**Reporte sobre la Implementación del Perceptrón con y sin Sesgo**

**Objetivo:**

El objetivo de esta actividad fue implementar un perceptrón para clasificar las compuertas lógicas AND y OR. Se crearon dos versiones del perceptrón:

1. **Perceptrón sin sesgo**, que se entrena para clasificar correctamente la compuerta lógica AND.
2. **Perceptrón con sesgo**, que se entrena para clasificar correctamente la compuerta lógica OR.

Ambas versiones utilizan la función de activación sigmoide y ajustan los pesos y el sesgo mediante el algoritmo de retropropagación, con una tasa de aprendizaje arbitraria. Al final del entrenamiento, se imprime el valor final de los pesos y el sesgo, y se verifica que el perceptrón haya aprendido correctamente las respectivas compuertas lógicas.

**Explicación General del Perceptrón:**

Un **perceptrón** es un modelo matemático simple que intenta simular el funcionamiento de una neurona. En este modelo, las entradas se multiplican por pesos, se suman y luego se pasan a través de una función de activación para obtener la salida. La principal tarea del perceptrón es aprender los valores de los pesos para que las salidas del modelo coincidan con las salidas deseadas (en este caso, los resultados de las compuertas lógicas).

La función de activación utilizada es la **sigmoide**, que es una función no lineal que mapea cualquier valor real entre 0 y 1. Esto es ideal para problemas de clasificación binaria como el de las compuertas lógicas.

**Implementación del Perceptrón Sin Sesgo (PerceptronSinSesgo):**

1. **Entrenamiento del Perceptrón Sin Sesgo:** En la clase PerceptronSinSesgo, la única diferencia con el perceptrón estándar es que el sesgo se fija en cero. El entrenamiento se realiza ajustando únicamente los pesos de acuerdo con el error de cada predicción. El algoritmo se repite durante el número de épocas especificado, o hasta que el error total sea menor que un valor umbral (en este caso, 0.01).
2. **Funcionamiento del Perceptrón Sin Sesgo:**
   * Inicialización: Los pesos se inicializan aleatoriamente y el sesgo se fija en cero.
   * Entrenamiento: Para cada entrada, se calcula la suma ponderada de las entradas multiplicadas por sus respectivos pesos. Luego, se aplica la función de activación para obtener la salida. El error se calcula como la diferencia entre la salida deseada y la salida predicha.
   * Actualización de pesos: Los pesos se ajustan en función del error, la tasa de aprendizaje y las entradas correspondientes. El sesgo no se actualiza, ya que está fijo en cero.
   * Convergencia: Si el error total es menor que 0.01, se considera que el perceptrón ha aprendido correctamente.

**Implementación del Perceptrón con Sesgo (PerceptronConSesgo):**

1. **Entrenamiento del Perceptrón con Sesgo:** La clase PerceptronConSesgo extiende la clase Perceptron, lo que significa que utiliza el mismo procedimiento general de entrenamiento. La diferencia es que el sesgo se ajusta durante el entrenamiento. En lugar de fijarlo en cero, el sesgo también se actualiza en cada iteración del entrenamiento, igual que los pesos.
2. **Funcionamiento del Perceptrón con Sesgo:**
   * Inicialización: Al igual que en el perceptrón sin sesgo, los pesos se inicializan aleatoriamente. Sin embargo, en este caso, el sesgo es también una variable que se inicializa aleatoriamente y se ajusta durante el proceso de entrenamiento.
   * Entrenamiento: Para cada entrada, se calcula la suma ponderada de las entradas multiplicadas por los pesos y se le añade el sesgo. Luego, se aplica la función de activación para obtener la salida. El error se calcula y se utiliza para ajustar tanto los pesos como el sesgo.
   * Convergencia: Al igual que en el caso del perceptrón sin sesgo, el algoritmo se detiene si el error total es suficientemente pequeño (menor que 0.01).

**Códigos Implementados:**

1. **Clase Perceptron:** Esta clase es la base para ambos tipos de perceptrón. Contiene métodos para inicializar los pesos, calcular la suma ponderada de las entradas, y aplicar la función de activación sigmoide. Además, incluye el método train para entrenar el perceptrón mediante el algoritmo de retropropagación.
2. **Clase PerceptronSinSesgo:** Esta clase hereda de la clase Perceptron, pero modifica la forma en que se entrenan los pesos al no utilizar un sesgo. La función train ajusta únicamente los pesos durante el entrenamiento.
3. **Clase PerceptronConSesgo:** Similar a PerceptronSinSesgo, pero con la diferencia de que también ajusta el sesgo durante el entrenamiento. La clase sobrescribe el método train para incorporar la actualización del sesgo.

**Proceso de Entrenamiento:**

1. **Entrenamiento para la compuerta lógica AND:** El perceptrón sin sesgo se entrena para aprender los resultados de la compuerta AND, que tiene como entradas dos valores binarios (0 o 1) y una salida binaria. Los patrones de entrada son:

yaml

CopiarEditar

Entrada: [0, 0] -> Salida: 0

Entrada: [0, 1] -> Salida: 0

Entrada: [1, 0] -> Salida: 0

Entrada: [1, 1] -> Salida: 1

1. **Entrenamiento para la compuerta lógica OR:** El perceptrón con sesgo se entrena para aprender los resultados de la compuerta OR, que también tiene dos entradas y una salida binaria. Los patrones de entrada son:

yaml

CopiarEditar

Entrada: [0, 0] -> Salida: 0

Entrada: [0, 1] -> Salida: 1

Entrada: [1, 0] -> Salida: 1

Entrada: [1, 1] -> Salida: 1

**Resultados Esperados:**

1. **Perceptrón Sin Sesgo:** Después de entrenar el perceptrón sin sesgo, el modelo debería ser capaz de clasificar correctamente las entradas de la compuerta AND. Al final del entrenamiento, los pesos finales y el valor del sesgo (que debe ser 0) se imprimen, junto con los cálculos de la suma ponderada, la activación y el error para cada patrón de entrada.
2. **Perceptrón Con Sesgo:** El perceptrón con sesgo debería clasificar correctamente las entradas de la compuerta OR. Al igual que en el perceptrón sin sesgo, se imprimirán los valores de los pesos y el sesgo finales, junto con los cálculos paso a paso.

**Conclusión:**

La actividad permitió implementar y entender cómo funciona un perceptrón para clasificar problemas de clasificación binaria. La diferencia entre un perceptrón con y sin sesgo radica en la presencia de un término adicional (el sesgo) que se ajusta durante el entrenamiento. El uso de la función de activación sigmoide es esencial para aplicar el perceptrón en problemas de clasificación binaria, y el entrenamiento muestra cómo los pesos y el sesgo se ajustan iterativamente para minimizar el error en las predicciones.

A través de este proceso, se validó que tanto el perceptrón sin sesgo como el perceptrón con sesgo son capaces de aprender correctamente las compuertas lógicas AND y OR, respectivamente, y el algoritmo de retropropagación permite ajustar los parámetros del modelo para lograr una clasificación correcta.