

# EVOLVING RANDOM TOPOLOGIES OF SPIKING NEURAL NETWORKS FOR PATTERN RECOGNITION

Gustavo López-Vázquez<sup>1</sup>, Manuel Ornelas-Rodríguez<sup>1</sup>, Andrés Espinal<sup>2</sup>,  
Jorge A. Soria-Alcaraz<sup>2</sup>, Alfonso Rojas-Domínguez<sup>1</sup>, Héctor J. Puga-  
Soberanes<sup>1</sup>, J. Martín Carpio<sup>1</sup> and Horacio Rostro-González<sup>3</sup>

<sup>1</sup>Division of Postgraduate Studies and Research, National Technology of  
México / León Institute of Technology. León, Guanajuato, México

<sup>2</sup>Department of Organizational Studies (DCEA), University of Guanajuato.  
Guanajuato, Guanajuato, México

<sup>3</sup>Department of Electronics (DICIS), University of Guanajuato.  
Salamanca, Guanajuato, México

## ABSTRACT

*Artificial Neural Networks (ANNs) have been successfully used in Pattern Recognition tasks. Evolutionary Spiking Neural Networks (ESNNs) constitute an approach to design third-generation ANNs (also known as Spiking Neural Networks, SNNs) involving Evolutionary Algorithms (EAs) to govern some intrinsic aspects of the networks, such as topology, connections and/or parameters. Concerning the practicality of the networks, a rather simple standard is commonly used; restricted feed-forward fully-connected network topologies deprived from more complex connections are usually considered. Notwithstanding, a wider prospect of configurations in contrast to standard network topologies is available for research. In this paper, ESNNs are evolved to solve pattern classification tasks, using an EA-based algorithm known as Grammatical Evolution (GE). Experiments demonstrate competitive results and a distinctive variety of network designs when compared to a more traditional approach to design ESNNs.*

## KEYWORDS

*Artificial Neural Networks, Spiking Neural Networks, Evolutionary Spiking Neural Networks, Evolutionary Algorithms, Grammatical Evolution*

## 1. INTRODUCTION

The Artificial Neural Networks (ANNs) are computational entities constituted by mathematical models that drew inspiration from the neurological processes occurring in the biological brain; insight in the field of Neuroscience has been harnessed to be recreated in the Computational Intelligence field, resulting in a powerful approach to solve a great variety of computational problems [1, 2, 3]. Usually depicted as a set of interconnected units arranged into layers, ANNs exchange some kind of data [4] and are subject to a training procedure for them to learn from the

incoming information [5]. Ever since the proposal of the first models of ANNs [6, 7], researchers have found improvements on their computational power, connections [8], learning capabilities [4, 5] and even resemblance to natural networks [9], leading to a profound interest in examining and expanding their scope.

Attempts in this matter have caused the conception of Spiking Neural Networks (SNNs) [10], advanced models of so-called third-generation ANNs catalogued for the inclusion of the time factor in their computation. This enhancement sets the SNNs one step ahead (with respect to older ANNs models) regarding their likeness to biological neurons, since it allows them to deal with spatio-temporal information. Their capabilities have been properly displayed in the literature [10, 11, 12].

But in spite of the advantages on their performance, ANNs are not without a fundamental issue related to their conformation: their design [13, 4, 10]. The ANNs problem-solving capabilities are strongly correlated to their internal construction (i.e., their topology, connectivity, parameterization, etc.), and so the importance of a good design process is evident. Traditionally, the ANNs are empirically devised in accordance to the problem at hand, with aid of the designer experience and trial-and-error techniques, a time-consuming and ever-perfectible task. However, some evolution-oriented approaches have been introduced to address the problem; by using Evolutionary Algorithms, the designer is able to establish well-known mechanisms to automatically optimize whichever facet of the network design specification. Praxis in this manner has led to the formulation of Evolutionary Artificial Neural Networks (EANNs) that require less *a priori* comprehension of the networks from the designer, to produce evolved ANNs useful for solving problems while helping to alleviate the design issues [14, 15].

This work proposes a framework to automatically evolve the design of Spiking Neural Networks by means of an Evolutionary Algorithm known as Grammatical Evolution, allowing for a higher degree of freedom in determining their design criteria; generated Evolutionary Spiking Neural Networks (ESNNs) possess unrestricted partially-connected network topologies containing various types of arbitrary connections in order to provide solutions to Pattern Recognition problems. Section 2 locates some important background concepts related to the present paper. Section 3 pinpoints the methodology employed to fulfill the purpose of this work, Section 4 shows the experimental results obtained by applying the methodology and Section 5 gives some insight on the discussion and conclusions achieved.

## 1.1. Related Work

Evolutionary Algorithms such as Genetic Programming and Evolutionary Strategies are employed to define design traits in Artificial Neural Networks such as training [16, 17] and topology [18]. Furthermore, utilizing Grammatical Evolution as a mean to define multiple design criteria such as topologies and neural model parameters of second-generation ANNs is considered in [19], and for third-generation ANNs in [20]. In [21] third-generation partially-connected topologies are generated with GE. Although some of these methods may consider any conceivable amount of designing criteria, the traditional approach seem to remain abided; restricted (i.e., connections between hidden units are not allowed) feed-forward network topologies figure as the standard for EANNs generation in the related work. As the evolutionary process aims to drive the design towards the best solution possible, a wider allowance in options may benefit the design process, preserving or improving the performance of the networks while considering unexplored forms of solutions.

