

Redes neuronales y algoritmos genéticos en la docencia en las escuelas de ingenieros: Un problema práctico de diseño de pantallas acústicas

José Francisco Capdevila, José Manuel Ruiz,
José Luis Salinas, Enrique A. Sánchez Pérez
UNIVERSITAT POLITÈCNICA DE VALÈNCIA
jocapro@cam.upv.es
joruimo@cam.upv.es y zuaritx@yahoo.es
josail@cam.upv.es
easancpe@mat.upv.es

Abstract

En este trabajo desarrollamos una técnica basada en redes neuronales y algoritmos genéticos para construir un instrumento matemático para el diseño de pantallas acústicas adaptadas. Para hacer esto, usamos la información de una empresa especializada en la construcción de estos dispositivos. En realidad, solucionamos el problema siguiente: si se quiere atenuar el ruido en un cierto punto, ¿cuál es la mejor pantalla para hacerlo?

In this paper we develop a technique based on neural networks and genetic algorithms for constructing a mathematical tool for the design of adapted acoustic screens. In order to do this, we use the information of a company specialized in the construction of these devices. Actually, we solve the following problem: if we want to attenuate the noise in a certain point, which is the best screen for doing it?

Keywords: Modelos Matemáticos; Redes Neuronales, Pantallas Acústicas; Ingeniería.

1 Introducción

La intención de este trabajo es desarrollar un modelo utilizando las redes neuronales para aportar un ejemplo de aplicación real en el mercado analizando la respuesta de pantallas acústicas. La construcción pretende ser simple, puesto que el objetivo es facilitar una práctica de modelización para la docencia en una asignatura de especialización en un currículo de una carrera relacionada con la ingeniería civil. Se pretende que el futuro ingeniero pueda comprender el funcionamiento en su globalidad empezando por los elementos más simples, atendiendo más a las propiedades formales del modelo construido (cómo empezar, fases a analizar, tratamiento de datos, evolución, pasos, obtención de predicciones,...) que a la eficacia del algoritmo, como sería más propio en el caso de que un grupo de programadores pretendiera desarrollar un paquete informático para introducirlo en el mercado.

El problema que se presenta tiene que ver con la construcción de un modelo de calidad en la atenuación de los productos comerciales de apantallamiento (pantallas acústicas) de un proveedor del mercado a partir de sus catálogos comerciales. Así, el objetivo para el diseñador es presentar un instrumento de uso sencillo que permita al usuario la evaluación del producto que más le pueda interesar según la oferta de la empresa que facilita las pantallas. En definitiva, si se pretende atenuar el ruido en un cierto punto hay que preguntarse cuál es la mejor pantalla entre las que se ofertan en el catálogo.

La herramienta matemática será construida mediante redes neuronales artificiales (abreviadamente RNA, en español, ANN, Artificial Neural Networks, en inglés) para el ajuste de datos a partir de las características estructurales de las pantallas ofertadas en el catálogo. A partir de este ajuste se optimizará, usando un cierto criterio de error, la función definida por la red neuronal mediante algoritmos genéticos. En el apartado tercero se muestra el diagrama de flujo del procedimiento, así como las características técnicas del modelo construido.

2 Selección de las técnicas matemáticas utilizadas

Como se ha comentado, el equipo de trabajo ha seleccionado las redes neuronales artificiales y los algoritmos genéticos como instrumento para la modelización. Las herramientas informáticas actuales permiten la ejecución de programas que realizan una gran cantidad de cálculos en tiempo real. Esto facilita el uso de algoritmos numéricos relativamente simples para resolver problemas de optimización y los hace preferibles a otros procedimientos clásicos de optimización.

Ante la pregunta de qué ofrece una red neuronal diríamos que, a partir de unos datos conocidos, se obtiene una función (la propia red), que nos ajusta según los datos de entrada un output determinado. En este sentido, no es muy diferente de cualquier otro procedimiento de ajuste. Sin embargo, los requisitos exigibles sobre la propia función de ajuste, que aparecen como hipótesis en otros métodos (continuidad, derivabilidad,...), se reducen al simple supuesto de existencia de la función. Con respecto a los algoritmos genéticos, las razones que justifican su uso son similares. Estas técnicas se puede aplicar también con éxito en la modelización de problemas que aparecen en distintos ámbitos de interés para la ingeniería civil (redes de tráfico, flujos de mareas, datos de generación de núcleos comerciales, flujos dinámicos, aproximación de desniveles en sistemas de información geográfica, ...). El equipo ha empleado el paquete informático MATLAB 7.01 24704 R14 SP1[©]. Facilitamos a continuación una presentación esquemática de las técnicas matemáticas indicadas.

