

Perceptron para la gráfica de la compuerta and.

Desarrollo:

En su navegador, ejecute el código siguiente(también puede obtener el código en el archivo “perceptronANDsimple.html”):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
<script>
  // Datos de entrenamiento (X) y etiquetas (y)
  const X = [[0, 0], [0, 1], [1, 0], [1, 1]];
  const y = [0, 0, 0, 1];

  // Inicialización de pesos y bias
  let w = [Math.random(), Math.random()];
  let b = Math.random(); //bias (b) o sesgo equivale a w3
  const eta = 0.1; // Tasa de aprendizaje

  // Función de propagación hacia adelante
  const forward = (x) => {
    const sum = w[0] * x[0] + w[1] * x[1] + b;
    return sum > 0 ? 1 : 0;
  };

  // Entrenamiento
  for (let epoch = 0; epoch < 100; epoch++) { // Número de
iteraciones
    for (let i = 0; i < X.length; i++) {
      const y_pred = forward(X[i]);
      const e = y[i] - y_pred;
```

```

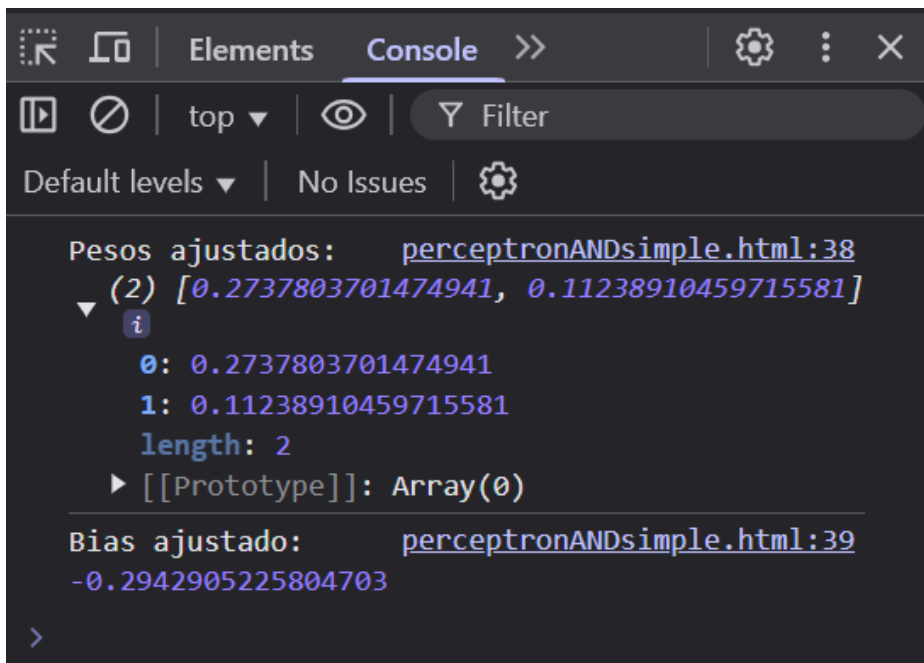
        w[0] = w[0] + eta * e * X[i][0];
        w[1] = w[1] + eta * e * X[i][1];
        b = b + eta * e;
    }
}

console.log("Pesos ajustados:", w);
console.log("Bias ajustado:", b);

</script>
</html>

```

Como resultado de la ejecución de este archivo, en la consola del navegador tendrá un resultado similar a este (recuerde que puede mostrar la consola del navegador con la tecla f12):



En mi ejecución obtuve:

$w_0=0.18$

$w_1=0.38$

Bias o $w_2= -0.5$

Por tanto, para la función $w_0x_0+w_1x_1+w_2$ o cambiando variables

$$w_0x + w_1y + w_2$$

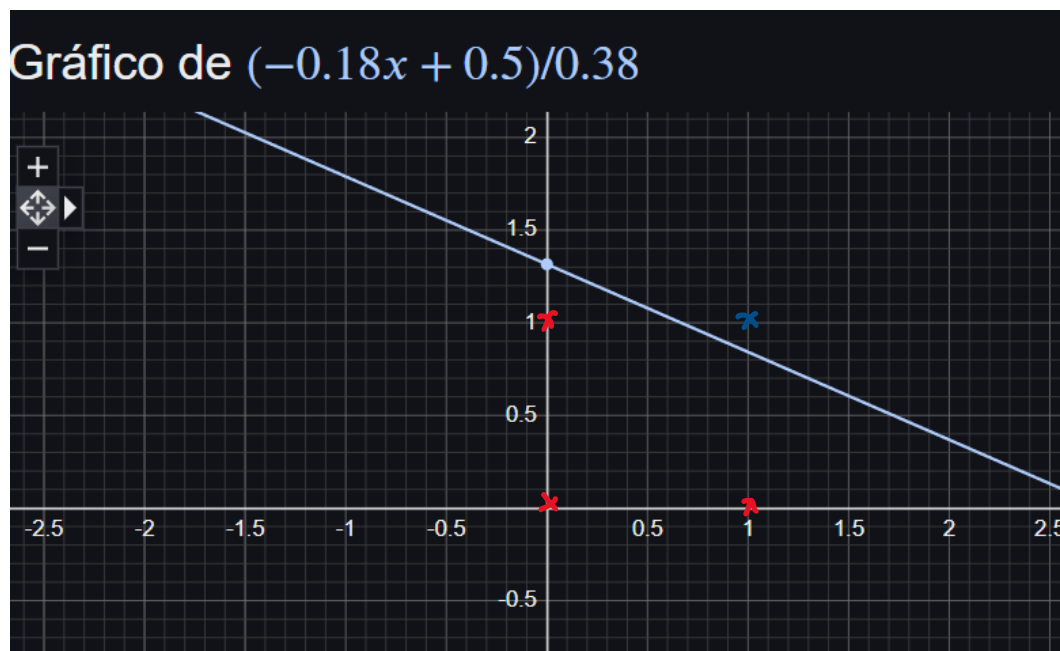
$$0.18x + 0.38y - 0.5$$

Despejando y, tenemos la función

$$Y = (-0.18x + 0.5) / 0.38$$

.

Graficando esa función en Chrome:



Los valores rojos y azules graficados corresponden a

X0	X1	and	
0	0	0	rojo
0	1	0	rojo
1	0	0	rojo
1	1	1	azul

Como vemos separa correctamente ambos conjuntos.