Universidad de Guadalajara

Centro universitario de ciencias exactas e ingenierías

Sistemas inteligentes

Impartida por: *M.C* JOSE DE JESUS HERNANDEZ BARRAGAN

Actividad 4

Detección de colores con Red Neuronal MLP

Realizada por: OMAR BARUCH MORON LOPEZ

Omar Baruch Morón López
Smart Systems

OBJETIVOS

El objetivo de esta práctica es realizar por medio de una red neuronal MLP (perceptrón multicapa) la clasificación de colores, gracias al sensor de TCS230 que convierte el color en frecuencias medibles.

Desarrollo:

Por medio del sensor TCS230 se obtiene la configuración RGB por medio de 3 frecuencias, en nuestro caso cada frecuencia será una entrada de nuestra red neuronal como se muestra en la siguiente imagen:

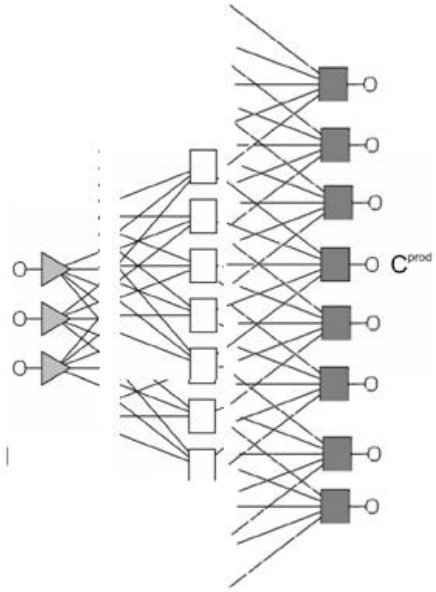

Ilustración 1: Estructura Neuronal Implementada

En esta estructura consideraremos a las entradas como los parámetros “R” “G” “B” y a las neuronas de salida como cada color que clasificara.

En el anexo podemos ver el set de entrenamiento utilizado y se puede ver la composición RGB de cada prueba de color utilizada.

La detección de la frecuencia que as u vez se parametrizaba entre 1 a 255 se programó por medio de la IDE de Arduino con el código anexo 4 y por medio del puerto serial se obtenían los datos en Matlab con el código anexo 2. Y así se podía obtener el RGB que leía el sensor a cada color.

Resultados

Como resultado del entrenamiento con:

Capas ocultas = 7

Aprendizaje=.8;

Iteración=10,000

Se obtuvieron los siguientes errores en cada neurona de salida o en otras palabras en la detección de cada color.

