



UNIVERSIDAD DE CÓRDOBA  
ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA (EPSC)

GRADO EN INGENIERÍA INFORMÁTICA  
ESPECIALIDAD: COMPUTACIÓN  
CUARTO CURSO. PRIMER CUATRIMESTRE

INTRODUCCIÓN A LOS MODELOS COMPUTACIONALES (IMC).

## Práctica 3. Redes neuronales de funciones de base radial.

*Adrián López Ortiz*

*DNI: 46265190T*

[p42loora@uco.es](mailto:p42loora@uco.es)

Curso académico 2019-2020  
Córdoba, 18 de noviembre de 2019

# Contenidos

Índice de figuras	2
Índice de tablas	3
Índice de algoritmos	4
1. Descripción de los pasos a realizar para llevar a cabo el entrenamiento de las redes RBF	5
2. Experimentos y análisis de resultados	7
2.1. Descripción de las bases de datos . . . . .	7
2.2. Descripción de los valores de los parámetros considerados. . . . .	8
2.3. Resultados de los experimentos . . . . .	9
2.3.1. Variaciones del ratio de RBFs (5,15,25 y 50 %), con Regulación L1 y $\eta = 10^5$ para todas las bases de datos. . . . .	9
2.3.2. Variar la $\eta$ y el tipo de regulación para los problemas de clasificación, usando la mejor arquitectura obtenida para cada uno de ellos. . . . .	11
2.3.3. Comparación de resultados entre distintas inicializaciones para la función KMeans. . . . .	13
2.3.4. Calculando para la mejor arquitectura de la base de datos de clasificación noMNIST el CCR como si fuera de regresión. . . . .	15
2.3.5. Comparación de las matrices de confusión en noMNIST. . . . .	15
Referencias bibliográficas	16

## Índice de figuras

1.	Diferencia entre funciones de proyección, izqda., y funciones locales, drcha.. . . .	5
2.	Matriz confusión práctica 3 . . . . .	15
3.	Matriz confusión práctica 2 . . . . .	15

## Índice de tablas

1.	Variaciones del ratio RBFs en la Función Seno. . . . .	10
2.	Variaciones del ratio RBFs en la Base de datos Quake. . . . .	10
3.	Variaciones del ratio RBFs en la Base de datos Parkinsons. . . . .	10
4.	Variaciones del ratio RBFs en la Base de datos noMNIST. . . . .	11
5.	Variaciones del ratio RBFs en la Base de datos Vote. . . . .	11
6.	Variaciones de $\eta$ para la regularización L1 en la base de datos noMNIST. . . . .	12
7.	Variaciones de $\eta$ para la regularización L2 en la base de datos noMNIST. . . . .	12
8.	Variaciones de $\eta$ para la regularización L1 en la base de datos vote. . . . .	13
9.	Variaciones de $\eta$ para la regularización L2 en la base de datos vote. . . . .	13
10.	Comparación entre distintas inicializaciones en la función Seno. . . . .	14
11.	Comparación entre distintas inicializaciones en la base de datos Quake. . . . .	14
12.	Comparación entre distintas inicializaciones en la base de datos Parkinsons. . . . .	14
13.	Comparación entre distintas inicializaciones en la base de datos vote. . . . .	14
14.	Comparación entre distintas inicializaciones en la base de datos noMNIST. . . . .	14
15.	Comparación de resultados entre regresión y clasificación en la base de datos noMNIST. . . . .	15

## Índice de algoritmos

## 1. Descripción de los pasos a realizar para llevar a cabo el entrenamiento de las redes RBF

La red neuronal implementada en esta práctica está compuesta por nodos **RBF** (Radial based functions). Las **neuronas de capa oculta** de este tipo de red neuronal son **funciones de base radial**, es decir, **funciones locales**.

En cambio, en las redes neuronales que vimos en las anteriores prácticas, (**Perceptrón Multi-capas**), se usan en las neuronas de capa oculta funciones de proyección, es decir, de tipo Sigmoide.

Mientras que **las funciones de proyección** hacen una división del espacio de entrada en regiones amplias (mediante una frontera, que puede ser una función lineal o no lineal), **las funciones locales** dividen el espacio de entrada en regiones locales.

