A Guide for the Upper Bound on the Number of Continuous-Valued Hidden Nodes of a Feed-Forward Network

Rua-Huan Tsaih[1,⋆] and Yat-wah Wan[2]

[1] Department of Management Information Systems,
National Chengchi University, Taipei, Taiwan 116
tsaih@mis.nccu.edu.tw

[2] Graduate Institute of Global Operations Strategy and Logistics Management,
National Dong Hwa University, Taiwan 974
ywan@mail.ndhu.edu.tw

Abstract. This study proposes and validates a construction concept for the realization of a real-valued single-hidden layer feed-forward neural network (SLFN) with continuous-valued hidden nodes for arbitrary mapping problems. The proposed construction concept says that for a specific application problem, the upper bound on the number of used hidden nodes depends on the characteristic of adopted SLFN and the observed properties of collected data samples. A positive validation result is obtained from the experiment of applying the construction concept to the m-bit parity problem learned by constructing two types of SLFN network solutions.

Keywords: Bound, hidden nodes, single-hidden layer feed-forward neural network, preimage, parity problem.

1 Bound on the Number of Hidden Nodes

With regard to the realization of a real-valued single-hidden layer feed-forward neural network (SLFN) with continuous-valued hidden nodes for arbitrary mapping problems, this study proposes and validates the *construction concept*. The proposed construction concept says that for any learning problem with specific data relationship observed among input vectors and target values, knowledge of the characteristic of adopted SLFN and the observed data properties helps find a better upper bound on the number of used hidden nodes of network solution than the one obtained from the conventional construction method that misses the characteristic and ignores the relationship.

The question of the necessary number of hidden nodes for a feed-forward neural network has been addressed in [1][3][5][6]. [3] argued that fewer hidden nodes is generally regarded as desirable for preventing over-learning, but the necessary number of hidden nodes is not known in general. Both of [1] and [6]

⋆ Corresponding author.

C. Alippi et al. (Eds.): ICANN 2009, Part I, LNCS 5768, pp. 658–667, 2009.

obtained a bound of N-1 for N distinct samples, but both assumed that the hidden layer of nodes produced a binary-valued output. To obtain the training advantages of back propagation of errors, most models use continuous-valued hidden node outputs, as we do here. The methods of [1] and [6] do not seem to generalize to our situation.

[5] adopted standard SLFNs with any nonlinear, continuous-valued activation function that has a limit at each infinity and claimed that N distinct samples can be fit perfectly through a SLFN with N hidden nodes. However, the number N is a loose upper bound on the number of hidden nodes of a SLFN solution for N distinct samples; for instance, [12] and [13] stated that the m-bit parity problem is solvable by a SLFN with merely $\lceil (m+1)/2 \rceil$ hidden nodes and with the sigmoid activation functions at hidden nodes, in which, and hereafter, $\lceil x \rceil$ denotes the smallest integer which is larger than or equal to x. In fact, instead of a loose and universal upper bound applied to the number of hidden nodes of SLFN solutions for all learning problems, most researchers and practitioners desire a concept to help find a better upper bound on the number of used hidden nodes of SLFN solution for a specific learning problem, as we do here. The discussion of [5][12][13] does not seem to provide such a concept. To address such challenge, we propose the construction concept.

In Section 2, this study shows that the conventional construction method of [5] misses the characteristic of the adopted SLFN and ignores the data relationship among input vectors and target values. This study then explores characteristics of SLFN through the preimage analysis, in which the preimage of a given output is the collection of inputs for the output. In Section 4, we set up the experiment of parity problem to validate the construction concept. The parity problem is a challenging benchmark for testing neural network learning algorithm. Some, but not exhaustive, recent studies of the parity problem can be found in [2][4][7][8][9][12][15]. Conclusions and future work are presented at the end.

2 A Conventional Construction Method

List of notations used in mathematical representations: Characters in bold represent column vectors, matrices or sets; $(\cdot)^{\mathrm{T}}$ denotes the transpose of $(\cdot)$.

- N $\equiv$ the amount of training samples;
- I $\equiv$ the amount of input nodes;
- J $\equiv$ the amount of hidden nodes;
- $\mathbf{x}$ $\equiv$ $(x_1, x_2, \cdots, x_I)^{\mathrm{T}}$: the input vector, in which x_i is the ith input component, with i from 1 to I;
- $\mathbf{a}$ $\equiv$ $(a_1, a_2, \cdots, a_J)^{\mathrm{T}}$: the hidden activation vector, in which a_j is the activation value of the jth hidden node, with j from 1 to J;
- y $\equiv$ the activation value of the output node and $y = f(\mathbf{x})$ with f being the map function of $\mathbf{x}$ and y;
- w_{ji}^{H} $\equiv$ the weight between the ith input variable and the jth hidden node, in which the superscript H throughout the paper refers to quantities related to the hidden layer;

