Applied Mathematics & Information Sciences
An International Journal

© 2012 NSP
Natural Sciences Publishing Cor.

Analysis on complexity of neural networks using integer weights

Bao Jian, Zhou LiangJie and Yan Yi

Institute of Software and Intelligent Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Received: Aug. 15, 2011; Revised Nov. 24, 2011; Accepted Dec. 16, 2011
Published online: 1 May 2012

Abstract: Integer neural network has been already extensively applied, but the result of application is always due to its operators because of the lack of theoretical guidance. In our paper, we will introduce a theoretical result that an integer network can present a good performance on approximating continuous function. Firstly we will make a quantitative analysis on integer networks capability of approximating a function from function spaces or . Then two new formulas will be first given to calculate the approximation error. Correspondently, we will give two new formulas to calculate the number of neurons in a three-layer network which has certain significance to engineering practice to construct a reasonable network.

Keywords: Artificial neural networks, Complexity, Integer weights, Approximation.

1. Introduction

Since Warren McCulloch, a neurophysiologist, and Walter Pitts, an logician, proposed the concept of artificial neurons and neural network mathematical model in 1943[15, 16] the researches on neural network became an important and practical area in computer science. After the development over half of a century, neural networks achieved undeniable success in a wide range, including pattern recognition, automatic control, signal processing, aid decision making, artificial intelligence, etc, as the statement of the paper of R.P.Lippmann about pattern classification[12] and the paper of A.Lapedes and R.Farber relating to nonlinear time series prediction[11].

Previous theoretical research always didn't care about the limit of the structure of network, the number of neurons, the precision of weights[2,4,7,8,10,13]. Such results have been crucial in building a solid theoretical foundation for neural network techniques.

However, ultimate success of any technique is determined by the success of its practical applications in real-world problems. We must consider the structure of networks and the precision of weights for the applications in real-world problems. In many case, we often encounter such situation: we can only offer a limited accuracy for the weight or a small number of neurons[17,24]. Most software implementations use a limited precision for number representation. Digital hardware implementations use a few bits to store the number of weights and this translates into a limited precision for the weights. Even analog implementations which are often cited for their ability to implement easily real numbers as analog quantities are limited in their precision by issues like dynamic range, noise, VLSI area and power dissipation problems, as the statement of the paper of Draghici.S.[5,24]. In this paper, we pay our attention to the neural networks with integer weights.

Some experimental and statistical methods[3,9,20–22, 1,14,17,25]have represented that a integer network can also provided a good performance for certain applications. But these results lack theoretical support, which hindered the application of the integer weight network.

With further researches, the theoretical analysis on limited precision weight network has been paid much more attention in recent years[5,23,25]. Here is a generalized problem arising, the capability of approximating continuous function by integer neural network. In this paper, we focus on this problem. Firstly, we will demonstrate that integer neural networks can't approximate any continuous function with arbitrarily small error. So we put our attention on finding a best approximation or a good approximation which our integer neural network could achieve. In

* Corresponding author: e-mail: zlj844230@gmail.com

© 2012 NSP
Natural Sciences Publishing Cor.

this paper, we will show: (i) a good approximation could be achieve by integer neural network; (ii) how to construct the integer neural network for a good approximation?

In this section we discuss how our work relates to current research trends. The rest of the paper is organized as follows. Section 2 reviews some preliminary concepts and definition that will be used at the following sections. Section 3 expounds and proofs the main results. Finally, Section 4 summarizes the main general results of the paper.

2. Preliminaries

A network is an acyclic graph which has several input nodes, hide nodes and some output nodes. In this paper, we focus on the network with one hide layer, so in our network, the nodes were divided into three layer, the nodes from different layer was connected by a line with a weight. Each node (also called neuron) calculates a function of the weighted sum of its m inputs as follows:

$$f(X) = \sigma(W^T X + \theta) \tag{1}$$

Where $f(X)$ is the output of the node, W^T is the weight vector, X is the input vector, θ is the threshold of the node and the $\sigma(\cdot)$ is the activation function. Here, without loss of generality, assume that only one node in the output layer, the neural network can be represent by the equation:

$$G(X) = \sum_{i=1}^{N} \alpha_i \sigma(W_i^T X + \theta_i) \tag{2}$$

Where $G(X)$ is the output of the network, X is the input of the network, α_i is the gain of the ith hide node.

