An Integrative Neural Network with Feedback Control for Classification

Chang Liu*, Jing Hua†, Lixin Wang‡, Ruwei Dai‡

*Wayne State University, Detroit, MI 48202
†Princeton University, Princeton, New Jersey 08544
‡Chinese Academy of Science, Beijing, China, 100080

Abstract—This paper presents a novel integrative neural network, which contains a feedback loop for adaptive control of learning. Instead of designing a single classifier for the classification task, a finite number of classifiers are simultaneously applied and all outputs from the individual classifiers are processed by the integrative neural network. The stability conditions and supervised learning algorithms are derived for the artificial neural network. We have applied it to unstrained handwritten numbers recognition. Experiments are carried out and the results are compared to that of multi-layer perceptron. They show that the proposed integrative neural network with feedback control has a better classification rate with no decrease of reliability. This type of neural network scheme provides an alternative approach for ensemble learning.

I. INTRODUCTION

An Artificial Neural Network (ANN) is an information processing paradigm. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs learn by example, involving adjustments to the synaptic connections that exist between the neurons. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an “expert” in the category of information it has been given to analysis. This expert can then be used to provide projections given new situations of interest and answer “what if” questions.

Typical neural networks include feed-forward networks, which allow signals to travel one way only; from input to output, and feedback networks, which can have signals travelling in both directions by introducing loops in the network. For adaptive control and learning, feedback control from output neurons to input neurons may help to provide better learning and classification capability. It has been proved that feedback control is a typical characteristic in human neural circuits [1].

In this paper, we present an Integrative Neural Network with Feedback Control (INNFC). The advantage of this neural network is to introduce the output decision layer to input layer in order to help the network convergence and increase the learning capability. The integrative neural network can be used to construct an integrated classification system by integrating a number of individual classifiers. This approach provides a new scheme for ensemble learning, which combines a set of classifiers and then classify new data points by taking a weighted or unweighted vote of their predictions. The original ensemble method is Bayesian averaging, but more recent algorithms include error-correcting output coding [2], [3], Bagging [4], [5], and boosting [6].

The rest of the paper is organized as follows. We will analyze the choice for the integrative classifier in Section II and propose the integrative neural network with feedback in Section III. The mathematical proof and the algorithm are given in Section V and VI, respectively, while Section IV presents some mathematical preliminaries. The experimental results are shown in Section VII.

II. INTEGRATIVE CLASSIFICATION

The classifiers, especially the integrative classifier, in the IPRS can be achieved by different approaches, such as the voting method [7] [8] [9] and Multi-Layer Perceptron (MLP) [10] [11] [12]. The latter merits in the provision of supervised training to the integrative classifier. Suppose that $m$ classifiers $C_i (i = 1, 2, \ldots, m)$ are designed to classify an input pattern into one of $n$ categories. The output of classifier $i$ is denoted by $O_{ij} (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$. Then an artificial neural network is desired as an integrative network for further processing of all the classifiers’ outputs, i.e., the outputs of $m$ classifiers $O_{ij}$ are the inputs of integrative network. The task of integration is to process $O_{ij}$ and then make decision which category the original input belongs to. The weights of integrative network can be trained by a supervised learning algorithm with a large amount of samples. In the learning process, which category the sample belongs to is decided by the teacher. Therefore, the teacher plays an important role for learning in this architecture.

Generally speaking, the integrative classifier in Fig.1 can be realized by MLPs. As we know, there are two major kinds of neural networks.

1) Rumelhart’s MLP and the BP algorithm.

BP algorithm led to the resurgence of artificial neural networks and connectionism. It differs from the former linear perception in that the network is a nonlinear mapping. In the algorithm, the weight is changed according to the backward error from the next layer.

2) Competitive MLP and the competitive supervised learning algorithm, which are proposed to alleviate the defects of the BP algorithm.
In classification, one neuron’s ith output layer corresponds to one category and different categories are represented by different neurons. Although this approach simplifies the problem, it suffers some shortcomings. In practice, the number of training and testing samples are so large that even samples in the same category are distinct. They distribute in different regions in the sample space, sometimes even unconnected. The one-category-one-output assumption makes the learning difficult and the learned regions intersected which results in the poor generalization capability of the MLPs. Therefore, here we propose the competitive MLP. It is a major characteristic of multiple output units corresponding to one category. The algorithm is based on competitive supervised learning [13] [14].

