

Agent Swarm Classification Network ASCN

Chi-kin Chow and Hung-tat Tsui

Visual Signal Processing and Communications Laboratory

Department of Electronic Engineering

The Chinese University of Hong Kong

{ckchow1, httsui}@ee.cuhk.edu.hk

Abstract – In this paper we introduced a newly RBF Classification Network – “Agent Swarm Classification Network ASCN”, which is trained by a Multi-agent systems (MAS) approach. MAS can be regarded as a swarm of independent software agents interact with each other to achieve common goals, complete concurrent distributed tasks under autonomous control. By treating each neuron as an agent, the weights of neurons can be determined through a set of pre-defined simple agent behavior. Three sets of experiments are examined to observe the effectiveness of the proposed method.

Keywords: Multi-Agent System, RBF neural network, Classifier

1 Introduction

Radial Basis Function (RBF) network consists of 3 layers: *Input Layer*, *Hidden Layer* and *Output layer*. Due to its simple architecture, RBF network has been applied successfully in a number of applications including image processing [6], speech recognition [1], adaptive equalization [3] and time series analysis [9]. Besides of the objective on regression, a wide range of pattern recognition applications, such as fault detection and diagnosis [2], image recognition [7] and speech recognition [5], are also involving RBF classifier. It is due to the facts that the RBF classifier is faster to be well-trained.

Different from the previous training algorithms that clusters the samples with same class and then minimizing the least mean output error [10], we proposed to model an RBF classifier as an RBF regression network with an extra hard-thresholding layer. Further, by using a Multi-Agent System approach, the parameters of the classifier can be determined.

We proposed a self-constructed RBF classifier called “Agent Swarm Classification Network” (ASCN) in this paper. It comprises of SVC [11, 4, 8] advantages that the centers and variances of the neurons are fixed. In addition, by using the MAS-based approach for training, the time for weight determination can be greatly reduced.

This paper is organized as follows. In section 2, we decompose a generalized MAS into 3 stages. Modeling of ASCN by a multi-agent system and hence the characteristic of agents are discussed in Section 3. The strategy for pattern noise removal is introduced in Section 4. 3 sets of

classification experiments will be examined so as to illustrate the performance of the proposed algorithm in Section 4. And a conclusion is drawn in Section 5.

2 General Architecture of MAS

Multi-agent system is a group of problem-solver entities, which consists of a predefined rule set. Moreover, the whole system composes of the following characteristics:

- Each agent has incomplete capabilities to solve a problem.
- There is no global system control.
- Data is decentralized.
- Computation is asynchronous.

In general, the behavior of an agent inside the MAS can be decomposed into 3 stages. Initially, the agent collects information from the environment and its neighbor(s), to conclude its local environment observation LEO. Base on the current LEO, the agent will draw a decision from its decision-making unit, which is benefit to its current status. Afterward, the agent will execute the decision by adjusting its internal parameters. Since the agent swarm keep interact with the environment, there is no beginning nor terminate stage. In addition, as all agents in the swarm undergo those 3 stages in parallel instead of sequentially, a more complicated swarm behavior is constructed from a set of simple rules.

3 RBF Network as MAS

In this section, we introduce an agent-based representation for evolving RBF networks and describe the associated multi-agent system. By considering each training sample as a neuron, a RBF network with optimal size can be achieved if:

1. The number of zero-magnitude neurons is maximized.
2. Positive value is returned for the positive class sample.
3. Negative value is returned for the negative class sample.

Instead of applying a passive approach that determines the set of zero-magnitude neurons, we request each neuron actively minimizing its magnitude as much as possible. Therefore, the neurons can be regarded as an agent swarm, while the criteria listed at the above can be treated as the agent behaviors.

3.1 Parameter of ASCN agent

An agent's internal abilities are defined by a set of parameters. As we regard each neuron inside the ASCN as an agent, 9 parameters are required to describe its features. 4 of them are adjustable during the iterations while the remaining 5 are predefined at the beginning of the swarm evolution. The follows are the details of the agent parameters.