A continuación, en la imagen 1 se muestra claramente la diferencia entre las funciones de proyección ya usadas con las funciones locales que usaremos a lo largo de esta práctica.

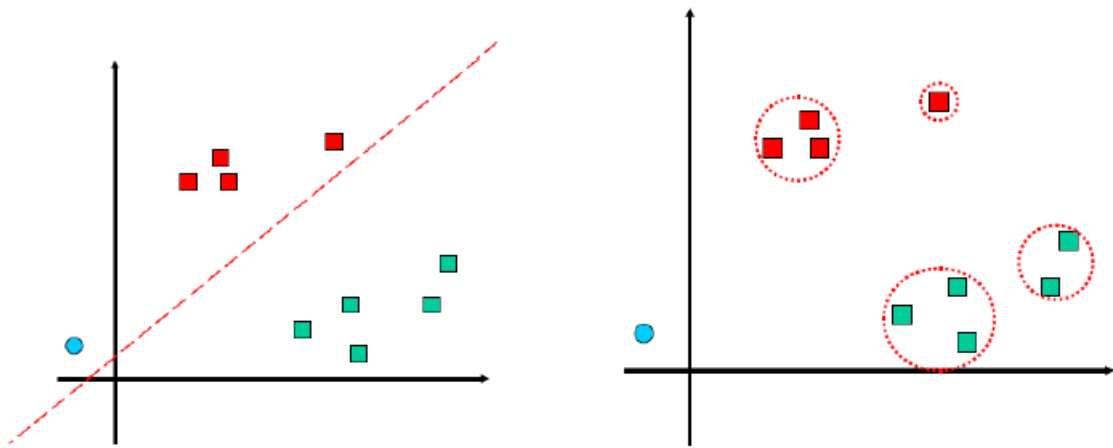


Figura 1: Diferencia entre funciones de proyección, izqda., y funciones locales, drcha..

Dichas funciones RBF dependen de la distancia que hay entre la entrada y un centro. Cada vez que se presente un nuevo patrón, se calcula la distancia a cada uno de los centros existentes. Se activará la neurona oculta correspondiente con el centro a menor distancia de la entrada. Para calcular la distancia de un patrón entre centros se usarán los radios.

En las anteriores prácticas, con el Perceptrón se entrenaba a la red con el algoritmo de retro-propagación del error, el cual contenía derivadas, pero cómo en este caso, calcular las derivadas respecto a cada centro es un ejercicio de computación muy costoso se propone un método híbrido para realizar el entrenamiento:

- En primer lugar, se hará *Clustering* mediante un algoritmo, véase *KMeans*, para calcular la posición de los centros. En este punto hay que asignar un número concreto de RBFs, en función de los patrones de la base de datos y del ratio de RBF respecto al número de patrones. Este paso es de vital importancia para el entrenamiento de la red.
- Luego, se ajustarán los radios de cada centro creado, teniendo en cuenta que ya que estamos usando los radios, se deberá de tomar el cociente de la distancia media de cada centro respecto al resto de estos, dividida entre dos.
- Por último, obtendremos los pesos de la capa de salida de dos formas: mediante regresión logística o mediante la inversa o la pseudoinversa de Penrose, (la inversa solo en el improbable caso de que tengamos una matriz cuadrada).

## 2. Experimentos y análisis de resultados

En este apartado se expondrá todos los recursos utilizados durante los experimentos para obtener el mejor modelo de red RBF, tanto las bases de datos utilizadas como los parámetros considerados y sus valores. También se mostrarán los resultados obtenidos de estos experimentos y se hará un análisis de estos.

