CONVERGENCE ANALYSIS OF A CASCADE ARCHITECTURE NEURAL NETWORK

Tuan A. Duong, Allen R. Stubberud[1], Taher Daud, and Anil Thakoor
Center for Space Microelectronics Technology
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109
[1]Department of Electrical and Computer Engineering, University of California, Irvine
Irvine, CA 92697

Abstract:

In this paper, we present a mathematical foundation, including a convergence analysis, for cascading architecture neural net works. From this, a mathematical foundation for the cascade correlation learning algorithm can also be found, Furthermore, it becomes apparent that the cascade correlation scheme IS a special case of an efficient hardware learning algorithm called Cascade Error Projection. Our analysis also shows that the convergence of the cascade architecture neural network is assured because it satisfies a Liapunov criterion, in an added hidden unit domain rather than in the time domain. Moreover, this analysis also allows us to predict that other methods (such as the conjugate gradient descent and Newton's second order) are good candidates as additional learning techniques. The final choice of a learning technique depends on the constraints of the problems (e.g., speed, performance, and hardware implementation) which may make one technique much more suitable than others. Simulation results help to validate the proposed CEP learning algorithm developed in this paper.

1. Introduction

Many ill-defined problems in areas such as pattern recognition, pattern classification, vision, and speech recognition require practical solutions. Typically, these problems are too complex to be solved by linear techniques thus non-linear methods, such as neural network methods are used. Usually, the practical value of a neural network method is closely related to the paradigm used to train the neural network. Currently, there are several neuromophic learning paradigms reported in the literature [Albus 1971, Cohen et al. 1983, Duong 1995a, Fahlman et aL. 1990, Fukushima et al. 1982, Hinton et al. 1984, Hopfield 1982, Jackson 1988, Kohonen 1989, Kosko 1988, Rosenblatt 1958, Rumelhart, et al. 1986, Widrow 1962] which are wide] y used. The majority of these are supervised learning techniques, the Error Backpropagation (EBP)[Rumelhart, et al. 1986] learning algorithm being one of the most popular. In real world applications, EBP often suffers convergence problems [Fahlman, Lebiere, 1990]. Recently, a technique called "cascade correlation" (CC)[Fahlman, Lebiere. 1990. Hoehfeld, Fahlman 1992] has showed encouraging results as a learning algorithm. This method appears to be fast and reliable, but thus a only empirical studies of its convergence properties have been provided. A mathematical foundation for this algorithm has been needed so that from this a convergence analysis can be developed. Such an analysis is herein provided for a learning algorithm, called cascade error projection (CEP), of which cascade correlation is a special case. CEP is a simple learning method using a one-layer perception approach followed by a deterministic calculation for another layer. This simple procedure offers a very fast, reliable, and implementable learning algorithm in hardware. The architecture for CEP is given in Figure 1.

Shaded squares and circles indicate frozen weights; squares indicate calculated weights, and circles indicate learned weights. The analysis is based only on the set of weights that is connected to the new hidden unit (n+]). in this case, only the blank squares and circles must be determined in order to decrease the energy level.

