行政院國家科學委員會專題研究計畫 成果報告

利用 preimage 分析萃取規則之實作
研究成果報告(精簡版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 97-2410-H-004-117-
執 行 期 間 ：97 年 08 月 01 日至 98 年 07 月 31 日
執 行 單 位 ：國立政治大學資訊管理學系

計 畫 主 持 人 ：蔡瑞煌

計畫參與人員： 學士級-專任助理人員：沈軒豪

報 告 附 件 ：出席國際會議研究心得報告及發表論文

處 理 方 式 ：本計畫可公開查詢

中 華 民 國 98 年 07 月 17 日

行政院國家科學委員會補助專題研究計畫 ■ 成 果 報 告
□期中進度報告

利用 preimage 分析萃取規則之實作

計畫類別：■ 個別型計畫　　□ 整合型計畫
計畫編號：NSC 97－2410－H－004－017－
執行期間：　97 年 8 月 1 日至 98 年 7 月 31 日

計畫主持人：蔡瑞煌
共同主持人：
計畫參與人員： 沈軒豪

成果報告類型(依經費核定清單規定繳交)：■精簡報告　□完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
■出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立政治大學資訊管理學系
中　華　民　國　98　年　7　月　15　日

報告內容

ARTIFICIAL NEURAL NETWORKS AS A FEATURE DISCOVERY TOOL

This study proposes the *preimage analysis* and its associated *belief justification* process regarding the application of continuous-valued single-hidden layer feed-forward neural network (SLFN) to discovering features – certain relationships between explanatory (input) and observed (output) variables – embedded in the (training) data. The preimage analysis explicitly specifies the *preimage* of network and discloses its preimage-related properties. The preimage of a given output of an SLFN is the collection of all inputs each of that generates the output.

The seminal publication of (Rumelhart & McClelland, 1986) states that Artificial Neural Networks (ANN) can be trained primarily through examples; ANN can do the general pattern-recognition; and ANN can learn general rules of optimal behavior. Since then, these claims stimulate studies in many fields to develop various ANNs as modeling tools to check the validity of the claims. Lots of experimental results are positive. For instance, Sgroi & Zizzo (2007) state that ANNs "are consistent with observed laboratory play in two very important senses. Firstly, they select a rule for behavior which appears very similar to that used by laboratory subjects. Secondly, using this rule they perform optimally only approximately 60% of the time." (page 717)

Results of these researches infer that well-trained ANNs possess features embedded in (training) data. Thus, some practitioners may go one step further to conduct researches that address the issue of extracting features embedded in data from well-trained ANNs. For instance, based upon the empirical data, one wants to identify the relative influence of factors for pricing newly issued securities. The practitioner first gets several well-trained networks and then from these networks extracts certain features. The rule (1) is one of such feature examples. Hopefully, the extracted rules could depict features embedded in data and could help identify the significant factor.

Rule: If the input sample is in some sub-region of the input space, then the predicted price value is given by a corresponding linear regression equation. (1)

Note that the feature-discovery practitioner (and researcher) may be merely interested in any (ANN or statistical) tool that can help analyze the data to discover something interesting or significant, instead of in the interaction of human cerebral activities and its explanation that results in the fundamentals of ANN. Regardless, the task faced by a feature-discovery practitioner is not easier because ANN simply behaves as a *black box*, i.e., a system that produces "certain outputs from certain inputs without explaining why or how" (Rabuñal, Dorado, Pazos, Pereira, & Rivero, 2004). Nevertheless, ANN researchers have spent substantial efforts leading to a huge amount of works that explore various "mechanisms, procedures, and algorithms designed to insert knowledge into ANNs (knowledge initialization), extract rules

from trained ANNs (rule extraction), and utilize ANNs to refine existing rule bases (rule refinement)." (Andrews, Diederich, & Tickle, 1995, page 373) Some, but not exhaustive, studies for these purposes are presented in (Andrews, Diederich, & Tickle, 1995; Setiono & Liu, 1997; Tickle, Andrews, Golea & Diederich, 1998; Tsaih, Hsu, & Lai, 1998; Taha & Ghosh, 1999; Zhou, Chen, & Chen, 2000; Saito & Nakano, 2002; Setiono, Leow, & Zurada, 2002; Baesens, Setiono, Mues, & Vanthienen, 2003)

Despite these studies, the black box nature of ANN persists and any feature-discovery intention is better to cope with following issues. First, the approach should disclose *true* properties from ANN. Most of above studies do not seem to generalize to the situation since they are contrived to explore restrictedly with limited data such that the extracted rules are dubious and unlikely to be considered as patterns of knowledge embedded in data. Consider the rule stated in (1). To spot its premise, most works use either training data or generated data, which the trained network itself yields. Such an approach is data sensitive and requires extensive amount of data to be accurate. The limited number of (training or generated) data instances leads to a suspicion of the generalization of rule (1) in interpolating and extrapolating any unexplored data values.

Second, since features learned by an ANN distribute over the entire network as weight values, the approach should effectively disclose the entire black box in order to observe interesting but unknown features embedded in data. With such an entire disclosure, the practitioner can

simultaneously investigate several kinds of features to get a better understanding about the (training) data. The above studies do not seem to generalize to the situation, since they base upon predefined schemes like rule (1) for extracting rules.

Third, it is possible to have a *sub-perfect* learning result since, in most studies (e.g., financial market tests), it is characteristically more difficult to determine the best architecture of SLFN. Besides, noises in the training samples prevent networks from perfect fittings. The possibility of a sub-perfect learning result leads to a conservative attitude on embracing the obtained preimage-related properties. The practitioner with such understanding has conservatism in the straightforward feature acquisition. The above studies do not seem to cope with such conservatism.

Fourth, the practitioner normally has some personal beliefs when he conducts the feature-discovery experiment. Any such belief, if available, is in the form of tacit knowledge about the relationship between the explanatory and observed variables. However, to draw parallels between the beliefs and the observed features needs professional interpretations due to the tacit nature of the former and the complex nature of the latter. That is, the practitioner has to contrast the similarities and differences between beliefs and observed features and then to infer properly. Even if the beliefs and observed features suggest different views, the practitioner may find a way to arrive at a justified belief embodying the observed features. Such a belief

justification is a kind of knowledge internalization stated by Nonaka & Takeuchi (1995). The above studies do not seem to provide such discussions.

This study addresses these issues. Specifically, for feature discovery via well-trained SLFNs, this study proposes the preimage analysis that explicitly specifies the preimage of network to disclose its preimage-related properties. For complex preimage-related properties, this study then proposes the following belief justification process, in which the practitioner's (prior) beliefs are refined based upon the examination results of preimage-related properties. The beliefs lead to propositions of the experiment. Based upon the propositions, the practitioner first picks up relevant explanatory and observed variables and collects the sample accordingly. Then he trains SLFNs and, after the training, applies the preimage analysis to the selected SLFNs. For each belief, the practitioner inspects relevant preimage-related properties. Such inspection leads to a belief justification process. If there is no such prior belief, the rule (2) and the observed preimage-related properties exclusively make statements about features embedded in data.

Research findings of this study are summarized as follows:

(I) the preimage analysis is not data intensive and the inspected preimage-related properties hold globally for any data point;

(II) the inspected preimage-related properties help explore the understanding of SLFN itself;

(III) the inspected preimage-related properties provide further insights about rule (2) and thus about features embedded in data. In rule (2), **x** is the vector of explanatory variables; y is the network's response; f: $\mathbf{X} \rightarrow \mathbf{Y}$ is the function of the SLFN and $y \equiv f(\mathbf{x})$; y' is a constant; and the preimage f^{-1} is the inverse function of f.

Rule: If ($\mathbf{x} \in$ the $f^{-1}(y')$ region), then ($y = y'$). (2)

(IV) several kinds of features can be simultaneously investigated through inspecting preimage-related properties.

The remainder of this paper is organized as follows. Section II starts with the list of notations used in the study and then gives the proposed preimage analysis. From the preimage analysis, we find that $rank(\mathbf{W}^H)$ determines characteristics of the preimage-related properties, in which $\mathbf{W}^H$ is the matrix of weights between the input variables and the hidden nodes and $rank(\mathbf{D})$ is the rank of matrix **D**. Hereafter, SLFN-p denotes an SLFN whose $rank(\mathbf{W}^H)$ equals p. Section III shows the application of preimage analysis to the two SLFN-1 network solutions of the 3-bit parity problem. Some implications regarding the feature-discovery application via SLFN-1 networks are offered in Section VI. It is readily seen that the preimage-related properties of SLFN-1 are easy to understand. However, learning algorithms adopted in most studies likely result in SLFN-p networks with $p \geq 2$ and these SLFN networks own complex preimages and preimage-related properties. In Section V, the details of the belief justification process is

illustrated through the feature-discovery application to the bond pricing experiment, which releases an SLFN-3 network. Some further discussions and future work are presented at the end.

THE PROPOSED PREIMAGE ANALYSIS

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

I ≡ the amount of input nodes;

J ≡ the amount of hidden nodes;

$\mathbf{x}$ ≡ $(x_1, x_2, \ldots, x_I)^{\mathrm{T}}$: the input vector, in which x_i is the i^{th} input component, with i from 1 to I;

$\mathbf{a}$ ≡ $(a_1, a_2, \ldots, a_J)^{\mathrm{T}}$: the hidden activation vector, in which a_j is the activation value of the j^{th} hidden node, with j from 1 to J;

y ≡ the activation value of the output node and $y = f(\mathbf{x})$ with f being the function mapping $\mathbf{x}$ to y;

w_{ji}^{H} ≡ the weight between the i^{th} input variable and the j^{th} hidden node, in which the superscript H throughout the paper refers to quantities related to the hidden layer;

$\mathbf{w}_j^H$ ≡ $(w_{j1}^H, w_{j2}^H, \ldots, w_{jI}^H)^{\mathrm{T}}$;

$\mathbf{W}^H$ ≡ $(\mathbf{w}_1^H, \mathbf{w}_2^H, \ldots, \mathbf{w}_J^H)^{\mathrm{T}}$, the $J \times I$ matrix of weights between the input variables and the hidden nodes;

w_{j0}^{H} ≡ the bias value of the j^{th} hidden node;

$\mathbf{w}_j^O$ ≡ the weight between the j^{th} hidden node and the output node, in which the superscript *O* throughout the paper refers to quantities related to the output layer;

$\mathbf{w}^O$ ≡ $(w_1^O, w_2^O, \ldots, w_J^O)^{\mathrm{T}}$; and

w_0^O ≡ the bias value of the output node.

Without any loss of generality, assume the *tanh* activation function is adopted in all hidden nodes. Denote the collection of w_{j0}^H, $\mathbf{w}_j^H$, $\mathbf{w}^O$, and w_0^O by $\boldsymbol{\theta}$. Given $\boldsymbol{\theta}$, the resulting *f* of SLFN is the composite of the following mappings: the *activation mapping* $\Phi_A: \Re I \rightarrow (\text{-}1, 1)^J$ that maps an input **x** to an activation value **a** (i.e., $\mathbf{a} = \Phi_A(\mathbf{x})$); and the *output mapping* $\Phi_O: (\text{-}1, 1)^J \rightarrow (w_0^O - \sum_{j=1}^{J}|w_j^O|, w_0^O + \sum_{j=1}^{J}|w_j^O|)$ that maps an activation value **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 $(\text{-}1, 1)^J$, 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 R^I the *input space* and $(\text{-}1, 1)^J$ the *activation space*.

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

$$\Phi_O^{-1}(y) \equiv \{\mathbf{a} \in (\text{-}1, 1)^J \mid \sum_{j=1}^{J} w_j^O a_j = y - w_0^O\}, \tag{3}$$

$$\Phi_A^{-1}(\mathbf{a}) \equiv \bigcap_{j=1}^{J}\{\mathbf{x} \in \Re^I \mid \sum_{i=1}^{I} w_{ji}^H x_j = \tanh^{-1}(a_j) - w_{j0}^H\}, \tag{4}$$

where $\tanh^{-1}(x) \equiv 0.5\ln\left(\frac{1+x}{1-x}\right)$. Formally, the followings are defined for every given $\boldsymbol{\theta}$:

(a) A value $y \in \Re$ is *void* if $y \notin f(\{\Re^I\})$, i.e., for all $\mathbf{x} \in \Re^I$, $f(\mathbf{x}) \neq y$. Otherwise, *y* is *non-void*.

