

The Enhancement of Evolving Spiking Neural Network with Dynamic Population Particle Swarm Optimization

Nur Nadiyah Md. Said, Haza Nuzly Abdull Hamed^(✉),
and Afnizanfaizal Abdullah

Faculty of Computing, Universiti Teknologi Malaysia,
Johor Bahru, Johor, Malaysia
nnadiyah35@live.utm.my, {haza,afnizanfaizal}@utm.my

Abstract. This study presents an integration of Evolving Spiking Neural - Network (ESNN) with Dynamic Population Particle Swarm Optimization (DPPSO). The original ESNN framework does not automatically modulate its parameters' optimum values. Thus, an integrated framework is proposed to optimize ESNN parameters namely, the modulation factor (*mod*), similarity factor (*sim*), and threshold factor (*c*). DPPSO improves the original PSO technique by implementing a dynamic particle population. Performance analysis is measured on classification accuracy in comparison with the existing methods. Five datasets retrieved from UCI machine learning are selected to simulate the classification problem. The proposed framework improves ESNN performance in regulating its parameters' optimum values.

Keywords: Evolving · Spiking · Neural network · Dynamic · Population · Particle swarm · Optimization · Parameter

1 Introduction

Neural network inspired by the human brain is gaining popularity nowadays due to its capability in solving various problems. Artificial neural network (ANN) is a group of processing components in a collective network resembles the features of a biological neural network [1]. The characteristic of an early neural network is restricted to a single layer, which inspires researchers in improving the neural network architecture. It leads to the creation of a new generation of neural network [2]. Spiking neural network is the third generation of a neural network model. Evolving Spiking Neural Network (ESNN) is one of the well-known SNN categories that evolves from the latest spiking neurons [3].

However, ESNN is dependent on parameter tuning. Thus, an optimizer is needed to help automate the process of determining the ESNN's parameters combination [4]. Few studies integrate ESNN with several optimizers [4, 5]. However, some of the optimizers known today require a large computational cost which has driven researchers to establish more effective methods.

Particle Swarm Optimization (PSO) introduced by Kennedy and Eberheart [6, 7] based on swarm population is inspired by the nature of birds' flocking behavior. PSO

has the ability to solve various issues of optimization [8–11]. However, despite the recognition received by PSO, this optimizer also has drawbacks. This study proposes a dynamic population concept in PSO known as Dynamic Population Particle Swarm Optimization (DPPSO), which will be implemented as an ESNN parameter optimizer. ESNN-DPPSO improve classification accuracy compared to the other ESNN integrated techniques.

2 Evolving Spiking Neural Network

Evolving spiking neural network (ESNN) was originally proposed as a visual pattern identification system. The earliest ESNN was derived from Thorpe’s neuron model whereby the earlier spikes represent the significance of the output generated [12, 13].

ESNN evolves its form whereby for each input pattern, a new neuron is created and linked to the connectivity of neurons [12]. ESNN classifier determines the mapping by computing the distance between neurons to a particular class marker. Hence, it is appropriate to be utilized for time-invariant information categorization [14].

A single input value is encoded to multiple neurons by calculating the intersection of a Gaussian function [5] using the following mathematical equations.

$$\mu = lmin + ((2 * i - 3)/2) * ((lmax - lmin)/(M - 2)) \quad (1)$$

$$\sigma = (1/(\beta(lmax - lmin)))/M - 2 \text{ where } 1 \leq \beta \leq 2 \quad (2)$$

where:

μ = The center of a Gaussian intersection

σ = The width of variable interval [$lmin$, $lmax$]

There are three parameters for ESNN consisting of modulation factor (mod), similarity factor (sim), and threshold (c). The mod shows how the sequence of temporal spike’s arrival time affects the neuron, whereas sim regulates the neuron distance and cluster in the output and the c determines when a neuron should produce an output spike. Details on ESNN can be found in [13].

3 Dynamic Population Particle Swarm Optimization

Although PSO has been widely used, there are a few drawbacks encompassing being trapped in a local optima and premature convergence [15]. This has influenced researchers to further improve the algorithm [16–19].

The proposed dynamic population PSO is implemented with a dynamic number of particles instead of a fixed population. The basic PSO element is retained to leverage its ability on fast convergence [15]. Algorithm 1 explains the details of DPPSO.