## 2. BACKGROUND

### 2.1. Artificial Neural Networks

The Artificial Neural Networks (ANNs) encompass a range of computational models distinguished by their connectivity traits; usually depicted as an interconnected group of computing units (or neurons), the ANNs convey information to other units through their links (or synapses) mimicking the manner a biological brain does, with the objective to fulfill a learning process to ultimately solve a specific problem.

Fundamentally, ANNs are conceived as a composition of linked neurons disposed in groups (or layers) of at least three types: input layer, hidden layer(s), output layer. External stimuli is provided to the network through the input layer, forwarded throughout the hidden layers, and egressed in the output layer. The connectivity links are modulated (or weighed) in order to enhance or inhibit the sharing information between the units; a learning process is intended to find the most proper regulation of the synapses to comply with a desired output to solve a problem.

Model-wise, ANNs are categorized in three groups (or generations), in relation to the type of data they are able to compute. First-generation ANNs support digital data by using threshold units [6, 7], second-generation ANNs comprise both digital and analogic data as they incorporate continuous activation functions, and third-generation ANNs in addition to digital and analogical data are able to handle spatio-temporal data since their computing units –spiking neurons– encode information into short electrical pulses spread along a time lapse.

### 2.2. Spiking Neural Networks

The development of the third generation of ANNs has involved an endeavour to comprehend the information-processing capabilities of the brain in order to mimic their behaviour. By introducing a previously obviated element, –the pulsating response of a neuron over time [22] –, the Spiking Neural Networks are inherently provided with fundamental neurological elements, viz., neuronal coding, signal transmission, refractoriness or synaptic plasticity. As modelling approaches appear to describe such traits [9], their computing capabilities are assessed altogether.

#### 2.2.1. Spike Response Model

Particularly, the Spike Response Model (SRM) is of specific interest in this work, as it serves as basis for the produced SNNs. In SRM, the state of a neuron  $j$  (i.e., its internal electric potential) over time  $t$  is determined by Eq. (1). When the value of  $x_j(t)$  reaches the threshold  $\theta$  from below, the neuron fires a spike (see Fig. 1).

$$x_j(t) = \sum_{i \in \Gamma_j} w_{ji} y_i(t) \quad (1)$$

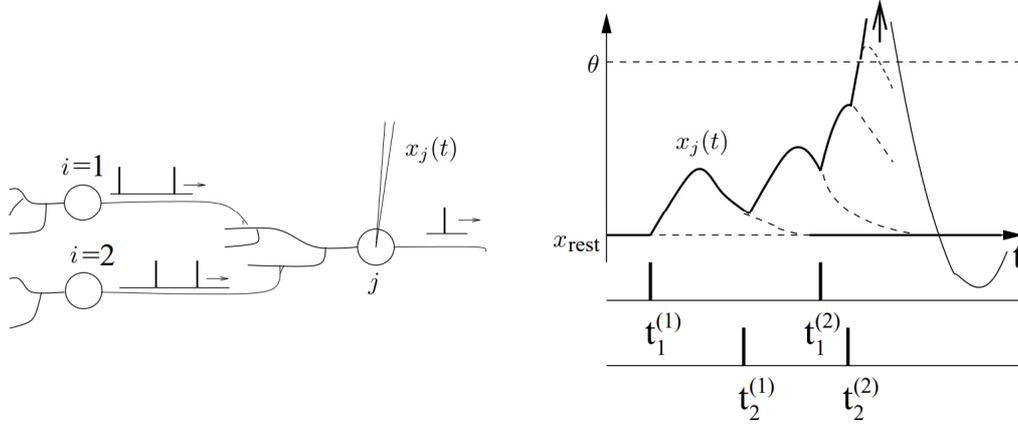


Figure 1. Linear summation of presynaptic spikes of neuron  $j$ . Adapted from [9]

In Eq. (1),  $w_{ji}$  acts as a modulation parameter of the specific synapse. The set of spikes from the connecting presynapses of  $j$  ( $\Gamma_j$ ) contribute to the linear summation  $y_i(t)$  described by Eq. (2).

$$y_i(t) = \begin{cases} \frac{t-t_i-d_{ji}}{\tau} \exp\left[1 - \frac{t-t_i-d_{ji}}{\tau}\right] & \text{if } (t - t_i - d_{ji}) > 0 \\ 0 & \text{else} \end{cases} \quad (2)$$

where  $\tau$  is a time constant that defines the decay of the postsynaptic potential.  $t_i$  and  $d_{ji}$  are the firing time and synaptic delay parameter of the neuron  $i$ , respectively.

### 2.3. Evolutionary Spiking Neural Networks

In accordance to the generalization Evolutionary Artificial Neural Networks (EANNs) [23], Evolutionary Spiking Neural Networks (ESNNs) can be specified as a paradigm that involves the utilization of Evolutionary Algorithms to determine the design of Spiking Neural Networks. This approach incorporates the advantages of a metaheuristic process for optimization purposes, and the computing capabilities of third-generation artificial neural models.

In this approach, a population-based algorithm is commonly used to subject a wide range of aspects in the network design to be optimized; the Evolutionary Algorithms capabilities are employed to evolve design criteria such as parameters [24], topology [25], learning rule [26] or a combination of those [27].

## 3. METHODOLOGY

The methodology employed in this paper is portrayed in Fig. 2. In it, an encoded rendering of the dataset information is provided to the Grammatical Evolution –the selected Evolutionary Algorithm–, to ultimately provide Evolutionary Spiking Neural Networks to solve the Pattern Classification problem. The Grammatical Evolution per se is operated internally by the definition of a problem-related Grammar, a Target Specification, a Mapping Process, and a Metaheuristic

Optimization Engine guided by an Objective Function; these elements are revisited in this section.

