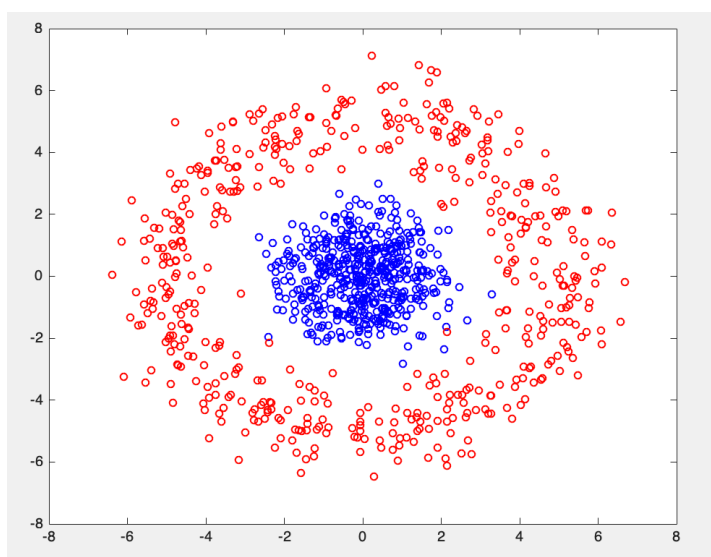




Trabajo Redes Neuronales

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 (overfitting 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 overfitting. Por último, aprenderás como hay que seleccionar el valor del parámetro de regularización para que la regularización sea efectiva y verdaderamente elimine el overfitting.

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:

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

- 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:

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

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:

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

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

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

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

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

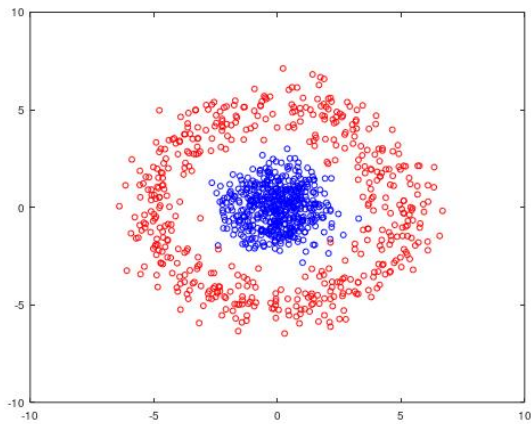
- Una vez ejecutado el descenso del gradiente con 1000 iteraciones, los pesos óptimos son:

$$\text{Theta1} = \begin{bmatrix} -5.4435 & -1.7436 & 1.3479 \\ -5.3530 & 2.1157 & 0.9082 \\ 5.4395 & 0.3274 & 2.2212 \end{bmatrix} \text{ y } \text{Theta2} = [-2.2104 \quad -4.7331 \quad -4.8126 \quad 4.8225]$$

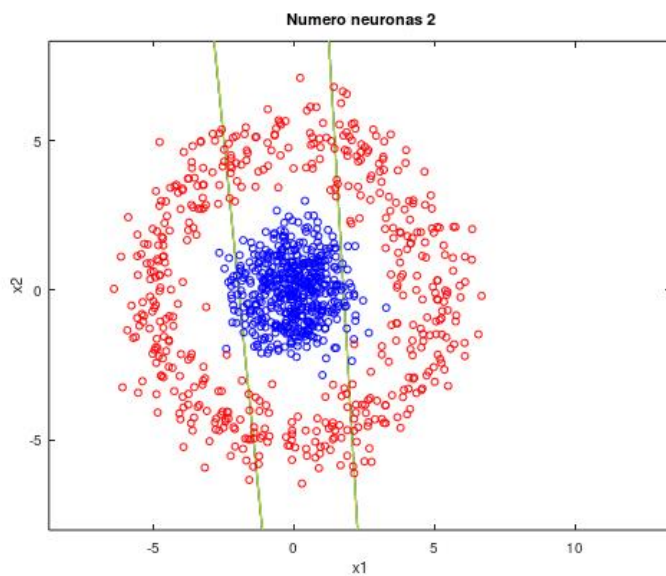
Nota: La función `fminunc` os podría dar algún warning, lo puedes ignorar o usar la función `fmincg` (equivalente a `fminunc`) que tenéis disponible en el material. Tenga en cuenta que los resultados parciales que se muestran han sido obtenidos con la función `fminunc`.



