding layer is shown in fig.1.. It can be observed that all the nodes are functionally identical and differ only in the directions of data movement, which depend on the position of a node in the DG. The above DG can be mapped onto a linear systolic array in a straightforward manner. A projection can be taken in the vertical direction and the schedule hyper planes can be chosen in a direction parallel to the horizontal.

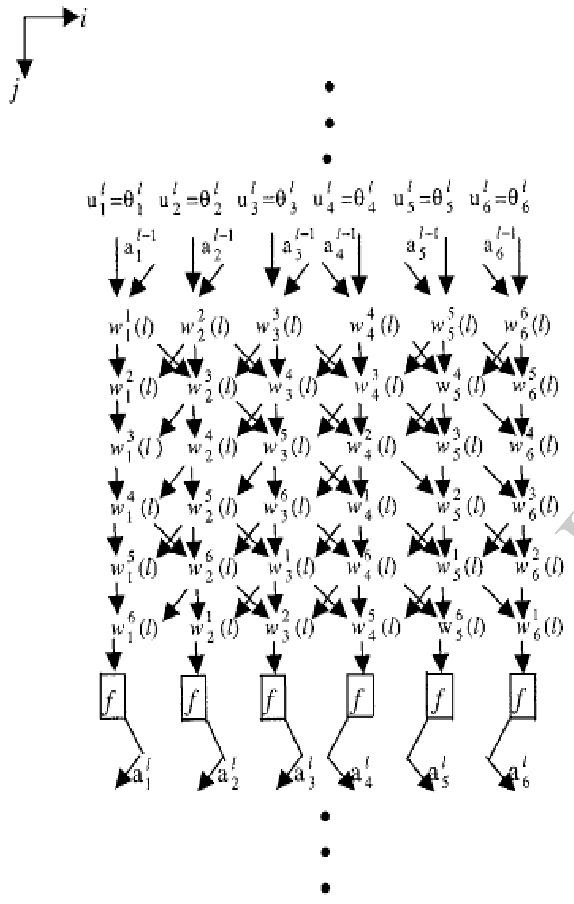

Fig.1. DG for the Recall phase[1]

BP net of six neurons per layer onto a six-processor array. It may be observed that the first and the last links of the first and the last processor, respectively, are shorted. Thus, for i=1, Pii-1=Pi and similarly for i=N, Pi+1=Pi . The operations performed by a processor in the i th iteration are as given in Algorithm 1.

Algorithm 1: In the following algorithm it is assumed that the processors in the linear array are represented as Pf (k) , where f (k)=k, 1≤k≤N≤2, represents the processors P1 , P2 , ..., PN_2, and Pf (k) for f (k)=(N&k+1), 1≤k≤N≤2, represents the processors PN, PN&1, ..., PN_2+1 . Using these notations, the algorithm for computing the state values of layer l neurons from the state values of layer (l&1) is as follows:

1. Each processor P_i initializes a variable $u_i^l=\theta_i^l$.
2. P_i multiplies a_i^{l-1} with $w_i^i(l)$ and adds the product to u_i^l.
3. Afterwards, each processor performs the following operations

(a) In any cycle, processor $P_{f(k)}$ receives data a_j^{l-1} from $P_{f(k+1)}$ and updates: $u_i^l=u_i^l+w_i^j(l)\,a_j^{l-1}$. $P_{f(k)}$ then sends a_j^{l-1} to $P_{f(k-1)}$.

(b) In addition to the above,

(i) for (f_k-1) clock cycles, starting with the second, processor $P_{f(k)}$, $1\leqslant k\leqslant N/2$, receives data a_n^{l-1} from $P_{f(k-1)}$ and forwards it to $P_{f(k+1)}$.

(ii) In the $(N/2+1)$th clock cycle, $P_{f(k)}$ sends the data a_j^{l-1} received from $P_{f(k+1)}$ in the previous cycle back to $P_{f(k+1)}$.

(iii) Again for (f_k-1) cycles, this time starting with the $(N/2+2)$th, processor $P_{f(k)}$, $1\leqslant k\leqslant N/2$, receives data a_n^{l-1} from $P_{f(k-1)}$ and forwards it to $P_{f(k+1)}$.

4. After $(N-1)$ clock cycles, the ith processor would have accumulated the weighted sum $u_i^l=\sum_{j=1}^{N} w_i^j(l)\,a_j^{l-1}$ for the ith neuron. It then computes a_i^l by applying the Sigmoid function to this weighted sum.