Algorithm 1: DPPSO

```

initialize DPPSO parameters
calculate reduction cycle
initialize particle with random value
initialize  $P_{best}$  and  $G_{best}$  with the lowest value
while not reaching the maximum iteration do
    get particle fitness
    if particle fitness >  $P_{best}$ 
         $P_{best}$  = particle fitness
    end if
    if  $P_{best}$  >  $G_{best}$ 
         $G_{best}$  =  $P_{best}$ 
    end if
    if current iteration == reduction cycle value
        remove lowest fitness
    end if
    update particle based on PSO equation
end while

```

The DPPSO has four parameters; inertia weight (ω), velocity towards global best (C_1), velocity towards particle best (C_2) and reduction cycle. First, the parameters are initialized. Next, the reduction is calculated. The particle is initialized with a random value. Particle best (P_{best}) is the best solution obtained by the particle itself while the global best (G_{best}) is measured by the best solution achieved globally. P_{best} and G_{best} are initialized each with the lowest value. If the particle fitness value is greater than P_{best} , the P_{best} value is updated. If the P_{best} value is greater than G_{best} , the G_{best} value is updated. If the current iteration value is equal to the reduction cycle value, particle with the lowest fitness value is removed. The particle position value is updated using the PSO rule in Eq. 3.

$$v_p(i,j) = \omega v_p(i,j) + C_1(P_{best}(i,j) - x(i,j)) + C_2(G_{best} - x(i,j)) \quad (3)$$

4 Integration of ESNN-DPPSO

The integrated ESNN-DPPSO is described in Algorithm 2. The algorithm begins with the initialization of particles and its fitness at the lowest value which is -9999 . The iteration is set to 2000. The fitness is updated throughout the iteration. The reduction cycle (reductionCycle) removes a particle with the lowest fitness value and is set according to Eq. 4.

$$\text{Reduction cycle} = \text{Iteration} / (\text{reductionCycle} + 1) \quad (4)$$

If the current fitness value is better than the P_{best} value, the P_{best} is updated. The ESNN parameters are updated by training the fitness value of each particle with DPPSO. PSO rule is implemented to update the position of particles.

Algorithm 2: ESNN-DPPSO

```

for all particles do
    initialize all DPPSO parameters
    initialize fitness
end for
while not reaching maximum iteration do
    for all particles do
        calculate reduction cycle
        get fitness from ESNN
        if current fitness > PBest fitness then
            PBest = current particle
        end if
        for all ESNN parameters do
            update parameter
        end for
        if iteration == reduction cycle
            remove lowest particle
        end if
    end for
end while

```

The proposed integrated framework is illustrated in Fig. 1.

4.1 Experiment Setup

In this study, five standard datasets retrieved from UCI machine learning repository [20] are selected to simulate the classification problems and are implemented to the proposed framework. The datasets are Iris, heart, breast cancer, Pima Indian diabetes, and wine datasets. The datasets are normalized using Eq. 5.

$$X = (X_i - X_{min}) / (X_{min} - X_{max}) \quad (5)$$

where,

- X = initial value of variable
- X_i = variable value after normalization
- X_{max} = variable maximum value
- X_{min} = variable minimum value

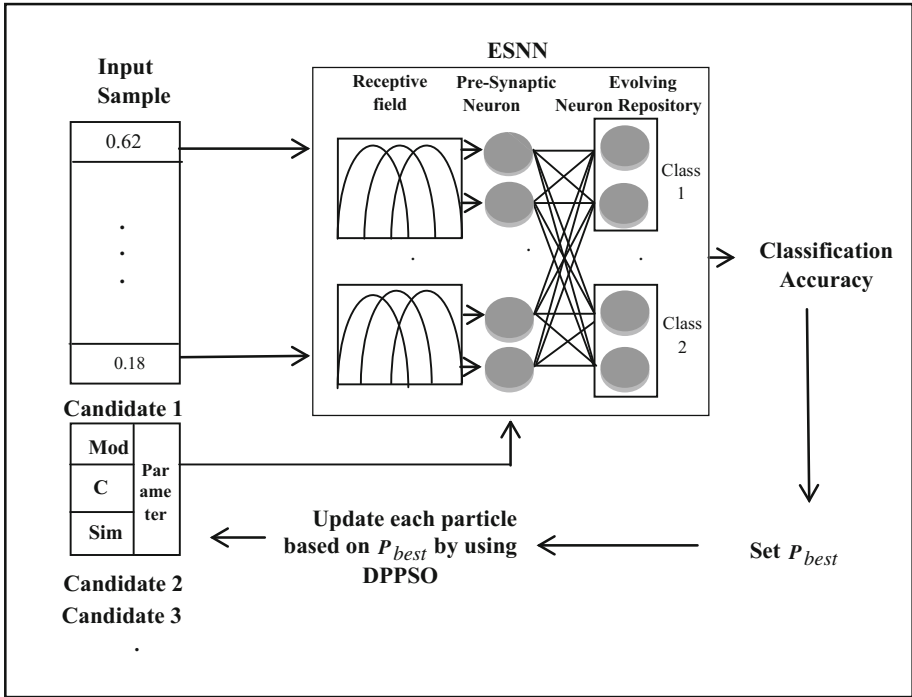


Fig. 1. ESNN-DPPSO framework

In addition, DPPSO has its own parameters. Table 1 shows the respective value of each selected DPPSO parameter.

