行政院國家科學委員會專題研究計畫　成果報告

學習演算法與學習理論

計畫類別：個別型計畫
計畫編號：NSC93-2416-H-004-015-
執行期間：93 年 08 月 01 日至 94 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：錡慧珊

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 94 年 6 月 30 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

學習演算法與學習理論

計畫類別：■ 個別型計畫　　□ 整合型計畫
計畫編號：NSC － 93 － 2416 － H － 004 － 015
執行期間：93 年 8 月 1 日 至 94 年 7 月 31 日

計畫主持人：蔡瑞煌
共同主持人：
計畫參與人員：錡慧珊

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　94 年 6 月 30 日

目錄

Learning of Neural Networks and Theories of Learning

中文摘要

本研究乃探索 3 層順傳遞類神經網路（3-layer feed-forward Neural Networks）之行為潛能（potential），以能了解網路之知識內涵（knowledge）以及在學習過程中知識內涵之調整方式。

Abstract

This research explores the general potential of a 3-layer feed-forward neural network to identify its embedded knowledge and the knowledge adjustment within the learning process.

Keyword：3-layer feed-forward neural networks, potential, generalized delta rule, cramming.

1. The Black Box Dilemma

Although the learning process of human beings is still too complicated to be understood clearly nowadays, the work in Neural Networks is beginning to contribute to our further understanding of the learning process of human beings.

Hebb's work in the neurophysiology has inspired the idea of computer modeling of brain cell activities. Hebb claims that the connection between two cells that are active simultaneously will be strengthened. The Hebbian rule is expressed in equation (1), where Δw_{oi} is the change of strength or weight of a connection between the input and output units, η is a constant that reflects learning rate, a_i is the activation value level of the input unit, and a_o is the activation value level of the output unit.

$$\Delta w_{oi} = \eta\, a_o\, a_i \qquad (1)$$

Later, Rumelhart, et al. in (Rumelhart, Hinton & Williams, 1986) propose another learning rule, the generalized delta rule that generalizes the delta rule, for layered feed-forward neural networks. The fundamental form of the delta rule is in equation (2), where t is the target activation value level of the output unit.

$$\Delta w_{oi} = \eta\, (t - a_o)\, a_i \qquad (2)$$

Unfortunately, from then on, learning algorithms proposed in neural networks to change the nature of connections are logical but arbitrary, and there are no researches exploring the analogy between theories of learning in human beings and the learning in neural networks. Moreover, the dilemma of the network being a black box discourages lots of researchers who are interested in applying neural networks to real world for discovering some knowledge.

To simplify the presentation, without lose of the generality, the neural network considered in this study is a feed-forward neural network with one hidden layer and one output node. We first explore network's general potential that engages itself in a class of behaviors. This exploration tries to derive a concept for discovering the knowledge embedded in the network. We then use this concept to examine the learning process of the layered feed-forward neural network in the viewpoint of knowledge adjustment.

The notations used here are as follows. We denote the output value of the neural network by y, and the activation value of the i^{th} hidden node by a_i, with i from 1 to p. $\mathbf{x}^T \equiv (x_1, x_2, \ldots, x_m)$ where x_j is the j^{th} input element, with j from 1 to m. ${}_2w_{i0}$ stands for the bias of the i^{th} hidden node; ${}_2\mathbf{w}_i{}^T \equiv ({}_2w_{i1}, {}_2w_{i2}, \ldots, {}_2w_{im})$ for the weights between the i^{th} hidden node and input layer, with i from 1 to p; ${}_3w_0$ for the bias of the output node; ${}_3\mathbf{w}^T \equiv ({}_3w_1, {}_3w_2, \ldots, {}_3w_p)$ for the weights between the output node and all hidden nodes; ${}_2\mathbf{w}^T \equiv ({}_2\mathbf{w}_1{}^T, {}_2\mathbf{w}_2{}^T, \ldots, {}_2\mathbf{w}_p{}^T)$; $\mathbf{w}^T \equiv ({}_2\mathbf{w}^T, {}_3\mathbf{w}^T)$. Character in bold represents a column vector, a matrix or a set, and the superscript T indicates the transposition.

