Using Multithreshold Quadratic Sigmoidal Neurons to Improve Classification Capability of Multilayer Perceptrons

Cheng-Chin Chiang and Hsin-Chia Fu

Abstract— **This letter proposes a new type of neurons called multithreshold quadratic sigmoidal neurons to improve the classification capability of multilayer neural networks. In cooperation with single-threshold quadratic sigmoidal neurons, the multithreshold quadratic sigmoidal neurons can be used to improve the classification capability of multilayer neural networks by a factor of four compared to committee machines and by a factor of two compared to the conventional sigmoidal multilayer perceptrons.**

I. Introduction

Recently, many researchers [1], [4]–[6] have studied the recognition capability of feedforward neural networks. In general, the main results obtained in these studies are the derivations on the lower or upper bounds on the number of hidden neurons required to learn the recognition of a given training set S containing fixed number of patterns. For examples, it has been proved [4], [5] that a committee machine containing at most $k-1$ hidden neurons can dichotomize an arbitrary dichotomy defined on any training set with k patterns. Sontag [6] also proved that if the direct input-to-output connections or the continuous sigmoid activation function is used, then a network containing k hidden units can dichotomize an arbitrary dichotomy defined on any training set with at least $2k$ patterns.

In [2], we proposed a new activation function called quadratic sigmoid function (QSF). Here, we refer to a neuron using the quadratic sigmoid activation function as the single-threshold quadratic sigmoidal neuron because there is an extra parameter which we named threshold in each neuron (to be described later). In this letter, an extended type of neurons called multithreshold quadratic sigmoidal neurons are proposed to improve the capability of multilayer neural networks. By using multithreshold quadratic sigmoidal, we will prove that a single-hidden-layer neural network with one multithreshold quadratic sigmoidal output neuron and $k+1$ single-threshold quadratic sigmoidal hidden neurons can dichotomize an arbitrary dichotomy defined on any training set with at least $4k+1$ patterns.

The rest of this letter is organized as follows. In Section II, we formally define the multithreshold quadratic sigmoidal neurons and propose a hybrid single-hidden-layer network architecture composited by multithreshold quadratic sigmoidal neurons and single-threshold quadratic sigmoidal neurons. In Section III, theoretical studies on the classification capability of the proposed network are presented. Finally, Section IV provides some conclusions and suggestions for future work on this research.

Manuscript received October 4, 1992. This work was supported in part by the National Science Council under Grant NSC82-0408-E009-427 and in part by Computer & Communication Research Laboratories ITRI under Grant 37H2200.

C.-C. Chiang is with the Computer & Communication Laboratories, ITRI, Chutung, Hsinchu, Taiwan 310, Republic of China.

H.-C. Fu is with the Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsinchu, Taiwan 300, Republic of China.

IEEE Log Number 9208459.

II. Multithreshold Quadratic Sigmoidal Neurons

In [2], we use single-threshold quadratic sigmoidal neurons for multilayer neural networks and apply it to continuous-valued function approximations. In comparison with conventional sigmoidal multilayer networks, we obtained satisfactory results, such as faster learning, smaller network size, and better generalization capability for our networks. In this letter, we extend the single-threshold quadratic sigmoidal neurons to another type of neurons called multithreshold quadratic sigmoidal neurons to improve the capability. Each multithreshold quadratic sigmoidal neuron, say neuron i, contains n weights $(w_{i,j}, 1 \le j \le n)$, one bias $(w_{i,0})$, and $n+1$ thresholds $(\theta_{i,j})$, where n denotes the input degree of the neuron i. Within each multithreshold quadratic sigmoidal neuron, the extended QSF is used as its activation function. Let vectors $\boldsymbol{w}_i = (w_{i,0}, w_{i,1}, \cdots, w_{i,n})$, $\boldsymbol{\Theta}_i = (\theta_{i,0}, \theta_{i,1}, \cdots, \theta_{i,n})$, and vector $\boldsymbol{x}$ represent the augmented input vector, i.e., $(1, x_0, x_1, \cdots, x_n)$. The extended QSF is then defined as

