Training Simultaneous Recurrent Neural Networks Using the PSO-QI Algorithm to Learn MIMO Systems

Bipul Luitel, Ganesh Kumar Venayagamoorthy*

Real-Time Power and Intelligent Systems Laboratory, Missouri University of Science Technology, Rolla, MO-65409, USA

Abstract

Training a single simultaneous recurrent neural network (SRN) to learn all outputs of a multiple-input-multiple-output (MIMO) system is a difficult problem. A new training algorithm developed from combined concepts of swarm intelligence and quantum principles is presented. The training algorithm is called particle swarm optimization with quantum infusion (PSO-QI). To improve the effectiveness of learning, a two-step learning approach is introduced in the training. The objective of the learning in the first step is to find the optimal set of weights in the SRN considering all output errors. In the second step, the objective is to maximize the learning of each output dynamics by fine tuning the respective SRN output weights. To demonstrate the effectiveness of the PSO-QI training algorithm and the two-step learning approach, two examples of an SRN learning MIMO systems are presented. The first example is learning a benchmark MIMO system and the second one is the design of a wide area monitoring system for a multimachine power system. From the results, it is observed that SRNs can effectively learn MIMO systems when trained using the PSO-QI algorithm and the two-step learning approach.

Key words: MIMO systems, PSO, Power system, Quantum principles, SRN, Two-step learning, Wide area monitor

1. Introduction

Simultaneous recurrent neural networks (SRNs) are a class of neural network architectures where the recurrence is instantaneous (Geib and Serpen, 2004). SRN is appropriate for approximating complex nonlinear systems with less number of neurons because it models the response of a dynamic nonlinear system even with fixed weights. Furthermore, SRNs have the capability of approximating non-smooth functions which cannot be approximated by conventional Multilayer Perceptrons (MLPs). An Elman SRN has its feedback from the hidden layer output to the context layer inputs and in this study it is represented in vector notation as:

$$H(k,n) = f(A * I(k) + B * H(k,n-1) + K) \qquad (1)$$

$$O(k) = g(C * H(k,R) + K') \quad n = 1,2,\ldots,R \qquad (2)$$

where, I is the set of inputs, H is the set of neuron outputs from the hidden layer and O is the set of outputs from the output layer. A is the set of weights from input layer to the hidden layer, B is the set of weights from context layer to the hidden layer, C is the set of weights from hidden layer to the output layer, n is the index of internal recurrence, k is the index of the input sample, R is the number of internal recurrences, K and K' are the biases and f and g are neuron activation functions in the hidden and output layers respectively.

Another important feature of recurrent neural networks is their ability to implement associative memory (Michel and Farrell, 1990). Use of various architectures of neural networks have been studied for associative memory (Kwan, 2002). Unlike other associative memories that store patterns, this work demonstrates SRNs that store dynamics that is stimulated by a stimulus pattern that has the same dynamics. Since SRNs have connectivities from its hidden or output layer to the input, this architecture helps them store information and hence act as associative memory. Although SRNs are powerful neural network architectures, the training process is intensive and more difficult when there are multiple outputs to learn i.e. learning of multiple-input-multiple-output (MIMO) system. Traditional training algorithms such as back propagation through time suffer from local minima and hence it is hard to train SRNs using these techniques because of the recursive calculations involved (Cai et al., 2007). Computational intelligence (CI) based algorithms have gained popularity in training of neural networks because of their ability to find a global solution in a multi-dimensional search space. Swarm and evolutionary based algorithms such as Particle Swarm Optimization (PSO) (Del Valle et al., 2008) have shown promises in training of SRNs. In this study, quantum principle obtained from Quantum PSO (QPSO) (Sun et al., 2004) has been combined with

*Corresponding Author
Email addresses: iambipul@ieee.org (Bipul Luitel), gkumar@ieee.org (Ganesh Kumar Venayagamoorthy)

standard PSO to form a new hybrid algorithm called as PSO with Quantum Infusion (PSO-QI). For training, a two step learning approach is introduced to improve the ability of SRNs to learn multiple outputs.