Definition 1.*A function $\sigma : \mathrm{R}^1 \to \mathrm{R}^1$ is called a sigmoid function, if it satisfies*

$$\begin{cases} \lim_{x\to-\infty} \sigma(x) = 0 \\ \lim_{x\to+\infty} \sigma(x) = 1 \end{cases} \tag{3}$$

Definition 2.*A function $\sigma : \mathrm{R}^1 \to \mathrm{R}^1$ is called a step function, if it satisfies*

$$\begin{cases} \sigma(x) = 0 \ x < 0 \\ \sigma(x) = 1 \ x \geq 0 \end{cases} \tag{4}$$

Here, we can see that step function is a special form of sigmoid function and is the most typical and the simplest form.

Definition 3.*The signal-to-noise ratio (SNR) of a network is defined as:*

$$SNR = 10 \times \lg \frac{\|f\|}{\|f - G\|} \tag{5}$$

Where f is the target function, G is the neural network.

Definition 4.*The p-norms of a measurable function f from (S, Σ, μ) is defined as:*

$$\begin{cases} \|f\|_p = \sqrt[p]{\int_S |f|^p \, \mathrm{d}\mu} < \infty \ 1 \leq p < \infty \\ \|f\|_p = \max(|f|) < \infty \qquad p = \infty \end{cases} \tag{6}$$

Where (S, Σ, μ) be a measure space.

Definition 5.*The L^p spaces are function spaces defined using natural generalizations of p-norms for finite-dimensional vector spaces*

Definition 6.*$f(x)$ is a continuous function on $[a, b]$, for any $x, y \in [a, b]$ if it satisfies*

$$|f(x) - f(y)| \leq M|x - y|^{\alpha} (0 < \alpha \leq 1) \tag{7}$$

We say $f(x) \in \mathrm{Lip}_M \alpha$, here $\mathrm{Lip}_M \alpha$ is the α exponent M coefficient Lipschitz function class.

Definition 7.*The symbol $C[a, b]^n$ represent n-dimension continuous function space on $[a, b]^n$.*

Definition 8.*The symbol $\pm\mathrm{nbits}$ represent n-bits integer number between $+(2^n - 1)$ and $-(2^n - 1)$.*

Definition 9.*The symbol $+\mathrm{nbits}$ represent n-bits integer number between $+(2^n - 1)$ and 0.*

Then we will establish the mathematical model for our network which we will discuss in the following section. First of all, we must choose a kind of function as the activation function. Sigmoid function is most widely used [4, 6, 19]. But using the sigmoid function directly will make some difficulty for the analysis, because the sigmoid function has many uncertain characteristics and we can't use the limit theory to simplify the problem [4, 6, 19] owing to integer weights. So we use the step function firstly and then try to analyze the relation between the sigmoid network and the step network. Next we need a rule to measure the error of the network. In practical applications, we always don't care the absolute error, for example, if the absolute error is 5 and the target function amplitude is 10, or if the absolute error is 5 and the target function amplitude is 10000, we can see the later system is more accurate than the previous one. If the neural network was treated as a signal system and the output was regarded as the superposition of the noise and the signal. We can use the signal-to-noise ratio (SNR) to describe the performance of our neural network.

3. Main results

In this section, we begin to show our main results. First we will show a simple result that many sigmoid active functions can't make the integer weight network approximate any continuous function with arbitrarily small error. Let's see a kind sigmoid function:

$$\mathrm{NDF} = \left\{ \sigma : \sigma(x) = \begin{cases} 0 \ x \leq 0 \\ 1 \ x \geq 1 \end{cases} ; \right\} \tag{8}$$

© 2012 NSP
Natural Sciences Publishing Cor.

Theorem 1.*The functions set*

$$FS^n = \left\{ \sum_{i=1}^{N} \alpha_i \times \sigma(W_i^T X + \theta_i) : \theta_i, \alpha_i \in \mathrm{Z}; \right. \\ \left. W_i \in \mathrm{Z}^n; X \in [0,1]^n; \sigma \in \mathrm{NDF} \right\}$$

is not dense on $C[0,1]^n$ *for the* $\|\cdot\|_\infty$.