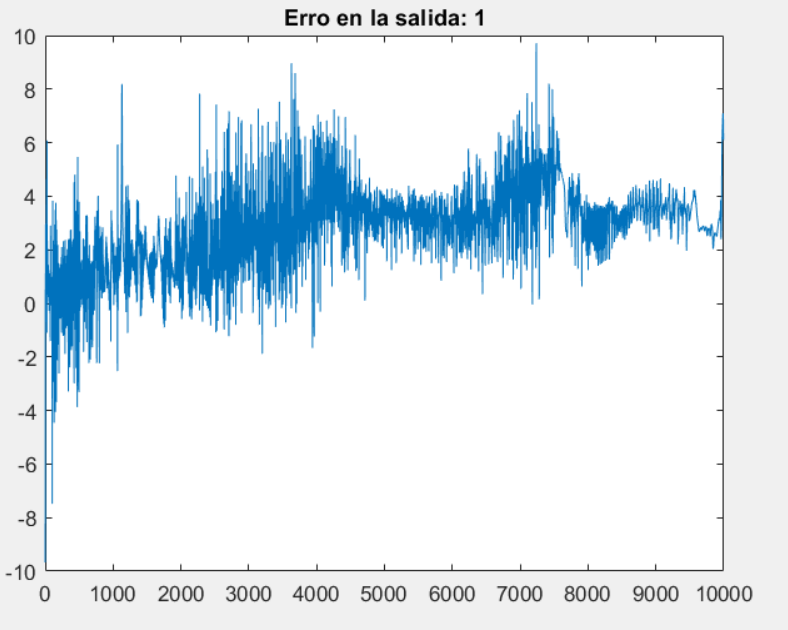

Ilustración 2:Color Blanco

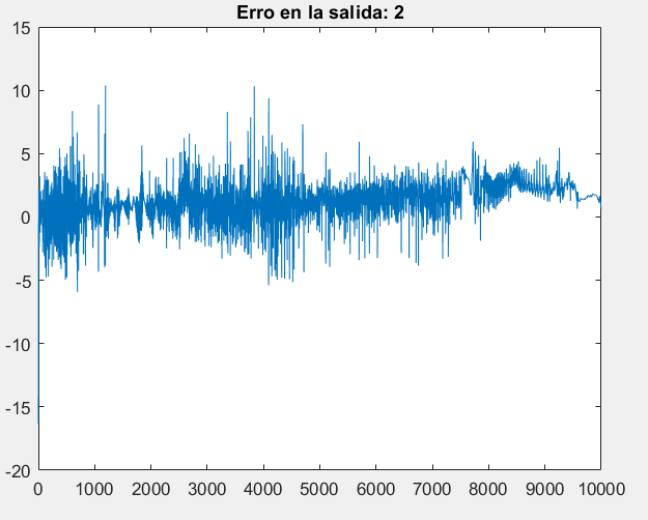

Ilustración 3:Color Verde

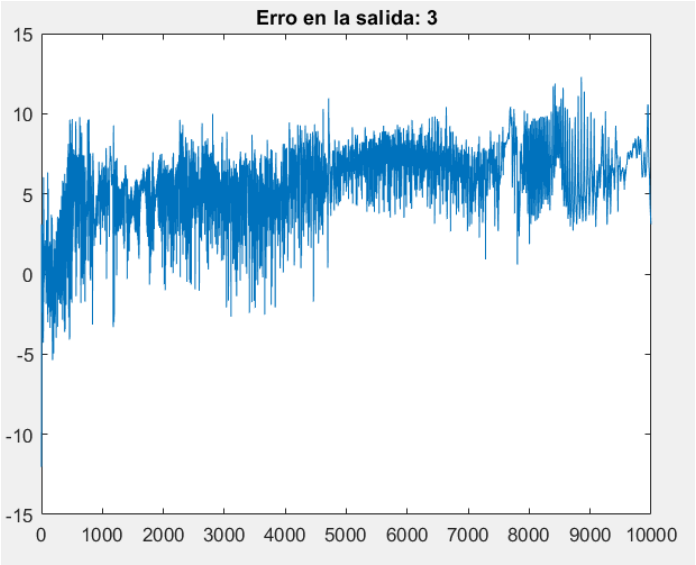

Ilustración 4: Color Azul

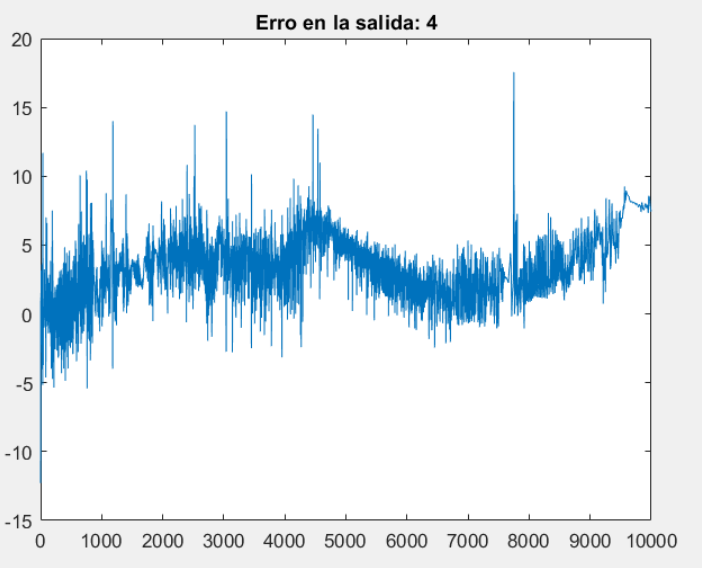

Ilustración 5: Color Cyan

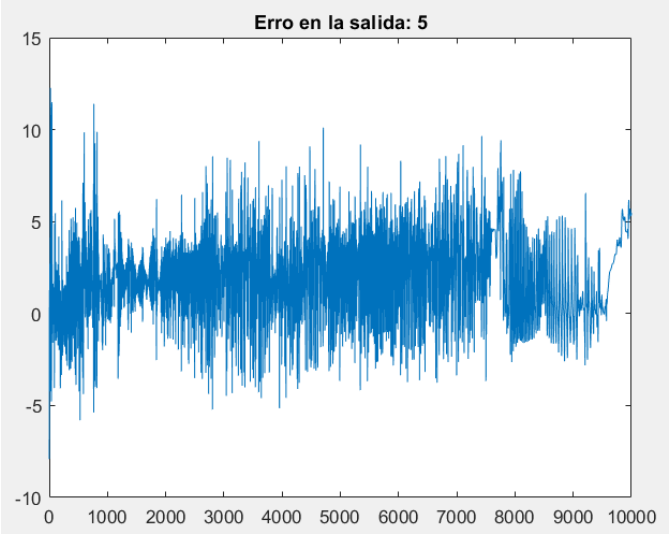

Ilustración 6: Color Negro

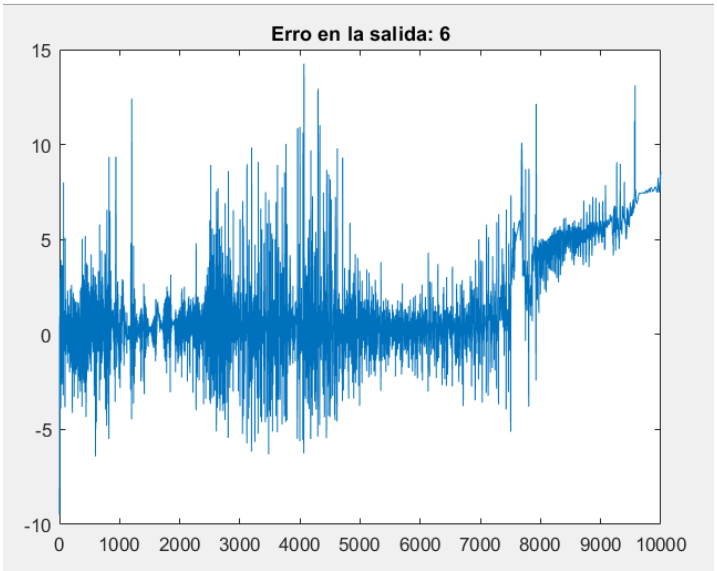

Ilustración 7: Color Amarillo

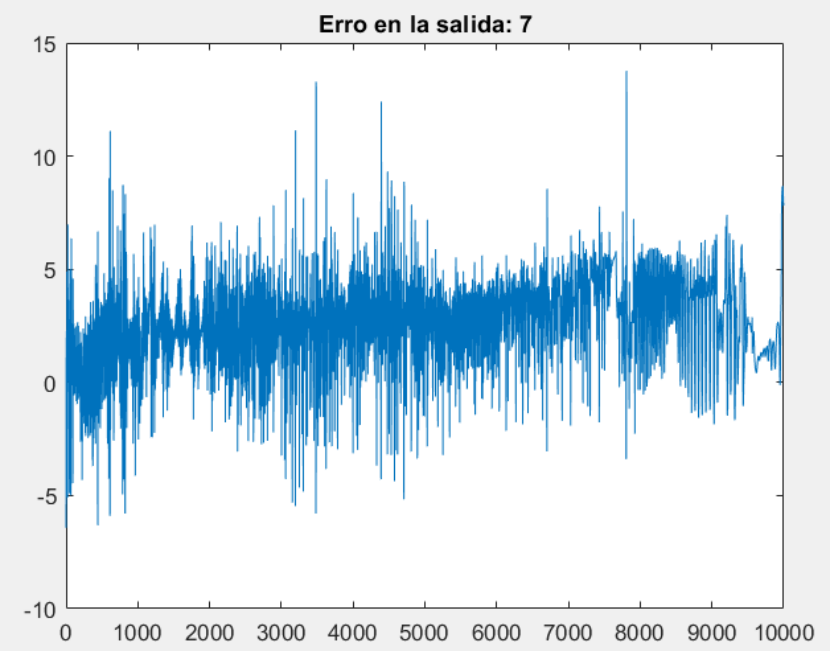

Ilustración 8:Color Rojo

