# PCA-PSO-BP Neural Network Application in IDS

Lan SHI[a], YanLong YANG[b] JanHui LV[c]

College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China

[a]shilan@mail.neu.edu.cn, [b]yangyanlongedu@163.com, [c]lvjianhui2012@163.com

**Keywords:** BP neural network; particle swarm optimization; principal components analysis, intrusion detection

**Abstract.** BP neural network has two disadvantages, one is to fall into local minimum value easily; the other is the slow convergence. We propose in this paper an approach, including three main operations. Firstly, the algorithm of particle swarm optimization (PSO) is applied to improve back propagation (BP) neural network. Secondly, principal components analysis (PCA) method is used to deal with the original information. Thirdly, after optimization of BP neural network, we employ it into the intrusion detection system. The simulation results reveal that the new proposed BP neural network is superior to the traditional BP neural network.

## Introduction

Neural network plays an important role in the domain of intelligent control, and it has made great progress in every domain, such as neurosciences, mathematics, statistics, computer science etc. Now, BP neural network application is one of the most widespread applications, this is due to the fact that BP neural network algorithm is simple and plastic. But there exits two disadvantages, falling into local minimum easily and having a slow convergence. For these problems many researchs scholars proposed a lot of solutions that including Conjugate gradient, Newton, Causs-Newton, Levenberg-Marguard methods etc [1] [2] [3] [4]. Also the problems above have a great improved. But the calculated quantity of these methods is relatively large and the methods fail at dealing with large-scale data. In addition, some experts so far have proposed some hybrid intelligent algorithms, like combining BP neural network with artificial immune or particle swarm algorithms. These combinations make BP neural network a better efficiency and more widely application. According to this idea, this paper adopts particle swarm optimization (PSO) to optimize the BP neural network. And then apply it to the intrusion detection. Most of the data from network is multidimensional and noisy. So in this paper, data is processed by principal components analysis method first.

The paper is organized as follows. Section 2 introduces the principle of BP neural network and PSO. The detailed design of PCA-PSO-BP neural network is designed in section 3. Intrusion detection system and simulation results are designed in section 4 and section 5. Finally we concludes this paper.

## BP neural network and PSO introduction

Since Rulmhart and parallel distributed processing (PDP) group put forward BP algorithm in 1986[5], and then the artificial neural network researchers began to pay attention on BP neural network. But the biggest flaw of BP neural network is slow learning efficiency and slow convergence. Particle swarm optimization algorithm is put forward by Kennedy and Eberhar in the United States in 1995, a kind evolutionary algorithm based on intelligence [6]. And its aims at to simulate the unpredictable movement of birds. The main idea is to regard a solution from the problem as a particle $i$. The particle $i$ refers to one bird. The particle's process of searching for its optimal solution refers to the process of searching for food. The above process can be described in Figure.1.

 

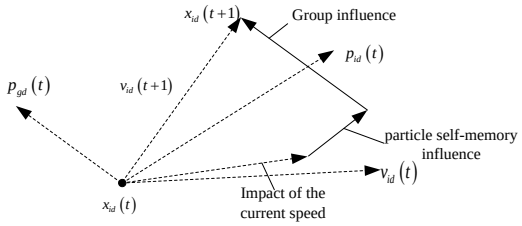


Fig.1. Particle position update of PSO algorithm

Among them, $p_{best}$ represents the d-th dimensional components of the best position, which refers to i-th individual particle. $p_{gd}$ represents the d-th dimensional components of the optimal value $g_{best}$ in the particle swarm; $V_{id}(t)$ represents the d-th dimensional components of particle i's speed after iterating for a t-th time; $x_{id}(t)$ represents the d-th dimensional components of particle i's position after a t-th time. Particle swarm optimization has the advantages of smaller calculation and the rapid convergence in dealing with the high-dimensional coples functions more extremes. So particle swarm optimization can optimize the BP neural network [7].

