**Práctica 4. Clasificación multiclase y redes neuronales**

**Parte A. Clasificación multiclase**

**Descripción del problema**

Se ha proporcionado un dataset que contiene 5000 ejemplos de entrenamiento de números escritos a mano. El fichero en formato *.mat* significa que los datos se han guardado de formato binario/octal. Estas matrices se pueden leer utilizando la librería ***scipy.io.*** concretamente la función ***loadmat***.

A computer code on a black background

Description automatically generated

**Visualización del conjunto de números**

Existen 5000 ejemplos de entrenamiento en el fichero proporcionado para la práctica, donde cada ejemplo es una imagen de 20 X 20 pixeles. Cada pixel se encuentra representado por un número *float* indicando la intensidad de la escala en esa localización.

Cada uno de estos ejemplos de entrenamiento se convierte en una sola fila en nuestros datos matriz X. Esto nos da una matriz ***X*** de 5000 por 400 donde cada fila es un ejemplo de entrenamiento para una imagen de dígitos escritos a mano.

La segunda parte del conjunto de entrenamiento es un vector ***y*** de 5000 dimensiones.

que contiene etiquetas para el conjunto de entrenamiento etiquetadas como "0" a "9" en su orden natural.

Se puede visualizar un conjunto de números de la siguiente forma:

A computer code with text

Description automatically generated

El resultado es el siguiente

A close-up of numbers

Description automatically generated

**Regresión logística vectorizada**

Se usarán múltiples modelos logísticos ***one-vs-all*** para construir el clasificador multiclase. Como hay 10 clases diferentes (una clase por cada número), se entrenará 10 clasificadores.

Se implementará una versión vectorizada de regresión logística que no emplee ningún bucle. Se usará el código del ejercicio anterior como punto de partida para implementar versiones vectorizadas del cálculo del coste regularizado (***compute\_cost\_reg***) y del cálculo del gradiente regularizado (***compute\_gradient\_reg***)

**Función del cálculo del sigmoide:**

**A math equation with numbers and symbols

Description automatically generated**

Al implementarlo en el código será de la siguiente manera:

A computer screen shot of text

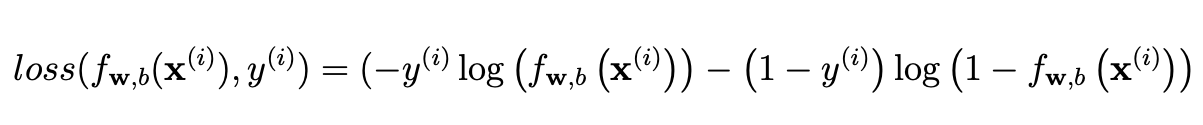
Description automatically generated

**Función del cálculo del coste**

**A mathematical equation with numbers and symbols

Description automatically generated**

donde *loss* es:



A black and white math equation

Description automatically generated

y g es la función sigmoide

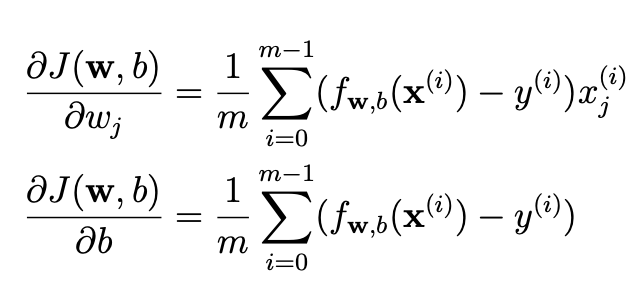
Por tanto, para implementar la función del cálculo del coste

A computer code with many letters

Description automatically generated with medium confidence

**Función para el cálculo del gradiente**

El gradiente para la regresión logística con múltiples variables viene definido por:



Para la implementación será de la forma:

A computer screen shot of a code

Description automatically generated

**Cálculo de los apartados en un modelo de regresión logística regularizada**

Una vez calculado los costes y el gradiente aplicamos los métodos para desarrollar las funciones que calculen los costes de un modelo de regresión logística regularizado y el gradiente.

**Coste del modelo de regresión logística regularizado**

Para el coste del modelo de regresión logística regularizado la fórmula es la siguiente:

A black numbers on a white background

Description automatically generated

Por lo que el nuevo método es:

A computer code with red and blue text

Description automatically generated

**Gradiente del modelo de regresión logística regularizado**

Con respecto al gradiente del modelo de regresión logística regularizado la fórmula viene definida por:

A math equations with numbers and symbols

Description automatically generated with medium confidence

Por lo que se añade al método anterior lo siguiente:

A computer code with text

Description automatically generated with medium confidence

Se han pasado todos los tests existentes de la anterior práctica.

**Parte B. Redes neuronales**