

# Relaciones entre dos modelos de computación natural

Vázquez, Juan Carlos / Constable, Leticia Edith  
*Universidad Tecnológica Nacional, Facultad Regional Córdoba*  
*Departamento de Ingeniería en Sistemas de Información*

## Abstract

*Durante diez años, varios proyectos relacionados se han interesado en modelos de computación no convencionales, en sus fundamentos, aplicaciones y posibles relaciones. En particular este artículo describe dos de ellos (redes neurales artificiales y autómatas celulares unidimensionales) y discute las relaciones encontradas. Partiendo de una mirada distinta, se piensa a las redes neurales artificiales como sistemas evolutivos que se trata de emparentar con el modo en que los autómatas celulares van cambiando en el tiempo. Mediante herramientas de software construidas para implementar los modelos, se procedió por experimentación computacional, se registraron los resultados obtenidos de las corridas en diversos formatos y, mediante el análisis de los mismos, se descubrió cierta relación para casos específicos. En el proceso, se ha apoyado la formación de una interesante cantidad de recursos humanos y los productos generados han servido de aplicación en diversos campos, dando lugar a transferencias y nuevos proyectos.*

## Palabras Clave

Redes neurales artificiales, autómatas celulares, computación natural.

## Introducción

Desde hace ya algunos años se está dando el nombre de *computación natural* a una gran cantidad de modelos de computación no convencionales inspirados en ejemplos de la naturaleza.

La idea que subyace en estos modelos, es la de imitar el modo en el cual la naturaleza ha dado soluciones a sus problemas durante milenios, ya sea aplicando sus estrategias (selección natural principalmente [1], pero también clonación vs reproducción sexual, codificación genética de información, etc.) o utilizando como modelo a los organismos que surgieron a través de la evolución (sistema nervioso, sistema inmune y otros) y hasta en algunos casos, a sus modos de interactuar y organizarse en comunidades

(colonias de bacterias, de hormigas, modos de propagación de plagas o de una especie vegetal).

Resumidamente, su método general es al parecer el de adaptación por prueba y error, logrando generación tras generación, época tras época, el éxito de los que mejor se adaptan al contexto, tomando como éxito a la sustentabilidad en el tiempo de los beneficios logrados. En ingeniería esto es un proceso de optimización.

Los ingenieros y científicos, atacan los problemas de la realidad que tienen entre manos construyendo modelos. Para ello, los sistemas bajo estudio son descritos con ecuaciones matemáticas (ya sean lineales, no lineales, diferenciales, integrales, de estructura, etc.) o mediante el diseño de dispositivos abstractos (máquinas de estado finito, autómatas, pseudolenguajes, grafos, croquis y otros); así, pueden interrogar luego al modelo en una primera instancia, para validarlo al contrastar sus respuestas contra hechos conocidos de la realidad y luego, para simular nuevas situaciones en un intento comprender, explicar y predecir comportamientos del sistema, que podrá ser después eventualmente controlado.

Las redes neurales artificiales (RNA), los algoritmos genéticos, la programación evolutiva, los autómatas celulares (AC) y otros muchos dispositivos computacionales no convencionales, implementan estrategias diseñadas por la naturaleza en su evolución (o sus resultados), para modelar sistemas discretos en los que se estima que su complejidad, no linealidad e incertidumbre son esenciales, por lo que un modelo matemático exacto o es muy complicado de ser construido, o sencillamente es imposible hacerlo.

El proyecto *RNA-AC*, intenta establecer una relación esencial entre distintos abordajes. El enfoque utilizado es pensar en las redes neurales artificiales como sistemas que durante su aprendizaje, evolucionan algún autómata celular y a los autómatas celulares en su evolución espacio-temporal, como cálculos aproximados de algún sistema de ecuaciones diferenciales que generan como resultados pesos sinápticos de una red neural entrenada. Se propone determinar un fenotipo común (comportamiento), para luego desarrollar un genotipo (explicación formal) que lo explique y lo ponga de manifiesto. Esto puede redundar en la aplicación de las mejores herramientas de modelado para cada problema, dejando la optimización de su implementación a un proceso automático de conversión.

