行政院國家科學委員會專題研究計畫　成果報告

使用倒傳遞神經網路之資料挖掘方法論

計畫類別：個別型計畫
計畫編號：NSC94-2416-H-004-017-
執行期間：94 年 08 月 01 日至 95 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

報告類型：精簡報告
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 95 年 8 月 8 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

使用倒傳遞神經網路之資料挖掘方法論

計畫類別：■ 個別型計畫　　□ 整合型計畫
計畫編號：NSC － 94 － 2416 － H － 004 － 017
執行期間：94 年 8 月 1 日 至 95 年 7 月 31 日

計畫主持人：蔡瑞煌
共同主持人：
計畫參與人員：

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　95 年 7 月 30 日

目錄

使用倒傳遞神經網路之資料挖掘方法論

中文摘要

本研究乃探索如何從 3 層順傳遞類神經網路（three-layer feed-forward Neural Networks）裡挖掘規則和知識，進而更新使用者之知識。

Abstract

This study proposes a rule/feature extracting methodology regarding the decision making process for the user with some domain expertise, a process that can update the user's prior belief of his/her interested feature in a specific interested area. Within the process, there are mechanisms designed for extracting relevant (nonlinear regression) rules from an odd number of well-trained three-layer feed-forward neural networks, identifying their associated features, and updating the corresponding prior belief. Instead of the data analysis, the mathematical programming analysis is adopted here to identify each rule premise and its associated feature. With a bond pricing example, we verify the proposed methodology and document that the proposed methodology is effective for (financial or social scientific) applications in which there are few or no observations within a certain region of the interested area.

Keyword：three-layer feed-forward neural networks, rules, features, prior belief.

1 Introduction

Extracting knowledge from a well-trained neural network is a difficult but significant issue attributed to that the corresponding mathematical representation of the relations between the explanatory variables and the response is often nonlinear and complicated. Nevertheless, developing algorithms to extract rules from well-trained neural networks has been a prevalent topic [1][12]. The significance of these applications may be especially pronounced for exploring new financial market or new financial instruments.

From the literature related to extracting rules from the trained neural network [9] [11][13][7][8], the extracted (regression) rules typically take the following syntax:

$$\text{If } (\mathbf{x} \in \text{the } k^{th} \textbf{ region}), \text{ then } (y' = f_k(\mathbf{x})), \quad (1)$$

where $\mathbf{x}$ is the vector of independent variables and y' is the approximated value of the dependent variable via the $f_k(\mathbf{x})$ function. In equation (1), the expression of $\mathbf{x} \in$ the k^{th} **region** is the premise for the rule to be applied. To identify the region depicted in the premise of a single rule, most previous work implement data analysis on the training or generated data set. The generated data set contains data instances yielded from the trained network. To extract a comprehensible multivariate polynomial representation in $f_k(\mathbf{x})$, the activation function of each hidden node is approximated by an approximation function, which may be a piecewise linear function [8] or a multivariate polynomial function with the power not be restricted to be an integer [7]. The approach of multivariate polynomial rules is analogous to the traditional statistical approach of parametric regression, which strips away nonessential details.

In this study, we propose and verifies a decision making process for users with domain expertise, a process that can update the user's prior belief of his/her interested feature. Within the process, three proposed mechanisms are adopted for extracting comprehensible multivariate polynomial representations from the well-trained three-layer layered feed-forward neural network and further examining the extracted rules to gain, for example, the feature of first order partial differential relation of the problem domain. With the features extracted from some odd number of well-trained networks, we further propose a procedure for updating the user's prior belief of his/her interested feature.

There are three distinctive characteristics of this study. First, the problem we are interested is a nonlinear regression problem with continuous variables. With this, our study differs from these classification studies with linear setting [10][2].

Second, the training data, which typically are the historical financial market observations, are frequently perturbed by some external forces and are thus contaminated with unknown noises. Accordingly, multiple well-trained networks are incorporated to enhance the accuracy of the extracted rules and features.

Third, instead of the data analysis, the mathematical programming analysis is adopted here to identify the rule premise and justify the extracted feature. Specifically, with respect to each trained network, the interested area is partitioned into disjoint regions for each of which there is a comprehensible multivariate polynomial representation of the relationship between the explanatory variables and the response. The mathematical programming analysis is used to identify the extant rule premise. With data analysis for extracting rules or justifying extracted features, the number of (training or generated) data instances is always finite and thus the resulted

rule premise covers merely discrete points. The feature exhibited at each extant region is also identified through solving a corresponding mathematical programming problem accompanying the comprehensible multivariate polynomial representation corresponding to that region.

