



Práctica 4: Extreme Learning Machine

Francisco Fernández-Navarro
Modelos computacionales

Enunciado de la práctica

Descarga el fichero ELM.zip del campus virtual. Para descomprimirlo, ejecuta en la línea de comandos de Matlab lo siguiente:

```
unzip ELM.zip
```

Aspectos a tener en cuenta:

1. El fichero comprimido contiene dos ficheros: la base de datos `handwriting.mat` y el fichero de código `P4_ELMStudents.m`.
2. La base de datos asociada a la práctica se conoce como MNIST de Dígitos reducida. A diferencia del MNIST original, que contiene 60,000 imágenes de entrenamiento y 10,000 imágenes de prueba de dígitos escritos a mano, esta versión reducida contiene 500 imágenes para cada dígito del 0 al 9, lo que suma un total de 5,000 imágenes. Estas imágenes son una colección de dígitos escritos a mano.
3. Por la estructura de la base de datos, los 500 primeros números son el 0, los 500 siguientes el 1, y así sucesivamente. Al no incluirse la etiqueta de clase en la base de datos original, lo primero que hace el script es generar esa etiqueta de clase.
4. Después de generar las etiquetas de clase para nuestros datos, el siguiente paso en nuestro proceso es preparar los datos para entrenar y evaluar nuestros modelos de aprendizaje automático. En nuestro caso, en esta fase nos limitaremos a escalar los datos para dejarlos todos con el mismo rango.
5. Una vez que hemos escalado nuestros datos, creamos tres conjuntos de datos diferentes: el conjunto de entrenamiento (llamado 'TrainVal'), el conjunto de validación (llamado 'Val') y el conjunto de prueba (llamado 'Test'). Además de estos tres conjuntos, tenemos un cuarto conjunto llamado 'Train' que se crea incorporando parte de los datos del conjunto de validación al conjunto de entrenamiento (TrainVal).

El objetivo de esta práctica es implementar el modelo ELM (Extreme Learning Machine) neuronal en dos fases diferentes del código. La primera fase es la que implementamos el modelo ELM ocurre dentro de lo que llamamos 'validación anidada'. La validación anidada es una técnica que nos permite encontrar los mejores hiperparámetros para nuestro modelo. En este caso, estamos interesados en determinar los valores óptimos para dos hiperparámetros: C y L. Cada uno de estos hiperparámetros afecta cómo funciona el modelo ELM y es esencial encontrar los valores que maximizan su rendimiento.

En la segunda fase, utilizamos el modelo ELM con el conjunto completo de entrenamiento. En este punto, ya hemos encontrado los valores ideales para C y L gracias a la validación anidada. Ahora, el objetivo es entrenar el modelo

con todo el conjunto de datos de entrenamiento. Una vez entrenado, podemos calcular dos métricas importantes: el Error Cuadrático Medio (MSE, por sus siglas en inglés) y la Tasa de Clasificación Correcta (CCR, por sus siglas en inglés) en el conjunto de prueba.

Parte Optativa: En relación al artículo 'A multi-class classification model with parametrized target outputs for randomized-based feedforward neural networks', publicado en la revista 'Applied Soft Computing' por el profesor Francisco Fernández-Navarro, les ofrecemos una emocionante oportunidad. Si logran implementar con éxito uno de los modelos que se describen en el artículo, podrán aumentar su calificación en 1 punto en el próximo parcial. Es importante destacar que esta tarea es completamente opcional y no es un requisito para aprobar la parte práctica de la asignatura, sino una oportunidad para aquellos que deseen desafiarse y mejorar su calificación.