A New Stochastic PSO Technique for Neural Network Training

Yangmin Li and Xin Chen

Department of Electromechanical Engineering,
Faculty of Science and Technology,
University of Macau, Av. Padre Tomás Pereira S.J., Taipa, Macao SAR, P.R. China
ymli@umac.mo, ya27407@umac.mo

Abstract. Recently, Particle Swarm Optimization(PSO) has been widely applied for training neural network. To improve the performance of PSO for high-dimensional solution space which always occurs in training NN, this paper introduces a new paradigm of particle swarm optimization named stochastic PSO (S-PSO). The feature of the S-PSO is its high ability for exploration. Consequently, when swarm size is relatively small, S-PSO performs much better than traditional PSO in training of NN. Hence if S-PSO is used to realize training of NN, computational cost of training can be reduced significantly.

1 Introduction

As an attempt to model the processing power of human brain, artificial neural network is viewed as an universal approximation for any non-linear function. Up to now many algorithms for training neural network have been developed, back-propagation (BP) method is a popular one. Instead of BP, this paper introduces a new particle swarm optimization (PSO) for training of NN. Since PSO was firstly developed in 1995 [1], it has been an increasingly hot topic in community of artificial intelligence. Due to PSO's advantages in terms of simple structure and easy implementation in practice, PSO is widely used in many fields which involve optimization algorithms[2] [3][4].

Up to now there are more and more literatures referring to training of neural network with PSO. Normally it is accepted that PSO-NN has the following advantages:

1) There are no restricts for the PSO method which is critical for BP that the transfer function in hidden layer should be differentiable. So more transfer functions can be selected to fulfill different requirements.
2) Comparing with BP, PSO training algorithm is not easy to be trapped into local minima.

But as a stochastic method, PSO method suffers from the "curse of dimensionality", which implies that its performance deteriorates as the dimensionality of the search space increases. To overcome this problem, a cooperative approach is proposed in which the solution space is divided into several subspaces with lower dimension, and several swarms in these subspaces cooperate with each other to find the global best solution [5]. But such method induces a problem named stagnation. Since the performance of PSO is determined by exploration and exploitation of particles, it is reasonable that improving exploration ability can make particles explore solution space more efficiently in order to improve PSO performance. In this paper, we propose a stochastic PSO(S-PSO) to accomplish training NN which fulfills requirements with relative small size but high efficiency.

2 Stochastic PSO with High Exploration Ability

Owing to many literatures involving NN training using PSO, we just introduce the main idea of PSO training briefly. Given a multi-layer neural network, all its weights are combined together to form a vector which is viewed as a solution in a solution space. Then a swarm is proposed whose members (particles) represent such solution candidates. According to a certain criterion, which is normally in the form of minimal mean square error between patterns and outputs of NN, all particles congregate to a position on which the coordinate represents the best solution they found. Therefore the dimension of solution space is the same as the number of weights of NN. Take a three-layer fully connected feed-forward network as an example. If there are five neurons in the hidden layer and bias in hidden and output layers, even the NN has one input and one output, there are sixteen weights to be trained. Now let's estimate the swarm size sufficient for training.

Consider the traditional PSO updating principle with constriction coefficient K expressed as follows:

$$\begin{aligned} &v_i(n+1) = K\left[v_i(n) + c_{i1}r_{i1}(n)(P_i^d(n) - X_i(n)) + c_{i2}r_{i2}(n)(P_i^g(n) - X_i(n))\right] \\ &X_i(n+1) = X_i(n) + v_i(n+1), \end{aligned} \tag{1}$$

where $X_i = [\, x_{i1} \quad x_{i2} \quad \cdots \quad x_{iD} \,]$ denotes current position; v_i denotes current velocity; c_1 and c_2 represent the acceleration coefficients; $P_i^d(n)$ represents the best position found by particle i so far, $P_i^g(n)$ represents the global best position found by particle i's neighborhood. Obviously the random exploration ability is constricted within a subspace spanned by $\{P_i^d(n) - X_i(n), P_i^g(n) - X_i(n)\}$. That means the direction of exploration is restricted. At the same time, the intension of exploration behavior is totally determined by rate of decreasing of $P_i^d(n) - X_i(n)$ and $P_i^g(n) - X_i(n)$. To maintain exploration ability, there always need many particles within a swarm so that swarm size is several times, even ten times as dimension of solution space. For a simple SISO NN with a five neurons hidden layer, swarm size should be as many as sixteen. Hence the crucial shortage of PSO-NN is that the swarm size is so large that computational burden is unacceptable.