- i. Magnitude M – Weight of the neuron, which is initialized as one.
- ii. Position vector $\vec{\mu}$ – Position of the training sample.
- iii. Variance σ – This parameter describes the correlation among the agent with its neighbor(s). To simplify the training process, all neurons are set to be the same value. Instead of predefined the value by the user, it is suggested by the following equation:

$$\sigma = p^{-\frac{1}{D}} (-8 \ln 0.8)^{-\frac{1}{2}}$$

where p is the swarm size and D is the swarm dimension

- iv. Local Environment Observation O – The classification value corresponding to its position vector.
- v. Neighbor Linkage \vec{N} – It is a p by I vector, i.e. $N_i = [n_{i,1} \ n_{i,2} \ \dots \ n_{i,p}]$. $n_{i,j}$ equals to 1 if Agent A_j is the neighbor of A_i . Otherwise, $n_{i,j}$ is set to be zero. Therefore, $N_i \bullet N_i^T$ represents the number of neighbors that A_i communicates with.
- vi. Magnitude Decay Rate α – This parameter controls the magnitude decrement rate of an agent. It is changed adaptively instead of a constant decay rate along the iterations. The adaptive strategy will be discussed at the Section 3.4.
- vii. Excited Magnitude Q – This parameter is used for preventing classification error. It is adjustable and initialized as 1.
- viii. Opponent Bias B – This parameter represents the minimum classification value such that no classification error occurred. The Opponent bias of agent A_i is defined as:

$$B_i = \sum_j Q_j G_{i,j} \text{ where } G_{i,j} = e^{-\frac{\|\mu_i - \mu_j\|}{2\sigma^2}} \text{ for all } C_i \neq C_j$$

- ix. Class C – The class of the corresponding training sample.

3.2 Neighbor of ASCN agent

Due to the concept of decentralize, no agent is able to collect all information from the whole swarm. Alternatively, they can only communicate with limited

amount of agents inside the swarm. We named the agents that can be communicated by agent A_i as the neighbors of A_i . It is obvious that as the size of neighbor increase, the more information can be collected and hence a more accurate decision can be drawn. On the other hand, more time is needed to analysis the collected information. Therefore, for a robust MAS, the agents should achieve the global objective by communicating with minimum number of neighbors. In the case of ASCN, the neighbors of an agent A_i are defined as follows:

1. Calculate the Contribution of agent A_j : $\{G_{i,j}\}$, i.e. for $G_{i,j} = \exp(-\|\mu_i - \mu_j\| / 2\sigma^2)$ all $j \neq i$ and $C_i = C_j$.
2. Sort $\{G_{i,j}\}$ in the descending order.
3. Assume $R(j)$ be the ranking of A_j .
4. By giving the decentralize error threshold β , agent A_k is the neighbor of A_i if:

$$\sum_{m=1}^k G_{i,m} \leq \beta \sum_{m=1}^k G_{i,m} \text{ for } R(k) \geq R(h)$$

It is found that more neighborhood linkages are established at case when β is smaller.

3.3 LEO of ASCN agent

The LEO of A_i is defined as:

$$O_i = M_i + \sum_{j=1}^{N_i} M_{i,j} e^{-\frac{\|\mu_i - \mu_{i,j}\|}{2\sigma^2}} \quad (1)$$

where $I_{i,j}$ is the agent index of j^{th} neighbor of i^{th} agent.

The LEO is formed as a linear equation since the positions and the variances of all agents are fixed, which speed up the process.

3.4 Decision Making Unit of ASCN agent

By recalling the objective of swarm, each agent is minimizing its magnitude under the constraint that no classification error occurred. Therefore, the behavior of an agent A_i can be summarized as follows:

1. Determine the classification error E_i , i.e. $E_i = O_i - B_i$
2. If $E_i \geq 0$ then

$$\alpha_i \leftarrow 2\alpha_i; \quad \partial M_i \leftarrow \alpha_i \times M_i$$
- Else

$$\alpha_i \leftarrow 0.5\alpha_i; \quad \partial M_i \leftarrow E_i$$
- EndIf

3.4.1 Proof of Classification Error Free

The inequality:

$$\sum_j M_j G_{i,j} - B_i \geq 0 \quad (2)$$

For all $C_i = C_j$ is satisfied when the swarm is converged. Since $B_i = \sum_k Q_k G_{i,k}$ and $Q_k \geq M_k$, Eq. (2) can be rewritten as:

$$\sum_j M_j G_{i,j} - \sum_k M_k G_{i,k} \geq 0 \quad (3)$$

Hence $C_i \times \sum_j M_j G_{i,j} \geq 0$ and no classification error occurred.

