Red Pulsante con Aprendizaje Hebbiano para Clasificación de Patrones Ralos

Iván Peralta[1*], José T. Molas[1], César E. Martínez[1,2], and Hugo L. Rufiner[1,2,3]

[1]Laboratorio de Cibernética, Facultad de Ingeniería,
Universidad Nacional de Entre Ríos,
CC 47 Suc. 3, E3100, Ruta 11, km. 10, Oro Verde, Entre Ríos, Argentina
[2]Centro de I+D en Señales, Sistemas e Inteligencia Computacional (SINC(i)),
Facultad de Ingeniería y Cs. Hídricas, Universidad Nacional del Litoral
[3]CONICET, Argentina
[*]bioivanperalta@gmail.com

Resumen En las últimas décadas se ha intentado desarrollar Redes Neuronales Artificiales más realístas que intenten imitar con mayor precisión el funcionamiento de sus contrapartes biológicas. Es así como nacieron las Redes Neuronales Pulsantes. Uno de los principales usos de estas redes es la clasificación de patrones. Sin embargo su aplicabilidad en el mundo real ha sido limitada debido a la falta de métodos de entrenamiento eficientes. En este trabajo se presenta un nuevo modelo de red pulsante pensado para clasificar patrones ralos. El mismo puede entrenarse mediante reglas de aprendizaje hebbiano no supervisado. Se describe su estructura, funcionamiento y el algoritmo propuesto para su entrenamiento. Además, se reportan resultados de prueba con patrones generados artificialmente y se discute la factibilidad de su implementación en un dispositivo lógico programable tipo FPGA.

Keywords: neurona pulsante, red neuronal pulsante, algoritmo de aprendizaje, aprendizaje no supervisado, patrones ralos.

1. Introducción

Las redes neuronales artificiales, inspiradas por las neuronas biológicas, están compuestas por unidades básicas denominadas neuronas. Estas se encuentran interconectadas mediante distintos pesos que determinan la intensidad con que interactúan dichas neuronas. La búsqueda de una analogía más cercana a la realidad biológica ha dado lugar en las últimas dos décadas a la aparición de las denominadas *Redes Neuronales Pulsantes* (SNN: *Spiking Neural Networks*). El uso de las SNN se encuentra en crecimiento debido a su habilidad para afrontar diferentes problemas en distintas áreas tales como clasificación de patrones, control maquinal, procesamiento de imágenes, etc. Estas redes reproducen más fielmente los sistemas neuronales biológicos tratando, por un lado, de imitar la transferencia de información entre neuronas a través de pulsos tal como se realizan en las sinapsis biológicas con los Potenciales de Acción, como así también, el procesamiento dinámico de las señales dentro de las neuronas.

Se han desarrollado innumerables trabajos que utilizan las SNN en diferentes aplicaciones. En [2] se describe un ejemplo de aplicación en ingeniería biomédica en donde se analizan tres algoritmos de entrenamiento de una SNN para la detección de epilepsia y convulsiones a través de la clasificación de patrones de EEG el cual es un problema de reconocimiento de patrones complicado, en [3] se presenta MuSpiNN, otro modelo de SNN, para afrontar el problema anterior. También se han utilizado en sistemas de control, en [12] se utiliza una SNN para controlar los movimientos de un robot para evitar obstáculos mediante señales ultrasónicas. Más recientemente en [1] se diseña una red capaz de memorizar secuencias de eventos y en [6] se analiza una SNN dinámica para el reconocimiento de patrones espacio/espectro temporales. A su vez se ha demostrado que este tipo de redes puede mapearse con mayor facilidad que las redes tradicionales dentro de dispositivos lógicos programables tipo FPGA [9,10].

Este trabajo fue motivado por la necesidad de desarrollar un clasificador eficiente para un tipo especial de patrones, originados a partir de *representaciones ralas* de señales de interés. Este tipo de representaciones o códigos surgen al analizar las señales mediante diccionarios discretos que utilizan una gran cantidad de átomos, pero que describen cada una de ellas en términos de una pequeña fracción de estos elementos [8]. Así, la codificación lograda posee la mayoría de sus coeficientes igualados a cero (o casi cero) [5]. Dado que este trabajo se enfoca en el entrenamiento de una red pulsante para el reconocimiento de patrones ralos, sin importar el método para obtener dicha representación, se utilizarán inicialmente patrones ralos artificiales de tipo binario, generados en forma aleatoria para cada clase.