2 The Proposed Process for Updating The User's Prior Belief of The Interested Feature in The Interested Area

The proposed process for updating the user's prior belief of the interested feature in the interested area is applied to any regression problem which has continuous variables and nonlinear requirement and has been coped with three-layer feed-forward neural networks. y denotes value of the network output, x_i denotes the i^{th} explanatory variable, with i from 1 to m, m is the number of input nodes, and $\mathbf{x}^T \equiv (x_1, x_2, \dots, x_m)$. ${}_2\mathbf{w}_j^T \equiv ({}_2w_{j1}, {}_2w_{j2}, \dots, {}_2w_{jm})$ stands for the weights between the j^{th} hidden node and the input layer, with j from 1 to p, where p is the number of used hidden nodes, and ${}_3\mathbf{w}^T \equiv ({}_3w_1, {}_3w_2, \dots, {}_3w_p)$ for the weights between the output node and all hidden nodes. ${}_2\theta_j$ is the bias of the j^{th} hidden node and ${}_3\theta$ is the bias of the output node. Hereafter, characters in bold represent column vectors and the subscript T indicates transposition. The activation function $tanh(t)$ is used in all hidden nodes and the linear activation function is used in the output node. Namely, for the c^{th} sample ${}_c\mathbf{x}$, the activation value of the j^{th} hidden node ${}_ch_j$ and the output value ${}_cy$ are computed as ${}_ch_j = tanh({}_2\mathbf{w}_j^T {}_c\mathbf{x} + {}_2\theta_j)$ and ${}_cy = \sum_{j=1}^{p} {}_3w_j\, {}_ch_j + {}_3\theta$, respectively

Below are four assumptions regarding the proposed prior-belief-updating process:

Assumption 1: The user should have an odd number of obtained networks, each of which has an acceptable forecast performance. With this assumption, the proposed process focuses upon updating the user's prior belief of the interested feature in the interested area via the obtained acceptable networks.

Assumption 2: The user should have a list of interested features, each of which may be either existing or absent in the literature, and the associated interested areas. Furthermore, based upon the user's knowledge and preference, each interested feature is either convinced or unconvinced. Namely, a feature belongs to one of the following four groups: (i) the convinced existing ones, (ii) the unconvinced existing ones, (iii) the convinced absent ones, and (iv) the unconvinced absent ones. The user is not interested in the information extracted from the obtained networks about any unconvinced absent feature, but the unconvinced existing feature or the convinced absent feature. On the other hand, the convinced existing feature may serve to determine if the obtained networks and the adopted approximation function are suitable for the prior-belief-updating process. Therefore, the listed interested features include the convinced existing ones, the unconvinced existing ones, and the convinced absent ones.

Assumption 3: There exists a (computer) round-off effect in the *tanh* function and $|tanh(x)| = 1$ when $|x| > \psi$. In other words, it is impossible to have $tanh(x) \in (-1, -tanh(\psi)) \cup (tanh(\psi), 1)$. The constant ψ may be different due to different level of precision of the computer. For instance, in our PC simulation environment with Pentium 4, ψ is (approximately) 19.0615474653985.

Assumption 4: As mentioned in [7] and [8], an approximation of the activation function is usually required for extracting comprehensible rules from the trained network. We here further

assume that, with respect to the interested feature that covers the ν^{th} order (partial) differential relation, the user should adopt a piecewise polynomial approximation function whose maximal power is (ν+1) to approximate the *tanh* function.

Table 1 presents the proposed prior-belief-updating process. Based upon Assumption 2, there is a list of interested features and their associated interested areas. With respect to each interested feature, in Table 1, there are four steps designed for updating the user's prior belief. Step 1 adopts the rule-extracting mechanism to divide the interested area into several disjoint regions and obtain the comprehensible polynomial representation of each disjoint region. The (extracted) rule should be a specific comprehensible multivariate polynomial representation of the relationship between the explanatory variables and the response when the vector of explanatory variables x is in some region. Step 2 uses the feature-extracting mechanism to identify the feature exhibited at each extant disjoint region. If a disjoint region exhibits a null result, the proposed partition mechanism in Step 3 further splits that region into two (sub-) regions, at each of which a specific feature definitely exhibits. The prior-belief-updating procedure is proposed in Step 4.

Note that Assumption 1 states that the user should have some odd number of obtained acceptable networks. The adoption of multiple networks serves as a stabilization measure to the conclusion of extracted features.

For each obtained network, the rule-extracting mechanism divides the interested area into several regions and the feature-identifying mechanism identifies a specific feature at each region, a feature that conforms or deviates with the interested feature. The division of the interested area and the features exhibited in the divided regions may be different regarding to different networks. Let the consistent area of a network regarding the interested feature be the union of regions exhibiting the conformed feature. With respect to each interested feature, the following three types of areas regarding all obtained network should be identified:

(i) The consistent area of the interested feature (hereafter the CA) is the region at which all obtained networks exhibit unanimously the conformed feature. Namely, the CA is the intersection of the corresponding consistent areas of all obtained networks.

(ii) The accordant inconsistent area of the interested feature (hereafter the AIA) is the area in which all networks unanimously exhibit the deviated feature.

(iii) The discordant area of the interested feature (hereafter the DA) is the area in which some networks exhibit the conformed feature and the others exhibit the deviated feature.

