Sequential Multi-Layers Covering Neural Network

Wang Renwu[1,2], Yang hongshan[1], Chen Jiaxun[1]
[1]College of Information Sciences and Technology, DongHua University
[2]Business school of East China Normal University, Shanghai
P. R. China
[1]No.1882 West YanAn Rd., Shanghai
P. R. China

Abstract: - A new architecture of neural networks has been preliminarily proved in this paper for the Sequential Multi-Layers covering Neural Network (SMNN). In SMNN, the training samples are separated by hidden neurons until the original same class training sets are empty. In the paper, it is proved that the target sets could be linearly separated in polynomial time. Some algorithms have been proposed in light of the ideas. We have used our algorithms with good results, e.g. voice classic. We present only a couple of examples: tow-dimension spiral line, which is difficult for BP network, and real data set.

Key-Words: - Neural Network; Topologic Space; SMNN

1 Introduction

Some kinds of neural network have been proposed in the last few decades. Among them, Feed-Foreword layered is being used widely. However, it has some drawbacks inside:

(1)The architecture of neural network should be pre-determined before training. It is a very difficult task especially for real-world problem.

(2)The presentation of knowledge in the neural network can not give the hierarchy concept, which is not compatible with the real world. As we all know, over time human knowledge of the same objects transform remarkably.

(3)For the training process, when new training samples are added to the network, the network should wholly change to new weights. The time cost is tremendous.

Therefore, some solutions have been proposed to deal with these drawbacks [1]. Zhang Lin has given a covering algorithm approach [2]. He used geometric precepts to cover the training samples. S. J. Wang proposed BPR (Biomimetic Pattern Recognition) theorem [3]. The big difference is that coverage for feature space instead of dividing the training set in pattern recognition.

The goal of this paper is to introduce the basic ideas and experiences. In this paper, we focus on the preliminary proofs of the SMNN using topological theory [4] and its special implementations.

The remainder of this paper is organized as follows. Section 2 gives the description of the SMNN theorem. In Section 3, special implementations are given. Section 4 is about the experiment result and its analysis.

2 The preliminary theorem of SMNN

Please, leave two blank lines between successive sections as here.

For convenience of discussion, we give some symbols as below:

(1) R^n, is the n-dimension Euclid space. It is Hausdorff topologic space.

(2) $S_{i\in\{1..m\}}$, is the i category training samples, which is a discrete space.

(3) $\{C_i^m\} \subset R$, is the i category. M represents m categories.

(4) $o \equiv \{0,0,....\}^n \subset R^n$, is the sequence number of neural network's output.

(5) $p_{i,j}$, is the layer number of j neuron for i category type.

(6) ξ_i^x, is the x special sample in the S_i.

(7) y_i, i=1…m, is the output of neural network, which is the binary values {0, 1}.

Some primitive definitions and proofs will be given as follows:

Definition 1. SMNN is the neural network with layer number, which uses constructive method to cover category sets. The symbol of training set mapping to category is $\{\bigcup \xi_i^x\} \rightarrow \{C_i^m\}$, with the corresponsive neural mapping $\{\bigcup\{\psi(\xi_i^x)_{p_{ij}}\}\} \rightarrow \{C_i^m\}$.

When a testing sample is inputted into neural network, the output is:

$Y = \{y_i \mid \max\{p_j \mid p_{j=1..m} \wedge y_{j=1..m} == 1\}\}$,
$y_i = \{x \mid x = 0 \ or \ 1\}, p_{ij} \in [0,1].$

Theorem 1. For training set $S_{i\in\{1..m\}}$ in $S_{i\in\{1..m\}}$, is logical compact space.

Proof. From the defined symbol, we know $S_{i\in\{1..m\}} \subset R^n$ is a discrete space. Therefore, there exists $\xi_i^x, \xi_i^y \in S_{i\in\{1..m\}}$. In measure metric space P, there exists $p(\xi_i^x, \xi_i^y) < M$,
$\forall \xi_i^x \in S_{i\in\{1..m\}}, \exists M, M = N + p(0, \xi_i^x)$
$\rightarrow S_{i\in\{1..m\}} \subset [-N, N]^n, M \in R.$ So, $S_{i\in\{1..m\}}$ is closed set on $[-N, N]^n$.
In addition, $S_{i\in\{1..m\}}$ is a discrete space. Therefore, $S_{i\in\{1..m\}}$ is logical compact space.