El resto del trabajo se organiza como se detalla a continuación. La Sección 2 introduce la estructura de la SNN, los modelos neuronales y su funcionamiento interno. La Sección 3 describe la codificación rala y las reglas de entrenamiento de la SNN. La Sección 4 describe el método, las condiciones experimentales de prueba de la red, los resultados obtenidos y una discusión sobre la factibilidad de reproducir este modelo de SNN en un dispositivo lógico programable tipo FPGA. Finalmente, la Sección 5 resume las conclusiones y posibles líneas de trabajo futuros.

2. Estructura de la SNN

2.1. Conexiones y tipos de neuronas

La estructura de la SNN consiste de dos capas de neuronas: una Capa Detectora de entrada y una Capa Integradora de salida. La primera capa se conecta por un lado con el vector patrón de entrada ν, el cual es un vector en $\mathbb{B}^N$ donde $\mathbb{B}$ es el conjunto $\{0, 1\}$, y por otro lado con la capa integradora. La cantidad de neuronas integradoras $\mathbf{I}$ es igual a la cantidad de detectoras $\mathbf{D}$ y se tienen tantas neuronas integradoras como clases de patrones $\mathbf{C}$ se deseen clasificar, es decir, $\mathbf{I} = \mathbf{D} = \mathbf{C}$. En la Figura 1 se presenta un esquema de la SNN.

Entre cada coeficiente ν_n del vector patrón ν y cada neurona d de la capa detectora existe un peso de interconexión W_{nd} que pondera que tan importante

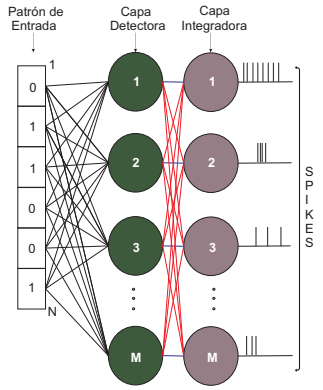

Figura 1. Diagrama estructural de la SNN, con N coeficientes para el vector patrón de entrada, M neuronas detectoras y M neuronas integradoras.

es la presencia de un "1" (uno) en dicho coeficiente para la clasificación del patrón por parte de la SNN. Cuanto mayor es el peso de interconexión, mayor importancia tendrá ese coeficiente para la clasificación del patrón dentro de alguna de las clases existentes. Si el coeficiente posee un valor "0" (cero) el mismo no estimula la neurona en la clasificación de ese patrón en particular. De la misma manera, existen pesos de interconexión entre la capa detectora y la capa integradora pero en este caso se diferencian dos tipos de conexiones: excitatorias –color azul– e inhibitorias –color rojo–. Cada neurona detectora excita a su correspondiente integradora a través de un peso positivo (W_{di} con $i = d$) e inhibe al resto mediante una conexión con pesos negativos (W_{di} con $i \neq d$). La comunicación entre las capas, así como también el vector de entrada con la capa detectora, se produce mediante spikes.

Cada neurona que integra la capa detectora recibe N conexiones; una por cada coeficiente del vector de entrada ν. Una neurona d de esta capa posee N registros internos correspondientes a los coeficientes v_n los cuales se denominan R_{nd}, a su vez posee un registro S_d que almacena la suma de los registros anteriores. Inicialmente todos estos registros se encuentran con valor cero.

El funcionamiento de la unidad esta temporizado en forma discreta con el tiempo k. Si en el instante k_0 se le presenta a la neurona detectora un vector patrón $\nu^{(0)}$, cada registro interno R_{nd} asociado a un coeficiente $\nu_n = 1$ se incrementará en el tiempo con una recta de pendiente igual al peso de interconexión entre el registro y el coeficiente. Este incremento en el registro se hará hasta llegar a un número preestablecido de iteraciones de subida k_f, luego cada registro regresará a cero en la iteración siguiente para esperar el próximo patrón de entrada $\nu^{(1)}$, a su vez antes de la presentación del próximo vector patrón se deja una iteración extra de espera. Por lo tanto, se presentará un vector patrón a la SNN cada $k_f + 2$ iteraciones y durante el período de tiempo entre una pre-

