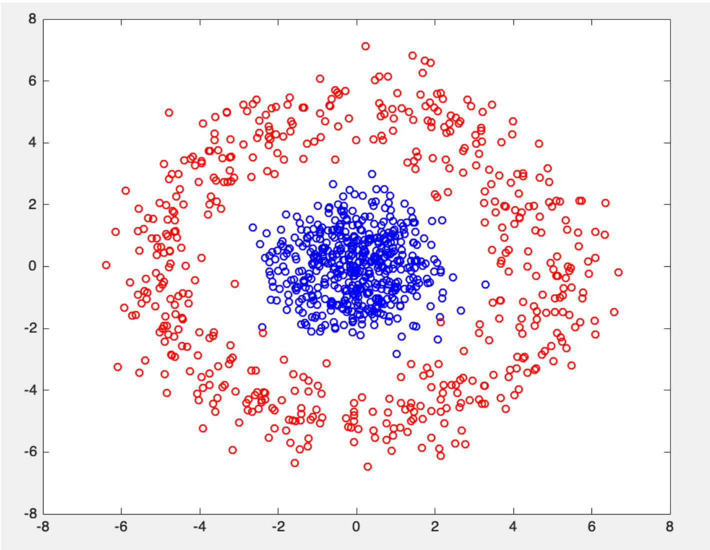
Inteligencia Artificial
Grado en Ingeniería Informática en Sistemas de Información - Curso 2019/20
HOMEWORK #3: Redes Neuronales

Bienvenidos a la tercera tarea de la asignatura Inteligencia Artificial. Las redes neuronales son métodos de machine learning que son capaces de captar los patrones ocultos en un conjunto de datos con una gran precisión. Su mayor desventaja es que tienden al sobreajuste (overfiting en inglés) en su entrenamiento. En esta tarea aprenderás de una forma práctica el concepto de overfitting y cómo la regularización evita el sobreajuste. Además, aprenderás cómo afecta el número de neuronas de la capa oculta a la tasa de acierto de la clasificación y cómo este parámetro está relacionado con el overfiting. Por último, aprenderás como hay que seleccionar el valor del parámetro de regularización para que la regularización sea efectiva y verdaeramente elimine el overfitting.

Problema

En este problema implementaremos una red neuronal para el conjunto de datos que se muestra en la figura.



Como se puede observar los datos no son separables linealmente y se necesita una frontera de decisión mucho más compleja que una simple recta. Las redes neuronales son capaces de aprender fronteras de decisión altamente no lineales a diferencia de la regresión logística.

1. Cargue y visualice el conjunto de datos usando la función `plotData`.
2. Implemente una red neuronal con una única capa oculta y haga una predicción de todo el conjunto de datos, imprimiendo por pantalla la tasa de acierto y la gráfica con la frontera de decisión usando para ello la función `plotDecisionBoundary`.

   Con el objeto de verificar el código tenga en cuenta las siguientes soluciones parciales:

   Para una red neuronal con 2 neuronas en la capa oculta, y los siguientes pesos iniciales:

   initial_Theta1=$\begin{bmatrix} -0.0893 & -0.0789 & 0.0147 \\ 0.1198 & -0.1122 & 0.0916 \end{bmatrix}$ y initial_Theta2=$[0.0406\ -0.0743 - 0.0315]$

   - La función de coste J vale 0.6932
   - Las derivadas tienen los siguientes valores:

$$\frac{\partial J}{\partial \theta_1} = \begin{bmatrix} 0.000140 & -0.000026 & 0.000175 \\ 0.000134 & -0.000209 & 0.000198 \end{bmatrix}$$

$$\frac{\partial J}{\partial \theta_2} = [-0.002888 \quad -0.001551 \quad -0.002538]$$

- Una vez ejecutado el descenso del gradiente con 1000 iteraciones, los pesos óptimos son:

Theta1 = $\begin{bmatrix} -4.7805 & 0.7160 & -1.6307 \\ 5.3456 & 1.1665 & -1.5917 \end{bmatrix}$ y Theta2 = $[-7.7130 - 8.4793\ 9.5412\ ]$

3. Observe los diferentes comportamientos del modelo para diferentes números de neuronas en la capa oculta. Experimente con 1, 2, 3, 4, 5, 20 y 50 e imprima en cada caso tanto la tasa de acierto cuando se predice todo el conjunto de datos como la frontera de decisión.

4. Implemente una red neuronal con una única capa oculta y 10 neuronas en la capa oculta que no tenga overfitting. Haga una predicción de todo el conjunto de datos, imprimiendo por pantalla la tasa de acierto y la gráfica con la frontera de decisión usando para ello la función `plotDecisionBoundary`.

   Con el objeto de verificar el código tenga en cuenta las siguientes soluciones parciales:

   Para una red neuronal con 3 neuronas en la capa oculta, un valor de lambda 3 y los siguientes pesos iniciales:

   initial_Theta1 = $\begin{bmatrix} -0.069081 & -0.077998 & 0.094653 \\ -0.096356 & -0.080743 & 0.003974 \\ 0.077658 & 0.039837 & 0.048649 \end{bmatrix}$ y

   initial_Theta2 = $[-0.083138 \quad 0.10883 \quad 0.0098122 \quad 0.043136\ ]$

   - La función de coste J vale 0.693192
   - Las derivadas tienen los siguientes valores:

$$\frac{\partial \mathrm{J}}{\partial \theta_1} = \begin{bmatrix} -0.00031339 & -0.00032803 & 0.00038878 \\ -1.3783\mathrm{e}-05 & -0.00025836 & 2.8666\mathrm{e}-06 \\ -4.1149\mathrm{e}-05 & 0.00012527 & 3.6738\mathrm{e}-05 \end{bmatrix}$$

$$\frac{\partial \mathrm{J}}{\partial \theta_2} = [-0.00088329 \quad -0.00011447 \quad -0.0003759 \quad -0.00039345]$$

   - Una vez ejecutado el descenso del gradiente con 1000 iteraciones, los pesos óptimos son:

   Theta1 = $\begin{bmatrix} 5.6479 & 0.50794 & 2.2628 \\ -5.7102 & -1.7211 & 1.5416 \\ 5.3684 & -2.1967 & -0.76961 \end{bmatrix}$ y Theta2 = $[-6.8321 \quad 4.7718 \quad -4.7226 \quad 4.6393\ ]$

   Nota: La función *fminunc* os podría dar algún warning, lo puedes ignorar o usar la función *fmincg* (equivalente a fminunc) que tenéis disponible en el material.