



UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA
Universidad de Córdoba



Introducción a los Modelos Computacionales

Práctica 3

Juan José Méndez Torrero
31018712-S
i42metoj@uco.es

2 de diciembre de 2018

Índice

1. Introducción	4
2. Descripción del problema	5
3. Descripción de los parámetros	6
4. Descripción de los conjuntos de datos	7
5. Resultados	8
5.1. Regresión	8
5.1.1. Conjunto de datos SIN	8
5.1.2. Conjunto de datos QUAKE	8
5.1.3. Conjunto de datos PARKINSONS	9
5.2. Clasificación	9
5.2.1. Conjunto de datos VOTE	9
5.2.2. Conjunto de datos NOMNIST	11
6. Análisis	14
6.1. Regresión	14
6.1.1. Conjunto de datos SIN	14
6.1.2. Conjunto de datos QUAKE	14
6.1.3. Conjunto de datos PARKINSONS	15
6.2. Clasificación	16
6.2.1. Conjunto de datos VOTE	16
6.2.2. Conjunto de datos NOMNIST	17
7. Bibliografía	20

Índice de figuras

1.	Errores cometidos por el conjunto de datos SIN	8
2.	Errores cometidos por el conjunto de datos QUAKE	9
3.	Errores cometidos por el conjunto de datos PARKINSONS	9
4.	Errores cometidos por el conjunto de datos VOTE	10
5.	CCR conseguido por el conjunto de datos VOTE	10
6.	Errores cometidos por el conjunto de datos VOTE con regularización tipo L1	10
7.	Errores cometidos por el conjunto de datos VOTE con regularización tipo L2	11
8.	Errores cometidos por el conjunto de datos NOMNIST	11
9.	CCR conseguido por el conjunto de datos NOMNIST	12
10.	Errores cometidos por el conjunto de datos NOMNIST con regulari- zación tipo L1	12
11.	CCR conseguido por el conjunto de datos NOMNIST con regulariza- ción tipo L2	12
12.	Gráfica error/iteración del conjunto de datos SIN	14
13.	Gráfica error/iteración del conjunto de datos QUAKE	15
14.	Gráfica error/iteración del conjunto de datos PARKINSONS	15
15.	Gráfica error/iteración del conjunto de datos VOTE	16
16.	Gráfica CCR/iteración del conjunto de datos VOTE	16
17.	Matriz de confusión del conjunto de datos VOTE	17
18.	Gráfica error/iteración del conjunto de datos NOMNIST	18
19.	Gráfica CCR/iteración del conjunto de datos NOMNIST	18
20.	Matriz de confusión del conjunto de datos NOMNIST	19

1. Introducción

Para esta práctica, hemos utilizado el lenguaje de programación Python, el cuál incluye una serie de librerías que nos permiten realizar modelos de redes neuronales de manera fácil y sencilla. En este documento veremos cómo realizar una RBF (*Radial Basis Function*) junto con la librería *sklearn* en Python. Además, este documento se dividirá en los siguientes apartados:

- **Descripción del problema:** En este apartado daremos una breve descripción del problema que hemos tenido que abarcar.
- **Descripción de los parámetros:** En este apartado explicaremos qué parámetros son necesarios para la creación de una red neuronal basada en RBFs.
- **Descripción de los conjuntos de datos:** Aquí daremos una descripción concisa sobre qué tipo de patrones tenemos en los diferentes conjuntos de datos usados en esta práctica junto con qué tipo de problema debemos enfrentar con cada uno de ellos.
- **Resultados:** En este apartado veremos, en forma de tablas, los resultados obtenidos por nuestra red con diferentes conjuntos de datos y configuraciones.
- **Análisis:** Por último, daremos una conclusión explicando los resultados obtenidos con cada uno de los conjuntos de datos utilizados.

2. Descripción del problema

A diferencia de las dos prácticas anteriores, esta vez utilizaremos una librería de Python llamada *sklearn*, la cuál nos permite realizar algoritmos de clasificación de manera sencilla. En esta práctica tendremos que crear un tipo de red llamada RBF(Radial Basis Function), la cual se basa en una aproximación local.

Para realizar esta red, haremos uso de un entrenamiento híbrido, es decir, una parte no supervisada, que será el clustering, y otra supervisada, que será una regresión logística o una inversión de una matriz.

