# Letters

## Training of a Feedforward Multiple-Valued Neural Network by Error Backpropagation With a Multilevel Threshold Function

Vijayan K. Asari

***Abstract*—A technique for the training of multiple-valued neural networks based on backpropagation learning algorithm employing a multilevel threshold function is proposed. The optimum threshold width of the multilevel function and the range of the learning parameter to be chosen for convergence are derived. Trials performed on a benchmark problem demonstrate the convergence of the network within the specified range of parameters.**

***Index Terms*—Backpropagation, multilevel threshold function, multiple-valued neural network.**

### I. Introduction

The binary model of the artificial neurons does not describe the complexity of biological neurons fully since the neurons actually handle continuous data. However, analog neurons implemented in an integrated chip require high-precision resistors and are easily affected by electrical noise. Because of the problems associated with the binary and analog neurons, research on multilevel neural networks for modeling the biological neurons has attracted great attention [1]–[3]. Multiple-valued logic establishes a balance between the quantized integrity of binary and the information density of analog signaling. Multiple-valued logic neuron has robust separation ability and a very fast operation speed in pattern recognition when compared to an ordinary linear neuron. Tang *et al.* introduced multiple-valued algebraic system of learning incorporating a weighted sum and piecewise linear functions [4]. An ARTMAP based multiple-valued neural network for the recognition and prediction of multiple-valued patterns is presented in [5]. A self-organizing neural network for the recognition of multiple-valued patterns is explained in [6].

In this letter, it is shown that a multilayer feedforward multiple-valued neural network can be trained by using a multilevel nonlinear threshold function with backpropagation learning rule. Experiments are performed on various sets of multiple-valued patterns to observe the convergence characteristics of a multilayer quaternary network with different hidden layer nodes.

### II. Training of Multilayer Networks

#### A. Multilevel Threshold Function

A $q$-level continuous nonlinearity can be constructed by the summation of $(q-1)$ number of shifted sigmoids as

$$f_Q(\text{Net}) = \sum_{i=1}^{q-1}\left[\frac{a_i}{1+\exp((-\text{Net}+T_i)/\alpha)}\right] \tag{1}$$

where $a_i$s are positive finite constants and $T_i$s are the transition points [2]. The parameter $\alpha$ determines the slope and it is chosen in the range $0 < \alpha < 1$, where $\alpha$ is called the slope parameter. For balanced $q$-level neurons, the generalized expression for the activation function with $a_i = A\,\forall\, i$, can be written as

$$f_Q(\text{Net}) = \sum_{i=1}^{q-1}\left[\frac{A}{1+\exp((-\text{Net}+T_i)/\alpha)}\right] - \frac{(q-1)A}{2}. \tag{2}$$

If the transition points $T_i$ are at equal intervals, i.e., $T_{i+1} - T_i = D$, where $D$ is the threshold width

$$T_i = iD - \frac{qD}{2} \qquad \text{for } 1 \le i \le q-1. \tag{3}$$

For a quaternary network with $A = 2$, (2) reduces to

$$f_Q(\text{Net}) = \sum_{i=1}^{3}\left[\frac{2}{1+\exp((-\text{Net}+T_i)/\alpha)}\right] - 3 \tag{4}$$

where $T_i = (iD - 2D)$. In order to apply the backpropagation algorithm for the training of the multilayer network, it is necessary that the nonlinear function is differentiable [7].

The derivative of the multilevel threshold function of (2) is given by

$$f'_Q(\text{Net}) = \frac{A}{\alpha}\sum_{i=1}^{q-1}\left[\frac{1}{1+\exp((-\text{Net}+T_i)/\alpha)}\right] \cdot \left[1 - \frac{1}{1+\exp((-\text{Net}+T_i)/\alpha)}\right]. \tag{5}$$

Since the $f'_Q(\text{Net})$ exists, the activation function $f_Q(\text{Net})$ can be used for training a multilevel network with backpropagation learning rule. The convergence of the network depends on the value of the threshold width $D$. If the threshold width is too large, the network requires more number of training cycles for convergence. On the other hand, if the threshold width is too small, a small change in Net can cause the output to change by more than one level and this process may lead to an oscillation in the output. A criterion for choosing an appropriate value of $D$ for the multilevel continuous threshold function is discussed below.