$\mathbf{w}_j^H \equiv (w_{j1}^H, w_{j2}^H, \ldots, w_{jI}^H)^{\mathrm{T}}$, the vector of weights between the input variables and the jth hidden node;
$\mathbf{W}^H \equiv (\mathbf{w}_1^H, \mathbf{w}_2^H, \ldots, \mathbf{w}_J^H)^T$, the $J \times I$ matrix of weights between the input variables and the hidden nodes;
$w_{j0}^H \equiv$ the bias value of the jth hidden node;
$w_j^O \equiv$ the weight between the jth hidden node and the output node in which the superscript O throughout the paper refers to quantities related to the output layer;
$\mathbf{w}^O \equiv (w_1^O, w_2^O, \ldots, w_J^O)^{\mathrm{T}}$; and
$w_0^O \equiv$ the bias value of the output node.

The construction method of [5] works for *any* activation function g as long as $g(x_{01}) \neq \lim_{x\to+\infty} g(x)$. Let $\mathbf{x}^c$ and t^c be the cth input pattern and the corresponding target value, respectively, $c = 1, \ldots, N$. Without the loss of generality, assume that $\mathbf{x}^c \neq \mathbf{x}^d$ for $1 \leq c \neq d \leq N$. Let $\mathbf{T} \equiv (t^1, t^2, \ldots, t^N)^{\mathrm{T}}$ be the N-dimensional vector of target values for the N input samples; $x_{01} > x_{02}$ be two arbitrary pre-specified constants. The construction method first arbitrarily chooses an I-dimensional vector $\mathbf{w}$ such that

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}^1 < \mathbf{w}^{\mathrm{T}}\mathbf{x}^2 < \ldots < \mathbf{w}^{\mathrm{T}}\mathbf{x}^N. \tag{1}$$

For this $\mathbf{w}$, the construction method then calculates $\mathbf{w}_j^H$ and w_{j0}^H from eqt. (3), in which the values of $\mathbf{w}_j^H$ and w_{j0}^H are independent of the target outputs $\{t^c\}$:

$$\mathbf{w}_j^H = \{ \begin{array}{ll} \mathbf{0}, & \text{if } j = 1; \\ \frac{x_{02}-x_{01}}{\mathbf{W}^{\mathrm{T}}\mathbf{X}^j - \mathbf{W}^{\mathrm{T}}\mathbf{X}^{j-1}}\mathbf{w}, & \text{if } 2 \leq j \leq N; \end{array} \tag{2}$$

$$w_{j0}^H = \{ \begin{array}{ll} x_{02}, & \text{if } j = 1; \\ \frac{x_{01}\mathbf{W}^{\mathrm{T}}\mathbf{X}^j - x_{02}\mathbf{W}^{\mathrm{T}}\mathbf{X}^{j-1}}{\mathbf{W}^{\mathrm{T}}\mathbf{X}^j - \mathbf{W}^{\mathrm{T}}\mathbf{X}^{j-1}}, & \text{if } 2 \leq j \leq N; \end{array} \tag{3}$$

Let a_j^c be the jth *activation value* for the cth input, i.e., the output of the jth hidden node for input $\mathbf{x}^c$. Then $a_1^c \equiv g(x_{02})$ and $a_j^c \equiv g(\frac{x_{02}-x_{01}}{\mathbf{W}^{\mathrm{T}}\mathbf{X}^j - \mathbf{W}^{\mathrm{T}}\mathbf{X}^{j-1}}\mathbf{w}^{\mathrm{T}}\mathbf{x}^c + \frac{x_{01}\mathbf{W}^{\mathrm{T}}\mathbf{X}^j - x_{02}\mathbf{W}^{\mathrm{T}}\mathbf{X}^{j-1}}{\mathbf{W}^{\mathrm{T}}\mathbf{X}^j - \mathbf{W}^{\mathrm{T}}\mathbf{X}^{j-1}})$ $\forall\ 2 \leq j \leq N$. Let $\mathbf{a}^c \equiv (g(x_{02}), a_2^c, \ldots, a_N^c)^{\mathrm{T}}$ and $\mathbf{M} \equiv (\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^N)^{\mathrm{T}}$. [5] showed that the N samples in $\{\mathbf{x}\}$ space are mapped to N distinctive points in the activation space such that the $N \times N$ matrix $\mathbf{M}$ is invertible. With w_0^O set to zero, $\mathbf{w}^O = \mathbf{M}^{-1}\mathbf{T}$ can always be found to match $w_0^O + \sum_{j=1}^N w_j^O a_j^c$ to t^c without any error.

[5] ends up at the construction method with neither discussion on the characteristic of the adopted SLFN nor on the data relationship among input vectors and target values. For instance, the output value y of the constructed SLFN for an arbitrary input $\mathbf{x}$ can be represented as $w_1^O g(x_{02}) + \sum_{j=2}^N w_j^O g(w_{j0}^H + \sum_{i=1}^I w_{j0}^H x_i)$, since a_1 always equals $g(x_{02})$. Thus $w_1^O g(x_{02})$ can serve as the bias of the output node such that there are only $N-1$ effective hidden nodes. Furthermore, from eqt. (3), vectors $\mathbf{w}_j^H$ for $2 \leq j \leq N$ are linearly dependent. Therefore, the constructed SLFN has the weight vectors (from the input layer) of all its (effective) hidden nodes linearly dependent on each other.

