ISSN 1870-4069

Spiking Neural Networks Codification Using Bio-Inspired Computation

Carlos Alberto López-Herrera, Héctor Gabriel Acosta-Mesa, Efrén Mezura-Montes

Universidad de Veracruz,
Instituto de Investigación en Inteligencia Artificial,
Mexico

carlosalberto.lopezherrera91@gmail.com,
{heacosta,emezura}@uv.mx

Abstract. In Spiking Neural Networks, the codification of analog signals constitutes a primordial pre-processing step. Hereof, Ben's spiker algorithm, as a temporal coding schemes, is one of the most recently used methods. Nevertheless, having optimal parameters is of great importance. In this paper, the performances of two evolutionary algorithms and one swarm intelligence algorithm are contrasted in said optimization task. Moreover, a comparison against a Grid Search implementation is also presented. Our findings showed that Differential Evolution outperformed its counterparts. Furthermore, it is also proved that the same transformation capabilities, as the Grid Search, are being reached.

Keywords: Ben's spiker algorithm, differential evolution, particle, swarm optimization, genetic algorithm, grid search.

1 Introduction

Spiking Neural Networks (SNNs), the third generation of Artificial Neural Networks (ANN), were introduced as a more biologically realistic approximation [1] regarding how information is spread, compared to past generations. In the brain, the interaction between neurons is done by transmitting action potentials (or spike trains) to other nearby neurons [2].

Since all real-world signals are characterized as analog and temporal, it becomes indispensable to implement a technique capable of transforming them into spike trains and preserve as much information as possible in order to harness the usage of SNNs.

These encoding methods are often divided into two approaches: Rate and Temporal coding schemes [3]. The Rate coding strategy focuses on how information is encoded (count, density or population rate) [3, 4].

On the other hand, Temporal coding methods encode signals based on the timing of significant events [5, 6]. Furthermore, it has been noted that rate coding suffers from wide periods of latency between spikes, which may not be suitable for some SNNs applications [4]. For that reason, temporal encoding has been used in more recent works [4].

Algorithm 1 BSA encoding

input: S signal, FIR filter, threshold
$L \leftarrow \mathrm{length(S)}$, $F \leftarrow \mathrm{length(FIR)}$, $\mathrm{Out} \leftarrow \mathrm{zeros}(L)$, $\mathrm{Shift} \leftarrow \min(S)$
$S \leftarrow S - \mathrm{Shift}$
for $t = 1 : (L - F)$ **do**
  $E1 \leftarrow 0$, $E2 \leftarrow 0$
  for $k = 1 : F$ **do**
    $E1 \leftarrow E1 + \mathrm{abs}(S(t+k) - \mathrm{FIR}(k))$, $E2 \leftarrow E2 + \mathrm{abs}(S(t+k-1))$
  if $E1 \leq (E2 * \mathrm{threshold})$ **then**
    $\mathrm{Out}(t) = 1$
    for $k = 1 : F$ **do**
      $S(t+k+1) \leftarrow S(t+k+1) - \mathrm{FIR}(k)$
output: *Out*, *Shift*

Algorithm 2 BSA decoding

input: *Spikes*, *FIR* filter, *Shift*
$L \leftarrow \mathrm{conv(Spikes, FIR)} + \mathrm{Shift}$
output: *Out*

For a more in-depth analysis for these schemes, [3] provides a comprehensive review of the subject. Moreover, there are many temporal coding algorithms that have been proposed: Step-Forward (SF), Threshold-Based Representation (TBR), Moving Window (MW) and Ben's Spiker Algorithm (BSA), to name a few [3, 7]. Primarily, the latter has been used to encode data streams (e.g., Electroencephalography) [3, 7].

First introduced in [8], BSA is an extension of Hough Spiker Algorithm (HSA). The core idea behind this technique is that an analog signal can be constructed using the convolution of a spike train and a FIR filter [7, 9]. Hence, BSA uses a suitable filter to produce a spike train based on the comparison of two errors.

The first one involves the sum of differences between the signal and the filter. The second one represents the aggregated value of the signal; a spike is produced whenever the first error is smaller than the weighted (by a threshold) second error [5] (Algorithm 1).

