

## NEUROCOMPUTACION. PRACTICA 2

**Objetivo general:** Conocer que en la mayoría de los problemas que puedan ser abordados mediante RN, es necesario el análisis de los datos de los que se dispone. Los problemas presentados serán de aproximación de funciones reales mediante RNs, pero pueden ser trasladados a muchas otras clases de problemas.

La aplicación de las RN, la mayoría de las veces no es directa, sino que necesita de un análisis de los datos para actuar de una u otra forma. A continuación se propone la realización de una práctica de aproximación de funciones mediante RNs, para observar la necesidad de analizar los datos y actuar de una u otra forma según los casos concretos. Aunque esta práctica no debe ser entregada, sí debe ser realizada, para el aprendizaje de algunas características, tanto del problema concreto como de aplicación de la RN en Matlab.

Los ficheros necesarios para la realización de la práctica se encuentran en el directorio:  
Material/Datos/FicheroFuncionesReales/FuncionesReales.rar

### La siguiente práctica debe ser realizada, pero no entregada

Descargar y seleccionar los ficheros Funcion1Entrena y Funcion1Test (almacenados en FuncionesReales.rar). Estos ficheros contienen datos de una función real definida en  $[0, 1] \times [0, 1] \times [0, 1] \rightarrow [0, 10]$  que se desea aproximar mediante una RN feedforward.

Construir una RN apropiada para la aproximación de dicha función, y observar que el proceso de aprendizaje no se produce, la causa es que las salidas deseadas pertenecen al intervalo  $[0, 10]$ , sin embargo si la función de activación en la capa de salida es p.e. la sigmoideal, las salidas obtenidas, siempre van a pertenecer al intervalo  $[0, 1]$ .

La solución a éste problema consiste en asignar como función de activación en la capa de salida, la función de activación identidad ('purelin'). Comprobar que con dicha función de activación en la capa de salida, es posible la aproximación de la función mediante la RN construida.

### La siguiente práctica debe ser realizada y entregada

Del directorio Material\Datos\FicheroFuncionesReales se puede descargar el material necesario para la realización de la Práctica 2. Los ficheros que se utilizarán son: Funcion2Entrena y Funcion2Test (almacenados en FuncionesReales.rar).

Los dos ficheros contienen datos de E/S de una función real, definida en  $[0, 1] \times [0, 1] \rightarrow [9, \infty]$ . Se desea construir una RN feedforward que aproxime dicha función, sin embargo cuando se intenta el aprendizaje de una RN con estos datos, se aprecia que es inviable debido al rango que tiene el recorrido de dicha función, por lo que los datos necesitan previamente ser preprocesados, con el objetivo de poder obtener en el proceso de aprendizaje de la RN feedforward, unos resultados medianamente aceptables.

Pueden ser considerados diversos tipos de preprocesamiento; en esta práctica se considerarán dos concretos, de forma que puedan ser comparados. Los dos tipos de preprocesamiento que deben realizarse son:

- I) Aproximación por intervalos. Para resolver el problema mediante esta técnica, deberemos aplicar una transformación lineal por intervalos a los datos de salida. Se considerarán los intervalos  $[0, 10]$ ,  $(10, 100]$ ,  $(100, 1000]$  ... Las funciones lineales que deben usarse para realizar la transformación de los datos serán de la forma:

$$y = ax + b.$$

Consideremos el intervalo  $[0, 10]$ , tratamos de hallar una transformación lineal, de forma que cuando un punto pertenezca a  $[0, 10]$ , como máximo me pueda equivocar en 1 unidad, por lo que la pendiente de la recta que aproxima con estas condiciones, tiene de tangente  $a = 1/10 = 0.1$ . Otro dato del que se dispone, es que el extremo inferior del segmento de la recta que estamos buscando tiene su origen en el  $(0, 0)$ . Con estos dos datos, ya podemos hallar la ecuación ( $y = ax + b$ ) de transformación de los datos pertenecientes al intervalo  $[0, 10]$ .

Para la obtención de la función lineal, para transformar los datos que pertenecen al intervalo  $(10, 100]$  consideraré que como máximo yo me pueda equivocar en 10 unidades, por lo que  $a = 1/100 = 0.01$ . Además el 2º segmento de la función lineal que estamos buscando tendrá su origen, en el extremo del

segmento obtenido mediante la función anterior, por lo que se obtendrá, mediante la sustitución de  $x=10$  en la función obtenida para el intervalo  $[0, 10]$ .

Con estos datos se halla la función lineal para el intervalo  $(10, 100]$ .

Las ecuaciones lineales para el resto de los intervalos se obtienen de forma análoga.

A continuación se evaluarán diferentes RNs, tomando los datos de entrada iniciales de Funcion2Entrena, y los datos obtenidos después de aplicar las transformaciones lineales a los datos de salida de Funcion2Entrena.

Una vez seleccionada la mejor red, ésta será aplicada a los datos de Funcion2Test (donde previamente se les ha aplicado las transformaciones lineales a los datos de salida de Funcion2Test).

Finalmente se aplicarán las transformaciones inversas a los datos de salida de la red, y se obtendrá el error cometido.

- II) Realizar una transformación logarítmica:  $y = \log(x+1)$ , y seguir los mismos pasos que en la transformación anterior.

Las siguientes actividades se realizarán independientemente para las dos formas de preprocesamiento:

a) Desarrollar y aplicar un programa \*.m, que al recibir como argumento el nombre de uno de los ficheros de que se dispone (de entrenamiento o validación), cree otro fichero con las mismas entradas y las salidas procesadas, en el formato requerido por Matlab.

b) Construir una RN feedforward, para el aprendizaje de los ejemplos del nuevo fichero de entrenamiento obtenido mediante la ejecución de la actividad anterior, intentando optimizar la topología y los parámetros del método de aprendizaje Backpropagation.

c) Una vez entrenada la red, desarrollar un programa \*.m, que usando la RN entrenada y el fichero de validación, muestre en pantalla por columnas, los datos de entrada, los datos de salida esperados y los datos de salida obtenidos después de realizar la transformación inversa al preprocesamiento, así como el error cometido .

**ESTA PRÁCTICA SE ENTREGARÁ CONJUNTAMENTE CON LA PRÁCTICA 1 A TRAVÉS DE LA Web**

**HAY QUE ENTREGAR EN SOPORTE INFORMÁTICO:**

- 1.- Código fuente de todos los ficheros de software creados
- 2.- Documentar:
  - 2.1.- El problema que se desea abordar
  - 2.2.- Las dos formas de resolverlo
  - 2.3.- Conclusiones obtenidas sobre la aplicación de las dos formas de preprocesamiento
  - 2.4.- Forma de utilizar los programas.

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