

## 1. Resumen

Se implemento un perceptrón con distintas reglas de aprendizaje para su comparación y se le intento enseñar las funciones booleanas de **AND**, **OR** y **XOR**. Para esto se realizaron diversos experimentos con distintas tasas de aprendizaje y reglas de aprendizaje. La evolución del aprendizaje del perceptrón se ve reflejada mediante el error  $E$  obtenido con el número de iteraciones correspondiente. Los resultados obtenidos son presentados y discutidos mas adelante.

## 2. Detalles de Implementación/Experimentación

Se utilizó el lenguaje de programación C++ para la representación del perceptrón y la realización del programa principal de experimentos. El perceptrón se modelo mediante una clase con métodos relevantes para el estudio como por ejemplo **entrenar** y **procesar** que entrenaban al perceptrón y probaban los resultados obtenidos respectivamente. Se utilizaron como pesos iniciales 0.1 y de tasa de aprendizaje se utilizó 0.01, 0.1, 0.2, 0.5 y 0.99 para los experimentos. Para la primera sección de experimentos se dejó como numero máximo de iteraciones 10 y para la segunda 100000.

## 3. Presentación y discusión de los resultados

### 3.1. Perceptrón con $n$ entradas

#### 3.1.1. Gráficas del error $E$

A raíz de las gráficas 1,2 y 3 se puede observar que las funciones **AND** y **OR** convergen rápidamente a un error aceptable, es decir, el perceptrón “aprende” correctamente estas funciones y se esto puede comprobar usando el método **procesar** de nuestra clase **Perceptron**. Sin embargo la función **XOR** no converge a un error aceptable, peor aún, esta parece empeora mientras mas iteraciones pasan hasta que se estabiliza en un error constante. La función **XOR** no puede aprenderse con un solo perceptrón pues esta es una función no separable linealmente.

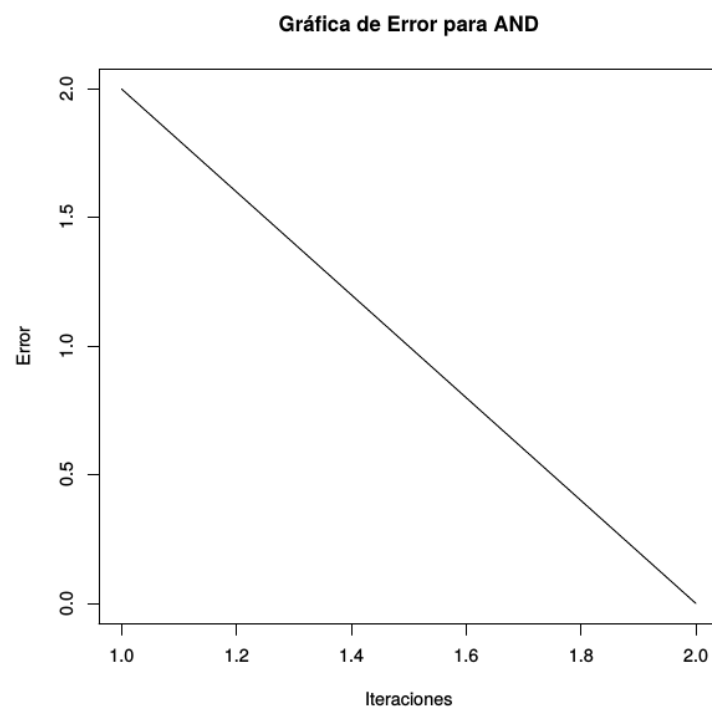


Figura 1: Gráfica de error para la función AND con una tasa de aprendizaje de 0,1.

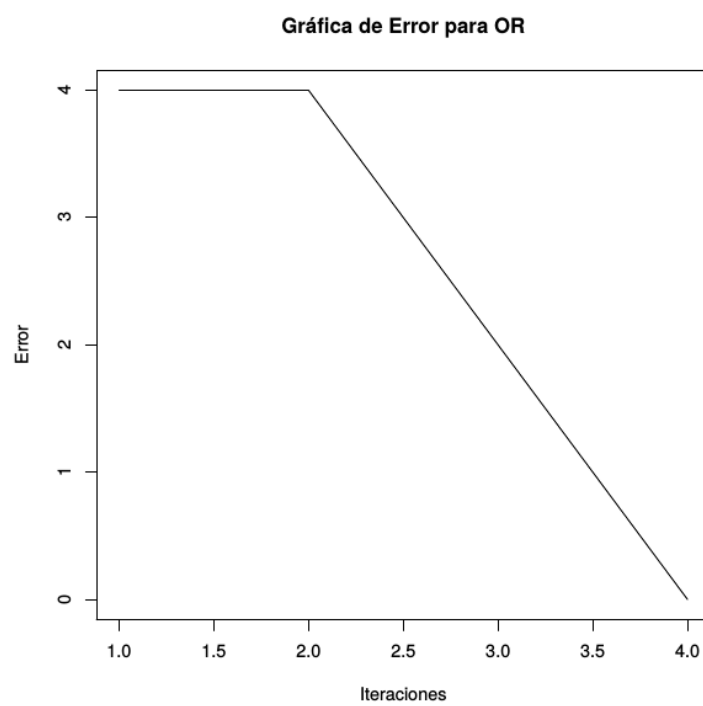


Figura 2: Gráfica de error para la función OR con una tasa de aprendizaje de 0,1.

