

# NEUROCOMPUTACION. PRACTICA 1.

**Objetivo general:** Adquisición de destrezas en la construcción, entrenamiento y uso de Redes Neuronales (RNs) feedforward, mediante el uso de las toolbox de RNs de Matlab

Del directorio Material/Datos/Iris/ se puede descargar el material necesario para la realización de la Práctica 1, Además de información sobre la Base de Datos Iris, se proporcionan 2 ficheros relacionados con el problema de las plantas Iris, (Iris1.txt, Iris2.txt). En el fichero Iris1.txt aparecen los ejemplos que deberán ser usados para el entrenamiento y validación de RNs Backpropagation que permitan la clasificación de datos Iris. En el fichero Iris2.txt están contenidos los modelos de test que deberán ser utilizados en el proceso de evaluación de las RNs.

**La práctica se compondrá de los siguientes apartados:**

**1.-** Analizar y diseñar una función Matlab (definida en un fichero script **ProcesamientoIris.m**) que:

Lea los ficheros ASCII Iris1.txt (entrenamiento y validación) e Iris2.txt (test), elimine las cabeceras, cree y almacene en el directorio de trabajo los siguientes ficheros de texto:

I) **Entrena.txt** se obtiene a partir de Iris1.txt, eliminando las cabeceras y el formato será de 4 columnas de datos correspondientes a las entradas que aparecen en Iris1.txt

II) **Salida2.txt** se obtiene a partir de Iris1.txt, eliminando las cabeceras; el formato será de 2 columnas de datos correspondientes a las salidas que aparecen en Iris1.txt. Estas salidas son el resultado de sustituir cada string de salida “sestosa”, “versicolor”, “virginica” por 2 valores binarios (1 0), (0 1), (1 1) dependiendo del caso.

III) **Salida3.txt** se obtiene a partir de Iris1.txt, eliminando las cabeceras; el formato será de 3 columnas de datos correspondientes a las salidas que aparecen en Iris1.txt. Estas salidas son el resultado de sustituir cada string de salida “sestosa”, “versicolor”, “virginica” por 3 valores binarios (1 0 0), (0 1 0), (0 0 1) dependiendo del caso.

IV) **TestEntrada.txt** se obtiene a partir de Iris2.txt, eliminando las cabeceras y el formato será de 4 columnas de datos correspondientes a las entradas que aparecen en Iris2.txt

V) **TestSalida2.txt** se obtiene a partir de Iris2.txt, eliminando las cabeceras; el formato será de 2 columnas de datos correspondientes a las salidas que aparecen en Iris2.txt. Estas salidas son el resultado de sustituir cada string de salida “sestosa”, “versicolor”, “virginica” por 2 valores binarios (1 0), (0 1), (1 1) dependiendo del caso.

VI) **TestSalida3.txt** se obtiene a partir de Iris2.txt, eliminando las cabeceras; el formato será de 3 columnas de datos correspondientes a las salidas que aparecen en Iris2.txt. Estas salidas son el resultado de sustituir cada string de salida “sestosa”, “versicolor”, “virginica” por 3 valores binarios (1 0 0), (0 1 0), (0 0 1) dependiendo del caso.

## Notas:

Los ficheros de texto obtenidos deben estar preparados para ser recuperados en el formato adecuado, columnas por filas.

El script ProcesamientoIris.m contendrá varias funciones. Algunas de las funciones que pueden utilizarse para esta fase pueden ser: fopen save, textscan, fseek, feof, fgetl, isempty, strend, length, strncmpi, dlmwrite, estructuras cell,... aunque cada uno puede utilizar aquellas funciones que cree convenientes.

**2.** Construir un script **PracticaIris.m** cuyo objetivo sea analizar diferentes topologías, funciones de activación y funciones de entrenamiento con RNs backpropagation, para la clasificación de los datos Iris, utilizando los ficheros obtenidos en la fase anterior.

El método que se debe seguir para comparar la efectividad de las redes entrenadas, estará basado en el % de aciertos obtenidos sobre el conjunto TestEntrada.txt. Para obtener el % de aciertos deberéis construir una función específica.

Las fases que se seguirán en este apartado serán las siguientes:

**2.1** Se utilizará el entorno gráfico GUI mediante la ejecución de la función nprtool, grabando el script correspondiente en un fichero \*.m (poner el % de test a cero, ya que después se utilizan los conjuntos de test)

**2.2.** A partir de dicho script, se realizarán las modificaciones pertinentes, para detectar la topología más adecuada (número de capas, número de neuronas en las capas ocultas), para ello se realizarán varios ensayos (alrededor de 10 con cada topología), anotando para cada topología el % de aciertos medio obtenido con los datos de test.

**2.3.** Una vez elegida la topología más adecuada, se deberán de crear nuevas redes neuronales, que serán entrenadas con diferentes funciones de activación y entrenamiento (para cada caso realizar alrededor de 10 ensayos), con objeto de anotar el % de aciertos medio obtenido con los datos de test.

**2.4.** Finalmente se grabará la RN entrenada que mejor resultado proporcione, de forma que pueda ser utilizada “a posteriori” por el conjunto de test.

- Al margen de las variaciones que deben ser consideradas, otras características como el número de iteraciones, pueden ser modificadas.

- Obviamente el script deberá constar de varias funciones, para permitir lanzarla y obtener el resultado deseado.

- Para poder automatizar los procesos anteriores, **deberán utilizarse las cell de arrays**, de forma que se permita la ejecución iterativa de la combinación de los diferentes parámetros, para la obtención de una RN óptima que resuelva el problema de la clasificación de los datos Iris.

- En la documentación se presentarán las topologías y métodos de entrenamiento de las RNs más idóneas para la resolución del problema, justificando de forma detallada la elección, proporcionando mediante tablas, los valores de los distintos parámetros utilizados, las gráficas más representativas etc., y **el % de acierto (Obligatorio)**.

**ESTA PRÁCTICA SE ENTREGARÁ CONJUNTAMENTE CON LA PRÁCTICA 2 A TRAVÉS DE LA Web**

**HAY QUE ENTREGAR EN SOPORTE INFORMÁTICO:**

- 1.- Código fuente de todos los ficheros\*.m
- 2.- Ficheros resultantes de la primera fase: Entrena.txt, Salida2.txt, ...
- 2.- Documentación sobre los ficheros \*.m
- 3.- Comparación de los resultados obtenidos con el % de aciertos de los conjuntos de test, y presentación de la elección que ha proporcionado mejores resultados.
- 3.- Explicación de los métodos utilizados para el entrenamiento.

**FECHA TOPE DE ENTREGA: 21 de Abril de 2014**