Extended QSF:

$$f(\mathrm{net}_i, \boldsymbol{\Theta}_i) = \frac{1}{1+\exp(\mathrm{net}_i^2 - g(\boldsymbol{\Theta}_i, \boldsymbol{x}))} \tag{1}$$

where

$$\mathrm{net}_i = \boldsymbol{w}_i \boldsymbol{x} = w_{i,0} + \sum_{j=0}^{n} w_{i,j} x_j, \qquad \text{and}$$

$$g(\boldsymbol{\Theta}_i, \boldsymbol{x}) = \theta_{i,0} + \sum_{j=0}^{n} \theta_{i,j} x_j.$$

In the original QSF for single-threshold quadratic sigmoidal neurons, the function $g(\Theta, \boldsymbol{x})$, is simply taken as $g(\boldsymbol{\Theta}, \boldsymbol{x}) = \theta_i$, i.e.,

$$\text{QSF}: \quad f(\mathrm{net}_i, \theta) = \frac{1}{1+\exp(\mathrm{net}_i^2 - \theta)}. \tag{2}$$

From (1) and (2), we can see that both multithreshold quadratic sigmoidal neurons and single-threshold quadratic sigmoidal neurons contain quadratic terms (net_i^2) in their activation function. Thus, both multithreshold quadratic sigmoidal neurons and single-threshold quadratic sigmoidal neurons can exhibit second-order characteristics as conventional second-order neural networks [3] to some extent. The second-order property is very helpful for a network in solving nonlinearly separable problems such as XOR, and parity problems.

With the multithreshold quadratic sigmoidal neurons, a multilayer neural network with more powerful recognition capability can be constructed. In this paper, we will consider only the single-hidden-layer hybrid network architecture which contains one multithreshold quadratic sigmoidal neuron in the output layer and many single-threshold quadratic sigmoidal neurons in the hidden layer. Throughout this letter, we will assume that the input layer, hidden layer, and output layer are the zeroth, first, and second layer, respectively. Thus, a superscript $(l)(l = 1, 2)$ on any parameter is used to denote the layer number.

III. Classification Capability of the Proposed Hybrid Neural Network Architecture

Before presenting the capability study of the proposed neural network, another activation function called the quadratic Heaviside function has to be introduced first. The quadratic Heaviside function

is an extension of conventional Heaviside function and is defined as

Quadratic Heaviside:

$$\mathcal{H}_q(\boldsymbol{w}_i, \boldsymbol{x}, \theta_i) = \begin{cases} 0 & \text{if } \theta_i - (\boldsymbol{w}_i \boldsymbol{x})^2 < 0 \\ 1 & \text{if } \theta_i - (\boldsymbol{w}_i \boldsymbol{x})^2 \geq 0 \end{cases}. \tag{3}$$

Let $\mathcal{H}(x)$ denote the conventional Heaviside function, i.e., $\mathcal{H}(x) = 0$ for $x < 0$ and $\mathcal{H}(x) = 1$ for $x \geq 0$, then

$$\mathcal{H}_q(\boldsymbol{w}_i, \boldsymbol{x}, \theta_i) = \mathcal{H}(q(\boldsymbol{w}_i, \boldsymbol{x}, \theta_i)) \tag{4}$$

where $q(\boldsymbol{w}_i, \boldsymbol{x}, \theta_i) = \theta_i - (\boldsymbol{w}_i \boldsymbol{x})^2$. Thus, the following lemma can be easily derived.

Lemma 1: For any positive constant k,

$$\mathcal{H}_q(\boldsymbol{w}_i, \boldsymbol{x}, \theta_i) = \mathcal{H}_q(k\boldsymbol{w}_i, \boldsymbol{x}, k^2\theta_i), \qquad \text{for all } \boldsymbol{x} \in \Re^n.$$

Let $\sigma(x)$ be the sigmoidal function, i.e., $\sigma(x) = 1/(1 + \exp(-x))$. We also can prove the following lemma.

