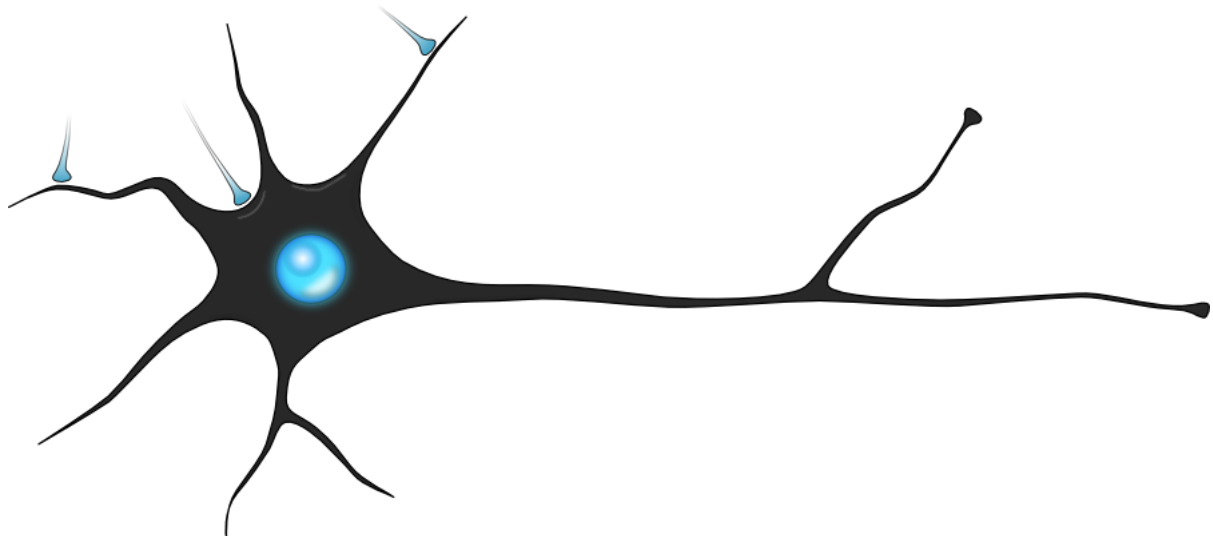




PRÁCTICA 3:

REDES NEURONALES DE FUNCIONES DE BASE RADIAL.



Víctor Monserrat Villatoro
i32moviv@uco.es
45887876R



1.	Proceso para llevar a cabo el entrenamiento de las redes RBF.	3
2.	Experimentos y análisis de resultados.	4
2.1.	Descripción de las bases de las bases de datos utilizadas.	4
2.1.1.	Función seno.	4
2.1.2.	Base de datos CPU.	4
2.1.3.	Base de datos forest.	4
2.1.4.	Base de datos Iris.	4
2.1.5.	Base de datos Digits.	5
2.2.	Descripción de los valores de los parámetros considerados.	5
2.3.	Resultados obtenidos.	6
2.3.1.	Función seno.	6
2.3.2.	Base de datos CPU.	6
2.3.3.	Base de datos forest.	6
2.3.4.	Base de datos Iris.	7
2.3.5.	Base de datos Digits.	8
2.4.	Análisis de resultados.	9
2.4.1.	Función seno.	9
2.4.2.	Base de datos CPU.	9
2.4.3.	Base de datos forest.	9
2.4.4.	Base de datos Iris.	10
2.4.5.	Base de datos Digits.	11
3.	Referencias bibliográficas.	13



1. Descripción del proceso para llevar a cabo el entrenamiento de las redes RBF.

En el proceso para llevar a cabo el entrenamiento de las redes RBF se utilizará un entrenamiento híbrido, que tiene una parte no supervisada (clustering) y otra supervisada (regresión logística o inversión de la matriz).

- Fase 1: ajuste de los centros.

