Spiking neural network approaches PCA with metaheuristics

J. Enríquez-Gaytán, F. Gómez-Castañeda[✉],
L.M. Flores-Nava and J.A. Moreno-Cadenas

This Letter presents meaningful results that demonstrate the reduction of dimensionality by spiking neural networks (SNNs) on benchmarking data. This experimental scheme includes metaheuristics, namely, the artificial bee colony algorithm (ABC algorithm) for finding optimal conductance values in the SNNs. Therefore, the objective function in the used ABC algorithm leads the SNNs to compute the principal component analysis (PCA), efficiently. The eigendecomposition of the information drawn by the SNNs in the training phase is the base of the formulated objective function. In these experiments, the Izhikevich model represents the spiking neurons, which have biological plausibility with parameters for reproducing a uniform firing rate. The visualisation of clusters in the 3D PCA space, whose sample values are compared with the PCA function in Matlab, is also shown; this comparison demonstrates an acceptable error in the MSE sense.

Introduction: Spiking neural networks (SNNs) represent the third neural model generation. They define both neural and synaptic functions, which are studied as dynamic systems in neuroscience [1] and computing entities in engineering [2]. In principle, the value of the synaptic conductance between neurons in the SNN can modify its magnitude according to the Spike-Timing Dependent Plasticity (STDP) rule [3]. In that biological scenario, the SNN can do correlation tasks over particular spiking patterns and can also perform principal component analysis (PCA) based on extended Hebbian rules [4]. The purpose of this Letter is to demonstrate that a single-layer SNN computes by architecture the first three principal components of the training data and that the artificial bee colony (ABC) algorithm finds the values of the synaptic conductance efficiently in a non-supervised manner. The objective function of the ABC algorithm uses statistical information from the SNN, i.e. the first three proportions of the total variance.

Training SNNs with metaheuristics: There are exploratory works for showing the possibility to train SNNs using metaheuristic methods, some of which are inspired by the collective and intelligent behaviour of specific living beings found in nature. They can combine diversification and intensification in their search for finding optimal solutions, jointly with an objective function. A representative case is given in [5], where the classic ABC algorithm takes the part of one supervised training process. On the other hand, advanced optimisation for designing SNNs based on evolutionary metaheuristics has also been proposed [6]. Multi-layer neural models, but with continuous-sigmoid neurons, may follow metaheuristic strategies [7]. In particular, the classic ABC algorithm achieves efficient optimisation in a broad set of engineering complex problems [8], becoming suitable in this Letter to train SNNs for PCA.

Classic ABC algorithm: It is efficient for solving global optimisation problems related to multi-dimensional functions compared with other metaheuristic approaches. It finds the solution vector $\boldsymbol{x}$ that minimises the objective function $F(\boldsymbol{x})$, where $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$. The values for the elements of $\boldsymbol{x}$ are real numbers and restricted to some interval. The parameters of the classic ABC algorithm in this work are in Table 1.

Table 1: Values of parameters used in the classic ABC algorithm

iteration	1000
bee colony	6
local search abandoning limit	200
upper bounds	10
lower bounds	0

Objective function: The objective function should contain valuable information about the problem, which is subject to the optimisation process by the ABC algorithm. In this Letter, the objective function supports a non-supervised process, where it depends on the proportion of total variance π_k, which measures the quality of the kth principal component [9]. Therefore, it is suitable to choose π_1, π_2, and π_3 as the optimisation parameters, leading to the below objective function:

$$F = 1/\pi_1 + 1/\pi_2 + 1/\pi_3$$

where π_1, π_2, and π_3 are computed as follows:

$$\pi_1 = \lambda_1/(\lambda_1 + \lambda_2 + \lambda_3)$$
$$\pi_2 = \lambda_2/(\lambda_1 + \lambda_2 + \lambda_3)$$
$$\pi_3 = \lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$$

In these equations, λ_1, λ_2, and λ_3 are the first three eigenvalues of the correlation matrix of the data provided by the SSN under training.

The proposed objective function decreases to a minimum until π_1, π_2, and π_3 get their optimal value.

The eigenvalues λ_1, λ_2, and λ_3 are computed in Matlab using the Singular Value Decomposition (SVD) method applied to the matrix of values provided by the SNN. The SVD method establishes the product of three matrices, namely $\boldsymbol{USV}^{\mathrm{T}}$; where $\boldsymbol{U}$ and $\boldsymbol{V}$ are the left and right singular matrices, respectively. $\boldsymbol{S}$ is a diagonal matrix, whose elements are sorted from-high-to-low, i.e. $s_1 > s_2 > \cdots > s_q > 0$.