#### B. Choice of the Threshold Width

The change in the Net value of a neuron $j$ in $l$th layer designated as $\Delta\text{Net}_{jl}$ due to one weight update operation is given by

$$\Delta\text{Net}_{jl} = \sum_{i=0}^{N_{l-1}-1} y_{i(l-1)}\Delta w_{jil} \tag{6}$$

where $y_{i(l-1)}$, is the output of the $i$th neuron in the $(l-1)$th layer and $N_{l-1}$, is the number of neurons in the $(l-1)$th layer. The weight update for a neuron in a multilayer network is given by

$$\Delta w_{jil} = \eta\delta_{jl}y_{i(l-1)} \tag{7}$$

where the error signal $\delta_{jl}$ is defined as

$$\delta_{jl} = f'(\text{Net}_{jl})\sum_{k=0}^{N_{l-i}-1}\delta_{k(l+1)}w_{kl(l+1)} \tag{7.a}$$

for a neuron in the hidden layer and

$$\delta_{jl} = f'(\text{Net}_j)(o_j - y_j). \tag{7.b}$$

Manuscript received June 25, 2001; revised July 7, 2001.

The author is with the Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA 23529 USA.

Publisher Item Identifier S 1045-9227(01)08899-3.

for a neuron in the output layer, where $o_j$ represents the desired output. From (6) and (7)

$$\Delta \text{Net}_{jl} = \sum_{i=0}^{N_{l-1}-1} \eta \delta_{jl} y_{i(l-1)}^2 = \eta \delta_{jl} \sum_{i=0}^{N_{l-1}-1} y_{i(l-1)}^2. \tag{8}$$

The maximum change in $\text{Net}_{jl}$, for a given $\eta$ occurs when $y_{i(l-1)} = \pm x_{\max}$ for all $i$ and $\delta_{jl}$ assumes the maximum value. For a $q$-level network using balanced threshold function, $x_{\max} = (q-1)A/2$ and hence

$$(\Delta \text{Net}_{jl})_{\max} = \eta (\delta_{jl})_{\max} x_{\max}^2 N_{l-1}. \tag{9}$$

Assume that the difference between the actual and desired outputs at the $j$th neuron of the output layer $L$ is equal to $mA$. From (7b) we have

$$(\delta_{jL})_{\max} = (f'(\text{Net}_{jL}))_{\max} mA. \tag{10}$$

$(f'(\text{Net}_{jl}))_{\max}$ occurs at the transition points $T_i$ and using (5) it can be shown that

$$(f'(\text{Net}_{jl}))_{\max} = \frac{A}{4\alpha}. \tag{11}$$

The value of the threshold width, $D$ should be chosen such that the resulting $(\Delta \text{Net}_{jl})_{\max}$ does not cause more than $m$ transition points in order to avoid oscillations in the output signal. That is

$$mD \geq \eta (\delta_{jl})_{\max} N_{l-1} x_{\max}^2. \tag{12}$$

From (10) to (12), we get

$$mD \geq \eta \frac{A}{4\alpha} mAN_{l-1} x_{\max}^2 \tag{13}$$

or

$$D \geq \eta N_{l-1} x_{\max}^2 \frac{A^2}{4\alpha}. \tag{14}$$

Equation (14) gives a sufficient condition for the convergence of the $q$-level network. For a balanced quaternary network using the threshold function of (2) with $x_{\max} = 3$ and $A = 2$, (14) reduces to $D \geq (9\eta N_{L-1}/\alpha)$.

### *C. Choice of the Learning Rate Parameter*

Assume that for the balanced $q$-level network using the threshold function of (2), $D$ is selected such that

$$D = kA \tag{15}$$

where $k$ is an integer. The slope parameter $\alpha$ is chosen in the range $0 < \alpha < 1$. Once the appropriate values for $D$ and $\alpha$ are selected, the range of the values of the learning rate parameter $\eta$ can be determined using (14) and (15) as

$$0 < \eta \leq \frac{4k\alpha}{N_{l-1} x_{\max}^2 A} \tag{16}$$

where $x_{\max} = (q-1)A/2$. If $\eta$ is selected according to (16), then the convergence of the network for any set of patterns is ensured. However, a value of $\eta$ greater than the one given by (16) may also lead to convergence depending upon the patterns. For the balanced quaternary network using the threshold function of (4) with $x_{\max} = 3$ and $A = 2$, (16) reduces to

$$0 < \eta \leq \frac{2k\alpha}{9N_{l-1}}. \tag{17}$$

Hence the maximum value of the learning rate parameter for a particular layer can be written as

$$\eta = \frac{\gamma}{N_{l-1}} \tag{18}$$

where $\gamma$ is a real-valued parameter in the range $0 < \gamma \leq (2\alpha k/9)$.