Theorem 2. For category type $C_i^m \in R$ and training set S_i , there exist single mapping chains: $C^L : S_i \rightarrow \{C_i^m\}$.

Proof. For countable finite set S_i , there exists radix card (S_i) < χ_0 . Because S_i is greater than C_i^m , there exist single mapping chains C^L : $S_i \rightarrow \{C_i\}$.

Theorem 3. Using SMNN to constructive neural network, the covering type spaces procedure can be finished in finite time. That is the Covering space existence.

Proof. The constructive procedure is given here. For any point in S_i space, from the poincaré theory of УрЫсон[4], there exists function $f : S_i \rightarrow [0,1]$. For any sub-compact space of S_i, such as A0, A1, satisfies the condition: $f(A_1) = 1$, $f(A_0) = 0$. Meanwhile, there is a sequence PO := $< p_{i0}, C_i >$, ($\mathrm{PO_i} \prec \mathrm{PO_j} \rightarrow p_{i0} < p_{j0}$) . p_{i0} is the initial layer number. There always exists open covering space of Ai, symbolized as λ_x . There exists: $S_i \subset \bigcup \lambda_x$, $S_i \bigcap (\bigcup \lambda_x) \subset S_i$, so $f : f(\lambda_x) \rightarrow S_i$. The open covering spacing is finite space, because S_i is a finite discrete space. So, the covering procedure can be finished in polynomial time.

Theorem 4. For $S_{i\in\{1..m\}}, S_{j\in\{1..m\} i\neq j} \subset R^n$, Raidx card(S_i) < card(S_j), space access probability is $h_{i=1...m}$, if $card\ (S_i)/h_i < card\ (S_j)/h_j$, then the layer number should be $p_{i0} < p_{j0}$.

Proof. If $card(S_i) = \tau_i, \tau_1/h_1 < .. < \tau_n/h_n$, then the whole time cost is T,
$T = h_1\tau_1 + h_2(\tau_1 + \tau_2) + ... + h_n(\tau_1 + ... + \tau_n)$ If there exists a pair $\tau_i/h_i > \tau_{i+1}/h_{i+1}$, the new time cost is T_1. The margin value between T_1 and T is ΔT.

$$\begin{aligned}\Delta T &= T_1 - T \\ &= h_i(\tau_1 + \tau_2 ... + \tau_i + \tau_{i+1}) + h_{i+1}(\tau_1 + \tau_2 ... + \tau_{i-1} \\ &+ \tau_{i+1}) + -h_i(\tau_1 + \tau_2 ... + \tau_{i-1} + \tau_i) - h_{i+1}(\tau_1 \\ &+ \tau_2 ... + \tau_i + \tau_{i+1}) \\ &= h_i\tau_{i+1} - h_{i+1}\tau_i = \tau_i\tau_{i+1}(\frac{h_i}{\tau_i} - \frac{h_{i+1}}{\tau_{i+1}}) \geq 0\end{aligned} \quad (1)$$

So, the time cost increases with sequence changing. But it's only the time consuming analysis, more analysis should mention the knowledge value of the samples which should be done later.

3 Special Implementations

In this section, special implementations are provided to cover training sets in high-dimension space. The basic distance measure is the Euclid distance such as ||A||. The whole procedure is as follows. The basic idea:

(1)For training process, we use hyper-ball to cover same type samples, symbolized as:

$$\begin{cases} P(S_i, x_0) <= d, S_i \subset R^N, x_0 \in R^N, d \in R \\ P(S_i, x_0) = \| xi - x_0 \|^2, x_i \in A \end{cases} \quad (2)$$