All the aforementioned artificial neural networks actually perform a nonlinear mapping. However, cybernetics theory [15] shows that systems with feedback control behave better than those with open loop. In neural network-based classification, the difficulty for making a three-layer network to a close loop net is mainly the dimensional problem. For pattern classification by three-layer network, the dimension of output is “Category”, but both dimensions of input layer and the hidden unit layer are “feature”. It cannot connect two nodes together with distinct properties. Although there are neural networks with feedback such as the Jordan neural network [16] and Elman neural network [17], they cannot be implemented as single classifiers for pattern discrimination. For an integrative network, its input and output represent samples’ Category, therefore it can connect the same outputs from the output layer to the hidden unit layer as a feedback control loop, and such feedback is meaningful. Through these connections the integrative network becomes a close loop net, which does not perform a nonlinear mapping any more, but is a nonlinear dynamical system.

**III. AN INTEGRATIVE NEURAL NETWORK WITH FEEDBACK CONTROL (INNFC)**

As we have shown, we cannot add feedback connections in single MLP’s classifier but in integrative classifier. For this reason, we propose an Integrative Neural Network with Feedback Control (INNFC) for the integrative classifier. For this reason, we propose an Integrative Neural Network with Feedback Control (INNFC) for the integrative classifier (Fig.2).

In Fig.2, $W_{hi}$, $W_{oh}$ and $W_{ho}$ are the connection matrixes, which are from the input layer back to the input layer, respectively. We add some virtual neurons in the input layer, which are corresponding to those in the output layer. The connection matrix among the virtual ones and those in the hidden layer is $W_{ho}$. In each recurrence, the values of the virtual units are those of the output layer’s units at the previous step. The INNFC bears some similarity with the Jordan Neural Network. In INNFC, the input and the output in the INNFC both stand for the class attribute of a sample. They are identical in dimension and superposable.

The parameters of the INNFC are defined as follows:
- $W_{hi}$, $W_{oh}$ and $W_{ho}$ are $H \times I, O \times H$ and $H \times O$, respectively.
- $I$: number of the neural units in the input layer;
- $O$: number of the neural units in the output layer;
- $H$: number of the neural units in the hidden layer;
- $k$: recurrent time of the neural network;
- $\Omega$: number of the training samples;
- $i_{H \times 1}$: the input vector in the input layer;
- $y_{O \times 1}$: the output vector in the output layer;
- $g = (g_1, \cdots, g_H)^{t}$: the activation vector function in the hidden layer, satisfying $g((x_1, \cdots, x_H)^{t}) = (g_1(x_1), \cdots, g_H(x_H))^t$;
- $f = (f_1, \cdots, f_O)^{t}$: the activation vector function in the output layer, satisfying $f((x_1, \cdots, x_O)^{t}) = (f_1(x_1), \cdots, f_O(x_O))^t$.

The coordinate functions of activation functions are sigmoid ones, such as $\frac{1}{1+e^{-x}}$, which have the following properties: 1) monotonously increasing; 2) the function and its derivative are continuous and bounded.

The training set is $(a, y(a))(a = 1, \cdots, \Omega)$.
The recurrent equation of the INNFC is:
\[
\begin{align*}
y(k+1) &= f(W_{oh} \cdot g(W_{ho} \cdot y(k) + A \cdot i))(k \geq 0) \\
y(0) &= \alpha
\end{align*}
\]
where \(\alpha\) is an arbitrary initial value of the output layer. Generally we set it to be the zero vector.

IV. MATHEMATICAL PRELIMINARIES

Before we proceed to the discussion on the dynamic properties of INNFC, some mathematical preliminaries are presented for their use in the upcoming sections, which are concentrated in the linear space \(\mathbb{R}^n\).

A. Vector norm and induced matrix norm

Definition 1: A real-valued function \(\| \cdot \|\) on the linear space \(\mathbb{R}^n\) is called the vector norm function, such that

- A1) \(\|\alpha\| \geq 0, \forall \alpha \in \mathbb{R}^n; \|\alpha\| = 0\) if and only if \(\alpha = 0\);
- A2) \(\|a\alpha\| = \|a\|\|\alpha\|, \forall \alpha \in \mathbb{R}^n, \forall a;\)
- A3) \(\|\alpha + \beta\| \leq \|\alpha\| + \|\beta\|, \forall \alpha, \beta \in \mathbb{R}^n\) (triangle inequality).

