Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 30 (2012) 989 – 996

Procedia Engineering

www.elsevier.com/locate/procedia

International Conference on Communication Technology and System Design 2011

Collaborative Artificial Bee Colony Optimization Clustering Using SPNN

D.Shanthi[a], R.Amalraj[b], a*

[a]Mother Teresa University, Kodaikannal – 624 102,Tamilnadu,India
[b]Sri Vasavi College, Erode – 638 316, Tamil Nadu, India

Abstract

This paper deals with methods for clustering of continuous signals such as time series data sets. Centers of classes are determined with the help of the neural network with process input, which is an extension of the traditional artificial neural network into the time domain. Collaborative Artificial Bee Colony algorithm is based on the search of food behaviour of honey bees for training in a non-trajectory way. An Enhancement has been done to the original Artificial Bee Colony (ABC) algorithm and was used to discover suitable domain specific architectures. The C-ABC has great explorative search features and better convergence compared to the original algorithm and it was proved empirically that it avoids local minima by promoting exploration of the search space. In SPNN(Self Organizing Process Neural Network), the inputs and weights are related to instantaneous conditions. The proposed algorithm results in clustering the data sets with reduced error rate and better convergence rate. The tests are conducted on empirical data in matlab.

KeyWords: Neural Networks; C-ABC Algorithm; SPNN; Artificial Bee Colony; Optimization;

1. Introduction

In neural networks research at present, the most popular and effective model is a feedforward neural network. It is quite successful in many domains, such as pattern recognition, classification and clustering, adaptive control and learning, etc. Clustering is a common problem in signal processing and combinatorial analysis. When the class number is unknown, the method of merging samples into classes is called “clustering”. For clustering, the classification structure of research objects does not need to be known beforehand, and it can be classified according to similarities among the research objects, which are not restricted by the current level of study of research objects and prior knowledge. The feedforward neural network (such as a self-organizing mapping neural network without teaching) adopting the self-organizing competitive learning algorithm with process input is a young approach to obtain the perfect cluster, which is broadly applied to many fields including data mining, association analysis, etc.

2. SPNN Structure:

Self-organizing process neural networks have a two-layer structure consisting of the input layer and the competitive layer composed of process neurons. All nodes in the input layer and the competitive layer connect fully with one another, and their input signals and the connection weights of the network may be time-dependent functions. The

* D Shanthi. Tel.: +91-909-533-9048;
E-mail address: shanthi_mimina@yahoo.co.in.

1877-7058 © 2011 Published by Elsevier Ltd.
doi:10.1016/j.proeng.2012.01.955

network adaptively extracts connotative pattern characteristics in input functions, carries out self organization, and exhibits the action result in the competitive layer.

Without loss of generality, suppose that the input space of the network is $(C[0,T])^n$ where [0,T] is the signal input process interval. Suppose that the input function of the system is $X(t)=(x_1(t),x_2(t), \ldots ,x_n(t))$, and the output is a static vector representing a pattern class. The topological structure of the network

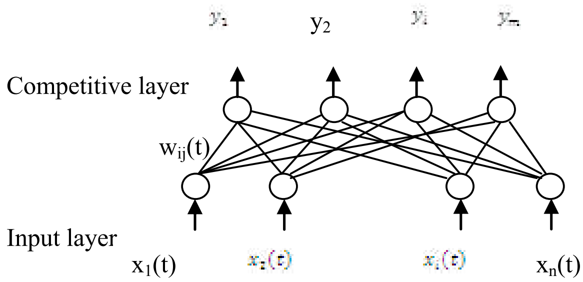

Fig 1. Self-organizing process neural network

$w_{ij}(t)$ (i=1,2,.. .,n; j=1,2,… ,m*)* is the connection weight function between the input node i and the competitive process neuron node j; y_j (j=1,2,… ,m) is the output of the process neuron j. In practice , they represent a group of similarity degree.

2.1. Learning Algorithm

Suppose that the training sample set for the network is $\{X^1(t),X^2(t),\ldots ,X^K (t)\}$ where $X^k(t) \in (C[O,T])^n$. All these samples belong to one of *m* given pattern classes according to some criteria. The output of the competitive layer node represents a pattern class and the weight function connected with the node includes basic characteristic information of the pattern class.

2.1.1. Competitive learning algorithm

2.1.1.1. Competitive learning rule

When the network is trained, the training samples $X^1(t),X^2(t) ,\ldots,X^k(t)$ are imported at the input terminal according to determinate or random order . The total weighted input signal from various nodes in the input layer to the process neuron node *j* in the competitive layer is

$$s_j(t)=\sum_{i=1}^{n} w_{ij}(t)x_i(t), j = 1,2, \ldots m \quad (1)$$

If the similarity coefficient of the k^{th} input sample vector $X^k(t)$ and the connection weight function vector $W_j(t)$ of the neuron node *j* is considered then the competitive layer is

$$r_j^k=\frac{(X^k(t),W_j(t))}{\|X^k(t)\|.\|W_j(t)\|} \quad (2)$$

$$r_j^k=\frac{\int_0^T \left(\sum_{i=1}^{n} x_i^k(t),W_{ij}(t)\right)dt}{\left(\int_0^T \left(\sum_{i=1}^{n}(x_i^k(t))^2\right)dt\right)^{\frac{1}{2}}\left(\int_0^T \left(\sum_{j=1}^{n}(w_{ij}(t))^2\right)dt\right)^{\frac{1}{2}}} \quad (3)$$

Where $W_j(t) = (w_{1j}(t), w_{2j}(t),,\ldots, w_{nj}(t)$ for j=1 , 2, …m.
Here, the node j* with the maximal similarity coefficient wins in the competition, i.e. j* satisfies

$$r_{j^*}^k = \max_{j\in\{1,2,\ldots,m\}}\{r_j^k\} \quad (4)$$

For the input sample vector $X^k(t)$, if the node j^* wins in the competition, then the weights are adjusted according to the following rule: when the network again encounters the input $X^k(t)$ or an input sample vector similar to $X^k(t)$ the winning probability of the node j^* is increased, i.e. $w_{ij}(t)$ (i=1,2,. . .,n;j=I,2,…,m) is adjusted so as to make the weight function $W_{j*}(t)$ move toward the sample $X^k(t)$ by algorithm adjusting, and finally make the output of the wining neuron j^* represent the pattern class that $X^k(t)$ represents.