### 2.1. Descripción de las bases de datos

Durante la práctica se trabaja con todas las bases de datos con las que hemos trabajado en las prácticas anteriores, excepto la del problema XOR, para poner a prueba la red y sacar conclusiones sobre que tipo de modelo de este es mejor. Las bases de datos utilizadas son:

- **Función Seno:** esta base de datos está compuesta por 120 patrones de train y 41 patrones de test. Ha sido obtenido añadiendo ruido aleatorio a la función seno.
- **Base de datos Quake:** esta base de datos está compuesta por 1633 patrones de train y 546 patrones de test. Se corresponde con una base de datos en la que el objetivo es averiguar la fuerza de un terremoto (medida en escala sismológica en Richter). Como variables de entrada, utilizamos la profundidad focal, la latitud en la que se produce y la longitud.
- **Base de datos Parkinsons:** esta base de datos está compuesta por 4406 patrones de train y 1469 patrones de test. Contiene, como entradas o variables independientes, una serie de datos clínicos de pacientes con la enfermedad de Parkinson y datos de medidas biométricas de la voz, y como salidas o variables dependientes, el valor motor y total el UPDRS (de las siglas en inglés Unified Parkinson's Disease Rating Scale).
- **Base de datos *vote*:** contiene 326 patrones de entrenamiento y 109 patrones de test. La base de datos incluye los votos para cada uno de los para cada uno de los candidatos para el Congreso de los EEUU, identificados por la CQA. Todas las variables de entrada son categóricas.
- **Base de datos *noMNIST*:** originalmente, esta base de datos estaba compuesta por 200,000 patrones de entrenamiento y 10,000 patrones de test, con un total de 10 clases. No obstante, para la práctica que nos ocupa, se ha reducido considerablemente el tamaño de la base de datos con objeto de realizar las pruebas en menor tiempo. Por lo tanto, la base de datos que se utilizará está compuesta por 900 patrones de entrenamiento y 300 patrones de test. Está formada por un conjunto de letras (de la a a la f ) escritas con diferentes tipografía y



simbologías. Están ajustadas a una rejilla cuadrada de  $28 \times 28$  píxeles. Las imágenes están en escala de grises en el intervalo  $[1,0; +1,0]$ . Cada uno de los píxeles es una variable de entrada (con un total de  $28 \times 28 = 784$  variables de entrada) y las clases se corresponden con la letra escrita (a, b, c, d, e y f, con un total de 6 clases). La figura 1 representa un subconjunto de los 180 patrones del conjunto de entrenamiento, mientras que la figura 2 representa un subconjunto de 180 letras del conjunto de test. Además, todas las letras, ordenadas dentro de cada conjunto, están colgadas en la plataforma Moodle en los ficheros *train img mnist.tar.gz* y *test img mnist.tar.gz*.

## 2.2. Descripción de los valores de los parámetros considerados.

Cada experimento se llevará a cabo con cinco diferentes semillas (1,2,3,4,5); de las cuáles se obtendrá la media y la desviación típica del error. Dicho error se calculará mediante la fórmula del MSE (Error cuadrático medio). También se calculará el CCR, todo esto, tanto para el conjunto de *train* como para el de *test*.

A continuación, se explicarán cada uno de los diez argumentos que puede recibir el programa por la línea de comandos:

- **Argumento -t, -train file:** Indica el nombre del fichero que contiene los datos de entrenamiento a utilizar. Sin este argumento, el programa no puede funcionar.
- **Argumento -T, -test file:** Indica el nombre del fichero que contiene los datos de test a utilizar. Si no se especifica este argumento, utilizar los datos de entrenamiento como test.
- **Argumento -c, -classification:** Booleano que indica si el problema es de clasificación. Si no se especifica, supondremos que el problema es de regresión.
- **Argumento -r, -ratio rbf:** Indica la razón (en tanto por uno) de neuronas RBF con respecto al total de patrones en entrenamiento. Si no se especifica, utilizar 0,1 capa oculta.
- **Argumento -l, -l2:** Booleano que indica si utilizaremos regularización de L2 en lugar de la regularización L1. Si no se especifica, supondremos que regularización L1.
- **Argumento -e, -eta:** Indica el valor del parámetro eta ( $\eta$ ). Por defecto, utilizar  $\eta = 1e-2$ .
- **Argumento -o, -outputs:** Indica el número de columnas de salida que tiene el conjunto de datos y que siempre están al final. Por defecto, utilizar  $o = 1$ .
- **(Kaggle) Argumento -p, -pred:** Booleano que indica si utilizaremos el modo de predicción.