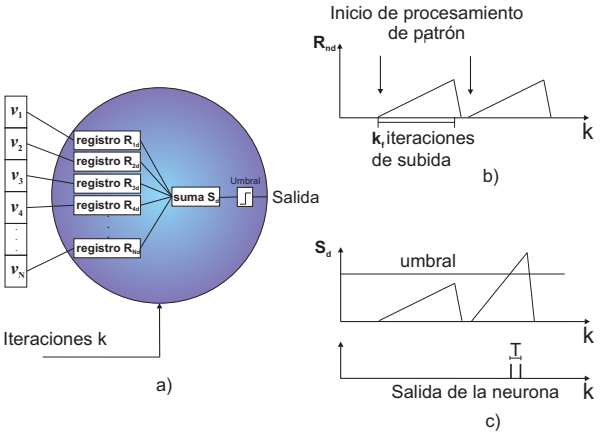

Figura 2. Diagrama estructural y funcional de la neurona detectora. a) representación de la estructura interna de la neurona. b) evolución de los registros una vez presentado un patrón con un $\nu = 1$ en el coeficiente correspondiente a ese registro. c) registro suma junto con el umbral, spikes a la salida y período refractario

sentación de un patrón y la siguiente, la neurona procesa dicha entrada para emitir pulsos ("spikes") en su salida si el registro S_d supera un cierto umbral. Una vez que la neurona emitió un pulso, existe un período refractario T en el que no se emiten pulsos por más que S_d se encuentre por encima del umbral. En la Figura 2 se resume la estructura y funcionamiento de la neurona detectora.

2.2. Neurona integradora

El modelo de neurona integradora propuesta para este trabajo posee una estructura muy similar a la neurona detectora, pero su funcionamiento interno es diferente. En la Figura 3 se describe el funcionamiento para la activación de varias de sus entradas. Cuando llega un pulso de alguna de las entradas, su registro R_{di} correspondiente se incrementará en el tiempo k con una recta de pendiente igual al peso de interconexión W_{di} entre estas neuronas, si $i = d$ el peso es positivo (conexión excitatoria) y si $i \neq d$ el peso es negativo (conexión inhibitoria). A diferencia del funcionamiento de la neurona detectora, este incremento en el registro se realiza sólo en una iteración, luego cada registro se acercará a cero pero con una pendiente W'_{di} inferior a W_{di} hasta recibir otro pulso en su entrada para volver a incrementarse. El registro S_i almacena la suma de todos los registros de esa neurona y si dicho registro supera un cierto umbral se emite un pulso en su salida. Al igual que en la neurona detectora, una vez que ésta emitió un pulso, existe un período refractario T en el que no se emiten pulsos por más que S_i se encuentre por encima del umbral. El valor del período refractario es el mismo para todas las neuronas integradoras pero es diferente del que poseen las neuronas detectoras.

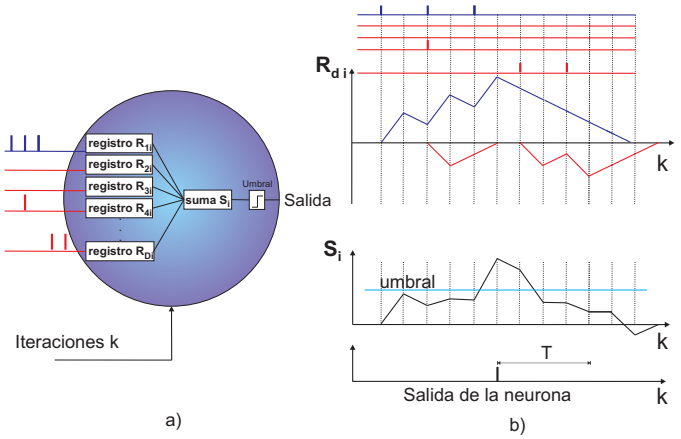

Figura 3. Diagrama Estructural y funcional de la neurona integradora. a) representación de la estructura interna de la neurona. b) integración de pulsos.

3. Algoritmo de aprendizaje

El método de aprendizaje de este trabajo se basa en la idea de Hebb [4]. El entrenamiento se realiza sobre las conexiones entre la capa detectora y los patrones de entrada, es decir, se determina el valor de los pesos W_{nd} descriptos en la Sección 2.1. Si se tiene una cantidad CE de vectores de cada clase para armar los patrones de entrenamiento y se denomina M_d a la matriz binaria de CE columnas y N filas formada por los patrones de entrenamiento de la clase d (recordar que $D = C$) en donde cada patrón forma un vector columna de la matriz. Si se denomina $Md(n, k)$ al n-ésimo coeficiente de la k-ésima columna de la matriz, entonces se puede describir la obtención de los pesos como:

$$W_{nd} = \begin{cases} \sum_{k=1}^{CE} M_d(n,k) & \text{si } \sum_{k=1}^{CE} M_d(n,k) > 0 \\ -\alpha \max_n \left\{ \sum_{k=1}^{CE} M_d(n,d) \right\} & \text{si } \sum_{k=1}^{CE} M_d(n,k) = 0 \end{cases} \quad (1)$$

donde α es una constante entera positiva.