Since the relatively low exploration ability is induced by constraints of direction and intension of relative distance between particles' current positions and the best solutions founded by them and their neighborhood, we try to introduce a random exploration velocity into updating principle which is independent with positions. Based on explicit representation (ER) of PSO [6], we propose a new stochastic PSO (S-PSO) with the following definition.

Definition of Stochastic PSO. A stochastic PSO (S-PSO) is described as follows: Given a swarm including M particles, the position of particle i is defined as $X_i = [\, x_{i1} \quad x_{i2} \quad \cdots \quad x_{iD} \,]^T$, where D represents the dimension of swarm space. The updating principle for individual particle is defined as

$$\begin{aligned} v_i(n+1) &= \varepsilon(n) \left[v_i(n) + c_{i1} r_{i1}(n)(P_i^d(n) - X_i(n)) \right. \\ &\qquad \left. + c_{i2} r_{i2}(n)(P_i^g(n) - X_i(n)) + \xi_i(n) \right] \\ X_i(n+1) &= \alpha X_i(n) + v_i(n+1) + \tfrac{1-\alpha}{\phi_i(n)}(c_{i1} r_{i1}(n) P_i^d(n) + c_{i2} r_{i2}(n) P_i^g(n)), \end{aligned} \tag{2}$$

where c_1 and c_2 are positive constants; $P_i^d(n)$ represents the best solution found by particle i so far; $P_i^g(n)$ represents the best position found by particle i's neighborhood; $\phi_i(n) = \phi_{i1}(n) + \phi_{i2}(n)$, where $\phi_{i1}(n) = c_{i1} r_{i1}(n)$, $\phi_{i2}(n) = c_{i2} r_{i2}(n)$.

Applying theory of stochastic approximation, we can prove that if the following assumptions hold,

1) $\xi_i(n)$ is a random velocity with continuous uniform distribution. It has constant expectation denoted by $\Xi_i = \mathbf{E}\xi_i(n)$,
2) $\varepsilon(n) \to 0$ with n increasing, and $\Sigma_{n=0}^{\infty} \varepsilon_n = \infty$,
3) $0 < \alpha < 1$,
4) $r_{i1}(n)$ and $r_{i2}(n)$ are independent variables satisfying continuous uniform distribution in $[0, 1]$, whose expectations are 0.5,
then the updating principle must converge with probability one. Let $P^* = \inf_{\lambda \in (\mathbf{R}^D)} F(\lambda)$ represent the unique optimal position in solution space. Then swarm must converge to P^* if $\lim_n P_i^d(n) \to P^*$ and $\lim_n P_i^g(n) \to P^*$.

Due to limitation of pages, we just introduce the main idea of the proof. If define $Y(n) = X(n) - P^*$, $\theta(n) = [\, v(n) \quad Z(n) \,]^T = [\, v(n) \quad Y(n) - \mathbf{E}_n Q^r(n) \,]^T$, where $Q^r(n) = \frac{1}{\phi(n)}[\phi_1(n)(P^d(n) - P^*) + \phi_2(n)(P^g(n) - P^*)]$, updating principle can be expressed as a standard form of stochastic ODE as

$$\theta(n+1) = \theta(n) + \varepsilon(n) H(n). \tag{3}$$

Applying Lyapunov theory on stochastic process, a Lyapunov function is defined as

$$L(\theta(n)) = \tfrac{1}{2}\theta^T \begin{bmatrix} 1 & 0 \\ 0 & \Phi \end{bmatrix} \theta = \tfrac{1}{2}(v^2(n) + \Phi Z^2(n)). \tag{4}$$

After some calculations, we know for $n > N_k$, where N_k is a large enough integer, there is a positive non-decreasing function $k(\theta(n))$ such that

$$E_n L(\theta(n+1)) - L(\theta(n)) \le -k(\theta(n)) + \mathbf{E}\left[b_1(n)(Q^r(n) - E_n Q^r(n))^2 \right], \tag{5}$$

where $b_1(n) = \Phi(1 - \alpha + \varepsilon(n)\phi(n))^2 + (\varepsilon(n)\phi(n))^2$. Therefore $\theta(n)$ returns to the neighborhood of $\{\theta | k(\theta(t)) = \Phi(1 - \alpha)\mathbf{E}(Q^r(t) - \mathbf{E}_n Q^r(t))^2\}$ infinitely often as $n \to \infty$.