*Proof.*We can find a continuous function f, let it meet the condition:

$$f \in C[0,1]^n, \quad f\left(\left(\underbrace{0,0,0,\cdots,0}_{n}\right)\right) = 0, \\ f\left(\left(\underbrace{1,1,1,\cdots,1}_{n}\right)\right) = \frac{1}{2} \tag{9}$$

From (8) we can easily get the result:

$$\alpha\sigma\left(W^T\left(\underbrace{1,1,1,\cdots,1}_{n}\right)+\theta\right) - \\ \alpha\sigma\left(W^T\left(\underbrace{0,0,0,\cdots,0}_{n}\right)+\theta\right) \in \mathrm{Z} \tag{10}$$

where $W \in \mathrm{Z}^n; \theta \in \mathrm{Z}; \alpha \in \mathrm{Z}; \sigma \in \mathrm{NDF}$. From (2)(10),for any network G we can get the following result:

$$G\left(\left(\underbrace{1,1,1,\cdots,1}_{n}\right)\right) - G\left(\left(\underbrace{0,0,0,\cdots,0}_{n}\right)\right) \\ = \sum_{i=1}^{N}\left[\alpha_i\sigma\left(W_i^T\left(\underbrace{1,1,1,\cdots,1}_{n}\right)+\theta_i\right)\right. \\ \left. -\alpha_i\sigma\left(W_i^T\left(\underbrace{0,0,0,\cdots,0}_{n}\right)+\theta_i\right)\right] \in \mathrm{Z} \tag{11}$$

So,from (9)(11)

$$\max\left(\left|G\left(\underbrace{1,1,1,\cdots,1}_{n}\right) - f\left(\underbrace{1,1,1,\cdots,1}_{n}\right)\right|, \right. \\ \left. \left|G\left(\underbrace{0,0,0,\cdots,0}_{n}\right) - f\left(\underbrace{0,0,0,\cdots,0}_{n}\right)\right|\right) \geq \frac{1}{4} \tag{12}$$

Then we can get the result

$$\|G - f\|_\infty = \max(|G(X) - f(X)|) \geq \frac{1}{4} \tag{13}$$

So the proposition holds.

Next, we will show a complexity result on the L^2 function space. Firstly, we construct a set $\mathrm{CS} = \{\langle a,b\rangle : a,b \text{ is coprime}; a < b; a,b \in +\mathrm{nbits}\}$ and the $|\mathrm{CS}|$ is defined the number of the elements of CS; So we easily see that $|\mathrm{CS}| < \infty$. Here we sort the elements from CS as following:

$$0 < \frac{a_1}{b_1} < \frac{a_2}{b_2} < \cdots < \frac{a_{|\mathrm{CS}|-1}}{b_{|\mathrm{CS}|-1}} < \frac{a_{|\mathrm{CS}|}}{b_{|\mathrm{CS}|}} < 1 \tag{14}$$

where $\langle a_i, b_i\rangle \in \mathrm{CS}; i = 1,2,3,\cdots,|\mathrm{CS}|-1,|\mathrm{CS}|-1$. Then we construct a function set:

$$\mathrm{G}' = \left\{ g_0' = \sigma(x), g_{|\mathrm{CS}|+1}' = \sigma(x-1), \right. \\ g_1' = \sigma(b_1 x - a_1), g_2' = \sigma(b_2 x - a_2), \cdots, \\ \left. g_{|\mathrm{CS}|}' = \sigma\left(b_{|\mathrm{CS}|}x - a_{|\mathrm{CS}|}\right)\right\} \tag{15}$$

Where σ is a step function.

Lemma 1.*For any function as following:*

$$f(x) = \sum_{i=1}^{n} \alpha_i \sigma(y_i x + \theta_i) : \alpha_i \in \mathrm{R}; y_i, \theta_i \in \pm\mathrm{nbits}; \\ \sigma \text{ is a step function} \tag{16}$$

on $[0,1]^n$ *can be represented by the linear combination of functions from* G' *except several points.*

*Proof.*For any functions $\sigma(yx+\theta) : y, \theta \in \pm\mathrm{nbits}$ on $[0,1]^1$:

1.if $y = 0, \theta \geq 0$ then $\sigma(yx+\theta) = g_0'$;
2.if $y = 0, \theta < 0$ then $\sigma(yx+\theta) = g_{|\mathrm{CS}|+1}'$ except $x = 1$;
3.if $y > 0, \frac{-\theta}{y} > 1$ then $\sigma(yx+\theta) = g_{|\mathrm{CS}|+1}'$ except $x = 1$;
4.if $y > 0, \frac{-\theta}{y} = 1$ then $\sigma(yx+\theta) = g_{|\mathrm{CS}|+1}'$;
5.if $y > 0, 0 < \frac{-\theta}{y} < 1$ then $\sigma(yx+\theta) = g_i'$ where $\frac{a_i}{b_i} = \frac{-\theta}{y}$;
6.if $y > 0, \frac{-\theta}{y} \leq 0$ then $\sigma(yx+\theta) = g_0'$;
7.if $y < 0, \frac{-\theta}{y} \geq 1$ then $\sigma(yx+\theta) = g_0'$;
8.if $y < 0, 0 < \frac{-\theta}{y} < 1$ then $\sigma(yx+\theta) = g_0' - g_i'$ where $\frac{a_i}{b_i} = \frac{-\theta}{y}$ except $x = \frac{-\theta}{y}$;
9.if $y < 0, \frac{-\theta}{y} = 0$ then $\sigma(yx+\theta) = g_{|\mathrm{CS}|+1}'$ except $x = 0$ and $x = 1$;
10.if $y < 0, \frac{-\theta}{y} < 0$ then $\sigma(yx+\theta) = g_{|\mathrm{CS}|+1}'$ except $x = 1$;