(b) A point $\mathbf{a} \in (\text{-}1, 1)^J$ 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, **a** is *non-void*. The set of all non-void **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 intersection of $\Phi_O^{-1}(y)$ and the non-void set on the activation space.

Given θ, the preimage analysis is conducted in the following four steps to specify the preimage:

Step 1: Derive the expression of $\Phi_O^{-1}(y)$;

Step 2: Derive the expression of the non-void set;

Step 3: Derive the expression of internal-preimage of a non-void output value y; and

Step 4: Derive the expression of preimage $f^{-1}(y)$.

From eqt. (3), with the given θ, $\Phi_O^{-1}(y)$ 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 (Tsaih, 1998). That is, for each point **a** of the activation space, there is only one output value y whose $\Phi_O^{-1}(y)$ hyperplane passes point **a**; all points on the same (internal preimage) hyperplane yield the same y value.

From eqt. (4), $\Phi_A^{-1}(\mathbf{a})$ is a separable function such that each of its components lies along a dimension of the activation space. Moreover, $\Phi_{Aj}^{-1}(a_j) \equiv \{\mathbf{x} \in \Re^I | \sum_{i=1}^{I} w_{ji}^H x_j = \tanh^{-1}(a_j) - w_{j0}^H\}$ 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 the given θ, the preimage of an activation value $\mathbf{a}$ by Φ_A^{-1} is the intersection of J specific hyperplanes.

The intersection $\bigcap_{j=1}^{J} \{\mathbf{x} \in \Re^I | \sum_{i=1}^{I} w_{ji}^H x_j = \tanh^{-1}(a_j) - w_{j0}^H\}$ 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 and the conditions that relates $\mathbf{a}$ with $\mathbf{x}$ can be represented as

$$\mathbf{W}^H\mathbf{x} = \boldsymbol{\omega}(\mathbf{a}), \tag{5}$$

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 $rank(\mathbf{W}^H : \boldsymbol{\omega}(\mathbf{a})) = rank(\mathbf{W}^H) + 1$ (c.f. (Murty, 1983)). In this case, the corresponding point **a** is void. Otherwise, **a** is non-void. Note that, for a non-void **a**, the solution of eqt. (5) 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 **a** in the activation space is non-void if its corresponding $rank(\mathbf{W}^H : \boldsymbol{\omega}(\mathbf{a}))$ equals $rank(\mathbf{W}^H)$. (b) The set of input values **x** mapped onto a non-void **a** forms an affine space of dimension $I - rank(\mathbf{W}^H)$ in the input space.

By definition, the non-void set equals $\{\mathbf{a} \in (-1, 1)^J \mid a_j = tanh(\sum_{i=1}^{I} w_{ji}^H x_j + w_{j0}^H) \text{ for } 1 \le j \le J, \mathbf{x} \in \Re I\}$. Check that $\mathbf{W}^H$ is a $J \times I$ matrix. If $rank(\mathbf{W}^H) = J$, Lemma 1 says that no activation point **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 **X** with a countable basis such that each point x of **X** has a neighborhood that is homomorphic with an open subset of $\Re^p$ (Munkres, 1975). 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 the relationship between the internal-preimage A(y) and the preimage $f^{-1}(y)$ of a non-void output value y.

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

The nature of A(y)	A single intersection-segment	Multiple disjoint intersection-segments
The nature of $f^{-1}(y)$	A single (I-1)-manifold	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 **x**; and the corresponding output value is the y value associated with this preimage manifold. Note that the preimage manifolds are aligned orderly because **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. For a SLFN-1 network, we can assume $\mathbf{w}_j^H \equiv \alpha_j \mathbf{w}$ for all j, in which **w** is a non-zero vector and α_js are constants; as for a SLFN-p network with $p > 1$, we can assume that vectors in the set of $\{\mathbf{w}_1^H, \mathbf{w}_2^H, \ldots, \mathbf{w}_p^H\}$ are linearly independent and $\mathbf{w}_j^H \equiv \sum_{k=1}^{P} \gamma_{jk} \mathbf{w}_k^H$ for all $j > p$.

APPLICATION TO SOME EXAMPLES OF SLFN-1 NETWORK

In this section, we show the application of the preimage analysis to the following two kinds of SLFN-1 network solutions of the 3-bit parity problem, in which the target output is 1 if the input vector contains an odd number of -1s and -1 otherwise: (1) the SLFN network solution with seven effective hidden nodes shown in Table 2, constructed by Huang & Babri (1998), and (2) the SLFN network solution with two hidden nodes shown in Table 3.

Table 2. An SLFN network solution of the 3-bit parity problem constructed by Huang & Babri (1998), in which w_0^O = 0.0 and $\mathbf{w} = (0.4, 0.5, 0.7)^T$.

j	w_j^O	w_{j0}^H	$\mathbf{w}_j^H$
1	-239.9515868	0.651132681	0.0 **w**
2	65.96703854	0.283261412	-0.459839086 **w**
3	15.8645681	-0.452481126	-1.839356344 **w**
4	-369.5491494	0.467197046	-0.919678172 **w**
5	465.7997072	0.835068315	-0.919678172 **w**
6	-45.18900519	1.202939584	-0.919678172 **w**
7	-110.3929208	2.122617756	-1.839356344 **w**
8	128.6377379	1.386875218	-0.459839086 **w**

Table 3. An SLFN network solution of the 3-bit parity problem, in which w_0^O = 0.0 and $\mathbf{w} = (1, 1, 1)^T$.

j	w_j^O	w_{j0}^H	$\mathbf{w}_j^H$
1	18.58899737	0.0	0.4 **w**
2	-30.7174688	0.0	0.2 **w**

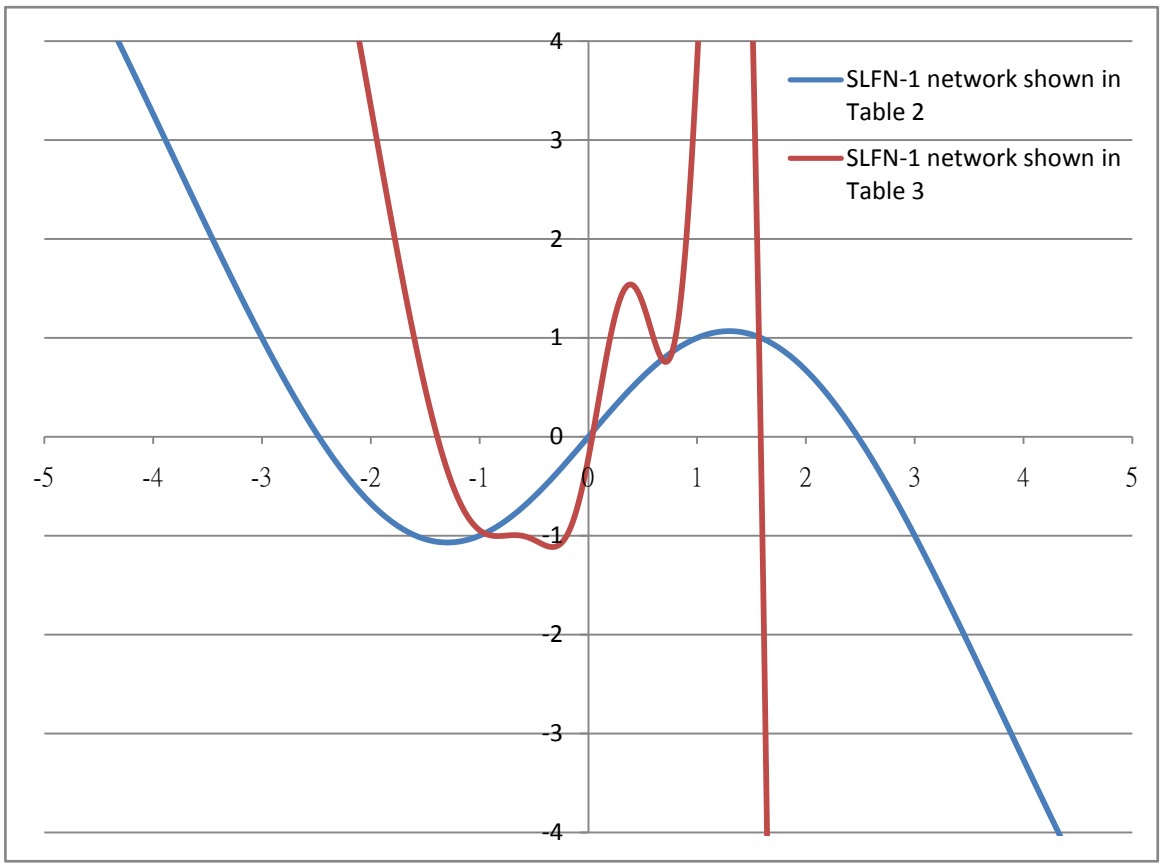
For the SLFN-1 network with $\mu \equiv 0.4x_1+0.5x_2+0.7x_3$ shown in Table 2, the preimage analysis states that its non-void set equals $\{\mathbf{a} \in (-1, 1)^8 | a_1 = \tanh(0.651132681), a_2 = \tanh(0.283261412-0.459839086\mu), a_3 = \tanh(-0.452481126-1.839356344\mu), a_4 = \tanh(0.467197046-0.919678172\mu), a_5 = \tanh(0.835068315-0.919678172\mu), a_6 = \tanh(1.202939584-0.919678172\mu), a_7 = \tanh(2.122617756-1.839356344\mu), a_8 = \tanh(1.386875218-0.459839086\mu), \mu \in \Re\}$, which is an 1-manifold in $(-1, 1)^8$; $\mathbf{A}(y)$ equals $\{\mathbf{a} \in (-1, 1)^8 | a_1 = \tanh(0.651132681), a_2 = \tanh(0.283261412-0.459839086\mu), a_3 = \tanh(-0.452481126-1.839356344\mu), a_4 = \tanh(0.467197046-0.919678172\mu), a_5 = \tanh(0.835068315-0.919678172\mu), a_6 = \tanh(1.202939584-0.919678172\mu), a_7 = \tanh(2.122617756-1.839356344\mu), a_8 = \tanh(1.386875218-0.459839086\mu), 65.96703854\ a_2 + 15.8645681\ a_3 - 369.5491494\ a_4 + 465.7997072\ a_5 - 45.18900519\ a_6 - 110.3929208\ a_7 + 128.6377379\ a_8 = y + 239.9515868 \tanh(0.651132681), \mu \in \Re\}$, which may consist of one or several 1-manifold segments in $(-1, 1)^8$; and $f^{-1}(y)$ equals $\{\mathbf{x} \in \Re^3 | 0.4x_1+0.5x_2+0.7x_3 = \mu, 65.96703854 \tanh(0.283261412-0.459839086\mu) + 15.8645681 \tanh(-0.452481126-1.839356344\mu) - 369.5491494\tanh(0.467197046-0.919678172\mu) + 465.7997072 \tanh(0.835068315-0.919678172\mu) - 45.18900519\tanh(1.202939584-0.919678172\mu) -$

$110.3929208\tanh(2.122617756-1.839356344\mu) + 128.6377379 \tanh(1.386875218-0.459839086\mu) = y + 239.9515868 \tanh(0.651132681)$, $\mu \in \Re\}$, which may consist of one or several 2-manifold segments in $\Re^3$.