El ajuste de los centros de la red puede hacerse mediante un procedimiento clustering. Se coloca una RBF en cada cluster y se utiliza el algoritmo de clustering K-medias. Tras el proceso de clustering, los centroides de cada cluster serán los centros de las RBF. En clasificación, escogemos número de clusters/número de clases patrones de cada clase. En regresión, escogemos aleatoriamente tantos patrones como clusters tengamos.

- Fase 2: ajuste de los radios.

Para el ajuste de los radios de cada centroide tomaremos la mitad de la distancia media al resto de centroides.

- Fase 3: pesos de la capa de salida.

- Caso 1: clasificación.

Los pesos se ajustarán utilizando regresión logística. Es un modelo lineal de clasificación, que aproxima la probabilidad de pertenencia a una clase a través de la función softmax. El objetivo es obtener los valores que maximizan la entropía cruzada. Además, utilizaremos regularización L1 y L2 para lograr que el máximo número de parámetros tienda a cero. El parámetro η establece la importancia de esta regularización.

- Caso 2: regresión.

Los pesos se ajustarán utilizando la pseudoinversa de Moore Penrose.



2. Experimentos y análisis de resultados.

2.1. Descripción de las bases de las bases de datos utilizadas.

2.1.1. Función seno.

Esta base de datos está compuesta por 120 patrones de entrenamiento y 41 patrones de test. Ha sido obtenido añadiendo cierto ruido aleatorio a la función seno. Tiene una variable independiente y otra dependiente.

2.1.2. Base de datos CPU.

Esta base de datos está compuesta por 109 patrones de entrenamiento y 100 patrones de test. Contiene, como entradas o variables independientes, MYCT, MMIN, MMAX, CACH, CHMIN, CHMAX, y, como salida o variable dependiente, el rendimiento relativo del procesador (PRP).

2.1.3. Base de datos forest.

Esta base de datos está compuesta por 387 patrones de entrenamiento y 130 patrones de test. Contiene, como entradas o variables independientes, una serie de datos meteorológicos y otras variables descriptoras (27) sobre incendios en bosques al norte de Portugal, y, como salida o variable dependiente, el área quemada.

2.1.4. Base de datos Iris.

Esta base de datos está compuesta por 112 patrones de entrenamiento y 38 patrones de test. Las clases son tres especies distintas de la flor iris, de manera que para cada flor se extraen cuatro medidas o variables de entrada (longitud y ancho de los pétalos y los sépalos, en cm). Las tres especies a distinguir son iris setosa, iris virginica e iris versicolor.



2.1.5. Base de datos Digits.

Esta base de datos está compuesta por 1274 patrones de entrenamiento y 319 patrones de test. Está formada por un conjunto de dígitos (del 0 al 9) escritos a mano por 80 personas distintas, y ajustados a una rejilla cuadrada de 16×16 píxeles. Aunque las imágenes originales estaban en escala de grises, éstas fueron binarizadas, con un valor de umbral fijo 1. Cada uno de los píxeles forman parte de las variables de entrada (256 variables) y las clases se corresponden con el dígito escrito (0, 1, ..., 9, con un total de 10 clases).

2.2. Descripción de los valores de los parámetros considerados.

Para la arquitectura de la red neuronal se han considerado los siguientes valores.

- Número de capas ocultas. Se generarán redes de una sola capa ocultas.
- Número de neuronas por capa oculta. Se van a generar redes con un número de neuronas en capa oculta igual al 5%, 10%, 25% y 50% del número de patrones de la base de datos.

La capa de entrada tendrá tantas neuronas como entradas hay en la base de datos y la de salida una, que será la salida de la red.

La actualización de pesos podremos controlarla a través de los valores de la tasa de aprendizaje η y el tipo de regularización L1 o L2.