So we easily know that functions(16) can be represented by the linear combination of function from G' except several points.

Next, we get another function set

$$\mathrm{GS} = \left\{g_0, g_{|\mathrm{CS}|+1}, g_1, g_2, \cdots, g_{|\mathrm{CS}|+1}, g_{|\mathrm{CS}|}\right\}$$

© 2012 NSP
Natural Sciences Publishing Cor.

where

$$\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{|\mathrm{CS}|-1} \\ g_{|\mathrm{CS}|} \\ g_{|\mathrm{CS}|+1} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} g'_0 \\ g'_1 \\ \vdots \\ g'_{|\mathrm{CS}|-1} \\ g'_{|\mathrm{CS}|} \\ g'_{|\mathrm{CS}|+1} \end{bmatrix} \tag{17}$$

for

$$\begin{vmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{vmatrix} \neq 0$$

So functions(16) also can be represented by the linear combination of function from GS except several points; in other word, for any $f(x)$, we can find a linear combination ϕ of functions from GS so that $\|f-\phi\|_2 = 0$.

Theorem 2.*The functions set* $\mathrm{NW} = \left\{\sum_{i=1}^{N} \alpha_i \sigma(y_i x + \theta_i) : \alpha_i \in \mathrm{Z}; y_i, \theta_i \in \pm\mathrm{nbits}\right\}$ *where σ is a step function, for any continuous function $t(x) \in L^2$ on $[0,1]^1$ there exist a $\phi^{**} \in \mathrm{NW}$ let*

$$\|t-\phi^{**}\|_2 \leq \sqrt{\|t\|_2^{\,2} - \sum_{i=0}^{|\mathrm{CS}|+1} \frac{\left(\int_0^1 t(x) g_i(x)\,\mathrm{d}x\right)^2}{\|g_i\|_2^{\,2}} + \frac{1}{4}} \tag{18}$$

*Proof.*Firstly, for the inner product $g_i \bullet g_j = \int_0^1 g_i(x) g_j(x)\,\mathrm{d}x = 0$ where $i \neq j$ and it is obvious that the elements in set GS is linear independence. So from Lemma1 we get the result that GS is a group of orthogonal basis of the function space:

$$\mathrm{GF}' = \left\{\sum_{i=1}^{n} \alpha_i \sigma(y_i x + \theta_i) : \alpha_i \in \mathrm{R};\ y_i, \theta_i \in \pm\mathrm{nbits}; \sigma \text{ is a step function}\right\}$$

For the target function $t(x)$ from the space L^2. From I. P. Natanson [18] results we can see the

$$\phi^* = \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \cdots & \alpha_{|\mathrm{CS}|} & \alpha_{|\mathrm{CS}|+1} \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_3 \\ \vdots \\ g_{|\mathrm{CS}|} \\ g_{|\mathrm{CS}|+1} \end{bmatrix} \tag{19}$$

is the best approximation element from GF' and $(t-\phi^*) \bullet g_i = 0$, where

$$\alpha_i = \frac{g_i \bullet t}{\|g_i\|_2^{\,2}} \tag{20}$$

So we get the equation

$$\|t-\phi^*\|_2^{\,2} = \|t\|_2^{\,2} - \sum_{i=0}^{|\mathrm{CS}|+1} \frac{\left(\int_0^1 t(x) g_i(x)\,\mathrm{d}x\right)^2}{\|g_i\|_2^{\,2}} \tag{21}$$

Then we construct our network function

$$\phi^{**} = \begin{bmatrix} \mathrm{round}(\alpha_0) & \mathrm{round}(\alpha_1) & \cdots & \mathrm{round}(\alpha_{|\mathrm{CS}|-1}) \\ \mathrm{round}(\alpha_{|\mathrm{CS}|}) & \mathrm{round}(\alpha_{|\mathrm{CS}|+1}) \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{|\mathrm{CS}|-1} \\ g_{|\mathrm{CS}|} \\ g_{|\mathrm{CS}|+1} \end{bmatrix} \tag{22}$$