Nuestra arquitectura será una capa de entrada, una capa oculta y otra capa de salida. En la capa oculta habrá tantas neuronas como RBFs haya, o lo que es lo mismo, el mismo número de centros que consigamos en el clustering. Además, hemos hecho uso de la función Gaussiana :

$$\varphi(\mathbf{x}, \mathbf{c}, \sigma) = \exp\left(-\frac{\text{distancia}}{2\sigma^2}\right)$$

El parámetro *distancia* será la distancia existente entre el centro de las RBFs al resto de patrones de nuestro conjunto de datos.

Para realizar el clustering, haremos uso de la función *K-Means*, que se encuentra dentro de la librería de *sklearn*. Los centros de cada una de las RBFs serán los centroides conseguidos por el algoritmo de K-Means. Además, los radios de cada cluster se calcularán dividiendo entre dos cada una de las distancias.

Por último, necesitaremos una matriz *R* en la que guardaremos las salidas. Dependiendo del problema, clasificación o regresión, haremos uso de un algoritmo. Para clasificación haremos uso de la regularización logística, L1 o L2. En cambio, para los problemas de regresión calcularemos la matriz de salida aplicando la pseudo-inversa de Moore Penrose.

3. Descripción de los parámetros

En este apartado se explicarán los parámetros necesarios para simular el entrenamiento con nuestra red. Para ejecutarlo, vamos a necesitar lo siguiente:

1. **train_data**: Es una variable la cual contiene los datos que vamos a considerar para el conjunto train.
2. **test_data**: Variable que guarda los valores considerados para el conjunto test.
3. **l2**: Booleano que controla el tipo de regularización a utilizar en un problema de clasificación. Por defecto utilizaremos el tipo *L1*.
4. **ratio_rbf**: Es un flotante entre 0 y 1 que se utilizará para calcular el número de RBFs necesarias.
5. **classification**: Booleano que controla si queremos hacer un problema de clasificación o de regresión. Por defecto, realizaremos un problema de regresión.
6. **eta**: Learning rate.
7. **outputs**: Variable para controlar cuántas columnas de datos hay para la salida.

4. Descripción de los conjuntos de datos

En esta sección explicaremos en qué consisten los conjuntos de datos utilizados en esta práctica. Como ya hemos visto en anteriores secciones, nuestra arquitectura permite resolver un problema tanto de regresión como de clasificación, es decir, en cada uno de los conjuntos de datos se dirá si se debe utilizar un problema de regresión o de clasificación. Los conjuntos de datos son los siguientes:

- **Conjunto de datos SIN:** Este conjunto de datos consta de 120 patrones para el conjunto de train y de 41 para test. Además, consta de dos atributos, uno de entrada y otro de salida. Este conjunto de datos sólo se podrá conseguir mediante regresión, ya que los patrones no están agrupados en clases, es decir, son salidas numéricas.
- **Conjunto de datos QUAKE:** Este conjunto de datos consta de 1633 patrones para el conjunto de train y de 546 patrones para el conjunto de test. Además, consta de 3 atributos de entrada y 1 de salida. Este conjunto de datos se resolvería mediante regresión, al no clasificar nada.
- **Conjunto de datos PARKINSONS:** Este conjunto de datos consta de 4406 patrones para el conjunto de train y de 1469 patrones para el conjunto de test. Ambos constan de 19 atributos de entrada y dos de salida. Este conjunto de datos se debe de resolver utilizando la regresión.
- **Conjunto de datos VOTE:** Este conjunto de datos consta de 326 patrones para el conjunto de train y 109 patrones para el conjunto de test, junto con 16 atributos de entrada y dos de salida. Se trata de un problema de clasificación, ya que podemos clasificar al votante como republicano o demócrata.
- **Conjunto de datos Nomnist:** Este conjunto de datos consta de 900 patrones para el conjunto de train y de 300 para el conjunto de test. Además, consta de 784 atributos de entrada y 6 de salida.

5. Resultados

En esta sección recogeremos los resultados obtenidos con los distintos conjuntos de datos. Para ello, hemos escogido los siguientes parámetros por defecto:

- Si estamos en un problema de clasificación, en tipo de regresión logística que utilizaremos será de L2.
- El valor del learning rate(η), será de 10^{-5}
- El porcentaje de neuronas que vamos a utilizar será el 5 %, 10 %, 25 % y 50 % del número de patrones del conjunto de datos

En este apartado, realizaremos los experimentos primero con los conjuntos de datos que son resueltos a través de regresión y seguidamente veremos los experimentos realizados a través de clasificación.