Here we assume ${}_3\mathbf{w}$ and ${}_2\mathbf{w}$ are non-zero vectors. Thus ${}_2\mathbf{W} \equiv ({}_2\mathbf{w}_1, {}_2\mathbf{w}_2, \ldots, {}_2\mathbf{w}_p)^T$ is a non-zero matrix.

Suppose that the $tanh(t)$ activation function defined in equation (3) is adopted in all hidden nodes and the linear activation function is used in the output node. That is, given the c^{th} stimulus ${}_c\mathbf{x}$, the activation value of the i^{th} hidden node ${}_ca_i \equiv tanh({}_2w_{i0} + \sum_{j=1}^{m} {}_2w_{ij}\, {}_cx_j)$ and the activation value of the output node ${}_cy = f({}_c\mathbf{x}) \equiv {}_3w_0 + \sum_{i=1}^{p} {}_3w_i\, {}_ca_i$.

$$tanh(t) \equiv \frac{e^t - e^{-t}}{e^t + e^{-t}} \qquad (3)$$

2. The Potentials Of Layered Feed-Forward Neural Networks

The network's general potential to engage itself in a class of behaviors can be observed from $\mathbf{f}^{-1}$, the inverse function of f.

A layered feed-forward neural network is generally viewed as a mechanism that provides a nonlinear mapping, $y = f(\mathbf{x})$, between the independent variables x_j's and the dependent variable y, where f is a nonlinear function derived from a given data set of samples $\{({}_1\mathbf{x}, {}_1t), \ldots, ({}_N\mathbf{x}, {}_Nt)\}$ with ${}_ct$, the observed value of y corresponding to ${}_c\mathbf{x}$. f can be viewed as the composite $g \circ \mathbf{h}$ where $\mathbf{h}$ is defined by $(\mathbf{h}(\mathbf{x}))_i$, the i^{th} component of $\mathbf{h}(\mathbf{x})$, $\equiv tanh({}_2w_{i0} + \sum_{j=1}^{m} {}_2w_{ij}\, x_j)$ for every $i \in \{1, 2, \ldots, p\}$, $a_i \equiv (\mathbf{h}(\mathbf{x}))_i$, and g is defined by $g(\mathbf{a}) \equiv {}_3w_0 + \sum_{i=1}^{p} {}_3w_i\, a_i$. The hidden-layer set is $(-1, 1)^p$ because the *tanh* activation function is used here. Unfortunately, from the expression of f, it is difficult to discover the associated general potentials that engages itself in classes of behaviors. Here we count on the inverse function $\mathbf{f}^{-1}(y)$, the set of all elements of R^m whose images under f are y.

According to (Tsaih & Lin, 2004), $\mathbf{g}^{-1}(y)$, the set of all elements of $(-1, 1)^p$ whose images under g are y, equals $\{\mathbf{a} | \sum_{i=1}^{p} {}_3w_i\, a_i = y - {}_3w_0, \mathbf{a} \in (-1, 1)^p\}$ and $\mathbf{f}^{-1}(y)$ equals $\{\mathbf{x} | {}_2\mathbf{W}\,\mathbf{x} = \boldsymbol{\omega}(\mathbf{a})$ with all $\mathbf{a} \in \mathbf{A}_{nv}(y)\}$ where the function $\boldsymbol{\omega} : (-1, 1)^p \rightarrow R^p$ is defined by $\boldsymbol{\omega}(\mathbf{a}) \equiv (\omega_1(a_1), \omega_2(a_2), \ldots, \omega_p(a_p))^{\mathrm{T}}$ with $\omega_i(a_i) \equiv tanh^{-1}(a_i) - {}_2w_{i0}$ for every i and $tanh^{-1}(x) \equiv 0.5\, ln(\frac{1+x}{1-x})$, $\mathbf{A}_{nv}(y) \equiv \{\mathbf{a} | \mathbf{a} \in \mathbf{g}^{-1}(y), rank({}_2\mathbf{W} : \boldsymbol{\omega}(\mathbf{a})) = rank({}_2\mathbf{W})\}$, and $rank({}_2\mathbf{W})$ is the rank of ${}_2\mathbf{W}$.