## Elementos del Trabajo y metodología

### Autómata Celular Unidimensional

La idea de los autómatas celulares nace con los trabajos de John Von Neumann a finales de la década de 1940, en un intento de formalizar el concepto de máquinas auto-replicas; este trabajo de Von Neumann quedó trunco y fue continuado luego de su muerte por A. W. Burks [2]. Si bien el trabajo de estos matemáticos se refería a arreglos bidimensionales de celdas, que J. H. Conway popularizó a principios de los años 1970 con su *Juego de la Vida*, en la década de 1980 Stephen Wolfram realizó una extensa investigación y discusión de los autómatas celulares unidimensionales, abarcándolos desde variados enfoques [3].

El autómata celular es un modelo dinámico donde el tiempo y el espacio se consideran discretos. En su versión teórica más simple el autómata celular, consiste de un arreglo unidimensional infinito de celdas contiguas que, sin pérdida de generalidad, se puede suponer que contienen cada una un valor entero entre  $0$  y  $k-1$  para  $k \in \mathbb{Z}^+$ ,  $k \geq 2$ .

En el caso  $k=2$ , el alfabeto de símbolos para

t	...	0	0	1	0	0	...
---	-----	---	---	---	---	---	-----

Fig. 1: Autómata celular unidimensional elemental.

una celda es el binario  $\{0, 1\}$  y el autómata celular obtenido se denomina *elemental*.

Los valores de las celdas evolucionan en etapas de tiempo discreto, usualmente de acuerdo a una regla sencilla idéntica para todas las celdas del autómata, comúnmente una expresión booleana en autómatas binarios u otra bajo aritmética módulo  $k$  para los no binarios; esta función (en la bibliografía es usual presentar autómatas deterministas), involucra solamente el valor contenido en la celda misma y en sus  $r$  vecinas inmediatas a ambos lados, en el instante anterior;  $r$  recibe el nombre de *rango de la regla* y al sub-arreglo formado por la celda y sus  $2r$  vecinas se lo denomina una *r-vecindad*.

t	...	0	0	1	0	0	...
t+1	...	0	1	0	1	0	...
t+2	...	1	0	1	0	1	...

$$a_i^{(t+1)} = a_{i-1}^{(t)} \vee a_{i+1}^{(t)}$$

Fig. 2: Autómata celular elemental regido por el operador *o inclusivo*, evolucionando.

La figura 2 muestra el autómata celular elemental ( $k=2$ ,  $r=1$ ) de la figura 1, su regla de evolución es la disyunción lógica de los valores de la celda inmediata anterior e inmediata posterior a la celda en proceso y el rango de la regla es  $1$ . El subíndice de la notación corresponde a la posición de la celda dentro del arreglo y el superíndice entre paréntesis indica la etapa del valor accedido. Nótese que para celdas binarias (sólo valores cero o uno) y una 1-vecindad (tres celdas), pueden establecerse 256 reglas distintas (ocho valores asignados a cada una

t	000	001	010	011	100	101	110	111
t+1	0	1	0	1	1	1	1	1

Fig. 3: Formato binario de la regla de disyunción.

de las posibles ocho configuraciones de tres ceros y unos, figura 3).

El arreglo se supone infinito, por lo que a los fines prácticos debe elegirse un número entero positivo  $N$  de celdas e imponerse una condición de contorno a las celdas extremas del arreglo finito, la cual puede consistir en

mantenerlas en un valor fijo (en la figura 2 se asumió 0) o en tomar el arreglo como circular, siendo la primer celda del arreglo la siguiente de la última y viceversa.

Así presentado, el arreglo del AC contiene en cada instante  $t$  una cadena binaria de largo  $N$  ( $\alpha_t \in \{0,1\}^N$ ) llamada *configuración*, que cambia en intervalos de tiempo discreto por lo que puede pensarse al AC como una implementación de una función:

$$\varphi: \{0,1\}^N \rightarrow \{0,1\}^N$$

Dada una configuración inicial arbitraria  $\alpha_0$ , a medida que transcurre el tiempo se genera una sucesión de configuraciones  $\alpha_0, \alpha_1, \dots, \alpha_t, \dots$  que se denomina *evolución espacio-temporal* del autómata celular.