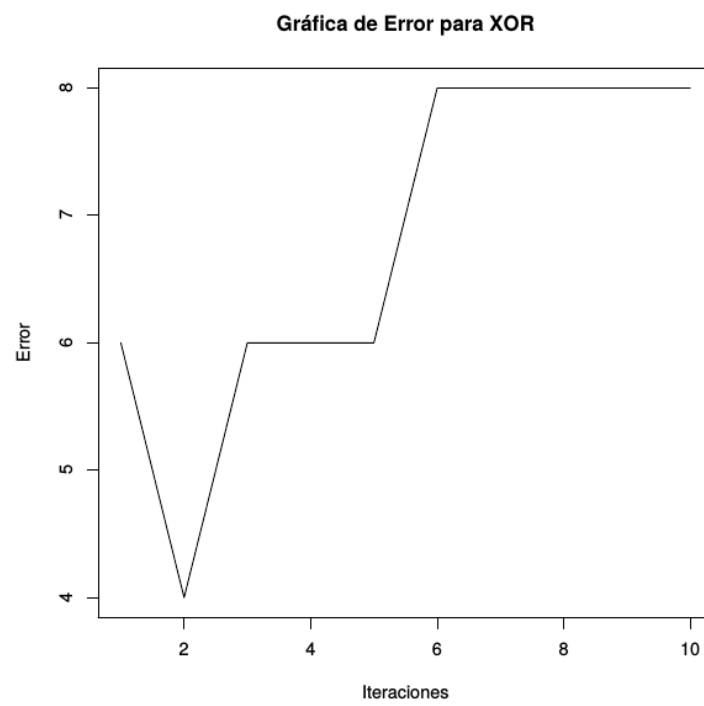


Figura 3: Gráfica de error para la función XOR con una tasa de aprendizaje de 0,1.

### 3.1.2. Pruebas con distintas tasas de aprendizaje

Se realizaron pruebas para las funciones AND y OR con los siguientes valores para la tasa de aprendizaje: 0,01, 0,1, 0,2, 0,5 y 0,99. Para una tasa de aprendizaje de 0,01 las funciones AND y OR tardan 4 y 6 iteraciones respectivamente, un resultado peor que obtenido con tasa de aprendizaje 0,1: 2 iteraciones para AND y 4 para OR (estos resultados se puede observar en las gráficas anteriores). Con tasas de aprendizaje superior o iguales a 0,2 se tienen 6 iteraciones para AND y 4 para la función OR. Esto nos indica que cuando la constante es demasiado pequeña se converge “gastando” un cierto numero de iteraciones pues la variación de los pesos varia menos pero si es muy grande el error fluctúa con mayor frecuencia lo que dificulta el aprendizaje y no asegura convergencia.

## 3.2. Regla de entrenamiento delta para una neurona artificial de $n$ entradas

### 3.2.1. Gráficas del error $E$

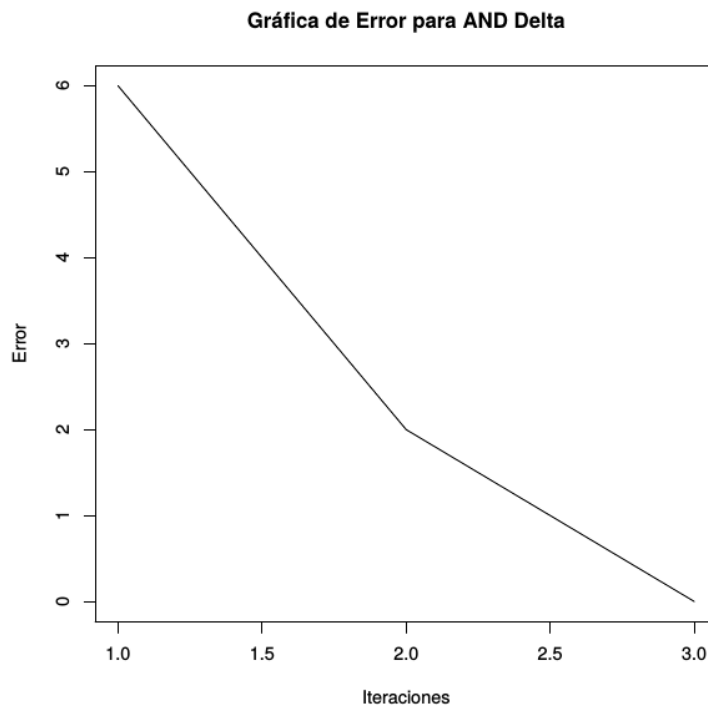


Figura 4: Gráfica de error para la función AND con una tasa de aprendizaje de 0,05 usando la regla de entrenamiento delta.