(2) For recognition, it is symbolized as:

$$\left.\begin{aligned} &\| xi - x_0 \|^2 <= d, x_i \in R^N, d \in R \\ &\rho_i = \max(\rho_i)_0^m \end{aligned}\right\} \rightarrow C_i \quad (3)$$

3.1 Randomized Partial Algorithm (RPA)

Step 1. Initial layer order sequence should be built following theorem 4. Selecting set S_i with its layer number is max. For any element $\xi_i^x \in S_i$, the set center is $\bar{\xi}_i = \sum_{x=1}^{card(S_i)} \xi_i^x / card(S_i)$. Then to calculate offset distance from $\bar{\xi}_i$ in different dimension, $D_{\min} = \{\min(D^1, D^2 ..... D^n)\}$.

Step 2. In high-dimension, neuron ψ should be made following the rules: Making $\bar{\xi}_i$ as center and $D_{\min}$ as distance. Let ψ divide samples of $S_{j \neq i}$, adjust D value to satisfy the formula below.

$$\psi = \begin{cases} \psi(a \in A) = 1 \wedge \psi(b \in \{S_i - A\}) = 0, A \subseteq S_i \\ \psi(a \in S_{i \neq j}) = 0 \end{cases}$$

Making neuron $\psi(A)$ as type of C_i.

Step3. The process jumps to Step 4 if set A is not empty after Step 2. Otherwise, the next operation is to continue to do the following the same way as we did in Step 2. $\xi_i^x = \{x \mid d_x = \max\{d(\xi_i^p, \bar{\xi}_i)\}, \xi_i^p \in S_i\}$. Now, let ξ_i^x be the center. Adjust D of the distance between $\bar{\xi}_i$ and samples of alien sets. The reason for selecting ξ_i^x is that current selected ξ_i^x gives $\bar{\xi}_i$ a bigger weight part than others (This will be proved in another paper).

Step 4. Removing the set A from S_i, $B = S_i - A$, $S_i = B$, we put $S_i\ (S_i \neq \phi)$ into training set sequence and put $p_{i(j+1)} = p_{ij}$^0.5 into layer sequence.

Step 5. Iterating step1-4 until there is only one type existing in the training set.

Step 6. For the last one training set, let's get the minimal distance between aliens set. Let the remaining set be a special class of its own with the layer number.

Following through Steps 1~6, we get sets, like neurons $\{\psi\}$, type identification $\{C_i\}$ and layer number $\{p_{ij}\}$. The high layer number of output indicates it is the special category.

3.2 Center Adaptive Selection Algorithm (CASA)

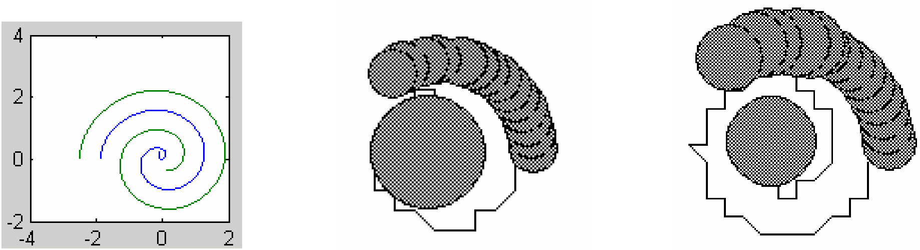
For each training sample, we calculate its center. Maybe the center pointer is not the real node in samples, therefore, we just adapt the center node to the nearest one in samples. The following algorithm steps are the same as from Step 2 to Step 6 in RPA.

4 Experiment result and its analysis

The tow-dimension spiral line is made to test the neural network. The spiral line is difficult to other classic neural networks, such as BP. The selected spiral curves are as follows.

$$\begin{cases} R_1 = 0.4 * \theta, \theta \in [-0.4\pi, 4\pi] \\ R_2 = 0.4 * (\theta + \pi), \theta \in [-0.4\pi, 4\pi] \end{cases} \quad (4)$$

For the training process, we carry the same number of samples from two lines. For the testing, we get the 1000 samples from each line and get the total number: 2000(2*1000) samples. The real data include character recognition and image classification. The character class has a to z letters with 20 features, 20000 sample in all, 16000 for training and 400 for testing. For image classification, we use 8 classes of images, including brace, vegetable, etc., and each class has 19 features. Use 210 images for training and 2100 samples for testing. The development environment is Visual C++ 6.0.



a b c

Figure 1. (a)Two spiral lines , R1,R2;(b)Partly covering space range of R1;(c) Partly covering space range of R1; The curve line is the training samples adjoined line. Current sample number is 100 each.