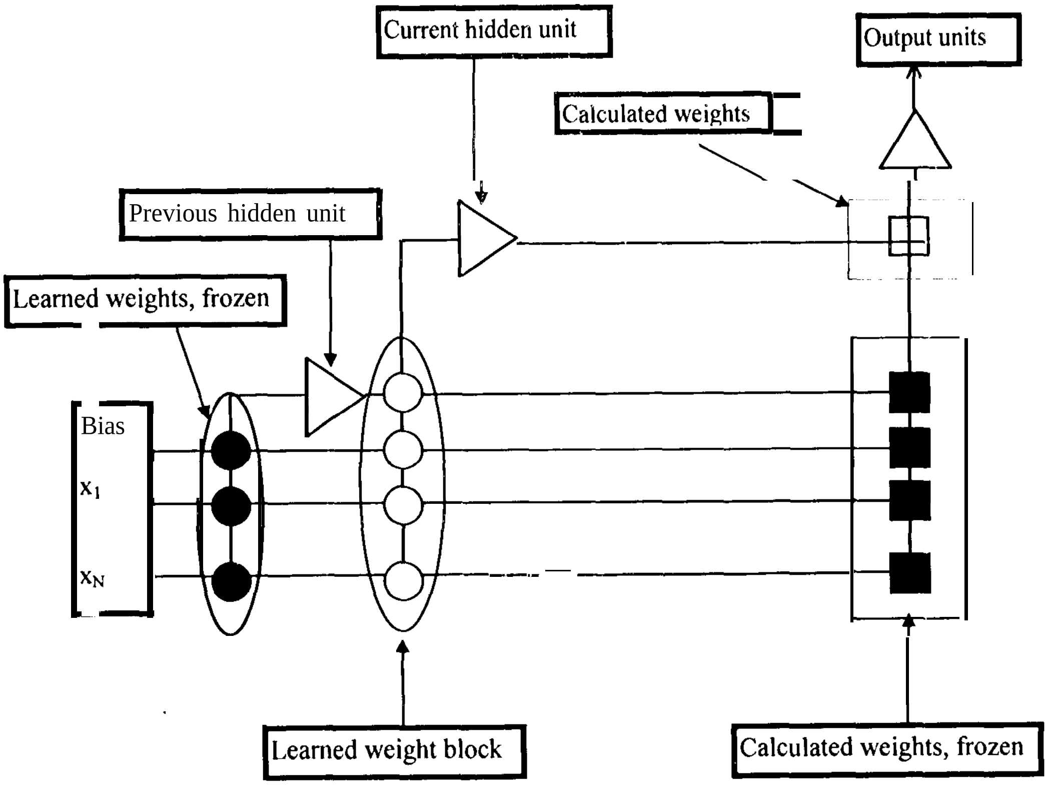

Figure 1. The architecture of cascade error projection includes inputs, hidden units, and output units. The shaded circles or squares indicate the learned or calculated weight set which has already been computed and frozen. A circle indicates that perception learning is used to obtain the weight, and a square indicates that the weight is deterministically calculated.

In this following sections of this paper an analysis of the structure and a learning technique is presented. First, a difference energy function AE between layers *n* and *(n+ 1) is* introduced. This function contains two sets of variables: (1) the set of weights between the input (including previously expanded inputs) and the current hidden unit, namely W_{ih}; (2) the set of weights between the current hidden unit and the output unit, namely W_{ho}. These two sets of variables are treated sequentially (not simultaneously). First, the difference energy function is maximized with respect to W_{ho} thus obtaining $\max_{W_{ho}}(\Delta E)$. Note that, the $\max_{W_{ho}}(\Delta E)$ is also a function of W_{ih}. We will show that there exists a solution set W_{ih}^{*}, obtained from an affine space which guarantees that the network reduces (or at least maintains constant) the present energy level when the new hidden unit is added. Thus, we can conclude that the network converges in the Liapunov sense as new units are added. From this we propose that the solution which is obtained in a non-linear space by learning techniques such as gradient descent, conjugate gradient, correlation, covariance or Newton's second order may also be suitable. The problems that are used to simulate the CEP are 5- to 8-bit parity problems.

II. STRUCTURE OF CASCADE ERROR PROJECTION

We start this section with a definition which will help to define the general structure of our neural network.

Definition:

For any k ∈ N, A^k is the set of all affine functions from $\Re^k$ to ‘N, that is, the set of all functions of the form A(X)= W^TX + b where W and X are vectors in $\Re^k$, and $b \in \Re$ is a scalar.

In this paper, X corresponds to the input of the network and *W* corresponds to the weight set which will vary with the dimension of the required cascade network. We start with the neural network in Figure 2 where we assume that the network contains n hidden units. We also assume that the learning cannot be further improved that is, the energy level cannot be further reduced with this structure. At this point, the new hidden unit *(n+1)* is added to the network and we choose the new weights to further reduce the energy level.