Consequently, the BSA decoding is achieved by the convolution of the encoded spike train signal and the FIR filter (Algorithm 2). Thus, it is evident that the composition of the FIR filter and the threshold value are of great importance. The configuration of this filter relies on two main parameters: Filter size and Cutoff frequency.

In this preliminary proof of concept, two main goals are pursued: To compare the performance of two well known evolutionary algorithms (EA) and one swarm intelligence algorithm (SI) for the optimization of the BSA parameters (Filter size, cutoff frequency and threshold) and to contrast the best performing EA or SI against a Grid Search (GS).

Moreover, the reason to choose GS as a comparative method is not only because it is a deterministic technique, but also because it was used in [7] as a optimization technique. In order to measure the BSA efficiency, three metrics criteria will be used:

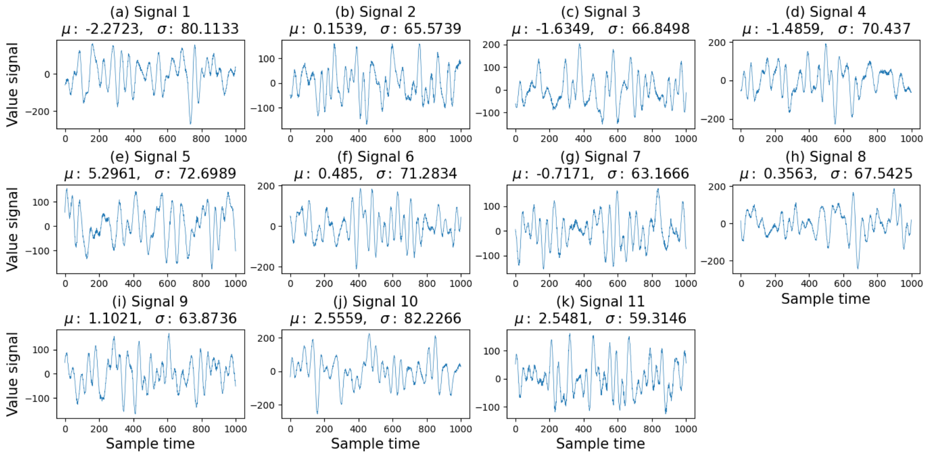

Fig. 1. All signals created with a length of 1001 elements and sampled at 1000 Hz.

Table 1. Range for each variable as established in [7].

Variable	Range	Increment
Filter size	17 - 81	4
Cutoff frequency	20 - 80	2
Threshold	0.8 - 1.1	0.01

– **Signal to Noise Ratio (SNR):** Measures the relation involving the original signal power and the noise signal power. Noise is considered as the difference between the original signal (s) and the decoded signal (r). Higher SNR values mean better results. It is defined as:

$$\text{SNR} = 10 \cdot \log_{10} \left(\frac{\sum_{t}^{N} s_t^2}{\sum_{t}^{N} (s_t - r_t)^2} \right). \tag{1}$$

– **Absolute Firing Rate (AFR):** Indicates the saturation of the spike train (sp). Lower AFR values mean a less saturated signal. It is defined as:

$$\text{AFR} = \frac{\sum_{t}^{N} |sp_t|}{N}. \tag{2}$$

– **Symmetric Mean Absolute Percentage Error (sMAPE):** A percentage error that measures accuracy between the original signal and the reconstructed one [10], [11]. Unlike SNR, this metric considers both, the original signal (s) and the reconstructed signal (r) as independent from each other. Lower sMAPE values mean better results. It is defined as:

$$\text{sMAPE} = \frac{1}{N} \sum_{t=1}^{N} \frac{|r_t - s_t|}{|r_t| + |s_t|} \cdot 100\%. \tag{3}$$

Table 2. Parameter values used for each EA and SI algorithm. These values were selected by a trial-and-error process.

(a) GA

Version	Canonical (Real representation)
Population	50
Crossover	SBX(η+10 - 90%
Mutation	Uniform - 60%
Parent selection	Probabilistic binary tournament - 60%
Elitism	1
Boundary management	Ran[13]
Generations	200

(b) DE

Version	DE/rand/1/bin
Representation	Real
NP	50
Cr	0.8
F	0.5
Boundary management	Ran[13]
Generations	200

(c) PSO

Version	Global-best PSO
Representation	Real
Cumulus size	50
W	0.65
C_1	1.2
C_2	1.4
Boundary management	Ran&RaB[14]
Generations	200

Table 3. Statistical results of 30 independent executions. Values in boldface indicate the best value. **H=1** means that a significant difference was found.