$$\bullet G_{i,j} = \exp(-\|\mu_i - \mu_j\| / 2\sigma^2)$$

3.5 Action Unit of ASCN agent

After determining the quantity of magnitude reduced, the agent take an action by simply decrease it magnitude, i.e. $M_i \leftarrow M_i - \partial M_i$. If the adjusted magnitude is larger than the value of excited magnitude, the Eq. (3) cannot be satisfied anymore. Therefore, the Opponent Bias of the neurons with opposite class to A_i have to be adjusted as:

$$B_i \leftarrow B_i + (M_i - Q_i) \times G_{ij}$$

for all $C_i \neq C_j$ and the excited magnitude of A_i is updated as $Q_i \leftarrow M_i$.

4 Probabilistic Noise Removal

Assume that every training sample $S = \{\mu | C\}$ has a Gaussian distribution with center μ and variance σ , the probability H_x that the point x belongs to the class C is $\exp(-\|x - \mu\|^2 / 2\sigma^2)$. For an general case that the pattern contains p training samples $S_i = \{\mu_i | y\}$ where $p \geq i$, H_x is defined as:

$$\sum_i e^{-\frac{\|x - \mu_i\|^2}{2\sigma^2}}$$

Since ASCN is applied on 2-classes classification problem, the value of y is either +1 or -1. Therefore, the probabilistic class at position x is

$$\sum_j C_j e^{-\frac{\|x - \mu_j\|^2}{2\sigma^2}}$$

And hence the training sample S_i is classified as an noise to the pattern if:

$$C_i \sum_j C_j e^{-\frac{\|x - \mu_j\|^2}{2\sigma^2}} < 0$$

5 Experimental Results

5.1 Benchmark Classification Problems

In the first experiment set, we employ the proposed RBF classification network ASCN to classify 2 two-dimensional benchmark classification problems: *Double Spiral* DS pattern and *Exclusive OR* XOR pattern. For each classification problem, 80 training samples with random positions are selected for the DS and XOR patterns, which are illustrated at fig. 1a and 2a respectively. By applying ASCN, the magnitudes of the agents, and hence the neurons' weights are determined. Fig. 1b and 2b show the normalized resultant magnitude plots of the DS and XOR agent swarms respectively. The agent with higher intensity indicates its has relatively higher magnitude. To verify the performance of ASCN, we used the resultant agent swarm to reconstruct the DS and XOR decision surfaces (fig. 1c and 2c). Afterward, the regressive class can be obtained by applying a hard-threshold function on the DS and XOR decision surfaces (fig. 1a and 2d). Those 2 classification problems are processed at the PC platform with 1.7GHz and 256MB memory. All pattern are well trained within 1 seconds and the number of resultant non-zero weight neurons are listed in Table 1.

Table 1 Experimental Results of Section 5.1

Pattern	N _n	Accuracy %	
		(Training Set)	(Testing Set)
Double spiral	37	100	100
XOR	20	100	100

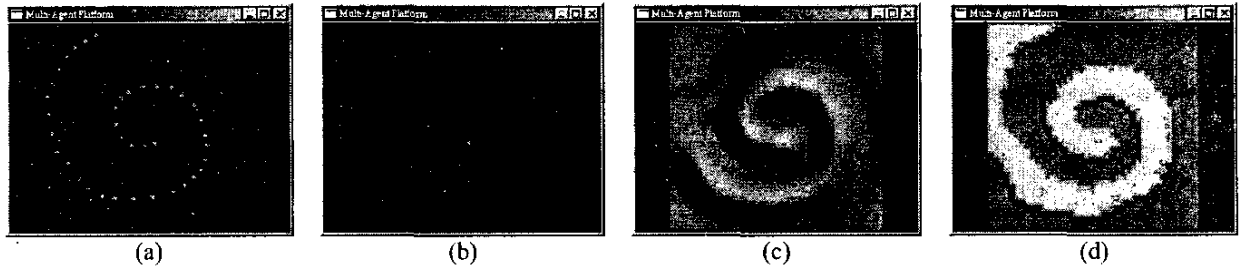


Figure 1 Experimental Results of Section 5.1 (a) Input Patterns. (b) Resultant Magnitude Plots after training. (c) Resultant Decision Surface. (d) Resultant Classification Surface