Lemma 2: Given an error tolerance $\epsilon > 0$, then

$$|\sigma(x) - \mathcal{H}(x)| \leq \epsilon, \qquad \text{for } |x| \geq \left| \log\left(\frac{1-\epsilon}{\epsilon}\right) \right|.$$

Proof: For $x < 0$, $\sigma(x)$ must be less or equal to ϵ, i.e., $1/(1 + \exp(-x)) \leq \epsilon$. Thus, it is easy to derive that $x \leq -\log((1-\epsilon)/\epsilon)$. On the other hand, for $x \geq 0$, $\sigma(x)$ must be larger than or equal to $1 - \epsilon$, i.e., $1/(1 + \exp(-x)) \geq 1 - \epsilon$. Thus, we derive $x \geq \log((1-\epsilon)/\epsilon)$. Therefore, we conclude that for $|x| \geq |\log((1-\epsilon)/\epsilon)|$, $|\sigma(x) - \mathcal{H}(x)| \leq \epsilon$. □

Similar to the multithreshold quadratic sigmoidal neurons, we can also define the extended quadratic Heaviside function for another type of neurons called multithreshold quadratic heaviside neurons as follows:

Extended Quadratic Heaviside:

$$f(\boldsymbol{w}_i, \boldsymbol{x}, \boldsymbol{\theta}_i) = \begin{cases} 0 & \text{if } g(\boldsymbol{\theta}_i, \boldsymbol{x}) - (\boldsymbol{w}_i \boldsymbol{x})^2 < 0 \\ 1 & \text{if } g(\boldsymbol{\theta}_i, \boldsymbol{x}) - (\boldsymbol{w}_i \boldsymbol{x})^2 \geq 0 \end{cases} \tag{5}$$

where

$$g(\boldsymbol{\theta}_i, \boldsymbol{x}) = \theta_{i,0} + \sum_{j=0}^{n} \theta_{i,j} x_j.$$

In the following, we will start the capability study from nonfeedforward single-hidden-layer networks which contain quadratic Heaviside neurons in hidden layer and one multithreshold Heaviside neuron in the output layer and use direct input-to-output connections. Then, we extend the results to the feedforward networks which contain many single-threshold quadratic sigmoidal neurons in the hidden layer and one multithreshold quadratic sigmoidal neuron in the output layer.

Suppose that a training set S consists of distinct vectors $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_p$, where $\boldsymbol{u}_i \in \Re^n$. Since the set

$$A = \Re^n - \cup_{i \neq j} \{s | s \cdot (\boldsymbol{u}_i - \boldsymbol{u}_j) = 0, s \in \Re^n\}$$

which cannot be empty, we can always find a vector $\boldsymbol{v}$ in A such that the new training set

$$S' = \{y_i | y_i = \boldsymbol{v}\boldsymbol{u}_i, 1 \leq i \leq p\}$$

contains no duplicated elements. Assume that a network containing h neurons in its first hidden layer can dichotomize a dichotomy which is induced from S onto S'. Let the weights of these h hidden neruons be $w_1, w_2, \cdots w_h (w_i \in \Re)$. Then, it is obvious that the network also can dichotomize the original dichotomy on S if we replace the weights of these h hidden neurons by $w_1\boldsymbol{v}, w_2\boldsymbol{v}, \cdots, w_h\boldsymbol{v}$.

Without loss of generality, let us assume that $y_1 < y_2 < \ldots < y_p$, where $y_i \in S'$, and take any dichotomy (S_-, S_+) on the original training set S. This will induce a partition on the set of y_i's into two subsets, corresponding to values $\boldsymbol{v}\boldsymbol{u}_i, u_i \in S_+$ and $\boldsymbol{v}\boldsymbol{u}_i, u_i \in S_-$. We shall assume that y_1 is of the first subset since we always can find a vector $\boldsymbol{v}$ for this purpose. Now, we can prove the following theorem.

Theorem 1: Given a training set

$$S = \{y_1, y_2, \cdots, y_{4k+1} | y_i \in \Re, 1 \leq i \leq 4k+1\}.$$

A single-hidden-layer network with direct input-to-output connections and containing at most k quadratic Heaviside hidden neurons and one multithreshold quadratic Heaviside output neuron can dichotomize an arbitrary dichotomy defined on S.