For the SLFN-1 network with $\mu \equiv x_1+x_2+x_3$ shown in Table 3, the preimage analysis states that its non-void set equals $\{\mathbf{a} \in (-1, 1)^2 | \ a_1 = \tanh(0.4\mu), a_2 = \tanh(0.2\mu), \mu \in \Re\}$, which is an 1-manifold in $(-1, 1)^2$; $\mathbf{A}(y)$ equals $\{\mathbf{a} \in (-1, 1)^2 | \ a_1 = \tanh(0.4\mu), a_2 = \tanh(0.2\mu), 18.58899737\ a_1 - 30.7174688 a_2 = y, \mu \in \Re\}$, which may consist of one or several 1-manifold segments in $(-1, 1)^2$; and $f^{-1}(y)$ equals $\{\mathbf{x} \in \Re^3 | \ x_1+x_2+x_3 = \mu, 18.58899737\tanh(0.4\mu) - 30.7174688\tanh(0.2\mu) = y, \mu \in \Re\}$, which may consist of one or several 2-manifold segments in $\Re^3$.

Fig. 1 shows the relationship between the value of μ and the output value y, regarding the SLFN-1 networks shown in Table 2 and Table 3. The relationship between the preimage $f^{-1}(y)$ and the output value y of these two SLFN-1 networks can be observed from Fig. 1. The y-μ graph also indicates the generalization of these two SLFN-1 networks.

Figure 1: The relationship between the value of μ and the output value y, regarding the SLFN-1 networks shown in Table 2 and Table 3.



IMPLICATIONS OF THE FEATURE-DISCOVERY APPLICATION OF THE SLFN-1 NETWORKS

In general, for any SLFN-1 network with $\mu \equiv \mathbf{w}^{\mathrm{T}}\mathbf{x}$ and $\mathbf{w}_j^H \equiv \alpha_j \mathbf{w}$ for all j, the preimage analysis states that its non-void set equals $\{\mathbf{a} \in (-1, 1)^J | \ a_j = \tanh(\alpha_j \ \mu + w_{j0}^H) \ \forall \ j, \mu \in \Re\}$, which is an 1-manifold in $(-1, 1)^J$; $\mathbf{A}(y)$ equals $\{\mathbf{a} \in (-1, 1)^J | \sum_{j=1}^{J} w_j^O \tanh(\alpha_j \ \mu + w_{j0}^H) = y - w_0^O, \ a_j = \tanh(\alpha_j \ \mu + w_{j0}^H)$ $\forall \ j, \mu \in \Re\}$, which may consist of one or several 1-manifold segments in $(-1, 1)^J$; and $f^{-1}(y)$ equals $\{\mathbf{x} \in \Re^I | \ \mathbf{w}^{\mathrm{T}}\mathbf{x} = \mu, \ \sum_{j=1}^{J} w_j^O \tanh(\alpha_j \ \mu + w_{j0}^H) = y - w_0^O, \ a_j = \tanh(\alpha_j \ \mu + w_{j0}^H) \ \forall \ j, \mu \in \Re\}$, which may consist of one or several $(I\text{-}1)$-manifold segments in $\Re^I$. These establish the following Lemma 5. Furthermore, $\mathbf{w}$ is the normal vector of the preimage hyperplane and $\mathbf{w}_j^H$ determines the

orientation of the activation hyperplane in the input space corresponding to the j^{th} 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.

Outcomes of the preimage analysis lead to an understanding of the SLFN-1 itself and further provides the following four insights about the usage of network and about the patterns embedded in (training) data. First, SLFN-1 networks possess the hyperplane-preimage property, which is their generalization. Therefore, the SLFN-1 should be used in the experiments desiring a hyperplane-preimage relationship.

Second, the act to adopt the SLFN-1 architecture at the learning stage does already set the hyperplane-preimage assumption and insert such feature into network. Third, when one gets a SLFN-1 from training, he/she can infer that the empirical data bear the hyperplane-preimage relationship. Fourth, the hyperplane-preimage relationship states that the observed variable of interest is a function of a certain factor obtained from some linear combination of explanatory variables. With such an insight, the practitioner may adopt a common regression method or other suitable tool for data analysis after he/she has properly transformed the explanatory variables into that factor variable. Then, the practitioner may get better understanding about the

application problem.

THE BOND PRICING EXPERIMENT

In this section, we illustrate the feature-discovery process of a practitioner, who knows bond-pricing mechanism well (but less than perfectly). Because bond pricing has been well-studied in the literature, the purpose of this experiment is to illustrate the belief justification process, not to discover extra features of bond pricing.

Before conducting the experiment, the practitioner has some personal beliefs and propositions of the experiment. Based upon the propositions, he first picks up relevant explanatory and observed variables and collects the sample accordingly. Below are the details of his experimental design.

Let the theoretical bond price p^c at time c is derived from (11), which serves as an example of knowledge regarding the data.

$$p^c \equiv \sum_{k=1}^{T_0} \frac{FR}{(1+r^c)^{k-c}} + \frac{F}{(1+r^c)^{T_0-c}}, \tag{11}$$

where r^c is the market rate of interest at time c; $F = 100$ is the face value of the bond; T^0 is the term to maturity at time $c = 0$; R is the coupon rate; and FR is the periodic coupon payment. Then garbled bond prices y^c are generated and used to simulate the set of data that may be observed by any practitioner. Namely, $y^c \equiv p^c + \varepsilon^c$ is the observed data, where ε^c is a white error term distributing as independent and identically distributed normal random variables of mean 0

and variance $(0.2)^2$, for all time c and bonds.

As depicted in Table 4, there are 18 hypothetical combinations of term to maturity and contractual interest rate and generate a set of price data with c = 1/80, 2/80, …, 80/80 through (11). The rate r^c is derived from a normal random number generator of $N(2\%, (0.1\%)^2)$. Accordingly, there are 1,440 training samples with input variables T^c, R and r^c, and the desired output variable y^c, where $T^c \equiv (T^0 - c)$ is the term to maturity at time c.

To examine the generalization of trained networks, the practitioner also generates 1,440 test samples by similar means, except that T^0, c, R and r^c are randomly and independently generated from {1, 2, …, 20} with a probability of 1/20 for each, {1/80, 2/80, …, 80/80} with a probability of 1/80 for each, [0.0%, 3.0%] with a probability density function $f(R)$ = 1/0.03, and $N(2\%, (0.1\%)^2)$, respectively. This setting results in varying instances among the test samples.

The Back Propagation learning algorithm of Rumelhart et al. (1986) is used to train 1,000 SLFNs, each of which has 4 hidden nodes and random initial weights and biases. Among these 1,000 SLFNs, the practitioner picks the three with the smallest *mean square error* (hereafter, MSE) for the test samples. Table 5 shows the (final) weights and biases of these three networks, hereafter named network I, II and III, respectively. The corresponding MSEs for the training samples are 0.414, 0.404 and 0.451, respectively, and the corresponding MSEs for the test samples are 0.429, 0.432 and 0.445, respectively. The average absolute deviation is approximately 0.6, which deviates from the specified error term standard deviation of 0.2.

rank($\mathbf{W}^H$) of all networks I, II and III are 3.

Take network I to illustrate the result of applying the preimage analysis to these three networks. $\Phi_O^{-1}(y) = \{\mathbf{a} | 15.1206a_1 - 34.366a_2 + 5.6589a_3 - 21.9999a_4 = y - 100.4744\}$, which is in the form of a linear equation. Thus, for each non-void value y, $\Phi_O^{-1}(y)$ is a hyperplane in $(-1, 1)^4$. Now

$$\mathbf{W}^{\mathrm{H}} = \begin{bmatrix} -0.0347 & -32.7223 & 18.8396 \\ 0.0544 & -36.8286 & 28.9511 \\ 0.0988 & 43.3354 & 16.8267 \\ -0.0643 & -36.4188 & 53.6646 \end{bmatrix} \quad (12)$$

and $\boldsymbol{\omega}(\mathbf{a}) = (\tanh^{-1}(a_1) + 0.1689, \tanh^{-1}(a_2) + 1.3535, \tanh^{-1}(a_3) + 2.1615, \tanh^{-1}(a_4) - 1.1698)^{\mathrm{T}}$. Thus the **a** vector satisfying the requirement of (13) corresponds to a non-void point; otherwise, a void point. Moreover, for each non-void **a**, the system of simultaneous linear equations $\mathbf{W}^H\mathbf{x} = \boldsymbol{\omega}(\mathbf{a})$ defines a point in the input space.

$$\tanh^{-1}(a_4) = 2.646686748 + 3.248238694 \tanh^{-1}(a_1) - 0.801390022 \tanh^{-1}(a_2) + 0.931270242 \tanh^{-1}(a_3). \quad (13)$$

Thus, the non-void set equals $\{\mathbf{a} | \tanh^{-1}(a_4) = 2.646686748 + 3.248238694 \tanh^{-1}(a_1) - 0.801390022 \tanh^{-1}(a_2) + 0.931270242 \tanh^{-1}(a_3)\}$, which is a 3-manifold in $(-1, 1)^4$. $\mathbf{A}(y)$ equals $\{\mathbf{a} | 15.1206a_1 - 34.366a_2 + 5.6589a_3 - 21.9999a_4 = y - 100.4744, \tanh^{-1}(a_4) = 2.646686748 + 3.248238694 \tanh^{-1}(a_1) - 0.801390022 \tanh^{-1}(a_2) + 0.931270242 \tanh^{-1}(a_3)\}$, which may consist of one or several 2-manifold segments in $(-1, 1)^4$. The corresponding preimage $f^{-1}(y)$ equals $\{(T^c, R, r^c)^{\mathrm{T}} | T^c = 8.619766318 - 16.38982949 \tanh^{-1}(a_1) + 11.9539307$

$\tanh^{-1}(a_2)$ - 2.216810863 $\tanh^{-1}(a_3)$, R = 0.012291155 + 0.01700793 $\tanh^{-1}(a_1)$ - 0.02138228 $\tanh^{-1}(a_2)$ + 0.017746672 $\tanh^{-1}(a_3)$, r^c = 0.046189981 + 0.052432722 $\tanh^{-1}(a_1)$ - 0.015121128 $\tanh^{-1}(a_2)$ + 0.026740939 $\tanh^{-1}(a_3)$, $5.6589a_3$ - 21.9999 tanh(2.646686748 + 3.248238694 $\tanh^{-1}(a_1)$ - 0.801390022 $\tanh^{-1}(a_2)$ + 0.931270242 $\tanh^{-1}(a_3)$) = y - 100.4744 - $15.1206a_1$ + $34.366a_2$, $-1 < a_1 < 1$, $-1 < a_2 < 1$, $-1 < a_3 < 1$}, which may consist of one or several 2-manifold segments in $\mathfrak{R}^3$.

As shown in Fig. 2, the preimage f^{-1} is a complex 2-manifold. According to his beliefs, the practitioner inspects relevant preimage-related properties. Take the following three beliefs as the illustration. First, the practitioner knows that the type of a bond, premium or discount, can be determined by comparing the market interest rate with the contract coupon rate. Specifically, if the coupon rate is greater than the market interest rate, then the bond is priced as premium, else as discount. This belief leads to an insight that the preimage of each reliable network should be parallel to the plane with this property that $r^c = R$. As shown in Fig. 3, the preimages of all three networks show the tendency predicted by the insight. Thus, he gives this belief a high credibility.

Second, the practitioner understands that one bond with a greater coupon rate than another should be priced higher at a given interest rate. From preimage of each network in Fig. 3, he observes that there is a positive relationship between coupon rate and interest rate. Namely, the high coupon rate implies a high price for a bond. Thus, the practitioner gives this understanding

a high credibility and conjectures further that the high interest rate results in the low bond price.

Third, from Fig. 4, the practitioner observes different curvatures for the premium bonds and discount bonds in networks I and II, but not in Network III. Namely, in the r^c and T^c coordinates, the preimages for premium bonds appear to be concave and those for discount bonds appear to convex for networks I and II. Thus, the practitioner gives low credibility to the insight that, when the bond price is held constant, the rate of increase (respectively decrease) in interest rate of a premium (respectively discount) bond increases as the maturity of the bound is getting shorter.