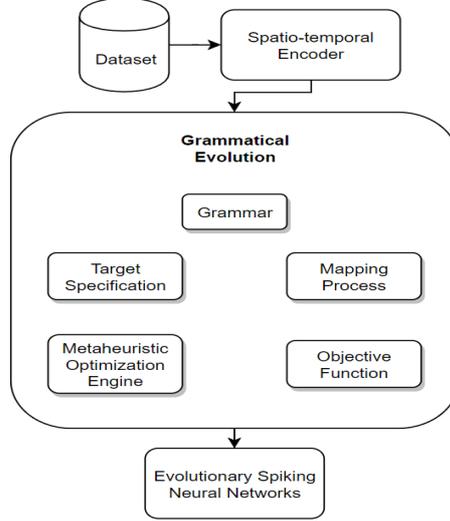


Figure 2. Flowchart of the methodology employed in this work

### 3.1. Spatio-temporal Encoder

The Spatio-temporal Encoder acts as a function to transform the information in the datasets into a comprehensible form to feed the spiking networks. Since the input data often stands for static information (whether digital or analogical), and third-generation ANNs are designed to process spatio-temporal data, a conversion process is necessary to accommodate the needs. The one-dimension encoder employed in this work is portrayed in Eq. (3) [28].

$$y(x) = \left[ \frac{(t_u - t_l)}{r} * x \right] + \left[ \frac{(t_l * x_M) - (t_u * x_m)}{r} \right] \quad (3)$$

where  $x$  is the original input data,  $x_M$  and  $x_m$  are the maximum and minimum values that  $x$  takes,  $y(x)$  is the spatio-temporal transformation over a period of time  $[t_l, t_u]$ , and  $r$  is the range between  $x_M$  and  $x_m$ . The observed encoding scheme was adjusted to retain a temporal range from 0.01 to 9 milliseconds (ms.).

### 3.2. Grammatical Evolution

The Grammatical Evolution (GE) [29] is the Evolutionary Algorithm used to guide the optimization process for the generated ESNNs in this work. The nature of GE involves several elements, described as follows.

#### 3.2.1. Grammar

The *Backus-Naur Form* Grammar 1 is proposed to govern the generation of random network topologies, and to determine the model parameters (weights and delays). It allows the inclusion of zero or more hidden layers (delimited by the closing curly bracket symbol `}`), composed by one or more neuron units (separated by the closing square bracket symbol `]`), that may contain one or

more synaptic connections (marked by the closing parenthesis symbol `)`). The output layer is composed by one computing unit with one or more synapses. Synaptic connections may be established with regard to a specific neuron index from a sequential list of integer numbers (`<neuronIndex>`), meaning that any given neuron is able to set up synapses with any neuron in the network, resulting in types of connections such as loops, between units in the same layer or in different layers. In this framework, the amount of elements in such list is determined as two times the number of input features minus one.

```

<architecture> = <hiddenLayers> | <outputLayer>
<hiddenLayers> = <hiddenLayers><hiddenLayer> | <hiddenLayer> |
<hiddenLayer> = <neurons>}
<outputLayer> = <neuron>
      <neurons> = <neurons><neuron> | <neuron>
      <neuron> = <synapses>]
<synapses> = <synapses><synapse> | <synapse>
<synapse> = <neuronIndex>@<weight>,<delay>
<neuronIndex> = 1 | 2 | 3 | 4 | 5 | 6 | 7...
<weight> = <sign><digit><digit><digit>.<digit><nonZeroDigit>
<delay> = <delayBound><digit>.<digit><nonZeroDigit>
<sign> = + | -
<delayBound> = 0 | 1
<nonZeroDigit> = 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<digit> = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

```

Grammar 1. Proposed BNF Grammar for designing ESNNs

### 3.2.2. Target Specification

The Target Specification alludes the manner in which the networks are conditioned to learn from the input data to denote a proper prediction of the classes in the dataset. Apropos of this work, the time of the first spike in the simulation of the lone output unit of the networks acts as the specification of the target, allocating each class a well-defined time; samples processed are expected to trigger a first-spike in the simulation of the output unit equal or near the time specified for its respective class.

### 3.2.3. Mapping Process

The mapping process employed in this framework is supported by a depth-first search algorithm to transform the genetic definition of the individuals in a population into words encoding the actual network (topology and parameters).

### 3.2.4. Metaheuristic Optimization Engine

The Grammatical Evolution is aided by a metaheuristic process to carry out the evolution of a population of individuals. Such process was designated to be the Differential Evolution (DE) [30] for the purposes of this work, as in concordance with [20]. The DE observes the following parameters: Real Search Space [0, 255], *Individual Dimension* = 500, *Function calls* = 1,000,000, *Population size* = 100, *Crossover Rate* = 10%, *Mutation* - DE/Rand/1.

### 3.2.5. Objective Function

The objective function is used to provide an indicative of how well-fitted a specific solution is to solve the problem. This measure is employed by the Metaheuristic Optimization Engine to guide the evolution towards the optimal. The Eq. (4) defines the accuracy error in the ratio of the proper classifications ( $C$ ) and the total amount of predictions ( $T$ ) made by a network on the design set, and so is applied in this framework. Lower error values indicate a better performance of the networks, so the function is designed for minimization.

$$E = 1 - \left(\frac{C}{T}\right) \quad (4)$$

## 4. EXPERIMENTAL RESULTS

The proposed method was implemented to comply with the experimental scheme detailed in this section. Twelve pattern recognition benchmark datasets from the UCI Machine Learning Repository [31] were considered: Balance Scale, Blood Transfusion Service Center (Blood), Breast Cancer Wisconsin (Breast Cancer), Japanese Credit Screening (Card), Pima Indians diabetes (Diabetes), Fertility, Glass Identification (Glass), Ionosphere, Iris Plant, Liver Disorders (Liver), Parkinson and Wine. Table 1 shows details of the datasets employed.