Finally, the elements: $s_1, s_2, \ldots, s_q$, which are called singular values, are related to λ_1, λ_2, and λ_3 as given by the following equations:

$$\lambda_1 = \frac{s_1^2}{m-1}, \quad \lambda_2 = \frac{s_2^2}{m-1}, \quad \lambda_3 = \frac{s_3^2}{m-1}$$

where m is the number of voltage vectors forming a training matrix.

Experimental SNN: Fig. 1 depicts the SNN in this Letter; there are three neurons: N_1, N_2, and N_3, whose firing rates (in spikes/s) are FR1, FR2, and FR3, respectively. They receive the synaptic currents (in pA): I_1, I_2, and I_3. The voltage vector (in volts) $\boldsymbol{e_i} = [e_{i1}, e_{i2}, e_{i3}, \ldots, e_{i(n-1)}, e_{in}]$ is read via the synapses, which are represented by the conductance vectors (in Ω^{-1}): $\boldsymbol{s}_1 = [s_{11}, s_{12}, \ldots, s_{1n}]$, $\boldsymbol{s}_2 = [s_{21}, s_{22}, \ldots, s_{2n}]$, and $\boldsymbol{s}_3 = [s_{31}, s_{32}, \ldots, s_{3n}]$.

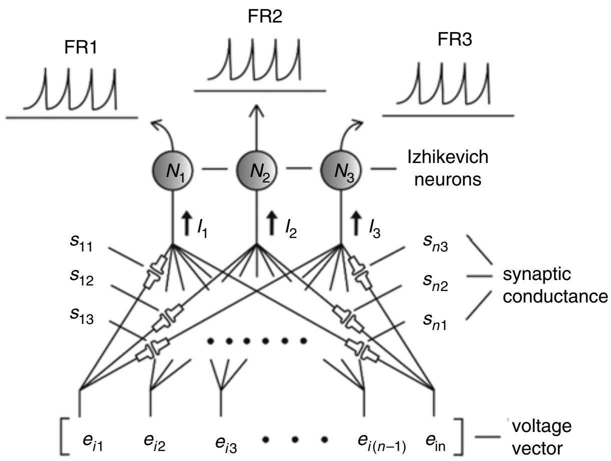

Fig. 1 *Experimental SNN with three Izhikevich neurons: N_1, N_2, and N_3 whose synaptic currents are I_1, I_2, and I_3, respectively*

The whole set of training vectors $\{\boldsymbol{e_i}, i = 1, \ldots, m\}$ defines a training matrix of dimension $m \times n$ with Q classes to be visualised in clusters in the PCA space. The synaptic currents I_1, I_2, and I_3 are given by the dot products: $I_1 = \boldsymbol{e_i}(\boldsymbol{s}_1)^{\mathrm{T}}$, $I_2 = \boldsymbol{e_i}(\boldsymbol{s}_2)^{\mathrm{T}}$, and $I_3 = \boldsymbol{e_i}(\boldsymbol{s}_3)^{\mathrm{T}}$, where $(\,)^{\mathrm{T}}$ denotes transpose.

Model of the Izhikevich neuron: This model is widely accepted for developing neuromorphic systems due to both its biological plausibility and its computational efficiency [1]. This model has the equations below:

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = k(V - V_{\mathrm{rest}})(V - V_{\mathrm{th}}) - U + I_{\mathrm{syn}}$$
$$\frac{\mathrm{d}U}{\mathrm{d}t} = a[b(V - V_{\mathrm{rest}}) - U]$$

where V and U are the membrane potential and the membrane recovery, respectively. The synaptic current I_{syn} fixes the spiking rate. There is a restriction for V, given as if $V \geq 35\,\mathrm{mV}$, then $V = c$ and $U = U + d$.

Table 2 presents the parameters and their corresponding values that were used in the experiments in this Letter for uniform spiking.

Table 2: Values of parameters used in the Izhikevich model

V_{rest}	−60 mV	a	0.03 ms^{-1}
V_{th}	−40 mV	b	−2
V_{peak}	35 mV	c	−50 mV
C	100 pF	d	100 pA

Scheme of the training phase: Fig. 2 shows the numerical processes for one epoch of the training phase, where it starts reading the training matrix and ends changing the conductance values of the SNN.