2. PSO-QI Algorithm and Two-Step Learning

2.1. Particle Swarm Optimization with Quantum Infusion

PSO-QI is a hybrid algorithm that uses the quantum principle from QPSO to create a new offspring in PSO. After the positions and velocities of the particles are updated using standard PSO equations, a randomly chosen particle from PSO's *pbest* (the previous particle position giving the best fitness value) population is utilized to carry out the quantum operation; and thus, create an offspring by mutating the *gbest* (the best particle among all the particles in the swarm). The fitness of the offspring is evaluated and the offspring replaces the *gbest* only if it has a better fitness. This ensures that the fitness of the *gbest* is equal to or better than its fitness in the previous iteration. Thus, it is improved and pulled toward the best solution over iterations.

According to the uncertainty principle, position and velocity of a particle in quantum world cannot be determined simultaneously. Thus QPSO differs from standard PSO mainly in the fact that exact values of x and v cannot be determined. Hence the probability of finding a particle at a particular position in the quantum search space is mapped into its actual position in the solution space by a technique called "collapsing". In Quantum Delta-Potential-Well based PSO (QDPSO) (Sun et al., 2004), a delta potential well based probability density function is used to avoid explosion and help the particles converge. By using Monte Carlo Simulation (Sun et al., 2004), the position equation in QDPSO is given by (3):

$$x(k) = J(k) \pm \frac{L(k)}{2} ln(1/u) \tag{3}$$

where u is a uniform random number in the interval [0,1]. The particle's local attractor point J has coordinates given by the following equation:

$$J_d(k) = \alpha_1 P_{gd}(k) + \alpha_2 P_{id}(k) \tag{4}$$

where P_{id} is the i^{th} *pbest* particle in d^{th} dimension and P_{gd} is d^{th} dimension of the *gbest* particle obtained from PSO. L is the length of the potential field given by:

$$L(k) = 2\beta|J(k) - x(k)| \tag{5}$$

The parameter β is the only parameter of the algorithm. It is called the creativity coefficient and is responsible for the convergence speed of the particle.

The Mean Best Position, *mbest*, is defined as:

$$mbest(k) = \frac{1}{S}\sum_{i=0}^{S} P_i(k) = \left(\frac{1}{S}\sum_{i=0}^{S} P_{i1}(k), \ldots, \frac{1}{S}\sum_{i=0}^{S} P_{iD}(k)\right) \tag{6}$$

where S is the size of the population, D is the number of dimensions and P_i is the *pbest* position of each particle. In QPSO, J in (5) is replaced by *mbest* to form (7) as follows:

$$L(k) = 2\beta|mbest(k) - x(k)| \tag{7}$$

By using (4) this can also be written as follows to show the mutation on *gbest*, where the addition or subtraction is carried out with 50% probability:

$$x(k+1) = \alpha_1 P_{gd}(k) + \alpha_2 P_{id}(k) \pm \beta|mbest(k) - x(k)| ln(1/u) \tag{8}$$

In PSO-QI, the position update equation (8) has been used to mutate the *gbest* particle obtained from PSO. The pseudocode for the PSO-QI algorithm is as follows:

```
Initialize position $x$, velocity $v$ and let $pbest$=$x$
repeat
  for $i = 1$ to $populationsize$ do
    Evaluate fitness
    if fitness $(i)$ < fitness $(pbest)$ then
      $pbest = x$ and $gbest = \min(pbest)$
    end if
    Update $v$ and $x$ using standard PSO equations
  end for
  Calculate $mbest$ using (6)
  Select a random particle $r$
  for $d$ from 1 to $dimensionsize$ do
    $\alpha_1, \alpha_2 = rand(0,1)$
    $J = (\alpha_1 * P_{rd} + \alpha_2 * P_{gd})/(\alpha_1 + \alpha_2)$
    $L = 2\beta * |mbest - x_{rd}|$ using (7)
    if $rand(0,1) > 0.5$ then
      using (3)
      $offspring = J - \frac{L}{2} * ln(1/u)$
    else
      $offspring = J + \frac{L}{2} * ln(1/u)$
    end if
    if fitness $(offspring)$<fitness$(gbest)$ then
      $gbest = offspring$
    end if
  end for
until termination criteria is met.
```