2.1 Redes neuronales artificiales

Las redes neuronales se emplean para ajustar una nube de puntos a una función que no se conoce mediante una familia de funciones que define la propia red, y que está determinada por una serie de operaciones simples (transformaciones lineales, sinusoidales, discretas y de otros tipos, ejecutadas en cada neurona), combinadas entre sí por una serie de conexiones que simulan el tejido nervioso. La obtención de los pesos (coeficientes numéricos) que afectan a cada una de las conexiones y que proporcionan una correlación óptima entre los datos de entrada y los output asociados es lo que se denomina el entrenamiento de la red. El resultado es una función que permite una predicción del valor del modelo en puntos próximos a los utilizados para el entrenamiento de la red. La arquitectura de la red (es decir el esquema propuesto de neuronas y de conexiones entre ellas) condiciona que el ajuste sea correcto. El funcionamiento de una red neuronal se basa en la realización de operaciones elementales en las neuronas, y en el flujo de información numérica a través de las interconexiones entre éstas. Este flujo está modulado por pesos, es decir, cada conexión es afectada por un valor entre cero y uno, y caso de que presente información innecesaria se cierra, o sea, se anula su valor; si es significativa, se verá afectada por el factor o peso correspondiente. Así, se va transfiriendo la información de una neurona a otra hasta llegar a la salida (nodo final, que puede ser único o múltiple). Cuando una red está ajustada con unos datos conocidos, y da resultados finales que describen el proceso real, se dice que la red está entrenada. La información del entrenamiento queda recogida en los parámetros de definición de la red. Normalmente, para conseguir una red neuronal que funcione correctamente, es necesario probar con varias arquitecturas. En nuestro caso, el equipo de trabajo comenzó a trabajar con un esquema conocido como perceptrón simple; con dos neuronas y dos niveles y tras una evaluación de resultados se propuso una arquitectura un poco más compleja. Como en cualquier procedimiento numérico, la extrapolación (obtención de valores de la función en puntos lejanos a los que se han utilizado para el entrenamiento de la red) puede dar resultados totalmente alejados de la realidad. Algunas ventajas de esta técnica, que la diferencia de otros métodos numéricos, son las siguientes:

- Buena tolerancia a fallos, dado que cada neurona percibe los fallos sólo como consecuencia del incremento de error, que es tratado en la fase de entrenamiento.
- Se puede operar en tiempo real, dado que la arquitectura utilizada puede ser tan simple como requieran las circunstancias.
- Se puede mejorar la eficacia de la red neuronal con la inserción de nuevas neuronas intermedias de forma ágil y rápida.

En la siguiente figura se muestra un esquema de una red neuronal de tres capas.

2.2 Algoritmos genéticos

Con un conjunto de datos iniciales (vectores de datos de entrada asociados a unos datos de salida) y una serie de transformaciones sencillas que pretenden simular las que se dan en el entrecruzamiento de la información genética que se da en los procesos biológicos, se pretende aproximar el óptimo de una función que se impone previamente (error). De cada generación producida a partir de los datos iniciales mediante esas transformaciones, se guarda el mejor, iniciando así un proceso iterativo que culmina cuando no se aprecian diferencias significativas en pasos sucesivos del algoritmo. El riesgo fundamental de esta técnica es que el dato solución

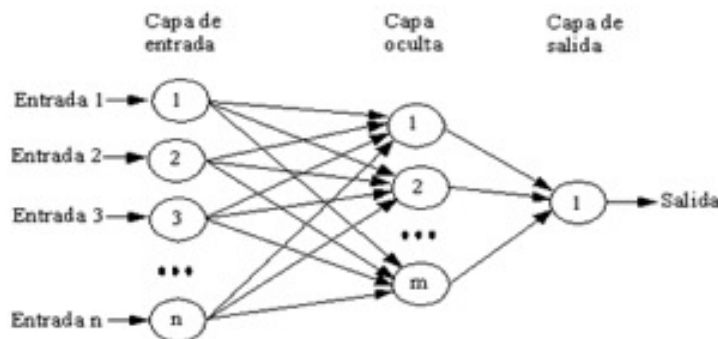


Figure 2.1: Esquema de las Redes

obtenido esté alejado del óptimo buscado por insuficiencia del proceso iterativo, o por que la función a la que tiende no permita la obtención de un resultado estable (convergencia asintótica, discontinuidad,...). Esto puede ser debido a una elección inadecuada de las familias de datos iniciales seleccionadas.

