

Clasificación de flores usando el dataset Iris por medio de una red neuronal

Orlando Hernandez Nuñez. Juan Alejandro Bernal Gallego

05 de Noviembre del 2019

1 Introducción

En el presente trabajo se describe el proceso llevado a cabo para entrenar una red neuronal con el objetivo de clasificar una flor atendiendo a sus características físicas.

2 Dataset

2.1 Descripción del dataset

El dataset con el que se realiza el entrenamiento de la red neuronal se compone de 100 registros o muestras, cada uno de ellos con la medida de longitud y ancho de pétalos y sépalos además de su respectiva clasificación.

sepalength	sepalwidth	petallength	petalwidth	class
5.1	3.5	1.4	0.2	Iris-setosa
7.0	3.2	4.7	1.4	Iris-versicolor
6.1	2.6	5.6	1.4	Iris-virginica

Fig. 1: Estructura y registros del dataset

2.2 Normalización

Antes de realizar el proceso de entrenamiento se optó por normalizar los datos con los cuales va a ser entrenada y evaluada la red neuronal. Este proceso se hizo mediante el método conocido como Min-Max descrito en la Formula 1:

$$v'_i = \frac{v_i - \min_A}{\max_A - \min_A}(\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A$$

donde v'_i será el nuevo valor del atributo que se esta actualizando; v_i es el valor actual; \min_A y \max_A es el mínimo y máximo valor respectivamente de los datos originales; new_max_A y new_min_A es el nuevo máximo y mínimo valor que puede tomar el atributo A .

En este caso se determinó que el valor mínimo y máximo será 0 y 1 respectivamente para cada uno de los atributos del *dataset*.

3 Red neuronal

3.1 Arquitectura de la red neuronal

La red neuronal posee tres capas, una capa de entrada donde se tomarán 4 valores, que corresponden a las características físicas del tamaño de una flor; una capa intermedia oculta la cual posee 1 2 ó 3 neuronas y finalmente, una capa de salida que esta compuesta de una neurona.

