Particle Swarm Optimization Based Learning Method for Process Neural Networks

Kun Liu, Ying Tan*, and Xingui He

Key Laboratory of Machine Perception, Ministry of Education,
Peking University, Beijing 100871, China
ytan@pku.edu.cn
http://www.cil.pku.edu.cn

Abstract. This paper proposes a new learning method for process neural networks (PNNs) based on the Gaussian mixture functions and particle swarm optimization (PSO), called PSO-LM. First, the weight functions of the PNNs are specified as the generalized Gaussian mixture functions (GGMFs). Second, a PSO algorithm is used to optimize the parameters, such as the order of GGMFs, the number of hidden neurons, the coefficients, means and variances of Gaussian functions and the thresholds in PNNs. In PSO-LM, the parameter space is transformed from the function space to real number space by using GGMFs. PSO can give a global search in the real number parameter space by avoiding the premature and gradient calculations in back propagation method. According to our analysis and several experiments, PSO-LM can outperform current basis function expansion based learning method (BFE-LM) for PNNs and the classic back propagation neural networks (BPNNs).

Keywords: Process Neural Networks, Learning Method, Particle Swarm Optimization, Gaussian Mixture Model.

1 Introduction

Recently, He et al. proposed this process neural networks (PNNs) to deal with process inputs, which can be related to time, locations etc. [1]. Inputs such as time series are usually sensitive to the time parameter, so the weights and thresholds should also be related to time in order to accumulate the effects of inputs more precisely. In PNNs, there exist process neurons (PNs). The inputs, connection weights and threshold of a PN can all be functions of time. Theoretically, the approximation capabilities of PNNs are better than classic artificial neural networks (ANNs) when solving for time-varying function problems. ANNs are considered as the special case of PNNs. PNNs have been applied in many actual tasks, such as simulation of oil reservoir exploitation [1] and traffic flow prediction [2].

In this study, we propose this new learning method for PNNs. Instead of transforming the inputs and weight functions, we specify the weight functions connected to the process neurons as the generalized Gaussian mixture functions (GGMFs), which theoretically can approximate any continuous functions. Afterward, instead of using the

* Corresponding Author.

L. Zhang, J. Kwok, and B.-L. Lu (Eds.): ISNN 2010, Part I, LNCS 6063, pp. 280–287, 2010.

classic back propagation method, we use PSO to optimize the parameters, such as the order and coefficients of GGMFs, the means and variances of Gaussian functions, the number of hidden neurons and their thresholds. This new learning method for PNNs is called PSO-LM for short.

The remainder of this paper is organized as follows. Section 2 describes the original PNN model and a standard PSO model. The PSO-LM is proposed in Section 3 with discussions about its qualities. Section 4 elaborates two experiments to compare the PSO-LM with the existing BFE-LM for PNNs and the BPNN. Concluding remarks are drawn in Section 5.

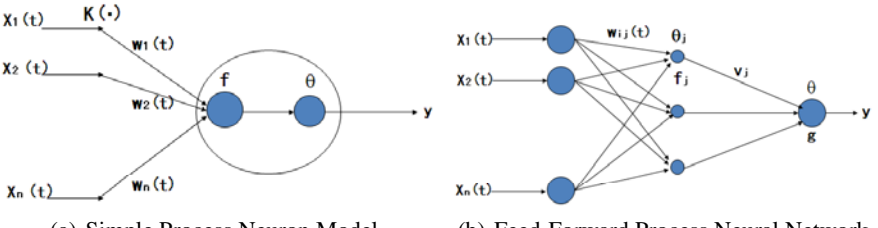
2 Background

2.1 Process Neural Networks

PNN is a generalized model of traditional neural networks, the inputs are related to instantaneous conditions [1]. The key of PNNs is the special structure of PNs. Fig. 1(a) shows a simple process neuron model (SPN), in which $x_i(t)(i = 1, 2, \ldots, n)$ are the input time-varying functions and $w_i(t)(i = 1, 2, \ldots, n)$ are the corresponding weight functions. $K(\cdot)$ is the aggregate function of the SPN, which can be summation or integration etc.. $f(\cdot)$ is the activation function, with a threshold value θ. The output of the SPN is a constant. A simple feed forward process neural network (PNN) is shown in Fig. 1(b).