- **(Kaggle) Argumento -m, -model file:** Indica el directorio en el que se guardarán los modelos entrenados (en el modo de entrenamiento, sin el flag p) o el fichero que contiene el modelo que se utilizará (en el modo de predicción, con el flag p).
- **Argumento -help:** Mostrar la ayuda del programa (utilizar la que genera automáticamente la librería *click*).

A continuación, mostraremos la arquitectura de la red así como los experimentos a realizar en esta.

### Arquitectura de la red:

- Para todas las bases de datos, considerar un número de neuronas en capa oculta ( $n_1$ ) igual al 5 %, 15 %, 25 % y 50 % del número de patrones de la base de datos. En esta fase, para problemas de clasificación, utilizar regularización L1 y un valor para el parámetro  $\eta = 10^5$ .
- Para los problemas de clasificación, una vez decidida la mejor arquitectura, probar los siguientes valores para  $\eta$ :  $\eta = 1$ ,  $\eta = 0.1$ ,  $\eta = 0.01$ ,  $\eta = 0.001$ , ... ,  $\eta = 10^{10}$ , junto con los dos tipos de regularización (L2 y L1). ¿Qué sucede?. Calcula la diferencia en número de coeficientes en *vote* y *noMNIST* cuando modificas el tipo de regularización (L2 Vs L1).
- Para problemas de regresión y de clasificación, comparar los resultados obtenidos con la inicialización propuesta para el algoritmo *sklearn.cluster.KMeans* (usando la mejor arquitectura y la mejor configuración para la regresión logística) con respecto a la inicialización “*k-means++*”.
- Finalmente, en alguno de los problemas de clasificación, probar a lanzar el script considerando el problema como si fuera un problema de regresión (es decir, incluyendo un *False* en el parámetro clasificación y calculando el CCR redondeando las predicciones hasta el entero más cercano). ¿Qué sucede en este caso?.

## 2.3. Resultados de los experimentos

A continuación se mostrarán los resultados obtenidos al realizar los experimentos exigidos por el guión de la práctica.

### 2.3.1. Variaciones del ratio de RBFs (5,15,25 y 50 %), con Regulación L1 y $\eta = 10^5$ para todas las bases de datos.

**-Función Seno:**

Ratio RBFs	MSE train	MSE test
5 %	0.013817 +- 0.000189	0.022182 +- 0.000245
15 %	0.011899 +- 0.000256	0.116910 +- 0.043303
25 %	0.011819 +- 0.000314	0.117961 +- 0.045791
50 %	0.011529 +- 0.000302	0.183935 +- 0.066504

Cuadro 1: Variaciones del ratio RBFs en la Función Seno.

Podemos ver que la arquitectura que menor MSE de test obtiene y por tanto, la más óptima para este problema es la que tiene un 5 % de Ratio RBFs.

**-Base de datos Quake:**

Ratio RBFs	MSE train	MSE test
5 %	0.028365 +- 0.000063	0.028475 +- 0.000223
15 %	0.026746 +- 0.000226	0.032581 +- 0.000994
25 %	0.026382 +- 0.000126	0.034395 +- 0.000413
50 %	0.026141 +- 0.000024	0.035853 +- 0.000408

Cuadro 2: Variaciones del ratio RBFs en la Base de datos Quake.

Podemos ver que la arquitectura que menor MSE de test obtiene y por tanto, la más óptima para este problema es la que tiene un 5 % de Ratio RBFs.

**-Base de datos parkinsons:**

Ratio RBFs	MSE train	MSE test
5 %	0.021629 +- 0.000247	0.024740 +- 0.000333
15 %	0.013970 +- 0.000288	0.019804 +- 0.000412
25 %	0.010341 +- 0.000080	0.022338 +- 0.001554
50 %	0.005221 +- 0.000087	0.054642 +- 0.004818

Cuadro 3: Variaciones del ratio RBFs en la Base de datos Parkinsons.

Podemos ver que la arquitectura que menor MSE de test obtiene y por tanto, la más óptima para este problema es la que tiene un 15 % de Ratio RBFs.