5.1. Regresión

5.1.1. Conjunto de datos SIN

Como hasta ahora, mostraremos una tabla en la cuál podremos observar las diferencias de error entre utilizar un porcentaje de neuronas u otro. En la tabla ?? podremos observar la mejor configuración que estará señalada en rojo.

Seno	Media Train	Desviación Train	Media Test	Desviación Test
5 %	0.013754	0.000339	0.021449	0.000725
10 %	0.012627	0.000091	0.030258	0.004050
25 %	0.010342	0.000015	2.577561	0.859481
50 %	0.010213	0.000182	3.444179	4.098181

Figura 1: Errores cometidos por el conjunto de datos SIN

Como vemos, los mejores resultados obtenidos han sido con un porcentaje de neuronas del 5 %, aunque hablando sólo de error de train la mejor configuración es con el 10 %, para el conjunto de test ofrece mejores resultados el primer porcentaje(5 %).

5.1.2. Conjunto de datos QUAKE

Igual que antes, se mostrará en rojo los mejores resultados obtenidos con este conjunto de datos. En la Tabla 2 podremos observar los diferentes resultados.

Quake	Media Train	Desviación Train	Media Test	Desviación Test
5 %	0.028474	0.000030	0.028027	0.000171
10 %	0.026973	0.000044	0.031856	0.001160
25 %	0.022237	0.000112	0.824889	0.275028
50 %	0.018456	0.000177	274.954570	125.385151

Figura 2: Errores cometidos por el conjunto de datos QUAKE

Como observamos en la Tabla 2, los mejores resultados se vuelven a obtener con un 5 % del número de patrones. Además, podríamos decir que obtenemos menor error con el 50 %, pero observando el error cometido por el conjunto de test, podríamos decir que con esta última configuración nuestro modelo está no está generalizando.

5.1.3. Conjunto de datos PARKINSONS

Como hasta ahora, mostraremos los mejores resultados en rojo. En la Tabla ?? podremos ver los resultados obtenidos por las diferentes configuraciones.

Parkinsons	Media Train	Desviación Train	Media Test	Desviación Test
5 %	0.001622	0.000068	0.001942	0.000063
10 %	0.001043	0.000011	0.001492	0.000021
25 %	0.000404	0.000006	0.000999	0.000035
50 %	0.000132	0.000003	0.001338	0.000064

Figura 3: Errores cometidos por el conjunto de datos PARKINSONS

Como vemos en la Tabla 3, podríamos decir que el mejor resultado lo obtenemos utilizando como neuronas el 25 % de los patrones del conjunto, aunque también podríamos decir que con el 50 % se seguiría consiguiendo muy buenos resultados. No hemos decidido en elegir el 25 % ya que conseguimos menor error en el conjunto de test, con lo que podríamos decir que nuestro modelo generaliza bastante bien, no como pasaba con el conjunto de datos QUAKE.

5.2. Clasificación

Para los conjuntos de test mostraremos en rojo las mejores configuraciones pero teniendo en cuenta también el CCR conseguido por nuestro modelo. Además, mostraremos los resultados obtenidos cambiando el learning rate entre 1 y 10^{-10} y cambiando el tipo de regularización diferenciando entre L1 y L2.

5.2.1. Conjunto de datos VOTE

Primero hablaremos sobre el error cometido y el CCR conseguido cambiando el porcentaje de patrones utilizados como neuronas.

Como vemos en las Tablas 4 y 5, los mejores resultados los obtenemos utilizando el 5 % de los patrones. Aunque utilizando mayor porcentaje conseguimos mejor CCR

Vote	Media Train	Desviación Train	Media Test	Desviación Test
5 %	0.038650	0.010732	0.051376	0.009356
10 %	0.006748	0.001227	0.066055	0.010699
25 %	0.006135	0.000000	0.064220	0.005802
50 %	0.006135	0.000000	0.064220	0.000000

Figura 4: Errores cometidos por el conjunto de datos VOTE

Vote	Media Train	Desviación Train	Media Test	Desviación Test
5 %	96.13	1.07	94.86	0.94
10 %	99.33	0.12	93.39	1.07
25 %	99.39	0.00	93.58	0.58
50 %	99.39	0.00	93.58	0.00

Figura 5: CCR conseguido por el conjunto de datos VOTE

para el conjunto de train, con el menor porcentaje conseguimos que nuestro modelo generalice mejor, siendo mayor el CCR en el conjunto de test que en el conjunto de train.

Ahora veremos los resultados variando el tipo de regularización y el learning rate. En las Tablas 6 y 7 podremos ver los respectivos resultados.