- Tipo de regularización. Mecanismo para lograr que el máximo número de parámetros β_{ji} tiendan a valores muy pequeños. Utilizaremos los tipos de regularización L1 y L2.
- Tasa de aprendizaje (η). Establece la importancia que se le da a la regularización. Se utilizarán como valores de η , 0, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001, 0.0000001, 0.00000001 y 0.000000001.



2.3. Resultados obtenidos.

2.3.1. Función seno.

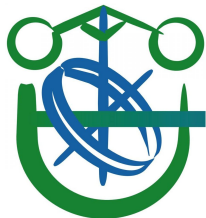
	MSE de entrenamiento		MSE de test	
	Media	σ	Media	σ
5% (6)	0.013816	0.000045	0.022342	0.000169
10% (12)	0.012473	0.000024	0.037887	0.002989
25% (30)	0.010363	0.000011	2.019609	0.553249
50% (60)	0.010357	0.000006	2.357787	0.728080

2.3.2. Base de datos CPU.

	MSE de entrenamiento		MSE de test	
	Media	σ	Media	σ
5% (5)	0.004108	0.000040	0.004417	0.000084
10% (11)	0.001731	0.001380	0.003105	0.000892
25% (27)	0.000454	0.000034	0.002258	0.000107
50% (55)	0.000238	0.000054	0.004520	0.002438

2.3.3. Base de datos forest.

	MSE de entrenamiento		MSE de test	
	Media	σ	Media	σ
5% (19)	0.001514	0.000007	0.009004	0.000041
10% (39)	0.001441	0.000070	0.009046	0.000164
25% (97)	0.001254	0.000075	0.009697	0.000269
50% (194)	0.001008	0.000078	0.011849	0.000333

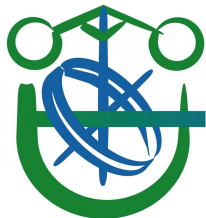


2.3.4. Base de datos Iris.

	MSE de entrenamiento		MSE de test		CCR de entrenamiento		CCR de test	
	Media	σ	Media	σ	Media	σ	Media	σ
6	0.0286	0.0067	0.0737	0.0482	97.143	0.6682	92.632	4.8238
11	0.0125	0.0071	0.0737	0.0197	98.75	0.7143	92.632	1.9693
28	0	0	0.0895	0.0316	100	0	91.053	3.1579
56	0	0	0.1053	0	100	0	89.474	0

Como podemos observar la mejor configuración ejecutada es aquella con un número de neuronas en capa oculta igual al 10% del número de patrones de la base de datos. Ahora cambiaremos el valor de eta y el tipo de regularización.

	Entrenamiento (L1)		Test (L1)		Entrenamiento (L2)		Test (L2)	
	Media	σ	Media	σ	Media	σ	Media	σ
0	33.036	0	34.211	0	85.179	0	97.895	0
1^{-1}	96.429	0	95.79	1.2892	96.429	0.5647	98.421	1.2892
1^{-2}	98.036	0.3572	94.737	0	96.429	0	95.79	1.2892
1^{-3}	98.036	0.3571	93.158	1.2892	97.5	0.6682	94.737	0
1^{-5}	98.750	0.7143	93.684	2.1053	98.036	0.3571	93.158	1.2892
1^{-6}	98.929	0.6682	94.211	1.9693	99.107	0.7986	92.105	2.3538
1^{-7}	98.571	0.4374	94.211	1.9693	99.464	0.7143	93.158	3.9386
1^{-8}	33.036	0	34.211	0	85.179	1.2111	97.895	1.9693
1^{-9}	33.036	0	34.211	0	85.179	1.2111	97.898	1.9693

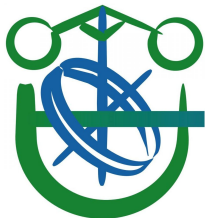


2.3.5. Base de datos Digits.

	MSE de entrenamiento		MSE de test		CCR de entrenamiento		CCR de test	
	Media	σ	Media	σ	Media	σ	Media	σ
64	0.2974	0.095	1.6633	0.2545	98.684	0.3418	91.787	0.9172
127	0	0	1.1392	0.1487	100	0	94.734	0.5016
319	0	0	1.2138	0.0803	100	0	95.047	0.3656
637	0	0	1.2383	0.1494	100	0	94.295	0.5016

Como podemos observar la mejor configuración ejecutada es aquella con un número de neuronas en capa oculta igual al 25% del número de patrones de la base de datos. Ahora cambiaremos el valor de eta y el tipo de regularización.