Take Network I in Figure 1 as an illustration of $\mathbf{f}^{-1}$ and associated potential. The computer's round-off effect occurs on the area of $\{\mathbf{x} | 18x_1 + 18x_2 - 18 > \psi\} \cup \{\mathbf{x} | 18x_1 + 18x_2 - 18 < -\psi\}$.[1] That is, y equals -1 when $\mathbf{x} \in \{\mathbf{x} | x_1 + x_2 > 1 + \frac{\psi}{18}\}$; 1 when $\mathbf{x} \in \{\mathbf{x} | x_1 + x_2 < 1 - \frac{\psi}{18}\}$. The range of non-vague[2] y is $\{-1\} \cup [-tanh(\psi), tanh(\psi)] \cup \{1\}$. For any non-vague y, $\mathbf{g}^{-1}(y)$ equals $\{a_1 \mid a_1 = -y\}$. $\mathbf{f}^{-1}(-1)$ equals $\{\mathbf{x} | x_1 + x_2 > 1 + \frac{\psi}{18}\}$, which consists of a open half space; for any $y \in [-tanh(\psi), tanh(\psi)]$, $\mathbf{f}^{-1}(y)$ equals $\{\mathbf{x} | 18x_1 + 18x_2 = tanh^{-1}(-y) + 18\}$, which consists of a line; $\mathbf{f}^{-1}(1)$ equals $\{\mathbf{x} | x_1 + x_2 < 1 - \frac{\psi}{18}\}$, which consists of a open half space.

The network with two hidden nodes can have various potentials. When either the associated ${}_3w_1$ or ${}_3w_2$ value is zero, the associated potential manages the mapping consisted in $\{a_1\}$ or $\{a_2\}$ and the knowledge within the potential is a linear one on the input space. When both of ${}_3w_1$ and ${}_3w_2$ values are non-zero, the associated potential manages the mapping consisted in $\{(a_1, a_2)\}$ and a different value of $rank({}_2\mathbf{W})$ results in a different associated nonlinear potential.

Take Network II and Network III in Figure 1 as illustrations of having nonlinear potential. For the a_1 and a_2 nodes of the II Network, the round-off effect occurs on areas of $\{\mathbf{x} | 18x_1 + 18x_2 - 18 > \psi\} \cup \{\mathbf{x} | 18x_1 + 18x_2 - 18 < -\psi\}$and of $\{\mathbf{x} | -18x_1 - 18x_2 - 18 > \psi\} \cup \{\mathbf{x} | -18x_1 - 18x_2 - 18 < -\psi\}$, respectively. Thus, a_1 equals 1 and a_2 equals -1 when $\mathbf{x} \in \{\mathbf{x} | x_1 + x_2 > 1 + \frac{\psi}{18}\}$; a_1 equals -1

---

[1] Assume $|tanh(x)| = 1$ when $|x| > \psi$ due to the computer's round-off effect. Given the precision of the computer, ψ is a fixed constant. Therefore, the range of $tanh(x)$ is $\{-1\} \cup [-tanh(\psi), tanh(\psi)] \cup \{1\}$. In other words, it is impossible to have $tanh(x) \in (-1, -tanh(\psi)) \cup (tanh(\psi), 1)$. In (Tsaih, 2004), the $\mathbf{f}^{-1}(y)$ is derived assuming that there is no computer's round-off effect. In this article, however, the $\mathbf{f}^{-1}(y)$ is described assuming that there is the computer's round-off effect.