Eta	Media Train	Media Test	Media Train	Media Test
1	0.055215	0.049541	94.48	95.05
0.1	0.044172	0.045872	95.58	95.41
0.01	0.040491	0.051376	95.95	94.86
0.001	0.038650	0.051376	96.13	94.86
0.0001	0.038650	0.051376	96.13	94.86
0.00001	0.038650	0.051376	96.13	94.86
0.000001	0.038650	0.051376	96.13	94.86
0.0000001	0.038650	0.051376	96.13	94.86
0.00000001	0.038650	0.051376	96.13	94.86
0.000000001	0.038650	0.051376	96.13	94.86
0.0000000001	0.038650	0.051376	96.13	94.86
0.00000000001	0.038650	0.051376	96.13	94.86

Figura 6: Errores cometidos por el conjunto de datos VOTE con regularización tipo L1

Como vemos en la Tabla 6, al utilizar el tipo L1 de regularización, el mejor resultado lo obtenemos con un learning rate de 10^{-5} , no variando el resultado con un learning rate menor. En cambio, como vemos en la Tabla 7, el mejor resultado lo

Eta	Media Train	Media Test	Media Train	Media Test
1	0.069939	0.073394	93.01	92.66
0.1	0.046626	0.051376	95.34	94.86
0.01	0.045399	0.045872	95.46	95.41
0.001	0.042331	0.049541	95.77	95.05
0.0001	0.039264	0.051376	96.07	94.86
0.00001	0.038650	0.051376	96.13	94.86
0.000001	0.038650	0.051376	96.13	94.86
0.0000001	0.038650	0.051376	96.13	94.86
0.00000001	0.038650	0.051376	96.13	94.86
0.000000001	0.038650	0.051376	96.13	94.86
0.0000000001	0.038650	0.051376	96.13	94.86
0.00000000001	0.038650	0.051376	96.13	94.86

Figura 7: Errores cometidos por el conjunto de datos VOTE con regularización tipo L2

conseguimos con un learning rate de 0.001. Hemos elegido esta como mejor configuración porque nos ofrece, conjuntamente, buenos resultados para los CCR tanto de train como de test.

5.2.2. Conjunto de datos NOMNIST

Para este conjunto de datos haremos como con el conjunto de datos VOTE, es decir, presentaremos primero los resultados obtenidos variando el porcentaje de patrones y seguidamente variando el learning rate junto con el tipo de regularización. En las tablas 8, 9, 10 y 11 podremos observar estos resultados.

Nomnist	Media Train	Desviación Train	Media Test	Desviación Test
5 %	0.727111	0.052871	0.844667	0.093680
10 %	0.224000	0.065224	0.862667	0.084127
25 %	0.000000	0.000000	0.929333	0.073587
50 %	0.000000	0.000000	0.815333	0.131193

Figura 8: Errores cometidos por el conjunto de datos NOMNIST

Como vemos en las Tablas 8 y 9, los mejores resultados obtenidos son con una configuración el 50 % de los patrones usados como neuronas. Como vemos, conseguimos un error de 0.00 junto con un CCR de 100 % para el conjunto de train, en cambio, para el conjunto de test conseguimos un valor un poco más bajo, siendo del 87.93 % de los patrones bien clasificados, esto quiere decir que nuestro modelo no generaliza tan bien como otros.

Nomnist	Media Train	Desviación Train	Media Test	Desviación Test
5 %	88.31	0.89	88.60	1.06
10 %	96.22	0.76	87.47	0.72
25 %	100.00	0.00	86.87	1.19
50 %	100.00	0.00	87.93	0.93

Figura 9: CCR conseguido por el conjunto de datos NOMNIST

Ahora es el turno de cambiar el learning rate y el tipo de regularización.

Eta	Media Train	Media Test	Media Train	Media Test
1	1.095333	0.974000	80.02	82.20
0.1	0.777556	0.752000	86.42	88.47
0.01	0.752889	0.836667	87.84	88.60
0.001	0.727556	0.843333	88.29	88.60
0.0001	0.725111	0.844667	88.33	88.60
0.00001	0.727111	0.844667	88.31	88.60
0.000001	0.727111	0.844667	88.31	88.60
0.0000001	0.727111	0.844667	88.31	88.60
0.00000001	0.727111	0.844667	88.31	88.60
0.000000001	0.727111	0.844667	88.31	88.60
0.0000000001	0.727111	0.844667	88.31	88.60

Figura 10: Errores cometidos por el conjunto de datos NOMNIST con regularización tipo L1