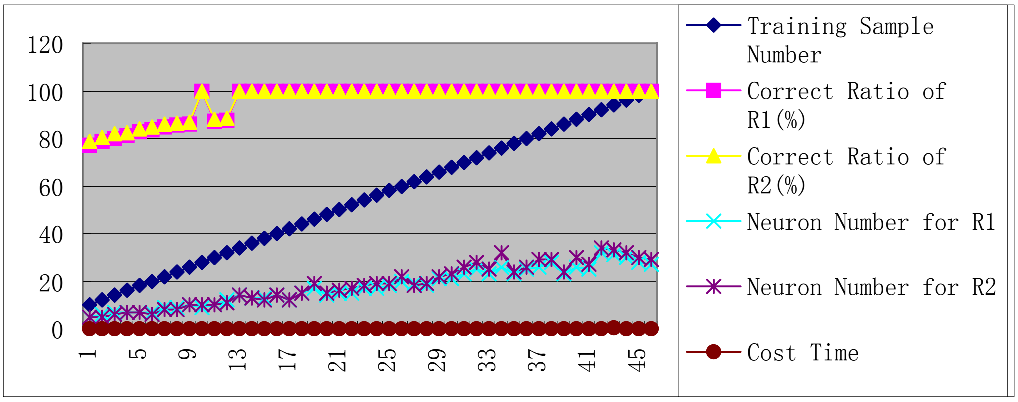


Figure 2. The result of same testing sets under different numbers of training set samples.

Table 1. The comparison of algorithms for different training set samples

		RPA	CASA
character recognition	correct rate(%)	73.743	77.069
	wrong rate (%)	25.106	21.730
	refuse rate (%)	1.150	1.200
	neural cells number	2271	2448
	time cost (second)	268.802	41.39
Image classify	correct rate(%)	70.143	78.095
	wrong rate (%)	29.619	21.048
	refuse rate (%)	0.238	0.857
	neural cells number	59	62
	time cost (second)	0.429	0.491

From Figure 2, we know that the SMNN can give a satisfactory experiment result, even though the problem is difficult to classic neural networks, such as BP. The time cost of SMNN is very short and the amount of neurons is dynamically changed according to the training sets. This character is very similar to human learning process, which is not determined before learning process but influenced by knowledge itself. From Table 1, we know CASA can make higher efficiency than RPA because CASA uses sample center to construct neural cells.

5 Conclusion

The preliminary principle and analysis of SMNN have been presented in this paper. This algorithm of the neural network is based on the model of human learning. It has no drawbacks of classic neural networks. It makes unnecessary to determine the structure and hidden layers of neural network. Moreover, the already trained neural network is not destroyed with new training samples coming in. The training process is very fast because it only adjusts its own neuron rather than the whole cells in networks. Currently, SMNN are being proved from theorems and tested real world problems. Of course, some features should be further researched and investigated, such as isotropy in local and anisotropy in global space.

References:

[1] B. Verma, Fast Training of Multilayer Perceptions, *IEEE T-NN* Nov. 1997, 1314-1320.

[2] Zhang L, Zhang B. A Geometrical Representation of McCulloch Pitts Neural Model and Its Applications. *IEEE Translation on Neural Networks*, 1999, 10 (4): 925～929

[3] S. J. Wang, Bionic(topological)pattern recognition-A new model of pattern recognition theory and its applications, *Acta Electron. Sinica* Vol.30, No.10, pp.1-4,2002.

[4] IM Singer and JA Thorpe, *Lecture Notes on Elementary Topology and Geometry* Springer-Verlag, New York, 1967.