[2] The output value y is non-vague if it can be obtained with the current network.

and a_2 equals 1 when $\mathbf{x} \in \{\mathbf{x}| x_1 + x_2 < -1-\frac{\psi}{18}\}$; both a_1 and a_2 equal -1 when $\mathbf{x} \in \{\mathbf{x}| -1+\frac{\psi}{18} < x_1 + x_2 < 1-\frac{\psi}{18}\}$. In sum, the range of non-vague y is $\{-1\} \cup [-tanh(\psi), tanh(\psi)] \cup \{1\}$. For any non-vague y, $\mathbf{g}^{-1}(y)$ equals $\{\mathbf{a} \mid -a_1 - a_2 = y + 1, a_1 \in \{-1, [-tanh(\psi), tanh(\psi)], 1\}, a_2 \in \{-1, [-tanh(\psi), tanh(\psi)], 1\}\}$. $\mathbf{f}^{-1}(-1)$ equals $\{\mathbf{x}| x_1 + x_2 > 1+\frac{\psi}{18}\} \cup \{\mathbf{x}| x_1 + x_2 < -1-\frac{\psi}{18}\}$, which consists of two parallel open half spaces; for any $y \in [-tanh(\psi), tanh(\psi)]$, $\mathbf{f}^{-1}(y)$ equals $\{\mathbf{x}| x_1 + x_2 = 1-\frac{tanh^{-1}(y)}{18}\} \cup \{\mathbf{x}| x_1 + x_2 = -1-\frac{tanh^{-1}(y)}{18}\}$, which consists of two parallel lines; $\mathbf{f}^{-1}(1)$ equals $\{\mathbf{x}| -1+\frac{\psi}{18} < x_1 + x_2 < 1-\frac{\psi}{18}\}$.