Statistic	GA	DE/rand/1/bin	Global-best PSO	Friedman test	
				p-value	**H**
Best	10.7257	**10.7826**	10.7825	2.46E-13	1
Mean	10.6966	**10.7726**	10.7294		
Median	10.7024	**10.7825**	10.7250		
Worst	10.6349	**10.7310**	10.5168		
Std Dev	0.0206	**0.0151**	0.0497		

The rest of this paper is structured as follows: In Section 2, the methodology for two experiments to be conducted is explained. Section 3 describes the experiments layout, as well as the corresponding results. In Section 4, a general discussion is made of the achieved results and the evidence observed. Section 5 consists of some conclusions attained as well as ideas for future work.

2 Methodology

Using the implementation of [7], eleven signals were created (Fig. 1). These are produced by a composition of sine signals ranging from 2 to 30 Hz with random power and random phase lags. Also, white noise was added with a strength of 3. All signals have a length of 1001 elements, sampled at 1000 Hz.

The first experiment consists in comparing performances for the optimization of the BSA parameters. Two commonly used EA are considered. Namely, Genetic Algorithm (GA) and Differential Evolution (DE). Also, a SI algorithm called Particle Swarm Optimization (PSO) is tested as well. Since SNR is highly recommended [6, 7, 12], this metric was used as the objective function for the three compared approaches. The first signal (Fig. 1a) was utilized.

In the second experiment, the optimization method[1] proposed in [7], where a GS is employed to find the optimal set of parameters, was applied to each remaining signal (Fig. 1b - 1k). After that, the best performing EA or SI from the previous experiment was used to the same task on the same signals.

This test aims at proving the transformation capabilities of an EA or SI against a deterministic and proved method. The GS ranges of each variable are shown in Table 1. Moreover, the parameters for each bio-inspired algorithm are presented in Table 2.

[1] github.com/KEDRI-AUT/snn-encoder-tools

Table 4. Details of the median run, by each algorithm. a) Metrics comparing the results of the algorithms. Values in boldface indicate a better result. b) Set of parameters found by each algorithm.

(a) Metrics achieved

	GA	DE/rand/1/bin	Global-best PSO
SNR	10.7024	**10.7825**	10.7250
sMAPE	13.6305	**13.2633**	13.5431
AFR	0.3337	0.3357	**0.3337**

(b) Parameters found

	GA	DE/rand/1/bin	Global-best PSO
Filter size	71	69	71
Cutoff frequency	49.7601	38.8649	49.1676
Threshold	0.9563	0.9572	0.9564

(a) Convergence graph.

(b) Visual comparison

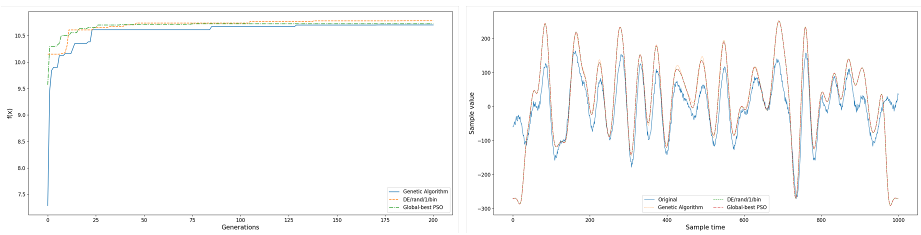

Fig. 2. Graphical results obtained. a) Convergence graph of the median execution by each algorithm. b) Visual comparison between the original signal and the reconstructed signals using the parameters found at the best execution.

3 Experiments and Results

3.1 Experiment 1

The focus of this experiment is to compare the performances of two common EA and one SI algorithm for the optimization of the BSA parameters on a given signal. To achieve this, 30 independent executions were performed per compared algorithm using the first signal (Fig. 1a). The same ranges of the variables (Table 1) were acknowledged in each algorithm implementation.

In Table 3, the statistical analysis of SNR values (objective function) obtained in all executions and the Friedman test results (95%-confidence) are presented. Furthermore, in Table 4 the details of the algorithms median run are shown. Table 4a refers to the metrics, whereas Table 4b presents the parameters found. Finally, the convergence graphs of the three algorithms are presented in Figure 2a. Also, Fig. 2b includes the contrast between the original signal and the reconstructed signal by the three compared algorithms.