La interpretación de esta regla es sencilla: cuanto más se active un coeficiente en los patrones de entrenamiento, mayor peso tendrá su conexión correspondiente. Si un coeficiente nunca se activó en el entrenamiento, entonces se le asigna un peso igual al negativo del máximo peso para esa clase escalado en un valor α. Esto hace que durante la evaluación cuando se presente un patrón cuyos coeficientes con valor 1 coincidan en su mayoría con las activaciones más frecuentes ocurridas durante el entrenamiento para alguna de las clases, el registro suma

de la neurona detectora correspondiente a esa clase supere el umbral y el patrón sea detectado.

3.1. Función de las neuronas integradoras

La neurona detectora emitirá mayor cantidad de pulsos si el patrón que está analizando corresponde con la clase de dicha neurona, dado que alcanzará su umbral más rápidamente que en caso contrario. La función de la neurona integradora consiste en integrar los pulsos provenientes de la detectora, esta integración maximizará el registro S_i en el caso de que el patrón pertenezca a la misma clase de este par de neuronas.

A menudo en la clasificación de patrones es necesario tener en cuenta la historia de los patrones que se van presentando a la red a lo largo del tiempo, un ejemplo de este tipo de clasificación es cuando la SNN está destinada a detectar patrones pertenecientes a clases que simulan distintos fonemas los cuales pueden abarcar varios patrones o solo uno [11]. Otra función de la neurona integradora consiste en recordar la clase de los patrones anteriores al analizado actualmente. Cuanto más leve sea la pendiente de decaimiento W'_{di}, mayor influencia tendrá el actual patrón en la detección de los siguientes pero si W'_{di} es muy pequeña, los registros S_i tardarán varias iteraciones en regresar a 0 y producirán pulsos que interferirán en la detección de patrones de otras clases. En el caso de la Figura 3, luego de los tres primeros pulsos de la entrada excitatoria vienen dos pulsos de una de las entradas inhibitorias, lo que supone que primeramente se presentó un patrón perteneciente a la misma clase de esa neurona y luego otro de una clase diferente. Si el valor de W'_{di} fuese más pequeño, entonces el valor del registro excitatorio permanecería elevado y podría generar pulsos a la salida e interferir en la detección del patrón de la segunda clase. Los pesos W_{di} negativos aumentan la especificidad de la detección (envío de pulsos inhibitorios hacia neuronas vecinas).

3.2. Determinación de los umbrales

Un punto importante es la determinación de los umbrales de los dos tipos de neuronas, sus valores no pueden ser muy elevados dado que no se tendría sensibilidad en la detección de los patrones. Por otro lado, si son muy pequeños se perdería especificidad entre las distintas clases de patrones. Cada neurona detectora posee un umbral proporcional al máximo valor alcanzado por sus registros R_{nd} durante todo el entrenamiento:

$$umbral_d = \beta k_f \max_n \{W_{nd}\} \tag{2}$$

donde $\beta < 1$ es la constante de proporcionalidad.

Cuanto mayor sean las iteraciones de subida k_f de una neurona detectora, mayor cantidad de pulsos excitarán la neurona integradora correspondiente y mayor el valor de sus registros internos. Por lo tanto, se fijó el umbral para estas neuronas como una proporción de k_f y es el mismo para todas las neuronas:

$$umbral_d = \gamma k_f \tag{3}$$

donde $\gamma > 1$ es la constante de proporcionalidad.