As for the a_1 and a_2 nodes of Network III in Figure 1, the computer's round-off effect occurs on areas of $\{\mathbf{x}| 18x_1 + 18x_2 - 18 > \psi\} \cup \{\mathbf{x}| 18x_1 + 18x_2 - 18 < -\psi\}$ and $\{\mathbf{x}| -18x_1 + 18x_2 - 18 > \psi\} \cup \{\mathbf{x}| -18x_1 + 18x_2 - 18 < -\psi\}$, respectively. Thus, a_1 equals 1 when $\mathbf{x} \in \{\mathbf{x}| x_1 + x_2 > 1+\frac{\psi}{18}\}$ and -1 when $\mathbf{x} \in \{\mathbf{x}| x_1 + x_2 < 1-\frac{\psi}{18}\}$; a_2 equals 1 when $\mathbf{x} \in \{\mathbf{x}| x_1 - x_2 > -1+\frac{\psi}{18}\}$ and -1 when $\mathbf{x} \in \{\mathbf{x}| x_1 - x_2 < -1-\frac{\psi}{18}\}$. In sum, the range of non-vague y is $\{-3\} \cup [-tanh(\psi)-2, tanh(\psi)] \cup \{1\}$. For any non-vague y, $\mathbf{g}^{-1}(y)$ equals $\{\mathbf{a} \mid -a_1 - a_2 = y + 1, a_1 \in \{-1, [-tanh(\psi), tanh(\psi)], 1\}, a_2 \in \{-1, [-tanh(\psi), tanh(\psi)], 1\}\}$. $\mathbf{f}^{-1}(-3)$ equals $\{\mathbf{x}| x_1 + x_2 > 1+\frac{\psi}{18}, x_1 - x_2 < -1-\frac{\psi}{18}\}$; for any $y \in [-tanh(\psi)-2, -2tanh(\psi)-1)$, $\mathbf{f}^{-1}(y)$ equals $\{\mathbf{x}| x_1 + x_2 = 1-\frac{tanh^{-1}(y+2)}{18}, x_1 - x_2 < -1-\frac{\psi}{18}\} \cup \{\mathbf{x}| x_1 - x_2 = -1-\frac{tanh^{-1}(y+2)}{18}, x_1 + x_2 > 1+\frac{\psi}{18}\}$, which consists of two half-lines; for any $y \in [-2tanh(\psi)-1, tanh(\psi)-2]$, $\mathbf{f}^{-1}(y)$ equals $\{\mathbf{x}| 18x_1 + 18x_2 = tanh^{-1}(a_1) + 18, -18x_1 + 18x_2 = tanh^{-1}(-a_1 - y - 1) + 18, a_1 \in [2tanh(\psi)-1, tanh(\psi)]\} \cup \{\mathbf{x}| x_1 + x_2 = 1-\frac{tanh^{-1}(y+2)}{18}, x_1 - x_2 < -1-\frac{\psi}{18}\} \cup \{\mathbf{x}| x_1 - x_2 = -1-\frac{tanh^{-1}(y+2)}{18}, x_1 + x_2 > 1+\frac{\psi}{18}\}$, which consists of a curve segment and two half-lines; for any $y \in (tanh(\psi)-2, -1)$, $\mathbf{f}^{-1}(y)$ equals $\{\mathbf{x}| 18x_1 + 18x_2 = tanh^{-1}(a_1) + 18, -18x_1 + 18x_2 = tanh^{-1}(-a_1 - y - 1) + 18, a_1 \in (-tanh(\psi), tanh(\psi))\}$, which consists of a curve segment; $\mathbf{f}^{-1}(-1)$ equals $\{\mathbf{x}| x_1 = \frac{\tanh^{-1}(a_1)}{18}, x_2 = 1, a_1 \in [-tanh(\psi), tanh(\psi)]\} \cup \{\mathbf{x}| x_1 + x_2 > 1+\frac{\psi}{18}, x_1 - x_2 > -1+\frac{\psi}{18}\} \cup \{\mathbf{x}| x_1 + x_2 < 1-\frac{\psi}{18}, x_1 - x_2 < -1-\frac{\psi}{18}\}$; for any $y \in (-1, -tanh(\psi))$, $\mathbf{f}^{-1}(y)$ equals $\{\mathbf{x}| 18x_1 + 18x_2 = tanh^{-1}(a_1) + 18, -18x_1 + 18x_2 = tanh^{-1}(-a_1 - y - 1) + 18, a_1 \in [-tanh(\psi), tanh(\psi)]\}$, which consists of a curve segment; for any $y \in [-tanh(\psi), 2tanh(\psi)-1]$, $\mathbf{f}^{-1}(y)$ equals $\{\mathbf{x}| 18x_1 + 18x_2 = tanh^{-1}(a_1) + 18, -18x_1 + 18x_2 = tanh^{-1}(-a_1 - y - 1) + 18, a_1 \in [-tanh(\psi), 1-2tanh(\psi)]\} \cup \{\mathbf{x}| x_1 + x_2 = 1-\frac{tanh^{-1}(y)}{18}, x_1 - x_2 > -1+\frac{\psi}{18}\} \cup \{\mathbf{x}| x_1 - x_2 = -1-\frac{tanh^{-1}(y)}{18}, x_1 - x_2 < 1-\frac{\psi}{18}\}$, which consists of a curve segment and two half-lines; for any $y \in (2tanh(\psi)-1, tanh(\psi)]$, $\mathbf{f}^{-1}(y)$ equals $\{\mathbf{x}| x_1 + x_2 = 1-\frac{tanh^{-1}(y)}{18}, x_1 - x_2 >$

$-1+\frac{\psi}{18}\} \cup \{\mathbf{x}|\ x_1 - x_2 = -1-\frac{tanh^{-1}(y)}{18},\ x_1 - x_2 < 1-\frac{\psi}{18}\}$, which consists of two half-lines; $\mathbf{f}^{-1}(1)$ equals $\{\mathbf{x}|\ x_1 + x_2 < 1-\frac{\psi}{18},\ x_1 - x_2 > 1+\frac{\psi}{18}\}$.