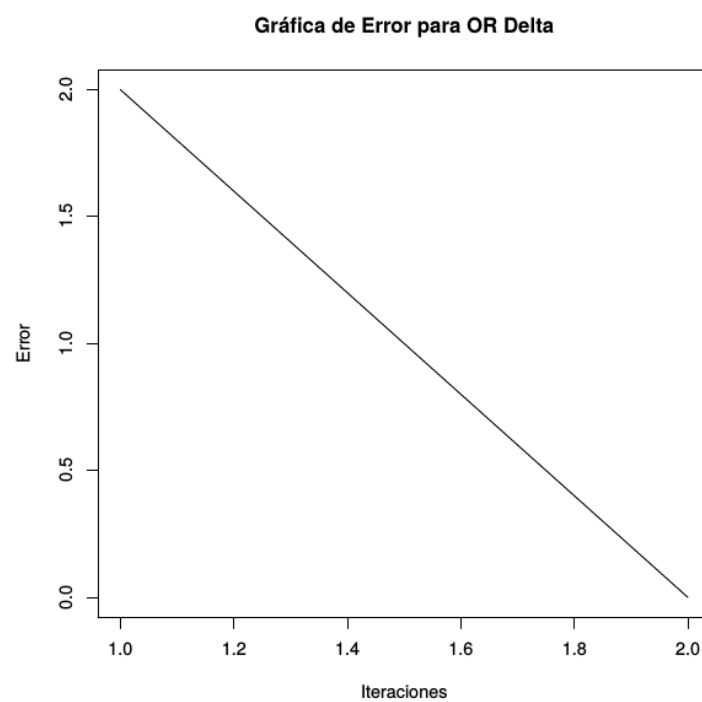


Figura 5: Gráfica de error para la función OR con una tasa de aprendizaje de 0,05 usando la regla de entrenamiento delta.

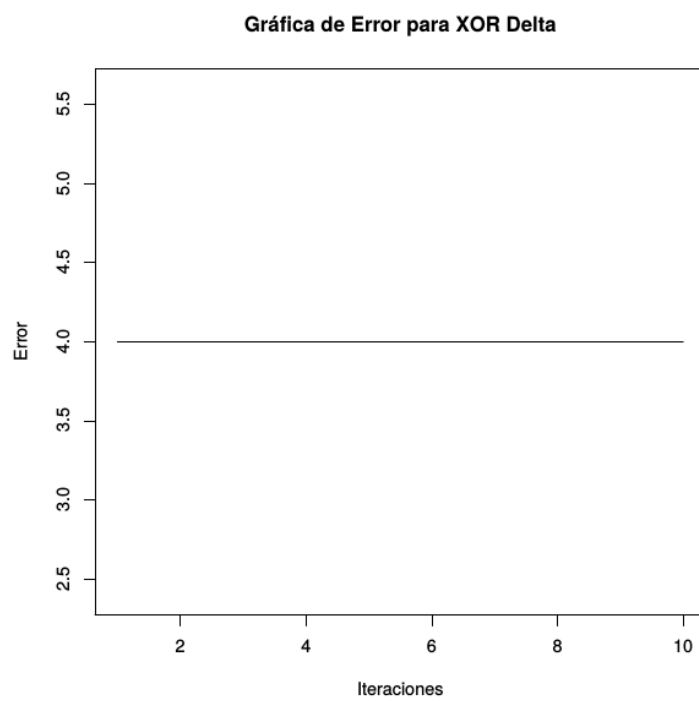


Figura 6: Gráfica de error para la función XOR con una tasa de aprendizaje de 0,05 usando la regla de entrenamiento delta.

Según los resultados que se pueden ver en las gráficas 4, 5 y 6 se nota que la regla de entrenamiento delta obtiene resultados superiores a los obtenidos mediante la regla de entrenamiento del perceptrón cuando la tasa de aprendizaje es muy pequeña. Observaremos en resultados posteriores que con una tasa de entrenamiento mayor (0,1) obtenemos resultados peores que los obtenidos por el perceptrón. Por otro lado, se observa que con la función XOR el error es constante.

### **3.2.2. Pruebas con distintas tasas de aprendizaje**

Se realizaron experimentos con tasa de aprendizaje de 0.01, 0.1, 0.2, 0.5 y 0.99; para un experimento se usó tasa de aprendizaje constante mientras que para otro esta era dividida según el numero de iteración. Al principio, el primer caso logra converger con 0.01 (con 9 y 90 iteraciones para AND y OR), 0.1 (10 y 10 iteraciones para AND y OR) y 0.2 (6 y 6 iteraciones con AND y OR). Se advirtió que para tasas de aprendizaje mayores o iguales a 0.313 no convergían ni AND ni OR. Usando decaimiento de la tasa de aprendizaje tenemos resultados bastante mediocres para las primeras tasas mas pequeñas como por ejemplo 1038 para el AND y no convergencia para el OR, sin embargo mientras mas subía la tasa menos iteraciones eran necesarias mediante este método. El error de XOR se mantenía constante para el primer caso mientras que para el segundo caso este pudo reducirse a 2 lo que no se había observado en ningún otro experimento.