Where $\mathrm{round}(\cdot)$ is a function to integer and satisfy $|\mathrm{round}(x) - x| \leq 0.5$. So, $\phi^{**} \in \mathrm{NW}$ and

$$\sum_{i=0}^{|\mathrm{CS}|+1} (\alpha_i - \mathrm{round}(\alpha_i))^2 \|g_i\|_2^{\,2} \leq \frac{1}{4} \sum_{i=0}^{|\mathrm{CS}|+1} \|g_i\|_2^{\,2} = \frac{1}{4} \tag{23}$$

Then

$$\begin{aligned} \|t-\phi^{**}\|_2^{\,2} &= \|t-\phi^* + \phi^* - \phi^{**}\|_2^{\,2} \\ &= \|t-\phi^*\|_2^{\,2} + 2(t-\phi^*) \bullet (\phi^* - \phi^{**}) + \\ &\qquad \|\phi^* - \phi^{**}\|_2^{\,2} \\ &= \|t-\phi^*\|_2^{\,2} + \|\phi^* - \phi^{**}\|_2^{\,2} \\ &= \|t-\phi^*\|_2^{\,2} + \sum_{i=0}^{|\mathrm{CS}|+1} (\alpha_i - \mathrm{round}(\alpha_i))\, g_i \\ &\qquad \bullet \sum_{i=0}^{|\mathrm{CS}|+1} (\alpha_i - \mathrm{round}(\alpha_i))\, g_i \\ &= \|t-\phi^*\|_2^{\,2} + \sum_{i=0}^{|\mathrm{CS}|+1} (\alpha_i - \mathrm{round}(\alpha_i))^2 \|g_i\|^2 \\ &\leq \|t-\phi^*\|_2^{\,2} + \frac{1}{4} \\ &= \|t\|_2^{\,2} - \sum_{i=0}^{|\mathrm{CS}|+1} \frac{\left(\int_0^1 t(x) g_i(x)\,\mathrm{d}x\right)^2}{\|g_i\|_2^{\,2}} + \frac{1}{4} \end{aligned} \tag{24}$$

© 2012 NSP
Natural Sciences Publishing Cor.

So (18) holds.

From above theorem, we can get an error estimates for a n-bits network, and then we want to know we need how many neurons in the hidden layer. From the proof of the theorem, we can see the number of the neurons related with the coefficient α_i and one function $g_i(x)$ need two neurons. So we need

$$\left\lceil \frac{|\text{round}(\alpha_i)|}{2^n-1} \right\rceil \times 2 \tag{25}$$

neurons to implement the ist item of ϕ^{**}.So, totally, we need the number of neurons is

$$N\text{-}number_2 = \sum_{i=0}^{|\text{CS}|+1} \left\lceil \frac{|\text{round}(\alpha_i)|}{2^n-1} \right\rceil \times 2 \tag{26}$$

for implementing ϕ^{**}.

Next, we will show some results on the L^∞ function space. We define a symbol $MS = \max\left(\frac{a_{i+1}}{b_{i+1}} - \frac{a_i}{b_i}\right)$ where $i = 0, 1, 2, \cdots, |\text{CS}|$.

Theorem 3.*The functions set* $\text{NW} = \left\{\sum_{i=1}^{N} \alpha_i \sigma(y_i x + \theta_i) : \alpha_i \in \text{Z}; y_i, \theta_i \in \pm\text{nbits}\right\}$ *where* σ *is a step function, a continuous function* $t(x) \in L^\infty$ *on* $[0,1]^1$ *and if* $t(x) \in \text{Lip}_{\frac{1}{2MS}} 1$ *and equation* $t(x) = \frac{c}{2} : c \in \text{Z}$ *have finite roots , we can get a function* $\phi^* \in \text{NW}$ *and let it satisfy*

$$\|t - \phi^*\|_\infty \le 1 \tag{27}$$

*Proof.*Firstly, for the target function $t(x)$,let $M = \lceil \max(t(x)) \rceil$ and $m = \lfloor \min(t(x)) \rfloor$, then we get a sequence number

$$\begin{aligned} m = y_0 < m + \frac{1}{2} = y_1 < m + 2 \times \frac{1}{2} < \cdots < y_i \\ = m + i \times \frac{1}{2} < \cdots < M = y_{2(M-m)} \end{aligned} \tag{28}$$