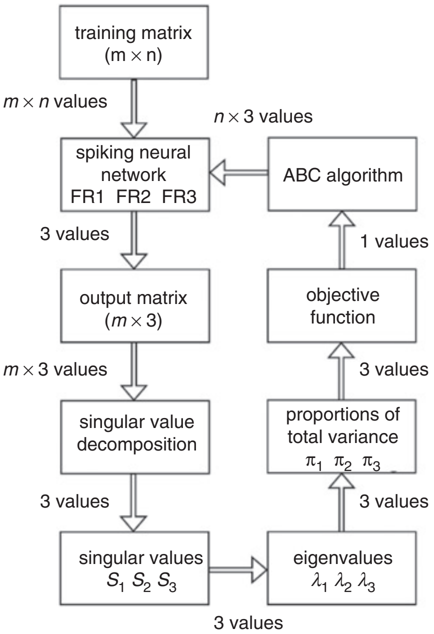

Fig. 2 *Numerical processes in one epoch of the training phase*

Experimental sessions: A visual presentation of the PCA with the first three components PC1, PC2, and PC3 on three types of experimental data demonstrates the effectiveness of the proposed training method. These three experimental sessions use available data sets from [10], namely, gas detection, wines classification, and numeral handwriting. In order to facilitate the training process, the spiking neurons were biased by a constant current of 55 pA, which sets their lowest firing rate at one spike/s. The values of the training data and those in the PCA space were normalised. Table 3 shows the parameters of the training matrices.

Table 3: Dimension of training matrices and number of classes

Session no.	Data set description	m	n	Q
1	gases detection	360	8	3
2	wines classification	900	11	2
3	numeral handwriting	7494	16	10

Session No.1 refers to electrical measurements on oxide sensors to detect dangerous gases, namely carbon monoxide, methane, and ethanol. The sensors transduce concentration (parts-per-million) into resistance (in Ω), whose value is normalised by a reference resistor.

Session No. 2 deals with a collection of red and white wines from 20 industrial manufactures. Both wines were evaluated concerning their quality according to fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulphur dioxide, total sulphur dioxide, density, pH, sulphates, and alcohol.

Session No. 3 takes into account the pixel positions that were activated by 44 individuals writing the numerals from '0' to '9' on a tablet with an array of 500 × 500 pixels.

Results by the SNNs: Figs. 3–5 show in a 3D space the first three components: PC1, PC2, and PC3 that were computed by the SNNs that were trained with the classic ABC algorithm for every experimental session.