Every dataset was randomly divided in two, and the halves balanced by including in each the same amount of instances of every class. One half was labelled as the Design set, and it was employed to run the Grammatical Evolution process in order to obtain the best possible network. The remaining half was labelled the Test set, and it was supplied to the best network obtained by the Design set to assess its accuracy on a previously unseen set.

For comparison purposes, two configurations were considered, performing forty independent experiments for each, as defined next:

- Configuration X. Conceived to replicate the configuration with the best results obtained in [20] (neural model parameterization was thoroughly reproduced). Restricted single-hidden-layer feed-forward topologies are achieved in all cases.
- Configuration Y. It emulates configuration X but employing the proposed Grammar 1 to broaden the amount of connection types available for selection in the evolutionary process. The following types are added: looped (within the same unit), unrestricted (between units in the same layer), supralayer (between units in non-adjacent layers).

Table 1. Details of the datasets employed for experimentation.

Dataset	Instances	Classes	Features
Balance Scale	625	3	4
Blood	748	2	4
Breast Cancer	683	2	19
Card	653	2	15
Diabetes	768	2	8
Fertility	100	2	9
Glass	214	6	9

Ionosphere	351	2	33
Iris Plant	150	3	4
Liver	345	2	6
Parkinson	195	2	22
Wine	178	3	13

Results obtained are summarized in Table 2 denoting average accuracies of classification on both labeled datasets. Highest values are emphasized in boldface. Furthermore, Figures (3, 4, 5, 6) show example networks (and their corresponding word) generated by the framework; solid lines represent traditional feed-forward connections, dashed lines portrait supralayer connections, and densely dotted lines mark unrestricted connections. E.g., Figure 3 corresponding word is portrayed in three lines to denote the synapses of the generated units, as described in section 3.2.1. The first line describes the connection of the hidden unit with the unit 10, a synapse weight of 915.59 and a delay of 11.61. The second line only denotes the separation of hidden layers, as there are not any more hidden units in this example. The third and last line shows the incoming synapses of the output unit (from units 1, 8, 4 and 3), and their respective weights and delay values. This very analysis of the corresponding words relating to the network topology can be made to Figures 4, 5 and 6 as well.

Table 2. Accuracy averages (and standard deviation) of Design and Test classification for both considered configurations on every experimentation dataset.

Dataset	Configuration	Design Accuracy	Test Accuracy
Balance Scale	X	<b>0.8957 ± 0.0068</b>	<b>0.8662 ± 0.0132</b>
	Y	0.8875 ± 0.0088	0.8595 ± 0.0169
Blood	X	0.7996 ± 0.0161	0.7710 ± 0.0155
	Y	<b>0.8021 ± 0.0108</b>	<b>0.7710 ± 0.0120</b>
Breast Cancer	X	<b>0.9751 ± 0.0059</b>	<b>0.9466 ± 0.0117</b>
	Y	0.9642 ± 0.0068	0.9457 ± 0.0115
Card	X	<b>0.8804 ± 0.0133</b>	<b>0.8585 ± 0.0155</b>
	Y	0.8782 ± 0.0121	0.8547 ± 0.0159
Diabetes	X	<b>0.7937 ± 0.0115</b>	0.7361 ± 0.0187
	Y	0.7932 ± 0.0145	<b>0.7370 ± 0.0168</b>
Fertility	X	<b>0.9395 ± 0.0192</b>	<b>0.8495 ± 0.0326</b>
	Y	0.9340 ± 0.0220	0.8350 ± 0.0376
Glass	X	<b>0.7121 ± 0.0193</b>	<b>0.6177 ± 0.0408</b>
	Y	0.6943 ± 0.0238	0.5968 ± 0.0386
Ionosphere	X	<b>0.9616 ± 0.0108</b>	<b>0.9001 ± 0.0215</b>
	Y	0.9526 ± 0.0114	0.8993 ± 0.0264
Iris Plant	X	<b>0.9920 ± 0.0078</b>	0.9373 ± 0.0259
	Y	0.9887 ± 0.0101	<b>0.9380 ± 0.0274</b>
Liver	X	<b>0.7677 ± 0.0227</b>	0.6604 ± 0.0326
	Y	0.7635 ± 0.0211	<b>0.6656 ± 0.0304</b>
Parkinson	X	<b>0.9242 ± 0.0161</b>	<b>0.8477 ± 0.0362</b>
	Y	0.9142 ± 0.0172	0.8321 ± 0.0327
Wine	X	<b>0.9665 ± 0.0170</b>	<b>0.8692 ± 0.0441</b>
	Y	0.9327 ± 0.0194	0.8483 ± 0.0597

```

100+915.59,11.61)]
}
10+393.87,07.53)80+717.86,08.21)40+828.89,11.13)30-279.67,06.85)]
    
```