Esquema general de un algoritmo genético:

1. Entrada de datos: los datos conocidos deben tener unas ciertas garantías de aleatoriedad, o en su defecto, se debe conocer cuáles son sus propiedades al respecto.
2. Fase de generación de la descendencia, donde se evalúa con la función de ajuste cada dato generado (creación de la descendencia).
3. Criterio de adaptación, que salvaguarda el dato óptimo para producir la descendencia de la siguiente iteración.
4. Llegada al óptimo. Si esto no ocurre (es decir si no se verifican las condiciones que producen la parada del proceso iterativo), se utiliza la descendencia obtenida para realizar una nueva iteración.

3 Construcción del modelo

En esta sección presentamos los diferentes pasos que condujeron al grupo de trabajo a la obtención del modelo para el problema presentado.

1. Recogida de datos. La obtención de datos se realizó utilizando las gráficas del catálogo técnico *TECHNISCHE SCHRIFT 9 Schalltechnische Eigenschaften von Verglasungen mit Plexiglas und Makrolon* de la empresa RÖHM GmbH, que representaban el amortiguamiento acústico (R en dB) frente a la frecuencia (f en Hz), para una densidad superficial del material y un espesor determinados. Se extrajeron un total de 12 puntos de cada gráfica. También fue necesario distinguir entre pantalla alveolar y maciza, puesto que el comportamiento es distinto, y tratar estos dos casos de forma paralela, con redes neuronales diferentes.
2. Normalización y adecuación de los datos. Como se ha dicho ya, el software utilizado para el diseño o, entrenamiento y simulación de las redes neuronales, así como la implementación de los algoritmos genéticos, fue la versión comercial del asistente matemático

MATLAB 7.01 24704 R14 SP1[©], con los correspondientes paquetes de herramientas para Redes Neuronales Artificiales (Toolboxes for Artificial Neural Networks) y Algoritmos Genéticos (Genetic Algorithms). Técnicamente, las redes escogidas fueron siempre del tipo propagación inyectiva en avance o propagación secuencial involutiva (Feed-forward Backpropagation), con la regla de aprendizaje (Adaptation Training Functions) del tipo gradiente con el momento (Gradient with Moment - LEARNNGDM en las opciones del paquete informático).

3. Diseño de la arquitectura de la red neuronal. Recuérdese que se han diseñado dos redes, una para pantallas alveolares y otra para pantallas macizas. Ambas han sido alimentadas con datos de entrada (inputs) en matrices de 3 filas (densidad, frecuencia y espesor) y tantas columnas como puntos tomados de las gráficas. Para no tener problemas de mal condicionamiento matricial de partida, las unidades de trabajo tomadas fueron: kg/m^2 para la densidad, kHz para la frecuencia y cm para el espesor, de modo que las tres magnitudes estuvieran en torno al mismo orden de magnitud (criterio de estabilidad). El formato de los resultados de las redes (outputs) objetivo para el entrenamiento era un vector fila de amortiguamientos, de nuevo con tantas columnas como observaciones anotadas, con la particularidad de que se normalizaron, quedando comprendidos sus valores entre 0 y 1 (condicionamiento puro de saturación). Se guardaron los valores máximo y mínimo del vector original, para cumplir: $[mín, máx] \rightarrow [0, 1]$ que se relaciona directamente por homotecia (condicionamiento puro lineal), y dichos valores se necesitarían posteriormente para deshacer la normalización. La razón de normalizar el vector de resultados de la cada neurona (outputs) vino impuesta por la arquitectura de la red elegida, comentada a continuación.

Como ya se ha dicho en el apartado anterior, la red neuronal era deseable que presentase un único resultado de salida (output), el amortiguamiento acústico, y tres entradas (inputs), recuérdese, la densidad superficial en kg/m^2 , la frecuencia en kHz y el espesor en cm . Así pues, en líneas generales, la arquitectura de redes final tras analizarlas y estudiarlas consistiría en una capa de entrada con tres neuronas, una última capa de salida con una sola neurona y la presencia opcional de capas intermedias u ocultas. Partimos de una red básica con dos capas de neuronas tipo lineal puro (purelin); esto es, se trataba sencillamente de un ajuste lineal que, al no ajustarse a la realidad del problema, no proporcionaba datos fiables. Por eso, se fue modificando poco a poco la arquitectura, observando el comportamiento de los datos de salida.