Let Ξ be the input space where $\Xi\subset$ [-1,1]', Ψ' be an output space where $\Psi\subset[-1,1]^m$, and Ω be a hidden output space where $\Omega\subset[-1,1]^q$. Thus, $\Xi x\Omega\subset[-1,1]^{N+q}$ forms the input space of the newly added hidden unit where N ıs the dimension of the input space, g is the dimension of the expanded input space (N+q is the dimension of the total input space to the hidden unit n+1), and m is the dimension of the output space. Let us define

$$f_h:[-1,1]^{N+q}x\Re^{N+q}\ -4[-1,1]$$

$$f_o:[-1,1]^{N+q+1}x\Re^{N+q+1}\longrightarrow\Psi'$$

where $\Re^{N+q}$ is the weight space of N+q dimensional real elements and similarly for $\Re^{N+q+1}$. The functions f_h and f_o are sigmoidal transfer functions which are defined by:

$$f(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$$

Other notation which we will use is defined as follows:

$\varepsilon_o^p = t_o^p$ – o:(n) denotes the error between output element o and training pattern p with target t and actual output $o(n)$ where n indicates that the output has n hidden units in the network;

$f'^{p}_{o}(n)$ denotes the output transfer function derivative with respect to *net. of'* the output element o and the training pattern p;

$f_h^p(n+1)$ denotes the function of hidden unit n+1 and training pattern p;

X^P denotes the input pattern p of vector X.

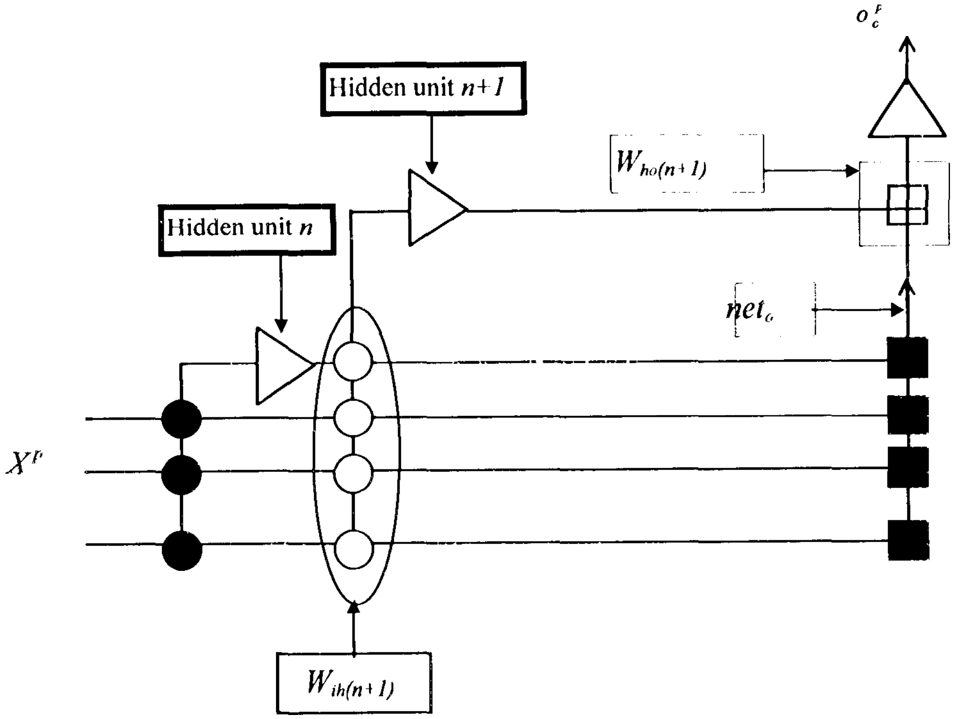

Figure 2: Assume that there are (n+]) hidden units in the network and the blank squares and circles are the weight components which determine the weight values by learning or calculating.

Theorem 1: In the cascade architecture, the maximum energy reduction between hidden unit n and@ *1)* with respect to w_{ho} is

$$\sum_{o=1}^{m} \frac{\{\sum_{p=1}^{P} \varepsilon_o^p(n) f_o'^p(n) f_h^p(n + 1)\}^2}{\sum_{p=1}^{P}\{f_o'^p(n) f_h^p(n + 1)\}} ,$$

where the energy function of the network is defined as

$$E = \sum_{p=1}^{P} E^p = \sum_{p=1}^{P}\sum_{o=1}^{m}(t_o^p - o_o^p)^2 = \sum_{p=1}^{P}\sum_{o=1}^{m}(\varepsilon_o^p)^2$$