**-Base de datos noMNIST:**

Ratio RBFs	MSE train	MSE test	CCR train	CCR test
5 %	0.030191 +- 0.001527	0.030285 +- 0.001682	87.93 % +- 0.78 %	88.53 % +- 1.15 %
15 %	0.000510 +- 0.000133	0.042340 +- 0.002695	100.00 % +- 0.00 %	86.53 % +- 1.05 %
25 %	0.000036 +- 0.000005	0.038075 +- 0.001651	100.00 % +- 0.00 %	87.33 % +- 0.56 %
50 %	0.000017 +- 0.000001	0.036296 +- 0.002478	100.00 % +- 0.00 %	87.87 % +- 1.00 %

Cuadro 4: Variaciones del ratio RBFs en la Base de datos noMNIST.

Podemos ver que la arquitectura que mayor CCR de test obtiene y por tanto, la más óptima para este problema es la que tiene un 5 % de Ratio RBFs.

**-Base de datos vote:**

Ratio RBFs	MSE train	MSE test	CCR train	CCR test
5 %	0.026243 +- 0.003441	0.035143 +- 0.001805	96.13 % +- 0.69 %	95.96 % +- 0.45 %
15 %	0.003583 +- 0.000001	0.067280 +- 0.010021	99.39 % +- 0.00 %	92.29 % +- 1.24 %
25 %	0.003581 +- 0.000001	0.058070 +- 0.005682	99.39 % +- 0.00 %	93.58 % +- 0.58 %
50 %	0.003580 +- 0.000000	0.057729 +- 0.000939	99.39 % +- 0.00 %	94.13 % +- 0.45 %

Cuadro 5: Variaciones del ratio RBFs en la Base de datos Vote.

Podemos ver que la arquitectura que mayor CCR de test obtiene y por tanto, la más óptima para este problema es la que tiene un 5 % de Ratio RBFs.

**2.3.2. Variar la  $\eta$  y el tipo de regulación para los problemas de clasificación, usando la mejor arquitectura obtenida para cada uno de ellos.**

Para los problemas de clasificación, una vez decidida la mejor arquitectura, probar los siguientes valores para  $\eta$ :  $\eta = 1$ ,  $\eta = 0.1$ ,  $\eta = 0.01$ ,  $\eta = 0.001$ , ... ,  $\eta = 10^{10}$ , junto con los dos tipos de regularización (L2 y L1). ¿Qué sucede?. Calcula la diferencia en número de coeficientes en *vote* y *noMNIST* cuando modificas el tipo de regularización (L2 Vs L1).

$\eta$	Coficiente	MSE train	MSE test	CCR train	CCR test
1	1	0.054578 +- 0.000689	0.051867 +- 0.001099	80.51 % +- 0.47 %	82.80 % +- 0.98 %
1e-1	10	0.035242 +- 0.001269	0.032533 +- 0.001539	86.40 % +- 0.68 %	89.13 % +- 1.00 %
1e-2	10e2	0.030856 +- 0.001515	0.030002 +- 0.001611	87.84 % +- 0.69 %	88.93 % +- 1.06 %
1e-3	10e3	0.030252 +- 0.001527	0.030245 +- 0.001691	87.87 % +- 0.81 %	88.67 % +- 1.21 %
1e-4	10e4	0.030195 +- 0.001529	0.030281 +- 0.001685	87.93 % +- 0.78 %	88.53 % +- 1.15 %
1e-5	10e5	0.030191 +- 0.001527	0.030285 +- 0.001682	87.93 % +- 0.78 %	88.53 % +- 1.15 %
1e-6	10e6	0.030191 +- 0.001527	0.030285 +- 0.001682	87.93 % +- 0.78 %	88.53 % +- 1.15 %
1e-7	10e7	0.030191 +- 0.001527	0.030285 +- 0.001682	87.93 % +- 0.78 %	88.53 % +- 1.15 %
1e-8	10e8	0.030190 +- 0.001527	0.030285 +- 0.001682	87.93 % +- 0.78 %	88.53 % +- 1.15 %
1e-9	10e9	0.030190 +- 0.001527	0.030285 +- 0.001682	87.93 % +- 0.78 %	88.53 % +- 1.15 %
1e-10	10e10	0.030190 +- 0.001527	0.030285 +- 0.001682	87.93 % +- 0.78 %	88.53 % +- 1.15 %

Cuadro 6: Variaciones de  $\eta$  para la regularización L1 en la base de datos noMNIST.