## Algorithm digital designing

Data from network usually are multidimensional. For this problem, this paper combines the BP neural network with principal components analysis method. It can reduce the data dimensions effectively. Now we have analyzed the advantages and the disadvantages of BP neural network, particle swarm optimization [8]. The PCA-PSO-BP Neural network algorithm digital steps are in the following:

**Step 1:** Principal components analysis deals with the captured data. Firstly, the captured data are normalized. We get the standard matrix of $x_1$, we have

$$x'_{ij} = \frac{x_{ij} - x_{j\min}}{x_{j\max} - x_{j\min}} \tag{1}$$

In the formula (1), $x_{j\max}$ represents the maximum value of the digital data, $x_{j\min}$ represents the minimum value of digital data. Then solve the coefficient matrix R. We have formula

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ r_{p1} & r_{p2} & \cdots & r_{pp} \end{bmatrix} \tag{2}$$

$$r_{ij} = \frac{\sum_{k=1}^{n} \left( x_{ki} - \bar{x}_i \right) \left( x_{kj} - \bar{x}_j \right)}{\sqrt{\sum_{k=1}^{n} \left( x_{ki} - \bar{x}_i \right)^2 \left( x_{kj} - \bar{x}_j \right)^2}} \tag{3}$$

Then calculate the characteristic equation of R. $\left| R - \lambda I_p \right| = 0$. We will get the values of $\lambda$, $\mu$, $e_m$, and $E_n$. Now we have got the principal components of number m and number m as an input for the BP neural network.

Among them

$$e_m = \frac{\lambda_i}{\sum_{i=1}^{p} \lambda_i} \tag{4}$$

$$E_m = \frac{\sum_{j=1}^{m} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \tag{5}$$

**Step 2:** BP neural network initializing. The number m as input layer of BP neural network determined by $\lambda$. It is matrix's dimension. The number N as output layer determined by the output properties. The number Q as hidden layer, but it has no exact mathematical theorem proving how much to take. So we usually take the experience function Q = (M+N)/2.

**Step 3:** Particle swarm optimization algorithm optimizes BP neural network. Each input of neural network seen as a particle of particle swarm optimization. We initialize the speed matrix and the position matrix of each particle. Position matrix will be given a random number between 0 and 1. Calculate the fitness of each particle. We have the formula

$$fitness = \frac{1}{2} \sum_{i=1}^{N} \left( y_{real} - y_i \right)^2 \tag{6}$$

In the formula (6), N represents the number of training samples; $y_{real}$ represents the true value of the first i samples; $y_i$ represents the predictive value of the first i samples. The optimal solution is the position of minimum fitness of particle.

**Step 4:** Compare the fitness of each particle. We get each particle extremes value and the global optimal value.

If Present < $P_{best}$, Present = $P_{best}$, $P_{best} = x_i$; else $P_{best}$ the same. If Present < $g_{best}$, Present = $g_{best}$, $g_{best} = x_i$; else $g_{best}$ the same.

Among them, Pr*esent* represents the current fitness of the particle, $P_{best}$ represents the particle itself extremes value, the $g_{best}$ represents the global optimal value.

**Step 5:** Update speed and position of each particle.

When the particle after one iteration, its speed $V_{id}$ and the $X_{id}$ will be update by the individual extreme value and the globe extreme value. The formula as follows:

$$V_{id}(k+1) = wV_{id}(k) + c_1 r_1 \left( P_{id}(k) - X_{id}(k) \right) + c_2 r_2 \left( P_{gd}(k) - X_{id}(k) \right) \tag{7}$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \tag{8}$$

Among them, w represents the inertia weight $c_1$ and $c_2$ represents the acceleration factor, $r_1$ and $r_2$ spread from 0 to 1.

**Step 6:** Update the individual extreme value and the globe extreme value. Comparing the individual moderate value and the globe moderate value before iteration, get the best represents the individual extreme value.

**Step 7:** Controlling the iteration stop.

Through the termination condition of the algorithm, we judge the algorithm stop or not. If number of iteration reach the maximum or reach the expectant error then turning to the **Step 7**, else go back to **Step 3**.