Proof

Let t_o^p be the target output of unit o given input pattern p, and let the actual output of unit o be given by:

$$o_o^p = f((X_i^p)^T W_{io} + \sum_{j=1}^{n+1} f_h^p(j) w_{ho}(j))$$

with

$$X_i^p = \begin{bmatrix} 1 \\ x_1^p \\ \cdot \\ \cdot \\ \cdot \\ x_N^p \end{bmatrix} \quad ; \qquad \text{and } X_h^p(n+1) = \begin{bmatrix} f_h^p(1) \\ \cdot \\ \cdot \\ \cdot \\ f_h^p(n+1) \end{bmatrix} \; ;$$

X_i^p (dimension *(N+1)*x*1*) denotes the original input vector of pattern *p*, and $X_h^p\,(n+1)$ (dimension *(n+1)*x*1*) denotes an expanded input vector with *(n+1)* hidden units.

Now let

$$i^p(n+1) = \begin{bmatrix} X_i^p \\ X_h^p(n+1) \end{bmatrix}$$

then

$$f((i^p(j))^T W_{ih}(j)) = f_h^p(j+1)$$

where $f_h^p(j+1)$ denotes the output of hidden unit *j+1* with the input pattern *p*.

Let *E(n)* and *E(n+1)* be the energy levels of the network with *n* and *n+1* hidden units, respectively. The objective in learning is to make the reduction in energy from *E(n)* to *E(n+1)* as large as possible (ignoring the overlearning phenomenon). The ideal case would be

$$\max\{E(n) -- E(n+1)\} = \max \Delta E$$

From Appendix A, we have

$$\mathrm{AE} = \sum_{o=1}^{m}\{-w_{ho}^{2}\sum_{p=1}^{P}[f'^{p}_{o} f_{h}^{p}(n+1)]^{2} + 2w_{ho}\sum_{p=1}^{P}[\varepsilon_{o}^{p} f'^{p}_{o} f_{h}^{p}(n+1)]\} \qquad (1)$$

From equation (1), the maximum AE with respect to w_{ho} is

$$\Delta \mathrm{E}_{\max} \quad \sum_{0\,=,}^{m} \frac{\{\sum_{p=1}^{P} \varepsilon_{o}^{p}(n) f_{o}'^{p}(n) f_{h}^{p}(n+1)\}^{2}}{\sum_{p=1}^{P}\{f_{o}'^{p}(n) f_{h}^{p}(n+1)\}^{2}} \quad \text{where} \quad w_{ho} = \frac{\sum_{p=1}^{P} \varepsilon_{o}^{p}(n) f'^{p}_{o}(n) f_{h}^{p}(n+1)}{\sum_{p=1}^{P}[f'^{p}_{o}(n) f_{h}^{p}(n+1)]^{2}} \qquad (2)$$

Theorem 2: There exists a weight subspace $W_{ho}(n+1)$ of the calculating weight space, from which the energy level is either reduced or remained the same as previous energy level. These cascading sequential subspaces ensure that the network converges in the Liapunov sense.

Proof:

From equation (2), we can rewrite:

$$\forall W_{ih}(n+1) \in \Re^{N+n} \quad n \quad \exists W_{ho} \in \Re^{m}, \Delta \mathrm{E} \geq 0$$

Therefore, the energy reduction is guaranteed or at worst the energy level is remained the same as before.

Theorem 3: The maximum reduction energy with respected to $W_{ih}(n + 1)$ is:

$$\sum_{p=1}^{P}\{\varepsilon_{o}^{p} f_{h}^{p}(n+1)\} \qquad \text{where} \quad f_{h}^{p}(n+1) = \varepsilon_{o}^{p} \qquad (3)$$

Proof:

From equation (2), it is:

$$\Delta \mathrm{E}_{\max} = \sum_{o=1}^{m} \frac{\{\sum_{p=1}^{P} \varepsilon_{o}^{p}(n) f_{o}'^{p}(n) f_{h}^{p}(n+1)\}^{2}}{\sum_{p=1}^{P}\{f_{o}'^{p}(n) f_{h}^{p}(n+1)\}^{2}}$$

let $\Theta = \Delta E_{\max}$ and in order to simplify this proof, we let f_o be a summation from which $f_o' = 1$, and m=1.