4. Experimentos y resultados

Como se mencionó anteriormente, en este trabajo no se aborda el problema de la obtención de los patrones ralos a partir de señales reales, sino que estos son generados en forma aleatoria y utilizados para entrenar y evaluar la SNN. Esto significa que se necesitan dos conjuntos de patrones: entrenamiento y evaluación. A su vez tendremos C clases o grupos de patrones. Los patrones utilizados se generaron artificialmente a partir del siguiente algoritmo:

1. Inicialmente se genera un conjunto de vectores binarios en $\mathbb{B}^N$ donde $\mathbb{B}$ es el conjunto $\{0, 1\}$ y N es un entero. Este inicio se hace favoreciendo la aparición de ceros respecto a la aparición de unos en el vector, esto lo hace un patrón ralo.
2. Se quitan aquellos vectores que posean todos sus elementos nulos.
3. Se aplica el algoritmo de *K-Means* para agrupar los vectores en C clases distintas. Este método permite agrupar vectores en distintas clases buscando el grado de parecido entre dichos vectores [7].
4. Si la cantidad de patrones que obtuvo la clase con menos elementos, es inferior a la cantidad de los patrones de entrenamiento y evaluación necesarios: se vuelve al paso 1 con mayor cantidad de vectores iniciales.
5. Se toman CE vectores de cada clase para armar los patrones de entrenamiento y CT vectores de cada clase para armar los patrones evaluación.

Se efectuaron 30 realizaciones del mismo experimento para obtener un resultado promedio de la tasa de reconocimiento global. En la Figura 4 se muestra un conjunto de patrones de 60 coeficientes utilizados para una de las realizaciones.

Con el fin de evaluar el desempeño de la capa integradora, en cada experimento se introducen los patrones para prueba con distintas modalidades. Por modalidad se entiende la cantidad de patrones de una misma clase que se introducen en forma consecutiva, siendo variados entre 1 y 5. En el primer caso los patrones son ubicados en forma alternada de manera que no pueda entrar un patrón de la misma clase que el patrón que ingresó anteriormente, para este caso el efecto de integración no será realizado. En el último caso, la introducción de los patrones se hace con menor mezcla de las clases, es decir, primero se introducen 5 patrones de una clase, luego se introducen los 5 patrones de la otra clase y así hasta terminar de introducir la totalidad de los patrones de prueba, esperando una integración máxima. Dado que se están utilizando patrones aleatorios, no se intentó un ajuste preciso de los parámetros de la SNN. La determinación se hizo mediante experimentación y no se utilizaron técnicas para la optimización de dichos parámetros. En la Tabla 1 se detallan los parámetros utilizados.

En la Figura 5 se muestran las tasas de reconocimiento para cada realización y la tasa de reconocimiento promedio para todo el experimento, en función de

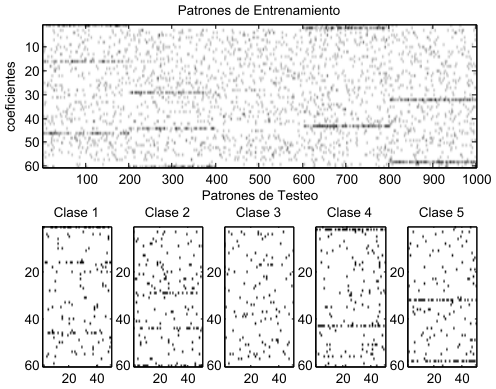

Figura 4. Patrones utilizados en una de las realizaciones del entrenamiento y prueba de la red. Arriba: 200 patrones de entrenamiento para cada clase. Abajo: 50 patrones de prueba para cada clase.

Tabla 1. Parámetros utilizados para la prueba de la SNN.

Referencia	Valor	Descripción
C	5	Cantidad de clases de patrones
D	5	Cantidad de neuronas de la capa detectora
I	5	Cantidad de neuronas de la capa integradora
k_f	8	Iteraciones de subida de la neurona detectora
T detectora	1	Periodo refractario de la neurona detectora
T integradora	4	Periodo refractario de la neurona integradora
W_{di}	16 (i == d)	Peso de interconexión entre capa detectora e integradora
	-13(i <> d)	
W'_{di}	5	Pendiente de descenso de los registros de la n. integradora
CE	200	Cantidad de patrones de entrenamiento por clase
CT	50	Cantidad de patrones de testeo por clase
α	4	Determinación de pesos neurona detectora
β	0,33	Determinación de umbral de neurona detectora
γ	1,5	Determinación de umbral de neurona integradora