Table 1. DPPSO parameter

Parameter	Value
Inertia weight	2.0
Velocity global best	2.0
Velocity best particle	2.0
Reduction cycle	4

5 Results and Discussion

The datasets are separated into two categories, training and testing. A 10-fold cross validation technique is used for these datasets. 90% of the datasets are used for training to avoid overfitting while the remaining datasets are used for testing. Overfitting can occur if the size of training is small. Next, each dataset is normalized followed by the training phase for classification. The training process for the integrated ESNN-DPPSO is shown in Algorithm 2. Three ESNN parameters; *mod*, *sim* and *c* are set in the range

of zero to one. These values are important as they influence the classification result. According to [21], a standard ESNN parameter value for parameter *mod* is in the range of 0.9 to 1.0. Parameter *c* is set in between 0.55 to 0.85. Parameter *sim* is set within the range of 0.3 to 0.6. Table 2 shows the result of an early experiment where the ESNN parameters are set manually.

Table 2. Classification accuracy result based on manual tuning of ESNN

Dataset	Manually set parameter value			Classification accuracy %
	Mod	C	Sim	
Iris	0.90	0.90	0.10	92.00
Diabetes	0.90	0.90	0.10	56.41
Wine	0.90	0.95	0.10	66.67
Breast cancer	0.90	0.35	0.10	94.41
Heart	0.85	0.75	0.10	74.44

Table 3 shows the classification accuracy using the ESNN-DPPSO method. The value of each parameter shown is the optimum value found by utilizing the optimizer DPPSO. It is known that there is no specific combination of parameters' values across all five datasets. The combination of ESNN parameters generated by ESNN-DPPSO produces a better classification result than the manually set values of the standard ESNN parameters.

Table 3. Classification accuracy result using ESNN-DPPSO

Dataset	Manually set parameter value			Classification accuracy %
	Mod	C	Sim	
Iris	0.85	0.70	0.52	92.00
Heart	0.98	0.97	0.45	70.37
Diabetes	0.74	0.61	0.53	75.23
Breast cancer	0.99	0.87	0.54	95.48
Wine	0.98	0.91	0.58	85.40

In ESNN-DPPSO, the best *mod* value found is within the range of 0.7 to 1.0. This value should not be low as it depicts the ESNN connection weight. On the other hand, parameter *c* influences the PSP threshold value within the range of 0.6 to 1.0. Meanwhile, the best *sim* value is within the range of 0.4 to 0.5 as it controls the evolving neurons where a lower value shows less neurons in the same range are being merged.

Thus, ESNN-DPPSO assists the search of best ESNN parameters' values combination automatically and reduces computational time. Next, a comparison study is

Table 4. Comparison between ESNN-DPPSO classification result with other methods

Dataset	Algorithm	Classification accuracy %
Iris	Standard ESNN parameter	89.33
	ESNN Modified parameter	92.00
	ESNN-DPPSO	92.00
	DE-ESNN	93.33
Breast cancer	Standard ESNN parameter	66.18
	ESNN Modified parameter	94.41
	ESNN-DPPSO	95.48
	DE-ESNN	91.18
Diabetes	Standard ESNN parameter	38.46
	ESNN Modified parameter	56.41
	ESNN-DPPSO	75.23
	DE-ESNN	–
Heart	Standard ESNN parameter	66.30
	ESNN Modified parameter	74.44
	ESNN-DPPSO	70.37
	DE-ESNN	62.71
Wine	Standard ESNN parameter	44.45
	ESNN Modified parameter	66.67
	ESNN-DPPSO	85.40
	DE-ESNN	–

carried out to evaluate the performance of the proposed ESNN-DPPSO with other methods. Table 4 shows the comparison of the results of various ESNN methods such as a standard ESNN [21], standard ESNN using manually set parameters' values, ESNN-DPPSO, and DE-ESNN.

Based on Table 4, ESNN-DPPSO shows the best accuracy results for breast cancer, diabetes, and wine datasets with the accuracy of 95.48%, 75.23%, and 85.40% respectively compared to other methods. The proposed method has improved ESNN method by at least one percent better classification accuracy compare the other techniques. The DE-ESNN has the best classification accuracy for iris dataset with the score of 93.33% because DE-ESNN uses different parameter suitable for DE such as mutation constant and crossover constant. The DE-ESNN classification accuracy result for diabetes and wine dataset are absent since it use different dataset; appendicitis and ionosphere.