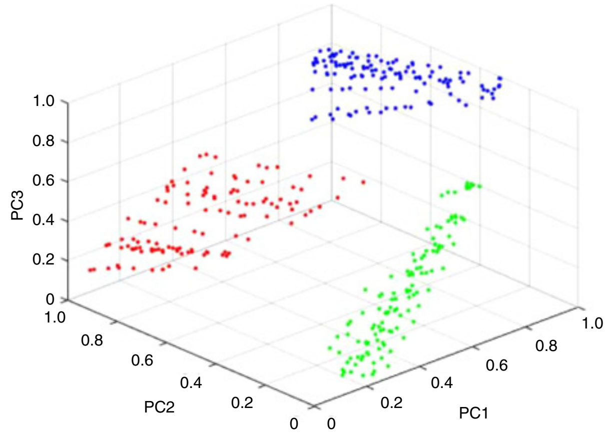

Fig. 3 *PCA space obtained in session No. 1*

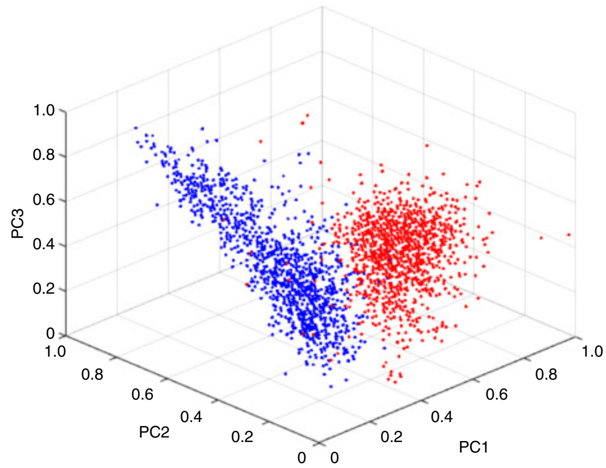

Fig. 4 *PCA space obtained in session No. 2*

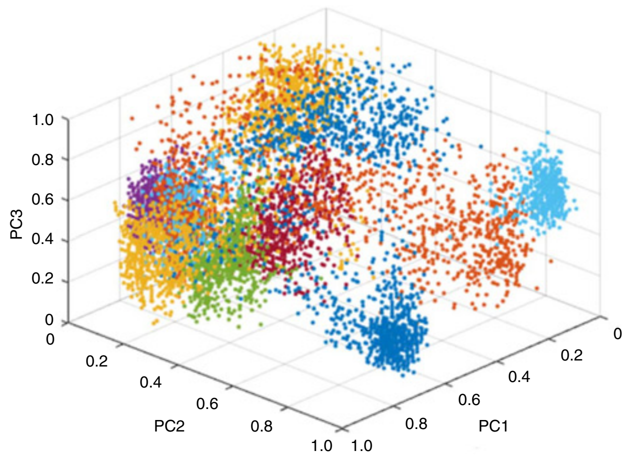

Fig. 5 *PCA space obtained in session No. 3*

For comparing the numerical values of these experimental results with those from the PCA function in Matlab, it was evaluated the mean square error (MSE) score according to the equation below:

$$\text{MSE} = \frac{1}{NS}\sum (EDX_i - EDY_i)^2$$

where EDX_i and EDY_i are the Euclidian distances in both PCA spaces, i.e. those created by the SNN and by Matlab, respectively. EDX_i and EDY_i measure the distance between the centroid of the cluster, and the samples of the same class. The discrete index i counts the samples, and NS is the number of them. The centroids of the clusters were found with the k-means function in Matlab. We computed values of the MSE score for all the experiments, which were always below 0.09; this quantity shows that the different SNN architectures considered, i.e. 8:3, 11:3 and 16:3, approach the PCA space with acceptable accuracy. Likewise, the synaptic conductance values found by the ABC algorithm were positive real numbers in [0, 100], and the number of epochs of the training phase was always lower than 2000 in all the experimental sessions.

Conclusion: This Letter introduced specific SNNs to compute accurate PCA spaces. The proposed metaheuristic training method might help to program neuromorphic hardware for visualising clusters of the same class contained in experimental information of high dimensionality.

Submitted: *20 January 2020* E-first: *28 February 2020*
doi: 10.1049/el.2020.0283
One or more of the Figures in this Letter are available in colour online.

J. Enríquez-Gaytán, F. Gómez-Castañeda, L.M. Flores-Nava and J.A. Moreno-Cadenas (*Department of Electrical Engineering, Cinvestav-IPN, Mexico City, Mexico*)

✉ E-mail: fgomez@cinvestav.mx

References

1 Izhikevich, E.M.: 'Dynamical systems in neuroscience: the geometry of excitability and bursting' (The MIT Press, Cambridge MA, USA, 2007)

2 Schliebs, S., and Kasabov, N.: 'Computational modeling with spiking neural networks', in Kasabov, N. (Ed.): 'Springer handbook of bio-/neuroinformatics' (Springer-Verlag, Berlin & Heidelberg, 2014), pp. 625–646

3 Sjöström, J., and Gerstner, W.: 'Spike-timing dependent plasticity', https://icwww.epfl.ch/~gerstner/PUBLICATIONS/STDP-Scholarpedia2010.pdf, accessed September 2018

4 Gerstner, W.: 'Hebbian learning and plasticity', https://pdfs.semanticscholar.org/f9fc/99a5c52aa5df1b530dfdeb25dfb6b10bdecf.pdf, accessed November 2018

5 Vazquez, R.A., and Garro, B.A.: 'Training spiking neural models using artificial bee colony', *Comput. Intell. Neurosci.*, 2015, **2015**, pp. 1–14

6 López-Vázquez, G., Ornelas-Rodriguez, M., Espinal, A., *et al.*: 'Evolutionary spiking neural networks for solving supervised classification problems', *Comput. Intell. Neurosci.*, 2019, **2019**, pp. 1–13

7 Ojha, V.K., Abraham, A., and Snasel, V.: 'Metaheuristic design of feedforward neural networks: a review of two decades of research', *Eng. Appl. Artif. Intell.*, 2017, **60**, pp. 97–116

8 Karaboga, D., and Basturk, B.: 'On the performance of artificial bee colony (abc) algorithm', *Appl. Soft Comput.*, 2008, **8**, pp. 687–697

9 Jolliffe, I.T., and Cadima, J.: 'Principal component analysis: a review and recent developments', *Phil. Trans. R. Soc.*, 2016, **374**, (2064), pp. 1–16, http://dx.doi.org/10.1098/rsta.2015.0202

10 Center for Machine Learning and Intelligent Systems, University of California, Irvine, USA, http://archive.ics.uci.edu/ml/datasets/Twin+gas+sensor+arrays, accessed January 2019, https://archive.ics.uci.edu/ml/datasets/wine+quality, accessed April 2019, https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits, accessed June 2019