1. Cargar y visualizar datos.



2. Red neuronal con una capa oculta de 2 neuronas con valores thetas preinicializados como dice el enunciado.



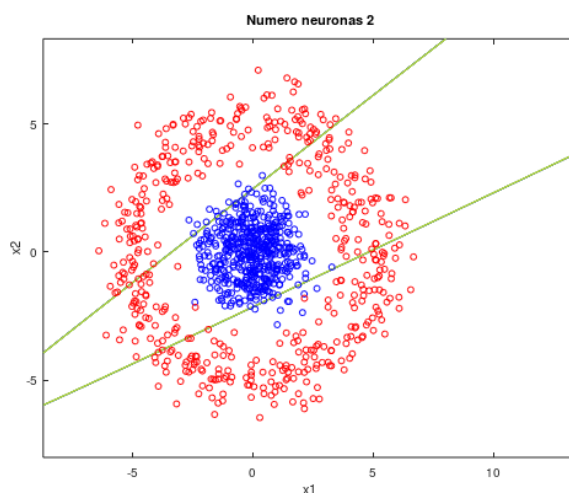
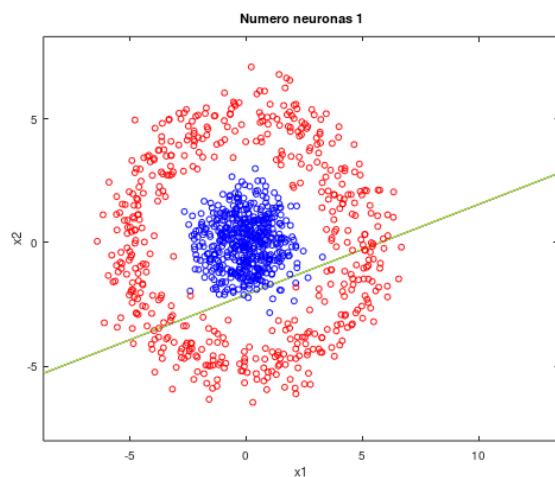
Tasa de acierto: 84.4%

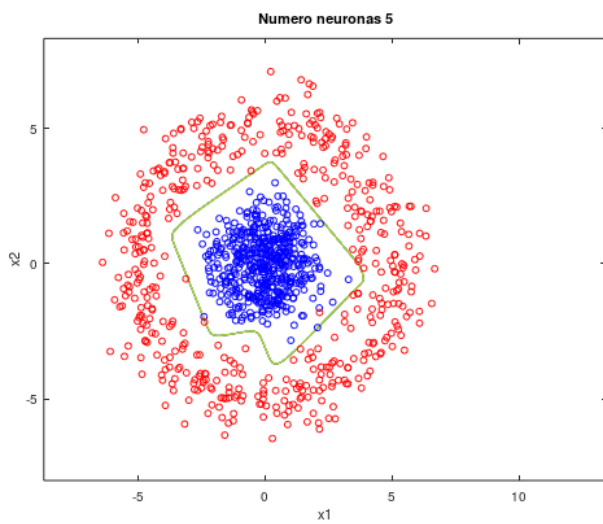
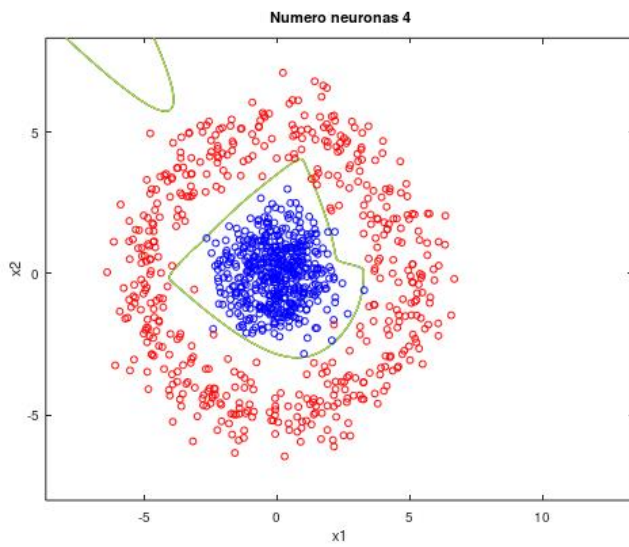
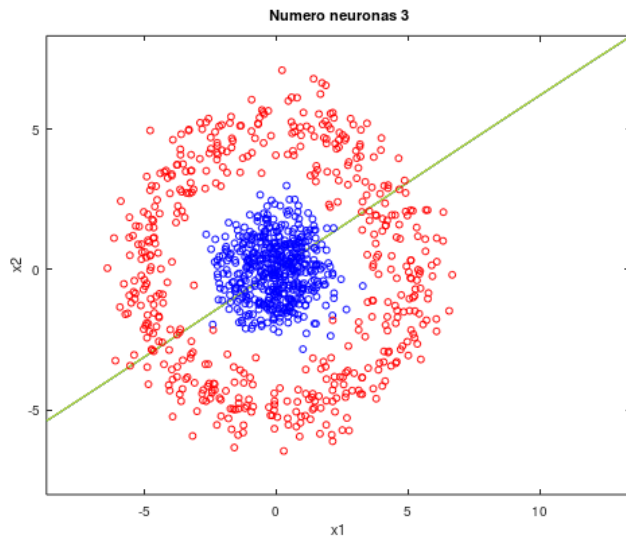