Eta	Media Train	Media Test	Media Train	Media Test
1	1.220444	1.452000	77.73	77.20
0.1	0.902000	0.818667	83.47	86.33
0.01	0.792222	0.690667	86.02	89.20
0.001	0.734889	0.775333	87.69	89.07
0.0001	0.740444	0.847333	88.20	88.67
0.00001	0.723556	0.852667	88.31	88.60
0.000001	0.723778	0.852667	88.31	88.60
0.0000001	0.721778	0.852667	88.33	88.60
0.00000001	0.719778	0.852667	88.36	88.60
0.000000001	0.719778	0.852667	88.36	88.60
0.0000000001	0.721778	0.852667	88.33	88.60

Figura 11: CCR conseguido por el conjunto de datos NOMNIST con regularización tipo L2

Como vemos en las Tablas 10 y 11, los mejores resultados los obtenemos con un learning rate del 0.0001, siendo prácticamente igual si utilizamos el tipo de regularización L1 o L2, aunque obtenemos una ligera mejora utilizando el tipo de regularización L1.

6. Análisis

Como en el apartado de experimentos, esta sección la dividiremos en dos: Regresión y Clasificación. En este apartado iremos conjunto de datos por conjunto de datos analizando el aprendizaje de los patrones en nuestro modelo.

6.1. Regresión

6.1.1. Conjunto de datos SIN

El problema con este conjunto de datos utilizado para regresión es el poco número de patrones para el entrenamiento y para el test.

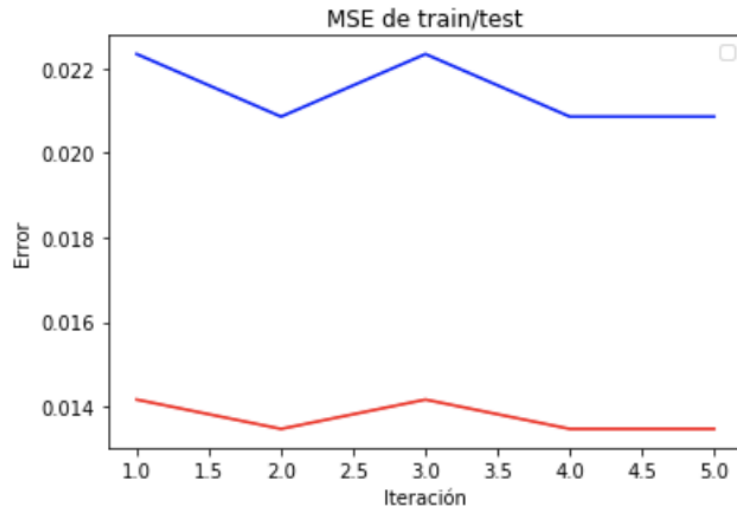


Figura 12: Gráfica error/iteración del conjunto de datos SIN

Como vemos en la Gráfica 12, el error producido por el conjunto de train(en rojo) es menor que el producido por el conjunto de test(en azul). Además, como es observable, el error empieza a estabilizarse desde la cuarta iteración, tanto para el conjunto de train como para el conjunto de test. Como nota, tenemos que decir que esta gráfica ha sido dibujada con la mejor configuración obtenida de la Tabla 1. Además, en esa Tabla(1), vemos cómo aumentando el porcentaje de patrones, el error de test comienza a elevarse, siendo esta acción no deseada para nuestro modelo.

6.1.2. Conjunto de datos QUAKE

Este conjunto de datos tiene un problema distinto al anterior conjunto, y es que, como vemos en la tabla 2, el modelo comienza a sobre-aprender conforme aumentamos el porcentaje de patrones. Ahora, con la mejor configuración conseguida, en la Gráfica 13 vemos cómo se comportan los errores tanto de train(rojo) como de test(azul).