## Model application

Now we put the algorithm into the intrude detection. The structure has six parts.

(1) Data capture module

The data capture module most use to capture the data from the network. Then the module put the data into the intrusion detection.

(2) Protocol analysis module

The protocol analysis module uses to analysis the protocol of the data.

(3) Pretreatment module

The pretreatment module will use the principal components analysis to deal with the data before it put into next module.

(4) Intrude match module

This module is the core of the whole system. According to the rules of the corresponding chain table, the module detect if the invade is happen. If the intrude is happen, it will touch off the response module. Else it turns to the BP neural network module.

(5) PCA-PSO-BP neural network module

Through the PCA-PSO-BP neural network module, we will take the anomaly detection. If the result departure the normal action, it will it will touch off the response module. At the same time it sent the action to the intrude action to the intrude model library. From this it can realize self-learning.

(6) Response module

When the system detects the intrude action it will give a warning and record the action.

**Simulation evaluation**

The configuration of computer uses in this simulation as follows: Intel(R) Core(TM)2 Quad CPU Q9500 @ 2.83GHz，4 GB of memory, 500 GB of hard disk, window 7 (32bit) operating system, C as the language, situation uses Matlab2007.

Now let detecting the recognition performance of TCP neural network and UDP neural network. They are designed by the PCA-PSO-BP neural network. The data is collected from Kdd Cup99. Usually we will take half of the data to use as practice and another half will use to simulation testing.

For the TCP protocol of input data, first we deal it use the PCA. So we get the results of have more affect to the variables. The results of TCP's PCA are in Table1.

Table1. Result of principal component analysis of TCP protocol

| NO. | Eigenvalues | Contribution Rate | Cumulative Contribution Rate |
|---|---|---|---|
| 1 | 24.9848 | 73.2569 | 73.2569 |
| 2 | 3.6776 | 10.7832 | 84.0401 |
| 3 | 1.9781 | 5.8001 | 89.8402 |
| 4 | 0.8041 | 2.3576 | 92.1978 |
| 5 | 0.4049 | 1.1871 | 93.3849 |
| 6 | 0.3767 | 1.1046 | 94.4895 |
| 7 | 0.1744 | 0.5114 | 95.0009 |

From Table1 we can see that the digital data with 41 dimensions has reduced to 7. But they have a high contribution rate of 95%. Then the result of simulation we can get:

Table2. Detecting result of traditional BP neural network relate to TCP protocol

| Type | samples number | Output number | Correct output | Error output number | Correct rate |
|---|---|---|---|---|---|
| Normal | 2000 | 1989 | 1911 | 78 | 95.55% |
| Teardrop | 2000 | 1971 | 1876 | 95 | 93.80% |
| Ipsweep | 500 | 482 | 470 | 12 | 94.00% |
| Back | 1000 | 965 | 929 | 36 | 92.90% |
| Smurf | 1000 | 963 | 922 | 41 | 92.20% |
| Total | 6500 | 6370 | 6108 | 262 | 93.97% |

Table3. Comparison of detecting result of traditional BP and TCP neural network

| Type | samples number | Output number | Correct output | Error output number | Correct rate |
|---|---|---|---|---|---|
| Normal | 2000 | 1688 | 1501 | 187 | 75.05% |
| Teardrop | 2000 | 1678 | 1420 | 258 | 71.00% |
| Ipsweep | 500 | 418 | 360 | 58 | 72.00% |
| Back | 1000 | 849 | 680 | 169 | 68.00% |
| Smurf | 1000 | 867 | 713 | 154 | 71.30% |
| Total | 6500 | 5500 | 4674 | 826 | 71.91% |

From the Table2 and Table3 we can get the different between tradition BP and the TCP-BP. In Tbale4, we can see the result clearly.