$\eta$	Coficiente	MSE train	MSE test	CCR train	CCR test
1	1	0.067085 +- 0.000893	0.065913 +- 0.001188	77.67 % +- 0.40 %	77.47 % +- 1.65 %
1e-1	10	0.045423 +- 0.000783	0.043739 +- 0.001242	83.36 % +- 0.40 %	86.40 % +- 1.18 %
1e-2	10e2	0.035838 +- 0.001008	0.033370 +- 0.001372	86.22 % +- 0.66 %	89.00 % +- 0.56 %
1e-3	10e3	0.031633 +- 0.001434	0.030144 +- 0.001554	87.40 % +- 0.82 %	89.07 % +- 0.98 %
1e-4	10e4	0.030373 +- 0.001514	0.030172 +- 0.001682	87.80 % +- 0.79 %	88.67 % +- 1.17 %
1e-5	10e5	0.030185 +- 0.001521	0.030305 +- 0.001692	87.96 % +- 0.82 %	88.53 % +- 1.15 %
1e-6	10e6	0.030167 +- 0.001521	0.030323 +- 0.001687	88.00 % +- 0.81 %	88.60 % +- 1.04 %
1e-7	10e7	0.030166 +- 0.001522	0.030326 +- 0.001679	87.98 % +- 0.81 %	88.60 % +- 1.04 %
1e-8	10e8	0.030164 +- 0.001522	0.030330 +- 0.001684	88.00 % +- 0.80 %	88.53 % +- 1.15 %
1e-9	10e9	0.030166 +- 0.001521	0.030325 +- 0.001682	88.00 % +- 0.80 %	88.53 % +- 1.15 %
1e-10	10e10	0.030163 +- 0.001522	0.030325 +- 0.001680	88.00 % +- 0.81 %	88.60 % +- 1.04 %

Cuadro 7: Variaciones de  $\eta$  para la regularización L2 en la base de datos noMNIST.

$\eta$	Coficiente	MSE train	MSE test	CCR train	CCR test
1	1	0.041246 +- 0.002281	0.045207 +- 0.001890	94.54 % +- 0.30 %	95.05 % +- 0.45 %
1e-1	10	0.029082 +- 0.002738	0.033900 +- 0.001612	95.71 % +- 0.51 %	96.15 % +- 0.37 %
1e-2	10e2	0.026545 +- 0.003358	0.034342 +- 0.001160	96.13 % +- 0.63 %	95.96 % +- 0.45 %
1e-3	10e3	0.026273 +- 0.003423	0.035034 +- 0.001692	96.07 % +- 0.68 %	95.96 % +- 0.45 %
1e-4	10e4	0.026251 +- 0.003429	0.035114 +- 0.001765	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-5	10e5	0.026243 +- 0.003441	0.035143 +- 0.001805	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-6	10e6	0.026242 +- 0.003442	0.035143 +- 0.001805	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-7	10e7	0.026242 +- 0.003442	0.035143 +- 0.001805	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-8	10e8	0.026242 +- 0.003442	0.035143 +- 0.001805	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-9	10e9	0.026242 +- 0.003442	0.035143 +- 0.001805	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-10	10e10	0.026242 +- 0.003442	0.035143 +- 0.001805	96.13 % +- 0.69 %	95.96 % +- 0.45 %

Cuadro 8: Variaciones de  $\eta$  para la regularización L1 en la base de datos vote.