Proof: Let us use the notation "$I_i < I_j$" for intervals to mean that $x < y$ for all $x \in I_i$, and all $y \in I_j$. Let

$$I_i < I_{i+1}, \qquad \text{for } i \leq i \leq 4k$$

be closed subintervals of $\Re$.

Denote

$$I^+ = \cup_0^{2k} I_{2i+1}, \qquad I^- = \cup_0^{2k} I_{2i}.$$

If we can construct a network with the stated architecture such that the constructed network outputs "1" for $x \in I^+$ and outputs "0" for $x \in I^-$, then the proof can be completed. Assume that

$$\begin{aligned} & \alpha_0 < I_1 < \beta_0 < I_2 < \gamma_0 < I_3 \\ & < \gamma_0' < I_4 < \beta_0' < I_5 < \alpha_1 < \cdots \\ & < \alpha_i < I_{4i+1} < \beta_i < I_{4i+2} < \gamma_i < I_{4i+3} \\ & < \gamma_i' < I_{4i+4} < \beta_i' < I_{4(i+1)+1} < \alpha_{i+1} < \cdots \\ & < \alpha_{k-1} < I_{4(k-1)+1} < \beta_{k-1} < I_{4(k-1)+2} \\ & < \gamma_{k-1} < I_{4(k-1)+3} < \gamma_{k-1}' < I_{4(k-1)+4} \\ & < \beta_{k-1}' < I_{4k+1} < \alpha_k. \end{aligned}$$

Let $\boldsymbol{w}_i = (w_{i,0}, w_{i,1})$ denote the bias and weight of the ith quadratic Heaviside hidden neuron, and $\theta_i^{(1)}$ denote the threshold of the ith quadratic Heaviside hidden neuron. Also let $\boldsymbol{u} = (u_0, u_1, \cdots u_k)$, and $\boldsymbol{\Theta} = (\theta_0^{(2)}, \theta_1^{(2)}, \cdots, \theta_k^{(2)})$ denote the hidden-to-output connection weight vector and the threshold vector of the multithreshold quadratic Heaviside neuron, respectively. Besides, use $v (\in \Re)$ to denote the direct input-to-output connection weight. Thus, the output of this network can be formulated by

$$O = f\left(u_0 + v \cdot x + \sum_{i=1}^{k} u_i \cdot h(w_{i,0} + w_{i,1}x, \theta_i), \boldsymbol{\Theta}\right) \tag{6}$$

where $x \in \Re$ is the input, $f(\,)$ denotes the extended quadratic Heaviside function (see (5)), and $h(\,)$ denotes the quadratic Heaviside function (see (3)). Now, let us set the parameters of this network as

$$\begin{aligned} u_0 &= 0, \qquad u_i = -\frac{\gamma_{i-1} + \gamma_{i-1}'}{2}. \\ v &= 1, \qquad \theta_0^{(2)} = \max\{\alpha_0^2, \alpha_k^2\}. \\ \theta_i^{(2)} &= \left(\frac{\gamma_{i-1}' - \gamma_{i-1}}{2}\right)^2 - \theta_0^{(2)} \qquad \text{for } i \neq 0. \\ w_{i,0} &= -\frac{\beta_{i-1} + \beta_{i-1}'}{2}. \\ w_{i,1} &= 1, \qquad \text{and} \\ \theta_i^{(1)} &= \left(\frac{\beta_{i-1}' - \beta_{i-1}}{2}\right)^2 \qquad \text{for } 1 \leq i \leq k. \end{aligned}$$

Based on these settings, given an input x in the interval $(\beta_{i-1}, \beta_{i-1}')$, according to (3), it is easy to prove that only the ith hidden neuron will output "1" and other hidden neurons output "0." Thus, $g(\boldsymbol{\Theta}, \boldsymbol{x}) = ((\gamma_{i-1}' - \gamma_{i-1})/2)^2$. According to (6) and (5), the output of the network becomes

$$\begin{aligned} O &= f(x + u_i, \boldsymbol{\Theta}) \\ &= \begin{cases} 1 & \text{if } \gamma_{i-1} \leq x \leq \gamma_{i-1}' \\ 0 & \text{if } \beta_{i-1} < x < \gamma_{i-1} \quad \text{or } \gamma_{i-1}' < x < \beta_{i-1}'. \end{cases} \end{aligned}$$

Therefore, if $x \in I_{4i+3}$ then the network outputs "1;" if $x \in I_{4i+2}$ or $x \in I_{4i+4}$, the network outputs "0."

