

# Comparison of Genetic Algorithm and Particle Swarm Optimizer When Evolving a Recurrent Neural Network

Matthew Settles<sup>1</sup>, Brandon Rodebaugh<sup>1</sup>, and Terence Soule<sup>1</sup>

Department of Computer Science, University of Idaho,  
Moscow, Idaho U.S.A

**Abstract.** This paper compares the performance of GAs and PSOs in evolving weights of a recurrent neural network. The algorithms are tested on multiple network topologies. Both algorithms produce successful networks. The GA is more successful evolving larger networks and the PSO is more successful on smaller networks.<sup>1</sup>

## 1 Background

In this paper we compare the performance of two population based algorithms, a genetic algorithm (GA) and particle swarm optimization (PSO), in training the weights of a strongly recurrent artificial neural network (RANN) for a number of different topologies. The goal is to develop a recurrent network that can reproduce the complex behaviors seen in biological neurons [1]. The combination of a strongly connected recurrent network and an output with a long period makes this a very difficult problem. Previous research in using evolutionary approaches to evolve RANNs have either evolved the topology and weights or used a hybrid algorithm that evolved the topology and used a local search or gradient descent search for the weights (see for example [2]).

## 2 Experiment and Results

Our goal is to evolve a network that produces a simple pulsed output when an activation ‘voltage’ is applied to the network’s input. The error is the sum of the absolute value of the difference between the desired output and the actual output at each time step plus a penalty (0.5) if the slope of the desired output differs in direction from the slope of the actual output.

The neural network is strongly connected with a single input node and a single output node. The nodes use a symmetric sigmoid activation function. The activation levels are calculated synchronically.

The GA uses a chromosomes consisting of real values. Each real value corresponds to the weight between one pair of nodes.

---

<sup>1</sup> This work supported by NSF EPSCoR EPS-0132626. The experiments were performed on a Beowulf cluster built with funds from NSF grant EPS-80935 and a generous hardware donation from Micron Technologies.

The GA is generational, 250 generations, 500 individuals per generation. The two best individuals are copied into the next generation (elitism). Tournament selection is used, with a tournament of size 3. The initial weights were randomly chosen in the range  $(-1.0, 1.0)$ . The mutation rate is  $1/(LN)^2$ . Mutation changes a weight by up to 25% of the weight's original value. Crossover is applied to two individuals at the same random (non-input) node. The crossover rate is 0.8.

The PSO uses position and velocity vectors which refer to the particles' position and velocity within the search space. They are real valued vectors, with one value for each network weight. The PSO is run for 250 generations on a population of 500 particles. The initial weights were randomly chosen in the range  $(-1.0, 1.0)$ . The position vector was allowed to explore values in the range of  $(-2.0, 2.0)$ . The inertial weight is reduced linearly from 0.9 to 0.4 each epoch [3].

Tables 1 and 2 show the number of successful trials out of the fifty. Successful trials evolve a network that produces periodic output with the desired frequency. Unsuccessful trials fail to produce periodic behavior.

Both the GA and PSO perform well for medium sized networks. The GAs optimal network size is around 3-4 layers with 5 nodes per layer. The PSOs optimal network is approximately 2x5. The GA is more successful with larger networks, whereas the PSO is more successful with smaller networks. A two-tailed z-test ( $\alpha$  of 0.05) confirms that these differences are statistically significant.

**Table 1.** Number of successful trials (out of fifty) trained using GA.

Layers	1	2	3	4
1 Node/Layer	0	0	0	0
3 Nodes/Layer	0	17	44	49
5 Nodes/Layer	5	41	50	50
7 Nodes/Layer	22	48	46	41
9 Nodes/Layer	36	49	40	—

**Table 2.** Number of successful trials (out of fifty) trained using PSO.

Layers	1	2	3	4
1 Node/Layer	0	4	23	38
3 Nodes/Layer	17	43	49	47
5 Nodes/Layer	39	50	40	32
7 Nodes/Layer	46	46	36	19
9 Nodes/Layer	49	41	17	—

### 3 Conclusions and Future Work

In this paper we demonstrated that GA and PSO can be used to evolve the weights of strongly recurrent networks to produce long period, pulsed output signals from a constant valued input. Our results also show that both approaches are effective for a variety of different network topologies. Future work will include evolving a single network that can produce a variety of biologically relevant behaviors depending on the input signals.

### References

1. Shepherd, G.M.: Neurobiology. Oxford University Press, New York, NY (1994)
2. Angeline, P.J., Saunders, G.M., Pollack, J.P.: An evolutionary algorithm that constructs recurrent neural networks. IEEE Transactions on Neural Networks 5 (1994) 54–65
3. Kennedy, J., Eberhart, R.: Swarm Intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, CA (2001)