3 Characteristics of SLFN

We apply the following preimage analysis to explore characteristics of the SLFN. Without any loss of generality, assume the *tanh* activation function is adopted in all hidden nodes.

Denote a particular collection of w_{j0}^H, $\mathbf{w}_j^H$, $\mathbf{w}^O$, and w_0^O by Θ. Given Θ, the mapping f of SLFN is the composite of the following mappings: the *activation mapping* $\Phi_A : \Re^I \to (-1,1)^J$ that maps an input $\mathbf{x}$ to an activation value $\mathbf{a}$ (i.e., $\mathbf{a} = \Phi_A(\mathbf{x})$); and the *output mapping* $\Phi_O : (-1,1)^J \to (w_0^O - \sum_{j=1}^J |w_j^O|, w_0^O + \sum_{j=1}^J |w_j^O|)$ that maps an activation value $\mathbf{a}$ to an output y (i.e., $y = \Phi_O(\mathbf{a})$). Note that, since the range of Φ_A and the domain of Φ_O are set as $(-1,1)^J$, the range in the output space $\Im \equiv (w_0^O - \sum_{j=1}^J |w_j^O|, w_0^O + |\sum_{j=1}^J w_j^O|)$ contains all achievable output values. For ease of reference in later discussion, we also call $\Re^I$ the *input space* and $(-1,1)^J$ the *activation space*.

Thus, $f^{-1}(y) \equiv \Phi_A^{-1} \circ \Phi_O^{-1}(y)$, with

$$\Phi_O^{-1}(y) \equiv \{\mathbf{a} \in (-1,1)^J | \sum_{j=1}^J w_j^O a_j = y - w_0^O\}, \tag{4}$$

$$\Phi_A^{-1}(\mathbf{a}) \equiv \cap_{j=1}^J \{\mathbf{x} \in \Re^I | \sum_{i=1}^I w_{ji}^H x_i = tanh^{-1}(a_j) - w_{j0}^H\}, \tag{5}$$

where $tanh^{-1}(x) = 0.5 \ln(\frac{1+x}{1-x})$. Formally, the followings are defined for every given Θ:

(a) A value $y \in \Re$ is *void* if $y \neq f(\{\Re^I\})$, i.e., for all $\mathbf{x} \in \Re^I$, $f(\mathbf{x}) \neq y$. Otherwise, y is *non-void*.
(b) A point $\mathbf{a} \in (-1,1)^J$ is *void* if $\mathbf{a} \notin \Phi_A(\Re^I)$, i.e., for all $\mathbf{x} \in \Re^I, \Phi_A(\mathbf{x}) \neq \mathbf{a}$. Otherwise, $\mathbf{a}$ is *non-void*. The set of all non-void $\mathbf{a}$'s in the activation space is named as the *non-void set*.
(c) The *image* of an input $\mathbf{x} \in \Re^I$ is $y \equiv f(\mathbf{x})$ for $y \in \Im$.
(d) The *preimage* of a non-void output value y is $f^{-1}(y) \equiv \{\mathbf{x} \in \Re^I | f(\mathbf{x}) = y\}$. The preimage of a void value y is the empty set.
(e) The *internal-preimage* of a non-void output value y is the collection $\{\mathbf{a} \in (-1,1)^J | \Phi_O(\mathbf{a}) = y\}$ on the activation space.

From eqt. (4), with the given Θ, $\Phi_O^{-1}(y)$ is the linear equation $\sum_{j=1}^J w_j^O a_j = y - w_0^O$, which is a hyperplane in the activation space. As y changes, $\Phi_O^{-1}(y)$ forms parallel hyperplanes in the activation space; for any change of the same magnitude in y, the corresponding hyperplanes are spaced by the same distance. The activation space is entirely covered by these parallel $\Phi_O^{-1}(y)$ hyperplanes, orderly in terms of the values of y. These parallel hyperplanes form a (linear) scalar field [14], that is, for each point $\mathbf{a}$ of the activation space, there is only one output value y whose $\Phi_O^{-1}(y)$ hyperplane passes point $\mathbf{a}$; all points on the same (internal preimage) hyperplane yield the same y value.

