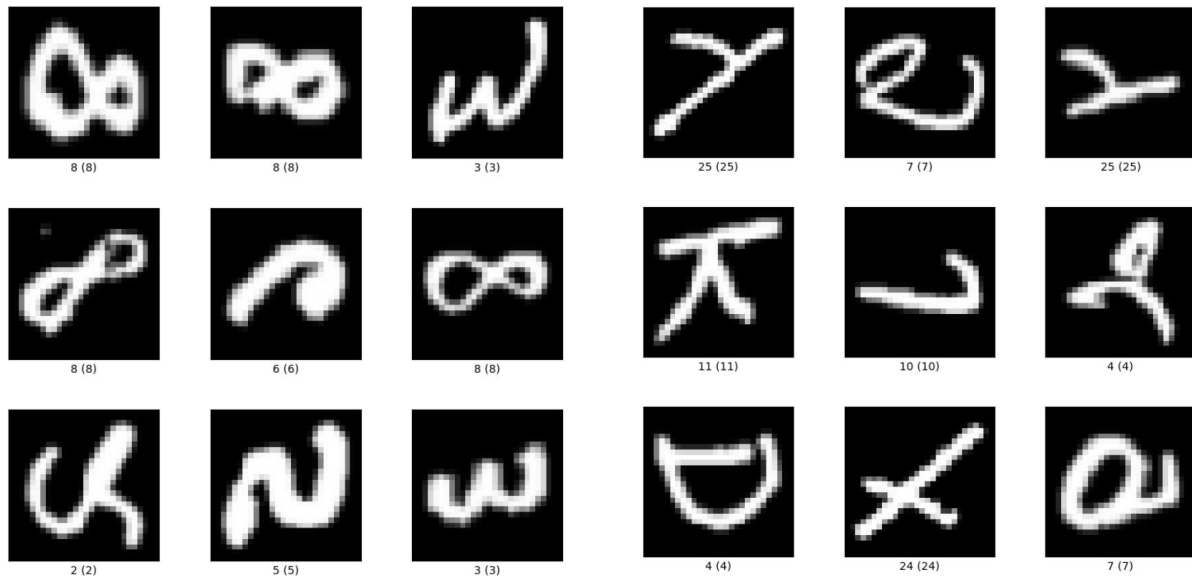


PROBLEMA 3: EMNIST - Clasificación de imágenes

1. Descripción del problema

El conjunto de datos [EMNIST](#) (Extended MNIST) [1] es un conjunto de caracteres (tanto letras como números) escritos a mano obtenidos de [NIST Special Database 19](#) en formato de escala de grises a 28x28 píxeles (formato original del MNIST¹ clásico).



Existen varias particiones del conjunto de datos. Las que contemplamos en esta práctica son:

- EMNIST ByClass: 814,255 muestras. 62 clases desbalanceadas. Se corresponde con el conjunto de datos completo, que incluye tanto dígitos como letras diferenciando mayúsculas y minúsculas.
- EMNIST ByMerge: 814,255 muestras. 47 clases desbalanceadas. Se corresponde con el mismo conjunto de datos anterior, pero colapsando algunas clases (para más información consultar [1]).
- EMNIST Balanced: 131,600 muestras. 47 clases balanceadas. Se corresponde con un subconjunto balanceado del conjunto anterior. El objetivo de este conjunto de datos es proporcionar una comparación justa, balanceada y lo suficientemente difícil para representar un reto.
- EMNIST Letters: 145,600 muestras. 26 clases balanceadas. Se corresponde con la partición del conjunto de datos original perteneciente a letras balanceada más grande posible. En esta versión, no se diferencia entre mayúsculas y minúsculas. Esto representa un reto pues para su correcta clasificación, se precisa que el clasificador aprenda a identificar con una misma etiqueta dos símbolos (mayúscula y minúscula).

¹ <http://yann.lecun.com/exdb/mnist/>

	Número de muestras	Número de clases	¿Clases balanceadas?
EMNIST ByClass	814,255	62	No
EMNIST ByMerge	814,255	47	No
EMNIST Balanced	131,600	47	Sí
EMNIST Letters	145,600	26	Sí

Claramente, la dificultad de los conjuntos de datos es decreciente. Principalmente por la disminución del número de clases, pero también por la presencia de clases desbalanceadas en los datos. Podéis elegir qué conjunto de datos utilizar o incluso experimentar con más de uno, pero tened en cuenta que a más dificultad, más posibilidad de proponer soluciones interesantes. Yo os aconsejo comenzar con el de EMNIST Letters y, si os animáis ir probando en orden ascendente, pero empezar por el más sencillo.

2. Descarga de datos

Con respecto a la descarga de datos, existen las siguientes posibilidades:

1. Descargar los datos de la [página oficial](#) y leerlos directamente de archivos locales en vuestro código. Se encuentran en formato Matlab (accesible desde Matlab y o Python usando la función `scipy.io.loadmat`) y formato binario.
2. Descargar los datos utilizando la librería `tensorflow_datasets`². Yo recomiendo encarecidamente esta segunda opción, dado que simplifica mucho el proceso de lectura de datos.

3. Modelos baseline

Dado que es un conjunto de datos antiguo, existen múltiples aproximaciones. Yo os propongo algunas a continuación, pero sois libres de investigar y proponer lo que se os ocurra:

1. En primer lugar, en [Kaggle](#) podéis encontrar múltiples soluciones explicadas en Notebooks. Os aconsejo que estudiéis las que os parezcan más interesantes y que tratéis de replicarlas y proponer mejoras sobre dichas soluciones, o quizás combinar algunas soluciones, etc.
2. Aproximaciones basadas en Deep Learning. La mayoría de los problemas de clasificación de imágenes obtienen los mejores resultados con modelos de *Deep Learning* basados en redes convolucionales³. Os animo a probar esta alternativa y a

² <https://www.tensorflow.org/datasets/catalog/emnist>

³ <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

experimentar con diferentes arquitecturas basadas en redes convolucionales. Este [blog](#) os puede ayudar

3. Uso de redes preentrenadas. En los últimos años se ha popularizado el uso de redes neuronales pre-entrenadas junto con técnicas de transferencia de conocimiento dado que proporcionan los mejores resultados en muchos problemas. Podéis probar a utilizar una red pre-entrenada de clasificación de imágenes y adaptarlo al dominio del EMNIST. Os recomiendo el siguiente [blog](#). Esta aproximación puede ser especialmente interesante para aquellos conjuntos que tienen un elevado número de clases.

Finalmente, os recomiendo el artículo [2] que recoge el estado del arte en el problema de reconocimiento de caracteres escritos, en concreto para MNIST y EMNIST. Podéis utilizar este paper para ver qué aproximaciones se están proponiendo en la literatura y hacer la sección de trabajo futuro o probar alguna de ellas.

Referencias

[1] Cohen, G., et al. "EMNIST: an extension of MNIST to handwritten letters. arXiv e-prints." *arXiv preprint arXiv:1702.05373* (2017). <https://arxiv.org/abs/1702.05373v1>

[2] Baldominos, A. et al. "A Survey of Handwritten Character Recognition with MNIST and EMNIST." *Applied Sciences* 9 (2019): 3169. <https://www.mdpi.com/2076-3417/9/15/3169>