



**ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA**
Universidad de Córdoba



4º Grado en Ingeniería Informática

Especialidad de Computación

Práctica 3: Redes neuronales de funciones de base radial

Introducción a los Modelos Computacionales

i72hisan@uco.es

Noelia Hinojosa Sánchez (46270988W)



Índice General

1. Introducción	3
2. Descripción entrenamiento de las redes RBF	4
3. Experimentos y análisis	5
3.1. Bases de datos utilizados	5
3.2. Valores de parámetros considerados	6
3.3. Resultados obtenidos	6
3.4. Análisis de resultados	7

Capítulo 1

Introducción

En esta práctica, se va a implementar una red neuronal de tipo RBF. En este caso, se utiliza python para su implementación en vez de C++ como en las dos prácticas anteriores.

A continuación se explicará el script implementado, para posteriormente analizar los resultados obtenidos.

Capítulo 2

Descripción entrenamiento de las redes RBF

Para el entrenamiento de la red neuronal de tipo RBF se van a realizar los siguientes pasos:

1. **Clustering:** Se aplicará un algoritmo de clustering. Este algoritmo establecerá los centros de las funciones RBF. Los centros calculados corresponderán a los pesos de la red desde capa de entrada a la capa oculta.
2. **Ajuste de radios:** Se ajustarán los radios de las funciones RBF mediante una heurística sencilla. En este caso se ha aplicado la media de las distancias hacia el resto de los centros.
3. **Aprendizaje:** Se ajustarán los pesos de la red neuronal (desde capa oculta a la capa de salida) mediante el aprendizaje. Para ello, se utilizan los siguientes métodos dependiendo del tipo de problemas al que nos enfrentamos.
 - **Problemas de regresión:** La pseudo-inversa de Moore Penrose.
 - **Problemas de clasificación:** Un modelo lineal de regresión logística.

Capítulo 3

Experimentos y análisis

3.1. Bases de datos utilizados

Para probar el código desarrollado, se han utilizado 5 datasets que se nos aportan con la documentación de la práctica:

- **Dataset SIN**

El problema del SIN consta de 120 patrones de entrenamiento y 41 de test. A estos datos se le han introducido un cierto ruido complicando la función inicial.

- **Dataset QUAKE**

En este problema el objetivo es averiguar la fuerza de los terremotos en la escala de Richter. Para ello tenemos 3 entradas (profundidad focal, latitud y longitud) y obtenemos una sola salida. Consta de 1633 patrones de entrenamiento y 546 de test.

- **Dataset PARKINSONS**

En este problema el objetivo es identificar si un paciente tiene la enfermedad de Parkinson o no. Consta de 4406 patrones de entrenamiento y 1469 de test.

- **Dataset ILDP**

En este dataset se clasifica un conjunto de pacientes siendo hepáticos o no hepáticos. Consta de 405 patrones de entrenamiento y 174 de test. Para este conjunto usaremos FairLearn debido a la diferencia de género

que contienen los datos, teniendo 441 patrones hombres y 142 mujeres.

- **Dataset noMNIST**

Este dataset se compone por un conjunto de imágenes de diferentes letras escritas con diferentes tipografías. Consta de 900 patrones de entrenamiento y 300 de test, utilizando letras de la A-F. Al ser cada imagen de 28x28 píxeles se tendrán 784 variables de entrada.

Los 3 primeros dataset son de regresión y los dos últimos corresponden a clasificación.

3.2. Valores de parámetros considerados

- Argumento **t**, **train_file**: Indica el nombre del fichero con los datos de entrenamiento a utilizar. Obligatorio.
- Argumento **T**, **test_file**: Indica el nombre del fichero con los datos de test a utilizar. Opcional.
- Argumento **c**, **classification**: Indica si el problema es de clasificación o no. Opcional.
- Argumento **r**, **ratio_rbf**: Indica la razón, en tanto por uno, de neuronas RBF con respecto al total de patrones en entrenamiento. Opcional.
- Argumento **l**, **l2**: Indica si se utilizará regularización de L2 en lugar de la regularización L1. Opcional.
- Argumento **e**, **eta**: Indica el valor del parámetro η . Opcional.
- Argumento **f**, **fairness**: Activa el cálculo de métricas de rendimiento por grupos. Opcional.
- Argumento **o**, **outputs**: Indica el número de columnas de salida que tiene el conjunto de datos y que siempre están al final. Opcional.

3.3. Resultados obtenidos

Los resultados obtenidos se encuentran en la carpeta results del repositorio para todos los datasets utilizados.

3.4. Análisis de resultados

En los problemas de regresión, el valor del parámetro η (factor de aprendizaje) no influye en el error cuadrático medio cometido por la red, por lo que se puede afirmar que dicho error dependerá del número de neuronas en la capa oculta de la red neuronal.

En el caso concreto del problema del QUAKE, como las variables de entrada son dependientes entre sí, cuanto menor sea el número de neuronas de la capa oculta en la red neuronal RBF, menor será el error cometido tanto en entrenamiento como en test.

Por otro lado, en el problema de Parkinson, nos encontramos con que todas las variables son independientes, de modo que cuanto menor sea el número de neuronas en la capa oculta de la red, mayor será el error cometido.

En los problemas de clasificación, el error CCR de entrenamiento y test depende, además del número de neuronas de capa oculta, del factor de aprendizaje. Es decir, se obtendrán unos mejores resultados en función del factor de aprendizaje (η).