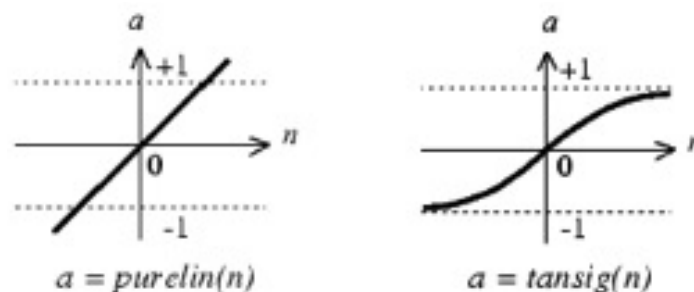


Figure 2.2: Esquema de las funciones de transferencia PURELIN y TANSIG

Finalmente, optamos por una arquitectura en tres niveles: un primer nivel con tres neuronas del tipo saturación tangencial (TANSIG), un segundo nivel con tres neuronas del

tipo lineal puro (PURELIN) y un tercero con una única neurona de tipo saturación tangencial (TANSIG). Como la función de transferencia de salida era de saturación tangencial (TANSIG) devolvía valores restringidos al intervalo $[-1, 1]$, que se imponían normalizar los datos de salida (outputs) de entrenamiento. Así pues, la red entrega como variable de salida un amortiguamiento adimensional que, como ya se ha mencionado, está comprendido entre 0 y 1.



Figure 2.3: Esquema de la arquitectura de la red neuronal

4. Entrenamiento. Como función de entrenamiento tomamos el algoritmo Levenberg-Marquardt (TRAINLM).
5. Construcción de las funciones de PREDICCIÓN Y ADAPTACIÓN (FITNESS). Una vez decidida la estructura y finalizado el entrenamiento de ambas redes (pantallas macizas y alveolares), se procedió a la implementación de las rutinas de predicción en formato Matlab. Se trata de funciones que tienen como variables de entrada la densidad superficial en kg/m^2 , la frecuencia en Hz y el espesor en mm ; unidades habituales de estas magnitudes. La rutina se encarga de transformar unidades, introducir los datos en la red neuronal y, una vez obtenida la salida (output) del amortiguamiento normalizado, se deshace la homotecia, obteniéndose como variable de salida el amortiguamiento acústico R predicho en dB.
6. Desnormalización de los datos. De los datos obtenidos inicialmente en las gráficas, se reservaron algunos, escogidos al azar, para comparar los datos reales con las simulaciones de las funciones de predicción. Los resultados se reflejan en las Tablas 2.1 y 2.2.

Table 2.1: Amortiguamientos Simulados mediante la RED de pantallas MACIZAS

| Densidad (kg/m ²) | Frecuencia (Hz) | Espesor (mm) | R Real (dB) | R Predicho (dB) | Error Relativo |
|-------------------------------|-----------------|--------------|-------------|-----------------|----------------|
| 9,6 | 1500 | 8 | 33,3 | 33,56 | +0,78% |
| 4,8 | 1000 | 4 | 27 | 25,77 | -4,54% |
| 7,2 | 250 | 6 | 22 | 20,926 | -4,92% |
| 12 | 667 | 10 | 30 | 29,08 | -3,08% |
| 15 | 833 | 6 | 33 | 31,19 | -5,47% |

Table 2.2: Amortiguamientos Simulados mediante la RED de pantallas ALVEOLARES

| Densidad (kg/m ²) | Frecuencia (Hz) | Espesor (mm) | R Real (dB) | R Predicho (dB) | Error Relativo |
|-------------------------------|-----------------|--------------|-------------|-----------------|----------------|
| 3,5 | 333 | 8 | 18 | 18,46 | +2,54% |
| 7,7 | 750 | 4 | 24,7 | 25,29 | +2,37% |
| 5 | 2000 | 16 | 25,7 | 26,14 | +1,72% |
| 6,7 | 500 | 40 | 24,7 | 24,59 | -0,44% |

Se observa, por tanto, que los errores relativos en la red para pantallas macizas rondan, salvo algunas excepciones, el 5%. En la simulación de pantallas alveolares los errores son

aún menores, llegando tan sólo a alcanzar el 2,5%. Estos resultados, teniendo en cuenta tanto la simplicidad del modelo, como la del planteamiento general del problema físico, son bastante aceptables, al menos desde el punto de vista docente.