	Entrenamiento (L1)		Test (L1)		Entrenamiento (L2)		Test (L2)	
	Media	σ	Media	σ	Media	σ	Media	σ
0	10.047	0	10.345	0	45.84	0.626	44.389	0.6455
1^{-1}	99.105	0.2360	94.984	0.3965	97.253	0.2047	93.730	0.2804
1^{-2}	100	0	94.922	0.4159	99.702	0.0587	94.922	0.3072
1^{-3}	100	0	94.796	0.3762	100	0	94.859	0.1536
1^{-5}	100	0	94.671	0.2804	100	0	94.922	0.2346
1^{-6}	100	0	94.859	0.3762	100	0	94.734	0.3072
1^{-7}	100	0	94.859	0.3762	100	0	94.608	0.2346
1^{-8}	10.047	0	10.345	0	45.84	0.626	44.389	0.6455
1^{-9}	10.047	0	10.345	0	45.84	0.626	44.389	0.6455



2.4. Análisis de resultados.

2.4.1. Función seno.

En la función seno podemos observar como el error medio para el conjunto de entrenamiento disminuye aumentando el número de neuronas en capa oculta. Sin embargo, para el conjunto de test llega un punto en el que empieza a empeorar. Podemos decir que se está sobreentrenando el modelo. Este punto se encuentra en la función seno por debajo de las 6 neuronas en la capa oculta.

2.4.2. Base de datos CPU.

En la base de datos CPU podemos observar como el error medio para el conjunto de entrenamiento disminuye aumentando el número de neuronas en capa oculta. Sin embargo, para el conjunto de test llega un punto en el que empieza a empeorar. Podemos decir que se está sobreentrenando el modelo. Este punto se encuentra en CPU entre 11 y 55 neuronas en la capa oculta.

2.4.3. Base de datos forest.

En la base de datos forest podemos observar como el error medio para el conjunto de entrenamiento disminuye aumentando el número de neuronas en capa oculta. Sin embargo, para el conjunto de test llega un punto en el que empieza a empeorar. Podemos decir que se está sobreentrenando el modelo. Este punto se encuentra en forest por debajo de las 19 neuronas en la capa oculta.



2.4.4. Base de datos Iris.

En la base de datos Iris podemos observar como tanto el error medio como el CCR para el conjunto de entrenamiento mejora aumentando el número de neuronas en capa oculta. Sin embargo, para el conjunto de test llega un punto en el que empieza a empeorar. Podemos decir que se está sobreentrenando el modelo. Este punto se encuentra en Iris entre 6 y 11 neuronas en la capa oculta. Se observa que al modificar el tipo de regularización el número de coeficientes se mantiene.

Si ejecutamos el script con el parámetro de clasificación a falso el CCR medio para el conjunto de test baja de 98.421% a un 94.74% y para el de entrenamiento se mantiene en un 96.429%, para la mejor configuración obtenida. Considerar la clasificación afecta positivamente a la clasificación del conjunto de test pero no para la clasificación del conjunto de entrenamiento.

El tiempo computacional necesario para entrenar la base de datos Iris con la mejor configuración obtenida es de 0.00824117660522 segundos.