From eqt. (5), $\Phi_A^{-1}(\mathbf{a})$ is a separable function such that each of its components lies along a dimension of the activation space. Moreover, $\Phi_{Aj}^{-1}(a_j) \equiv \{\mathbf{x} \in \Re^I | \sum_{i=1}^I w_{ji}^H x_i = tanh^{-1}(a_j) - w_{j0}^H\}$ is a monotone bijection that defines a one-to-one mapping between the activation value a_j and the input $\mathbf{x}$. For each a_j value, $\Phi_{Aj}^{-1}(a_j)$ defines an activation hyperplane in the input space. Activation hyperplanes associated with all possible a_j values are parallel and form a (linear) scalar activation field in the input space. That is, for each point $\mathbf{x}$ of the input space, there is only one activation value a_j whose $\Phi_{Aj}^{-1}(a_j)$ hyperplane passes point $\mathbf{x}$; all points on the $\Phi_{Aj}^{-1}(a_j)$ hyperplane are associated with the activation value a_j. Each hidden node gives rise to an activation field, and J hidden nodes set up J independent activation fields in the input space. Thus, with a given Θ, the preimage of an activation value $\mathbf{a}$ by Φ_A^{-1} is the intersection of J specific hyperplanes.

The intersection $\cap_{j=1}^J \{\mathbf{x} \in \Re^I | \sum_{i=1}^I w_{ji}^H x_i = tanh^{-1}(a_j) - w_{j0}^H\}$ can be represented as $\{\mathbf{x} | \mathbf{W}^H \mathbf{x} = \boldsymbol{\omega}(\mathbf{a})\}$, where $\omega_j(a_j) \equiv tanh^{-1}(a_j) - w_{j0}^H$ for all $1 \le j \le J$, and $\boldsymbol{\omega}(\mathbf{a}) \equiv (\omega_1(a_1), \omega_2(a_2), \ldots, \omega_J(a_J))^{\mathrm{T}}$. Given Θ and an arbitrary point $\mathbf{a}$, $\boldsymbol{\omega}(\mathbf{a})$ is simply a J-dimensional vector of known component values; the conditions that relates $\mathbf{a}$ with $\mathbf{x}$ can be represented as

$$\mathbf{W}^H \mathbf{x} = \boldsymbol{\omega}(\mathbf{a}), \tag{6}$$

which is a system of J simultaneous linear equations with I unknowns.

Let $rank(\mathbf{D})$ be the rank of matrix $\mathbf{D}$ and $(\mathbf{D}_1 \vdots \mathbf{D}_2)$ be the augmented matrix of two matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ (with the same number of rows). $\mathbf{W}^H \mathbf{x} = \boldsymbol{\omega}(\mathbf{a})$ is a set of inconsistent simultaneous equations if $rank(\mathbf{W}^H \vdots \boldsymbol{\omega}(\mathbf{a})) = rank(\mathbf{W}^H) + 1$ (c.f. [11]). In this case, the corresponding point $\mathbf{a}$ is void. Otherwise, $\mathbf{a}$ is non-void. Note that, for a non-void $\mathbf{a}$, the solution of eqt. (6) defines an affine space of dimension $I - rank(\mathbf{W}^H)$ in the input space. The discussion establishes Lemma 1 below.

Lemma 1. (a) An activation point $\mathbf{a}$ in the activation space is non-void if its corresponding $rank(\mathbf{W}^H \vdots \boldsymbol{\omega}(\mathbf{a}))$ equals $rank(\mathbf{W}^H)$. (b) The set of input values $\mathbf{x}$ mapped onto a non-void $\mathbf{a}$ forms an affine space of dimension $I - rank(\mathbf{W}^H)$ in the input space.

By definition, the non-void set equals $\{\mathbf{a} \in (-1,1)^J | a_j = tanh(\sum_{i=1}^I w_{ji}^H x_i + w_{j0}^H)$ for $1 \le j \le J, \mathbf{x} \in \Re^I\}$. Check that $\mathbf{W}^H$is a $J \times I$ matrix. If $rank(\mathbf{W}^H) = J$, Lemma 1 says that no activation point $\mathbf{a}$ can be void and leads to Lemma 2 below. For $rank(\mathbf{W}^H) < J$, Lemma 3 characterizes the non-void set, which requires the concept of manifold. A p-manifold is a Hausdorff space $\mathbf{X}$ with a countable basis such that each point x of $\mathbf{X}$ has a neighborhood that is homomorphic with an open subset of $\Re^p$ [10]. A 1-manifold is often called a curve, and a 2-manifold is called a surface. For our purpose, it suffices to consider Euclidean spaces, the most common members of the family of Hausdorff spaces.

Lemma 2. If $rank(\mathbf{W}^H)$ equals J, then the non-void set covers the entire activation space.

Lemma 3. If $rank(\mathbf{W}^H)$ is less than J, then the non-void set in the activation space is a $rank(\mathbf{W}^H)$-manifold.