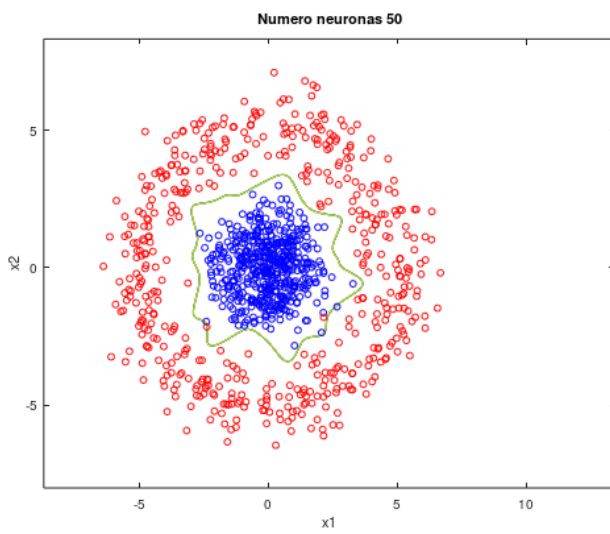
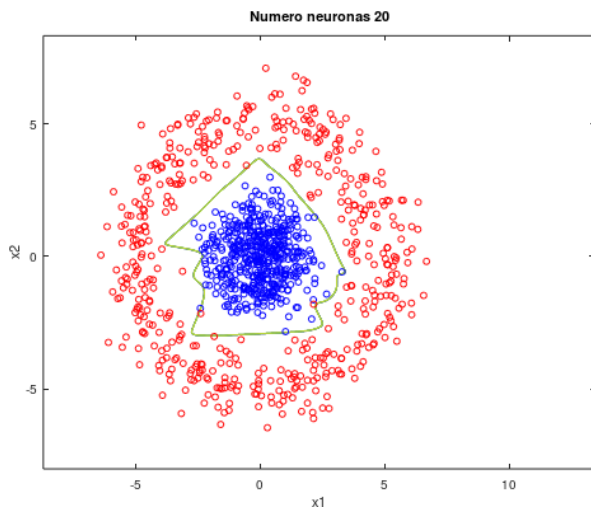
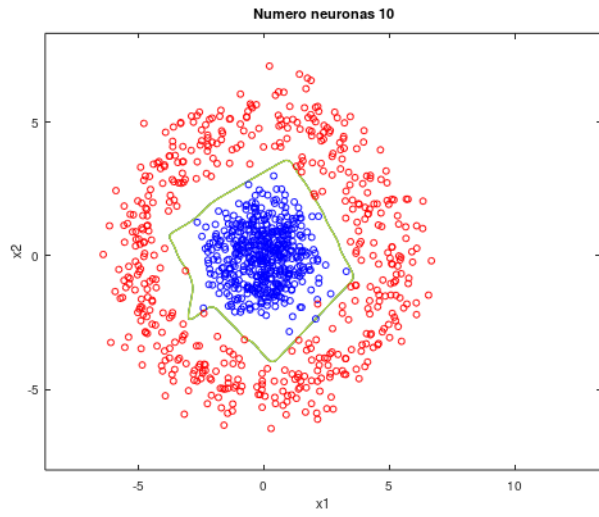
3. Con la red neuronal implementada en el apartado 3, rellene los resultados obtenidos de predecir TODO el conjunto de datos en la siguiente tabla.

Número de neuronas ocultas	Tasa de acierto
1	68.2%
2	85%
3	50.5%
4	99.5%
5	99.9%
10	99.9%
20	99.7%
50	99.8%

Mostrar gráficas con datos y frontera de decisión para número de neuronas en la capa oculta de 1, 2, 3, 4, 5, 10, 20 y 50. **Nota:** Con el objeto de distinguir las gráficas, ponga como título en cada gráfica el número de neuronas de la capa oculta.