For x in I_{4i+1}, according to (3), we can see that no hidden neuron will output "1." Thus, $g(\boldsymbol{\Theta}, \boldsymbol{x}) = \theta_0^{(2)} = \max\{\alpha_0^2, \alpha_k^2\}$. Then, the output of the network becomes

$$O = f(x, \boldsymbol{\Theta}) = \begin{cases} 1 & \text{if } -\sqrt{\theta_0^{(2)}} \le x \le \sqrt{\theta_0^{(2)}}, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, we can prove that for all x in I_{4i+1} $(1 \le i \le k)$, the network outputs "1." Finally, we can conclude that for $x \in I^+$, the network outputs "1" and for $x \in I^-$, the network outputs "0." □

In the following, we extend the results to the feedforward networks which use single-threshold quadratic sigmoidal hidden neurons and multithreshold quadratic sigmoidal output neuron. Before the extension, it is necessary to introduce the following auxiliary lemmas first.

Lemma 3: Let $\Phi(\boldsymbol{wy}, \theta)$ denote a quadratic sigmoid function, where $\boldsymbol{w} = (w_0, w_1)$ and $\boldsymbol{y} = (1, x)$. Given a compact domain $C \subset \Re - \{(\sqrt{\theta} - w_0)/w_1\}$ and an error tolerance $\epsilon > 0$, then

$$|\Phi(\lambda \boldsymbol{wy}, \lambda^2\theta) - \mathcal{H}_g(\boldsymbol{wy}, \theta)| \le \epsilon. \quad \text{if } \lambda \ge \sqrt{\frac{\left|\log\left(\frac{1-\epsilon}{\epsilon}\right)\right|}{m}}, \quad \text{for all } x \in C$$

where $m = \min(\{|\theta - (w_0 + w_1 x)^2| | x \in C\})$.

Proof: The quadratic sigmoid function $\Phi(\boldsymbol{wy}, \theta)$ can be regarded as a variant of sigmoid function, i.e., $\Phi(\boldsymbol{wy}, \theta) = \sigma(\theta - (w_0 + w_1 x)^2)$, where $\sigma(x)$ denotes the conventional sigmoid function. Thus, according to Lemma 2, we obtain that

$$|\sigma(\lambda^2\theta - (\lambda w_0 + \lambda w_1 x)^2) - \mathcal{H}(\lambda^2\theta - (\lambda w_0 + \lambda w_1 x)^2)| \le \epsilon, \quad \text{for } |\lambda^2\theta - (\lambda w_0 + \lambda w_1 x)^2| \ge \left|\log\left(\frac{1-\epsilon}{\epsilon}\right)\right|.$$

Let $m = \min(\{|\theta - (w_0 + w_1 x)^2| | x \in C\})$. Since x cannot be $(\sqrt{\theta} - w_0)/w_1$, thus $m > 0$. Therefore, the above equation can be rewritten as

$$|\Phi(\lambda \boldsymbol{wy}, \lambda^2\theta) - \mathcal{H}_g(\lambda \boldsymbol{wy}, \lambda^2\theta)| \le \epsilon. \quad \text{if } \lambda \ge \sqrt{\frac{\left|\log\left(\frac{1-\epsilon}{\epsilon}\right)\right|}{m}}, \quad \text{for all } x \in C.$$

By Lemma 1, we obtain that

$$|\Phi(\lambda \boldsymbol{wy}, \lambda^2\theta) - \mathcal{H}_g(\boldsymbol{wy}, \theta)| \le \epsilon. \quad \text{if } \lambda \ge \sqrt{\frac{\left|\log\left(\frac{1-\epsilon}{\epsilon}\right)\right|}{m}}, \quad \text{for all } x \in C$$

where $m = \min(\{|\theta - (w_0 + w_1 x)^2| | x \in C\})$.