Note that, if the ${}_3w_2$ value becomes zero due to the weight adjustment on Network III, then we get a linear potential like the one associated with Network I.

In closing, when there are more than one hidden nodes, the non-linearity of the $tanh^{-1}$ function and the variety of $rank({}_2\mathbf{W})$ fertilize the diverseness of the associated potential.

3. The Learning Processes Of Layered Feed-Forward Neural Networks

When the $\mathbf{w}$ is specified in the learning stage, $\mathbf{f}^{-1}(y)$ is determined; thus the associated potential is decided and used to assimilate all presented training samples. When the current potential cannot effectively deal with all presented training samples, there is an accommodation that implements a learning algorithm to alter the values of $\mathbf{w}$ and thus tune up the knowledge within the associated potential or form a new potential.

An adjustment of ${}_3\mathbf{w}$ leads to adjustments of $\mathbf{g}^{-1}(y)$ and thus $\mathbf{f}^{-1}(y)$. If the set $\{i: {}_3w_i = 0\}$ isn't changed, the adjustment of $\mathbf{f}^{-1}(y)$ merely alters a_is' weighting in the output value y and thus tunes up the knowledge within the current potential; otherwise, the adjustment of $\mathbf{f}^{-1}(y)$ results in a new potential.

An adjustment of ${}_2\mathbf{w}$ leaves an adjustment of $\mathbf{f}^{-1}(y)$. When the adjustment of ${}_2\mathbf{w}$ does not leave a change of $rank({}_2\mathbf{W})$, the adjustment of $\mathbf{f}^{-1}(y)$ leads to tuning up the knowledge within the associated potential. When the adjustment of ${}_2\mathbf{w}$ leaves a change of $rank({}_2\mathbf{W})$, the (dramatic) adjustment of $\mathbf{f}^{-1}(y)$ leads to forming a new associated potential. However, when values of p and m are fixed, the minimum of m and p is the upper bound of $rank({}_2\mathbf{W})$. No matter what the adjustment of ${}_2\mathbf{w}$ is, it is impossible to increase $rank({}_2\mathbf{W})$ beyond its upper bound.

Take as an illustration the learning process of $\{({}_1\mathbf{x}, {}_1t), ({}_2\mathbf{x}, {}_2t), ({}_3\mathbf{x}, {}_3t), ({}_4\mathbf{x}, {}_4t)\} = \{((1, 1)^T, -1), ((-1, 1)^T, 1), ((1, -1)^T, 1), ((-1, -1)^T, -1)\}$, where a mapping similar with the logical XOR is embedded. When the learning algorithm adopts the generalized delta rule proposed in (Rumelhart, Hinton & Williams, 1986), the learning process is recognized as a weight-tuning process of implementing a series of generalized delta rules. Each implementation of the generalized delta rule results in a tiny adjustment of both ${}_2\mathbf{w}$ and ${}_3\mathbf{w}$. A tiny adjustment of ${}_3\mathbf{w}$ hardly ever changes the set $\{i: {}_3w_i = 0\}$, and usually merely alters a_is' weighting in the output value y. Meanwhile, a tiny adjustment of ${}_2\mathbf{w}$ may lead to one of the following three situations:

1. an alternation without changing $rank({}_2\mathbf{W})$;
2. an alternation with an increase of $rank({}_2\mathbf{W})$;
3. an alternation with a decrease of $rank({}_2\mathbf{W})$.

As for a series of generalized delta rule, a_is' weighting in the output value y is still more likely being altered than the set $\{i: {}_3w_i = 0\}$. Meanwhile, the alteration of ${}_2\mathbf{w}$ may lead to one of the following six situations:

1. a tiny alternation of ${}_2\mathbf{w}$ without changing $rank({}_2\mathbf{W})$;
2. a dramatic alternation of ${}_2\mathbf{w}$ without changing $rank({}_2\mathbf{W})$;
3. a tiny alternation of ${}_2\mathbf{w}$ with an increase of $rank({}_2\mathbf{W})$;
4. a dramatic alternation of ${}_2\mathbf{w}$ with an increase of $rank({}_2\mathbf{W})$;
5. a tiny alternation of ${}_2\mathbf{w}$ with a decrease of $rank({}_2\mathbf{W})$;
6. a dramatic alternation of ${}_2\mathbf{w}$ with a decrease of $rank({}_2\mathbf{W})$.