Besides the vector norms, most discussions are with the induced matrix norms, whose definition is as follows:

Definition 2: Let \(\|\cdot\|\) be a given vector norm on \(\mathbb{R}^n\). Then for each \(n \times n\) matrix \(P\), the quantity \(\|P\|\) defined by \(\|P\| = \max_{\alpha} \|P\alpha\|\) is called the induced matrix norm of \(P\) corresponding to the vector norm \(\|\cdot\|\).

Based on the definition, the following form can be readily obtained. \(\|P\alpha\| \leq \|P\|\|\alpha\|\).

In general, there are three kinds of vector norms commonly in use. They are

1) \(\|\alpha\|_{\infty} = \max_i |a_i|;\)
2) \(\|\alpha\|_1 = \sum_i |a_i|;\) and
3) \(\|\alpha\|_2 = (\alpha, \beta) = \sqrt{\sum_{i=1}^{n} a_i^2};\)

The corresponding induced matrix norms are \(\|\cdot\|_{\infty}, \|\cdot\|_1, \|\cdot\|_2\) as follows, respectively:

1) \(\|P\|_{\infty} = \max_i \sum_j |p_{ij}|\) (row sum);
2) \(\|P\|_1 = \max_j \sum_i |p_{ij}|\) (column sum);
3) \(\|P\|_2 = \frac{\sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}^2}}{\|\lambda_{\text{max}}(P^TP)\|},\)

where \(\lambda_{\text{max}}\) is the maximum eigenvalue of the symmetric matrix \(P^TP\).

Note that the induced matrix norms are denoted on square matrixes in most literatures. However, we can extend them into the general cases, that is, the induced matrix norms are defined on \(m \times n\) matrix \(P\). Corresponding to the three vector norms \(V_1\)-(V3), it can be easily shown that (M1)-(M3) still hold. These extended ones are helpful for the discussion on the continuous distribution of the fixed points in Section V.

B. Contraction mapping theorem

Contraction mapping theorem is an important Theorem in Functional Analysis, which is sometimes also called the Banach (complete normed linear space) fixed point theorem. It is usually given in two forms: the global version and the local version. Because we analyze the absolute stability of INNFC, the global version is favorable. Therefore, we refer to the global version in this paper. Readers who are interested can find the complete description of this theorem in[18].

Theorem 1: Let \(T: \mathbb{R}^n \rightarrow \mathbb{R}^n\) be a mapping for which there exists a fixed positive constant \(\rho < 1\) such that
\[
\|Tz - Ty\| \leq \rho \|z - y\|, \forall z, y \in \mathbb{R}^n
\]
(2)
(for brevity, we write \(Tz\) instead of \(T(z)\)). Then

i) there exists exactly one fixed point \(z^f \in \mathbb{R}^n\) such that \(Tz^f = z^f\);

ii) for any \(z \in \mathbb{R}^n\), the sequence \(\{z_m\}_{m=1}^{\infty}\) in \(\mathbb{R}^n\) defined by \(z_{m+1} = Tz_m; z_0 = z\) converges to \(z^f\).

Moreover,
\[
\|z_m - z^f\| \leq \frac{\rho^m}{1 - \rho} \|z_1 - z_0\| = \frac{\rho^m}{1 - \rho} \|z_0\|. \quad (3)
\]
The mathematical preliminaries in this section are important for the following sections. Now we can go on with the discussion on the dynamic properties of the INNFC.

V. ANALYSIS OF STABILITY OF THE INNFC

This section is devoted to the analysis on the conditions of the stability of the INNFC. The model proposed in section III is
\[
\begin{align*}
y(k+1) &= f(W_{oh} \cdot g(W_{ho} \cdot y(k) + A \cdot i))(k \geq 0) \\
y(0) &= \alpha
\end{align*}
\]
where \(\alpha\) is an arbitrary initial value of the output. We aim to understand the eventual or asymptotic behavior of the INNFC as a dynamic system. For a discrete-time neural system, the goal is to understand the eventual state of the output vector \(y(k)\) as \(k\) becomes large.