Una de las características de interés de los AC, es el patrón de evolución espacio-temporal que desarrollan; a partir de una configuración inicial aleatoria se generan estructuras organizadas, contradiciendo la segunda ley de la termodinámica, ya que disminuye su entropía con el tiempo; si bien esto no es un sistema físico sino abstracto, esta aparente contradicción se debe a que la evolución del AC puede considerarse en general un fenómeno irreversible. Wolfram [4] basándose en estas estructuras catalogó los AC en **cuatro categorías**, estableció su relación con los cuatro tipos de lenguajes formales en la jerarquía de Chomsky [5] (regulares, independientes del contexto, dependientes del contexto y recursivamente enumerables) y con los cuatro modelos de máquinas abstractas que los reconocen (autómatas finitos, autómatas con pila, autómatas linealmente acotados y máquinas de Turing).



Fig. 4: Autómata tipo II de Wolfram.

Para nuestros propósitos, es de particular interés el AC *tipo II de Wolfram* que es una clase de autómatas que evolucionan con patrones irregulares o no tan claramente

definidos al principio, pero que a partir de un instante y en adelante llegan a una configuración de valores fijos o periódicos, como el que se muestra en la figura 4. Esta configuración podría pensarse como *el resultado* del cálculo efectuado por el AC durante su evolución.

### Redes Neuronales Artificiales

La Inteligencia Artificial (IA) propone para lograr un comportamiento inteligente de los programas, básicamente dos enfoques: el *enfoque simbólico* que intenta emular la forma en que *se cree* que los humanos piensan, tomando grandes cantidades de conocimientos (bases de conocimiento) y razonando a partir de ellos deductivamente con lógica (proposicional o de predicados clásicas, difusa, rebatible u otras); por otro lado, el *enfoque sub-simbólico* se orienta a emular con hardware o software la forma en la que la naturaleza resuelve los problemas; en particular, el *modelo conexionista* copia el hardware del sistema nervioso de los animales (claramente una simplificación del mismo), con la esperanza de que el comportamiento inteligente surja como una *propiedad emergente* de la complejidad generada por muchos elementos simples y discretos interactuando cooperativamente.

Una red neural artificial es uno de estos

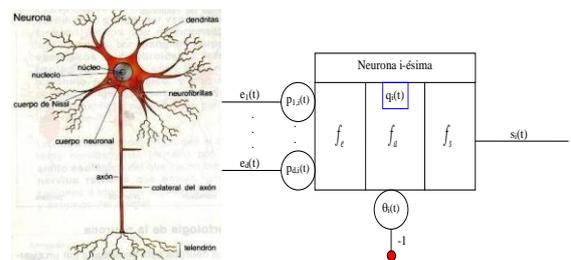


Fig. 5: Neurona biológica y modelo matemático de neurona artificial.

modelos inspirado en el funcionamiento del cerebro de los animales, respondiendo al enfoque sub-simbólico de la IA.

La idea es contar con una importante cantidad de elementos de modesto poder computacional (las neuronas) fuertemente interconectadas mediante canales por donde fluyen las señales (axones y dendritas). Se han desarrollado modelos que admiten

señales continuas y discretas de neuronas y redes; consideraremos aquí las discretas.

En un modelo general, cada neurona recibe entradas  $E(t)$ , aplica sobre ellas funciones (de entrada, activación y salida:  $f_e, f_a, f_s$  en la figura 5) y emite una salida  $S(t)$  en cada instante  $t$ . En las neuronas usuales al uso, no todas las funciones están presentes.

Se han diseñado infinidad de topologías distintas de conexionado de redes neurales y aún muchos tipos distintos de neuronas (en [6], [7] y otros puede encontrarse una extensa y variada clasificación de redes neurales).

A grandes rasgos, para determinar una típica red neural artificial se debe:

- Definir un conjunto finito y no vacío de elementos de cálculo (neuronas).
- Establecer la topología de la red (cómo interactuarán las neuronas entre ellas).
- Dar la dinámica de actualización de la red (sincrónica o asincrónica).
- Elegir una tarea a realizar por la red (pares entrada/salida de ejemplo).
- Generar un algoritmo para que la red aprenda la tarea (aprendizaje).

En nuestros proyectos se han utilizado redes perceptron multicapa (MLP) y redes de tipo Hopfield auto-asociativas, aunque aquí nos referiremos sólo al primer tipo.

