

Universidad de Guadalajara.

Centro Universitario de Ciencias Exactas e Ingenierías.

Ingeniería en computación.

Sección: D05.



Seminario de solución de problemas de inteligencia artificial II.

Practica 1. Ejercicio 3

Mtro. Diego Campos Peña

Montoya Vargas Roberto.

Instrucciones:

Realizar el ejercicio 3 del archivo Práctica1.pdf.

Realiza los programas en el lenguaje de tu elección.

Sube un reporte en formato PDF y los programas correspondientes en un repositorio de git.

Introducción:

Un algoritmo de retropropagación se ejecutará de forma repetida a través de varias iteraciones o épocas de entrenamiento hasta que el error de predicción llegue a un nivel aceptable o converja hacia un mínimo. Es crucial señalar que el éxito del entrenamiento de una red neuronal depende de esto.

Desarrollo:

Como paso inicial en la resolución del problema, vamos a desarrollar un algoritmo de retropropagación específicamente diseñado para nuestro problema de un perceptrón multicapa. Para lograr esto de manera efectiva, necesitamos tener una serie de especificaciones que nos permitan ejemplificarlo correctamente.

El archivo concentrlite.csv, proporcionado por el profesor, contiene dos clases distribuidas en forma concéntrica. Es crucial entender que nuestro objetivo con este algoritmo es capacitarlo para que pueda entrenar y probar el conjunto de datos que le proporcionamos, de modo que adquiera la capacidad de clasificar correctamente el perceptrón multicapa.

La primera etapa del programa, tras cargar las dependencias necesarias, implica la carga y preparación de los datos, los cuales provienen del archivo adjunto en la actividad. Luego, dividimos estos datos en características y etiquetas, tal como se mencionó anteriormente.

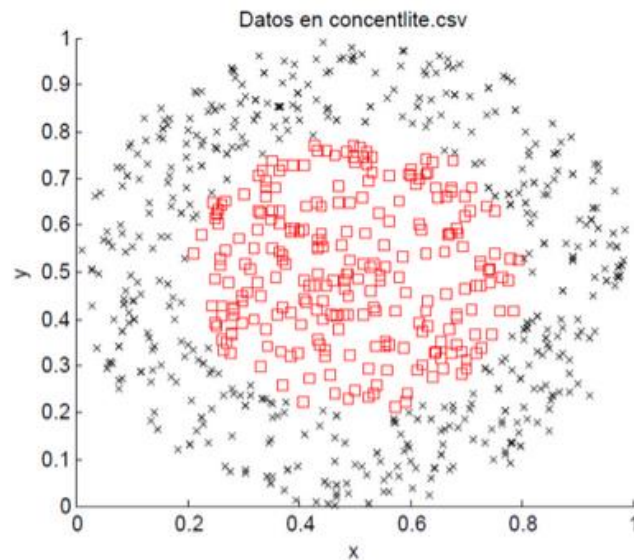
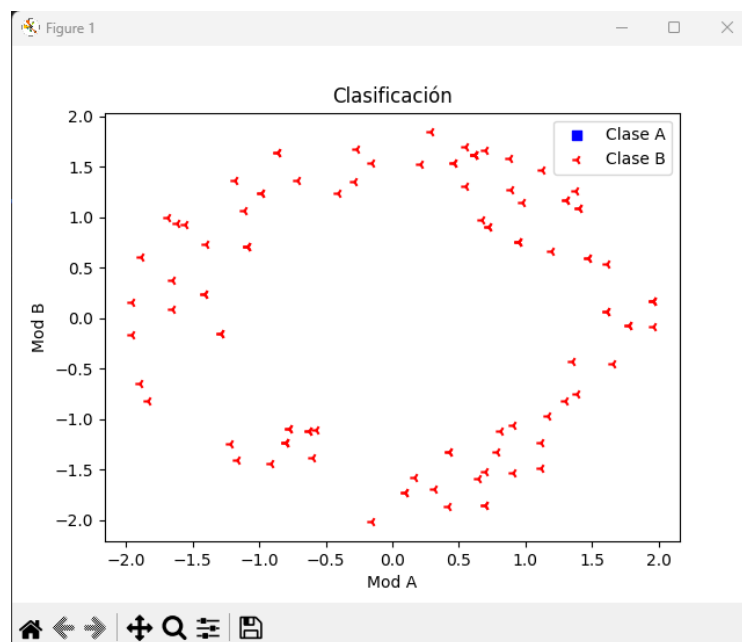


Figura 2. Distribución de clases para el dataset concentrite.

En la segunda fase de la estructuración de la red neuronal, establecemos su arquitectura, definiendo el tamaño de la capa de entrada, las capas ocultas y la capa de salida.



Conclusión:

En resumen, el algoritmo que hemos desarrollado nos brinda una implementación de una red neuronal de retropropagación diseñada para la clasificación binaria. Al cargar nuestros datos en esta implementación, podemos dividirlos entre datos de entrenamiento y de prueba, un paso crucial para entrenar perceptrones multicapa, ajustando los pesos de manera que la red pueda aprender y llevar a cabo tareas de manera automatizada.

Lo que hemos aprendido en esta práctica es la capacidad de la red para aprender de manera eficiente a partir de un conjunto de datos dado, así como el proceso de ajuste de los pesos y sesgos en función del error propagado a través de las capas de la red.