The maximum of@ with respected to $f_h(n+1)$ is obtained as follows:

$$\frac{\partial\Theta}{\partial f_h(n+1)} = \frac{2\{\sum_{p=1}^{P} \varepsilon_o^p f_h^p(n+1)\}\{\sum_{p=1}^{P} \varepsilon_o^p\}\{\sum_{p=1}^{P}[f_h^p(n+1)]^2\} - 2\{[\sum_{p=1}^{P} \varepsilon_o^p f_h^p(n+1)]^2 \sum_{p=1}^{P}(f_h^p(n+1)\}}{\{\sum_{p=1}^{P}[f_h^p(n+1)]^2\}^2}$$

A sufficient condition for equation (2) to be maximum with respect to $W_{ih}(n$ +- 1) is

$$\sum_{p=1}^{P}\{\varepsilon_o^p f_h^p(n+1)\} \qquad \text{where } f_h^p(n+1) = \varepsilon_o^p$$

Theorem 4: There exists a solution set of $W_{ih}^{*}(n+1)$ in subspace which is obtained from affine space. This solution is almost always guaranteed to reduce the energy level from the previous energy level.

Proof:

Let

$$\Gamma = \begin{bmatrix} \varepsilon_1 \\ \cdots \\ \cdots \\ \cdots \\ \varepsilon_P \end{bmatrix}$$

Then, $\Gamma \in \Psi$.

and

$$F_h(n+1) = \begin{bmatrix} f_h^1(n+1) \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ f_h^P(n+1) \end{bmatrix}$$

We can rewrite equation (2) in a matrix form as follows:

$$\Delta \mathrm{E} = \frac{\Gamma^T F_h(n+1) F_h^T(n+1)\Gamma}{\Gamma^T \Gamma} \tag{4}$$

Now let

$$F_h(n+1) = \Gamma \tag{5}$$

but

$$F_n(n+1) = F(IW_{ih}(n+1)) \tag{6}$$

with

$$I = \begin{vmatrix} (i'(n))"" \\ \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \\ (i^P(n))^T \end{vmatrix}$$

From (5) and (6), we let ${}^{*}_{ih}$ (n + 1) be a solution in affine space; then we have

$$IW_{ih}\ (\mathrm{n}\ +1)^{=}\,F_h^{-1}(\Gamma)$$

Finally, then the solution is

$$W^{*}_{ih}\ (\mathrm{n}\ +1) = I^{+}F_h^{-1}\ (\mathrm{r}) \tag{7}$$

where I^+ is the pseudo-inverse of I.

In equation (7), the existence of ${}^{*}_{ih}(n$ + 1) depends on the non-zero column matrix $I^{+}F_h^{-1}$ (13. The rank of I is at least 1 because of the non-linear combination of all previous dimensions ($i=1,n$). At the same time, the error surface still exists (if it is zero, then the energy is already zero). Therefore, the existence of ${}^{*}_{ih}(n+1)$ is almost always guaranteed. As shown, the existence in affine space is demonstrated; however, we are also interested in a non-linear space.

Let

$$F_h^*(n+1) = F\{IW_{ih}^*(n+1)\}$$

We should note that, $F_h^*(n+1)$ is a non-zero column matrix. However, the null space in non-linear space may be encountered where 1'7'Fh" (n + 1) = O. Therefore, from (4) the precise inequality is:

$$\text{AE} = \frac{\Gamma^{7} F_h^*(n+1)\, F_h^{*7}(n+1)\Gamma}{\Gamma^{7}\Gamma} \geq 0 \qquad (8)$$

From (8), there exists at least one solution obtained by the pseudo-inverse technique in affine space. This solution also indicates the lower bound of the reduction in energy that can be obtained by the hidden unit (n+ 1). Therefore, in non-linear space it can be shown there always exists a solution space when the error surface is projected to the new hidden unit for learning and the lower bound of energy reduction is $\frac{\Gamma^{7} F_h^*(n+1) F_h^{*7}(n+1)\Gamma}{\text{r}\ {}^{7}\Gamma}$. To obtain the maximum energy reduction, a straight forward approach is to obtain the closest match between $F_h(n+1)$ and Γ. One can use gradient descent [Duong, 95], maximum correlation/covariance [Fahlman, Lebiere. 1990], Newton's second order, or conjugate gradient techniques to obtain this. Finally,

$\Delta\text{E}(n) \geq 0$, with $\Delta\text{E}(n) = \text{E}(n) - \text{E}(n+1)$.