The current learning methods for PNNs include: basis function expansion method (BFE) [1] and numerical learning method (NL) [3]. The BFE includes two steps: first, transform the input data and the weight functions of the PNN into the space that expanded by some specified orthogonal basis functions. Therefore, there will be no time-varying parameters in the model. Second, adjust the parameters in the networks by back propagation method. However, how to choose appropriate basis functions and how many expansion items should be reserved are awkward to decide. Furthermore, transforming the input data in the first place certainly will lose information contained in the input signals, which can restrain the performance of PNNs.

The NL method is only suitable for small-scale discrete input signals. When the length of the input series increases, the number of parameters in NL will increase



(a) Simple Process Neuron Model (b) Feed-Forward Process Neural Network

Fig. 1. Process Neural Networks

rapidly, which can lead to huge computational complexity. Furthermore, these two learning methods both rely on the back propagation (BP) method, which will suffer from the premature convergence problem and gradient calculations.

2.2 Particle Swarm Optimization

Particle swarm optimization (PSO) [4], as a stochastic global optimization technique based on a social interaction metaphor, can be used to train ANNs effectively [5].

Bratton and Kennedy defined a standard PSO model (SPSO) in [6]. The update rules can be equivalently transformed and expressed as follows:

$$\begin{aligned} V_{id}(t+1) &= \omega V_{id}(t) + c_1\epsilon_1(P_{iBd}(t) - X_{id}(t)) \\ &\quad + c_2\epsilon_2(P_{nBd}(t) - X_{id}(t)), \end{aligned} \tag{1}$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1). \tag{2}$$

where $i = 1, 2, \cdots, n$, n is the number of particles in the swarm, $d = 1, 2, \cdots, D$, and D is the dimensionality of solution space. The learning factors c_1 and c_2 are nonnegative constants, r_1 and r_2 are random numbers uniformly drawn from the interval $[0, 1]$, which are all scalar quantities for each particle in each dimension. P_{iBd} and P_{nBd} are the locations of the best positions found so far by particle i and its neighbors in dimension d. ω is called inertia weight, which determines the effect of the inertia to the movements of particles.

3 The Proposed PSO-LM for PNNs

It is impossible to learn the weight functions directly from the function space. In this study, we assume that all weight functions are GGMFs based on the fact that all the continuous functions can be approximated by GGMFs. Afterwards, the constant parameters in the model are optimized by PSO.

3.1 Generalized Gaussian Mixture Functions

A standard Gaussian mixture model is defined in (3), and its qualities are expressed in Theorem 1. For the proof of the theorem, one can refer to [7].

$$f = \sum_{i=1}^{k} a_i N(u_i, \sigma_i), \tag{3}$$

where $N(u_i, \sigma_i)$ is the normal distribution with mean u_i and variance σ_i, $\Sigma_{i=1}^{k} a_i = 1$, $a_i \geq 0, i = 1, 2, \ldots, k$ and k is the order of the model.

Theorem 1 (Gaussian Mixture Model Approximation Theorem). *Finite Gaussian mixture model can approximate a non-negative Riemann integrable function in real number field by any degree of accuracy. Particularly, it is able to approximate any probability density functions by any degree of accuracy.* □

The generalized Gaussian mixture function is also defined as in (3), only with the difference that $a_i \in R$. Therefore, one can derive the following corollary. The proof of this corollary is straight according to the above theorem.

Corollary 1. *Generalized Gaussian mixture functions can approximate a continuous function in real number field by any degree of accuracy.* □

3.2 Learning Method for PNNs Based on PSO (PSO-LM)

Consider the feed-forward PNN shown in Fig. 1(b). Only the hidden neurons are simple process neurons, and the other parameters are all constants. Theoretically, the approximation capability of this PNN can be retained if the weight functions are specialized as generalized Gaussian mixture functions.