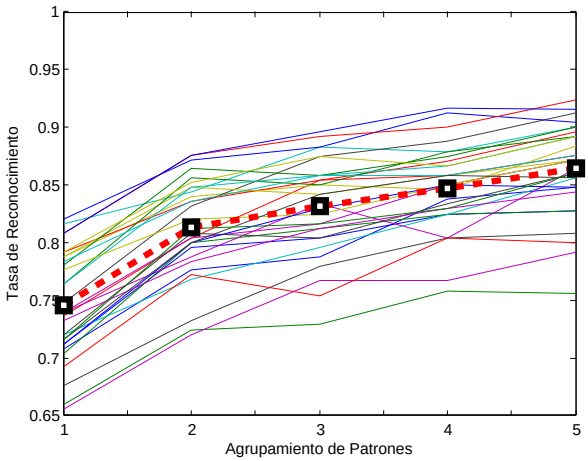

Figura 5. Tasas de reconocimiento en función de las 5 modalidades. En línea llena delgada: tasa de reconocimiento global de patrones para cada realización. En línea de trazos: tasa de reconocimiento promedio.

las 5 modalidades. Se aprecia una tasa de reconocimiento mínima superior al 65 % y máxima cercana al 83 % para la modalidad 1; mientras que la tasa de reconocimiento global promedio de todas las realizaciones se encuentra cercana al 75 %. A medida que aumenta la modalidad, las tasas de reconocimiento aumentan debido al funcionamiento de la capa integradora.

La utilización de pesos de interconexión entre capas de tipo enteros, las curvas de variación de tipo rectas así como la estructura basada en registros de esta SNN y la mayoría de los parámetros enteros, vuelven este diseño muy factible para mapearse en dispositivos electrónicos lógicos programables tipo FPGA con mínimos cambios. En [10] se ha desarrollado una SNN que posee las características mencionadas en una FPGA.

5. Conclusiones

En este trabajo se ha presentado el diseño e implementación de una Red Neuronal Pulsante, junto a un algoritmo de entrenamiento que permiten el reconocimiento de patrones ralos.

El desempeño obtenido es satisfactorio, sobre todo cuando se utilizan las capacidades de integración de estas redes mediante la presentación de patrones en forma consecutiva. Los resultados son bastante alentadores para la aplicación la cual fue motivación del presente trabajo.

Como trabajos futuros se puede mencionar que, con mínimos cambios, es posible trasladar la implementación lograda a un dispositivo portátil tipo FPGA.

Agradecimientos

Los autores desean agradecer a la *Agencia Nacional de Promoción Científica y Tecnológica* (bajo proyecto PAE 37122), la *Universidad Nacional de Entre Ríos* (PID NOVEL 6121), la *Universidad Nacional de Litoral* (PACT 2011 #58, CAI+D 2011 #58-511, #58-525), y al *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET).

Referencias

1. Borisyuk, R., Chik, D., Kazanovich, Y., Gomes, J.d.S.: Spiking neural network model for memorizing sequences with forward and backward recall. Biosystems (2013)
2. Ghosh-Dastidar, S., Adeli, H.: Improved spiking neural networks for EEG classification and epilepsy and seizure detection. Integrated Computer-Aided Engineering 14(3), 187–212 (2007)
3. Ghosh-Dastidar, S., Adeli, H.: A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. Neural Networks 22(10), 1419–1431 (2009)
4. Hebb, D.O.: The organization of behavior: A neuropsychological approach. John Wiley & Sons (1949)
5. Hyvärinen, A.: Sparse code shrinkage: Denoising of nongaussian data by maximum-likelihood estimation. Tech. rep., Helsinki University of Technology (1998)
6. Kasabov, N., Dhoble, K., Nuntalid, N., Indiveri, G.: Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition. Neural Networks (2012)
7. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, p. 14 (1967)
8. Olshausen, B., Field, D.: Emergence of simple cell receptive field properties by learning a sparse code for natural images. Nature 381, 607–609 (1996)
9. Pearson, M.J., Melhuish, C., Pipe, A.G., Nibouche, M., Gilhesphy, L., Gurney, K., Mitchinson, B.: Design and FPGA implementation of an embedded real-time biologically plausible spiking neural network processor. In: Field Programmable Logic and Applications, 2005. International Conference on. pp. 582–585 (2005)
10. Peralta, I., Molas, J.T., Martínez, C.E., Rufiner, H.L.: Implementación de una red neuronal pulsante parametrizable en FPGA. Anales de la XIV Reunión de Procesamiento de la Información y Control (2011)
11. Rufiner, H.L.: Análisis y modelado digital de la voz: técnicas recientes y aplicaciones. Ediciones UNL, Colección Ciencia y Técnica 284 (2009)
12. Wang, X., Hou, Z.G., Zou, A., Tan, M., Cheng, L.: A behavior controller based on spiking neural networks for mobile robots. Neurocomputing 71(4), 655–666 (2008)