Corollary 1: Let $\Psi(\boldsymbol{wy}, \boldsymbol{\Theta})$ and $\Omega(\boldsymbol{wy}, \boldsymbol{\Theta})$ denote an extended quadratic sigmoid function and an extended quadratic Heaviside function, respectively, where $\boldsymbol{w} = (w_0, w_1, \cdots, w_n)$, $\boldsymbol{\Theta} = (\theta_0, \theta_1, \cdots, \theta_n)$, and $\boldsymbol{y} = (1, y_1, y_2, \cdots, y_n)$. Given a compact set

$$C \subset \Re^n - \left\{(x_1, x_2, \cdots, x_n) \middle| \left(\theta_0 + \sum_1^n \theta_i x_i\right) - \left(w_0 + \sum_1^n w_i x_i\right)^2 = 0\right\}$$

and an error tolerance $\epsilon > 0$, then

$$|\Psi(\lambda \boldsymbol{wy}, \lambda^2\Theta) - \Omega(\boldsymbol{wy}, \Theta)| \le \epsilon. \quad \text{if } \lambda \ge \sqrt{\frac{\left|\log\left(\frac{1-\epsilon}{\epsilon}\right)\right|}{m}}, \quad \text{for all } \boldsymbol{y} \in C$$

where

$$m = \min\left(\left\{\left|\left(\theta_0 + \sum_1^n \theta_i y_i\right) - \left(w_0 + \sum_1^n w_i y_i\right)^2\right| \cdot |(y_1, y_2, \cdots, y_n) \in C\right\}\right).$$

Lemma 4: Let $\Phi(\boldsymbol{wy}, \theta)$ denote a quadratic sigmoid function. Suppose that vector $\boldsymbol{y} = (1, x)$ and $\boldsymbol{w}_c = (c, 0)$ such that $\partial\Phi(\boldsymbol{w}_c\boldsymbol{y}, \theta)/\partial\text{net} = \mu \ne 0$, where $\text{net} = \boldsymbol{wy} = w_0 + w_1 x$. Let $C \subset \Re$ be a compact domain. There exists a weight vector $\boldsymbol{w}_\lambda = (c - c/\lambda, 1/\lambda)$, such that

$$\lim_{\lambda\to\infty} \frac{\lambda}{\mu}[\Phi(\boldsymbol{w}_\lambda\boldsymbol{y}, \theta) - \Phi(\boldsymbol{w}_c\boldsymbol{y}, \theta)] + c \to x, \quad \text{for all } x \in C.$$

Proof: For the purpose of convenience, let $f(\text{net}, \theta)$ denote the quadratic sigmoid function, where $\text{net} = \boldsymbol{wy} = w_0 + w_1 x$. Thus, $\Phi(\boldsymbol{w}_c\boldsymbol{y}, \theta) = f(c, \theta)$. Since $\partial\Phi(\boldsymbol{w}_c\boldsymbol{y}, \theta)/\partial\text{net} = \mu \ne 0$, thus

$$\lim_{\lambda\to\infty} \frac{f\left(c + \frac{x-c}{\lambda}, \theta\right) - f(c, \theta)}{\frac{x-c}{\lambda}} \to \mu \ne 0.$$

Rearranging the terms in the above equation, we obtain that

$$\lim_{\lambda\to\infty} \frac{\lambda}{\mu}\left[f\left(c + \frac{x-c}{\lambda}, \theta\right) - f(c, \theta)\right] + c \to x.$$

In other words, there exists a weight vector $\boldsymbol{w}_\lambda = (c - c/\lambda, 1/\lambda)$, such that

$$\lim_{\lambda\to\infty} \frac{\lambda}{\mu}[\Phi(\boldsymbol{w}_\lambda\boldsymbol{y}, \theta) - \Phi(\boldsymbol{w}_c\boldsymbol{y}, \theta)] + c \to x, \quad \text{for all } x \in C. \qquad \square$$

With the above auxiliary lemmas, the following theorem can be proved.

Theorem 2: Given a training set

$$S = \{y_1, y_2, \cdots, y_{4k+1} | y_i \in \Re, 1 \le i \le 4k+1\}$$

a single-hidden-layer network containing at most $k+1$ single-threshold quadratic sigmoidal hidden neurons and one multithreshold quadratic sigmoidal output neuron can dichotomize an arbitrary dichotomy defined on S.