En redes MLP, las neuronas se organizan en capas y la comunicación es siempre hacia delante (feedforward); la capa inicial que recibe las entradas desde el exterior, se denomina *capa de entrada* (corresponde a la *percepción* en la metáfora del sistema nervioso), luego puede haber cero o más capas de *neuronas ocultas* (que realizan el *procesamiento*) y la última capa (*acción*) que se comunica con el exterior, se denomina *capa de salida*.

Cada neurona de la figura 6, responde al modelo indicado en la figura 5, siendo los parámetros  $p_{ij}$  los pesos sinápticos que

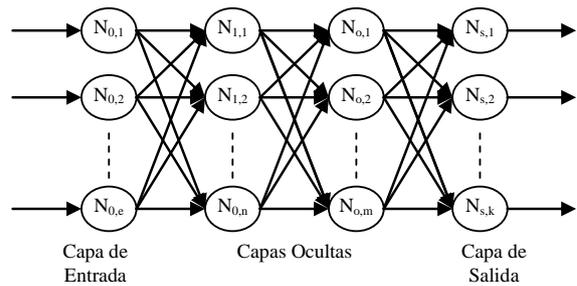


Fig. 6: Red perceptron multicapa.

miden la fuerza de conexión entre pares de neuronas de dos capas sucesivas.

La red, antes de ser productiva, debe pasar por una *etapa de entrenamiento* que, de alguna forma, calcule *el peso de cada conexión sináptica* y los *umbrales de cada neurona* de la red, de tal manera que la red **infera** la relación existente entre la entrada y la salida de los diferentes juegos de entrenamiento.

El proceso de aprendizaje en una red de este tipo se realiza mediante un proceso iterativo denominado **retropropagación de errores**. Matemáticamente, este algoritmo recurre a la técnica de **descenso por el gradiente**, que intenta disminuir en cada iteración el error producido por la red al ser aplicada a los ejemplos, medido como la diferencia entre la salida esperada y la salida calculada por la red; la función de error total se toma como el error medio cuadrático que produce la red y es una función sólo de los pesos sinápticos, supuesta fija la topología de conexionado y la cantidad y el modelo de las neuronas.

#### Idea de investigación.

El algoritmo de retropropagación de errores funciona modificando en cada iteración el valor de los pesos sinápticos (números reales) para que el error total de la red disminuya paulatinamente, logrando que las salidas reales calculadas desde las entradas de ejemplo, se acerquen tanto como se quiera a las salidas conocidas para esas entradas. La suposición de base es que, si la red aprende su tarea, en algún momento estos pesos dejarán de variar y se mantendrán fijos o se mantendrán variando

en forma oscilante (pequeña oscilación) alrededor de un valor fijo; si esto ocurre debería aproximadamente poder mostrarse esto como la evolución de un autómata celular del *Tipo II de Wolfram* y si esto es posible, tenemos la relación que estamos buscando: *puede verse el proceso de aprendizaje de la RNA como un proceso evolutivo, la evolución de un AC.*

#### Metodología.

En línea con los trabajos de Wolfram [3], que modela, codifica, ejecuta y visualiza los patrones generados por los AC traducidos a programas de computadoras y descubre similitudes de comportamiento, se utilizó la experimentación computacional, esto es:

- Se estudió y efectuó un desarrollo formal detallado de estos modelos desde la perspectiva de la teoría de autómatas y se analizaron sus propiedades, para determinar qué variables serían controladas durante el estudio.
- Se propusieron distintas codificaciones para representar la evolución del conocimiento que redes de tipo MLP logran durante su entrenamiento (seis funciones de conversión de los pesos sinápticos, secuenciación de neuronas según orden de capas, etc.).
- Se construyeron prototipos operativos de redes MLP con aprendizaje por retropropagación de errores (BP), redes Hopfield para reconocimiento de símbolos (casos simples) y de AC para muestreo.
- Se realizaron pruebas de evolución de variables bajo estudio (pesos sinápticos, estado de neuronas, configuraciones, etc.) y estudio de las mismas (límites de valores, periodicidad, estabilidad, etc.).
- Se determinó la representación gráfica a utilizar (granularidad, formato, colores, intervalos de muestreo y graficación, precisión de valores generados, etc.) de tal manera que sea uniforme y significativa en todos los modelos bajo estudio.

- Se diseñó e implementó un algoritmo para cotejo automático de resultados.
- Se experimentó con los programas construidos.