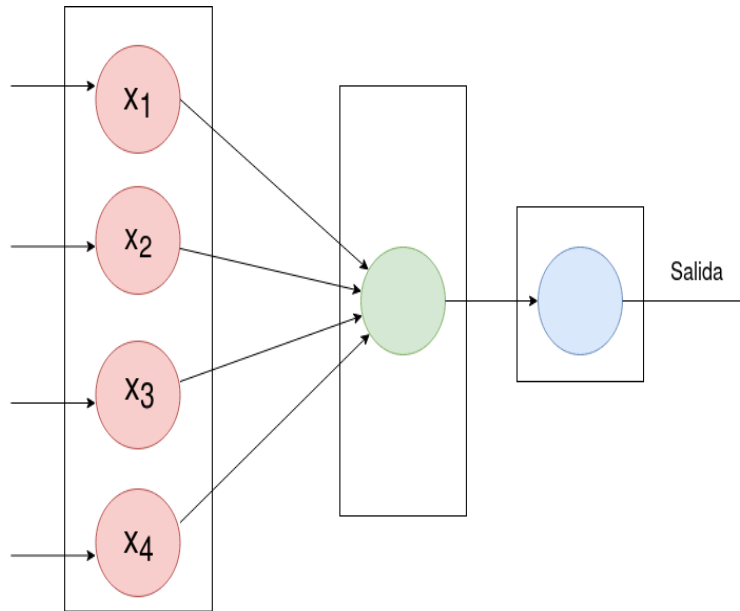


Fig. 2: Red neuronal con 1 neurona en la capa oculta

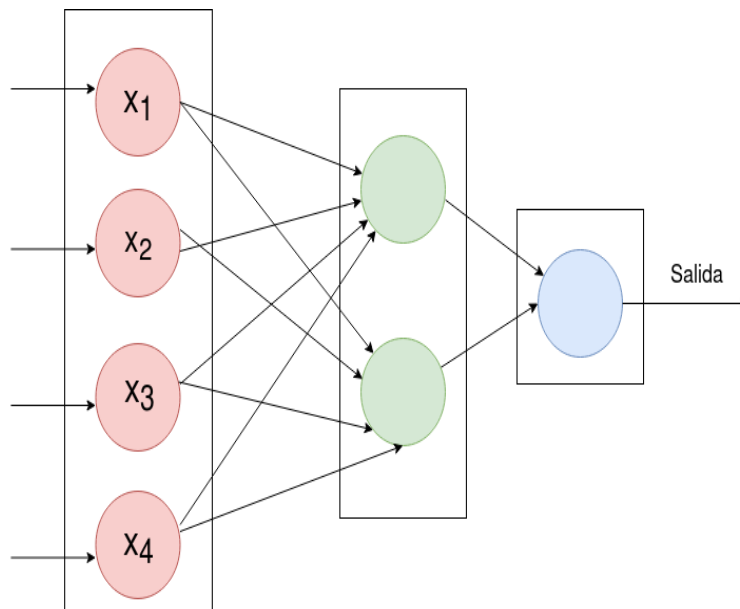


Fig. 3: Red neuronal con 2 neuronas en la capa oculta

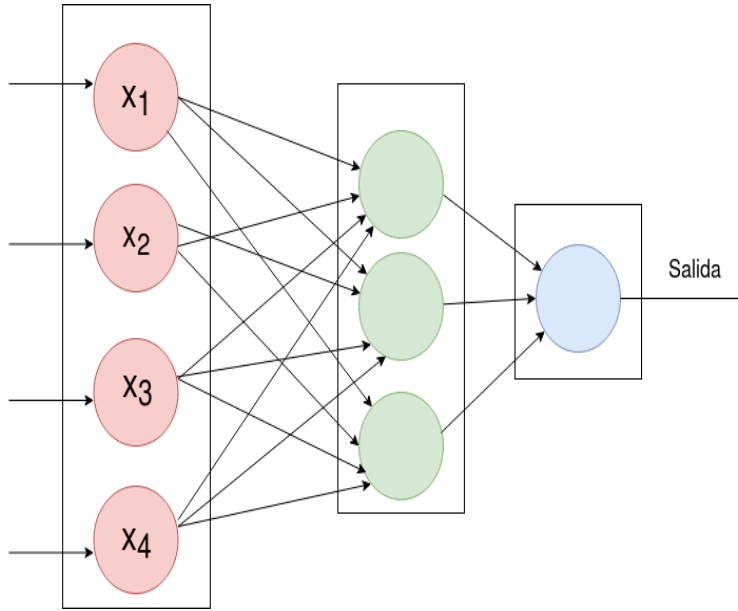


Fig. 4: Red neuronal con 3 neuronas en la capa oculta

3.2 Entrenamiento

De manera aleatoria se escoge el 80% del dataset para entrenar la red neuronal y el porcentaje restante es usado para evaluar los resultados obtenidos una vez hecho el entrenamiento. El entrenamiento se realizó con cada uno de los diseños mostrados en las figuras 2, 3 y 4. Como parámetros de entrada para el proceso de entrenamiento se toma el η o tasa de aprendizaje, el número de épocas a alcanzar y la función de activación. Los pesos al inicio del proceso de entrenamiento son generados de manera aleatoria. Las funciones de activación disponibles son umbral, identidad, lineal a trozos y la función sigmoide.

3.2.1 Entrenamiento con una neurona en la capa oculta

Con $\eta=0.2$ y hasta alcanzar la época 50000 se obtuvo un error de 0.201, en la gráfica se puede apreciar como la muestra de pruebas no alcanza a llegar al promedio de error que tiene la muestra entrenada.