After learning the following training samples $({}_1\mathbf{x}, {}_1t)$, $({}_2\mathbf{x}, {}_2t)$ and $({}_3\mathbf{x}, {}_3t)$, the network system is tuned to be like Network I in Figure 1 whose current potential deals with the mapping con-

sisted in $\{a_1\}$. The weight-tuning (without changing $rank({}_2\mathbf{W})$) corresponds to a tuning of knowledge within the associated potential.

Then, $({}_4\mathbf{x}, {}_4t)$ comes in, and Network I stuns in the weight-tuning process because any adjustment of weights leads to the same potential which cannot deal with the nonlinear mapping of the presented samples.

If we adopt the cramming mechanism proposed in (Tsaih, 1993 & 2003)[3] to recruit an extra hidden node[4] and get Network II from Network I, a new potential dealing with the mapping consisted in $\{(a_1, a_2)\}$ is formed by modeling on the (existing) potential dealing with $\{a_1\}$. Recruiting extra hidden nodes can increase $rank({}_2\mathbf{W})$ beyond its current upper bound and render the internal representation more complicated. In short, the cramming action corresponds to an initial creation of a new potential by modeling on the current potential.

Assume that the learning process proceeds further with extra training samples and the network system becomes Network III through the weight-tuning mechanism. Such weight-tuning with changing $rank({}_2\mathbf{W})$ corresponds to forming a new potential with encoding new (nonlinear) knowledge in terms of pre-existing potential.

4. Discussions And Future Work

In conclusion, it seems that a series of implementing the generalized delta rule can do modes of tuning and accretion proposed in (Rumelhart & Norman, 1981), and the hidden-node recruiting mechanism can get do the mode of structuring proposed in (Rumelhart & Norman, 1981; Norman, 1982). The discussion of these analogy deserves a future work.

For refining the potential, it is better to implement the weight-tuning mechanism accompanying with a mechanism of pruning hidden nodes. As mentioned in (Tsaih, 1998), the cramming mechanism may add excess hidden nodes that later become irrelevant. The irrelevant hidden nodes are useless with respect to the learning goal; in addition, they deteriorate the generalization of network system. Moreover, more training samples typically lead to more concise information about the knowledge. Therefore, it is necessary to prune irrelevant hidden nodes. This argument needs to be justified in the future. One future work is to examine if activities of adding and pruning neurons really occur in brains.

The outcomes of experiments with the learning procedure proposed in (Tsaih, 1993 & 2003) demonstrate that some training sample sequences cause much more difficulties in the learning process. Another further investigation is to (theoretically or numerically) identify the scenario in which the knowledge-tuning fails to achieve the learning goal and a new potential with encoding new knowledge is requested.

計畫成果自評：

此研究計畫成果豐碩，已被送到 IEEE 期刊審查。其延伸之研究亦在進行中。

References

[1] D. Norman, *Learning and Memory*, W.H. Freeman and Company, NY (1982).

[2] D. Rumelhart, Hinton, G. & Williams, R., Learning Internal Representations By Error

[3] In brief, the learning process proposed in (Tsaih, 1993 & 2003) is as follows: when encountering a new training case, the network system first checks if the knowledge obtained so far could assimilate it. If so, there is no accommodation. If not, the network system may add new hidden nodes to cope with this unfamiliar case; then, the network system does tuning and pruning to integrate the old and new knowledge.

[4] Adding a new hidden node corresponds to rendering the weights (synaptic strengths) between associated neurons in brain non-zero, while pruning a hidden node corresponds to rendering the weights between associated neurons zero.