Due to the classification standard proposed by K. Mat-suoka [19], the INNFC belongs to the class that the sample pattern is given as a constant input to the neural network. Generally speaking, even if the recurrent equation (4) have asymptotic stable fixed points, they depend on the initial value of \(y(0)\) as well as the structure of the neural network (i.e. the activation functions, connection matrixes) and the
Define two operations
\[
\begin{align*}
R_i &= \sum_{j=1}^{n} |c_{ij}| \\
C_i &= \sum_{j=1}^{n} |d_{ij}|
\end{align*}
\]
Then the eigenvalues of \( C \) are all in the union of \( n \) closed discs in the complex plane with centers \( c_{ii} \) and radii \( R_i(C_i), i = 1, \cdots, n \).
Define two operations
\[
\begin{align*}
R_{\text{max}}(C) &= \max_{i,j} \left( \sum_{i=1}^{m} |c_{ij}(x)| \right), \\
C_{\text{max}}(C) &= \max_{j,i} \left( \sum_{j=1}^{n} |d_{ij}(x)| \right),
\end{align*}
\]
where each entry of the matrix is bounded, which is assumed hereafter for function matrices. The operations are to find the maximum of the abstract sums of the row and the column of the function matrix \( C \), respectively. When the elements of the function matrix are all constant, they can be written as \( R_{\text{max}} \) and \( C_{\text{max}} \).

Lemma 2: Assume \( H = CD \), where \( H, C \) and \( D \) are \( n \times l \), \( n \times m \) and \( m \times l \) function matrices, respectively. Let \( H = (h_{ij}(x)), C = (c_{it}(x)), D = (d_{it}(x)) \). For given \( x_{ij}s \), let \( H_{\text{const}} = (h_{ij}(x_{ij})) \) we have
\[
\begin{align*}
R_{\text{max}}(H_{\text{const}}) &= R_{\text{max}}((h_{ij}(x_{ij}))) \\
&\leq R_{\text{max}}(H) \leq R_{\text{max}}(C)R_{\text{max}}(D),
\end{align*}
\]
and
\[
\begin{align*}
C_{\text{max}}(H_{\text{const}}) &= C_{\text{max}}((h_{ij}(x_{ij}))) \\
&\leq C_{\text{max}}(H) \leq C_{\text{max}}(C)C_{\text{max}}(D).
\end{align*}
\]
Proof: For given \( x_{ij}s \),
\[
\sum_{j=1}^{l} |h_{ij}(x_{ij})| \leq \sum_{j=1}^{l} \max_{x_{ij}} |h_{ij}(x)| = \max_{x_{ij}} \sum_{j=1}^{l} |h_{ij}(x)|
\]
and
\[
\sum_{i=1}^{m} |h_{ij}(x_{ij})| \leq \sum_{i=1}^{m} \max_{x_{ij}} |h_{ij}(x)| = \max_{x_{ij}} \sum_{i=1}^{m} |h_{ij}(x)|
\]
Therefore
\[
R_{\text{max}}(H_{\text{const}}) = R_{\text{max}}((h_{ij}(x_{ij}))) \leq R_{\text{max}}(H)
\]
As
\[
h_{ij}(x) = \sum_{t=1}^{m} c_{it}(x)d_{ij}(x),
\]
we obtain
\[
\sum_{j=1}^{l} |h_{ij}(x)| = \sum_{j=1}^{l} \left| \sum_{t=1}^{m} c_{it}(x)d_{ij}(x) \right| \\
\leq \sum_{j=1}^{l} \sum_{t=1}^{m} |c_{it}(x)||d_{ij}(x)|
\]
Similarly,
\[
\sum_{i=1}^{m} |h_{ij}(x)| \leq C_{\text{max}}(C)C_{\text{max}}(D).
\]
Therefore
\[
R_{\text{max}}(H) \leq R_{\text{max}}(C)R_{\text{max}}(D),
\]
and
\[
C_{\text{max}}(H) \leq C_{\text{max}}(C)C_{\text{max}}(D).
\]
Denote \( d_1 = \max_{i=1, \cdots, N} \{ f_i \} < \infty, d_2 = \max_{j=1, \cdots, L} \{ g_j \} < \infty \).
Then we can derive an important theorem on the absolute stability of the INNFC.