Después de analizar las gráficas anteriores, responda a las siguientes cuestiones:

¿Cuál es el mejor valor del parámetro “número de neuronas en la capa oculta”?

¿Por qué?

Respuesta:

2, porque tiene la mejor tasa de acierto, debido a que no es un número de neuronas excesivo (produce overfitting) ni demasiado pequeño (produce underfitting).

¿Por qué la red neuronal con 1 y 2 neuronas en la capa oculta no funciona bien?

Respuesta:

Porque un número muy pequeño de neuronas puede producir falta de conocimiento en la red neuronal (underfitting).

¿Qué comportamiento observas en la red neuronal cuando se aumenta el número de neuronas en la capa oculta?

Respuesta:

A medida que aumentamos de forma excesiva el número de neuronas, estamos sobreentrenando la red neuronal (sobreajuste – overfitting) y podemos observar una tasa de acierto casi del 100% porque se ajusta demasiado bien al modelo y no predice.

¿Qué harías para poder usar modelos de redes neuronales con un número elevado de neuronas en la capa oculta?

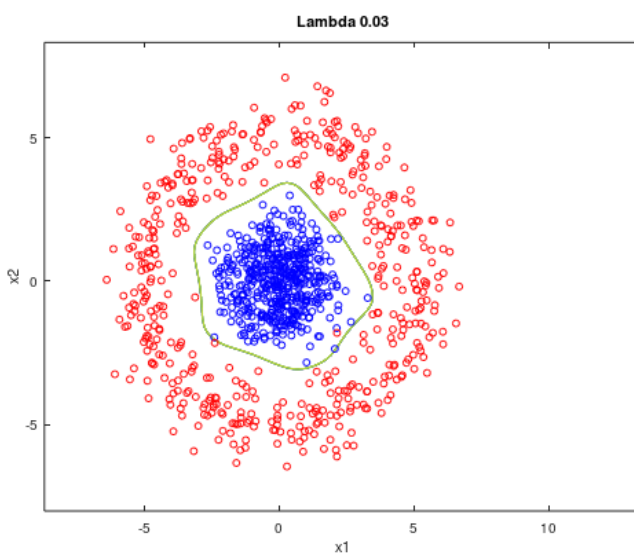
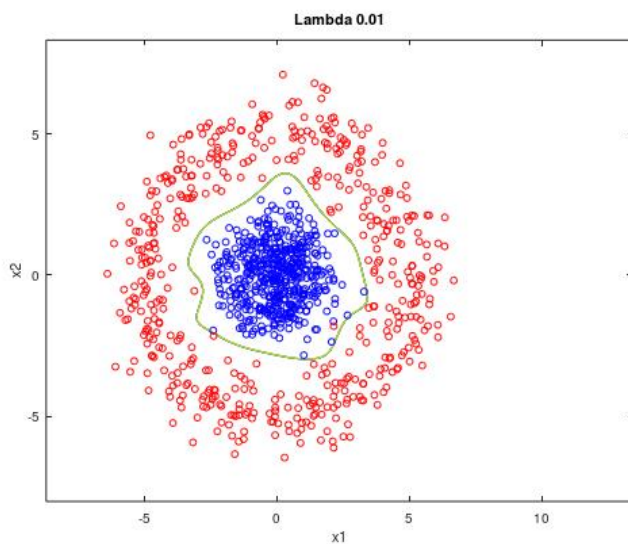
Respuesta:

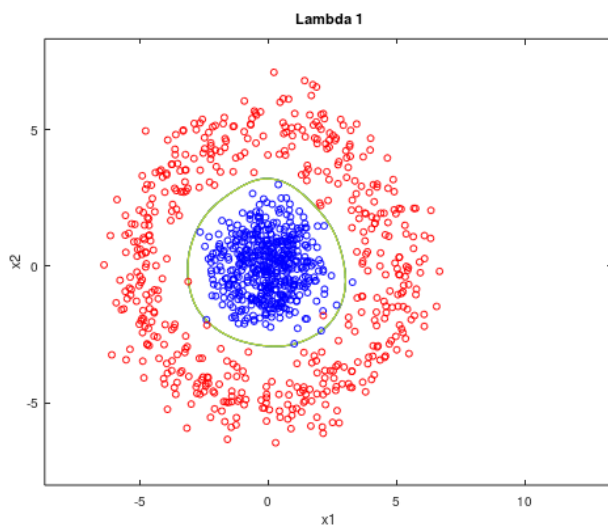
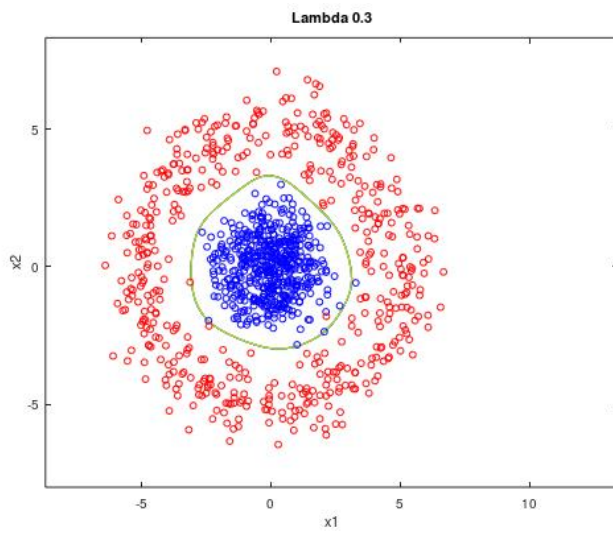
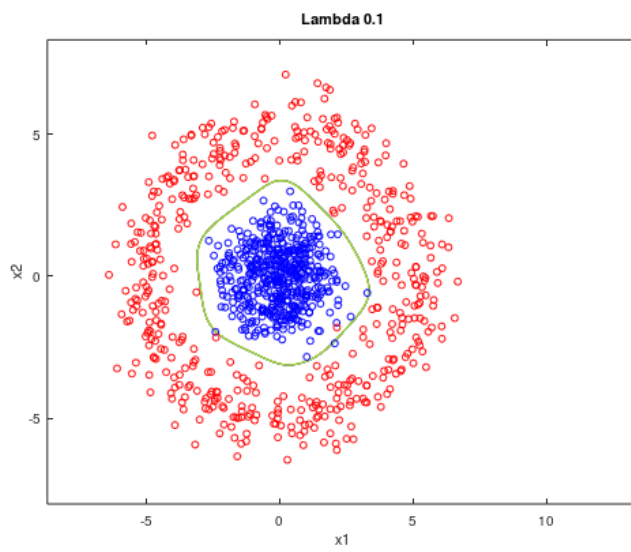
Aplicar regularización para evitar el sobreajuste al entrenar el modelo.

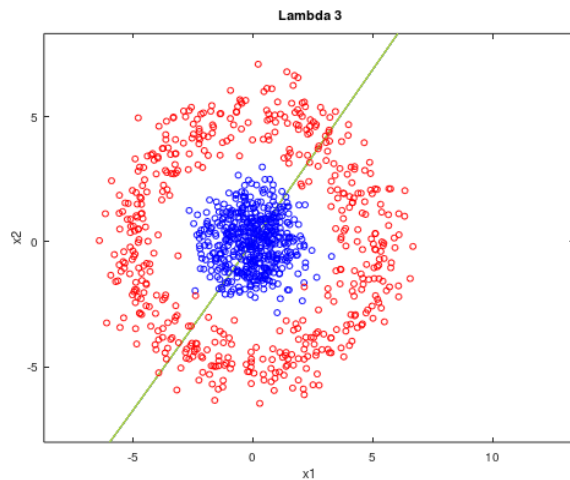


4. Con la red neuronal implementada en el apartado 4, mostrar gráficas con datos y frontera de decisión para parámetro de regularización λ 0.01, 0.03, 0.1, 0.3, 1 y 3. **Nota:** Con el objeto de distinguir las gráficas, ponga como título en cada gráfica el valor de λ .

Valor lambda	Tasa de acierto
0.01	99.9%
0.03	99.9%
0.1	99.7%
0.3	99.7%
1	99.8%
3	49.1%







¿Qué comportamiento observas en la red neuronal cuando se aumenta el parámetro de regularización λ ?

Respuesta:

Se tiende a subajustar los datos, es decir el modelo deja de ser sobreajustado. Pero si aumentamos mucho lambda, el modelo quedaría demasiado subajustado a los datos y no sería capaz de predecir.

¿Cuál es el mejor valor del parámetro λ ?

¿Por qué?

Respuesta:

En este caso el valor 1 es el mejor porque produce un modelo que generaliza sin problemas los datos nuevos, es el que menor sobreajuste genera sin subajustar demasiado los datos.