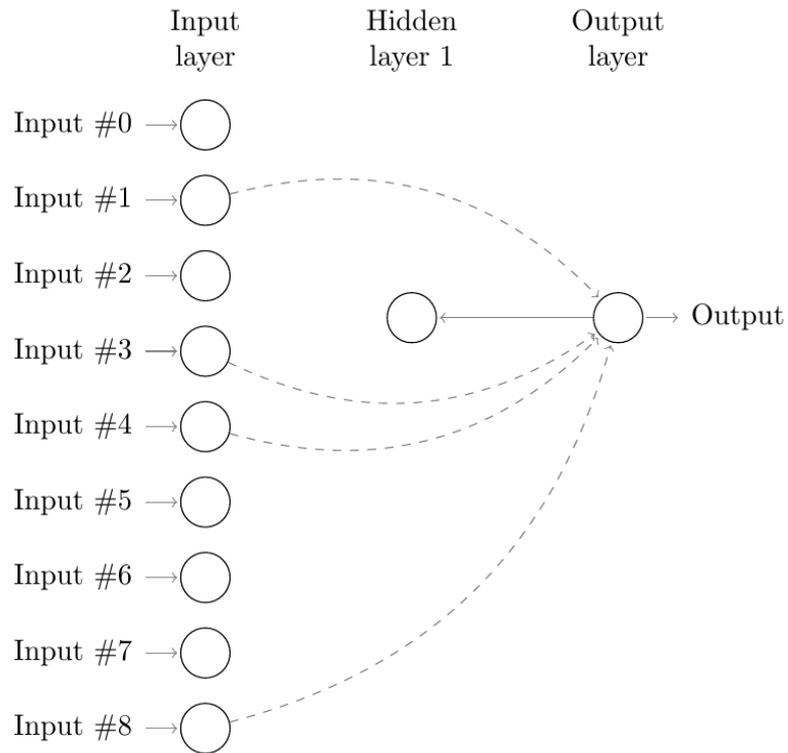


Figure 3. Example network and corresponding word. Dataset: *Fertility*. Configuration: *Y*. Accuracy of design 0.94. Accuracy of test: 0.9

```

40+194.43,15.69)50-007.07,15.06)]
30+535.53,16.25)]
30-898.03,16.14)]
10+460.06,08.81)10-700.05,03.67)60+407.65,05.88)10-145.91,01.09)60+620.16,09.33)]
}
60+736.34,13.72)]
}
00-210.55,04.48)20-839.22,19.96)20-026.57,16.02)20+696.79,07.75)40-500.08,09.67)30+675.36,09.06)]
    
```

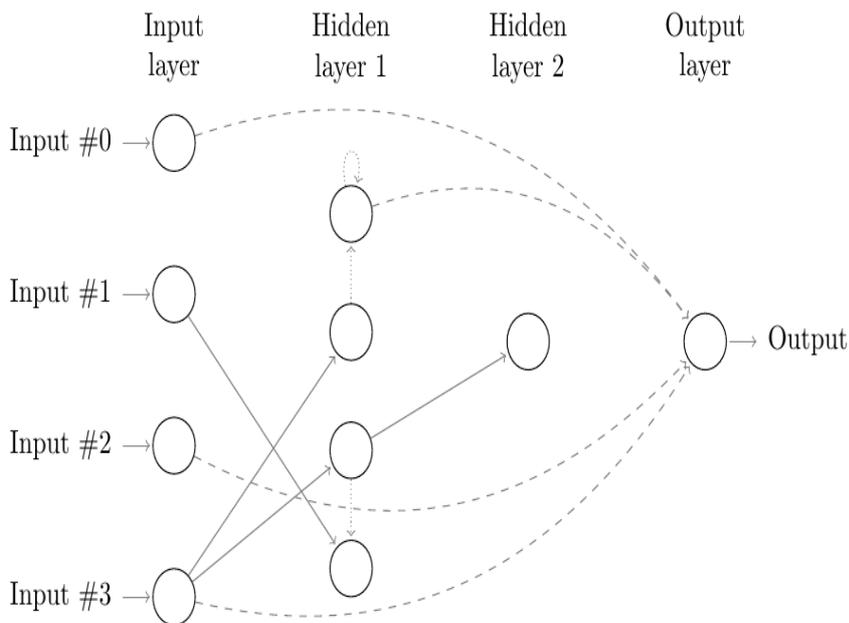


Figure 4. Example network and corresponding word. Dataset: *Iris Plant*. Configuration: *Y*. Accuracy of design 0.9867. Accuracy of test: 0.9867

```

100+453.91,04.75)]
}
80-674.73,04.59)]
10+551.15,02.62)90+810.11,11.51)]
110+764.01,02.96)150+973.28,07.31)]
110+087.89,17.76)]
160-472.34,00.28)]
10+042.42,08.43)]
}
60-244.88,04.33)110-251.07,12.96)80-039.78,10.13)70+309.49,08.45)]
}
100-608.01,13.03)110-916.05,02.25)]
}
160-115.67,12.13)20-119.79,08.79)30+030.82,09.76)160+712.73,04.01)]
    
```

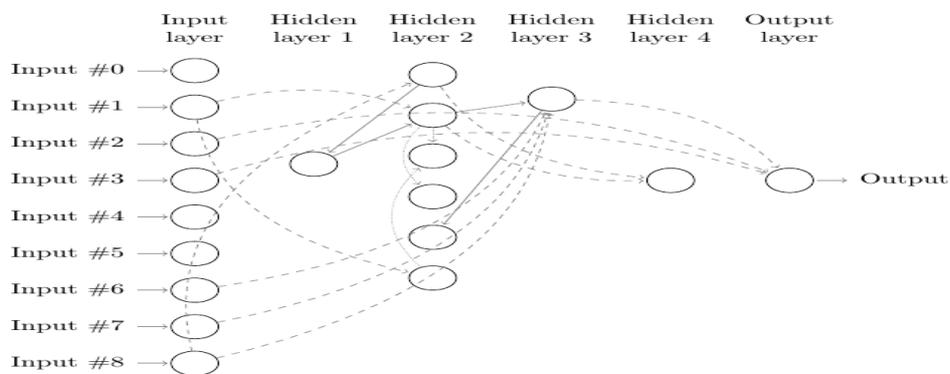


Figure 5. Example network and corresponding word. Dataset: *Glass*. Configuration: *Y*. Accuracy of design 0.6667. Accuracy of test: 0.6606

```
60-949.69,14.29]
60+568.81,11.59)40978.34,03.17)50-035.75,08.96)10-449.28,04.68)00-450.61,05.66)20+407.41,05.21)30+464.98,06.13)]
}
00+414.31,16.12)40+680.67,05.14)50+505.34,03.25)]
```