$\mathbf{A}(y)$, the intersection of $\Phi_O^{-1}(y)$ and the non-void set in the activation space, is the internal-preimage of y. Mathematically, for each non-void y, $\mathbf{A}(y) \equiv \{\mathbf{a}|rank(\mathbf{W}^H \vdots \boldsymbol{\omega}(\mathbf{a})) = rank(\mathbf{W}^H), \mathbf{a} \in \Phi_O^{-1}(y)\}$. Consider first $rank(\mathbf{W}^H) = J$. In this case, Lemma 2 says that the non-void set is the entire activation space. Thus, $\mathbf{A}(y)$ equals $\Phi_O^{-1}(y)$. If $rank(\mathbf{W}^H) < J$, then $\mathbf{A}(y)$ is a subset of $\Phi_O^{-1}(y)$. Thus, we have the following Lemma 4. Furthermore, $\mathbf{A}(y)$'s are aligned orderly according to $\Phi_O^{-1}(y)$ and all non-empty $\mathbf{A}(y)$'s form an *internal-preimage field* in the activation space. That is, there is one and only one y such that a non-void $\mathbf{a} \in \mathbf{A}(y)$; and for any $\mathbf{a}$ on $\mathbf{A}(y)$, its output value is equal to y.

Lemma 4. For each non-void output value y, all points in the set $\mathbf{A}(y)$ are at the same hyperplane.

Now the preimage of any non-void output value y, $f^{-1}(y)$, equals $\{\mathbf{x} \in \Re^I | \mathbf{W}^H\mathbf{x} = \boldsymbol{\omega}(\mathbf{a})$ with all $\mathbf{a} \in \mathbf{A}(y)\}$. If $rank(\mathbf{W}^H) = J$, then, from Lemma 2 and Lemma 1(b), the preimage $f^{-1}(y)$ is a $(I-1)$-manifold in the input space. For $rank(\mathbf{W}^H) < J$, from Lemma 3 and Lemma 1(b),

1. if $rank(\mathbf{W}^H) = 1$ and $\mathbf{A}(y)$ is a single point, then $f^{-1}(y)$ is a single hyperplane;
2. if $rank(\mathbf{W}^H) = 1$ and $\mathbf{A}(y)$ consists of several points, then $f^{-1}(y)$ may consist of several disjoint hyperplanes;
3. if $1 < rank(\mathbf{W}^H) < J$ and $\mathbf{A}(y)$ is a single ($rank(\mathbf{W}^H)$-1)-manifold, then $f^{-1}(y)$ is a single $(I-1)$-manifold; and
4. if $1 < rank(\mathbf{W}^H) < J$ and $\mathbf{A}(y)$ consists of several disjoint ($rank(\mathbf{W}^H)$-1)-manifolds, then $f^{-1}(y)$ consists of several disjoint $(I-1)$-manifolds.

Table 1 summarizes that the preimage $f^{-1}(y)$ is dictated by the property of its associated internal-preimage $\mathbf{A}(y)$.

Table 1. The relationship between the internal-preimage $\mathbf{A}(y)$ and the preimage $f^{-1}(y)$ of a non-void output value y

The nature of $\mathbf{A}(y)$	The nature of $f^{-1}(y)$
A single intersection-segment	A single $(I-1)$-manifold
Multiple disjoint intersection-segments	Multiple disjoint $(I-1)$-manifolds

The input space is entirely covered by a grouping of preimage manifolds that forms a *preimage field*. That is, there is one and only one preimage manifold passing through each $\mathbf{x}$; and the corresponding output value is the y value associated with this preimage manifold. Note that the preimage manifolds are aligned orderly because $\mathbf{A}(y)$'s are aligned orderly according to $\Phi_O^{-1}(y)$'s and the mapping of Φ_A^{-1} is a monotone bijection that defines a one-to-one mapping between an activation vector and an affine space.

Notice that $rank(\mathbf{W}^H)$ determines the characteristic of the non-void set and thus the characteristic of internal-preimage. Hereafter, SLFN-p denotes a SLFN whose $rank(\mathbf{W}^H)$ equals p. For instance, if we adopt a SLFN-1 network, then we can assume $\mathbf{w}_j^H$ equals $\alpha_j \mathbf{w}$ with $\alpha_j \neq 0$ for all j and $\alpha_{j_1} \neq \alpha_{j_2}$ for all $j_1 \neq j_2$. Since α_1 is non-zero, a_j can be represented as $tanh(\delta_j tanh^{-1}(a_1) + \delta_{j0})$, where $\delta_j = \alpha_j/\alpha_1$ and $\delta_{j0} = (\alpha_1 w_{j0}^H - \alpha_j w_{10}^H)/\alpha_1$. If we adopt a SLFN-2 network, then we can assume that $\mathbf{w}_1^H$ equals $\mathbf{w}_1$, $\mathbf{w}_2^H$ equals $\mathbf{w}_2, \mathbf{w}_j^H = \gamma_{j1}\mathbf{w}_1 + \gamma_{j2}\mathbf{w}_2$ with either γ_{j1} or γ_{j2} nonzero for all $j \geq 3$, and $\mathbf{w}_1$ and $\mathbf{w}_2$ are linearly independent.