Theorem 2: The neural network (8) is absolutely stable if one of the following inequalities,
1) for the vector norm \( \| \cdot \|_{\infty} \),
\[
d_1d_2R_{\text{max}}(W_{ho})R_{\text{max}}(W_{oh}) < 1;
\]
2) for the vector norm \( \| \cdot \|_1 \),
\[
d_1d_2C_{\text{max}}(W_{ho})C_{\text{max}}(W_{oh}) < 1;
\]
3) for the vector norm \( \| \cdot \|_2 \),
\[
d_1^2d_2^2R_{\text{max}}(W_{ho})C_{\text{max}}(W_{oh})R_{\text{max}}(W_{ho})C_{\text{max}}(W_{ho}) < 1
\]
holds.
Proof: Let \( Ty = f(W_{ho} \cdot g(W_{ho} \cdot y + W_{hi} \cdot i)) \). Then \( T \) is a mapping from \( \mathbb{R}^N \) to \( \mathbb{R}^N \).
For a given input sample pattern \( i, \forall x, y \in \mathbb{R}^N \),
\[
Tx - Ty = \left( \frac{\partial T}{\partial y} \right)_{const} \cdot (x - y)
= (\text{diag}(f_1', \cdots, f_N') \cdot W_{oh} \cdot \text{diag}(g_1', \cdots, g_L') \cdot W_{ho})_{const} \cdot (x - y)
\]
(21)
where \( \frac{\partial T}{\partial y} \) is the vector norm.

\[
(0 \leq \theta_{ij} \leq 1) \text{ based on the mean value theorem of differentiation.}
\]

1) When \( \| \cdot \|_{\infty} \) is considered,
\[
\| T\mathbf{x} - T\mathbf{y} \|_{\infty} \leq R_{\max}(\| \frac{\partial T}{\partial y} \|_{\text{const}}) \| \mathbf{x} - \mathbf{y} \|_{\infty}. \quad (22)
\]

From lemma 2,
\[
R_{\max}(\| \frac{\partial T}{\partial y} \|_{\text{const}}) \leq R_{\max}(\frac{\partial T}{\partial y}) \leq R_{\max}(\text{diag}(f'_1, \ldots, f'_N)).
\]

\[
W_{oh} \cdot \text{diag}(g'_1, \ldots, g'_{hl}) \cdot W_{ho} \leq R_{\max}(\text{diag}(f'_1, \ldots, f'_N)).
\]

\[
R_{\max}(W_{oh}) \cdot R_{\max}(\text{diag}(g'_1, \ldots, g'_{hl})) \cdot R_{\max}(W_{ho}) \leq d_1 d_2 R_{\max}(W_{oh}) R_{\max}(W_{ho})
\]

Therefore
\[
\| T\mathbf{x} - T\mathbf{y} \|_{\infty} \leq d_1 d_2 R_{\max}(W_{oh}) R_{\max}(W_{ho}) \| \mathbf{x} - \mathbf{y} \|_{\infty} \quad (24)
\]

2) When \( \| \cdot \|_1 \) is chosen, it is similarly proved that
\[
\| T\mathbf{x} - T\mathbf{y} \|_1 \leq d_1 d_2 C_{\max}(W_{oh}) C_{\max}(W_{ho}) \| \mathbf{x} - \mathbf{y} \|_2 \quad (25)
\]

3) When \( \| \cdot \|_2 \) is in use,
\[
\| T\mathbf{x} - T\mathbf{y} \|_2 \leq \sqrt{\lambda_{\max}(\| \frac{\partial T}{\partial y} \|_{\text{const}}^T \| \frac{\partial T}{\partial y} \|_{\text{const}})} \| \mathbf{x} - \mathbf{y} \|_2 \quad (26)
\]

From lemma 1,
\[
\lambda_{\max} \leq R_{\max}(\| \frac{\partial T}{\partial y} \|_{\text{const}}^T \| \frac{\partial T}{\partial y} \|_{\text{const}}) \quad (27)
\]

Due to lemma 2,
\[
R_{\max}(\| \frac{\partial T}{\partial y} \|_{\text{const}}^T \| \frac{\partial T}{\partial y} \|_{\text{const}}) \leq R_{\max}(\| \frac{\partial T}{\partial y} \|_{\text{const}}) \cdot R_{\max}(\| \frac{\partial T}{\partial y} \|_{\text{const}}) \quad (28)
\]

Therefore,
\[
\| T\mathbf{x} - T\mathbf{y} \|_2 \leq \sqrt{R_{\max}(W_{oh}) C_{\max}(W_{oh}) R_{\max}(W_{ho}) C_{\max}(W_{ho})} \| \mathbf{x} - \mathbf{y} \|_2 \quad (29)
\]

According to theorem 1, if one of the three inequalities holds, the mapping \( T \) is a global contraction with the corresponding vector norm.

For this reason, for a given input sample pattern, there exists a unique asymptotically stable fixed point, and the output converges to it from any arbitrary initial state exponentially. That is, INNFC is absolutely stable.