#### **Resultados**

Los experimentos realizados consistieron en la generación de archivos planos con los valores que adoptaban los pesos sinápticos durante el aprendizaje en redes MLP/BP, su posterior conversión a cadenas binarias según las seis codificaciones propuestas y análisis de aproximación a la evolución espacio-temporal de autómatas celulares mediante el programa de descubrimiento.

En todos los casos testeados hasta ahora, las únicas veces que se logró la concordancia buscada fue con archivos codificados según el tercer esquema de codificación, esto es, tomar el peso sináptico que es un número real entre 0 y 1, multiplicarlo por la unidad seguida de ceros para transformar las cifras más significativas a entero y luego convertir ese entero en binario puro según la representación larga de enteros del lenguaje C#. Con las otras codificaciones nunca obtuvimos concordancia.

El cuadro de la figura 7 muestra algunos de los resultados arrojados por el programa descubridor en aquellas oportunidades que tuvo éxito (resumidos). Se asumió un porcentaje de concordancia de más del 90% como éxito en la búsqueda de un patrón.

```
# 2a8_3221
# RNA-AC: BackPropagation
0      1 = 5.8759 %    0 = 94.124 %
1      1 = 93.688 %   0 = 6.3113 %

# 2a8_3241
# RNA-AC: BackPropagation
0      1 = 8.4819 %    0 = 91.518 %
1      1 = 91.064 %    0 = 8.9350 %

# 2a8_3241
# RNA-AC: BackPropagation
0      1 = 5.8991 %    0 = 94.100 %
1      1 = 92.228 %    0 = 7.7716 %

# Dig_15_15_20_5_1
# RNA-AC: BackPropagation
0      1 = 5.4086 %    0 = 94.591 %
1      1 = 94.093 %    0 = 5.9060 %
```

Figura 7: Resultados en corridas exitosas del descubridor de AC

```

# XOR_2241
# RNA-AC: BackPropagation
000000000000010      1 = 0.0 %      0 = 100.0 %
0000000000000101    1 = 0.0 %      0 = 100.0 %
00000000000001010   1 = 0.0 %      0 = 100.0 %
000000000000010101  1 = 0.0 %      0 = 100.0 %
0000000000000100000  1 = 0.0 %      0 = 100.0 %
0000000000000101011  1 = 0.0 %      0 = 100.0 %
0000000000000000000  1 = 0.0 %      0 = 100.0 %
00000000000001010110  1 = 0.0 %      0 = 100.0 %
00000000000000000001  1 = 100.0 %     0 = 0.0 %
000000000000000001110  1 = 100.0 %     0 = 0.0 %
00000000000000000001  1 = 100.0 %     0 = 0.0 %
00000000000000000100  1 = 100.0 %     0 = 0.0 %
00000000000000000000  1 = 0.0 %      0 = 100.0 %
00000000000000000001  1 = 100.0 %     0 = 0.0 %
00000000000000000011  1 = 0.0 %      0 = 100.0 %
000000000000000001110  1 = 0.0 %      0 = 100.0 %
000000000000000001101  1 = 0.0 %      0 = 100.0 %
00000000000000000011  1 = 0.0 %      0 = 100.0 %
00000000000000000000  1 = 0.0 %      0 = 100.0 %
000000000000000000010  1 = 100.0 %     0 = 0.0 %
000000000000000000000  1 = 100.0 %     0 = 0.0 %
000000000000000000001  1 = 0.0 %      0 = 100.0 %
000000000000000000001  1 = 100.0 %     0 = 0.0 %
000000000000000000000  1 = 100.0 %     0 = 0.0 %
0000000000000000000010  1 = 0.0 %      0 = 100.0 %
0000000000000000000010  1 = 0.0 %      0 = 100.0 %
0000000000000000000011  1 = 0.0 %      0 = 100.0 %
0000000000000000000011  1 = 0.0 %      0 = 100.0 %
0000000000000000000011  1 = 0.0 %      0 = 100.0 %
0000000000000000000011  1 = 0.0 %      0 = 100.0 %
0000000000000000000011  1 = 100.0 %     0 = 0.0 %
0000000000000000000011  1 = 0.0 %      0 = 100.0 %
0000000000000000000011  1 = 0.0 %      0 = 100.0 %
0000000000000000000011  1 = 0.0 %      0 = 100.0 %
... y siguen muchísimas más líneas.