For SLFN-1, the above preimage analysis states that the non-void set is an 1-manifold; $\mathbf{A}(y)$ equals $\{\mathbf{a} \in (-1,1)^J | w_1^O a_1 + \sum_{j=2}^J w_j^O tanh(\delta_j tanh^{-1}(a_1) + \delta_{j0}) = y - w_0^O, a_j = tanh(\delta_j tanh^{-1}(a_1) + \delta_{j0}) \ \forall\ j \geq 2,\ a_1 \in (-1,1)\}$; and $f^{-1}(y)$ equals $\{\mathbf{x} \in \Re^I | \sum_{i=1}^I w_{1i}^H x_i = tanh^{-1}(a_1) - w_{10}^H, w_1^O a_1 + \sum_{j=2}^J w_j^o tanh(\delta_j tanh^{-1}(a_1) + \delta_{j0}) = y - w_0^O,\ a_j = tanh(\delta_j tanh^{-1}(a_1) + \delta_{j0}) \ \forall\ j \geq 2,\ a_1 \in (-1,1)\}$. These establish the following Lemma 5. Furthermore, $\mathbf{w}$ is the normal vector of the preimage hyperplane of SLFN-1 and $\mathbf{w}_j^H \equiv \alpha_j \mathbf{w}$ determines the orientation of the activation hyperplane in the input space corresponding to the jth hidden node. Thus, we have Lemma 6.

Lemma 5. For SLFN-1, the preimage field is formed from a collection of preimage hyperplanes.

Lemma 6. For SLFN-1, the activation hyperplanes in the input space corresponding to all hidden nodes are parallel, and the preimage hyperplane is parallel with the activation hyperplane.

The above preimage analysis results in the following two hyperplane characteristics (i) and (ii) regarding all SLFN networks and one hyperplane characteristic (iii) for SLFN-1: (i) training samples with the same target value are allowed to be on the same activation hyperplane; (ii) activation points with the same target value are allowed to be on the same Φ_O^{-1} hyperplane; (iii) training samples with the same target value are allowed to be on the same preimage hyperplane.

4 The Experiment of m-Bit Parity Learning Problem

Through the application to the m-bit parity problem learned by constructing SLFN-2 and SLFN-1 network solutions, we show that the construction concept can help find network solutions perfectly fitting 2^m distinct samples with fewer used hidden nodes than the one obtained from the conventional construction method of [5].

For the m-bit parity problem, I is equal to m and we observe that the 2^m input samples are on the vertices of an m-dimensional hypercube with any two adjacent vertices having different target values. Without any loss of generality, take $x_i^c \in \{-1, 1\}$ for all c and i, and set the target value to t for odd number of $+1$'s in input, and to $-t$ otherwise.

In the case of constructing a SLFN-2 network solution, assume that $\mathbf{w}_1^H$ equals $\mathbf{w}_1, \mathbf{w}_2^H$ equals $\mathbf{w}_2$, $\mathbf{w}_j^H \equiv \gamma_{j1}\mathbf{w}_1 + \gamma_{j2}\mathbf{w}_2$ with either γ_{j1} or γ_{j2} nonzero for

Table 2. The mapping from the total eight input patterns to the following six activation vectors

$\mathbf{x}^c$	$\mathbf{w}_1^{\mathrm{T}}\ \mathbf{x}^c$	$\mathbf{w}_2^{\mathrm{T}}\mathbf{x}^c$	$\mathbf{w}_j^{\mathrm{T}}\mathbf{x}^c$	t^c
(1, 1, 1)	3	1	$3\gamma_{j1}+\gamma_{j2}$	$-t$
(-1, 1, 1)	1	3	$\gamma_{j1}+3\gamma_{j2}$	t
(1, -1, 1)	1	-1	$\gamma_{j1}-\gamma_{j2}$	t
(1, 1, -1)	1	-1	$\gamma_{j1}-\gamma_{j2}$	t
(-1, -1, 1)	-1	1	$-\gamma_{j1}+\gamma_{j2}$	$-t$
(-1, 1, -1)	-1	1	$-\gamma_{j1}+\gamma_{j2}$	$-t$
(1, -1, -1)	-1	-3	$-\gamma_{j1}-3\gamma_{j2}$	$-t$
(-1, -1, -1)	-3	-1	$-3\gamma_{j1}-\gamma_{j2}$	t