This algorithm is executed repeatedly with increasing values of l till the state values of all the output layer neurons have been determined. It is assumed that the state value a_i^l after its evaluation is stored in the processor Pi.
For calculating lower layer $ values, we use the formula

$$\delta^l=\delta^{l+1}W^{l+1}f^l(U^l).$$

The DG for $ commutation is shown in fig.2

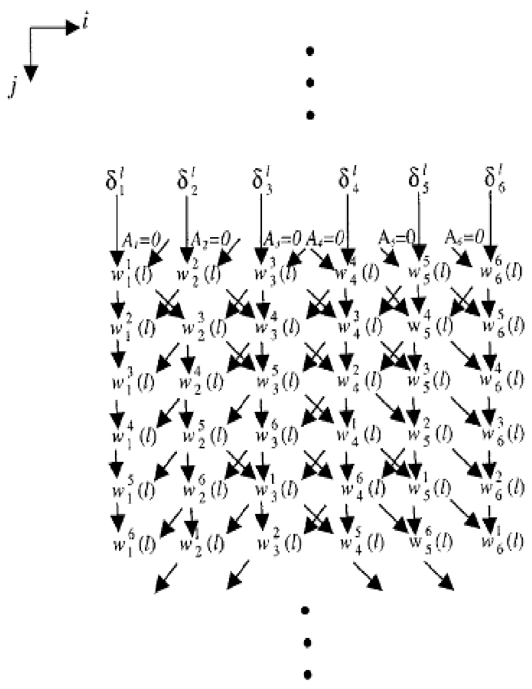

FIG.2. DG for $-value computation.[1]

Algorithm 2 is repeatedly executed with decreasing values of l, i.e., from l=L to l=2, in order to compute the $-values of all the hidden layer neurons.

ALGORITHM 2. 1. Each processor initializes an accumulator $A_i = w_i^j(l)\,\delta_i^l$.

2. P_i also initializes two more accumulators A_i^1 and A_i^N to zeros and sends them respectively to P_{i-1} and P_{i+1}.

3. In each of the following $(N-1)$ cycles,

(i) Processor P_i for $1 \leqslant i \leqslant N/2$ receives the accumulator A_j^1 from P_{i+1} and updates: $A_j^1 = A_j^1 + w_i^j(l)\,\delta_i^l$. P_i then sends A_j^1 to P_{i-1}.

(ii) Processor P_i for $(N/2+1) \leqslant i \leqslant N$ receives the accumulator A_k^N from P_{i-1} and updates: $A_k^N = A_k^N + w_i^k(l)\,\delta_i^l$. P_i then sends A_k^N to P_{i+1}.

4. Additionally, in the $(N/2+1)$th cycle, processor P_i, $2 \leqslant i \leqslant N/2$, initializes a variable $A_j^N = 0$ and sends it to P_{i+1} and processor P_i, $(N/2+1) \leqslant i \leqslant (N-1)$, initializes a variable $A_k^1 = 0$ and sends it to P_{i-1}. In the above, j and k are indices of the accumulators received in the previous cycle from processors P_{i+1} and P_{i-1}, respectively.

After deriving DGs for representing the operations in the different execution phases, steps are outlined to execute the back-propagation algorithm.

The main drawback of Back Propagation algorithm is that as the number of processors increases, the speed also increases

III. PAIRWISE CORRELATION

The correlation between spike trains is a useful measurement for revealing the relationship between neurons. Calculating correlations between all spike trains yields a correlation map where nodes represent neurons or electrodes and edges indicate thedegree of correlation between neural recordings (1 and 0 representcorrelated and uncorrelated relationships, respectively). A cross-correlogram based method is employed to reduce hardware costs and provide effective correlation analysis between spike trains.

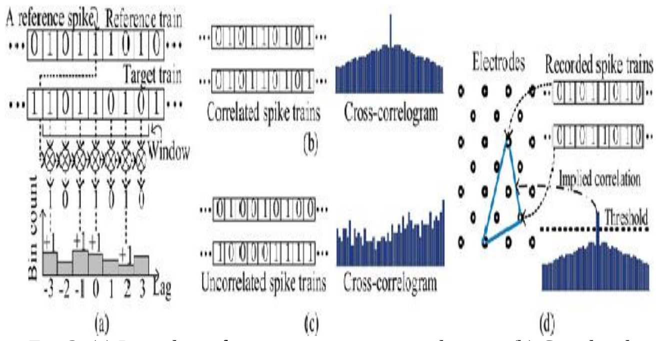

Fig. 3. (a) Procedure of computing a cross-correlogram. (b) Correlated spike trains (left) and the corresponding cross-correlogram (right). (c) Uncorrelated spike trains (left) and the corresponding cross-correlogram (right). (d) Correlation map obtained by calculating all the pair wise cross-correlograms of spike trains recorded by electrodes.[1]

The cross-correlogram is a representation of correlation between two spike trains. Fig. 3(a) summarizes the procedure to generate a cross-correlogram. A target and a reference spike train are aligned and divided equally into a series of bins in which "1" represents a spike.

The systolic array is a specialized form of parallel computing architecture. Identical processing units are organized in a regular network. Each processing unit only communicates with its neighboring units. Pipelines are inserted in communication channels, which make the data flow through the network rhythmically and regularly. The hardware architecture for calculating the crosscorrelogram between spike trains is illustrated in the right panel of Fig. 4. Spike trains, _x and _y, are fed into two delay chains. For the purpose of analysis, delay chains coordinate the spike trains and generate all signal pairs characterized by certain timing lags. One delayed spike signal, *yi* , is broadcasted to each logic AND gate as one input. The other input of each logic AND gate is delayed _x with a particular timing lag to *yi* . Logic AND gates are used to perform binary multiplications. Hardware adders accumulate the results of logic AND gates. Results of adders are stored in registers, "R." The number of pairs of logic AND gates and adders are equal to the window size.