```

Figura 8: Parte de la regla del AC deducida desde la evolución de pesos sinápticos

En el cuadro de la figura 8, se muestra un ejemplo de concordancia al 100%, entre los pesos sinápticos codificados y un AC, parcialmente ya que la vecindad es grande y conlleva una regla muy complicada, *pero la relación existe!!!*

Estas reglas complicadas siempre se encontraron en archivos representativos del resultado de aplicar la red MLP/BP al problema de determinar el XOR de valores codificados según el esquema 3. El título XOR\_2241 identifica el problema resuelto por la red y la topología de la red: dos neuronas en la capa de entrada, dos capas ocultas con 2 y 4 neuronas respectivamente y una única neurona en la capa de salida.

### Discusión

Los resultados obtenidos son alentadores ya que la relación entre los modelos bajo estudios experimentalmente existe y puede

detectarse con nuestros algoritmos; una nueva versión del programa que estudia los pesos sinápticos cambiando durante el entrenamiento de una RNA (traducidos a cadenas binarias) muestra además la regla específicamente, con lo que se espera poder formular expresiones lógicas que permitan atacar el problema formalmente. La tercera codificación con la que se encuentran concordancias es la más natural de las seis diseñadas ya que genera una representación binaria pura de los pesos sinápticos; la secuencia de pesos se definió en base al orden de las neuronas en cada capa, iniciando en la primer capa oculta, luego la segunda y así hasta llegar a la capa de salida.

Además de los programas construidos para experimentación, durante el progreso de la investigación se han construido y aplicado herramientas computacionales al modelado

de problemas sociales de las características expuestas en la introducción, a problemas de predicción de eventos naturales y a estimación de esfuerzos de ingeniería, pero los campos de aplicación posibles son muy vastos.

El riesgo para la salud de la vivienda urbana, que bajo un enfoque holístico de la salud comunitaria [8] y la vulnerabilidad social [9], resulta ser un fenómeno que los expertos en demografía y salud colectiva consideran complejo y altamente no lineal por involucrar factores humanos, sociales y geopolíticos; la ubicación y magnitud de incendios forestales en zonas serranas [10] representan un problema que a simple vista parece aleatorio e impredecible; estimar el costo y tiempo de desarrollo de proyectos de ingeniería siempre representa un desafío, sobre todo si lo que se construye tiene que ver con ideas como lo es el software. Los problemas indicados han sido abordados por nuestro equipo usando redes neurales artificiales y ha sido posible transferir los conocimientos y productos logrados a centros de investigación en Argentina, Brasil, Colombia, Paraguay, Cuba y a empresas. Pero las predilecciones de los consumidores, de los navegantes en la web, de los votantes, de los virus y las bacterias pueden ser también tratadas con técnicas de computación natural como las expuestas, especialmente cuando no exista un modelo matemático acertado.

### **Conclusión**

Varios proyectos relacionados durante los últimos años, han dedicado sus esfuerzos al estudio de modelos de computación no tradicionales, orientando parte de sus objetivos a investigación básica buscando relaciones entre dos de ellos: redes neurales artificiales y autómatas celulares, y también haciendo investigación aplicada, al construir herramientas computacionales para atacar problemas del mundo real, tanto en ciencias sociales, medioambiente y estimaciones de esfuerzos en proyectos de ingeniería. El método general ha sido el de estudiar los modelos a nivel formal,

luego implementarlos en programas de computadoras y finalmente experimentar con ellos. Del análisis de resultados se ha podido detectar la relación buscada y en el proceso se han transferido conocimientos y productos a otros centros de investigación y empresas. No menos importante, ha sido el apoyo brindado a la formación recursos humanos de grado y posgrado.

Queda mucho por hacer. En la actualidad (julio de 2013) hay tres proyectos vigentes en nuestro Departamento relacionados con redes neurales artificiales y autómatas celulares. Entre los objetivos involucrados se pueden mencionar los que se orientan: i) a la búsqueda de precisas relaciones entre los AC y las ecuaciones diferenciales, ii) a un enfoque alternativo a las redes neurales para apoyar el trabajo de los investigadores en ciencias sociales, efectuando búsquedas en espacios métricos por similaridad con algún método de interpolación, aún bajo estudio, que brinde una opción más eficiente, iii) el desarrollo de hardware y software para ser transferido a un centro de investigación en temas de producción animal, para la determinación en campo de la calidad de fibras textiles animales, iv) la recopilación de la información necesaria para predecir incendios forestales en la provincia de Córdoba y los programas necesarios para efectuar estas predicciones, v) el procesamiento de lenguaje natural para la determinación de implicaciones lógicas entre textos escritos, vi) el estudio, desarrollo y prueba de mejoras de los algoritmos de entrenamiento de redes neurales, vii) la construcción de entrenadores de redes y de simuladores con fines educativos, y muchos otros.