There are n input neurons. The following parameters will be optimized by PSO according to specific objectives: the order of GGMFs (k), the number of hidden neurons (m), the coefficients, means and variances of Gaussian functions corresponding to the weight functions that connect from input neurons to hidden neurons ($a_{ijl}, u_{ijl}, \sigma_{ijl}, i = 1, \ldots, n; j = 1, \ldots, m; l = 1, \ldots, k$), the thresholds of hidden neurons ($\theta_j, j = 1, \ldots, m$) and the output neuron (θ).

So, a particle in the swarm of PSO can be expressed as:

$$p = (k, m, a_{ijl}, u_{ijl}, \sigma_{ijl}, \theta_j, w_j, \theta), i = 1, \ldots, n; j = 1, \ldots, m; l = 1, \ldots, k. \quad (4)$$

Each position of the particle in the search space represents one process neural network. The PSO will search for a PNN that has appropriate structure and parameter values to optimize the objective function, which is the performance evaluation of the PNN. When the swarm of PSO converges to one position, which is supposed to be the best parameter setups of the PNN, the learning process terminates.

According to the above study, the PSO-LM for PNNs can be summarized as in Algorithm 1:

Algorithm 1. PSO-LM for PNNs

Step 1: Set up the objective function of the training as the fitness function of PSO and the generalized Gaussian mixture functions as the weight functions of PNs shown in Fig. 1(a).
Step 2: Set parameters of PSO (particle number p, w, c, search space borderlines).
Step 3: Each particle in PSO is formed according to (4).
Step 4: Run PSO.
Step 5: Stop criterion (maximum iterations or/and fixed learning precision).
Step 6: If the result is not satisfactory, go back to Step 3.

3.3 Discussions

The proposed PSO-LM has several advantages comparing with the existing learning methods for PNNs.

First, PSO-LM does not need any transformation to the input signals, which will not lose any information carried in inputs. BFE-LM extracts the principal components of the input data by using the basis function expansion, which will retain most of the information. However, the detailed information carried in inputs is the most needed information in some task. For example, in pattern recognition, two patterns might have the same first and second principal components, which will be reserved according to BFE-LM. But, it is the detailed differences carried in their third components that can distinguish them, which will be eliminated by BFE-LM. Thereafter, samples from different classes can be misclassified into the same pattern. Furthermore, the transformations also bring more computation.

Second, gradient information is no longer needed in PSO-LM. The structure and the parameters of the PNNs can be optimized simultaneously by PSO. The learning capability will be enhanced, which is guaranteed by the strong optimization capability of PSO.

Furthermore, PSO-LM is able to solve high dimensional data learning problems by using high dimensional GGMFs as the weight functions. PSO-LM can also solve multi-objectives training problems as PSO is good at doing multi-objective optimization.

The computational complexity of PSO-LM for PNNs can be expressed as $C = F \times P \times I$. F is the amount of one time feed-forward calculation of PNN which is also one time fitness calculation for PSO, P is the number of particles in PSO, and I is the maximum number of iterations. The generalization performance of PSO-LM can be verified by the following experiments.

4 Experiments

4.1 Traffic Flow Prediction

The data is from $PeMS$ system [8], which is a performance measurement system of freeway traffic from University of California. In this study, we chose traffic volume data recorded every five minutes in the time interval 10:00 to 15:00 during the weekdays from 2009/6/1 to 2009/6/21. The data is taken from five sequential loop detectors, whose serial numbers are 716810, 717907, 717898, 717896 and 717887 in District 7.

Experimental Setups. The objective is to predict the traffic flow of loop 717898, which lies in the middle of the five loops, in the next five minutes. We will use the previous five records from all the five loops to predict the next value of the third loop [3]. The data from the first two weeks which contains 120 samples is used to train the three models mentioned above, and the data of the third week which has 55 samples is used as the test data to test their generalization performances.