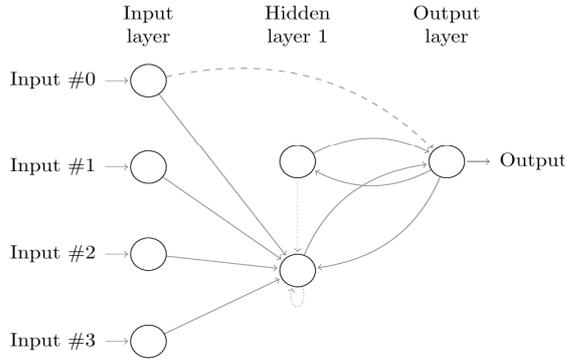


Figure 6. Example network and corresponding word. Dataset: *Balance Scale*. Configuration: *Y*. Accuracy of design 0.8878. Accuracy of test: 0.8882

#### 4.1. Statistical Analysis

A Wilcoxon signed-rank test was applied to the Test Accuracy of the obtained data in order to determine whether or not the results belonged to the same distribution, i.e. to distinguish on the efficacy of one configuration over the other. The null hypothesis of the Wilcoxon signed-rank test states that compared populations have the same distribution with the same median. Table 3 shows the results obtained by the Wilcoxon test for configurations X and Y.

Table 3. Results obtained by the Wilcoxon test for configurations X and Y.

VS	R <sup>+</sup>	R <sup>-</sup>	Exact P-value	Asymptotic P-value
X vs Y	63.0	15.0	>= 0.2	0.054614
Y vs X	15.0	63.0	>= 0.2	1

Based on a significance level of 0.05, as the Exact P-values obtained by the test are higher, the null hypothesis cannot be rejected, and a proper distinction of the performances of the configurations is not statistically clear. Ergo, a significant consideration on which configuration is best cannot be made, appraising both configurations as equal in terms of performance.

### 5. DISCUSSION AND CONCLUSIONS

In this paper, a framework to design Evolutionary Spiking Neural Network (ESNNs) was presented. Such framework is based on an Evolutionary Algorithm known as Grammatical Evolution, and is capable of producing SNNs with a wider range of connections types. A comparison against a similar framework generating more traditional networks was made to assess the behaviour of the generated networks. In general, the proposed method was found to match the pattern recognition capabilities of its contender, since accuracy results hardly differ and a statistically trustworthy argument to differentiate the methods was not found.

Unusual topologies with many kinds of connections were obtained; these were found to portrair competitive results in classification tasks, while exhibiting desirable traits such as attribute selection (by skipping connections in some of the input features) or complexity reduction (by bypassing connections between layers).

On the other hand, the SNN model employed (SRM) and the target specification led to network simulations with relatively few spiking activity, rendering recurrent connections likely unexplored or unused. As well, the existence of irrelevant connections for the pattern recognition process (i.e., those synapses not directly connected to the output unit) was not accounted by the evolutionary process, resulting in topologies apt to be "cleaned".

Further development might focus in employing spiking models and/or target specifications exhibiting a greater spiking activity, so the intrinsic potential of the recurrent connections could be assessed. Moreover, proper adaptations to the framework could aid the specification of reservoir computing.

#### ACKNOWLEDGEMENTS

Authors wish to thank *The National Technology of México* and *University of Guanajuato*. G. López-Vázquez and A. Rojas-Domínguez thank to the *National Council of Science and Technology of México* (CONACYT) for the support provided by means of the Scholarship for Postgraduate Studies (701071) and Research Grant (CÁTEDRAS-2598); respectively. A. Espinal wishes to thank *SEP-PRODEP* for the support provided to the Project 511-6/17-8074 "Diseño y entrenamiento de Redes Neuronales Artificiales mediante Algoritmos Evolutivos". H. Rostro-González thanks the *CONACYT* for project FC2016-1961 "Neurociencia Computacional: de la teoría al desarrollo de sistemas neuromórficos".

#### REFERENCES

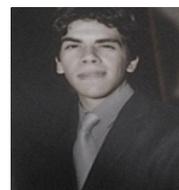
- [1] Guoqiang Peter Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4):451–462, 2000.
- [2] Markos Markou and Sameer Singh. Novelty detection: a review part 2:: neural network based approaches. *Signal processing*, 83(12):2499–2521, 2003.
- [3] Franco Scarselli and Ah Chung Tsoi. Universal approximation using feed-forward neural networks: A survey of some existing methods, and some new results. *Neural networks*, 11(1):15–37, 1998.
- [4] J. Stephen Judd. *Neural Network Design And The Complexity Of Learning. Neural Network Modeling And Connectionism Series*. MIT press, Cambridge (MA), 1990. ISBN 780262100458.
- [5] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.
- [6] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4):115–133, December 1943. ISSN 0007-4985.
- [7] Frank Rosenblatt. *The Perceptron, A Perceiving And Recognizing Automaton (Project PARA)*. Cornell Aeronautical Laboratory, 1957.