Theorem 2 gives sufficient conditions for the absolute stability of INNFC. It appears that the absolute stability depends on the vector norm we choose. However, it is not the case. From the equivalence between norms on finite space, it can be easily shown that it is absolutely stable with other norms whenever it is absolutely stable with certain norm. That is, this stable property is independent with the specified norm.

On the other hand, if one of the three inequalities is satisfied, the final output is dependent on the inner structure of the neural network and the input sample pattern. When the connection matrixes and the activation functions are determined, the final output can be viewed as the function of input. It is natural to wonder the relation between them. The following theorem 3 is the answer.

For the discussion, we write the mapping \( T(y, i) \) instead of \( T(y) \)

**Theorem 3:** The fixed points of the neural network(4) are continuously dependent on the input sample patterns if one of the conditions in theorem 2 is satisfied.

**Proof:** For simplicity, we just prove the case with norm \( \| \cdot \|_{\infty} \). The other two cases are similar.

Denote \( \rho_1 = d_1 d_2 R_{\max}(W) R_{\max}(B) \), \( \rho_2 = d_1 d_2 R_{\max}(W) R_{\max}(A) \).

According to theorem 2, the neural network is absolutely stable, i.e. for every input pattern, there is exactly one global asymptotically stable fixed point if \( \rho_1 < 1 \).

Let \( i_1, i_2 \) be two arbitrary inputs, \( x^{i_1}_f, x^{i_2}_f \) be the corresponding stable fixed points. From \( x^{i_1}_f = T(x^{i_1}_1, i_1), x^{i_2}_f = T(x^{i_2}_1, i_2) \),
\[
\| x^{i_1}_f - x^{i_2}_f \|_{\infty} = \| T(x^{i_1}_1, i_1) - T(x^{i_2}_1, i_2) \|_{\infty} \leq \| T(x^{i_1}_1, i_1) - T(x^{i_2}_1, i_1) \|_{\infty} + \| T(x^{i_1}_1, i_1) - T(x^{i_2}_1, i_2) \|_{\infty}
\]

Similar to the proof of theorem 2,
\[
\| T(x^{i_1}_1, i_1) - T(x^{i_2}_1, i_1) \|_{\infty} \leq \rho_1 \| x^{i_1}_1 - x^{i_2}_1 \|_{\infty}
\]

and due to the extended induced matrix norm,
\[
\| T(x^{i_1}_1, i_1) - T(x^{i_2}_1, i_1) \|_{\infty} \leq \rho_2 \| i_1 - i_2 \|_{\infty}
\]

Therefore, we have
\[
\| x^{i_1}_f - x^{i_2}_f \|_{\infty} \leq \rho_1 \| x^{i_1}_1 - x^{i_2}_1 \|_{\infty} + \rho_2 \| i_1 - i_2 \|_{\infty}
\]

From \( \rho_1 = d_1 d_2 R_{\max}(W) R_{\max}(B) \leq 1 \)
\[
\| x^{i_1}_1 - x^{i_2}_1 \|_{\infty} \leq \frac{\rho_2}{1 - \rho_1} \| i_1 - i_2 \|_{\infty}
\]

Therefore it can be readily verified that the absolutely stable fixed points are continuously dependent on the input patterns.

Theorem 2 and 3 are the bases of the application of INNFC. Next section will present a learning algorithm under one of the constraints(18)(19)(20).
VI. THE SUPERVISED LEARNING ALGORITHM OF INNFC

In previous section, we prove the sufficient condition on the absolute stability of INNFC. It is necessary to find a way to ensure the INNFC satisfying one of the three-inequality(18)-(20). That is to say, how we can implement it into the learning algorithm. Although there are methods in constrained optimization theory, such as the penalty function method, we found it was not always efficient. For this reason, we directly add the "regularization" in the algorithm so that the INNFC readily satisfies one of (18)-(20).

On the other hand, BP algorithm is common for the MLPs. We want to keep its merits in our algorithm. 

Considering equation (1), the partial differentiation of $y_i(k+1)$ consists in two parts: that under the assumption that $y_i(k)$ is constant and $\frac{\partial y_i(k+1)}{\partial y_i(k)}$. the former can be derived similarly to that in BP algorithm.