Proof: Consider each quadratic Heaviside hidden neuron i of the network constructed in Theorem 1. Since

$$\frac{\sqrt{\theta_i^{(1)}} - w_{i,0}}{w_{i,1}} = \frac{\sqrt{\left(\frac{\beta'_{i-1} - \beta_{i-1}}{2}\right)^2} - \left(-\frac{\beta'_{i-1} + \beta_{i-1}}{2}\right)}{1} = \beta'_{i-1},$$

and we have assumed that β'_{i-1} is not contained in any interval I_i for $1 \le i \le 4k+1$ in Theorem 1, thus according to Lemma 3, each term of quadratic Heaviside function $(h(w_{i,0} + w_{i,1}x, \theta_i))$ in (6) can be replaced by a single-threshold quadratic sigmoidal neuron with activation function $\Phi(\lambda \boldsymbol{wy}, \lambda^2\theta)$ for large enough λ. Let h_i denote

the output of the ith hidden neuron. In the proof of Theorem 1, we have seen that for any input in

$$\cup_{i=1}^{k}\{I_{4i+2} \cup I_{4i+3} \cup I_{4i+4}\},$$

one and only one hidden neuron will output "1." Thus, based on the settings in Theorem 1, the term $g(\boldsymbol{\Theta}, \boldsymbol{y}) - (\boldsymbol{wy})^2$ for multithreshold quadratic Heaviside output neuron is equal to $\theta_i^{(2)} + \theta_0^{(2)} - (x + a_i)^2 (i \neq 0)$, where $\boldsymbol{y} = (1, x, h_1, h_2, \cdots, h_k)$, and $\boldsymbol{w} = (a_0, c, a_1, a_2, \cdots, a_k)$. Since

$$\theta_i^{(2)} = \left(\frac{\gamma'_{i-1} - \gamma_{i-1}}{2}\right)^2$$

and

$$a_i = -\frac{\gamma'_{i-1} + \gamma_{i-1}}{2}.$$

In addition, we have also assumed that γ'_{i-1} and γ_{i-1} are not contained in any interval I_i for $1 \leq i \leq 4k + 1$ in Theorem 1, i.e., $x \neq \gamma'_{i-1}$ and $x \neq \gamma_{i-1}$. Thus, $\theta_i^{(2)} + \theta_0^{(2)} - (x + a_i)^2$ cannot be zero. Similarly, for any input in I_{4i+1}, no hidden neuron outputs "1." Hence, the term $g(\boldsymbol{\Theta}, \boldsymbol{y}) - (\boldsymbol{wy})^2 = \theta_0^{(2)} - x^2$, where $\theta_0^{(2)} = \max\{a_0^2, a_k^2\}$. Since both a_0 and a_k are not contained in any interval I_i for $1 \leq i \leq 4k + 1$ in Theorem 1, $\theta_0^{(2)} - x^2$ cannot be zero. Thus, we conclude taht $g(\boldsymbol{\Theta}, \boldsymbol{y}) - (\boldsymbol{wy})^2 \neq 0$ for x in $I_i (1 \leq i \leq 4k + 1)$ based on the settings in Theorem 1. Hence, by Corollary 1, we can use a multithreshold quadratic sigmoidal neuron to replace the multithreshold quadratic Heaviside output neuron of the network constructed in Theorem 1. For the linear term, $c \cdot x = 1 \cdot x = x$, in (6), according to Lemma 4, we can use another single-threshold quadratic sigmoidal neuron to approximation this linear function. Therefore, in summary, we can use $k + 1$ single-threshold quadratic sigmoidal hidden neurons and one multithreshold quadratic sigmoidal output neuron to implement the function of the nonfeedforward network constructed by Theorem 1. □