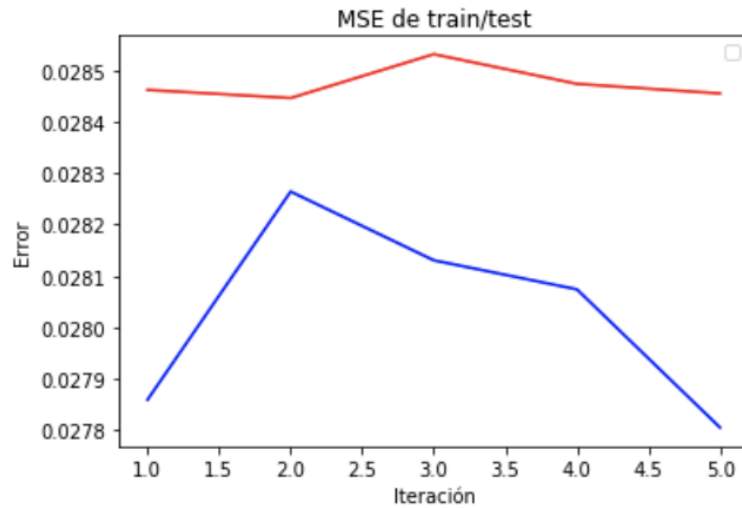


Figura 13: Gráfica error/iteración del conjunto de datos QUAKE

Como observamos, esta vez el error cometido por el conjunto de train es mayor que por el conjunto de test, y que conforme pasan las iteraciones, este error sigue disminuyendo, al contrario que con train el cual se empieza a estabilizar.

6.1.3. Conjunto de datos PARKINSONS

Como antes, para generar la gráfica de comportamiento del error, hemos escogido la mejor configuración de la Tabla 3, en la cual se ve que los mejores resultados son obtenidos utilizando un 50 % del número de patrones. En la Gráfica ?? se puede observar el comportamiento del error de train(rojo) y de test(azul) a lo largo de 5 iteraciones.

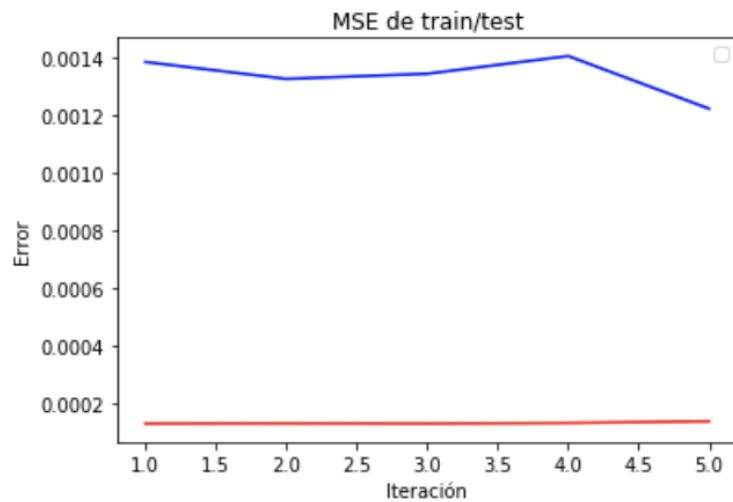


Figura 14: Gráfica error/iteración del conjunto de datos PARKINSONS

Como vemos en la Gráfica 14 los dos errores se mantienen prácticamente estables, es decir, no varían mucho de iteración en iteración, esto puede ser debido a la

cantidad de patrones que contiene el conjunto de datos. como vemos, en la última iteración, el error cometido por el conjunto de test empieza a bajar.

6.2. Clasificación

6.2.1. Conjunto de datos VOTE

Como antes, para ver la variación del error y del CCR con respecto a las iteraciones, escogeremos la mejor configuración de las Tablas 4 y 5. Las variaciones son observables en las gráficas 15 y 16.