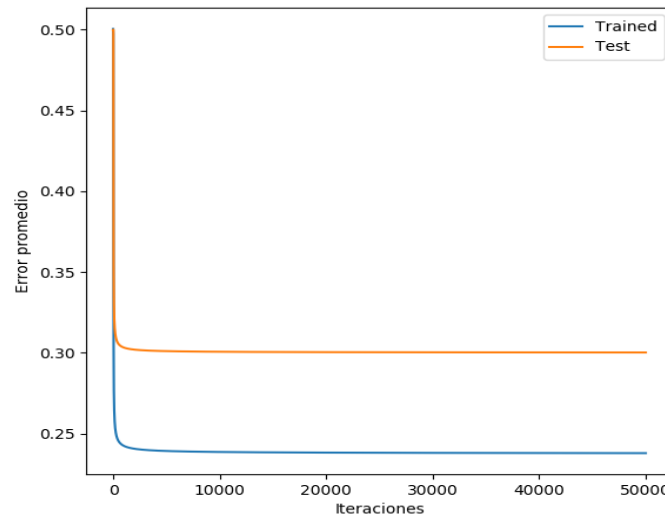


Fig. 5: Entrenamiento con 1 neuronas en la capa oculta

3.2.2 Entrenamiento con dos neuronas en la capa oculta

Con $\eta=0.08$ y hasta alcanzar la época 15000 se consiguió un error de 0.004, en la gráfica 6 se puede observar como el error del grupo de prueba obtiene un error menor hasta la época 5000, después de la muestra de entrenamiento alcanza a la de prueba, obteniendo errores promedio similares.

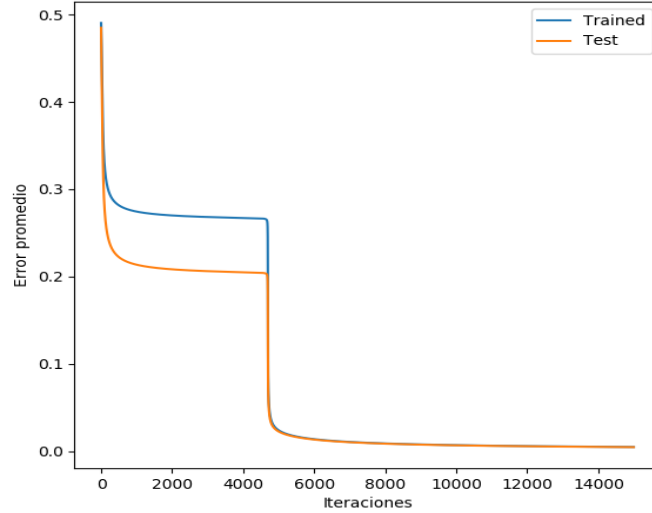


Fig. 6: Entrenamiento con 2 neuronas en la capa oculta

3.2.3 Entrenamiento con tres neuronas en la capa oculta

Los resultados obtenidos bajo esta arquitectura usando $\eta=0.35$ y función de activación sigmoide hasta alcanzar la época 15000 se observan en la figura 7. Se puede observar como el error promedio con los datos de entrenamiento y de prueba se reduce constantemente hasta alcanzar el valor de 0.001460.

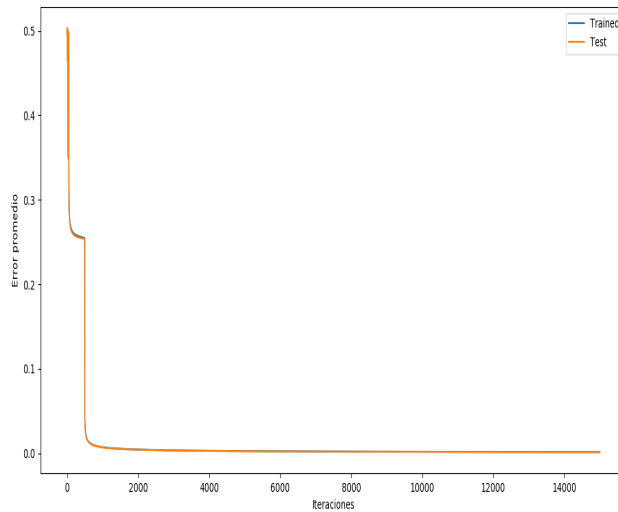


Fig. 7: Entrenamiento con 3 neuronas en la capa oculta