- [8] David Elizondo and Emile Fiesler. A survey of partially connected neural networks. *International Journal Of Neural Systems*, 8(5-6):535–558, 1997.
- [9] Wulfram Gerstner and Werner Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002. ISBN 9780521890793.
- [10] Wolfgang Maass and Michael Schmitt. On the complexity of learning for spiking neurons with temporal coding. *Information and Computation*, 153 (1):26–46, 1999.
- [11] Samanwoy Ghosh-Dastidar and Hojjat Adeli. Spiking neural networks. *International journal of neural systems*, 19(04):295–308, 2009.
- [12] Ammar Belatreche. *Biologically Inspired Neural Networks: Models, Learning, and Applications*. VDM Verlag, Saarbrücken, Germany, Germany, 2010. ISBN 9783639228267.
- [13] Edoardo Amaldi, Eddy Mayoraz, and Dominique de Werra. A review of combinatorial problems arising in feedforward neural network design. *Discrete Applied Mathematics*, 52(2):111–138, 1994.
- [14] Xin Yao. Evolving artificial neural networks. *Proceedings Of The IEEE*, 87(9):1423–1447, 1999. ISSN 0018-9219. <https://doi.org/10.1109/5.784219>.
- [15] Varun Kumar Ojha, Ajith Abraham, and Václav Sná~sel. Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, 60:97–116, 2017.
- [16] Ammar Belatreche, Liam P Maguire, Martin Mccinnity, and Qing Xiang Wu. An evolutionary strategy for supervised training of biologically plausible neural networks. In *Proceedings of the Sixth International Conference on Computational Intelligence and Natural Computing*, pages 1524–1527, 2003.
- [17] Altamirano, José S., Manuel Ornelas, Andrés Espinal, Raúl Santiago-Montero, Héctor Puga, Juan Martín Carpio, and Sergio Tostado. "Comparing Evolutionary Strategy Algorithms for Training Spiking Neural Networks." *Research in Computing Science* 96 (2015): 9-17.
- [18] Daniel Rivero, Julian Dorado, Juan Rabuñal, and Alejandro Pazos. Generation and simplification of artificial neural networks by means of genetic programming. *Neurocomputing*, 73(16-18):3200–3223, 2010.
- [19] Olga Quiroz-Ramírez, Andrés Espinal, Manuel Ornelas-Rodríguez, Alfonso Rojas-Domínguez, Daniela Sánchez, Héctor Puga-Soberanes, Martín Carpio, Luis Ernesto Mancilla Espinoza, and Janet Ortíz-López. Partially-connected artificial neural networks developed by grammatical evolution for pattern recognition problems. *Studies in Computational Intelligence*, 749:99–112, 2018. ISSN 1860949X. [https://doi.org/10.1007/978-3-319-71008-2\\_9](https://doi.org/10.1007/978-3-319-71008-2_9).
- [20] Andrés Espinal, Martín Carpio, Manuel Ornelas, Héctor Puga, Patricia Melín, and Marco Sotelo-Figueroa. Developing architectures of spiking neural networks by using grammatical evolution based on evolutionary strategy. In *Mexican Conference on Pattern Recognition*, pages 71–80. Springer, 2014.
- [21] Gustavo López-Vázquez, Manuel Ornelas-Rodríguez, Andrés Espinal, Jorge Soria-Alcaraz, Alfonso Rojas-Domínguez, Héctor Puga-Soberanes, J. Martín Carpio, and Horacio Rostro-González. Evolutionary spiking neural networks for solving supervised classification problems. *Computational Intelligence and Neuroscience*, 2019, "In Press".

- [22] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659 – 1671, 1997. ISSN 0893-6080. [https://doi.org/http://dx.doi.org/10.1016/S0893-6080\(97\)00011-7](https://doi.org/http://dx.doi.org/10.1016/S0893-6080(97)00011-7).
- [23] Shifei Ding, Hui Li, Chunyang Su, Junzhao Yu, and Fengxiang Jin. Evolutionary artificial neural networks: A review. *Artif. Intell. Rev.*, 39(3):251–260, 2013.
- [24] Andrés Espinal, Marco Sotelo-Figueroa, Jorge A. Soria-Alcaraz, Manuel Ornelas, Héctor Puga, Martín Carpio, Rosario Baltazar, and J.L. Rico. Comparison of PSO and DE for training neural networks. 2013 12<sup>th</sup> Mexican International Conference on Artificial Intelligence, 0:83–87, 2011. <https://doi.org/http://doi.ieeecomputersociety.org/10.1109/MICAI.2011.16>.
- [25] Maryam Tayefeh, Fattaneh Taghiyareh, Nafiseh Forouzideh, and Lucas Caro. Evolving artificial neural network structure using grammar encoding and colonial competitive algorithm. *Neural Computing and Applications*, 22 (1):1–16, 2013. ISSN 0941-0643. <https://doi.org/10.1007/s00521-012-0905-6>.
- [26] Heung Bum Kim, Sung Hoon Jung, Tag Gon Kim, and Kyu Ho Park. Fast learning method for back-propagation neural network by evolutionary adaptation of learning rates. *Neurocomputing*, 11(1):101–106, 1996.
- [27] Ioannis Tsoulos, Dimitris Gavrilis, and Euripidis Glavas. Neural network construction and training using grammatical evolution. *Neurocomputing*, 72(1-3):269–277, 2008.
- [28] Ammar Belatreche, Liam P. Maguire, and T. Martin McGinnity. Advances in design and application of spiking neural networks. *Soft Comput.*, 11(3):239–248, 2007.
- [29] M. O’Neill and C. Ryan. Grammatical evolution. *Trans. Evol. Comp*, 5(4):349–358, August 2001. ISSN 1089-778X. <https://doi.org/10.1109/4235.942529>. URL <http://dx.doi.org/10.1109/4235.942529>.
- [30] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. John Wiley & Sons, Hoboken, N.J, 2009. ISBN 978-0-470-27858-1. OCLC: ocn230183356.
- [31] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository. <http://archive.ics.uci.edu/ml>, 2017.