In Sontag's work [6], he had proved that the upper bound on the number of hidden neurons required by a feedforward sigmoidal network for dichotomizing a training set with $2k$ training patterns is k. For the multithreshold quadratic sigmoidal networks, according to Theorem 2, only at most $\lceil k/2 \rceil$ multithreshold quadratic sigmoidal hidden neurons are enough. Let us compare the multithreshold quadratic sigmoidal networks with the conventional sigmoidal networks in terms of the number of free parameters. Suppose that the input dimension is n. Given a training set with $4k + 1$ training patterns, then the multithreshold quadratic sigmoidal network requires at most $(k + 1)(n + 2) + 2(k + 1)$ free parameters. However, the sigmoidal network requires at most $2k(n + 1) + (2k + 1)$ free parameters. Thus, for problems with large input dimensions (n is large), the upper bound on the number of required free parameters for multithreshold quadratic sigmoidal networks is only one half of that of sigmoidal networks. For a training set with a large number of patterns (k is large), the ratio between the upper bounds on the numbers of required free parameters for multithreshold quadratic sigmoidal networks and sigmoidal networks is $(n + 4)/(2n + 4)$. Thus, the improved ratio (say ξ) for the multithreshold quadratic sigmoidal networks is $6/5 \leq \xi < 2$.

IV. Concluding Remarks and Future Works

In this letter, a new type of neurons called multithreshold quadratic sigmoidal neurons are proposed to improve the classification of capability of multilayer neural networks. Using a multithreshold quadratic sigmoidal neuron in the output layer and $k + 1$ single-threshold quadratic sigmoidal neurons in the hidden layer, a single-hidden-layer neural network can be constructed to dichotomize arbitrary dichotomy defined on any training set with at least $4k + 1$ training patterns. Thus, in comparison with the committed machines (feedforward and Heaviside activation function) [5], we can claim that the multithreshold quadratic sigmoidal neurons have improved the recognition capability of single-hidden-layer neural networks by a factor of 4. In fact, a version of a backprop-like learning algorithm for multithreshold quadratic sigmoidal neural networks also can be easily derived based on the gradient descent method. Research into two interesting topics on the multithreshold quadratic sigmoidal neural networks is underway:

1) Capability studies on more complicated architectures, such as nonfeedforward networks, networks with more layers, or networks with multithreshold quadratic sigmoidal neurons in hidden layers.
2) Practical application studies on multithreshold quadratic sigmoidal neural networks.

References

[1] E. B. Baum and D. Haussler, "What size net gives valid generalization?" *Neural Computation*, vol. 1, pp. 151–160, 1989.

[2] C. C. Chiang and H. C. Fu, "A variant of second-order multilayer perceptron and its application to function approximations," in *IJCNN '92*, Baltimore, MD, 1992, pp. III:887–III:892.

[3] C. L. Giles and T. Maxwell, "Learning, invariance, and generalization in high-order neural networks," *Appl. Opt.*, vol. 26, pp. 4972–4978, 1987.

[4] S. C. Huang and Y. F. Huang, "Bounds on the number of hidden neurons in multilayer perceptrons," *IEEE Trans. Neural Networks*, vol. 2, no. 1, pp. 47–55, 1991.

[5] N. J. Nilsson, *Learning Machines: Foundation of Trainable Pattern-Classifying Systems*. New York: McGraw-Hill, 1965.

[6] E. D. Sontag, "On the recognition capabilities of feedforward nets," Tech. Rep. SYCON 90-03, SYCON-Rutgrs Center Syst. Contr., Dep. Mathematics, Rutgers Univ., New Brunswick, NJ, Apr. 1990.

A Learning Law for Density Estimation

Dharmendra S. Modha and Yeshayahu Fainman

***Abstract*—Probability density functions are estimated by an exponential family of densities based on multilayer feedforward networks. The role of the multilayer feedforward networks, in the proposed estimator, is to approximate the logarithm of the probability density functions. The method of maximum likelihood is used, as the main contribution, to derive an unsupervised backpropagation learning law to estimate the probability density functions. Computer simulation results demonstrating the use of the derived learning law are presented.**

I. Introduction

The joint probability density function of an observation vector that is assumed to follow a random process, embodies all the

Manuscript received November 29, 1992. This work was supported by UCSD Academic Senate Grant RS56-G/Fainman.

The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0407.

IEEE Log Number 9208460.