Suppose that weight $w_{ps}$ connects the output of neuron $s$ to the input of neuron $p$. The sum of neuron $p$’s inputs is $net_p$ and the output is $o_p$.

$$net_p = \sum_s w_{ps} o_s,$$  \hspace{1cm} (33)

and

$$o_p = f(net_p),$$  \hspace{1cm} (34)

where $f$ is the activation function.

Now we compute $\frac{\partial y_i(k+1)}{\partial w_{ps}}$. Let $\frac{\partial y_i(k+1)}{\partial w_{ps}}|_{y_i(k)}$ be the partial differentiation considering $y_i(k)$ as constant. We have

$$\frac{\partial y_i(k+1)}{\partial w_{ps}} = \frac{\partial y_i(k+1)}{\partial y_i(k)}|_{y_i(k)} + \sum_{m=1}^{O} \frac{\partial y_i(k+1)}{\partial y_m(k)} \cdot \frac{\partial y_m(k)}{\partial w_{ps}}.$$ \hspace{1cm} (35)

Denote

$$\delta_{ip} = \frac{\partial y_i(k+1)}{\partial y_i(k)}|_{y_i(k)}$$ \hspace{1cm} (36)

Because

$$\frac{\partial y_i(k+1)}{\partial w_{ps}}|_{y_i(k)} = \frac{\partial y_i(k+1)}{\partial net_p}|_{y_i(k)} \cdot \frac{\partial net_p}{\partial w_{ps}}|_{y_i(k)}$$ \hspace{1cm} (37)

we obtain

$$\frac{\partial y_i(k+1)}{\partial w_{ps}}|_{y_i(k)} = \delta_{ip} \cdot o_s,$$ \hspace{1cm} (38)

(1) when neuron $p$ is in the output layer, if it is the $i$-th output unit,

$$\delta_{ip} = f'(net_p),$$ \hspace{1cm} (40)

otherwise,

$$\delta_{ip} = 0,$$ \hspace{1cm} (41)

(2) when $p$ is in the hidden layer,

$$\delta_{ip} = \frac{\partial y_i(k+1)}{\partial o_p}|_{y_i(k)} \cdot \frac{\partial o_p}{\partial net_p}|_{y_i(k)}$$

$$= g'(net_p) \sum_m \delta_{im} w_{mp}.$$ \hspace{1cm} (42)

According to (39),(40),(41),(42), we can derive $\frac{\partial y_i(k+1)}{\partial w_{ps}}|_{y_i(k)}$. It is obvious that it is a back propagation process. Now we compute $\frac{\partial y_i(k+1)}{\partial w_{ps}}$. For simplicity, denote $w_{in}^h$ as the weight from the $n$-th neuron in the hidden layer to the $i$-th one in the output layer, $w_{nm}^o$ as the weight from the $m$-th one in the output back to the $n$-th one in the hidden layer. From equation (1),

$$\frac{\partial y_i(k+1)}{\partial y_n(m)} = \frac{\partial}{\partial y_n(m)} \left( f_i \left( \sum_n w_{in}^hw_{nm}^o \right) \right)$$ \hspace{1cm} (43)

Due to (39)-(43), we can compute $\frac{\partial y_i(k+1)}{\partial w_{ps}}$, which is comprised of the back propagation (39)-(42), and the recurrence(43). The supervised learning algorithm is as follows. For simplicity, it is under the inequality (18). The other two are similar.

1. Initialization of $W_{hi}, W_{oh}$ and $W_{in}$, let $\Delta w_{ps}(0) = 0$.
2. Let the training time $u = 1$.
3. Let the sample number $\alpha = 1$, error $E = 0$.
4. Let the recurrent time $k = 0$, and $\frac{\partial y_i(0)}{\partial w_{ps}} = 0$.
5. Compute $y(k+1) + 1$ and $E_{stable} = \sum_{i=1}^{O} |y_i(k+1) - y_i(k)|$ according to (1).
6. Due to (39),(40),(41),(42) and (33), compute $\frac{\partial y_i(k+1)}{\partial w_{ps}}(i = 1, \ldots, O)$.
7. Let $k = k + 1$. If $E_{stable}$ is smaller than a positive constant or $k$ is larger than the specified recurrent time, continue; Otherwise return to step 5.
8. Compute $\frac{\partial E(\alpha)}{\partial w_{ps}} = \sum_{i=1}^{O} (y_i(\alpha) - y_i(\alpha)) \frac{\partial y_i(k+1)}{\partial w_{ps}}$.
9. Change the weight according to

$$\Delta w_{ps}(u) = -\alpha \frac{\partial E(\alpha)}{\partial w_{ps}} + \beta \Delta w_{ps}(u-1),$$ where $\alpha$ and $\beta$ are positive constants.
10. Let $Max1 = Max(\max(W_{hi}), Max2 = Max(\max(W_{ho})$.
11. Compute $Abs1(i) = \sum_{n=1}^{O} |w_{in}^h(i)|$ and $Abs2(n) = \sum_{m=1}^{O} |w_{nm}^o|.$
12. Regularize weights $w_{in}^h$ and $w_{nm}^o$.