In conclusion, we have shown that there exists a weight set $W^{*}_{ih}(n+1)$, obtained by the pseudo-inverse technique, which guarantees a reduction of the energy or at worst results in the same energy when the hidden unit (n+ 1) is added. From a network viewpoint, since the energy decreases or remains the same when the number of hidden units increases; therefore the network converges (in the Liapunov sense).

III. DISCUSSION:

From this analysis, we will show the relationship to Cascade Correlation learning algorithm, and then propose a new learning algorithm entitled “Cascade Error Projection” which is more suitable for our focus-hardware implementable learning algorithm.

- *Cascade Correlation*:

In equation (2) with $f_o' = 1$ and $m=1$, it ‘becomes:

$$\Delta E = -\frac{\{\sum_{p=1}^{P} \varepsilon_o^p f_h^p(n+1)\}^2}{\sum_{p=1}^{P}\{f_h^p(n+1)\}^2} \quad \text{where} \quad w_{ho} = \frac{\sum_{p=1}^{P} \varepsilon_o^p f_h^p(n+1)}{\sum_{p=1}^{P}[f_h^p(n+1)]^2}$$

In cascade correlation, to maximize the reduction energy (AE) by fine tuning the weight set $W_{ih}(n+1)$, the maximum correlation/covariance between the previous known error surface s:(n) and the additional unknown hidden unit $f_h^p(n+1)$ is used in cascade correlation [Fahlman, Lebiere. 1990]. Then, w_{ho} is obtained through perception learning using the previous weight components from the input/hidden units to the output units. In equation (3), w_{ho} can be viewed as a best weight component in a single dimension from a hidden unit $n+1$ to output units. However the perception learning technique may provide a best weight set in multiple dimensions. From this evident, cascade correlation is one among in powerful software based learning algorithms.

- *Cascade Error Projection:*

Recall equation (3):

$$\sum_{p=1}^{P}\{\varepsilon_o^p f_h^p(n+1)\} \qquad \text{where } f_h^p(n+1)=\varepsilon_o^p \text{ and } w_{ho}=\frac{\sum_{p=1}^{P}\varepsilon_o^p(n)f_h^p(n+1)}{\sum_{p=1}^{P}[f_h^p(n+1)]^2}$$

In order to obtain the maximum reduction of energy (ΔE), we build the objective function (new energy function) Φ with the known target ε_o^p as follows:

$$\Phi(n+1)=\sum_{p=1}^{P}\{f_h^p(n+1)-\varepsilon_o^p\}^2$$

The weights set W_{ih} *(n + 1)* are the variable parameters of function $f_h^p(n+1)$ where Φ(n+1) is to be minimized using a gradient descent technique.

The proposed learning algorithm CEP [Duong, 9s] is more practical to implemented in hardware (low quantization, less learning, simple design, and fast). Also from the theory, it is seen to be feasible to use the conjugate gradient technique, or even better to use Newton's second order approach, to get a better match between $\sum_{o=1}^{m}\varepsilon_o^p(n)$ and $f_h^p(n+1)$ [Battiti, 92]. However, the goal of our present analysis is to select a learning algorithm that best satisfies the given constraints.

The weight set W_{ih} *(n* + 1) can be obtained directly from the affine space by using the pseudo-inverse technique, which has been thoroughly studied [Haykin 1991]. However, in our approach, we are interested in a non-linear solution space in which the solution weight set can be obtained directly from a learning technique using analog/digital hardware. This learning approach offers a better solution from both a theoretical and implementable point of view. First, the solution which is obtained in the non-linear space is always better than a solution in linear space if obtained. Second, it is hard to solve a singular-valued decomposition problem using a linear hardware network, even though the solution is deterministically defined, and the cost of the complicated hardware required by the network may exceed the available resources.