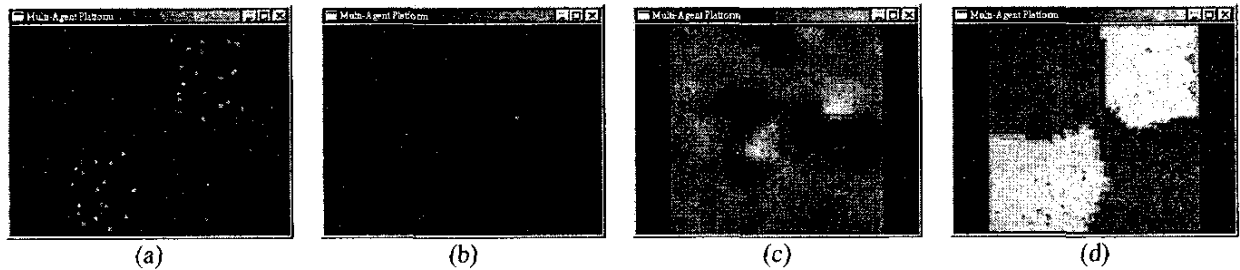


Figure 2 Experimental Results of Section 5.1 (a) Input Patterns. (b) Resultant Magnitude Plots after training. (c) Resultant Decision Surface. (d) Resultant Classification Surface

5.2 Large-Scaled Pattern Classification

In this experiment, the large-scale 700-dimensional patterns with training sample varied from 10,000 to 50,000 are trained by the ASCN. The experiment is executed at the platform same as Experiment 5.1. Table 2 shows the computational time required for the pattern with different swarm (sample) size.

Table 2. Computational Time of Section 5.2

Swarm Size	Computational Time (sec)
10,000	25.08
20,000	45.27
30,000	120.4
40,000	659.85
50,000	7595.5

5.3 2D Noisy Pattern Classification

In this experiment, 2 non-separable 2D patterns (fig. 3a and 4a) are examined. By adopting the proposed noise removal method, the noise samples are outlined (fig. 3b and 4b). Fig. 3c and 4c demonstrate the resultant classification surfaces. This experiment is undergone at the platform same as Section 5.1 with the process time is 2 seconds.

Table 3 Experimental Results of Section 5.3

Pattern Index	Accuracy %	
	(Training Set)	(Testing Set)
1	86	72
2	76	69

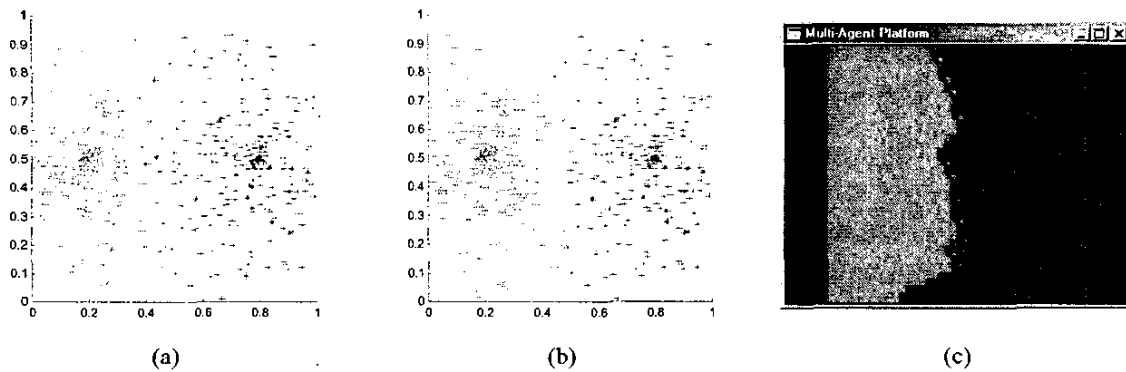


Figure 3 Experimental Results of Pattern 1 at Section 5.3 (a) Input noisy patterns. (b) Input pattern by the proposed noise removal algorithm (c) Resultant classification surface