The parameters of the three models are listed in Table 1. The number of $db1$ wavelet basis functions chosen in BFE-LM is ten, which is decided by trial-and-error and is also the number of neurons in the first hidden layer of BFE-LM. k is the order of the GGMFs and m is the number of neurons in hidden layer of PSO-LM, which will both be decided by PSO to optimize the objective function. Finally, the activation functions of neurons are all chosen to be $purelin$ function according to experiments. Notice that $k = 2$ and $m = 13$ is decided by PSO.

Table 1. Parameter Setups For Traffic Flow Experiment

	Structures	*Activation Functions*	*Parameters*
$BPNN$	5-25-1	purelin, purelin	
$BFE-LM$	5-10-25-1	purelin, purelin	$db1$ wavelet basis
$PSO-LM$	5-m-1	purelin, purelin	k=2, m=13

The evaluation of the models depends on the following two targets:

$$MPAE = \frac{1}{n}\sum_{t=1}^{n}\frac{|\ y_t - \widehat{y}_t\ |}{y_t} \tag{5}$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \widehat{y}_t)^2}{n}} \tag{6}$$

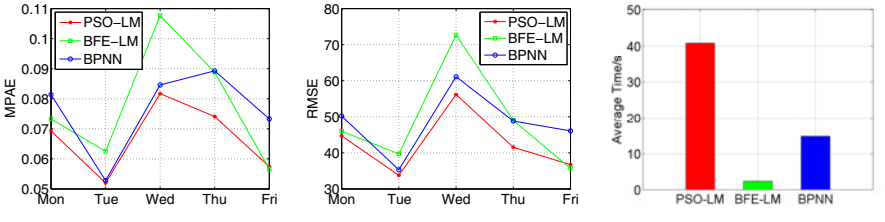
We will compare the models on each day of the weekdays independently. Each of the models ran 30 times independently, and the averaged results were recorded.

Experimental Results. The averaged results are shown in Fig. 2. As can be seen, PSO-LM outperforms BFE-LM and BPNN. According to the variance analysis, the p values of $MPAE$ and $RMSE$ are 0.013 and 0.034, which mens the improvements of the performance by PSO-LM is statistically significant.

The PSO-LM can learn the data well and predict the traffic flow more precisely, which indicate that PSO-LM can give PNNs more powerful learning capability and also good generalization capability. PSO-LM spent the most time to achieve the best performance for its global search strategy. BFE-LM consumes the least time, because the basis function expansion for the inputs retains only the principal components of the data, which makes it easier for NNs to learn. Nevertheless, BFE-LM can not guarantee good performances.

4.2 Spam Detection

In this experiment, the three models will be compared in identifying spam on a standard $Spambase$ data [9]. There are 4601 samples, each of which has 57 attributes and one



(a) MPAE (b) RMSE (c) Time Consuming

Fig. 2. Experimental Results For Traffic Flow Forecasting

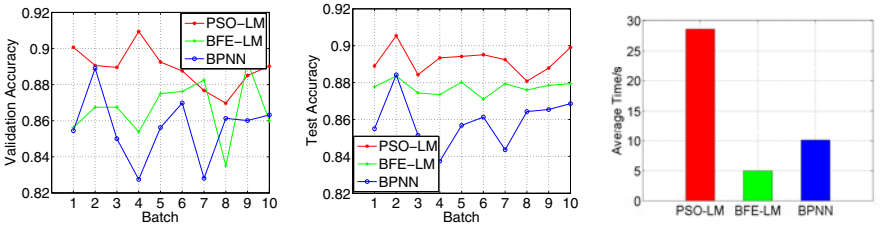
label. The label denotes whether the e-mail is considered as a spam (1) or not (0). In this study, we will use the first 48 attributes which denote the frequencies of 48 words occurring in the e-mails.