IMPLICATIONS AND FUTURE WORK

The bond pricing experiment shows that the practitioner should have domain knowledge to set up challenging propositions and collect the sample for network's training as well as SLFN knowledge to acquire reliable networks for feature discovery. And complex preimage-related properties make the practitioner conduct belief justification process. For SLFN-p with $p \leq 3$, the inspection of preimage-related properties could be conducted through the y-μ graph like Fig. 1 or the preimage graph in the input space like Fig. 2. For SLFN-p with $p > 4$, however, the inspection of preimage-related properties relies on certain skills and experiences of nonlinear mathematical (or numerical) analysis applied to the obtained f^{-1} representation. Such skills and experiences of nonlinear mathematical (or numerical) analysis for inspecting the complex preimage-related properties is one for future researches.

Other possible future avenues of further enquiry may be the application of the proposed preimage analysis to real world data and the externalization of belief into explicit knowledge through SLFNs.

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

Andrews, R., Diederich, J., & Tickle, A. B. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6), 373-389.

Arslanov, M. Z., Ashigaliev, D. U., & Ismail, E. E. (2002). N-bit parity ordered neural networks. *Neurocomputing*, 48, 1053-1056.

Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science*, 49(3), 312-329.

Hohil, M. E., Liu, D. R., & Smith, S. H. (1999). Solving the N-bit parity problem using neural networks. *Neural Networks*, 12(11), 1321-1323.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation*. Redwood City, CA: Addison-Wesley Publishing Company.

Huang, G., & Babri, H. (1998). 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.

Iyoda, E. M., Nobuhara, H., & Hirota, K. (2003). A solution for the N-bit parity problem using a single translated multiplicative neuron. *Neural Processing Letters*, 18 (3), 213-218.

Lavretsky, E. (2000). On the exact solution of the Parity-N problem using ordered neural networks. *Neural Networks*, 13(8), 643-649.

Liu, D. R., Hohil, M. E., & Smith, S. H. (2002). N-bit parity neural networks: new solutions based on linear programming. *Neurocomputing*, 48, 477-488.

Munkres, J. (1975). *Topology: a first course.* New Jersey: Prentice-Hall Englewood Cliffs.

Murty, K. (1983). *Linear Programming*. New York, NY: John Wiley & Sons.

Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company*. Oxford: Oxford University Press.

Rabuñal, J., Dorado, J., Pazos, A., Pereira, J., & Rivero, D. (2004). A new approach to the extraction of ANN rules and to their generalization capacity through GP. *Neural Computation*, 16, 1483-1523.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing: explorations in the microstructure of cognition*, Cambridge, MA: MIT Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representation by error propagation. In D. E. Rumelhart, and J. L. McClelland (Eds.), *Parallel distributed processing: explorations in the microstructure of cognition*, vol. 1: foundation. Cambridge, MA: MIT Press, 318-362.

Saito, K., & Nakano, R. (2002). Extracting regression rules from neural networks. *Neural Network*, 15(10), 1297-1288.

Setiono, R., Leow, W. K., & Zurada, J. M. (2002). Extraction of rules from artificial neural networks for nonlinear regression. *IEEE Transactions on Neural Networks*, 13(3), 564-577.

Setiono, R., & Liu, H. (1997). NeuroLinear: From neural networks to oblique decision rules. *Neurocomputing*, 17(1), 1-24.

Setiono, R. (1997). On the solution of the parity problem by a single hidden layer feedforward neural network. *Neurocomputing*, 16, 25-235.

Sgroi, D., & Zizzo, D. (2007). Neural Networks and bounded rationality. *Physica A*, 375, 717-725.

Taha, I. A., & Ghosh, J. (1999). Symbolic interpretation of artificial neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 11(3), 448-463.

Tickle, A. B., Andrews, R., Golea, M., & Diederich, J. (1998). The truth will come to light: directions and challenges extracting the knowledge embedded within trained artificial neural networks. *IEEE Transactions on Neural Networks*, 9(6), 1057-1068.

Tsaih, R., Hsu, Y., & Lai, C. (1998). Forecasting S&P 500 stock index futures with the hybrid AI system. *Decision Support Systems*, 23(2), 161-174.

Tsaih, R. (1998). An explanation of reasoning neural networks. *Mathematical and Computer Modelling*, 28, 37-44.

Urcid, G., Ritter, G. X., & Iancu, L. (2004). Single layer morphological Perceptron solution to the N-bit parity problem. *Lecture Notes in Computer Science*, 3287, 171-178.

Zhou, R. R., Chen, S. F., & Chen, Z. Q. (2000). A statistics based approach for extracting priority rules from trained neural networks. *Proceedings of the IEEE-INNS-ENNS International Join Conference on Neural Network*, Como, Italy, CA: IEEE Computer Society, 401-406.

Table 4: the 18 hypothetical bonds with different combinations of term to maturity and contractual interest rate.

Bond No.	**Term to maturity (T^0)**	**Contractual interest rate[a] (R)**	**Bond No.**	**Term to maturity (T^0)**	**Contractual interest rate (R)**	**Bond No.**	**Term to maturity (T^0)**	**Contractual interest rate (R)**
1	2	0.0%	7	2	1.5%	13	2	3.0%
2	4	0.0%	8	4	1.5%	14	4	3.0%
3	7	0.0%	9	7	1.5%	15	7	3.0%
4	10	0.0%	10	10	1.5%	16	10	3.0%
5	15	0.0%	11	15	1.5%	17	15	3.0%
6	20	0.0%	12	20	1.5%	18	20	3.0%

[a] Assume that coupon payments are made annually.

Table 5: final weights and biases of networks I, II and III, respectively.

Weights and Biases / Network	w_0^O	j	w_{j0}^H	w_j^O	w_{j1}^H	w_{j2}^H	w_{j3}^H
I	100.4744	1	-0.1689	15.1206	-0.0347	-32.7223	18.8396
		2	-1.3535	-34.3660	0.0544	-36.8286	28.9551
		3	-2.1615	5.6589	0.0988	43.3354	16.8267
		4	1.1698	-21.9999	-0.0643	-36.4188	53.6646
II	93.6583	1	0.4510	-23.3874	-0.0571	-33.4648	71.8090
		2	0.8413	36.9871	-0.0467	32.9078	-10.5926
		3	-1.1572	-10.4621	0.0699	45.5792	43.2855
		4	1.2874	-9.2684	-0.0458	17.5685	-87.314

							7
III	104.8248	1	0.7832	-14.0352	-0.0519	-27.1299	49.3836
		2	1.3108	-16.7297	-0.0571	-27.3748	14.5874
		3	-1.5287	-30.1819	0.0631	-37.1108	33.3026
		4	-0.6010	13.1504	-0.0524	-34.9042	36.9149

Figure 2: The preimage graphs of Network I. The numbers within the parentheses are values of *y*.

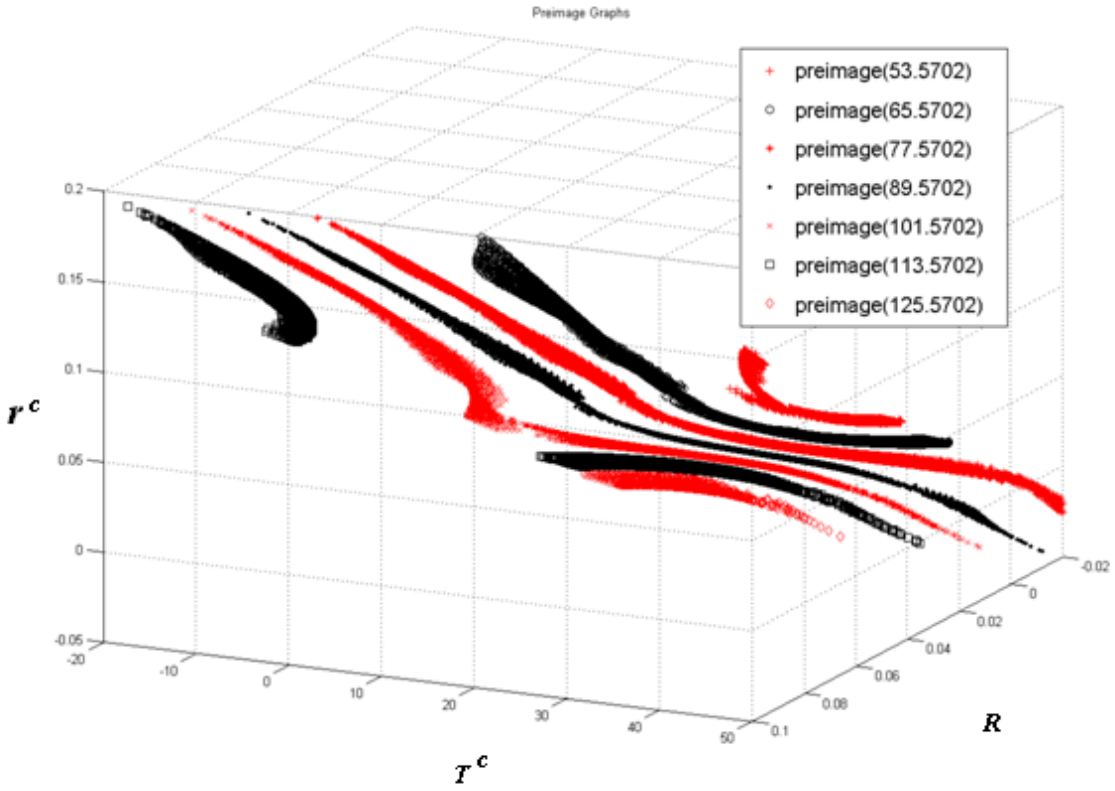


Figure 3: Preimage graphs along the r^c and R plane for networks I, II, III (from top to bottom), respectively. The numbers within the parentheses are values of y.