It can be proved that the following condition for asymptotic rate of change holds.

$$\lim_n \sup_{j \geq n} \max_{0 \leq t \leq T} \left| M^0(jT+t) - M^0(jT) \right| = 0, \tag{6}$$

where $M^0(t) = \sum_{i=0}^{m(t)-1} \varepsilon(i)\delta M(i)$, $\delta M(n) = H(n) - \mathbf{E}_n H(n)$. Hence as $n \to \infty$, the stochastic factor can not force $\theta(n)$ out of the vicinity of $\Phi(1-\alpha)\mathbf{E}(Q^r(t) - \mathbf{E}_n Q^r(t))^2\}$ infinitely often. That means $\theta(n)$ must converge to the set $\{\theta | k(\theta(t)) = \Phi(1-\alpha)\mathbf{E}(Q^r(t) - \mathbf{E}_n Q^r(t))^2\}$ with probability one. And if $lim_n Q^r(n) = 0$ or $P^d(n)$ and $P^g(n)$ converge to P^*, $\{\theta | k(\theta(t)) = \Phi(1-\alpha)\mathbf{E}(Q^r(t)$ $- \mathbf{E}_n Q^r(t))^2\}$ becomes $\{\theta | k(\theta(t)) = 0\}$, and the swarm must converge to P^*.

The following properties are obtained.

1) When n is less than N_k which makes equation (5) hold, the updating principle is nonconvergent so that particle will move away from the best position recorded by itself and its neighborhood. This phenomenon can be viewed as a strong exploration that all particles are exploring in the solution space. And when $n > N_k$, particle starts to converge.
2) An additional random velocity $\xi(n)$ independent with particle's position is very useful to maintain intension of exploration. That means when particles congregate too fast, particles can maintain certain exploration behavior to avoid being trapped into local minimum.

These two properties imply that S-PSO has strong exploration ability than traditional PSO, so that using S-PSO, we can accomplish training of NN with relative small swarm size.

3 Neural Network Training with S-PSO

To test the feasibility of S-PSO in training, we propose a test on non-linear artificial function approximation. As a comparison, other two ways of training, BP and traditional PSO are chosen. Since the test is to investigate dynamics of learning for the three algorithms, there is no need to build a complex neural network. So a standard one input, one output, three layers feed-forward neural network is chosen, whose hidden layer includes five neurons, just like Fig. 1 shows. There are sixteen weights to be optimized. In order to use BP, a differentiable sigmoid function and a linear function are chosen as transfer function in hidden layers and output layer. We use the way of batch training, that means in one epoch (iteration) the weights are updated once after all data are input to the net.

The parameters used in PSO training methods are arranged as follows. For S-PSO, $c_1 = c_2 = 3.5$, $\alpha = 0.95$, $\varepsilon(n)$ is of the form $\varepsilon(n) = 3.5/(n+1)^{0.4}$. For traditional PSO, $c_1 = c_2 = 2.05$, $K = 0.729$.

To speed up convergence of BP, a gradient descent with momentum weight is applied as BP learning, in which the momentum constant is set as 0.9. The NN toolbox in MATLAB 6.5 is used to build up such BP neural network.

Each weight in three NNs is initialized within $[0 \quad 1]$. A data set including 40 data is presented to NNs for training, in which data are sampled from a smooth

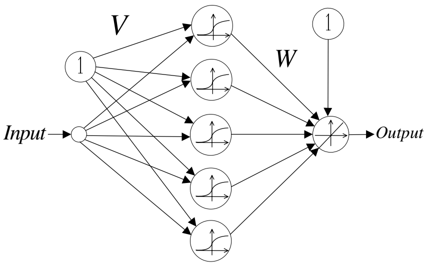
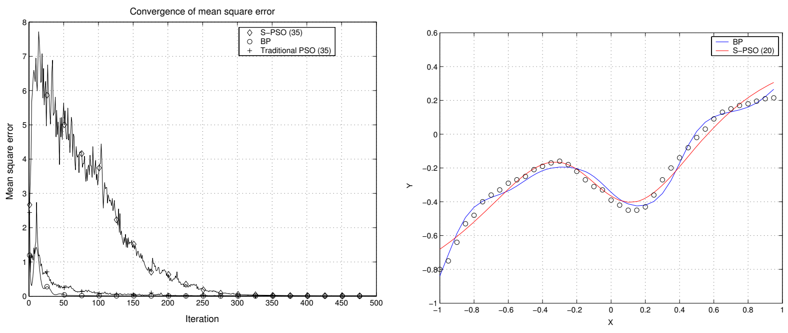