IV. SIMULATION

The problems that are simulated in this paper are 5- to 8-bit parity problems for which (1) there is no limited weight quantization (The weight resolution is the same as the floating point machine which is about 32-bit for floating point or 64-bit for double precision); and, (2) the limited weight quantization is from 3-to 6-bits. The details of the simulation can be found [Duong, 9s, Duong et al. 96]

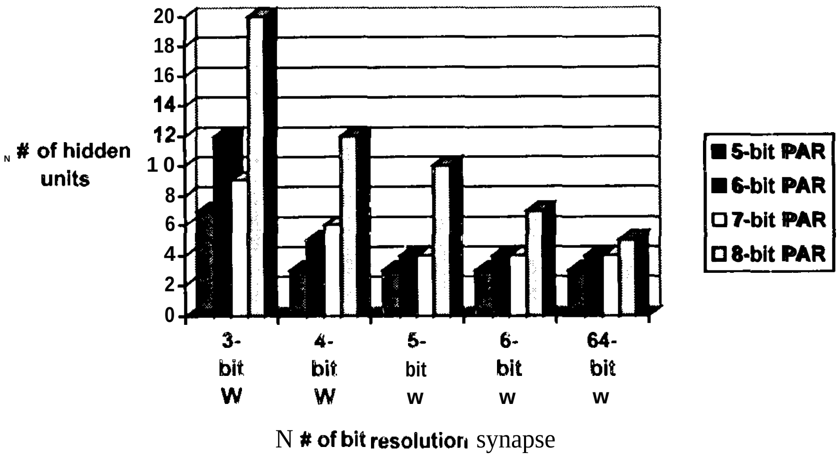

Figure 3: The chart shows CEP learning capability for 5- to 8-bit parity problems using round-off technique. x axis represents limited weight quantization (3-6 and 64-bit) and y axis shows the resulting number of hidden units (limited to 20). Each ‘hidden unit has 100 epoch iterations. As shown, the lager number of hidden units compensate for the lower weight resolution.

V. Conclusions

In this paper, we have shown that CEP is feasible for both a software- and a hardware-based learning algorithm. From this analysis, the way CC works can be understood in depth. Moreover, the theoretical analysis provides us with the general framework of the learning architecture, and the particular learning algorithm can be independently studied for its suitability in a given application associated with some constraint for each problem. (For example, in the hardware approach CEP is most advantageous, and for software, Covariant or Newton's second order method is more advantages).

Acknowledgments:

The research described herein was performed by the Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology and was jointly sponsored by the Ballistic Missile Defense Organization/Innovative Science and Technology Office (BMDO/IST), the Office of Naval Research (ONR), the Advanced Research Projects Agency (ARPA), and the National Aeronautics and Space Administration (NASA).

References:

Albus, J. S. 1971, "A theory of cerebella function," *Mathematical Biosciences vol. 10,* pp. 25-61.

Battiti, R 1992, "First and Second-Order Methods for Learning between Steepest Descent and Newton's Method," Neural computation, Vol. 4(2), pp. 141-166.

Cohen, M. and Grossberg, S. 1983, "Absolute stability of global pattern formation and parallel memmory stage by competitive neural networks, " *IEEE Trans. Systems, Man, Cybernetics, vol.* SMC-13, pp. 815-826.

Duong, T.A et al., 1996a, "Learning in neural networks: VLSI implementation strategies," In: Fuzzy logic and Neural Network Handbook, Ed: C.H. Chen, McGraw-Hill.

Duong, T. A. 1995, "Cascade Error Projection-An efficient hardware learning algorithm," Proceeding Int'l IEEE/ICNN in Perth, Western Australia, vol. 1, pp. 175-178.

Duong, T. A. et al., 1996b, "Cascade Error Projection-A New Learning Algorithm," Proceeding Int'l IEEE/ICNN in Washington D.C., vol. 1, pp. 229-234,

Fahlman, S. E., Lebiere, C. 1990, "The Cascade Correlation learning architecture," in *Advances in Neural information Processing Systems II,* Ed: D. Touretzky, Morgan Kaufmann, San Mateo, CA, pp. 524-532.