7. Criterios de la función ERROR. Una vez finalizado el apartado de la simulación, se procedió a definir la función de error, que posteriormente minimizaríamos mediante algoritmos genéticos. El problema que en última instancia se pretendía modelizar era el siguiente: dado un amortiguamiento acústico R_0 requerido y una frecuencia f_0 determinada, encontrar una pantalla; esto es, un espesor e y una densidad superficial σ , que cumpla las características. Se escogió, por simplicidad, un error de tipo cuadrático. Esencialmente, la función a optimizar era $f(\sigma, e) = (R_0 - R_{simulado})^2$, donde por supuesto, para calcular $R_{simulado}$, se llamaba a las rutinas de predicción para una frecuencia f_0 fija y una densidad σ y un espesor e variables.
8. Minimización de la función de ADAPTACIÓN (FITNESS) para la obtención del óptimo solución con ALGORITMOS GENÉTICOS. Como el dominio de las redes no estaba restringido de ninguna manera, al introducir la función adaptativa (fitness) en el Toolbox de algoritmos genéticos, éste proporcionaba valores de espesor y densidad que, en efecto, anulaban la función de error, pero que carecían la mayoría de las veces de sentido físico (resultados negativos o exageradamente grandes o pequeños) e incluso de viabilidad real (ratios $\frac{\sigma}{e}$ desproporcionados). Esto ocurría porque las redes neuronales predecían amortiguamientos acústicos lógicos para valores de espesor y densidad fuera de rango (es de sobra conocido que las redes neuronales tienen una gran potencia para la interpolación, pero fallan estrepitosamente en la extrapolación).

Se decidió entonces solucionar el problema modificando la función fitness de la siguiente manera. Se definieron unos límites viables desde el punto de vista técnico de la producción y manipulación de las pantallas acústicas, tanto para el caso de las macizas como para el de las alveolares. La función de error quedaba ahora penalizada, con valores del orden de 10^6 , para argumentos que no entraran dentro del rango escogido, de modo que los vectores incoherentes generados por los algoritmos genéticos no pasaran a la siguiente generación. Finalmente, se implementó una rutina en formato Matlab que automatizara todo el proceso, de modo que el usuario únicamente tuviera que introducir el amortiguamiento R solicitado en dB y la frecuencia en Hz . A parte del resultado en pantalla de espesor y densidad óptimos (en mm y kg/m^2 respectivamente) tanto para pantalla maciza como para alveolar, la función genera un fichero de MSWORD que detalla el proceso de optimización.

4 Algunos comentarios finales

La diversidad de opciones que presenta la herramienta Toolbox del paquete informático MATLAB[®] era prácticamente ilimitada para modificar los algoritmos genéticos. Afinando el límite de la optimización de la función de ajuste (fitness) a una tolerancia de la centésimas (10^{-2}), definiendo una población inicial genérica (dentro del rango escogido o zona de regularidad) y limitando el número de generaciones a 150 o el tiempo de espera a 3 minutos se obtenían resultados altamente satisfactorios.

Debido al carácter intrínsecamente aleatorio de los algoritmos genéticos, para una misma frecuencia y amortiguamiento, la rutina de optimización no devuelve siempre los mismos valores de espesor y densidad óptimos. Por otra parte, este hecho no se aleja de la realidad del problema,