Then we use the line group $y = y_i : i = 0, 1, 2, \cdots, 2(M-m)$ to intersect $t(x)$ at a group points $x_1 < x_2 < x_3 < \cdots < x_n$ where $n < +\infty$ because of $t(x) = \frac{c}{2} : c \in \text{Z}$ have finite roots;Not losing general, let $x_1 > 0$ and $x_n < 1$, so we construct a points group

$$0 = x_0 < x_1 < x_2 < \cdots < x_n < x_{n+1} = 1 \tag{29}$$

on x-axis;and $|t(x_i) - t(x_{i+1})| = \frac{1}{2} : i = 1, 2, \cdots, n-1$ because $t(x)$ is a continuous function; And for $t(x) \in \text{Lip}_{\frac{1}{2MS}} 1$, we can know

$$|t(x_i) - t(x_{i+1})| \le \frac{1}{2MS} |x_i - x_{i+1}| \Rightarrow MS \le |x_i - x_{i+1}| \tag{30}$$

So there is at least one element of CS in the interval $[x_i, x_{i+1}]$,get one element for every interval, we assume those element sequence is $0 < \frac{a'_1}{b'_1} < \frac{a'_2}{b'_2} < \frac{a'_3}{b'_3} < \cdots < \frac{a'_{n-1}}{b'_{n-1}} < 1$, we merge the same items then get the sequence

$$0 < \frac{a^*_1}{b^*_1} < \frac{a^*_2}{b^*_2} < \frac{a^*_3}{b^*_3} < \cdots < \frac{a^*_k}{b^*_k} < 1 \tag{31}$$

And the points sequence have the property

$$\begin{cases} \left| t\left(\frac{a^*_i}{b^*_i}\right) - t\left(\frac{a^*_{i+1}}{b^*_{i+1}}\right) \right| \le 1 \; i = 1, 2, 3, \cdots, k-1 \\ \left| t\left(\frac{a^*_1}{b^*_1}\right) - t(0) \right| \le 1 \\ \left| t\left(\frac{a^*_k}{b^*_k}\right) - t(1) \right| \le 1 \end{cases} \tag{32}$$

Then we define a function $T(x, y)$ map from R^2 to Z as

$$T(x,y) = \begin{cases} \lceil t(x) \rceil \text{ if } \lceil t(x) \rceil \le \lceil t(y) \rceil \\ \lceil t(y) \rceil \text{ if } \lceil t(x) \rceil > \lceil t(y) \rceil \end{cases} \tag{33}$$

So we can construct a function $\phi^* \in \text{NW}$ as

$$\begin{aligned} \phi^*(x) = T\left(0, \frac{a^*_1}{b^*_1}\right) \times [\sigma(x) - \sigma(a^*_1 x - b^*_1)] \\ + T\left(\frac{a^*_k}{b^*_k}, 1\right) \times [\sigma(a^*_k x - b^*_k) - \sigma(x-1)] \\ + \sum_{i=1}^{k-1} T\left(\frac{a^*_i}{b^*_i}, \frac{a^*_{i+1}}{b^*_{i+1}}\right) \times [\sigma(a^*_i x - b^*_i) - \\ \sigma(a^*_{i+1} x - b^*_{i+1})] \end{aligned} \tag{34}$$

For one $x \in [0,1]^1$, if $x \in \left[\frac{a^*_i}{b^*_i}, \frac{a^*_{i+1}}{b^*_{i+1}}\right)$,then

$$\phi^*(x) = T\left(\frac{a^*_i}{b^*_i}, \frac{a^*_{i+1}}{b^*_{i+1}}\right) \tag{35}$$

Similarly when $x \in \left[0, \frac{a^*_1}{b^*_1}\right)$ and $x \in \left[\frac{a^*_k}{b^*_k}, 1\right]$.So $\|t - \phi^*\|_\infty \le 1$ holds.

Next we will concern the complexity of the network, as proof above, every item need

$$\left\lceil \frac{\left| T\left(\frac{a^*_i}{b^*_i}, \frac{a^*_{i+1}}{b^*_{i+1}}\right) \right|}{2^n - 1} \right\rceil \times 2 \tag{36}$$

neurons, so we need the number of neurons is

$$\begin{aligned} N\text{-}number_\infty = \left\lceil \frac{\left| T\left(\frac{a^*_1}{b^*_1}, 0\right) \right|}{2^n-1} \right\rceil \times 2 + \\ \left\lceil \frac{\left| T\left(1, \frac{a^*_k}{b^*_k}\right) \right|}{2^n-1} \right\rceil \times 2 + \sum_{i=1}^{k-1} \left\lceil \frac{\left| T\left(\frac{a^*_i}{b^*_i}, \frac{a^*_{i+1}}{b^*_{i+1}}\right) \right|}{2^n-1} \right\rceil \times 2 \end{aligned} \tag{37}$$

© 2012 NSP
Natural Sciences Publishing Cor.