La estimación de la probabilidad de pertenencia a cada una de las clases se realiza a través de la función softmax:

$$P(\mathbf{x} \in C_j) = o_j = \frac{\exp(\beta_{j0} + \sum_{i=1}^n \beta_{ji}x_i)}{\sum_{l=1}^k \exp(\beta_{l0} + \sum_{i=1}^n \beta_{li}x_i)}$$

Los coeficientes de la ecuación del mejor modelo de red neuronal obtenido para la base de datos Iris son los siguientes (12, porque se ha añadido sesgo):

-0.61918319	3.18440504	-0.08263907	1.44679202
0.50036907	0.50061604	-0.4614048	-3.02674271
-0.96870941	0.74322737	-2.31618585	1.34494204



A continuación mostraremos la matriz de confusión en test del mejor modelo de red neuronal obtenido para esta base de datos.

		Clase estimada		
		0	1	2
Clase real	0	13		
	1		12	
	2			13

La clasificación del conjunto de test es perfecta, no se comete ningún error de clasificación.

2.4.5. Base de datos Digits.

En la base de datos Digits podemos observar como tanto el error medio como el CCR para el conjunto de entrenamiento mejora aumentando el número de neuronas en capa oculta. Sin embargo, para el conjunto de test llega un punto en el que empieza a empeorar. Podemos decir que se está sobreentrenando el modelo. Este punto se encuentra en Iris entre 127 y 637 neuronas en la capa oculta.

Se observa que al modificar el tipo de regularización el número de coeficientes se mantiene.

Si ejecutamos el script con el parámetro de clasificación a falso el CCR medio para el conjunto de test baja de 94.984% a un 50.08% y para el de entrenamiento de un 99.105% a un 42.01%, para la mejor configuración obtenida. Considerar la clasificación afecta positivamente a la clasificación de ambos conjuntos.
















El tiempo computacional necesario para entrenar la base de datos Digits con la mejor configuración obtenida es de 2.27336812019 segundos.



A continuación mostraremos la matriz de confusión en test del mejor modelo de red neuronal obtenido para esta base de datos.

		Clase estimada									
		0	1	2	3	4	5	6	7	8	9
Clase real	0	32								1	
	1		29	1		1					2
	2			32							
	3				31						1
	4					31			1		
	5			1			30				
	6							32			
	7					1			29	1	1
	8									30	1
	9			1	1		1				28

Ahora veremos qué dígitos ha clasificado mal, cómo los ha clasificado y qué representaban en realidad:

 (0, como 8).  (8, como 9).  (5, como 2).  (3, como 9).
 (1, como 4).  (1, como 9).  (1, como 9).  (1, como 2).
 (4, como 7).  (9, como 2).  (9, como 5).  (9, como 3).
 (7, como 4).  (7, como 9).  (7, como 8).

Podemos observar como el dígito en el que encuentra mayor dificultad a la hora de clasificar es en el 1. Esto puede ser debido a que es el número que más representaciones posibles tiene.



3. Referencias bibliográficas.

- <https://en.wikipedia.org/>
- <http://scikit-learn.org/>
- <https://www.scipy.org/>
- <http://archive.ics.uci.edu/>
- <https://msdn.microsoft.com/>
- <http://www.lab.inf.uc3m.es/>
- Auer, Peter; Harald Burgsteiner; Wolfgang Maass (2008). «A learning rule for very simple universal approximators consisting of a single layer of perceptrons». Neural Networks 21 (5): 786-795.
- Roman M. Balabin, Ravilya Z. Safieva, and Ekaterina I. Lomakina (2007). «Comparison of linear and nonlinear calibration models based on near infrared (NIR) spectroscopy data for gasoline properties prediction». Chemometr Intell Lab 88 (2): 183-188.
- Meshfree Approximation Methods with MATLAB, Gregory E. Fasshauser. Illinois Institute of Technology, USA.
- Redes de Neuronas de Base Radial, Redes de Neuronas Artificiales, UC3M, RAI 2012.
- Algoritmo K-medias, Tutorial sobre su funcionamiento a través de una aplicación.