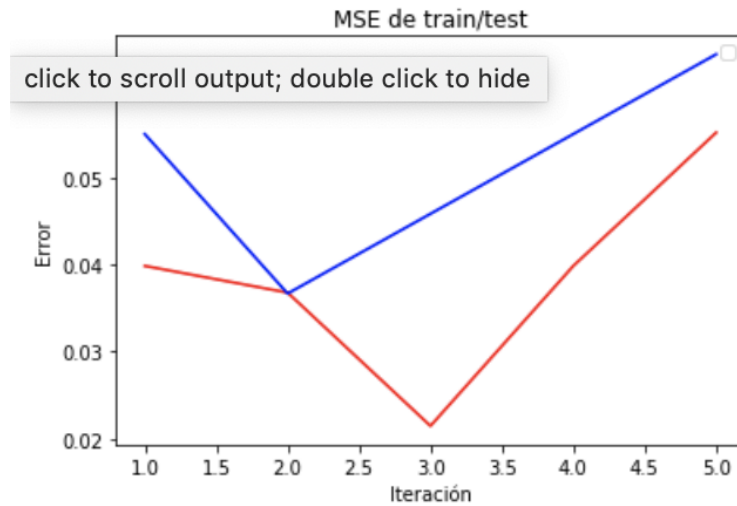


Figura 15: Gráfica error/iteración del conjunto de datos VOTE

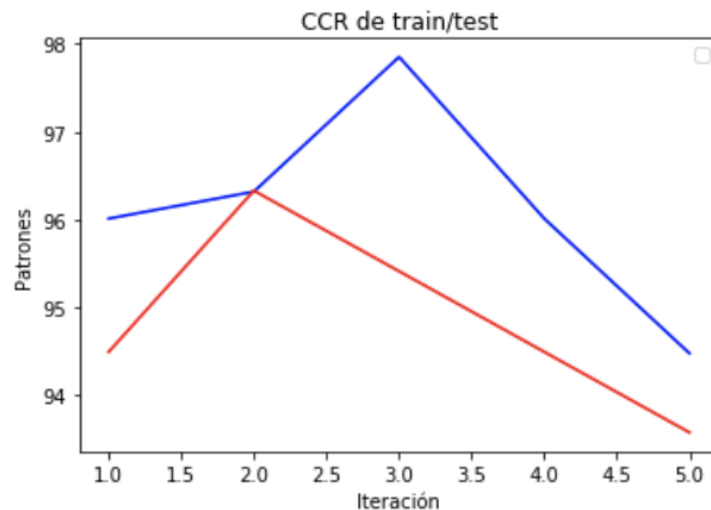


Figura 16: Gráfica CCR/iteración del conjunto de datos VOTE

Como vemos en las dos gráficas anteriores, existen muchas variaciones del error con respecto a las iteraciones, tanto para el conjunto de train(rojo), como para el

conjunto de test(azul). La principal novedad es que en el conjunto de test, aumenta el error al final, aunque por el contrario, el porcentaje de patrones bien clasificados también sube. En cambio, para el conjunto de train, el CCR empieza a disminuir desde la 4ª iteración hasta la 5ª.

Para comprender mejor estas gráficas, mostraremos a continuación la matriz de confusión conseguida por nuestro modelo. En la Matriz 17 puede verse el resultado.

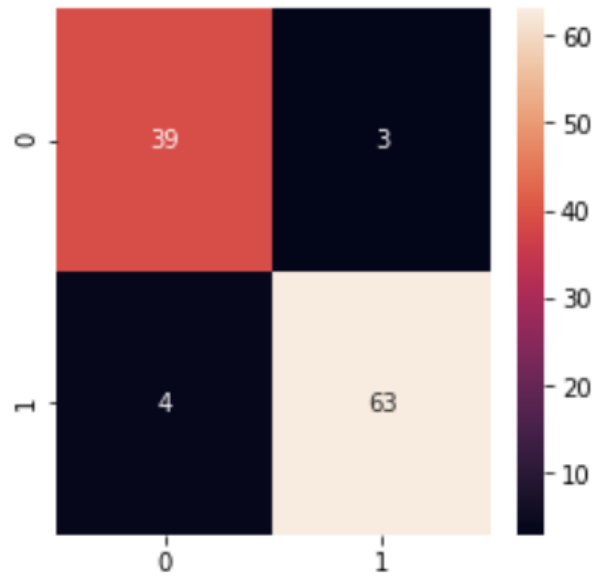


Figura 17: Matriz de confusión del conjunto de datos VOTE

Como vemos, existen patrones que no están bien clasificados, por ejemplo, hay 3 patrones que son clasificados como 1(republicano), siendo estos tres patrones de la clase 0(democrático). Además, hay 4 patrones que son, también, mal clasificados, pero esta vez, al contrario.

6.2.2. Conjunto de datos NOMNIST

Para este último conjunto de datos, haremos como con el conjunto de datos VOTE, primero se expondrá dos gráficas que nos explicarán mejor el comportamiento del error y del CCR con respecto a las iteraciones. Por último, se podrá ver la matriz de confusión resultante de este entrenamiento.

Como vemos en las Gráficas 18 y 19, el conjunto de train se mantiene prácticamente inalterable durante todas las iteraciones. En cambio, el conjunto de test, como vemos en la Gráfica 18, empieza a aumentar, aunque a partir de la segunda iteración, el error cometido comienza a bajar aunque lentamente. Con respecto al CCR pasa al contrario, primero empieza a disminuir lentamente aunque en la iteración 3 empieza a mejorar también lentamente hasta llegar a ser prácticamente del 90 % de patrones bien clasificados.