Here if we use the SNR to measure the performance of the network

$$SNR = 10 \times \lg \frac{\|f\|}{\|f - G\|} \geq 10 \times \lg \frac{\|f\|}{1} = 10 \times \lg \|f\| \tag{38}$$

From (38), we can see that a good result could be achieved if $\|f\|$ is big enough. So in our application, there is a proposal, we can enlarge our target function by a coefficient then use the network to approximate it.

With the sigmoid active function, generally, the network can perform the foregoing results when the bits number is large enough because of the limit equation which is derived from the definition of sigmoid function:

$$\begin{cases} \lim_{\lambda \to \infty} \sigma(\lambda x + \theta) = 1 & x > 0 \\ \lim_{\lambda \to \infty} \sigma(\lambda x + \theta) = 0 & x < 0 \\ \lim_{\lambda \to \infty} \sigma(\lambda x + \theta) = \sigma(\theta) & x = 0 \end{cases} \tag{39}$$

where σ is a sigmoid function.

For every neuron, there is an error corresponding, a network error is the sum of those neurons error. So if the number of neurons is great enough, the sigmoid network would not get a reasonable approximation degree. Two ways we can use to compensate the error. First we can use even more bits. Second we can add neurons at some appropriate position. If we need quantitative analysis on the relation among error, adding neurons and number of bits, we could try to use the experimental and statistics methods for specific sigmoid active function and get some useful statistics results. In this paper we won't to discuss this problem. This problem and the sigmoid active function integer network theoretical issues will be discuss in our further paper.

Then let's see the multidimensional case. Firstly, we construct an active function as following

$$\sigma(x_1, x_2, \cdots, x_n) = \begin{cases} 1 \; 1 \geq x_i \geq 0 : i = 1, 2, 3, \cdots, n \\ 0 \;\; \text{else} \end{cases} \tag{40}$$

And the network output was described as following

$$G(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{N} \alpha_i \sigma(w_{1i} x_1 - \theta_{1i}, \\ w_{2i} x_2 - \theta_{2i}, \cdots, w_{ni} x_n - \theta_{ni}) \tag{41}$$

Here, $(\theta_{1i}, \theta_{2i}, \cdots, \theta_{ni})$ is the threshold vector of the ith neuron, $(w_{1i}, w_{2i}, \cdots, w_{ni})$ is the weight vector of the ith neuron, α_i is the gain of the ith neuron . For every one-dimension modal character, we can get correspondent character in this modal, so we can use the above results directly.

Another method, I.P.Natanson[18] had a result that the set $[0,1]^n$ and the set $[0,1]^1$ have the same cardinality, in other word, we can find a one-one map from $[0,1]^n$ to $[0,1]^1$, an instance of these map could be found in I.P. Natanson's book [18], here we assume we can find a map $\delta(\cdot)$. If $\delta(\cdot)$ is continuous, for a n-dimension continuous function $F(X)$, there is a one-dimension function continuous $f(x)$ which satisfy $f(\delta(X)) = F(X)$. So we can convert the n-dimension problem to a one-dimension problem. There are two issues arising: 1) can we fine a continuous map? 2) If we can't find a continuous map, how to treat or utilize those usable maps? We will put that content in our further paper.

4. Conclusions

In this paper, we mainly analyzed the integer network using step active function, its performance and the construction method. We described the performance of n-bits networks by two theorems and concluded that an n-bits network can achieve a good performance. Then we gave the construction parameters by the $N\text{-}number_2$ and $N\text{-}number_\infty$. Finally we introduced some discussions about the sigmoid function network and the n-dimension network. For the sigmoid function network and the n-dimension network, there is vast space remained for further study and that is our main direction in further work. Neural network would have more extensive applications in the engineering fields if we have a strict theoretical system.

Acknowledgement

The author is grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] Bellido, A., Bellido, I., & Fiesler, E. (1993). Do Backpropagation Trained Neural Networks have Normal Weight Distributions ? : Springer-Verlag.

[2] Chen, T., Chen, H., & Liu, R. (1992). A constructive proof and an extension of Cybenko's approximation theorem.

[3] Coggins, R., & Jabri, M. (1994). Wattle: A trainable gain analogue VLSI neural network. ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 874-874.

[4] Cybenko, G. (1989). Approximation by superpositions of a sigmoid function. Mathematics of Control, Signals, and Systems (MCSS), 2(4), 303-314.