From these results, it is noticeable that DE/rand/1/bin outperformed both, GA and Global-best PSO. This is also validated by the Friedman test (Table 3). Furthermore, all parameters found are quite similar. On this regard, DE/rand/1/bin managed to get a better result using a lower filter size (Table 4).

Additionally, all algorithms showed a similar convergence dynamic (Fig. 2a): the exploration seems to decrease rapidly. Similarly, Fig. 2b shows that the reconstructed signals do not exhibit mayor differences among the algorithms implementations despite the variations observed by metrics.

Table 5. Detail of Wilcoxon rank-sum results based on SNR metric. **H=0** means no significant difference was found.

Statistic	GS	DE/rand/1/bin	Friedman test	
			p-value	**H**
Mean	9.1896	9.4786	0.2730	0
Std Dev	0.9367	0.9811		

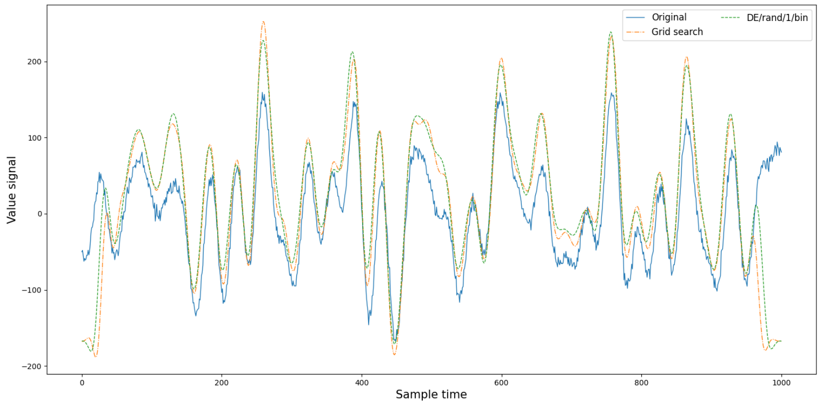

Fig. 3. Visual comparison between signal reconstruction from GS and DE for signal 2.

3.2 Experiment 2

The main goal of this experiment is to test the best performing bio-inspired algorithm out of the previous experiment against the implementation in [7] on ten different signals (Fig. 1b - 1k). Considering the fact that DE showed the best performance, it was selected for this experiment.

The parameters from the previous implementation were kept (Table 2b). In order to contrast the overall performance of GS and DE, Table 5 shows the statistical values obtained from SNR metrics of all signals, as well as the results of the Wilcoxon rank-sum test (95%-confidence).

Finally, a representative visual comparison of all tested methods on signal 2 (Fig. 1b) is presented in Fig. 3. From Table 5, it can be establish that no significant differences were found between GS and DE. This is also supported by the visual comparison highlighted in Fig. 3, where no mayor differences are visible.

4 Discussion

Firstly, experiment 1 shows that DE/rand/1/bin exhibited a better performance over the other algorithms tested (Table 3). Moreover, DE was able to lower the filter size variable the most. This is of great importance since BSA involves the deconvolution/convolution of a signal by the FIR filter. Hence, the smaller the filter size, the lower the cost and computational time.

Nevertheless, the quantitative improvements observed where not greatly reflected during visual comparison (Fig. 2b). Also, it seems that all bio-inspired algorithms converge fairly quick (Fig. 2a). Therefore, more efforts could be made in order to improve the explorations.

Finally, experiment 2 presents a direct contrast between GS and DE implementations applied to ten different signals. From there, the observations of GS and DE showed no significant differences overall (Table 5). This is also reaffirmed by the visual comparison performed between the reconstructed signal of GS and DE against the original signal. This evidence points to the ability of DE to match the performance of a GS with fixed ranges.

5 Conclusions and Future Work

SNNs models represent a paradigm shift from its predecessors; the key differentiation lies in how information is conveyed. Since SNNs use spike trains, a crucial question to be asked is: How can we translate analog signals to an impulse-based representation? In this paper, BSA was chosen as a method to transform analog signals to spike trains.