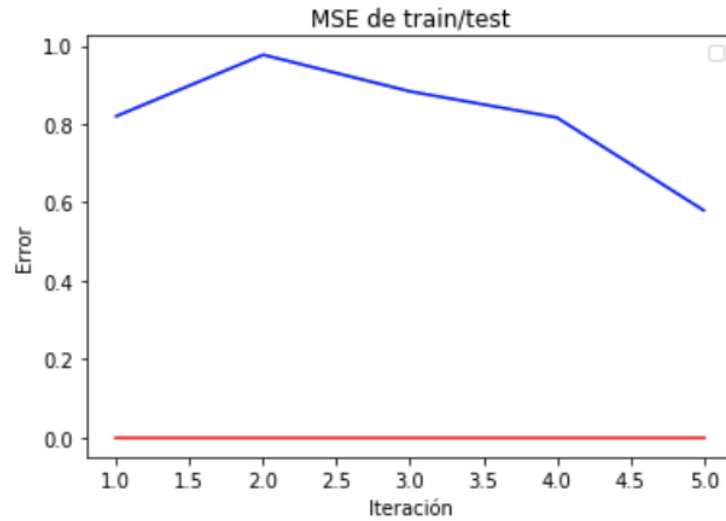


Figura 18: Gráfica error/iteración del conjunto de datos NOMNIST

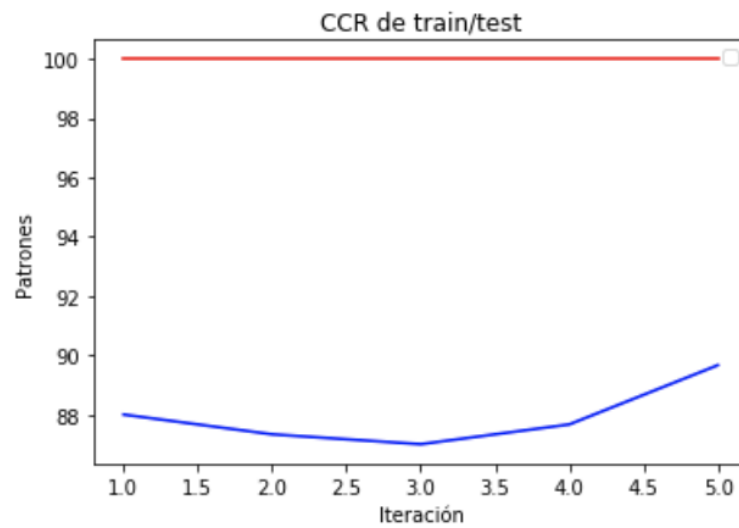


Figura 19: Gráfica CCR/iteración del conjunto de datos NOMNIST

Por último, en la Matriz 20 veremos con más precisión la cantidad de patrones bien o mal clasificados y en qué clases han sido clasificados.

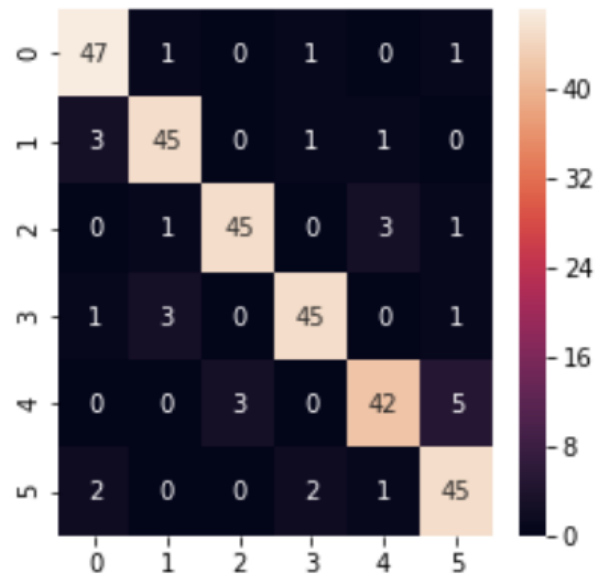


Figura 20: Matriz de confusión del conjunto de datos NOMNIST

Como se puede observar, en la clase que más falla es con la clase 4(E), en la cuál, clasifica como C 3 patrones que en realidad deberían de ser una E, además, clasifica 5 patrones en la clase 5(F) cuando debería de ser de la clase 4(E). Esto explica el casi 90 % de patrones bien clasificados.

7. Bibliografia

1. [Moodle](#)
2. [sklearn](#)
3. [Numpy](#)
4. [Pandas](#)
5. [Tutorial sklearn](#)
6. [Matplotlib](#)