### **Agradecimientos**

Se agradece especialmente el soporte brindado a los proyectos RNA-AC y relacionados, por la Facultad Regional Córdoba de la Universidad Tecnológica Nacional a través de su Secretaría de Ciencia y Técnica; el subsidio del programa PROTRI 2008 de Dirección de Promoción de Actividades Científicas del Ministerio de Ciencia y Tecnología de Córdoba; los incentivos otorgados a investigadores formados por la Agencia Nacional de Promoción Científica y Tecnológica del Ministerio de Ciencia y Técnica de

Argentina y, finalmente, a los investigadores del IIGHI-CONICET-UNNE de Resistencia (Chaco) y del programa SUPPRAD de la Carrera de Veterinaria de la Universidad Católica de Córdoba, que aportaron sus conocimientos para lograr aplicaciones tanto en ciencias sociales como en producción animal, y transferencias de resultados.

#### Referencias.

- [1] Darwin, C. (1859), *On the origins of species by means of natural selection, or the preservation of favoured race in the struggle for life*, Traducción: *El origen de las especies*, (1992) Editorial Planeta, Barcelona, España.
- [2] Von Neumann J., Burks A. (1966), *Theory of Self-Reproducing Automata*, Univesity of Illinois Press, Urbana and London, USA/UK.
- [3] Wolfram S. (1994), *Cellular Automata and Complexity (collected papers)*, Addison-Wesley, Massachusetts, USA.
- [4] Wolfram S. (1984), *Universality and Complexity in Celullar Automata*, Physica D, vol. 10, pp. 1-35.
- [5] Hopcroft J., Ullman J. (1979), *Introduction to Automata Theory, Language and Computation*, Addison-Wesley, New York, USA.
- [6] Del Brío M. / Molina Sanz A. (1997), *Redes Neuronales y Sistemas Borrosos*, RA-MA, Madrid, España.
- [7] Hilera J./ Martínez V. (1995), *Redes Neuronales Artificiales*, Addison-Wesley, Delaware, USA.
- [8] Rojas M. / Meichtry N. / Vázquez J. / Castillo J. / Ciuffolini B. (2008), *Capítulo 16: Desarrollo de un software basado en redes neuronales artificiales para la vigilancia del riesgo de la vivienda urbana para la salud Pobreza y Vulnerabilidad Social*, Libro: Enfoques y Perspectivas, Autores: Sánchez, Diniz, Ed. ALAP Editores/Copiar, ISBN: 978-85620-160-2-8, pág. 335-358, Río de Janeiro, Brasil.
- [9] Rojas M. / Meichtry N. / Amarilla Velazco B. / Vázquez J. / Castillo J. / Alfonso E. (2013), *Vulnerabilidad y Riesgo en Asunción, Paraguay*, Revista Internacional de Investigación en Ciencias Sociales, Vol. 9 – Nro. 1 – Julio de 2013; Asunción, Paraguay; 2013. ISSN: 2225-5117.
- [10] Vázquez J., Castillo J., Cardenas M., Gordillo R., Villena Ruiz S. (2012), *Predicción de Incendios Forestales en la Provincia de Córdoba*, Workshop de Investigadores en Ciencias de la Computación (WICC 2012); Posadas, Misiones, Argentina. Mayo de 2012. ISBN: 978-950-766-082-5 (CD)

#### Datos de Contacto:

Juan Carlos Vázquez.  
jcvazquez@sistemas.frc.utn.edu.ar

Leticia Edith Constable  
lconstable@sistemas.frc.utn.edu.ar

Dpto. Ingeniería en Sistemas de Información  
Facultad Regional Córdoba  
Universidad Tecnológica Nacional  
Maestro López esq. Cruz Roja Argentina  
5016 - Ciudad Universitaria - Córdoba.