$\eta$	Coficiente	MSE train	MSE test	CCR train	CCR test
1	1	0.053558 +- 0.001352	0.052546 +- 0.001058	93.25 % +- 0.58 %	93.21 % +- 0.45 %
1e-1	10	0.035543 +- 0.001928	0.039284 +- 0.002148	95.46 % +- 0.45 %	95.41 % +- 0.82 %
1e-2	10e2	0.029088 +- 0.002771	0.034598 +- 0.001412	95.58 % +- 0.57 %	96.33 % +- 0.00 %
1e-3	10e3	0.026734 +- 0.003232	0.034480 +- 0.001114	96.13 % +- 0.63 %	95.96 % +- 0.45 %
1e-4	10e4	0.026282 +- 0.003410	0.035050 +- 0.001715	96.07 % +- 0.68 %	95.96 % +- 0.45 %
1e-5	10e5	0.026243 +- 0.003443	0.035139 +- 0.001757	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-6	10e6	0.026240 +- 0.003446	0.035168 +- 0.001794	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-7	10e7	0.026239 +- 0.003446	0.035169 +- 0.001795	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-8	10e8	0.026239 +- 0.003446	0.035169 +- 0.001795	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-9	10e9	0.026239 +- 0.003446	0.035170 +- 0.001795	96.13 % +- 0.69 %	95.96 % +- 0.45 %
1e-10	10e10	0.026239 +- 0.003446	0.035170 +- 0.001795	96.13 % +- 0.69 %	95.96 % +- 0.45 %

Cuadro 9: Variaciones de  $\eta$  para la regularización L2 en la base de datos vote.

El coeficiente en ambas regularizaciones es la misma, ya que se utiliza para ambas la misma:

$$C = 1 / \eta$$

### 2.3.3. Comparación de resultados entre distintas inicializaciones para la función KMeans.

**-Función Seno:**

Kmeans	MSE train	MSE test
old	0.013817 +- 0.000189	0.022182 +- 0.000245
new	0.013800 +- 0.000108	0.022301 +- 0.000292

Cuadro 10: Comparación entre distintas inicializaciones en la función Seno.

**-Base de datos Quake:**

Kmeans	MSE train	MSE test
old	0.028365 +- 0.000063	0.028475 +- 0.000223
new	0.028215 +- 0.000051	0.028689 +- 0.000124

Cuadro 11: Comparación entre distintas inicializaciones en la base de datos Quake.

**-Base de datos Parkinsons:**

Kmeans	MSE train	MSE test
old	0.013970 +- 0.000288	0.019804 +- 0.000412
new	0.014983 +- 0.000093	0.021708 +- 0.000619

Cuadro 12: Comparación entre distintas inicializaciones en la base de datos Parkinsons.

**-Base de datos Vote:**

Kmeans	MSE train	MSE test	CCR train	CCR test
old	0.026243 +- 0.003441	0.035143 +- 0.001805	96.13 % +- 0.69 %	95.96 % +- 0.45 %
new	0.027882 +- 0.004197	0.043090 +- 0.009777	96.07 % +- 0.81 %	94.86 % +- 1.10 %

Cuadro 13: Comparación entre distintas inicializaciones en la base de datos vote.

**-Base de datos noMNIST:**

Kmeans	MSE train	MSE test	CCR train	CCR test
old	0.030191 +- 0.001527	0.030285 +- 0.001682	87.93 % +- 0.78 %	88.53 % +- 1.15 %
new	0.028673 +- 0.000568	0.029427 +- 0.000902	88.18 % +- 0.57 %	88.27 % +- 0.93 %

Cuadro 14: Comparación entre distintas inicializaciones en la base de datos noMNIST.

2.3.4. Calculando para la mejor arquitectura de la base de datos de clasificación noMNIST el CCR como si fuera de regresión.

Kmeans	MSE train	MSE test	CCR train	CCR test
clasificacion	0.030191 +- 0.001527	0.030285 +- 0.001682	87.93 % +- 0.78 %	88.53 % +- 1.15 %
regresion	0.871545 +- 0.032224	0.920240 +- 0.043723	48.29 % +- 2.62 %	48.73 % +- 4.16 %

Cuadro 15: Comparación de resultados entre regresión y clasificación en la base de datos noMNIST.

Cómo podemos observar, los datos obtenidos para el CCR para regresión son muy malos.