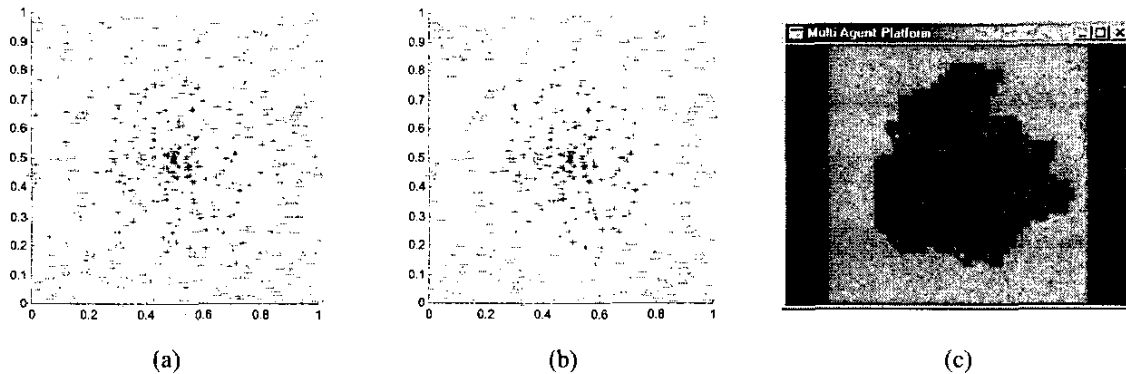


Figure 4 Experimental Results of Pattern 2 at Section 5.3 (a) Input noisy patterns. (b) Input pattern by the proposed noise removal algorithm (c) Resultant classification surface

6 Conclusion

By introducing an additional hard-thresholding layer to the output layer of RBF network, 2-classes classification network can be constructed. This paper proposed to model a classification RBF network by a MAS approach effectively. As we decompose the training algorithm to a

parallel and simple agent behavior, in addition to the neighborhood concept of MAS, the algorithm is accelerated and simpler to be implemented. 3 sets of classification problems were examined to illustrate the acceptable performance of ASCN. The results show that the proposed network ASCN is robust on classification.

References

- [1] Berthold, M.R., "A time delay radial basis function network for phoneme recognition", *IEEE World Congress on Computational Intelligence*, International Conference on Neural Networks 1994, Volume: 7, 27 June-2 July 1994.
- [2] Catelani, M., Fort A., Nosi, G., "Application of radial basis function network to the preventive maintenance of electronic analog circuits", *Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference*, 1999. IMTC/99, Volume: 1, 24-26 May 1999, Page(s): 510 -513 vol.1
- [3] Chen S., Mulgrew, B., Grant P.M., "A clustering technique for digital communications channel equalization using radial basis function networks", *IEEE Transactions on Neural Network*, Volume: 4, Issue: 4, July 1993, Page(s): 570 -590.
- [4] de Freitas N., Milo M., Clarkson P., Niranjani M., Gee A., "Sequential support vector machines", *Neural Networks for Signal Processing IX*, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop, 23-25 Aug. 1999, Page(s): 31 -40
- [5] Hoyt J.D., Wechsler H., "Detection of human speech in structured noise", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1994. ICASSP-94, 1994, Volume: ii, 19-22 April 1994, Page(s): II/237 - II/240 vol.2
- [6] Inhyok Cha, Kassam S.A., "RBFN restoration of nonlinearly degraded images", *IEEE Transactions on Image Processing*, Volume 5, Issue 6, June 1996, Page(s): 964 -975.
- [7] Kovacevic D., Loncaric S., "Radial basis function-based image segmentation using a receptive field" *Proceedings of tenth IEEE Symposium on Computer-Based Medical Systems*, 1997, 11-13 June 1997, Page(s): 126 - 130.
- [8] Paquet U., Engelbrecht A.P., "Training support vector machines with particle swarms", *Proceedings of the International Joint Conference on Neural Networks*, 2003. Volume: 2, July 20 -24, 2003, Page(s): 1593 -1598
- [9] Sun-Gi Hong, Sang-Keon Oh, Min-Soeng Kim, Ju-Jang Lee, "Nonlinear time series modelling and prediction using Gaussian RBF network with evolutionary structure optimization", *Electronics Letters*, Volume: 37, Issue: 10, 10 May 2001, Page(s): 639 -640.
- [10] Tarassenko I., Roberts S., "Supervised and unsupervised learning in radial basis function classifiers", *IEE Proceedings- Vision, Image and Signal Processing*, Volume: 141 Issue: 4, Aug. 1994, Page(s): 210 -216
- [11] V. N. Vapnik, C. Cortes, "Support Vector Networks", *Machine Learning*, 20(3), 273-297, 1995.