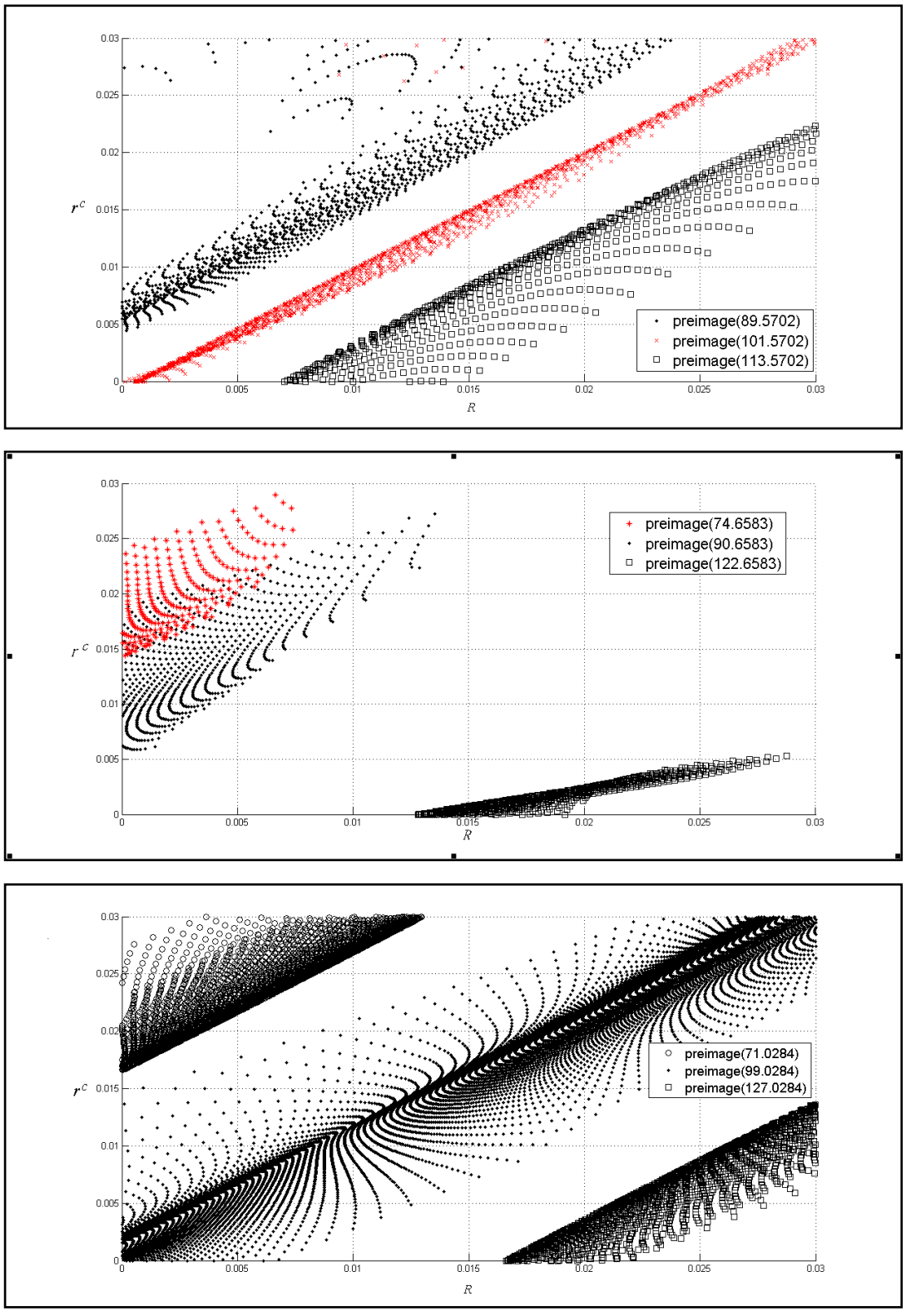
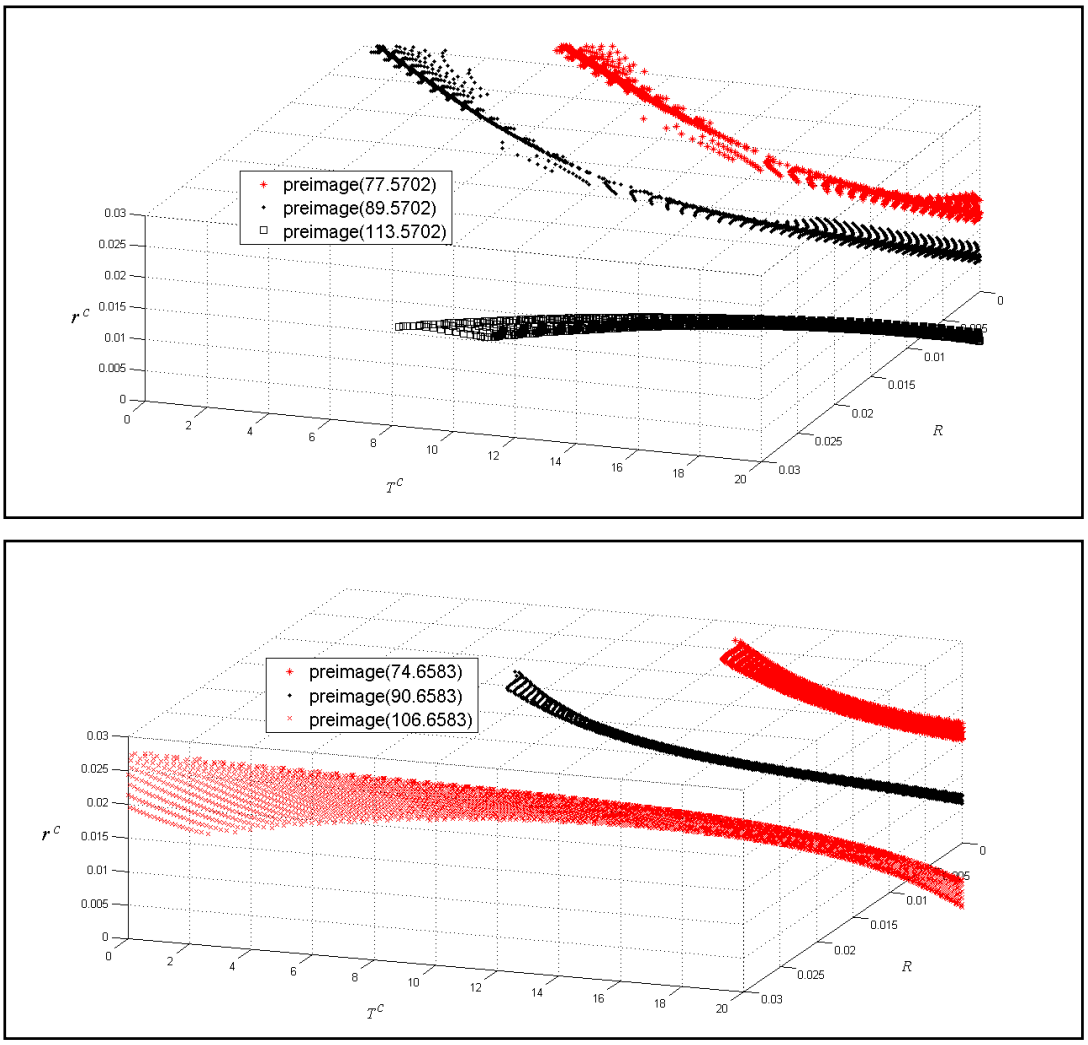


Figure 4: Preimages graphs along the plane of r^c vs. T^c for networks I, II, III (from top to bottom), respectively. The numbers within the parentheses are values of y.



計畫成果自評：

本研究內容與原計畫相符程度高，達成預期之研究目的。不過，也發現其後續研究之有趣以及困難處。本研究報告將加以修改後，投稿到學術期刊發表。

出席國際學術會議心得報告

計畫編號	NSC 97－2410－H－004－017
計畫名稱	利用 preimage 分析萃取規則之實作
出國人員姓名 服務機關及職稱	蔡瑞煌　國立政治大學資訊管理學系 教授
會議時間地點	14-19 June 2009, Atlanta, Georgia
會議名稱	2009 International Joint Conference On Neural Networks (IJCNN2009)
發表論文題目	Knowledge-Internalization Process for Neural-Network Practitioners

一、參加會議經過

我於 16/06/2009 凌晨到達 Atlanta 後，於會場上聆聽多場 Plenary Talk 及多篇論文之發表，亦於 16/06/2009 發表論文，於 17/06/2009 晚上離開 Atlanta 回國。附件是我所發表之論文。

二、與會心得

Plenary Talk 邀請了不少的 Neural Networks 學界裡之知名學者，例如 John Hopfield, Bernard Widrow, John Taylor, Walter Freeman 等人，來演講，我受益不少。我回國後，加以修改我所發表之論文，將投稿於期刊上。

Knowledge-Internalization Process for Neural-Networks Practitioners

***Abstract*—This study explores the knowledge-internalization process within which a neural-network practitioner embody the explicit knowledge obtained from extracting network's preimage, the set of input values for a given output value, into his/her tacit knowledge. With a number of well-trained single-hidden layer feed-forward neural networks, the practitioner first extracts the (nonlinear) preimage of each trained network. The practitioner then internalizes the explicit outcomes and the insights obtained from the preimage extracting process into his/her tacit knowledge bases. We use the experiment of bond-pricing analysis to illustrate the knowledge-internalization process. This study adds to the literature by introducing the knowledge-internalization process. Moreover, in contrast to the data analyses in previous studies, this study uses mathematical analyses to identify networks' preimages.**

I. BLACK-BOX DILLEMA AND KNOWLEDGE ACQUISITION

WHEN practitioners apply Artificial Neural Networks (ANN) to resolving social science issues, there is a dynamic human process of justifying personal belief toward the "truth". Reference [9] stated that ANNs can be trained (just as human children are taught), ANNs can learn primarily through example (as is often the case with humans), and ANNs can create general pattern-recognizing algorithms, learning general rules of optimal behavior. Since then, varieties of ANN have been developed and applied in many fields as modeling tools to see if the ANN does provide a model of human behavior and does approximate likely patterns of human behavior. At the beginning stage, a huge amount of experiments are conducted to see if the corresponding performances of the trained ANN are acceptable. Most experimental results are positive. For instance, [13] stated that ANNs "are consistent with observed laboratory play in two very important senses. Firstly, they select a rule for behavior which appears very similar to that used by laboratory subjects. Secondly, using this rule they perform optimally only approximately 60% of the time." (p. 717) Later, the excitement shifts to applying ANN to resolving the challenging issue of domain. For instance, through extracting rules or features from a well-trained ANN, one tries to identify risk factors for newly issued securities, which have a prohibitively small number of observations. There are several concerns, however, when one has such application.

On the one hand, the practitioner has to cope with the black box[1] image of ANN to obtain a better understanding of relations between the input to ANN and its output. Reading or understanding the knowledge in ANN is difficult because the knowledge is distributed over the entire network and the relation between the input to ANN and its output is multivariate and nonlinear. Nevertheless, there is a huge amount of work that explore various "mechanisms, procedures, and algorithms designed to insert knowledge into ANNs (knowledge initialization), extract rules from trained ANNs (rule extraction), and utilize ANNs to refine existing rule bases (rule refinement)." [1, p. 373] Some, but not exhaustive, recent studies can refer to [1]-[2], [10]-[12], [14]-[15], [17]-[18]. These studies are contrived by the *engineering* design with data analysis and approximation. For instance, to identify the premise of a single rule stated in (1), most work use either training data or generated data, which the trained network itself yields. Due to the finite number of (training or generated) data instances, however, such a data analysis covers only some finite countable points in the (presumed) region of the rule premise, instead of the entire region. Reference [11] implemented a piecewise linear approximation on each hidden node to divide the input space into sub-regions in each of which, a corresponding linear equation that approximates the network's output is defined as the consequent of the extracted rule to ensure the predicted value can be calculated from a comprehensible multivariate polynomial representation. Reference [3] solved the inversion problem through the back-propagation a union of polyhedra, which approximate (arbitrarily well) any reasonable set.

Rule: If the input sample is in some sub-region of the input space, Then the predicted value is given by a corresponding linear regression equation. (1)

On the other hand, instead of a knowledge acquisition process, practitioners conduct a knowledge internalization process within which they embody the explicit outcomes and the insights obtained from the experiment into their tacit knowledge. Knowledge is normally tacit -- highly personal and hard to formalize. Subjective insights, intuitions and hunches are common heard from the discussions and sometimes difficult to replicate as the validation process depends on certain skills and experience. It is not trivial to conduct such knowledge internalization even when some explicit outcome is extracted from ANN. Furthermore, in most social science applications (e.g., financial market tests), the knowledge internalization process needs to cope with the

Manuscript received Feb 12, 2009. This work was supported in part 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.

[1] A black box refers to a system that produces "certain outputs from certain inputs without explaining why or how." [7, pg. 1483] The black box image for many years has gradually discouraged the study or application of the ANN.

unlikely perfect learning due to the defect design of the architecture of ANN[2] and the garbled data. In literature, however, there are no discussions connecting to such knowledge internalization process.

II. THE KNOWLEDGE INTERNALIZATION

This study explores such a knowledge internalization process. Specifically, the ANN used here is the real-valued single-hidden layer feed-forward neural networks (hereafter also referred to as SLFN) with one output node. Furthermore, the following assumption is set to help average out noises in estimates from individual SLFNs and serve as a stabilization measure to the knowledge internalization process: The practitioner should have a number of SLFNs, each of which is perceived well-trained by the practitioner, although does not necessarily provide a globally optimal learning result.

In order to not trap in the criticisms due to adopting the data analysis and approximation, mathematical analysis is adopted here to explicitly specify the (nonlinear) preimage of each SLFN's mapping and thus the rule (2):

Rule: If ($\mathbf{x} \in$ the $f^{-1}(y')$ **region**), then ($y = y'$), (2)

where $\mathbf{x}$ is the vector of explanatory variables; y is the network's response; $f : \mathbf{X} \rightarrow \mathbf{Y}$ is the function of the trained SLFN and $y \equiv f(\mathbf{x})$; y' is a constant; and the preimage f^{-1} is the inverse function of f. The preimage $f^{-1}(y')$ also represents the collection of inputs of the given output value y'.