all $j \geq 3$, and $\mathbf{w}_1$ and $\mathbf{w}_2$ are linearly independent. Following the guide of the observed property of parity problem and the hyperplane characteristic (i), let $\mathbf{w}_1$ and $\mathbf{w}_2$ be assigned as in eqt. (6) and thus the total 2^m input patterns are mapped onto $2m$ activation vectors $\{\tilde{\mathbf{a}}^0, \ldots, \tilde{\mathbf{a}}^{2m-1}\}$, in which $\tilde{\mathbf{a}}^k \equiv (\tilde{a}_1^k, \ldots, \tilde{a}_J^k)^{\mathrm{T}}$. Table 2 illustrates the mapping regarding the 3-bit parity problem. Let $\tilde{a}_1^0 = tanh(m + w_{10}^H), \tilde{a}_2^0 = tanh(m-2+w_{20}^H), \tilde{a}_j^0 = tanh(m\gamma_{j1} + (m-2)\gamma_{j2} + w_{j0}^H), j = 3, \ldots, J$, and $\tilde{t}^0 \equiv -t$; for each $k = 1, \ldots, m-1, \tilde{a}_1^{2k-1} = tanh(m-2k+w_{10}^H), \tilde{a}_2^{2k-1} = tanh(m-2k+2+w_{20}^H), \tilde{a}_j^{2k-1} = tanh((m-2k)\gamma_{j1} + (m-2k+2)\gamma_{j2} + w_{j0}^H), j = 3, \ldots, J$, and $\tilde{t}^{2k-1} \equiv (-1)^{k+1}t; \tilde{a}_1^{2k} = tanh(m-2k+w_{10}^H), \tilde{a}_2^{2k} = tanh(m-2k-2+w_{20}^H), \tilde{a}_j^{2k} = tanh((m-2k)\gamma_{j1} + (m-2k-2)\gamma_{j2} + w_{j0}^H), j = 3, \ldots, J$, and $\tilde{t}^{2k} \equiv (-1)^{k+1}t; \tilde{a}_1^{2m-1} = tanh(-m+w_{10}^H), \tilde{a}_2^{2m-1} = tanh(-m+2+w_{20}^H), \tilde{a}_j^{2m-1} = tanh(-m\gamma_{j1} + (-m+2)\gamma_{j2} + w_{j0}^H), j = 3, \ldots, J$, and $\tilde{t}^{2m-1} \equiv (-1)^{m+1}t$.

$$w_{1j} = 1, i = 1, \ldots, m; \qquad w_{21} = -1, w_{2i} = 1, i = 2, \ldots, m. \tag{7}$$

Thus let $J = 2m, w_{j0}^H = 0\ \forall\ j, w_0^O = 0$, and $\mathbf{w}^O = \widetilde{\mathbf{M}}^{-1}\widetilde{\mathbf{T}}$, in which $\widetilde{\mathbf{M}} \equiv (\tilde{\mathbf{a}}^0, \ldots, \tilde{\mathbf{a}}^{2m-1})^{\mathrm{T}}$ and $\widetilde{\mathbf{T}} \equiv (\tilde{t}^0, \tilde{t}^1, \ldots, \tilde{t}^{2m-1})^{\mathrm{T}}$. Referring to [5], it is trivial to show that there exist non-zero values of γ_{j1} and γ_{j2} such that the square matrix $\widetilde{\mathbf{M}}$ is invertible and thus the corresponding inverse matrix $\widetilde{\mathbf{M}}^{-1}$ exists. By checking all 2^m samples, it is trivial to show that the above SLFN-2 network is a solution of the m-bit parity problem.

In the case of constructing a SLFN-1 network solution, assume that $\mathbf{w}_j^H$ equals $\alpha_j\mathbf{w}$ with $\alpha_j \neq 0$ for all $j, \alpha_{j_1} \neq \alpha_{j_2}$ for all $j_1 \neq j_2$, and a_j can be represented as $tanh(\delta_j tanh^{-1}(a_1) + \delta_{j0})$, where $\delta_j = \alpha_j/\alpha_1$ and $\delta_{j0} = (\alpha_1 w_{j0}^H - \alpha_j w_{10}^H)/\alpha_1$. Following the guide of the observed property of parity problem and the hyperplane characteristics (ii) and (iii), we pick $\mathbf{w}$ as $\mathbf{w}_1$ in eqt. (6) and assign a total of $\lceil (m+1)/2 \rceil$ adopted hidden nodes. Thus the total 2^m input patterns are mapped onto $m+1$ activation vectors, $\{\hat{\mathbf{a}}^0, \ldots, \hat{\mathbf{a}}^m\}$, in which $\hat{a}_j^k = tanh((m-2k)\alpha_j + w_{j0}^H), j = 1, \ldots, \lceil (m+1)/2 \rceil$ and $\hat{\mathbf{a}}^k \equiv (\hat{a}_1^k, \ldots, \hat{a}_{\lceil (m+1)/2 \rceil}^k)^{\mathrm{T}}, k = 0, \ldots, m$. Then we make the following assignments:

(I) when m is an odd number: let $w_{j0}^H = 0 \ \forall \ j, w_0^O = 0$ and $\mathbf{w}^O = \widehat{\mathbf{M}}^{-1}\widehat{\mathbf{T}}$, where $\widehat{\mathbf{M}} \equiv (\hat{\mathbf{a}}^0, \hat{\mathbf{a}}^1, \ldots, \hat{\mathbf{a}}^{\lceil (m+1)/2 \rceil - 1})^{\mathrm{T}}, \hat{a}_j^k = tanh((m-2k)\alpha_j) \ \forall \ j, k = 0, \ldots, m$, $\widehat{\mathbf{T}} \equiv (\hat{t}^0, \hat{t}^1, \ldots, \hat{t}^{\lceil (m+1)/2 \rceil - 1})^{\mathrm{T}}$, and $\hat{t}^k \equiv (-1)^{k+1} t$ for all k.