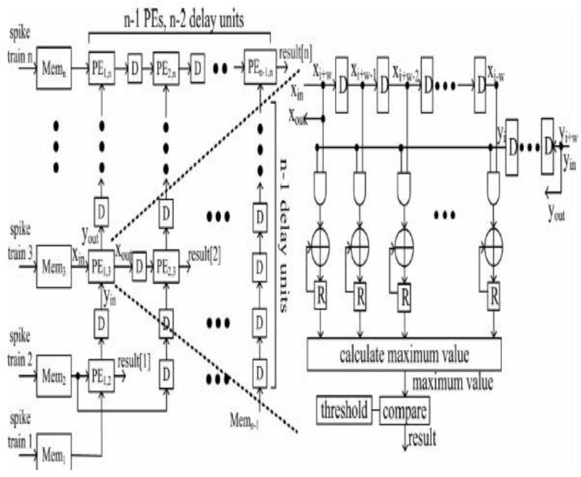

Fig.4. Architecture of the 1-D array for calculating cross-correlograms between spike trains (right), and architecture of the 2-D array for calculating correlation maps (left). In the right panel, "R" represents registers. The latency of the architecture for calculating correlation maps scales linearly with the number of recordings. As the number of recordings increases, the number of PEs of the architecture exhibits quadratic growth.

As the number of spike trains increases, the growth of the computational latency will be quadratic if using single correlation hardware. In this paper, we propose a 2-D systolic array that embeds much identical pair-wise cross-correlogram hardware to speed up the computation.

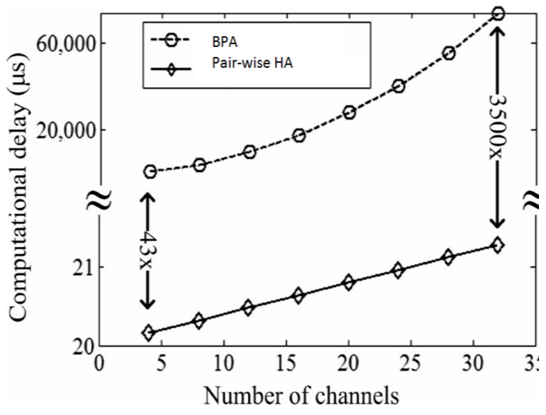

Fig.5. Computational delay comparisons between the proposed hardware architecture and the MATLAB software, which implements the crosscorrelogram- based algorithm on Intel Core I5 650 (at 3.2 GHz).

We compared the pair-wise correlation hardware architecture with a back propagation algorithm in terms of computational delay. The computational delay of the BPA is obtained by measuring the running time of the MATLAB program implementing the crosscorrelogram- based algorithm on Intel Core I5 650. The above graph shows that the computational delay of the BPA exhibits quadratic growth as the number of channels increases. The systolic array outperforms the BPA substantially as the number of channels increases. When the number of channels is 32, the systolic array is almost 3500 times faster than the BPA.

IV.CONCLUSION

In this paper, the pair-wise correlation hardware design utilizing hierarchical systolic arrays is proposed for constructing correlation maps from multiple spike trains. By adopting the largely parallel architecture, the delay of the hardware for constructing correlation maps scales linearly with the number of recordings, whereas the growth of delay is quadratic for a software-based back propagation approach.The computational delay can be reduced by three orders of magnitude when the hardware is adopted. This novel method leads to future devices for real-time monitoring and tracking of large-scale neural networks.

REFERENCES

[1] M. Kaiser, "A tutorial in connectome analysis: Topological and spatial features of brain networks," *NeuroImage*, vol. 57, no. 3, pp. 892–907, 2011.

[2]. H. Yoon, J. N. Hwang, and S. R. Maeng, Parallel simulation of multilayer neural networks on distributed memory multiprocessors, Microprocess. Microprogr. 29 (1990), 185_195.

[3]. A. Singer, Implementation of artificial neural networks on the connection machine, Parallel Comput. 14 (1990), 305_315.

[4] S. Ponten, F. Bartolomei, and C. Stam, "Small-world networks and epilepsy: Graph theoretical analysis of intracerebrally recorded mesial temporal lobe seizures," *Clin. Neurophysiol.*, vol. 118, pp. 918–927, 2007

.[5] A. Jackson, J. Mavoori, and E. Fetz, "Long-term motor cortex plasticity induced by an electronic neural implant," *Nature*, vol. 444, pp. 56–60, 2006.

[6]. S. Y. Kung and J. N. Hwang, A unified systolic architecture for artificial neural networks, J. Parallel Distrib. Comput. 6 (1989), 358_387.