

# Enseñanza de los Modelos de Computación Conexionista

Patricio García Báez, Javier Rodríguez González

Dept. de Estadística, Investigación Operativa y Computación  
Universidad de La Laguna

38071 La Laguna, S/C Tenerife

e-mail: pgarcia@ull.es

## Resumen

Presentamos en esta ponencia la experiencia docente llevada a cabo en la asignatura de «Introducción a los Modelos de Computación Conexionista» durante los últimos tres años en el Centro Superior de Informática [1] de la Universidad de La Laguna.

## 1. Introducción

Los modelos de computación conexionista en los últimos tiempos han resuelto multitud de aplicaciones de una manera fácil y cómoda, entre ellas parte de las relacionadas con la Inteligencia Artificial (IA). Su especial predisposición para el aprendizaje los hace aptos para multitud de tareas insondables desde otros puntos de vista. Es por lo que consideramos fundamental la enseñanza de dichas técnicas dentro de las titulaciones informáticas.

La asignatura de «Introducción a los Modelos de Computación Conexionista» (IMCC) surge en la titulación de Ingeniero Técnico de Informática de Sistemas con la entrada del plan de estudios del año 1996, titulación que es impartida por el Centro Superior de Informática. Se trata de una asignatura optativa que consta de 4,5 créditos encuadrada dentro del tercer curso, además de ello también forma parte de la oferta de créditos de libre elección de la Universidad de La Laguna, lo que posibilita que muchos alumnos de la titulación de Ingeniero Técnico de Informática de Gestión también la cursen, y en menor medida alumnos de otras facultades (Matemáticas y Física).

Una de las primeras dificultades que nos encontramos para confeccionarla fue la escasa presencia de asignaturas de este tipo en las titulaciones técnicas. En la mayoría de casos

dichos contenidos eran asumidos en otras asignaturas o simplemente no se impartían.

## 2. Contexto

Esta asignatura se plantea como una introducción a las Redes Neuronales Artificiales (RNA) y complementa la visión del alumno sobre el área de la IA junto a estas otras asignaturas:

- Introducción a la Inteligencia Artificial, 6 créditos.
- Introducción al Reconocimiento de Patrones, 6 créditos.
- Robótica, 4,5 créditos.
- Heurísticas, 6 créditos.

Todas las asignaturas anteriores son también optativas y están encuadradas en el tercer curso. Cabe destacar entre ellas a «Introducción a la Inteligencia Artificial», que se centra más sobre los aspectos simbólicos de la IA.

Como asignatura previa en la que se imparten unos conocimientos iniciales sobre RNA tenemos a «Teoría de Autómatas y Lenguajes Formales», que está situada en segundo curso como troncal. En dicha asignatura, dentro de los diferentes modelos de computación existentes, se imparte un tema dedicado fundamentalmente a Redes Neuronales Formales de McCulloch y Pitts.

También hay que destacar que en la titulación de Ingeniero en Informática, que necesita como requisito de acceso poseer una titulación bien como Ingeniero Técnico de Informática de Sistemas o bien de Gestión, existe una asignatura optativa, «Modelos Conexionistas y Autómatas» con 7,5 créditos, en la que se ven conceptos más avanzados de RNA junto a otras que completan los estudios sobre IA.

Los profesores, firmantes de esta ponencia, que impartimos IMCC estamos encargados

además de la asignatura ya reseñada de «Modelos Conexionistas y Automatas» así como parte de las asignaturas de «Introducción a la Inteligencia Artificial» (3º) e «Inteligencia Artificial e Ingeniería del Conocimiento» (4º). Además nuestras investigaciones están centradas en el campo de las RNA.

### 3. Programa

La asignatura que se imparte durante el primer cuatrimestre del tercer curso esta organizada semanalmente mediante dos horas impartidas en el aula, que abarcan teoría y problemas, y una hora mas que se imparte en el laboratorio de prácticas, siendo flexibles en dicha asignación en la medida que los contenidos lo requieran y los medios lo permitan.

El temario que se imparte es el siguiente. En el se ha detallado el porcentaje de tiempo, sumando teoría+problemas+prácticas, empleado en cada tema:

0. Introducción (10%)  
Introducción. Inspiración biológica. Historia. Áreas. Propiedades y Capacidades.
1. Características de las RNA (20%)  
Modelo de neurona. Arquitecturas de red. Tipos de aprendizaje. Tipos de problemas. Codificación e/s. Fases de desarrollo.
2. Modelos (45%)  
Redes Neuronales Formales. Perceptrones Simples. Redes Multicapas. Redes Recurrentes. Sistemas Autoorganizados. Redes para tratamiento temporal. Otros modelos.
3. Neuro-Software (5%)  
Marcos formales. Programación directa. Librerías. Entornos de desarrollo. Herramientas de software.
4. Neuro-Hardware (5%)  
VLSI. Chips. Neuro-Tarjetas. Neurocomputadores. Máquinas de propósito general.
5. Aplicaciones (15%)  
Reconocimiento de Patrones. Procesado de Señales. Visión. Finanzas. Otras.