For the input of UDP, we deal with the same as the TCP. So the result as follows:

Table4. Result of principal component analysis of UDP protocol

| NO. | Eigenvalues | Contribution Rate | Cumulative Contribution Rate. |
|---|---|---|---|
| 1 | 14.9715 | 38.7569 | 38.7569 |
| 2 | 13.6469 | 35.3279 | 74.0848 |
| 3 | 6.4008 | 16.5906 | 90.6754 |
| 4 | 1.1783 | 3.0503 | 93.7257 |
| 5 | 0.3124 | 0.8087 | 94.6118 |
| 6 | 0.1529 | 0.3958 | 95.0076 |

Table5. Detecting result of UDP neural network

| Type | samples number | Output number | Correct output | Error output number | Correct rate |
|---|---|---|---|---|---|
| Normal | 2000 | 1987 | 1898 | 89 | 94.90% |
| Smurf | 2000 | 1961 | 1879 | 82 | 93.95% |
| Ipsweep | 500 | 482 | 450 | 32 | 90.00% |
| Neptune | 1000 | 951 | 927 | 24 | 92.70% |
| Total | 5500 | 5381 | 5154 | 227 | 93.71% |

Table6. Detecting result of traditional BP neural network relate to UDP neural network

| Type | samples number | Output number | Correct output | Error output number | Correct rate |
|---|---|---|---|---|---|
| Normal | 2000 | 1676 | 1498 | 178 | 74.90% |
| Smurf | 2000 | 1724 | 1510 | 214 | 75.50% |
| Ipsweep | 500 | 396 | 339 | 57 | 67.80% |
| Neptune | 1000 | 826 | 735 | 91 | 73.50% |
| Total | 5500 | 4622 | 4082 | 540 | 74.22% |

From the Table5 and Table6 we can get the different between traditional BP and the UDP-BP. We can get the result from Table7.

Table7. Comparison of detecting result of traditional BP and UDP neural network

| Algorithm | False rate | Detection rate | Missing report rate |
|---|---|---|---|
| Tradition BP | 8.90% | 84.03% | 10.34% |
| UDP BP | 4.45% | 97.84% | 3.94% |

Among them

Detection rate = (number of be detected samples)/ (total number of samples);

Missing report rate = (number of testing for normal abnormal data sample)/ (Total number of samples of normal data);

False positive rate = (number of testing for normal data sample)/ (Total number of samples of normal data).

## Summary

In this paper, we made two improvements. One is that take principal components analysis method to deal with the original. The other is combine the particle swarm optimization algorithm with BP neural network. The result from simulation indicates that the false positive rate and missing rate of TCP or UDP has an obviously reducing. Thus, the PCA-PSO-BP neural network is worth popularization and application.

## References

[1] JinMing S, ZhuTing Y: Weighted D-S evidence theory and conjugate gradient descent diesel engine fault diagnosis based on BP neural network, Vol. 16 (2013), p. 111-112.

[2] Masud U, Baig M: An analysis of Newton's method in wireless systems using Gabor frames. Multitopic Conference (INMIC) 2012 15th International. IEEE[C], 2012, p.132-135.

[3] Zhao Y, Juang B H: Nonlinear Compensation Using the Gauss–Newton Method for Noise-Robust Speech Recognition, IEEE Transactions on Audio, Speech, and Language Processing, Vol. 20(2012), p. 2191-2206.

[4] Yang T, Qun Z, Fanlin Z: Data fitting method on hypersonic vehicle aerodynamic odeling, 31st Chinese Control Conference (2012), p. 7031-7041.

[5] David E. Rumelhart, James L. McClelland: *Rulmhart and parallel distributed processing*, edtied by Massachusetts institute of technology (1986).

[6] LI Xiang-fei, ZOU En, ZOU Li-hua: Chaos BP hybrid learning algorithm for feedforward neural network, Vol.19(2004), Vol. 4, p. 462-464.

[7] Jianhui Lv, Xingwei Wang, Min Huang: Research on Routing Algorithm Based on Limitation Arrangement Principle in Mathematics, mathematical problems in engineering, Vol. 2014, p. 1-11.

[8] Sha D Y, Hsu C Y: A hybrid particle swarm optimization for job shop scheduling problem, Computers & Industrial Engineering, Vol. 51(2006), NO.4, p. 791-808.