2.2. Two-Step Learning

In this training approach, SRN is trained for all network weights in Step 1. After the first step, input weights are kept fixed and output weights are fine tuned to maximize the learning of the respective output dynamics. This means, for M outputs, M searches are carried out. This drastically reduces the dimension of the problem in Step 2, and hence lesser computations and a fast tuning process. For an Elman Network, the hidden node outputs may be initially computed and fixed in Step 2, thus further reducing the computations. Step 1 can be viewed as global exploration search and Step 2 a local exploitation search.

3. Examples

SRNs training two MIMO systems are illustrated in this letter. Mean squared error (MSE) between the actual and the predicted output is considered as the fitness function of a particle. For Step 1, it is given by (9):

$$MSE = \frac{1}{M}\sum_{m=1}^{M} MSE_m \tag{9}$$

where M is the number of outputs and MSE_m given by (10):

$$MSE_m = \frac{1}{N}\sum_{k=1}^{N} (Y_m(k) - \hat{Y}_m(k)) \tag{10}$$

is the fitness of Step 2, where $Y_i(k)$ is the actual output of the system and $\hat{Y}_i(k)$ is the predicted output from the SRN at sample k and N is the number of samples. The PSO parameters used in the study are: c_1, $c_2 = 2$, w is linearly decreasing from 0.9 to 0.4 and a population size of 30 particles. β parameter of PSO-QI is linearly increasing from 0.5 to 1. The results of PSO-QI are compared with that obtained using PSO. In both cases, the networks are trained for 200 iterations in Step 1 and 25 in Step 2.

3.1. Case I: Identification of MIMO System

The first case studied is a MIMO plant given by the following equation (Kumpati and Kannan, 1990):

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} \frac{y_1(k)}{1+y_2^2(k)} \\ \frac{y_1(k)y_2(k)}{1+y_2^2(k)} \end{bmatrix} + \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{11}$$

where $u_i(k)$ and $y_i(k)$ are the i^{th} input and output at instant k. The inputs to the SRN are the present values of system inputs and outputs. The SRN outputs are the one-step ahead prediction of the system outputs. An SRN of 4 inputs, 10 hidden nodes and 2 output nodes is then trained on 100 samples of uniform random numbers between [0,1] and tested on 100 data samples of input vector $[sin(2\pi k/25), cos(2\pi k/25)]$. The dimension of the SRN training problem is 160 and 10 in Steps 1 and 2 respectively. The testing results obtained are shown in Figs. 1 and 2. The MSEs are shown in Table 1.

3.2. Case II: Design of a Wide Area Monitoring System

In the second case study, a wide area monitor (WAM) used for the two area four machine power system described in (Venayagamoorthy, 2007) and shown in Fig. 4 is considered. The WAM is modeled by an SRN with 8 input nodes, 15 hidden nodes and 4 output nodes (405 weights). The inputs to the SRN are the current deviations in reference voltage $Vref$, caused by the pseudorandom binary signal excitations and speed deviations of the four machines. The outputs are the one-step ahead predictions of their speed deviations. The operating conditions are similar to (Venayagamoorthy, 2007). The loads are modified to have a power transfer of 510 MW during training and 458 MW during testing from Area 1 to 2. The voltage of all the generators is 1.03 pu in both operating conditions. The dimension of the SRN training problem is 405 and 15 in Step 1 and 2 repectively. The outputs of the SRNs for two generators (G1 and G4) are shown in Figs. 5 and 6. The fitness curve obtained for the two step process is shown in Fig. 3. The results show improvement from PSO to PSO-QI and from Step 1 to Step 2. The MSEs are shown in Table 1.