The function representation f and preimage f^{-1} of the obtained SLFN are instances of explicit outcomes that can be "easily communicated and shared in the form of hard data, scientific formula, codified procedures, or universal principles" [6, p. 8]. When the practitioner conducts the experiment without domain expertise, the explicit outcomes and the insights obtained within the extracting process make some statement exclusively. But when there is certain prior belief, the practitioner focuses on the credibility of belief, the extent to which the belief can be generalized within the preimage-extracting process and the subsequent examination process. At the end of knowledge internalization process, the practitioner can have a posterior belief that has "high credibility" if the explicit outcomes and the insights obtained within the extracting process corroborate the belief; and "low credibility" if some corresponding result contradicts and weakens the belief. That is, when the explicit outcomes and the insights obtained within the extracting process are totally consistent with the practitioner's prior belief, he/she may accord his/her belief a high credibility. Conversely, an inconsistent result triggers the following examinations of SLFNs and belief instead of an immediate rating:

(i) Investigating whether there exist factors or noises leading to a defect design of the SLFN such that all well-trained SLFNs are not suitable for the purpose of rating the credibility of belief.
(ii) Examining whether some of the obtained SLFNs are optimal to the extent that they are suitable for the purpose of rating the credibility of belief.
(iii) Consolidating the explicit outcomes the obtained insights amongst all reliable SLFNs.
(iv) Contrasting the belief with the consolidated outcome of reliable SLFNs.

Only when the practitioner feels certain that he/she can eliminate the first possibility, should he/she rate the credibility of belief. Furthermore, the practitioner would conservatively follow the explicit outcomes and the obtained insights.

Section III uses the experiment of bond-pricing analysis, which in literature is a nonlinear regression problem with continuous variables, to illustrate the knowledge internalization process. At the end, conclusion and future work are offered.

III. THE BOND PRICING EXPERIMENT

In order to simulate the set of data that may be observed by a representative practitioner, who know about the bond-pricing mechanism well (but less than perfectly), garbled training samples of bond price $y_t = p_t + \varepsilon_t$ are generated and used. p_t is the theoretic value of the bond at time t and is derived from (3), which serves as an example of complete domain knowledge with respect to the bond pricing model, and ε_t is a white error term provided by a normal random number generator of $N(0, (0.2)^2)$. Namely, y_t is perturbed by a white noise.

$$p_t \equiv \sum_{k=1}^{T_0} \frac{C}{(1+r_t)^{k-t}} + \frac{F}{(1+r_t)^{T_0-t}} \qquad (3)$$

According to (3), p_t is determined by (i) r_t, the market rate of interest at time t; (ii) F, the face value of the bond, which generally equals 100; (iii) T_0, the term to maturity at time $t = 0$; and (iv) C, the periodic coupon payment, which equals $F \times r_c$. As depicted in Table 1, we use 18 hypothetical combinations of term to maturity and contractual interest rate and generate a set of price data with t = 1/80, 2/80, …, 80/80 through (3). The rate r_t is derived from a normal random number generator of $N(2\%, (0.1\%)^2)$. Accordingly, we have 1,440 training samples with input variables T_t, r_c and r_t, and the desired output variable y_t, where $T_t \equiv (T_0 - t)$ is the term to maturity at time t.

To examine the generalization of trained networks, we also generate 1,440 test samples by the similar means, except that T_0, t, r_c and r_t are randomly and independently generated from {1, 2, …, 20} with a probability of 1/20 for each, {1/80, 2/80, …, 80/80} with a probability of 1/80 for each, [0.0%, 3.0%] with a probability density function $f(r_c)$ = 1/0.03 and $N(2\%, (0.1\%)^2)$, respectively. This setting results in varying instances among the test samples.

We, as the representative practitioner, adopt the Back Propagation learning algorithm [8] to train 1,000 SLFNs, each of which has 4 hidden nodes and random initial weights and biases. Among the 1,000 SLFNs, we pick the three with the smallest mean square error (hereafter, MSE) for the test samples. Table 2 shows the (final) weights and biases of these three SLFNs, hereafter named network I, II and III,

[2] In most social science studies, it is characteristically more difficult to determine the best architecture of ANN.

respectively. The corresponding MSEs for the training samples are 0.414, 0.404 and 0.451, respectively; and the corresponding MSEs for the test samples are 0.429, 0.432 and 0.445, respectively. The average absolute deviation is approximately 0.6, which deviates from the specified error term standard deviation of 0.2. The pricing error is unrelated to theoretic prices p_t. but related to observed prices y_t.

By definition, for each SLFN, the h^{th} activation value a_h equals $tanh(b_h^H + \sum_{i=1}^{3} w_{hi}^H x_i)$, h = 1, ..., 4; the output y equals $b^o + \sum_{h=1}^{4} w_h^o a_h$; and the function f equals $b^o + \sum_{h=1}^{4} w_h^o tanh(b_h^H + \sum_{i=1}^{3} w_{hi}^H x_i)$. Hereafter, let $(\cdot)^{\mathrm{T}}$ be the transpose of $(\cdot)$ for $(\cdot)$ to be a vector or a matrix. Furthermore,

w_h^o ≡ the weight of the h^{th} activation value for the output, where the superscript *o* throughout the paper indicates quantities related to the output layer;

b^o ≡ the bias of the output node;

w_{hi}^H ≡ the weight of the i^{th} input for the h^{th} hidden node, where the superscript *H* throughout the paper indicates quantities related to the hidden layer;

$\mathbf{w}_{h\cdot}^H \equiv (w_{h1}^H, w_{h2}^H, w_{h3}^H)^{\mathrm{T}}$, the 3x1vector of weights between the h^{th} hidden node and the input layer;

$\mathbf{W}^H \equiv (\mathbf{w}_{1\cdot}^H, \mathbf{w}_{2\cdot}^H, \mathbf{w}_{3\cdot}^H, \mathbf{w}_{4\cdot}^H)^{\mathrm{T}}$, the 4x3 matrix of weights between the hidden nodes and the input layer; and

b_h^H ≡ the bias of the h^{th} hidden node.

For ease of reference in later discussions, we also call R^3 the *input space* and $(-1, 1)^4$ the *activation space*.

For each SLFN, $f^{-1}(y) = \mathbf{\Phi}_{tanh}^{-1} \circ \mathbf{\Phi}_o^{-1}(y)$, with

$$\mathbf{\Phi}_o^{-1}(y) \equiv \{\mathbf{a} \in (-1, 1)^4 | \sum_{h=1}^{4} w_h^o a_h = y - b^o\}, \quad (4)$$

$$\mathbf{\Phi}_{tanh}^{-1}(\mathbf{a}) \equiv \bigcap_{h=1}^{4} \{\mathbf{x} \in \Re^3 | \mathbf{x}^{\mathrm{T}} \mathbf{w}_{h\cdot}^H = tanh^{-1}(a_h) - b_h^H\}, \quad (5)$$

where Ω is a subset of $\Re$ and $tanh^{-1}(x) \equiv 0.5\ln\left(\frac{1+x}{1-x}\right)$. Formally, the followings are defined for every SLFN:

(i) A value $y \in \Re$ is *void* if $y \notin f(\{\Re^3\})$, i.e., for all $\mathbf{x} \in \Re^3$, $f(\mathbf{x}) \neq y$. Otherwise, y is *non-void*.

(ii) A point $\mathbf{a} \in (-1, 1)^4$ is *void* if $\mathbf{a} \notin \mathbf{\Phi}_{tanh}(\{\Re^3\})$, i.e., for all $\mathbf{x} \in \Re^3$, $\mathbf{\Phi}_{tanh}(\mathbf{x}) \neq \mathbf{a}$. 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*.

(iii) The *image* of an input $\mathbf{x} \in \Re^3$ is $y \equiv f(\mathbf{x})$ for $y \in \Omega$.

(iv) The *preimage* of a non-void output value y is the set $f^{-1}(y) \equiv \{\mathbf{x} \in \Re^3 | f(\mathbf{x}) = y\}$. The preimage of a void value y is the empty set.

(v) The *internal-preimage* of a non-void output value y is the set $\{\mathbf{a} \in (-1, 1)^4 | \Phi_o(\mathbf{a}) = y\}$ on the activation space.

Given the weights and biases of each SLFN, the preimage-extracting phase conducts the following steps, where $rank(\mathbf{D})$ is the rank of the 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):

Step 1: Derive the expression of $\mathbf{\Phi}_o^{-1}(y)$;

Step 2: Derive the expression of the non-void set that is defined as $\{\mathbf{a} | rank(\mathbf{W}^H \vdots \boldsymbol{\omega}(\mathbf{a})) = rank(\mathbf{W}^H)\}$;

Step 3: Derive the expression of $\mathbf{A}(y)$ that is defined as $\{\mathbf{a} | \mathbf{a} \in \mathbf{\Phi}_o^{-1}(y)$ AND $rank(\mathbf{W}^H \vdots \boldsymbol{\omega}(\mathbf{a})) = rank(\mathbf{W}^H)\}$; and

Step 4: Derive the expression of $f^{-1}(y)$ that is defined as $\{\mathbf{x} | \mathbf{W}^H\mathbf{x} = \boldsymbol{\omega}(\mathbf{a})$ with all $\mathbf{a} \in \mathbf{A}(y)\}$.

Take Network I to illustrate the explicit outcomes and the insights obtained within the extracting process. $\mathbf{\Phi}_o^{-1}(y) = \{\mathbf{a} | 15.1206a_1 - 34.366a_2 + 5.6589a_3 - 21.9999a_4 = y - 100.4744\}$. $\mathbf{\Phi}_o^{-1}(y)$ is in the form of linear equation. Thus, for each non-void value y, $\mathbf{\Phi}_o^{-1}(y)$ is a hyperplane in $(-1, 1)^4$. As y changes, $\mathbf{\Phi}_o^{-1}(y)$ forms parallel hyperplanes in $(-1, 1)^4$; for any y changes of the same magnitude, the corresponding hyperplanes are spaced by the same distance. The activation space is entirely covered by these parallel $\mathbf{\Phi}_o^{-1}(y)$ hyperplanes, orderly in terms of the values of non-void y. Furthermore, the center of these parallel hyperplanes is the $\mathbf{\Phi}_o^{-1}(100.4744)$ hyperplane. These parallel hyperplanes form a (linear) scalar field: For each point $\mathbf{a}$ of the activation space, there is only one output value y whose $\mathbf{\Phi}_o^{-1}(y)$ hyperplane passes point $\mathbf{a}$; all points on the same $\mathbf{\Phi}_o^{-1}(y)$ hyperplane are associated with the same y value.

Note that the function $\mathbf{x}^{\mathrm{T}} \mathbf{w}_{h\cdot}^H = tanh^{-1}(a_h) - b_h^H$ within the h^{th} component in the right-hand side of (5) is a separable function from the one within the other components. Given an activation value a_h, $\{\mathbf{x} \in \Re^3 | \mathbf{x}^{\mathrm{T}} \mathbf{w}_{h\cdot}^H = tanh^{-1}(a_h) - b_h^H\}$ defines a hyperplane in the input space, since all $\mathbf{w}_{h\cdot}^H$ and b_h^H are given constants. For the h^{th} hidden node, the hyperplanes associated with various a_h values are parallel and form a (linear) scalar activation field in the input space [16]: For each point $\mathbf{x}$ of the input space, there is only one activation value a_h whose corresponding hyperplane passes point $\mathbf{x}$; all points on this hyperplane are associated with the same a_h value. Furthermore, each hidden node gives rise to an activation field in the input space, and four hidden nodes set up four independent activation fields in the input space.