After obtaining these three types of areas, the user can follow the procedure shown in Figure 2 to update his/her prior belief of the interested feature in the interested area. The result could be as follows:

(i) The user's prior belief of the interested feature is strengthened in the CA. Namely, the user's posterior belief of the interested feature in the CA is stronger than his/her prior belief.

(ii) The user's prior belief of the interested feature is weakened in the AIA. Namely, the user's posterior belief of the interested feature in the AIA is weaker than his/her prior belief.

(iii) If the CA is the whole interested area, then the user's prior belief of the interested feature in the interested area is strengthened. Namely, the user's posterior belief of the interested feature in the interested area is stronger than his/her prior belief.

(iv) If the AIA is the whole interested area, then the user's prior belief of the interested feature in the interested area is weakened. Namely, the user's posterior belief of the interested feature in the

interested area is weaker than his/her prior belief.

(v) For each DA, the voting mechanism is used to derive a (plausible) feature. Specifically, the number of networks with the exhibited features $_2F_1$ and $_2F_2$ on each DA are exhaustively counted. The left subscript of 2 corresponds to a second order differential relation depicted in the interested feature; the right subscript of 0, 1, or 2 corresponds to a null result, a positive differential relation, or a negative differential relation, respectively. The percentage of these two numbers reflects either supportive or disconfirming strength on updating the user's prior belief in the DA.

(vi) If the convinced existing feature is supported by the conformed feature in the whole interested area, the user may claim that the adopted approximation function is acceptable. Otherwise, the user may claim either that the adopted approximation function g is unacceptable or that the obtained networks are not useful in the rule/feature extraction.

3 Conclusion and Managerial implications

This study adds to the literature by introducing a methodology for extracting rule/feature from three-layer feed-forward networks and updating the user's prior belief based upon the extracted features. Mechanisms are developed to extract rules and features within each relevant region of the user's interested area. The mathematical programming analysis, instead of a data analysis, is adopted to identify the premise of each multivariate polynomial rule. The mathematical programming analysis is also adopted in the identification of the feature exhibited in each separate region. With the mathematical programming analysis, these mechanisms can provide both rules and features in the region with few or no data observations. Furthermore, the prior-belief-updating procedure accompanied with multiple networks helps average out noises and thus mitigate inaccurate estimates of individual networks. Moreover, the proposed methodology may identify the relationship which differs within differential domains. For example, from the perspective of the bond literature, the magnitude of the discount or premium for fixed rated bonds decreases as its life gets shorter.

Accordingly, we can extract any higher order nonlinear regression rules and corresponding features from trained neural networks with the *tanh* activation function in hidden nodes. The extracted feature could describe the differential relation that a social scientific application concerns. The proposed methodology could be useful in exploring newly issued financial instruments with limited data instances, and this is a future work.

On the other hand, issues worthy of future studies include the application of the proposed methodology to real world data, the elimination of redundant constraints from the premise of a rule, and the integration of extracted rules.

計畫成果自評：

此研究計畫成果豐碩，已被送到相關研討會和期刊審查。其延伸之研究亦在進行中。

References

1. 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.
2. 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.
3. Luenberger, D. (1984). *Linear and Nonlinear Programming*. Reading, Reading, MA: Addison-Wesley.
4. Malkiel, B. G. (1962). Expectations, bond prices, and the term structure of interest rates. Quarterly Journal of Economics, 76(2), 197-218.
5. MathWorks, Inc. (2005). Optimization Toolbox For Use with MATLAB, accessed July 28, 2005, available at http://www.mathworks.com/access/helpdesk/help/pdf_doc/optim/optim_tb.pdf.
6. Rumelhart, D. E., Hinton, G. E., & Williams, R. (1986). Learning internal representation by error propagation. Parallel Distributed Processing. Cambridge, MA: MIT Press, 1, 318-362.
7. Saito, K., & Nakano, R. (2002). Extracting regression rules from neural networks. Neural Network, 15(10), 1297-1288.
8. 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.
9. Setiono, R., & Liu, H. (1997). NeuroLinear: From neural networks to oblique decision rules. Neurocomputing, 17(1), 1-24.
10. Setiono, R., &, Liu, H. (1996). Symbolic representation of neural networks. IEEE Computer. 29(3), 71–77.
11. Taha, I. A., & Ghosh, J. (1999). Symbolic interpretation of artificial neural networks. IEEE Transactions on Knowledge and Data Engineering, 11(3), 448-463.
12. 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.
13. 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, 3, 401-406.

Table 1. The proposed rule-feature-extracting procedure for a three-layer feed-forward neural network.

Step 1: Apply the rule-extracting mechanism to divide the interested area into several disjoint regions, associated with each of which there is a specific polynomial representation approximating the response.
Step 2: Apply the feature-extracting mechanism to identify the exhibited feature in each extant disjoint region.
Step 3: Apply the partition mechanism to split any disjoint region with a null result in Step 2 into two (sub-) regions, each of which exhibits exactly one specific feature.
Step 4: Update the prior belief.