

Práctica 1B: Reconocimiento de letras y dígitos mediante el Análisis de Componentes Principales (PCA)

Ingeniería del Conocimiento

Curso académico 2023-2024

1 Introducción

Durante la práctica anterior, se ha implementado el algoritmo de las componentes principales (PCA) y se ha aplicado en el reconocimiento de imágenes del conjunto de datos MNIST. En este caso, siendo una práctica introductoria, únicamente se ha realizado la resolución de un problema de clasificación binaria (dos clases, en este caso eran la clase “0” y la clase “5”).

En esta práctica vamos a realizar distintas pruebas para evaluar al algoritmo PCA cambiando los conjuntos de datos e intentando extraer algunas conclusiones a partir de estas. De esta manera, realizaremos:

1. Clasificación binaria entre clases “más difíciles” (comparar entre números parecidos).
2. Clasificación multiclase entre todas las clases (10) del conjunto de datos de reconocimiento de dígitos (MNIST ¹).
3. Clasificación multiclase entre todas las clases (26) del conjunto de datos de reconocimiento de letras (EMNIST ²).

2 Clasificaciones binarias

Dado que el problema a resolver hasta ahora ha sido la clasificación entre las clases “0” y “5”, se propone realizar las siguientes comparaciones:

- (1) Clasificación binaria entre la clase “1” y clase “7”.
- (2) Clasificación binaria entre la clase “2” y clase “7”.

¹LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. “*Gradient-based learning applied to document recognition.*” Proceedings of the IEEE, 86 (11): 2278-2324, 1998.

²Cohen, G., Afshar, S., Tapson, J., Van Schaik, A. “*EMNIST: an extension of MNIST to handwritten letters*”, 2017.

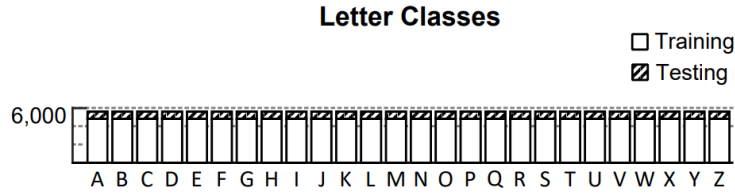


Figure 1: Distribución por clases del conjunto de datos EMNIST

- (3) Clasificación binaria entre la clase “4” y clase “9”.

En estos tres casos se utilizará la distribución utilizada en la práctica anterior (80 imágenes para train y 20 para test). Incluir estas tres pruebas en una función `main1.m`.

3 Clasificación multiclase MNIST

Por otro lado, se propone realizar la clasificación multi-clase, es decir, en lugar de utilizar imágenes de dos clases, utilizar las 10 clases que conforman el dataset. Para ello se deberá probar:

- (1) La distribución utilizada en la práctica anterior (80 imágenes para train y 20 para test).
- (2) Doblar la distribución (160 para train y 40 para test).

Incluir estas dos pruebas en una función `main2.m`.

4 Clasificación multiclase EMNIST

El conjunto de datos EMNIST, es una extensión del conjunto MNIST, donde se recogen también, además de imágenes de dígitos escritos a mano, letras mayúsculas y minúsculas (Figura 2). En este caso, utilizaremos la parte de EMNIST conformada sólo por letras mayúsculas. Originalmente, este conjunto de datos está compuesto por 145.600 ejemplos, distribuidos homogéneamente – como podemos observar en la Figura 1– en 26 clases. Al igual que en la práctica anterior utilizaremos una pequeña parte de este conjunto de datos.

Para cargar los datos se suministra un fichero de datos de MATLAB denominado `emnist-letters.mat`. A partir de ahí, con las siguientes líneas de código obtendremos un mini-conjunto de datos con 1000 imágenes de entrenamiento y 104 de test (4 para cada clase en test, 4×26).

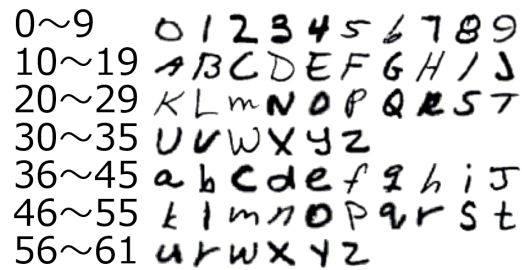


Figure 2: Clases del conjunto extendido inicial EMNIST

```
clear
load('emnist-letters.mat')
N_EX_TRAIN = 1000

XTr = reshape(dataset.train.images.', 28, 28, 1, []);
YTr = dataset.train.labels;
XTr = XTr(:,:,:,1:N_EX_TRAIN);
YTr = YTr(1:N_EX_TRAIN);

XTe = reshape(dataset.test.images.', 28, 28, 1, []);
YTe = dataset.test.labels;
XTe = XTe(:,:,:,1:100:20800);
YTe = YTe(1:100:20800);
```

El objetivo de esta parte es obtener una clasificación al igual que en los casos anteriores. Incluir esta prueba en una función `main3.m`.

5 Conclusiones

Una vez realizadas las pruebas, deberemos extraer algunas conclusiones a partir de algunas preguntas que planteamos:

- ¿Cómo clasifica el algoritmo realizado la distinción entre las clases “más conflictivas”? ¿Porqué consideras que clasifica bien a pesar de que sean números parecidos?
- ¿Qué ocurre cuándo tiene que clasificar 10 clases? ¿Cambia algo el algoritmo cuando tenemos el doble de datos de entrenamiento?
- ¿Qué ocurre cuando tratamos de clasificar 26 clases con un entrenamiento realizado con 1000 ejemplos? ¿Porqué crees que funciona de esa manera? ¿Has probado a realizar un entrenamiento con más ejemplos (2000, 5000, 10000, etc.? Si lo has probado, comenta qué es lo que ocurre.

6 Memoria

El trabajo deberá de sintetizarse en una memoria. Las partes de esa memoria deberán de ser las siguientes:

1. **Introducción.** Qué problema tenemos que resolver y el método que utilizaremos.
2. **Explicación del problema.** Explicación de los conjuntos de datos que utilizaremos. Explicación de cada una de las pruebas que vamos a realizar.
3. **Resultados.** Explicación de los resultados a nivel cuantitativo así como a nivel **cuantitativo cualitativo**, dando **respuesta a las preguntas de la Sección 5 (Conclusiones)**.
4. **Conclusión final.**

La extensión orientativa de la memoria es de unas 4 hojas (sin contar portada ni índice).

7 Entrega y fecha

En la evaluación se considerará principalmente la memoria. Los archivos a entregar serán los siguientes:

- [M1B_Apellido1Apellido2.pdf](#): Memoria de la Práctica 1B.
- [Codigo1B_Apellido1Apellido2.zip](#): Carpeta comprimida con los ficheros necesarios para la práctica ³ (NO incluir los datasets).

La práctica se entregará vía Tareas de MiAulario, con **fecha límite el 17 de noviembre** a las 23:55.

³Los ficheros `aprendeBase.m`, `creaPrototipos.m`, `clasificar.m`, `main1.m`, `main2.m`, `main3.m`.