$\bigcap_{h=1}^{4} \{\mathbf{x} | \mathbf{x}^{\mathrm{T}} \mathbf{w}_{h\cdot}^H = tanh^{-1}(a_h) - b_h^H\}$ in (5) can be denoted by $\{\mathbf{x} | \mathbf{W}^H\mathbf{x} = \boldsymbol{\omega}(\mathbf{a})\}$, where $\boldsymbol{\omega}(\mathbf{a}) \equiv (\omega_1(a_1), \omega_2(a_2), \omega_3(a_3), \omega_4(a_4))^{\mathrm{T}}$ and $\omega_h(a_h) \equiv tanh^{-1}(a_h) - b_h^H$ for all $1 \leq h \leq 4$. Given the activation values of $\mathbf{a}$, $\boldsymbol{\omega}(\mathbf{a})$ is simply a vector of known component values and the representation

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

is a system of four simultaneous linear equations with three unknowns. Furthermore, $\mathbf{W}^H\mathbf{x} = \boldsymbol{\omega}(\mathbf{a})$ is a set of inconsistent simultaneous equations if $rank(\mathbf{W}^H \vdots \boldsymbol{\omega}(\mathbf{a})) = rank(\mathbf{W}^H) + 1$ [5, p. 108], and thus the corresponding point $\mathbf{a}$ is void. The discussion establishes Lemma 1 below.

Lemma 1. An activation value $\mathbf{a}$ is void if $rank(\mathbf{W}^H \vdots \boldsymbol{\omega}(\mathbf{a})) = rank(\mathbf{W}^H) + 1$; otherwise, $\mathbf{a}$ is non-void.

Now $\mathbf{W}^H$ equals $\begin{bmatrix} -0.0347 & -32.7223 & 18.8396 \\ 0.0544 & -36.8286 & 28.9511 \\ 0.0988 & 43.3354 & 16.8267 \\ -0.0643 & -36.4188 & 53.6646 \end{bmatrix}$ and $\boldsymbol{\omega}(\mathbf{a})$ equals $(tanh^{-1}(a_1) + 0.1689, tanh^{-1}(a_2) + 1.3535, tanh^{-1}(a_3) + 2.1615, tanh^{-1}(a_4) - 1.1698)^T$. Thus the **a** vector satisfying the requirement of (7) corresponds to a non-void point; otherwise, a void point. Moreover, for each non-void **a**, (6) defines a *point* in the input space.

$$tanh^{-1}(a_4) = 2.646686748 + 3.248238694tanh^{-1}(a_1) - 0.801390022tanh^{-1}(a_2) + 0.931270242tanh^{-1}(a_3) \quad (7)$$

Thus, the non-void set, which consists of all non-void **a**'s, equals $\{\mathbf{a} | tanh^{-1}(a_4) = 2.646686748 + 3.248238694 tanh^{-1}(a_1) - 0.801390022tanh^{-1}(a_2) + 0.931270242tanh^{-1}(a_3)\}$, which is a (non-linear) 3-manifold in $(-1, 1)^4$. A *p*-manifold is a Hausdorff space **X** with a countable basis such that each point x of **X** has a neighborhood that is homomorphic with an open subset of $\Re^p$. A 1-manifold is often called a curve, and a 2-manifold is called a surface [4]. For our purpose, it suffices to consider Euclidean spaces, the most common members of the family of Hausdorff spaces.

The internal-preimage of a non-void value y, $\mathbf{A}(y) \equiv \{\mathbf{a} | 15.1206a_1 - 34.366a_2 + 5.6589a_3 - 21.9999a_4 = y - 100.4744, tanh^{-1}(a_4) = 2.646686748 + 3.248238694tanh^{-1}(a_1) - 0.801390022tanh^{-1}(a_2) + 0.931270242tanh^{-1}(a_3)\}$, is a 2-manifold in $(-1, 1)^4$. The group of the (non-empty) $\mathbf{A}(y)$s forms an *internal-preimage field* in the activation space – there is one and only one $\mathbf{A}(y)$ passing through each non-void **a**; for any **a** on $\mathbf{A}(y')$, its output value is equal to y'; and $\mathbf{A}(y)$s are geometrically aligned orderly according to the positions of $\boldsymbol{\Phi}_o^{-1}(y)$s, since the $\mathbf{A}(y)$ is the intersection of $\boldsymbol{\Phi}_o^{-1}(y)$ and the non-void set. Thus, the non-void value y can be represented as $15.1206a_1 - 34.366a_2 + 5.6589a_3 - 21.9999tanh(2.646686748 + 3.248238694tanh^{-1}(a_1) - 0.801390022tanh^{-1}(a_2) + 0.931270242tanh^{-1}(a_3)) + 100.4744$.

Any point that is in $\mathbf{A}(y)$ links to the preimage $f^{-1}(y)$. Thus, for any non-void y, the corresponding preimage $f^{-1}(y)$ of Network I equals $\{(T_t, r_c, r_t)^T | T_t = 8.619766318 - 16.38982949tanh^{-1}(a_1) + 11.9539307tanh^{-1}(a_2) - 2.216810863tanh^{-1}(a_3), r_c = 0.012291155 + 0.01700793tanh^{-1}(a_1) - 0.02138228tanh^{-1}(a_2) + 0.017746672tanh^{-1}(a_3), r_t = 0.046189981 + 0.052432722tanh^{-1}(a_1) - 0.015121128tanh^{-1}(a_2) + 0.026740939tanh^{-1}(a_3), 5.6589a_3 - 21.9999tanh(2.646686748 + 3.248238694tanh^{-1}(a_1) - 0.801390022tanh^{-1}(a_2) + 0.931270242tanh^{-1}(a_3)) = y - 100.4744 - 15.1206a_1 + 34.366a_2, -1 < a_1 < 1, -1 < a_2 < 1, -1 < a_3 < 1\}$, which is a 2-manifold in the input space.

The input space is entirely covered by an orderly grouping of $f^{-1}(y)$'s that forms a *preimage field* in the input space. That is, there is one and only one $f^{-1}(y)$ passing through each **x**; the corresponding output value of the network to this **x** is the y value associated with this $f^{-1}(y)$; and $f^{-1}(y)$'s are aligned orderly, though not necessarily with equal space.

Similar extracting processes can be applied to Network II and III to have their corresponding preimages shown in (8) and (9), respectively.

$$f^{-1}(y) = \{(T_t, r_c, r_t)^T | T_t = 15.23359752 - 4.059672115tanh^{-1}(a_1) - 10.05010689tanh^{-1}(a_2) + 4.275432481tanh^{-1}(a_3), r_c = -0.002434911 - 0.002726715tanh^{-1}(a_1) + 0.015945201 tanh^{-1}(a_2) + 0.008425543tanh^{-1}(a_3), r_t = 0.004697943 + 0.009427004tanh^{-1}(a_1) - 0.000560628tanh^{-1}(a_2) + 0.007326189 tanh^{-1}(a_3), - 10.4621a_3 - 9.2684 tanh(0.13672399 - 0.685087351tanh^{-1}(a_1) + 0.78937922 tanh^{-1}(a_2) - 0.687474652tanh^{-1}(a_3)) = y - 93.6583 + 23.3874a_1 - 36.9871a_2, -1 < a_1 < 1, -1 < a_2 < 1, -1 < a_3 < 1\} \quad (8)$$

$$f^{-1}(y) = \{(T_t, r_c, r_t)^T | T_t = 22.39422393 - 2.792522056tanh^{-1}(a_1) - 7.007049818tanh^{-1}(a_2) + 7.210230755tanh^{-1}(a_3), r_c = 0.007440778 + 0.021281602tanh^{-1}(a_1) - 0.036533434 tanh^{-1}(a_2) - 0.015555372tanh^{-1}(a_3), r_t = 0.011763577 + 0.029006307tanh^{-1}(a_1) - 0.027434499tanh^{-1}(a_2) - 0.000968028tanh^{-1}(a_3), - 30.1819a_3 + 13.1504 tanh(-1.599920472 + 0.474275763tanh^{-1}(a_1) + 0.629597938tanh^{-1}(a_2) + 0.129397058tanh^{-1}(a_3)) = y - 104.8248 + 14.0352a_1 + 16.7297a_2, -1 < a_1 < 1, -1 < a_2 < 1, -1 < a_3 < 1\} \quad (9)$$

Since the dimension of the input space in all networks in Table 2 is three, we can also draw the three-dimensional preimage of each obtained SFLN based upon its corresponding mathematical expression.

With the explicit outcomes and insights obtained from the above preimage-extracting process, we, as the representative practitioner, conduct the examination of several beliefs.

We know that premium bonds and discount bonds could be determined by comparing the market interest rate with the contract coupon rate. Specifically, if the coupon rate is greater than the market interest rate, then the bond is priced with a premium. Else the bond is traded at a discount. This belief leads to an insight that the preimage of a reliable SLFN should be parallel to the plane with this property that $r_t = r_c$. As shown in Fig. 1, these obtained preimages reveal the tendency of such insight. Thus, this belief has "high" credibility.

Another common sense is that one bond with a greater coupon rate than another should be priced higher at a given interest rate. Namely, the high coupon rate implies a high price for one bond. Preimage of each SLFN in Fig. 1 does demonstrate a positive relationship between coupon rate and interest rate in the same. Thus, we could conjecture that the high interest rate results in the low bond price and this common known sense has "high" credibility.

Furthermore, observing Fig. 2, we find that the curvatures for the premium bonds and discount bonds are different in Networks I and II. In the r_t and T_t coordinate, the preimages for premium bonds appear to be concave and those for discount bonds appear to convex. This may imply another insight that is consistent with the viewpoint in the bond pricing but has "low" credibility. Namely, for keeping constant price, the increased (or decreased) magnitude of interest rate is getting great when the maturity for a premium

(or discount) bond is getting short. In other words, the interest rate is more significantly influence on short-term bonds than long-term bonds for keeping the same price.

IV. CONCLUSIONS AND FUTURE WORK

This study adds to the literature by introducing the knowledge internalization process for the SLFN practitioner. A possible future avenue of further enquiry may be exploring newly issued financial instruments with limited instances using the process set out here. In addition, the knowledge internalization process applied to real world data and the externalization of belief into explicit knowledge are all issues meriting further study.

This study also demonstrates the appropriateness of adopting mathematical analysis in place of the more usual data analysis to extract the accurate preimage from SLFNs. With the extraction of preimages for each SLFN, the practitioner may obtain certain insights for any unexplored data which none knows the suited tools to analyze, because SLFN is known for its ability of being an universal approximator. Further, with these obtained insights, the practitioner may adopt the most suited tools for the empirical data, the tools that may help him/her more easily and properly to establish the model for the variables. Then, he may get more useful insights and understanding. For example, when the dependent variable of interest is monotone along with a certain factor linearly or nonlinearly combined by these explanatory (or independent) variables, the extraction of preimages may be a great benefit. In such a case, these extracted preimages appear to be parallel with this linear combined factor. With such an insight, the practitioner may adopt the common regression method or other suited tools for the data after he properly transform the variables. Of course, this argument is one of future researches.

REFERENCES

[1] R. Andrews, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge-Based Systems*, Vol.8, no. 6, pp. 373-389, 1995.

[2] B. Baesens, R. Setiono, C. Mues, and J. Vanthienen, "Using neural network rule extraction and decision tables for credit-risk evaluation," *Management Science*, Vol. 49, no. 3, pp. 312-329, 2003.

[3] F. Maire, "Rule-extraction by backpropagation of polyhedral," *Neural Networks*, Vol. 12, pp. 717-725, 1999

[4] J. Munkres, *Topology: a first course*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[5] K. Murty, *Linear Programming*. New York, NY: John Wiley & Sons, 1983.

[6] I. Nonaka, and H. Takeuchi, *The Knowledge-creating Company*. Oxford: Oxford University Press, 1995.

[7] J. Rabuñal, J. Dorado, A. Pazos, J. Pereira, and D. Rivero, "A new approach to the extraction of ANN rules and to their generalization capacity through GP," *Neural Computation*, Vol. 16, pp. 1483-1523, 2004

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representation by error propagation," In D. E. Rumelhart, & J. L.McClelland (Eds.), *Parallel distributed processing: explorations in the microstructure of cognition*, vol. 1: foundation. Cambridge, MA: MIT Press, pp. 318-362, 1986.

[9] D. E. Rumelhart, and J. L. McClelland, Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundation. Cambridge, MA: MIT Press, 1986

[10] K. Saito, and R. Nakano, "Extracting regression rules from neural networks," *Neural Network*, Vol. 15, no. 10, pp. 1297-1288, 2002

[11] R. Setiono, W. K. Leow, & J. M. Zurada, "Extraction of rules from artificial neural networks for nonlinear regression," *IEEE Transactions on Neural Networks*, Vol. 13, no. 3, pp.564-577, 2002.