(II) when m is an even number: let $w_{j0}^H = \alpha_j \ \forall \ j, w_0^O = 0$ and $\mathbf{w}^O = \widehat{\mathbf{M}}^{-1}\widehat{\mathbf{T}}$, where $\widehat{\mathbf{M}} \equiv (\hat{\mathbf{a}}^0, \hat{\mathbf{a}}^1, \ldots, \hat{\mathbf{a}}^{\lceil (m+1)/2 \rceil - 1})^{\mathrm{T}}, \hat{a}_j^k = tanh((m-2k+1)\alpha_j) \ \forall \ j, k = 0, \ldots, m$, $\widehat{\mathbf{T}} \equiv (\hat{t}^0, \hat{t}^1, \ldots, \hat{t}^{\lceil (m+1)/2 \rceil - 1})^{\mathrm{T}}$, and $\hat{t}^k \equiv (-1)^{k+1} t$ for all k.

Referring to [5], it is trivial to show that there exist non-zero values of α_j such that the set $\{\hat{\mathbf{a}}^0, \ldots, \hat{\mathbf{a}}^m\}$ are linearly independent and thus the square matrix $\widehat{\mathbf{M}}$ is invertible. By checking all 2^m samples, it is trivial to show that the above SLFN-1 network is a solution of the m-bit parity problem.

5 Conclusions and Future Work

This study derives three hyperplane characteristics and several properties of the SLFN through the preimage analysis. Regarding the m-bit parity learning problem, we observe that the 2^m input samples are on the vertices of an m-dimensional hypercube with any two adjacent vertices having different target values. Accordingly, the construction concept helps set up SLFN-1 and SLFN-2 solutions, each of which uses fewer hidden nodes than the ones used by the conventional construction method of [5].

Note that most learning algorithms (or construction methods) lead to a SLFN-p solution with $p \geq 2$ and complex preimages. Extending from this study, one may further argue that the construction concept can help identify the *true* upper bound on the number of used hidden nodes of such SLFN solutions for a specific learning problem. This argument is one of future researches. Most training is a trade-off of learning performance and generalization performance, and the hidden layer plays a key role in this issue. Therefore, another future research is to develop and validate a construction concept that involves with the generalization.

Acknowledgments. This study is supported by the National Science Council of the R.O.C. under Grants NSC 92-2416-H-004-004, NSC 93-2416-H-004-015, and NSC 43028F.

References

1. Arai, M.: Bounds on the number of hidden units in binary-valued three-layer neural networks. Neural Networks 6, 855–860 (1993)
2. Arslanov, M.Z., Ashigaliev, D.U., Ismail, E.E.: N-bit parity ordered neural networks. Neurocomputing 48, 1053–1056 (2002)
3. Hertz, J., Krogh, A., Palmer, R.: Introduction to the Theory of Neural Computation. Addison-Wesley Publishing Company, Redwood City (1991)

4. Hohil, M.E., Liu, D.R., Smith, S.H.: Solving the N-bit parity problem using neural networks. Neural Networks 12(11), 1321–1323 (1999)
5. Huang, G., Babri, H.: Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. IEEE Transactions on Neural Networks 9, 224–229 (1998)
6. Huang, S.C., Huang, Y.F.: Bounds on the number of hidden neurons in multilayer perceptrons. IEEE Transactions on Neural Networks 2, 47–55 (1991)
7. Iyoda, E.M., Nobuhara, H., Hirota, K.: A solution for the N-bit parity problem using a single translated multiplicative neuron. Neural Processing Letters 18(3), 213–218 (2003)
8. Lavretsky, E.: On the exact solution of the Parity-N problem using ordered neural networks. Neural Networks 13(8), 643–649 (2000)
9. Liu, D.R., Hohil, M.E., Smith, S.H.: N-bit parity neural networks: new solutions based on linear programming. Neurocomputing 48, 477–488 (2002)
10. Munkres, J.: Topology: a first course. Prentice-Hall, Englewood Cliffs (1975)
11. Murty, K.: Linear Programming. John Wiley & Sons, NY (1983)
12. Setiono, R.: On the solution of the parity problem by a single hidden layer feedforward neural network. Neurocomputing 16, 225–235 (1997)
13. Sontag, E.: Feedforward nets for interpolation and classification. J. Comput. System Sci. 45, 20–48 (1992)
14. Tsaih, R.: An explanation of reasoning neural networks. Mathematical and Computer Modelling 28, 37–44 (1998)
15. Urcid, G., Ritter, G.X., Iancu, L.: Single layer morphological Perceptron solution to the N-bit parity problem. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) CIARP 2004. LNCS, vol. 3287, pp. 171–178. Springer, Heidelberg (2004)