Las referencias [2] y [3] indican algunos de los libros de apoyo utilizados en el desarrollo de dichos temas, además en [4] tenemos otra referencia para poder profundizar mas en algunos aspectos.

Las prácticas están basadas principalmente en la utilización de una herramienta, Stuttgart Neural Network Simulator (SNNS) [5], que describiremos mas adelante. El programa de prácticas, según los temas cubiertos y las herramientas utilizadas, se organiza como sigue:

- Familiarización con el SNNS  
Temas 0, 1 y 3; SNNS.
- Tipos de Aprendizajes y Problemas  
Tema 1; Applets Java.
- Diseño de conjuntos de datos  
Tema 1; SNNS.
- Perceptrón simple y Adaline  
Tema 2; Codificación directa.
- Red multicapa  
Tema 2; SNNS.
- Red recurrente  
Tema 2; Codificación directa.
- Red autoorganizada  
Tema 2; SNNS.
- Otros modelos de RNA.  
Tema 2; SNNS y Applets Java.
- Familiarización con otros entornos de desarrollo. Neuro-hardware  
Temas 3 y 4. Tarjeta Neurocomputadora ZISC.

La asignatura se complementa con la presentación de un trabajo final realizado de forma individual por parte de cada alumno que cubre el tema 5, centrado en el Análisis y Desarrollo de una aplicación basada en RNA.

### 4. Metodología

En esta asignatura se pretende mas que tener un conocimiento exhaustivo de los distintos modelos de RNA, saber reconocer que un problema puede ser abordado con esta perspectiva, los métodos de análisis, diseño y evaluación de estas aplicaciones así como saber elegir los paradigmas y herramientas de apoyo utilizables.

El conocimiento adquirido mediante la exposición de la teoría en el aula trata de ser

asentado mediante las prácticas en el laboratorio. Para ello se distinguen dos tipos diferentes de prácticas, las tuteladas y las no tuteladas.

Durante la hora que dura una práctica tutelada los alumnos ejecutan un programa demostrativo o resuelven un determinado problema utilizando el conocimiento visto en clase y siguiendo los pasos que indica el profesor. Para ello habitualmente se hace uso de distintos Applets Java presentes en Internet [6] o bien de ejemplos sobre el SNNS.

En las prácticas no tuteladas al alumno ha de resolver un determinado problema satisfaciendo los requerimientos expuestos por el profesor. Normalmente se extienden más de una semana y el alumno trabaja tanto fuera como dentro del horario de laboratorio. En este último caso el profesor actúa en el laboratorio resolviendo dudas y realizando correcciones. El problema a trabajar es el mismo para todos los alumnos y una vez concluido éstos deben de entregar un pequeño informe del trabajo realizado.

La decisión de que prácticas serán tuteladas o no puede variar según el curso académico, aunque normalmente las primeras dos prácticas son tuteladas y durante las restantes se tratan de intercalar para facilitar el seguimiento de la asignatura por parte de los alumnos. Habitualmente a final de curso acaba habiendo un 50% de prácticas tuteladas y otro 50% de no tuteladas.

El trabajo final de la asignatura trata de cubrir todo el temario y, junto con la valoración de las prácticas, constituye la forma de evaluar los conocimientos adquiridos. Este trabajo es planteado ya desde mediados de curso, en él cada alumno ha de buscarse un problema y tratar de resolverlo con las pautas indicadas a lo largo de la asignatura. Para facilitar el trabajo a realizar se les ofrece una lista de enlaces que incluyen descripciones de problemas y conjuntos de datos [7], aunque gran parte de los alumnos prefieren montarse sus propios problemas, diseñando ellos mismos u obteniendo de otras fuentes sus conjuntos de datos. Finalmente ha de entregarse un informe en el que se detalle el problema elegido, el trabajo desarrollado así como se evalúe los resultados obtenidos.

## **5. Herramientas utilizadas**

Durante la realización de las prácticas se hace uso de diversas herramientas tendentes a facilitar el desarrollo de las mismas. A continuación se describen las más utilizadas.

### **5.1.Applets Java**

Existen en Internet multitud de páginas webs que incluyen Applets Java que implementan uno o varios modelos de RNA, en la referencia [6] se puede encontrar una lista de enlaces a algunas de estas páginas. La ventaja de este tipo de aplicaciones es que son muy ilustrativas y fáciles de utilizar, y habitualmente están bien documentadas y su código fuente está disponible.