2.3.5. Comparación de las matrices de confusión en noMNIST.

```
Matriz de confusión del Test
[[46  1  1  0  1  1]
 [ 2 44  0  2  2  0]
 [ 0  1 46  0  1  2]
 [ 1  3  0 44  0  2]
 [ 0  1  2  0 44  3]
 [ 2  1  1  0  0 46]]
```

Figura 2: Matriz confusión práctica 3

```
adrian@adrian-UBUNTU:~/Escritorio/IMC/p2/practica2/Debug$ ./practica2 -t /home/a
drian/Escritorio/IMC/p2/basesDatosPr2IMC/dat/train_nomnist.dat -T /home/adrian/E
scritorio/IMC/p2/basesDatosPr2IMC/dat/test_nomnist.dat -l 1 -h 64 -v 0 -d 2 -f 1
-s
0.00289731;0.000222087;0.100629;0.0135516;100;0;85.4;1.73845
Matriz de confusión
| 46 0 0 2 0 2 |
| 4 37 2 2 4 1 |
| 0 0 43 1 4 2 |
| 1 4 2 42 0 1 |
| 1 5 2 1 40 1 |
| 3 0 1 3 0 43 |
adrian@adrian-UBUNTU:~/Escritorio/IMC/p2/practica2/Debug$
```

Figura 3: Matriz confusión práctica 2

Podemos concluir que con esta red clasifica mejor para esta base de datos.



## Referencias bibliográficas

- [1] Servicio on-line proporcionado por la web del periódico digital el País para acortar URLs. [Consulta: Viernes, 17 de Noviembre de 2019]. Disponible en: <http://cortas.elpais.com/>
- [2] César Hervás-Martínez. Asignatura: Introducción al Aprendizaje Automático(IAA). Curso académico 2018-19.*Introducción a las Redes Neuronales Artificiales (Parte I)*. [Consulta: Viernes, 17 de Noviembre de 2019]. Disponible en: <http://cort.as/-S94w>
- [3] César Hervás-Martínez. Asignatura: Introducción al Aprendizaje Automático(IAA). Curso académico 2018-19.*Introducción a las Redes Neuronales Artificiales (Parte II)*. [Consulta: Viernes, 17 de Noviembre de 2019]. Disponible en: <http://cort.as/-S954>
- [4] César Hervás-Martínez. Asignatura: Introducción a los Modelos Computacionales (IMC). Curso académico 2019-20.*Algoritmo de retropropagación del error*. [Consulta: Viernes, 17 de Noviembre de 2019]. Disponible en: <http://cort.as/-S95O>
- [5] Pedro Antonio Gutiérrez Peña. Asignatura: Introducción a los Modelos Computacionales (IMC). Curso académico 2019-20.*Guión de la Práctica 1*. [Consulta: Viernes, 17 de Noviembre de 2019].
- [6] Pedro Antonio Gutiérrez Peña. Asignatura: Introducción a los Modelos Computacionales (IMC). Curso académico 2019-20.*Presentación de la Práctica 1*. [Consulta: Viernes, 17 de Noviembre de 2019].
- [7] *Página web para crear tablas en LaTeX*. [Consulta: Viernes, 17 de Noviembre de 2019]. Disponible en: <https://www.tablesgenerator.com/>
- [8] Pedro Antonio Gutiérrez Peña. Asignatura: Introducción a los Modelos Computacionales (IMC). Curso académico 2019-20.*Guión de la Práctica 2*. [Consulta: Viernes, 17 de Noviembre de 2019].
- [9] Pedro Antonio Gutiérrez Peña. Asignatura: Introducción a los Modelos Computacionales (IMC). Curso académico 2019-20.*Presentación de la Práctica 2*. [Consulta: Viernes, 17 de Noviembre de 2019].
- [10] Pedro Antonio Gutiérrez Peña. Asignatura: Introducción a los Modelos Computacionales (IMC). Curso académico 2019-20.*Guión de la Práctica 3*. [Consulta: Viernes, 17 de Noviembre de 2019].
- [11] Pedro Antonio Gutiérrez Peña. Asignatura: Introducción a los Modelos Computacionales (IMC). Curso académico 2019-20.*Presentación de la Práctica 3*. [Consulta: Viernes, 17 de Noviembre de 2019].