6 Conclusion

In this study, a new ESNN-DPPSO optimizer technique is proposed. This simulation optimizer shows a promising result in parameter optimization. The proposed optimizer, ESNN-DPPSO, searches for optimum ESNN parameters' values. The experiment result shows that ESNN-DPPSO has a better performance result in comparison to the

standard ESNN and ESNN-DE for most datasets. For future work, more improvements on DPPSO could be carried out in order to obtain a better classification accuracy result.

Acknowledgment. This research work was supported by Universiti Teknologi Malaysia under the Research University Grant with vot. Q.J130000.2528.11H80.

References

1. Yegnanarayana, B.: Artificial Neural Networks. PHI Learning Pvt. Ltd., New Delhi (2009)
2. Huang, W., Hong, H., Song, G., Xie, K.: Deep process neural network for temporal deep learning. In: International Joint Conference on Neural Networks (IJCNN), pp. 465–472 (2014)
3. Dhoble, K., Nuntalid, N., Indiveri, G., Kasabov, N.: Online spatio-temporal pattern recognition with evolving spiking neural networks utilising address event representation, rank order, and temporal spike learning. In: The 2012 International Joint Conference Neural Networks (IJCNN), pp. 1–7 (2012)
4. Saleh, A.Y., Shamsuddin, S.M., Hamed, H.N.B.A.: Parameter tuning of evolving spiking neural network with differential evolution algorithm. In: International Conference of Recent Trends in Information and Communication Technologies, p. 13 (2014)
5. Hamed, H.N.A., Kasabov, N., Shamsuddin, S.M.: Quantum-inspired particle swarm optimization for feature selection and parameter optimization in evolving spiking neural networks for classification tasks. In: InTech (2011)
6. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceeding of IEEE International Conference on Neural Network, vol. 4, pp. 1942–1948 (1995)
7. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: IEEE Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 (1995)
8. Mao, C., Lin, R., Xu, C., He, Q.: Towards a trust prediction framework for cloud services based on PSO-driven neural network. *IEEE Access* **5**, 2187–2199 (2017)
9. Chen, Y.C., Jiang, J.R.: Particle swarm optimization for charger deployment in wireless rechargeable sensor networks. In: 26th International Telecommunication Networks and Applications Conference (ITNAC), pp. 231–236 (2016)
10. Kaur, H., Prabahakar, G.: An advanced clustering scheme for wireless sensor networks using particle swarm optimization. In: 2nd International Conference on Next Generation Computing Technologies (NGCT), pp. 387–392 (2016)
11. Pal, D., Verma, P., Gautam, D., Indait, P.: Improved optimization technique using hybrid ACO-PSO. In: 2nd International Conference on Next Generation Computing Technologies (NGCT), pp. 277–282 (2016)
12. Kasabov, N.: Evolving spiking neural networks for spatio- and spectro-temporal pattern recognition. In: 2012 6th IEEE International Conference on Intelligent Systems (IS), pp. 27–32 (2012)
13. Wysoski, S.G., Benuskova, L., Kasabov, N.: Adaptive learning procedure for a network of spiking neurons and visual pattern recognition. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 1133–1142. Springer, Heidelberg (2006). doi:[10.1007/11864349_103](https://doi.org/10.1007/11864349_103)
14. Schliebs, S., Defoin-Platel, M., Kasabov, N.: Analyzing the dynamics of the simultaneous feature and parameter optimization of an evolving spiking neural network. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2010)

15. Saxena, N., Tripathi, A., Mishra, K.K., Misra, A.K.: Dynamic-PSO: an improved particle swarm optimizer. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 212–219 (2015)
16. Kaur, R., Arora, M.: A novel asynchronous Mc-Cdma multiuser detector with modified particle swarm optimization algorithm (MPSO). In: 2nd International Conference on Next Generation Computing Technologies (NGCT), pp. 420–425 (2016)
17. Soni, N., Bhatt, R., Parmar, G.: Optimal LFC system of interconnected thermal power plants using hybrid particle swarm optimization-pattern search algorithm (hPSO-PS). In: 2nd International Conference on Communication Control and Intelligent Systems (CCIS), pp. 225–229 (2016)
18. Song, K., Li, C., Yang, L.: Parameter estimation for multi-scale multi-lag underwater acoustic channels based on modified particle swarm optimization algorithm. In: IEEE Access (2017)
19. M'hamdi, B., Tegar, M., Mekhaldi, A.: Optimal design of corona ring on HV composite insulator using PSO approach with dynamic population size. IEEE Trans. Dielectr. Electr. Insul. **23**, 1048–1057 (2016)
20. UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml/>
21. Hamed, H.N.B.A., Nuzly, H.: Novel integrated methods of evolving spiking neural network and particle swarm optimisation. Ph.D. dissertation, Auckland University of Technology (2012)