## AUTHOR

**GUSTAVO LÓPEZ-VÁZQUEZ** was born in León, Guanajuato, México, in 1989. He received the B.Sc. degree in computational systems engineering from the León Institute of Technology in 2014. From March to October 2016 he participated in a knowledge co-creation project in the Kanazawa Institute of Technology, Japan with the work Feature Extraction, classification and modeling of music using Deep Learning. He is currently in course to obtain the M.Sc. degree in computational science in the León Institute of Technology. His research interests includes Evolutionary Artificial Neural Networks, Reservoir computing and Deep Learning.



**MANUEL ORNELAS-RODRÍGUEZ** received his B.Sc. degree in electric-mechanic engineering from the Leon Institute of Technology in 1991 and his Ph.D. degree in physics (optics) in 2002 from the University of Guanajuato. Since 2002 he is a researcher at the Leon Institute of Technology in Leon, Guanajuato, Mexico. His research interests include optimization, pattern recognition and computer vision.



**ANDRÉS ESPINAL** received the B.Sc. degree in computer systems engineering and the M.Sc. degree in computer science from the Instituto Tecnológico de León in 2009 and 2011, respectively. And the Ph.D. degree in computer science from the Tijuana Institute of Technology in 2017. He is currently a full-time Professor with the Organizational Studies Department, University of Guanajuato México. His research interest include evolutionary algorithms, artificial neural networks, computer vision, image processing and bio-inspired algorithms.



**JORGE SORIA-ALCARAZ** received the B.Sc. degree in computer systems engineering and the M.Sc. degree in computer science from the Instituto Tecnológico de León in 2008 and 2010, respectively. He has carried out research stays at the Monterrey Institute of Technology and Higher Education in 2009 and at the University of Stirling, U.K. in 2014. He is currently a full-time professor with the Organizational Studies Department, University of Guanajuato. His research interests include evolutionary algorithms, autonomous search, hyperheuristics, adaptive operator selections, heuristic optimization, and bio-inspired algorithms.



**ALFONSO ROJAS-DOMÍNGUEZ** was born in Mexico City, in 1978. He received the BSc deg. in telecommunications engineering from the National Autonomous University of Mexico in 2001; the M.Sc. degree in intelligence engineering in 2003 and the Ph.D. degree in electronics engineering in 2007 from the University of Liverpool, UK. From 2008 to 2014 he was a researcher with recognized institutions, such as the Institute of Biotechnology (IBT-UNAM, '08-'11), the Center for Research in Mathematics (CIMAT, '12-'13), the Center for Research and Advanced Studies (CINVESTAV-IPN, '13-'14). Since 2014 he is a Research Fellow with the National Council of Science and Technology (CONACYT) of Mexico and is currently associated with the Technology Institute of León, Guanajuato, Mexico. His main research interests are machine learning, pattern recognition and computer vision. His current work includes SVMs, Evolutionary algorithms and Deep Learning.



**HÉCTOR J. PUGA-SOBERANES** was born in Mexico City, in 1960. He received the BSc deg. in physics and mathematics from the National Polytechnic Institute in 1993; the M.Sc. degree in physics (Optic) from the University of Guanajuato, México in 1995 and the Ph.D. degree in physics (Optic) in 2002 from the University of Guanajuato, México. Since 2002 he is associated with the research division of the Technology Institute of León, Guanajuato, Mexico. Dr. Puga has been a member of the National Researchers System of the National Council of Science and Technology (SNI-CONACYT) of Mexico since 2004. His main research interests are optic metrology and intelligent systems, in particular, optimization whit heuristics, metaheuristic and hyperheuristics. His current work includes SVMs and Evolutionary algorithms.



metaheuristic and

**JUAN MARTÍN CARPIO VALADEZ** was born León, Gto., México, in 1961. He received the BSc deg. in mathematics from the Autonomous University of Nuevo Leon, México, in 1985; the M.Sc. degree in physics (Optic) from the University of Guanajuato, México in 1987 and the Ph.D. degree in physics (Optic) in 1995 from the University of Guanajuato, México. Since 1994 he is associated with the research division of the Technology Institute of León, Guanajuato, Mexico. Dr. Carpio has been a member of the National Researchers System of the National Council of Science and Technology (SNI-CONACYT) of Mexico since 1992. His main research interests are optic metrology and data fitting with functions and orthogonal polynomials, intelligent systems, in particular, optimization whit heuristics, metaheuristic and hyperheuristics. His current work includes SVMs and Evolutionary algorithms and Grammatical Evolution.



His current work

**HORACIO ROSTRO-GONZÁLEZ** received the B.S. degree in electronic engineering from Instituto Tecnológico de Celaya, México in 2003, the M.E. degree (Hons.) in electrical engineering from the Universidad de Guanajuato, México in 2006, and the D.Sc. degree in computational neuroscience from the University of Nice Sophia Antipolis, France, in 2011. From 2011 to 2012, he was a Research Fellow with the University of Cyprus, where he was involved in the FP7 EU SCANDLE Project. Since 2012, he has been a Full Professor with the Department of Electronics, Universidad de Guanajuato, México. He has authored around 40 journals and conferences proceeding papers. His current research interests include reconfigurable electronic, neuromorphic engineering, computational neuroscience, bio-inspired algorithms, parallel computing, and signal processing.

