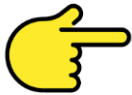


REDES NEURONALES Y LÓGICA DIFUSA



Material
Código Predicción

DOCENTE: ALDO ALARCÓN SUCASACA
2021



Señor estudiante: Por favor, replicar el código de
predicción de ventas

Código Sistema Inteligente

```
index.html.html  script.js  x
1
2 <html lang="es">
3 <head>
4   <title>SISTEMA INTELIGENTE PREDICCIÓN DE VENTAS</title>
5   <meta charset="UTF-8">
6   <meta name="title" content="SISTEMA INTELIGENTE">
7   <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs/dist/tf.min.js"></script>
8   <script src="https://cdn.jsdelivr.net/npm/chart.js@2.7.3/dist/chart.bundle.min.js"></script>
9
10 </head>
11 <body>
12 <center><table border="0"></center>
13 <tbody>
14 <tr>
15   <td>REPETICIONES</td>
16   <td><input type="number" id="repeticiones" value="50"/></td>
17 </tr>
18 <tr>
19   <td>VX_DÍA DE VENTA</td>
20   <td><input type="number" id="nuevoValX" value="5"/></td>
21 </tr>
22 <tr>
23   <td></td>
24   <td><input type="button" value="Calcular" name="calcular" id="calcular" on
25     click="learnLinear()" /> </td>
26 </tr>
27 <tr>
28   <td>VY_¿Cuánto serán mis ventas?</td>
29   <td><span id="valy"></span> </td>
30 </tr>
31 <tr>
32   <td>EPOCAS</td>
33   <td><span id="epocas"></span> </td>
34 </tr>
35 </tbody>
36 </table>
37 <canvas id="myChart" width="400" height="300"></canvas>
38 <script>
39
40   var valX = [1 , 2 , 3 , 4 , 5 , 6];
41   var valY = [100, 110, 90, 80, 150, 130];
42   var datosGrafica=deArrayAMatriz(valX, valY);
43   // Inicializamos la Gráfica
```

```
44     var grafica = new Chart(document.getElementById("myChart"), {
45         type: 'scatter',
46         data: {
47             datasets: [{
48                 label: "PREDICIENDO MIS VENTAS UNAMAD",
49                 data: datosGrafica,
50                 borderColor: "orange",
51             }]
52         },
53         options: {
54             responsive: false
55         }
56     });
57
58
59     async function learnLinear() {
60         const model = tf.sequential();
61         model.add(tf.layers.dense({units: 1, inputShape: [1]}));
62         model.compile({loss: 'meanSquaredError', optimizer: 'sgd'});
63         const xs = tf.tensor2d(valX, [6, 1]);
64         const ys = tf.tensor2d(valY, [6, 1]);
65         var epocas = +document.getElementById("repeticiones").value;
66         var nuevoValX = +document.getElementById("nuevoValX").value;
67
68         for (i = 0; i < epocas; i++) {
69             await model.fit(xs, ys, {epochs: 1});
70             var prediccionY = model.predict(tf.tensor2d([nuevoValX], [1, 1])).dataSync()[0];
71
72             document.getElementById("valY").innerText = prediccionY;
73             document.getElementById("epocas").innerText = i+1;
74
75             datosGrafica.push({x:nuevoValX,y:prediccionY});
76             grafica.data.datasets[0].data = datosGrafica;
77             grafica.update();
78         }
79     }
80     function deArrayAMatriz(arx, ary) {
81         var data = [];
82         for (i = 0; i < arx.length; i++) {
83             data.push({x: arx[i], y: ary[i]});
84         }
85         return data;
86     }
87
88     </script>
89 </body>
90 </html>
```

```
index.html.html  script.js

1
2   var valX = [1 , 2 , 3 , 4 , 5 , 6];
3   var valY = [100, 110, 90, 80, 150, 130];
4   var datosGrafica=deArrayAMatriz(valX, valY);
5
6   var grafica = new Chart(document.getElementById("myChart"), {
7     type: 'scatter',
8     data: {
9       datasets: [{
10         label: "MIS VENTAS",
11         data: datosGrafica,
12         borderColor: "black",
13       }]
14     },
15     options: {
16       responsive: false
17     }
18   });
19
20
21 async function learnLinear() {
22   const model = tf.sequential();
23   model.add(tf.layers.dense({units: 1, inputShape: [1]}));
24   model.compile({loss: 'meanSquaredError', optimizer: 'sgd'});
25
26   const xs = tf.tensor2d(valX, [6, 1]);
27   const ys = tf.tensor2d(valY, [6, 1]);
28
29   var epocas = +document.getElementById("repeticiones").value;
30   var nuevoValX = +document.getElementById("nuevoValX").value;
31
32   for (i = 0; i < epocas; i++) {
33     await model.fit(xs, ys, {epochs: 1});
34     var prediccionY = model.predict(tf.tensor2d([nuevoValX], [1, 1])).dataSync()[0];
35     document.getElementById("valy").innerText = prediccionY;
36     document.getElementById("epocas").innerText = i+1;
37
38     datosGrafica.push({x:nuevoValX,y:prediccionY});
39     grafica.data.datasets[0].data = datosGrafica;
40     grafica.update();
41   }
42 }
43
44 function deArrayAMatriz(arx, ary) {
45   var data = [];
46   for (i = 0; i < arx.length; i++) {
47     data.push({x: arx[i], y: ary[i]});
48   }
49   return data;
50 }
```