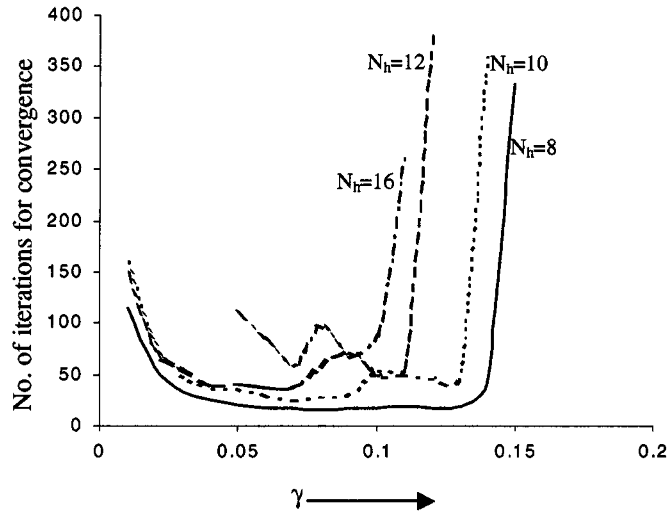


Fig. 1. Effect of $\gamma$ on convergence of multilayer quaternary network with $D = 2$.

## III. Experimental Results

Various experiments are performed on a quaternary network with one hidden layer to evaluate the performance of the proposed technique in a pattern-mapping problem. The network is presented with a sequence of character patterns together with the corresponding desired outputs and it is trained to identify the individual patterns. The quaternary network had eight input and eight output neurons. The gray levels of the patterns are encoded as $-3$, $-1$, 1, and 3. The effect of the learning rate on the convergence characteristics is studied. Fig. 1 depicts the number of iterations required for convergence versus the real valued parameter $\gamma$ for $D = 2$ with $N_h = 8, 10, 12$ and $16$. For this experiment the values of the slope parameter $\alpha$, the momentum term $\varepsilon$, and the number of training patterns $P$ are 0.7, 0.25, and 40, respectively. It is observed that, the network fails to converge for values of $\gamma$ greater than a critical value $\gamma_c$. The parameter $\gamma$ is related to the learning parameter $\eta$ by (18), which in turn related to $D$ by (14). In Fig. 1, the values of $\gamma$ less than 0.1 are found to be more appropriate for the smoother gradient descent learning of the network. However, for smaller values, i.e., $\gamma < 0.03$, the learning is very slow. In another experiment, the plots depicting the root mean square error versus the number of iterations for $D = 2$ with $P = 24$, 32, 40 and 48 are obtained. The values of other parameters for this experiment are $N_h = 10$, $\gamma = 0.07$, $\varepsilon = 0.25$ and $\alpha = 0.7$. Fig. 2 demonstrates the characteristic behavior of the backpropagation learning with gradient descent in the mean square error surface. It can be noted that the network performance improves rapidly during the initial iterations and then undergoes a prolonged phase. This may be attributed to the fact that the magnitude of the weight update $\Delta w$ is more when Net is near the transition points and nonupdates are observed to occur more

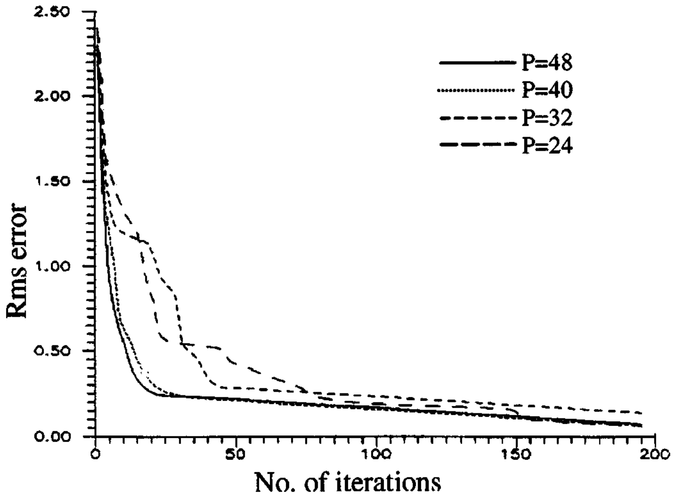


Fig. 2. Convergence characteristics of multilayer quaternary network with $D = 2$.

frequently when the network approaches convergence. The simulation results presented in this section provide preliminary evidence for the convergence properties of a multilevel network using backpropagation algorithm.