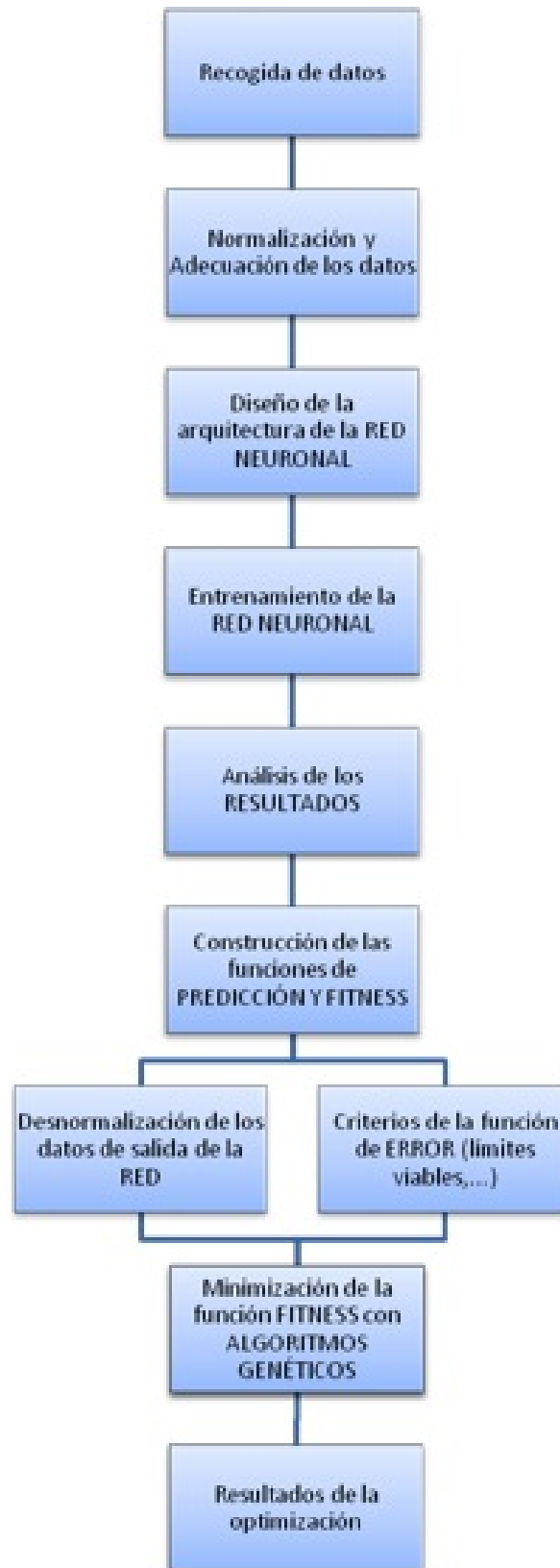


Figure 2.4: Diagrama de flujo del proceso de optimización

ya que, como es lógico, no existe una única pantalla que cumpla unas propiedades determinadas de atenuación acústica.

5 Aplicación didáctica y conclusiones

Desde el punto de vista de la utilización didáctica de esta práctica, hay varios aspectos de interés que indicamos a continuación.

- a- Las técnicas matemáticas utilizadas, y ésta es posiblemente una de las ventajas del proyecto, son sencillas de entender a partir de conocimientos elementales de matemáticas; ni siquiera es necesario que los alumnos hayan realizado cursos de especialización en técnicas numéricas, es suficiente con que conozcan las asignaturas básicas (generalmente, contenidos de cálculo y álgebra).
- b- Sin embargo, tanto las redes neuronales como los algoritmos genéticos son procedimientos que no suelen figurar entre los contenidos de las asignaturas de una carrera relacionada con la ingeniería civil. Se podría plantear la misma práctica utilizando otras técnicas numéricas alternativas; son necesarias una técnica de ajuste y otra de optimización.
- c- El modelo ha sido realizado siguiendo un esquema de complejidad creciente, en el sentido que se partió de una red simple que se fue complicando en pasos sucesivos. Esto la hace especialmente adecuada para utilizarla para un proyecto de curso que se realice a lo largo de una serie de prácticas consecutivas.

Referencias

- [1] Barro, Senén y Mira, José. *Computación Neuronal*. Ed. Servicio de Publicaciones e Intercambio Científico (Campus Universitario Sur), Univ. de Compostela, 1995
- [2] Hilera, José R y Martínez, Víctor J. *Redes neuronales artificiales: Fundamentos, módulos y aplicaciones*. Ed. Ra-Ma, 1995
- [3] http://es.wikipedia.org/wiki/Algoritmo_gen%C3%A9tico
- [4] http://es.wikipedia.org/wiki/Redes_neuronales
- [5] Isasi Viñuela, Pedro y Galván León, Inés M. *Redes de neuronas artificiales: un enfoque práctico*. Ed. Pearson Prentice Hall, 2004
- [6] Russell, Stuart y Norvig, Peter. *Inteligencia artificial: Un enfoque moderno*. Ed. Pearson Prentice Hall, 2004