[5] Draghici, S. (2002). On the capabilities of neural networks using limited precision weights. Neural networks, 15(3), 395-414.

[6] Ebert, T., Bänfer, O., & Nelles, O. (2010). Multilayer perceptron network with modified sigmoid activation functions. Artificial Intelligence and Computational Intelligence, 414-421.

[7] Funahashi, K. I. (1989). On the approximate realization of continuous mappings by neural networks. Neural networks, 2(3), 183-192.

[8] Hahm, N., & Hong, B. I. (2004). An approximation by neural networks with a fixed weight. Computers & Mathematics with Applications, 47(12), 1897-1903.

© 2012 NSP
Natural Sciences Publishing Cor.

[9] Hoehfeld, M., & Fahlman, S. E. (1992). Learning with limited numerical precision using the cascade-correlation algorithm. Neural Networks, IEEE Transactions on, 3(4), 602-611.
[10] Kolmogorov, A. N. (1963). On the Representation of Continuous Functions of Several Variables in the Form of Super Positions of Continuous Functions of One Variable and Additive Functions: SLA Translations Center.
[11] Lapedes, A., & Farber, R. (1987). Non-linear signal processing using neural networks: prediction and system modelling Technical Report LA-UR-87-2662. Los Alamos National Laboratory, Los Alamos, New Mexico, 87545.
[12] Lippmann, R. P. (1987). An introduction to computing with neural networks. IEEE Assp magazine, 4(2), 4-22.
[13] Lorentz, G. (1976). The 13th problem of Hilbert.
[14] Lundin, T., Fiesler, E., & Moerland, P. (1996). Connectionist quantization functions.
[15] McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biology, 5(4), 115-133.
[16] McCulloch, W. S., & Pitts, W. (1947). How we know universals: the perception of auditory and visual forms. Bulletin of mathematical Biophysics, 9, 127-147.
[17] Moerland, P., & Fiesler, E. (1996). Hardware-friendly learning algorithms for neural networks: an overview.
[18] Natanson, I. (1961). Theory of functions of a real variable, GITTL, Moscow, 1957. English transl., Ungar, New York.
[19] T.Chen. (1992). Approximation to nonlinear functional in Hilbert space by superposition of sigmoid funcions. Kexue Tongbao, 13, 1167-1169.
[20] Tang, C. Z., & Kwan, H. (1993). Multilayer feedforward neural networks with single powers-of-two weights. Signal Processing, IEEE Transactions on, 41(8), 2724-2727.
[21] Vincent, J., & Myers, D. (1992). Weight dithering and wordlength selection for digital backpropagation networks. BT Technology journal, 10, 124-133.
[22] Xie, Y., & Jabri, M. A. (1992, 7-11 Jun 1992). On the training of limited precision multi-layer perceptrons. Paper presented at the Neural Networks, 1992. IJCNN., International Joint Conference on.
[23] Zhang, Z., Ma, X., & Yang, Y. (2003). Bounds on the number of hidden neurons in three-layer binary neural networks. Neural networks, 16(7), 995-1002.
[24] Misra, J., & Saha, I. (2010). Artificial neural networks in hardware: A survey of two decades of progress. Neurocomputing, 74(1-3), 239-255.
[25] Johansson, C., & Lansner, A. (2009). Implementing plastic weights in neural networks using low precision arithmetic. Neurocomputing, 72(4-6), 968-972.

Bao Jian is a full tenured professor, one of founder of Institute of Software and Intelligent Technology at the Hangzhou Dianzi University, Hang Zhou. Her main research fields include computer architecture and intelligent control. In past 5 years, She published more than 20 articles published in scholarly journals, took charge of 5 provincial/ministry scientific research projects and has been awarded the ”ZheJiang Science and Technology Award” more than 5 times.

Zhou LiangJie is a postgraduate at Institute of Software and Intelligent Technology at the Hangzhou Dianzi University, Hang Zhou. His research fields are neural network, applied mathematics and intelligent control. In past 2 years, studying at the institute, he had participated in a number of practical projects about the engineering application of neural networks and have accumulated a lot of application experience of applying neural networks in embedded system.

Yi Yanis Director and Full Professor in the Institute of Intelligent and Software Technology at Hangzhou DianZi University. He received B.S. in automatic control engineering form Zhejian Sci-Tech University in 1984, M.S. in computer engineering from Beijin University of Postal Telecommumications in 1990 . His areas of research are related to embeded system, advanced manufacturing system, intelligent control & automation, and intelligent instruments.

© 2012 NSP
Natural Sciences Publishing Cor.