Propagation. In Rumelhart, D. & McClelland J. (Eds.), *Parallel Distributed Processing*: *Explorations in the Microstructure of Cognition*, Vol. 1. Cambridge, MA: MIT Press (1986).
[3] D. Rumelhart, & Norman, D., Analogical processes in learning. In J. Anderson (Ed.), *Cognitive skills and their acquisition*, Hillsdale, NJ: Erlbaum (1981).
[4] R. Tsaih, *The Softening Learning Procedure*, Mathematical and Computer Modelling, Vol. 18, No. 8, pp. 61-64, 1993.
[5] R. Tsaih, *An Explanation of Reasoning Neural Networks*, Mathematical and Computer Modelling, Vol. 28, No. 2, pp. 37-44, 1998.
[6] R. Tsaih, *The Evolution of Internal Representation*, Mathematical and Computer Modeling, Vol. 38, pp. 339-350, 2003.
[7] R. Tsaih & C. Lin, *The layered Feed-forward Neural Networks and its Rule Extraction*, Advances in Neural Networks – ISNN 2004, Part I, pp. 377-382, 2004.

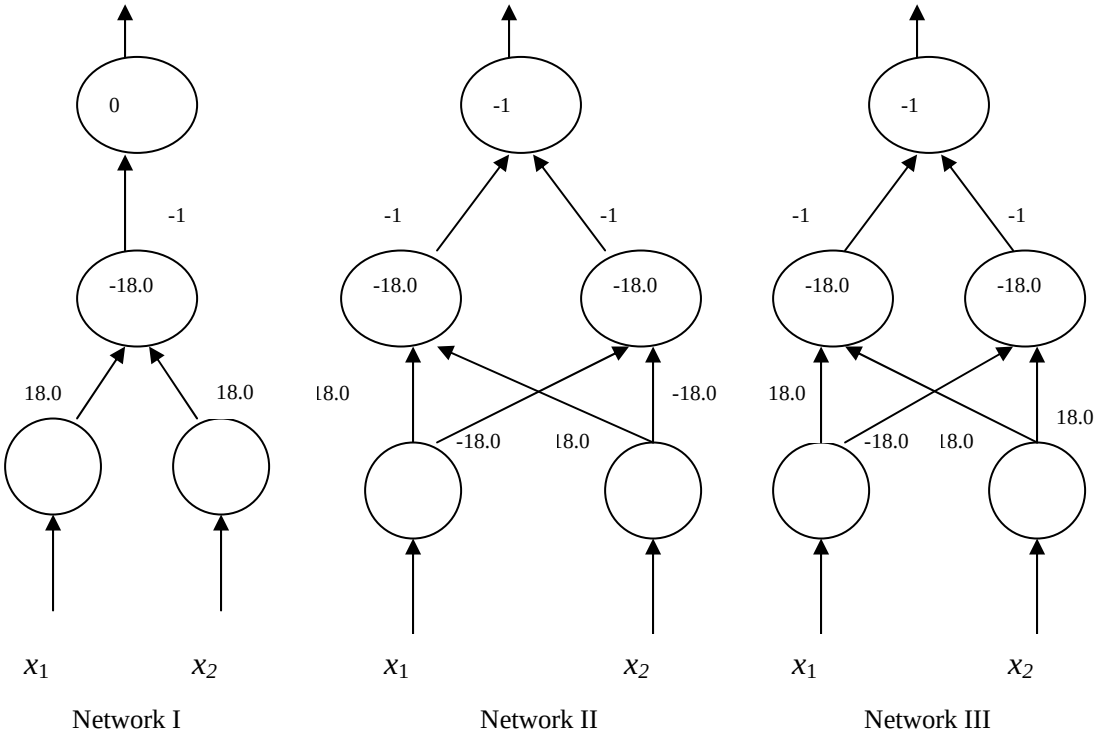

Figure 1. Three layered feed-forward neural networks. The number in the circle denotes the bias value, and the number along each line is the weight value.