Fukushima, K., Miyake, S. 1982, "Neocognitron: A new algorithm for pattern recognition telerant of deformations and shifts in position," *Pattern Recognition,* 15 (6), pp. 455-469.

Haykin, S. 1991 *Adaptive Filter Theory,* Prentice-Hall, Inc. 2nd Ed.

Hinton, G. F., Sejnowski, 'f. J., and Ackley, D.H. 1984, "Boltzmann machines: Constrain satisfaction networks that learn," *CMU Technical Report # CMU-CS-84-119,* Canergie Mellon University, Pittsburgh PA.

Hoehfeld, M. and Fahlman, S. 1992, learning with limited numerical precision using the cascade-correlation algorithm," *IEEE Trans. Neural Networks,* vol.3, *No. 4,* pp 602-611.

Hopfield, J. J. 1982, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. USA,* vol. *79,* pp. 2554-2558.

Jackson. I. R., 1988, "Convergence properties of radial basis functions", Constructive Approximation, vol. 4,243-264.

Kohonen, T. 1989, "Self f-Organizat ion and Associat ive Memory," *Springer-Verlag,* Berlin Heidelberg.

Kosko, B. 1988, "Bidirectional associative memories," *IEEE Trans. Systems, Man and Cybernetics,* vol. 18, N# 1, pp. 49-60.

Rosenblatt, F. 1958, "The perception: A probabilistic model for information storage and organization in the brain," *Psychology Review,* vol. *65,* pp. *386-408.*

Rumelhart, D. E., and McClelland, J. L. 1986, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundation.* MIT Press, Cambridge, MA.

Widrow, B. 1962, "Generalizat ion and information storage in networks of ADALINE neurons," Ed: G.T. Yovitt, "Self-Organizing Systems," *Spartan Books,* Washington DC.

Appendix A:

The energy function of the network is defined as

$$\mathrm{E} = \sum_{p=1}^{P} E^{p} = \sum_{p=1}^{P}\sum_{o=1}^{m}(t_o^p - o_o^p)^2$$

Assume that the network currently has n hidden units, and that the energy no longer decreases with any search technique (gradient descent search, or exhausted search, etc.). A new hidden unit is now **added to the network.** We expect that

$$\mathrm{E}(n+1) \leq \mathrm{E(n)}$$

This is equivalent to

$$\sum_{p=1}^{P}\sum_{o=1}^{m}\{t_o^p - f(net_o^p + w_{ho}f_h^p(n+1))\}^2 \leq \sum_{p=1}^{P}\sum_{o=1}^{m}\{t_o^p - f(net_o^p)\}^2$$

with $\quad o_o^p = f(net_o^p)$

Expanding and rearranging, we have

$$\sum_{p=1}^{P}\sum_{o=1}^{m}\{[f(net_o^p + w_{ho}f_h^p(n+1)) - f(net_o^p)][f(net_o^p + w_{ho}f_h^p(n+1)) + f(net_o^p) - 2t_o^p]\} \leq 0$$

(i)

Assume that $w_{ho}f_h^p(n+1)$ is small so that

$$f\{net_o^p + w_{ho}f_h^p(n+1)\} \approx f(net_o^p) + f'(net_o^p)w_{ho}f_h^p(n+1) \qquad \text{(ii)}$$

From (i) and (ii), it can be shown that

$$\sum_{p=1}^{P}\sum_{o=1}^{m}\{w_{ho}f'^{p}_{o}\,f^{p}_{h}(n+1)[w_{ho}f'^{p}_{o}\,f^{p}_{h}(n+1)-2(t^{p}_{o}-o^{p}_{o})]\}\le 0$$

with $f'(net^{p}_{o}) = f'^{p}_{o}$

or

$$\sum_{o=1}^{m}\{w_{ho}^{2}\sum_{p=1}^{P}[f'^{p}_{o}\,f^{p}_{h}(n+1)]^{2}-2w_{ho}\sum_{p=1}^{P}[f'^{p}_{o}\,f^{p}_{h}(n+1)\varepsilon^{p}_{o}]\}\le 0$$