Fig. 1. Feed-forward neural network

Table 1. Comparative Results

Algorithm (swarm size)	Average of MSE	Minimum of MSE	Maximum of MSE	Std. Dev.
S-PSO (20)	0.6636×10^{-2}	0.9527×10^{-3}	0.02109	0.1078×10^{-2}
S-PSO (35)	0.5465×10^{-2}	0.1155×10^{-2}	0.01078	0.1040×10^{-2}
Traditional PSO (20)	0.01067	0.1408×10^{-2}	0.02151	$0.1187\times^{-2}$
Traditional PSO (35)	0.3648×10^{-2}	$0.6556\times^{-3}$	0.01056	0.5102×10^{-3}
Backpropagation	0.2367×10^{-2}	0.7082×10^{-3}	0.5743×10^{-2}	$0.2468\times^{-3}$



(a) (b)

Fig. 2. Evolution processes for three-path planning and results of function approximation

continuous curve with little disturbances. Each training algorithm is tested 30 runs. And each run includes 1500 iterations. To investigate performance of PSO under different swarm sizes, two swarm sizes are tested. The one is 20 which is a little bit greater than the dimension of solution space, the other is 35 which is more than twice as the dimension. All test result is listed in Tab. 1.

We observe that S-PSO training with a swarm including 20 particles performs much better than traditional PSO. We think the random velocity $\xi(n)$ brings obvious improvement on the performance. But when swarm size increases to 35 which is more than twice as solution dimension, the performance of traditional

PSO catches up with S-PSO, because larger size make traditional PSO explore better and originally traditional PSO converges faster than S-PSO. Such fast convergence can be observed in Fig. 2 (a) in which traditional PSO converges as fast as BP, while due to divergence property mentioned previously, S-PSO converges much slower than the other two. The result of function approximation in one of runs is shown in Fig. 2 (b), in which the black circles denote the data presented to NN, and the red line and blue line denote outputs of neural networks after training using S-PSO and BP respectively.

4 Conclusion

PSO is considered as an important evolutionary technique for optimization, which becomes more and more popular in training neural network due to its advantages mentioned above. But since the dimension of solution space is the same as the number of weights in NN, normally the swarm size is so large that computational cost becomes a large burden. This paper proposes a new stochastic PSO (S-PSO) which has an advantage of including an independent random velocity $\xi(n)$ to improve the exploration ability of swarm, so that S-PSO with relative small swarm size can accomplish training of NN. Hence applying such S-PSO, the computational cost of PSO training can be reduced significantly, meanwhile the advantages of PSO training are maintained as well.

Acknowledgement

This work was supported by the Research Committee of University of Macau under Grant no. RG082/04-05S/LYM/FST.

References

1. Kennedy, J., Eberhart, R. C.: Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Network, Perth, Australia (1995) 1942-1948
2. Juang, C. F.: A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design. IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics 34(2) (2004) 997 - 1006
3. Li, Y. and Chen, X.: Mobile Robot Navigation Using Particle Swarm Optimization and Adaptive NN, Advances in Natural Computation, Eds by L. Wang, K. Chen and Y.S. Ong, Springer, LNCS 3612 (2005), 628-631.
4. Messerschmidt, L., Engelbrecht, A. P.: Learning to Play Games Using a PSO-Based Competitive Learning Approach. IEEE Transactions on Evolutionary Computation 8(3) (2004) 280 - 288
5. Bergh, F., Engelbrecht, A. P.: A Cooperative Approach to Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation 8(3) (2004) 225 -239
6. Clerc, M., Kennedy, J.: The Particle Swarm: Explosion, Stability, and Convergence in a Multi-Dimentional Complex Space. IEEE Transactions on Evolutionary Computation 6(1) (2002) 58-73