Práctica 4:

Entrenamiento de Redes neuronales

Grupo 13:

David Ortiz Fernández.

Andrés Ortiz Loaiza.

En esta práctica retomamos el guion de la práctica anterior. Comenzamos cargando el fichero de datos, que consiste nuevamente en 5000 imágenes de 20x20 pixeles de tamaño, cada uno de los cuales almacena la intensidad de dicho pixel en la escala de grises.

Se ha desarrollado una función que da pesos iniciales aleatorios cercanos a cero a las thetha iniciales para un correcto funcionamiento.

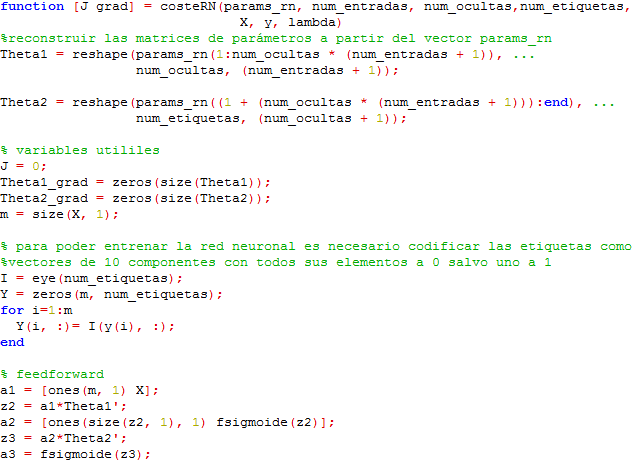
Posteriormente se ha desarrollado el código de una función destinada a calcular el coste y el gradiente realizando la propagación hacia delante y posteriormente la retropropagación hacia atrás para minimiza esta función de coste (costeRN).

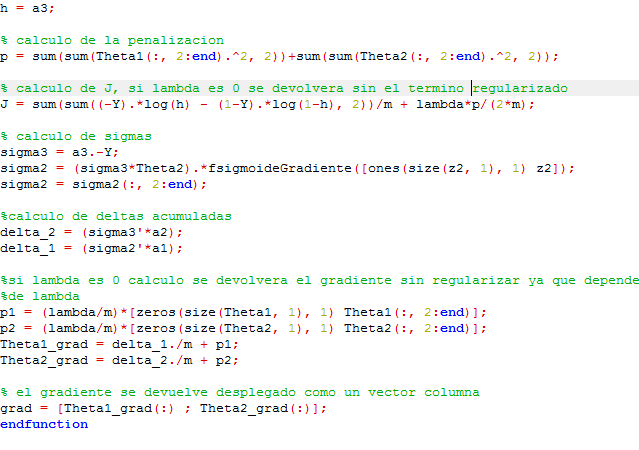
Después de esto, mediante la invocación de la función “checkNNGradients”, se ha comparado que el cálculo de gradientes mediante los dos métodos es similar, lo cual nos indica el buen funcionamiento de este.

A continuación, con estos valores se han obtenido los parámetros de la red neuronal óptimos, es decir sus thetas óptimos y de esta manera se puede calcular la probabilidad de pertenencia.

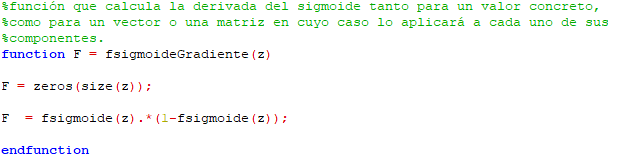
Se adjuntan dichas funciones entre cuyos comentarios se puede apreciar paso a paso:

1. **Función de coste y cálculo de gradiente**

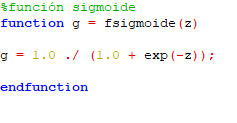


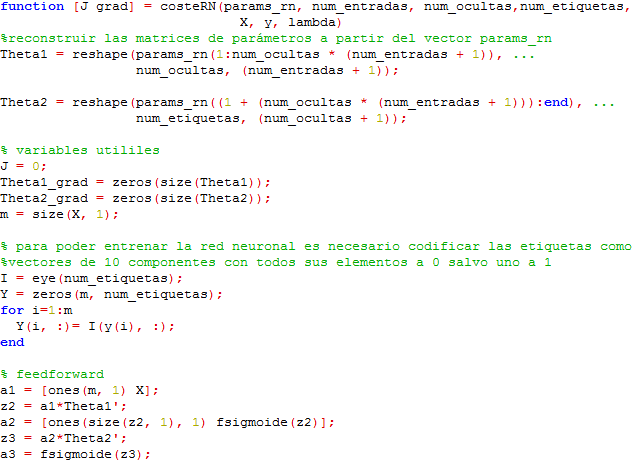


**FsigmoideGradiente:**

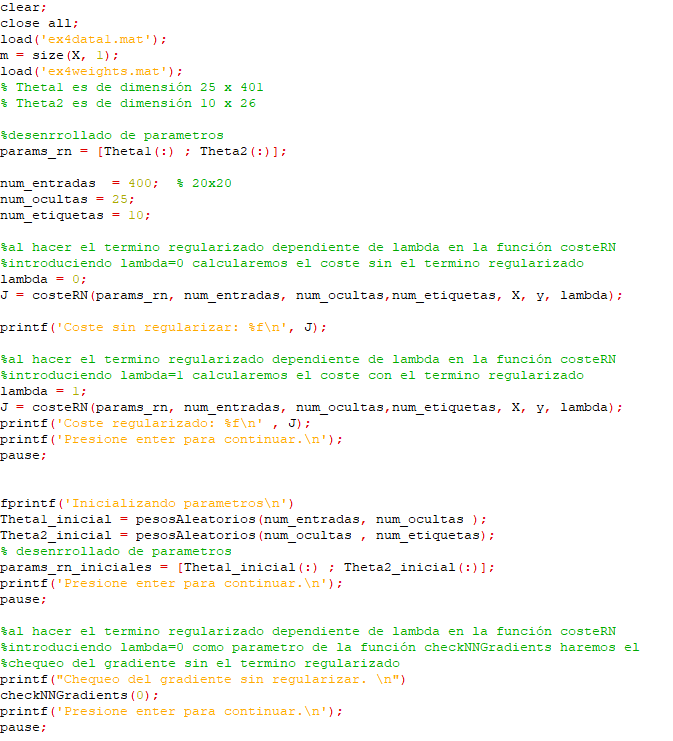
****

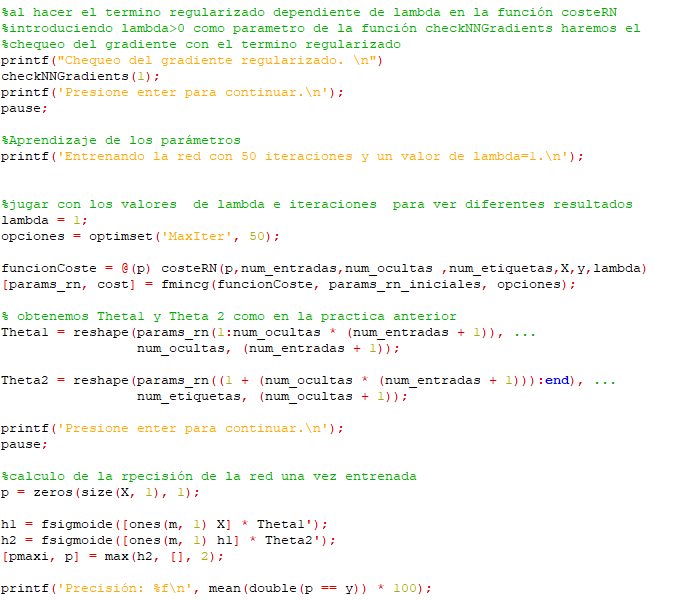
**Función fsigmoide :**

****

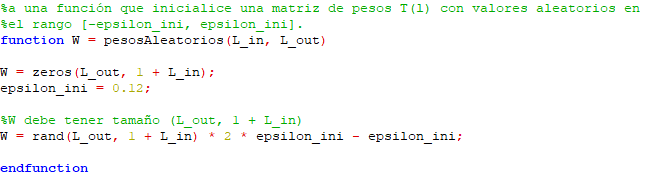
****

**Flujo principal de la práctica:**

****

****

**Función PesosAleatorios :**

****

**Resultados :**

Coste sin regularizar (con lambda 0): 0.287629

Coste regularizado (con lambda 1): 0.383770

Chequeo del gradiente sin regularizar: 2.27415e-011

Chequeo del gradiente regularizado:2.29153e-011

Precisión: 95.36000

Tras el entrenamiento de la red neuronal se ha obtenido así una precisión muy alta, en torno al 95%, lo cual indica la alta probabilidad de predicción correcta de la red neuronal desarrollada. También se ha podido apreciar, gracias a la función “checkNNGradients”, el correcto funcionamiento de la propagación hacia delante y retropropagación, al descender en cada iteración el gradiente de manera correcta, esto se consigue comparando los valores obtenidos al aproximar los cálculos obtenidos al aplicar las derivadas parciales con el método de aproximación de la pendiente.