[12] R. Setiono, and H. Liu, "NeuroLinear : From neural networks to oblique decision rules," *Neurocomputing*, Vol. 17, no. 1, pp. 1-24, 1997.

[13] D. Sgroi, and D. Zizzo, "Neural Networks and bounded rationality," *Physica A*, Vol. 375, pp. 717-725, 2007.

[14] I. A. Taha, and J. Ghosh, "Symbolic interpretation of artificial neural networks," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, no. 3, pp. 448-463, 1999.

[15] A. B. Tickle, R. Andrews, M. Golea, & J. Diederich, "The truth will come to light: directions and challenges extracting the knowledge embedded within trained artificial neural networks," *IEEE Transactions on Neural Networks*, Vol. 9, no. 6, pp. 1057-1068, 1998.

[16] R. Tsaih, "An explanation of reasoning neural networks," *Mathematical and Computer Modelling*, Vol. 28, pp. 37-44, 1998.

[17] R. Tsaih, Y. Hsu, and C. Lai, "Forecasting S&P 500 stock index futures with the hybrid AI system," *Decision Support Systems*, Vol. 23, no. 2, pp. 161-174, 1998.

[18] R. R. Zhou, S. F. Chen, and Z. Q. Chen, "A statistics based approach for extracting priority rules from trained neural networks," *Proceedings of the IEEE-INNS-ENNS International Join Conference on Neural Network*, Como, Italy, CA: IEEE Computer Society, pp. 401-406, 2000.

TABLE I

THE 18 HYPOTHETICAL BONDS WITH DIFFERENT COMBINATIONS OF TERM TO MATURITY AND CONTRACTUAL INTEREST RATE.

Bond No.	**Term to maturity (T_0)**	**Contractual interest rate[a] (r_c)**	**Bond No.**	**Term to maturity (T_0)**	**Contractual interest rate (r_c)**	**Bond No.**	**Term to maturity (T_0)**	**Contractual interest rate (r_c)**
1	2	0.0%	7	2	1.5%	13	2	3.0%
2	4	0.0%	8	4	1.5%	14	4	3.0%
3	7	0.0%	9	7	1.5%	15	7	3.0%
4	10	0.0%	10	10	1.5%	16	10	3.0%
5	15	0.0%	11	15	1.5%	17	15	3.0%
6	20	0.0%	12	20	1.5%	18	20	3.0%

[a] Assume that coupon payments are made annually.

TABLE II

FINAL WEIGHTS AND BIASES OF NETWORKS I, II AND III, RESPECTIVELY.

Weights and Biases / **Network**	b^o	h	b_h^{H}	w_h^{o}	w_{h1}^{H}	w_{h2}^{H}	w_{h3}^{H}
I	100.4744	1	-0.1689	15.1206	-0.0347	-32.7223	18.8396
		2	-1.3535	-34.3660	0.0544	-36.8286	28.9551
		3	-2.1615	5.6589	0.0988	43.3354	16.8267
		4	1.1698	-21.9999	-0.0643	-36.4188	53.6646
II	93.6583	1	0.4510	-23.3874	-0.0571	-33.4648	71.8090
		2	0.8413	36.9871	-0.0467	32.9078	-10.5926
		3	-1.1572	-10.4621	0.0699	45.5792	43.2855
		4	1.2874	-9.2684	-0.0458	17.5685	-87.3147
III	104.8248	1	0.7832	-14.0352	-0.0519	-27.1299	49.3836
		2	1.3108	-16.7297	-0.0571	-27.3748	14.5874
		3	-1.5287	-30.1819	0.0631	-37.1108	33.3026
		4	-0.6010	13.1504	-0.0524	-34.9042	36.9149

Note: b^o denotes the bias of the output node; b_h^{H} denotes the bias of h^{th} hidden node; w_h^{o} denotes the weight between the output node and the h^{th} hidden node; and w_{hi}^{H} denotes the weight between the h^{th} hidden node and the i^{th} input node.

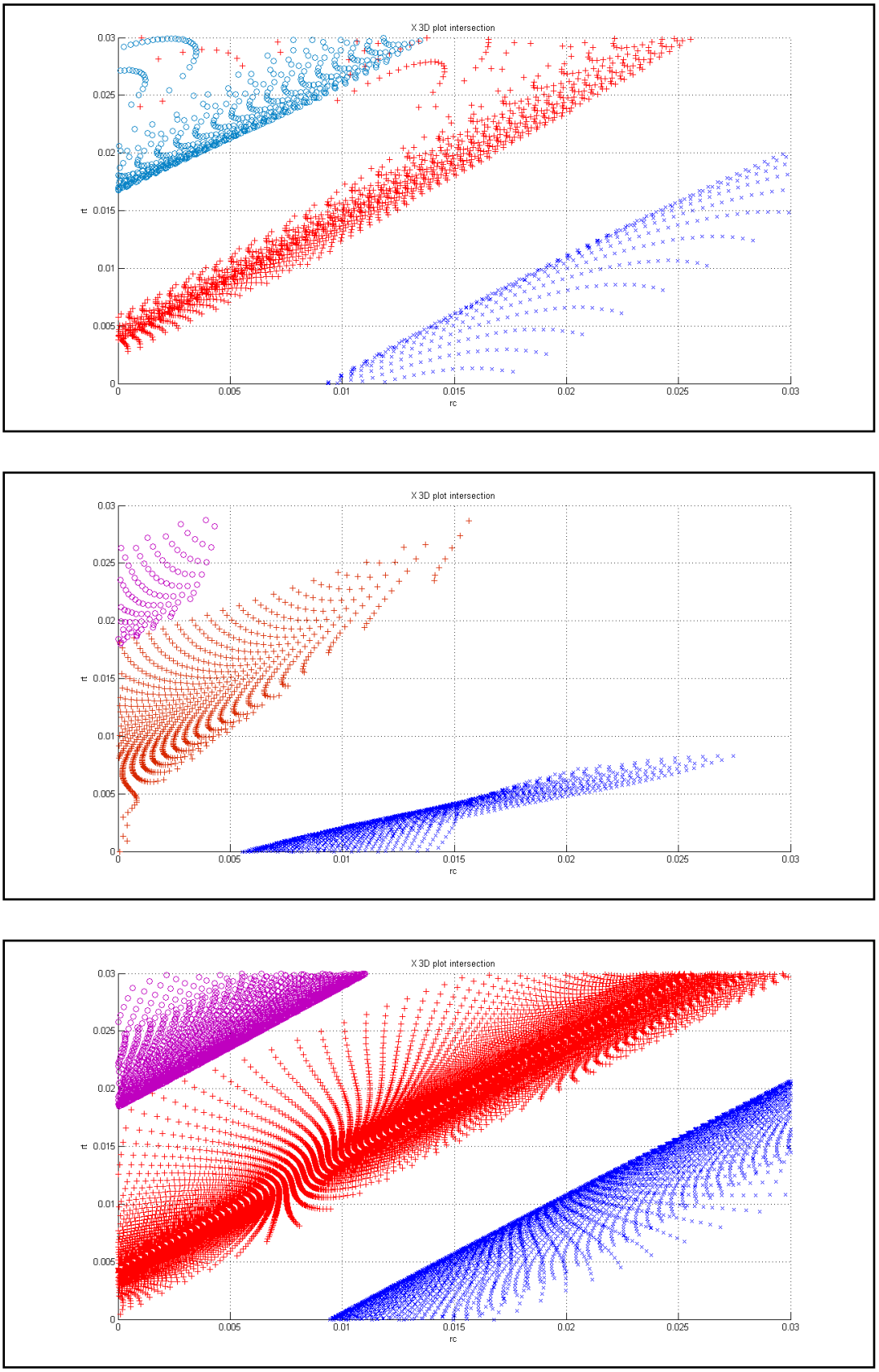


Fig. 1. Preimage graphs along the r_t and r_c plane for network I, II, III (from top to bottom), respectively.

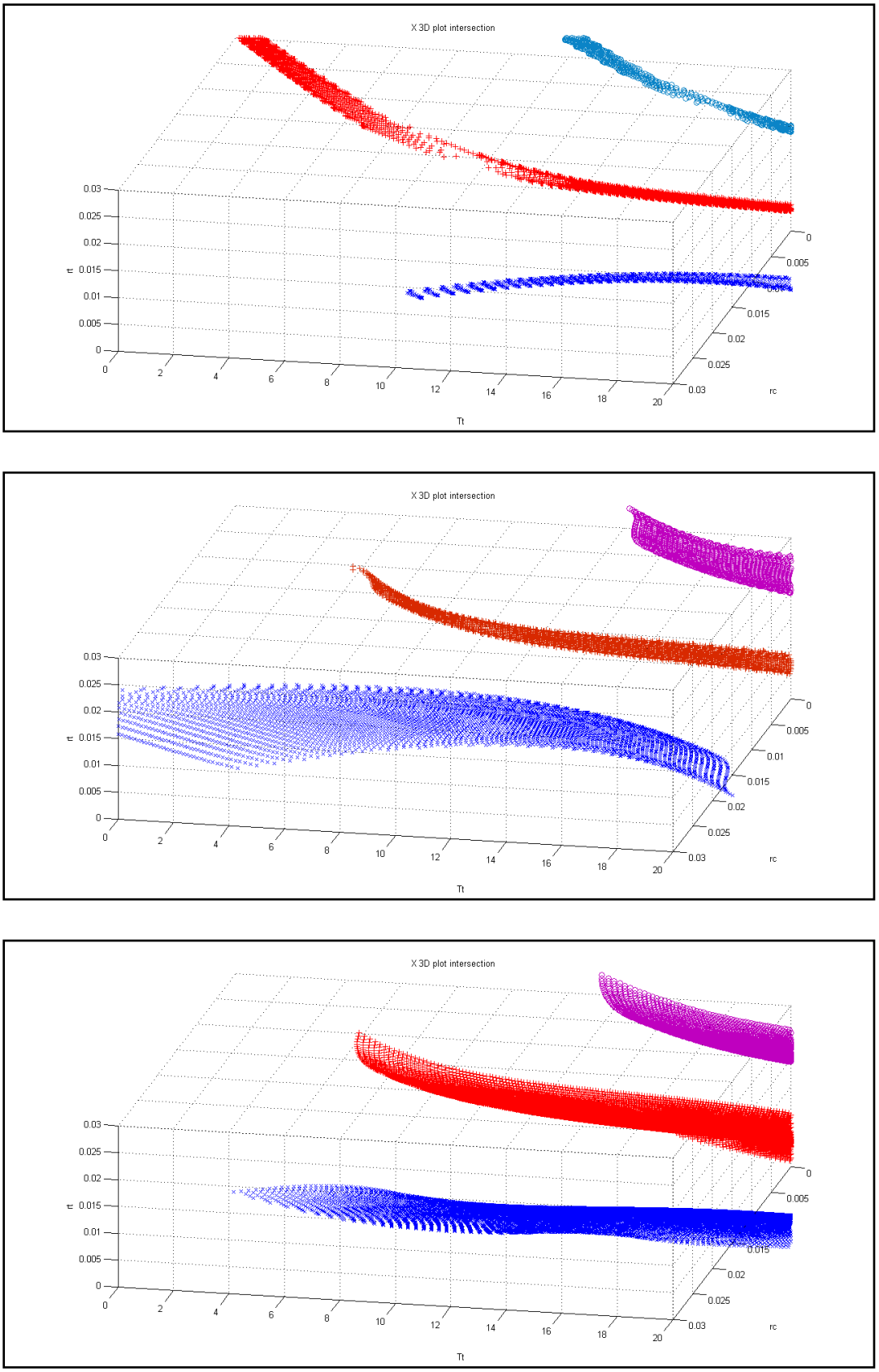


Fig. 2. Preimages graphs along the plane of r_t vs. T_t for network I, II, III (from top to bottom), respectively.