Experimental Setups. We use cross-validation in this experiment. First, 601 samples are randomly chosen to join a test data. Then, we use the rest 4000 samples to perform the standard ten times cross-validation to train and validate the models. Finally, the trained models will be tested on the test data to show their generalization capabilities. Each model will run 30 times independently to get statistical results. The parameters in this experiment are shown in Table 2. All the parameters are with the same meanings as in Table 1.

Table 2. Parameter Setups For $Spambase$ Experiment

	Structures	*Activation Functions*	*Parameters*
$BPNN$	48-15-1	tansig, logsig	
$BFE-LM$	1-10-5-1	tansig, logsig	$db1$ wavelet basis
$PSO-LM$	1-m-1	purelin, logsig	k=4, m=16

Experimental Results. The averaged results are shown in Fig. 3. The validation accuracies of the ten times cross-validation are shown in Fig. 3(a). As can be seen, PSO-LM can obviously outperform the other two models, which indicates that the PSO-LM has the best learning capabilities. The test accuracies are shown in Fig. 3(b). Similarly, PSO-LM has the best generalization capability. Time consuming comparison is shown in Fig. 3(c). As the same reason as in the above experiment, the PSO-LM needs more time to gain better performance that the other models can not achieve. Furthermore, the p values of the validation and test accuracy are 0.0486 and 0.0491, which means the performance improvements made by PSO-LM are statistically significant.



(a) Validation Accuracy (b) Test Accuracy (c) Time Consuming

Fig. 3. Experimental Results For Spam Detection

5 Conclusion

This paper proposed a new learning method for process neural networks based on the Gaussian mixture weight functions and particle swarm optimization (PSO-LM). The weight functions were specified as the generalized Gaussian mixture functions. The structure and parameters in the model were then optimized by PSO. According to experiments, PSO-LM had better performance on time-series prediction and pattern recognition than BFE-LM and BPNNs. Although it needed more computations for the global search strategy of PSO, it achieved better performances that other methods can not gain.

Acknowledgement

This work is supported by National Natural Science Foundation of China (NSFC), under grant number 60875080 and 60673020, and partially financially supported by the Research Fund for the Doctoral Program of Higher Education (RFDP) in China. This work is also in part supported by the National High Technology Research and Development Program of China (863 Program), with grant number 2007AA01Z453.

References

1. He, X.G., Xu, S.H.: Process Neural Networks. Science Press, Beijing (2007)
2. He, S., Hu, C., Song, G.J., Xie, K.Q., Sun, Y.Z.: Real-Time Short-Term Traffic Flow Forecasting Based on Process Neural Network. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008, Part II. LNCS, vol. 5264, pp. 560–569. Springer, Heidelberg (2008)
3. Wu, T.S., Xie, K.Q., Song, G.J., He, X.G.: Numerical Learning Method for Process Neural Network. In: Yu, W., He, H., Zhang, N. (eds.) ISNN 2009. LNCS, vol. 5551, pp. 670–678. Springer, Heidelberg (2009)
4. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. of the IEEE International Conference on Neural Networks 1995, pp. 1942–1948 (1995)
5. Van Den Bergh, F., Engelbrecht, A.P.: Training Product Unit Networks Using Cooperative Particle Swarm Optimization. In: Proc. of the IEEE International Joint Conference on Neural Networks 2001, vol. 1, pp. 126–131 (2001)
6. Bratton, D., Kennedy, J.: Defining a Standard for Particle Swarm Optimization. In: Proc. of the IEEE Swarm Intelligence Symposlum (SIS), pp. 120–127 (2007)
7. Yuan, L.H., Li, Z., Song, J.S.: Probabillty Density Function Approximation Using Gaussian Mixture Model. Radio Communications Technology 33(2), 20–22 (2007)
8. Freeway Performance Measurement System (PeMS), `http://pems.eecs.berkeley.edu`
9. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. In: School of Information and Computer Science. University of California, Irvine (2007), `http://archive.ics.uci.edu/ml/datasets/Spambase`