Redes Neuronales (POO)

Pablo Antonio Stack Sánchez
Programación y Algoritmos

26 de noviembre de 2019

1. Introducción

Las redes neuronales son un modelo computacional inspirado en el funcionamiento del sistema nervioso humano. Las unidades básicas son las neuronas, que generalmente se organizan en capas, como se muestra en la siguiente figura.

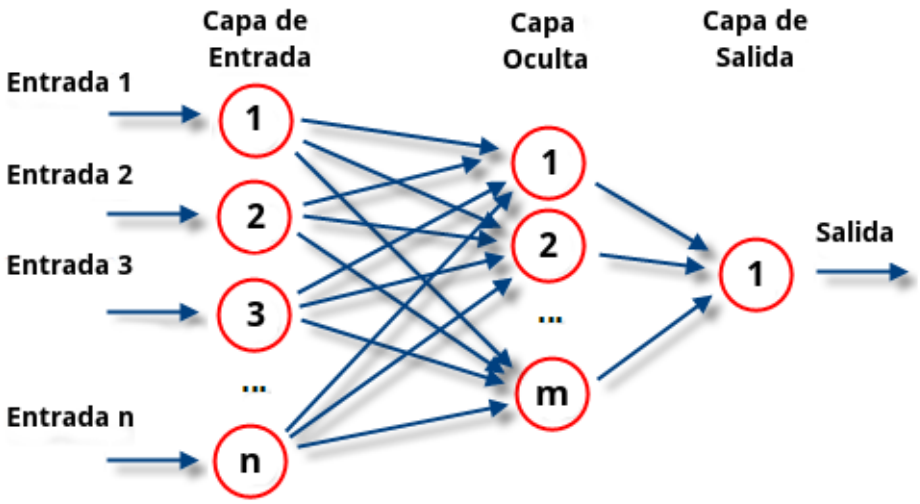

Figura 1: Red Neuronal

Cada neurona está conectada con otras a través de enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso:

$$a^l = \sum_{i=0}^{m} W^l a^{l-1} + b$$

En donde a^l es el valor de salida de la función de activación $\sigma = \frac{1}{1+e^{-x}}$ y b es e bias.
La red aprende examinando los registros individuales, generando una predicción para cada registro y realizando ajustes a los pesos cuando realiza una predicción incorrecta. Este proceso se repite muchas veces y la red sigue mejorando sus predicciones.

2. Metodología

Se programó la red neuronal utilizando clases y el paradigma orientado a objetos, se crearon tres clases diferentes:

1. Neurona: La clase Neurona se encarga de iniciaizar aleatoriamente los pesos y los bias.
2. Capa: Vector de neuronas que se encarga de calcular los pesos y las activaciones de las neuronas.
3. Red_Neuronal: Vector de capas que se encarga de hacer la propagación hacia adelante y hacia atrás.

Para la red neuronal se utilizó el error cuadrático medio:

$$\frac{1}{n}\sum_{i=0}^{n}(y-\hat{y})^2$$

En donde y es el valor esperado y $\hat{y}$ el valor predicho.
Se crearon métodos para entrenar y guardar los pesos y los bias resultantes. También se creó un método para cargar estos pesos obtenidos del entrenamiento en una nueva red neuronal para el proceso de prueba. Se entrenó con 80 % de los datos y se probó con el resto.

3. Resultados

Para tener una idea del proceso de entrenamiento de la red se graficó el error, el resultado se observa en la siguiente figura.

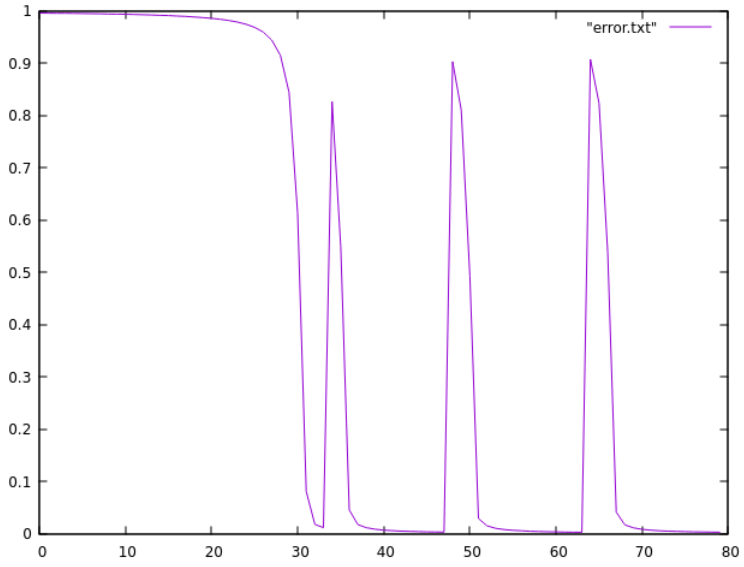

Figura 2: Red Neuronal

Se observa que el error disminuye conforme el entrenamiento avanza. El último error de entrenamiento fue: 0.00248388.
Para la prueba de clasificación se utilizó los 20 datos restantes, en primera instancia se obtuvo un accuracy del 100 % . Después de analizar los valores esperados de los datos se observó que el valor de los ultimos 20 datos es 1.
Posteriormente se decidió clasificar todos los datos y se obervó que los valores predichos por la red neuronal son siempre 1. Por lo que el accuracy de la red neuronal programada es del 50 %.

4. Conclusiones

La propagación hacia atrás es un algoritmo dificíl de programar en c++, hay que ser muy cuidadoso con los índices. A pesar de haber programado este método de actualización de pesos basado en el descenso de gradiente no se obtuvo un buen rendimiento de la red.
Probablemente se pueda mejorar la forma en la que se calcula el error, sería interesante probar con otra función de activación y ver como cambian los resultados de activación, probablemente al ser un problema binario de clasificación la función de entropia cruzada tenga un buen desempeño.
El código se podría mejorar acomodando en distintos archivos las clases programadas y creando una clase para las funciones de activación y para el calculo del error.
Se dejó la arquitectura de la red con 8 neuronas en la primera capa, 16 en la intermedia y 1 de salida, pero la configuración puede ser cambiada por el usuario, se pueden agregar más capas y neuronas.