## IV. Conclusion

In this letter, a novel technique for the training of multiple-valued neural networks based on backpropagation learning algorithm using a multilevel threshold function has been presented. A suitable multilevel threshold function has been developed. The backpropagation learning algorithm has been used for training the multilevel network. The value of the threshold width of the multilevel function suitable for the convergence of the network has been derived. The range of learning rate parameter to be chosen for ensuring convergence of the network has been computed for a chosen value of threshold width and slope parameter. Experiments have been performed using various sets of quaternary patterns with different values of threshold width, slope, learning parameter, and hidden layer size. These experiments show that the multiple-valued network performs satisfactorily if the learning rate parameter is chosen within the proposed range. An important advantage of the multiple-valued neural network is that it can be implemented in very large scale integration with reduced number of neurons and synaptic weights.

## References

[1] J. Yuh and R. W. Newcomb, "Circuits for multilevel nonlinearities," in *Proc. Int. Joint Conf. Neural Networks*, vol. II, 1992, pp. 27–32.
[2] A. Ngom, I. Stojmenovic, and V. Milutinovic, "STRIP—A strip-based neural-network growth algorithm for learning multiple valued functions," *IEEE Trans. Neural Networks*, vol. 12, pp. 212–227, Mar. 2001.
[3] T. Watanabe, M. Matsumoto, M. Enokida, and T. Hasegawa, "A design of multiple-valued logic neuron," in *Proc. 20th Int. Symp. Multiple-Valued Logic*, 1990, pp. 418–425.
[4] Z. Tang, C. Qi-xin, O. Ishizuka, and H. Matsumoto, "Algorithm and implementation of a learning multiple-valued logic network," in *Proc. 23rd Int. Symp. Multiple-Valued Logic*, 1993, pp. 202–207.
[5] K. V. Asari and C. Eswaran, "A supervised learning neural network for self-organized mapping of multiple-valued patterns," *J. Neural, Parallel, Sci. Comput.*, vol. 7, pp. 441–452, 1999.
[6] ——, "Bidirectional multiple-valued neural network for pattern recognition and associative recall," *J. Imaging Syst. Technol.*, vol. 11, pp. 125–129, 2000.
[7] D. E. Rumelhart, D. E. Hinton, and R. J. Williams, "Learning representations by backpropagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

# Qualitative Analysis of a Recurrent Neural Network for Nonlinear Continuously Differentiable Convex Minimization Over a Nonempty Closed Convex Subset

Xue-Bin Liang

***Abstract*—We investigate the qualitative properties of a recurrent neural network (RNN) for minimizing a nonlinear continuously differentiable and convex objective function over any given nonempty, closed, and convex subset which may be bounded or unbounded, by exploiting some key inequalities in mathematical programming. The global existence and boundedness of the solution of the RNN are proved when the objective function is convex and has a nonempty constrained minimum set. Under the same assumption, the RNN is shown to be globally convergent in the sense that every trajectory of the RNN converges to some equilibrium point of the RNN. If the objective function itself is uniformly convex and its gradient vector is a locally Lipschitz continuous mapping, then the RNN is globally exponentially convergent in the sense that every trajectory of the RNN converges to the unique equilibrium point of the RNN exponentially. These qualitative properties of the RNN render the network model well suitable for solving the convex minimization over any given nonempty, closed, and convex subset, no matter whether the given constrained subset is bounded or not.**

***Index Terms*—Closed convex subsets, convex minimization, global convergence, global existence of solutions, global exponential convergence, recurrent neural networks (RNNs), uniform convexity.**

## I. Introduction

Since the seminal work of Tank and Hopfield [1] and Kennedy and Chua [2], there has been considerable investigation in the literature of artificial neural networks to construct recurrent neural networks (RNNs) for solving linear and nonlinear programming problems (see, e.g., [3] and references therein). It is the parallel and distributive structure inherent in the artificial neural network scheme that renders the neural network, especially the RNN, a preferable approach for solving a numerous variety of problems in mathematical programming and optimization. For the success of the RNN in practice, it is critical for the RNN to have qualitative properties such as the existence of equilibrium of the network, the correspondence between the equilibrium of the network and the solution to the original problem to be solved, the global existence and boundedness of the network's solution trajectory from any initial point in the whole space, and the convergence of the network's solution trajectory to the equilibrium set or some equilibrium point of the network.

In the letter, we consider the following nonlinear minimization problem formulated as

$$\text{minimizing } E(x) \quad \text{subject to } x \in \Omega \tag{1}$$

Manuscript received April 11, 2001; revised August 9, 2001.
The author is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: xbliang@ee.udel.edu).
Publisher Item Identifier S 1045-9227(01)10188-8.