### **5.2.Stuttgart Neural Network Simulator**

El SNNS es un simulador multiplataforma de RNA. Se distribuye como software libre y aunque inicialmente se desarrolló en el Instituto para Sistemas Distribuidos y Paralelos de Alto Rendimiento de la Universidad de Stuttgart en 1989 su mantenimiento es llevado hoy en día desde la Universidad de Tübingen [5]. Entre sus ventajas podemos destacar la gran cantidad de modelos neuronales que implementa, la flexibilidad existente para el diseño de redes complejas, la existencia de una interface gráfica de usuarios así como de una modalidad de ejecución en modo batch, goza de una potente documentación y se agradece la presencia de utilidades capaces de manipular conjuntos de datos, evaluar resultados o generar código en C.

Es de resaltar sobre todo su interface gráfica de usuario, que permite conocer los modelos conexionistas y comenzar a trabajar con ellos de forma muy rápida a usuarios inexpertos.

### **5.3.Diversos Lenguajes de Programación**

Una de las desventajas de utilizar simuladores de RNA es el hecho que esconden el funcionamiento real de la red y tienden a fijar en el alumno una visión de ésta como «caja negra». Es por ello que en algunas de las prácticas se le exige a los alumnos la codificación de alguna RNA en alguno de los lenguajes de programación que ya dominan.

Habitualmente optan por los lenguajes de C, Pascal o Matlab.

#### 5.4. Tarjeta Neurocomputadora ZISC

En el laboratorio disponemos de una tarjeta neurocomputadora ZISC (Zero Instruction Set Computer) de IBM [8]. Se trata de una tarjeta de bajo coste que dispone de 7 chips ZISC36, capaces de emular cada uno a 36 neuronas tipo RBF (Radial Basis Function). Con ella se incluye un kit de desarrollo para poder programarla en lenguaje C, aunque también están disponibles herramientas para trabajar con ella mediante Java, vía CORBA o emularla por software.

Trabajando con esta tarjeta los alumnos aprenden las prestaciones y limitaciones de este tipo de hardware.

## 6. Evaluación

### 6.1. Evaluación del alumno

Durante la realización de cada una de las prácticas el alumno es evaluado y al final de curso se obtiene una nota global de prácticas en base a ponderar la nota obtenida en cada una de las prácticas por la dificultad de éstas. En la nota final de la asignatura influyen a partes iguales tanto la nota global obtenida en las prácticas como la del trabajo final.

El porcentaje de alumnos que concluyen con éxito la asignatura sobre los que aparecen inicialmente matriculados se sitúan entorno a un 70%, siendo el número de alumnos matriculados en estos tres últimos cursos de 18, 22 y 29 respectivamente.

### 6.2. Evaluación de la asignatura

Como medida orientativa se muestra a continuación una tabla que resume la opinión de los alumnos, dichos datos han sido extraído de encuestas que se han realizado a final de curso en los dos últimos años.

Pregunta (0-10)	1999/2000 Media	2000/2001 Media
-----------------	--------------------	--------------------

	(Desv.)	(Desv.)
¿Ha encontrado la asignatura interesante?	8,1 (1,1)	8,4 (1,0)
¿Se tratan cuestiones realmente útiles?	7,4 (0,7)	7,9 (1,8)
Interés de la asignatura en relación con las otras	6,2 (1,2)	7,2 (1,3)
Grado de esfuerzo requerido en relación con las otras	5,5 (1,2)	5,0 (1,6)
¿Le parece adecuada la forma de evaluar la asignatura?	8,2 (0,6)	9,3 (0,7)

## 7. Conclusiones

La asignatura Introducción a los Modelos de Computación Conexionista pretende que los alumnos de las titulaciones Informáticas del Centro Superior de Educación obtengan conocimientos básicos y útiles sobre los modelos conexionistas. En general, los alumnos que la cursan se sienten satisfechos de haberlo hecho, como indican las encuestas y el progresivo aumento de matriculados desde sus inicios. Por otro lado, entre las dificultades encontradas tenemos su bajo número de créditos (4,5) en relación a otras optativas, lo que la hace poco atractiva para algunos alumnos así como la carencia de buena bibliografía escrita en lengua española.

## Referencias

- [1] <http://www.csi.ull.es/>
- [2] J.A. Freeman, D.M. Sakapura. *Redes neuronales. Algoritmos, aplicaciones y técnicas de programación*. Addison-Wesley / Díaz de Santos, 1993.
- [3] Laurene Fausett. *Fundamentals of Neural Networks. Architectures, algorithms and applications*. Prentice-Hall, 1994.
- [4] Simon Haykin. *Neural Networks. A comprehensive foundation*. Mcmillan, 1994.
- [5] <http://www-ra.informatik.uni-tuebingen.de/SNNS/>

- [6] <ftp://ftp.csi.ull.es/pub/assignas/IMCC/applets/>
- [7] <ftp://ftp.csi.ull.es/pub/assignas/IMCC/ds.html>
- [8] <http://www.fr.ibm.com/france/cdlab/zisc.htm>