4. Conclusion

MIMO SRNs are trained using PSO-QI by using a two-step learning approach. The results of the study show that hard-to-train MIMO SRNs can be successfully trained with better accuracy. Due to the dimension reduction in the second step, this method of SRN training can be useful for large number of inputs and hence highly scalable. The network is trained for less iterations in the second step providing a significant improvement in fitness for a small overhead in time. Although implicit studies have not been carried out to see the associative memory property of SRNs, their ability to store system dynamics in a similar fashion has been demonstrated, which leads the authors to believe their ability to be used for associative memory in their future works.

Acknowledgment

The funding provided by the National Science Foundation, USA under the CAREER grant ECS #0348221 and EFRI #0836017 is gratefully acknowledged.

References

Cai, X., Prokhorov, D., Wunsch, D., 2007. Training Winner-Take-All Simultaneous Recurrent Neural Networks. IEEE Transactions on Neural Networks 18 (3), 674–684.

Del Valle, Y., Venayagamoorthy, G., Mohagheghi, S., Hernandez, J., Harley, R., 2008. Particle swarm optimization: Basic concepts, variants and applications in power systems. IEEE Transactions on Evolutionary Computation 12 (2), 171–195.

Geib, J., Serpen, G., 25–29 July 2004. Computational promise of simultaneous recurrent network with a stochastic search mechanism. In: Proc. IEEE International Joint Conference on Neural Networks. Vol. 3. pp. 2239–2244.

Kumpati, S., Kannan, P., 1990. Identification and control of dynamical systems using neural networks. IEEE Trans on Neural Networks 1 (1), 4–27.

Kwan, H., 2002. Multilayer recurrent neural networks [character recognition application example]. In: Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on. Vol. 3.

Michel, A., Farrell, J., 1990. Associative memories via artificial neural networks. IEEE Control Systems Magazine 10 (3), 6–17.

Sun, J., Feng, B., Xu, W., 19–23 June 2004. Particle swarm optimization with particles having quantum behavior. In: Proc. CEC2004 Evolutionary Computation Congress on. Vol. 1. pp. 325–331.

Venayagamoorthy, G., 2007. 2007 special issue: Online design of an echo state network based wide area monitor for a multimachine power system. Neural Networks 20 (3), 404–413.