$$w_{in}^h = \frac{\lambda_1 w_{in}^h}{\sqrt{Abs1(i) \cdot Max2}},$$

$$w_{nm}^o = \frac{\lambda_2 w_{nm}^o}{\sqrt{Abs2(n) \cdot Max1}}.$$

$\lambda_1$ and $\lambda_2$ are positive and satisfy $\lambda_1 \lambda_2 < 1/d_1 d_2$.
13. Compute the error $E(\alpha) = \sum_{i=1}^{N} (y_i(\alpha) - x_i(\alpha))^2 / 2$.
14. When $\alpha$ is less than the sample amount $\Omega$, let $\alpha = \alpha + 1$ and return to step 4. Otherwise continue.
15. Let $E = E/\Omega$. If $E$ is less than $\varepsilon$, or $u$ is greater than the specified training time, finish the training. Otherwise return to step 3 and begin a new training.
TABLE I
THE EXPERIMENTAL RESULTS OF C1, C2 AND C3

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Error</th>
<th>Recog. with rejection</th>
<th>Rejection</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>&lt; 0.08</td>
<td>86.71%</td>
<td>9.60%</td>
<td>95.9%</td>
</tr>
<tr>
<td>C2</td>
<td>&lt; 0.09</td>
<td>87.72%</td>
<td>9.95%</td>
<td>97.4%</td>
</tr>
<tr>
<td>C3</td>
<td>&lt; 0.05</td>
<td>91.10%</td>
<td>7.10%</td>
<td>98.1%</td>
</tr>
</tbody>
</table>

Step 12 is the most important, which ensures the constrained weights satisfy inequality (18). When the training finishes, the INNFC (1) is absolutely stable. It should be noticed that the algorithm is for the INNFC with one hidden layer. If it has more hidden layers, simple modification can get the corresponding algorithm.

VII. Experiments

To demonstrate the superiority of the proposed INNFC, experiments were performed with the totally unconstrained handwritten numeral database. The integration was based on three single classifiers, C1, C2 and C3. C1 used contour and point features and trained by 8000 samples. C2 used lattice feature and trained by 10000 samples. C3 used direction feature and trained by 6000 samples. They were all tested by another 10000 samples. The training algorithm was the competitive supervised learning [13]. The results are shown in Table I.

Reliability was according to

\[
\text{Reliability} = \frac{\text{Recognition with rejection}}{1 - \text{Rejection}}
\]

The integration was based on C1, C2 and C3. To verify the INNFC’s effectiveness, it was compared with the MLP. Their input units, hidden units and output units were 30, 10 and 10, respectively. Two experiments were carried out. One was 2000 training samples and 8000 testing samples. The other was 3000 training samples and 7000 testing samples. The results were shown in Table II and III, respectively.

Table II and III showed that the integration of multiple classifiers improved the system’s generalization capability and reliability greatly. On the other hand, the recognition with rejection of the INNFC increased 2.31% and 0.74%, whose reliability almost did not decrease. This proved the superiority of the INNFC.

VIII. Conclusions

The INNFC is a close loop system in the sense of cybernetics because the feedback connections make it a dynamic one. For its application in pattern recognition, it shows absolute stability. We prove the sufficient condition for its stability, then provide the supervised learning algorithm with the derived inequalities, which ensures the IPRS’s absolute stability.

To verify the proposed model’s superiority, experiments were carried out on the unconstrained handwritten numeral database. Three single classifiers and INNFC were trained by the supervised learning. The experimental results were convincing. However, we have not tested the algorithm on many different data sets, thus the overall performance has not yet been verified. In future work, we plan to apply it to other pattern recognition tasks, such as brain tumor recognition.

ACKNOWLEDGEMENT

This research was supported in part by the MTTC grant MTTC05-135/GR686, the MTTC grant MTTC05-154/GR705, and Wayne State University Faculty Research Award.

REFERENCES