We tested two evolutionary algorithms and one swarm intelligence algorithm to optimize parameters for the already mentioned technique. It was found that DE/rand/1/bin performed better than its counterparts. Yet, results did not significantly improve the transformation capabilities.

On the other hand, the second experiment compared results between DE and the implementation in [7]. In such reference, a GS was used to find optimal parameters. Nevertheless, this method restricts the search space in order to lower the computational cost and time.

This also implies a certain prior knowledge in order to narrow the variables. In contrast, the DE implementation did not required any increment restriction. Having said that, our findings showed that DE could be considered as a optimizer of BSA parameters. Finally, future work directions could include different paths:

1. Perform experiments with more specialized, bio-inspired algorithms.
2. Implement a parameter tuning method in the algorithms calibration.
3. Evaluate the use of surrogate models for signal transformation.
4. Asses the possibility of a bio-inspired algorithm optimization using the classification performance of a generic SNN as objective function.

Acknowledgments. The first author acknowledges support from Consejo Nacional de Ciencia y Tecnología (CONACyT), Mexico through scholarship No. 1075919 to pursue graduate studies at University of Veracruz.

References

1. Maass, W.: Networks of spiking neurons: The Third Generation of Neural Network Models. Neural Networks, vol. 10, no. 9, pp. 1659–1671 (1997). DOI: 10.1016/s0893-6080(97)00011-7.

2. Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., Maida, A.: Deep Learning in Spiking Neural Networks. Neural Networks, vol. 111, pp. 47–63 (2019). DOI: 10.1016/j.neunet.2018.12.002.
3. Auge, D., Hille, J., Mueller, E., Knoll, A.: A Survey of Encoding Techniques for Signal Processing in Spiking Neural Networks. Neural Processing Letters, vol. 53, no. 6, pp. 1–18 (2021). DOI: 10.1007/s11063-021-10562-2.
4. Kasabov, N.: Evolving Spiking Neural Networks for Spatio-and Spectro-temporal Pattern Recognition. International Conference Intelligent Systems (2012). DOI: 10.1109/is.2012.6335110.
5. Dupeyroux, J., Stroobants, S., de-Croon, G.: A Toolbox for Neuromorphic Sensing in Robotics (2021) DOI: 10.1109/EBCCSP56922.2022.9845664.
6. Tan, C., Šarlija, M., Kasabov, N.: Spiking Neural Networks: Background, Recent Development and the neuCube Architecture. Neural Processing Letters, vol. 52, no. 2, pp. 1675–1701 (2020). DOI: 10.1007/s11063-020-10322-8.
7. Petro, B., Kasabov, N., Kiss, R. M.: Selection and Optimization of Temporal Spike Encoding Methods for Spiking Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 2, pp. 358–370 (2020). DOI: 10.1109/tnnls.2019.2906158.
8. Schrauwen, B., Van Campenhout, I.: BSA, a Fast and Accurate Spike Train Encoding Scheme. In: Proceedings of the International Joint Conference on Neural Networks, vol. 4, pp. 2825–2830 (2003). DOI: 10.1109/ijcnn.2003.1224019.
9. Moinnereau, M. A., Brienne, T., Brodeur, S., Rouat, J., Whittingstall, K., Plourde, E.: Classification of Auditory Stimuli from EEG Signals with a Regulated Recurrent Neural Network Reservoir (2018). DOI: 10.48550/ARXIV.1804.10322.
10. Armstrong, J. S., Forecasting, L. R.: From Crystal Ball to Computer. vol. 348 (1985)
11. Hyndman, R. J., Athanasopoulos, G.: Forecasting: Principles and Practice, o texts: Melbourne, 2nd edition (2018)
12. Sengupta, N., Kasabov, N.: Spike-time Encoding as a Data Compression Technique for Pattern Recognition of Temporal Data. Information Sciences, vol. 406–407, pp. 133–145 (2017). DOI: 10.1016/j.ins.2017.04.017.
13. Lampinen, J.: A Constraint Handling Approach for the Differential Evolution Algorithm. In: Proceedings of the Congress on Evolutionary Computation, vol 2, pp. 1468–1473 (2002). DOI: 10.1109/cec.2002.1004459.
14. Clerc, M.: Confinements and Biases in Particle Swarm Optimization. Technical Report hal-00122799 (2006)