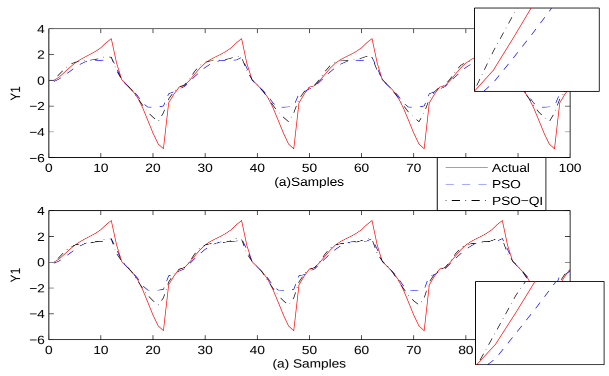

Figure 1: Output of SRN for Y1 in (a) Step 1 and (b) Step 2

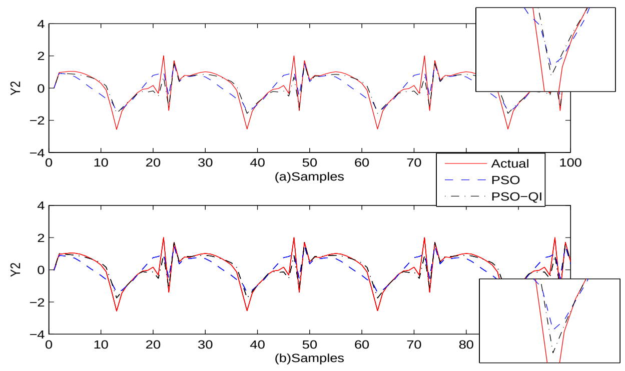

Figure 2: Output of SRN for Y2 in (a) Step 1 and (b) Step 2

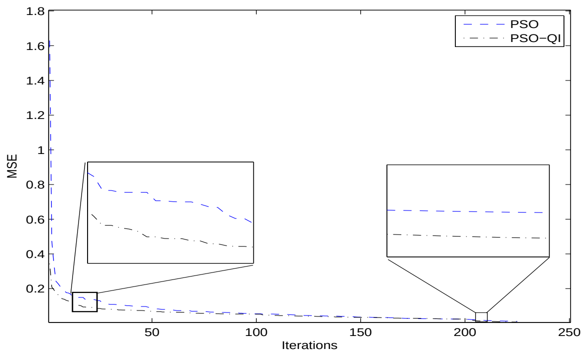

Figure 3: Comparison of fitness in Step 1 and 2

Table 1: MSE results for the case studies

		PSO		PSO-QI	
		Step 1	Step 2	Step 1	Step 2
Case I	Y1	1.2494	1.1487	0.7132	0.6407
	Y2	0.3150	0.2996	0.1427	0.1012
Case II	G1	0.0147	0.0075	0.0188	0.0057
	G2	0.0281	0.0111	0.0218	0.0068
	G3	0.0225	0.0118	0.0181	0.0067
	G4	0.0344	0.0167	0.0319	0.0105

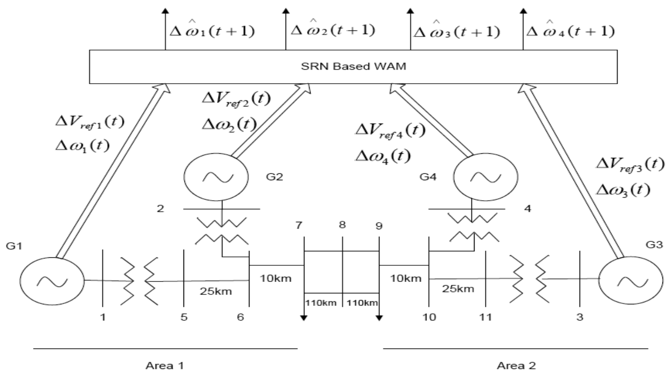

Figure 4: Wide area monitor in a two area four machine system

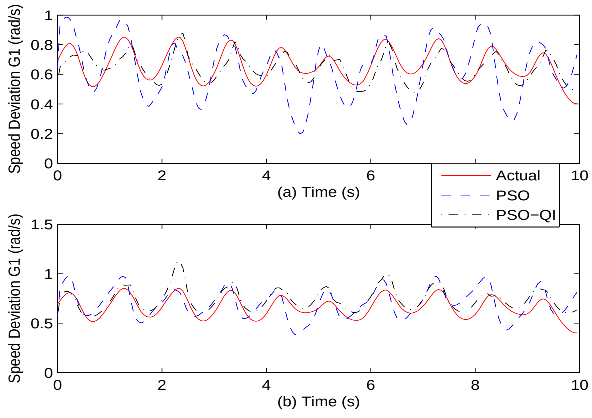

Figure 5: SRN output for G1 (Area 1) in (a) Step 1 and (b) Step 2

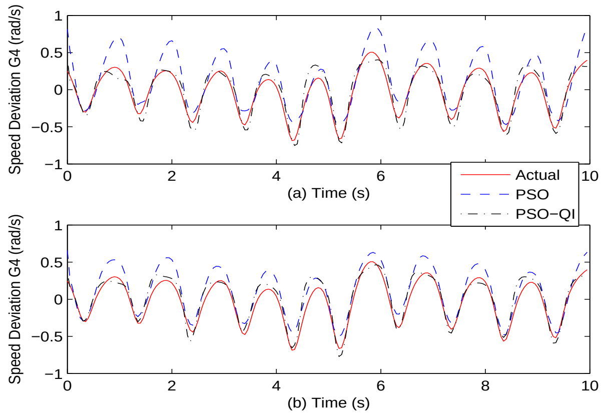

Figure 6: SRN output for G4 (Area 2) in (a) Step 1 and (b) Step 2