Omar Baruch Morón López
Robotics student of the University of Guadalajara
Smart Systems

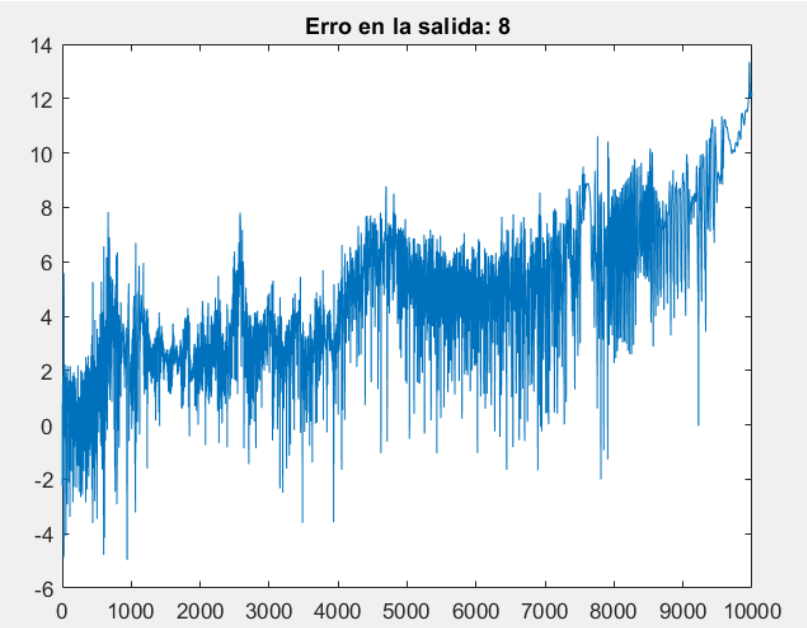

Ilustración 9: Color Morado

Análisis y conclusión:

En usa red neuronal de 3 entradas se complica un poco más poder analizar el clostring de los colores, además de necesitar muchas más muestras para el set de entrenamiento.

Para temas de análisis neuronal con datos censados es importante considerar el ambiente en donde se realiza el set de entrenamiento y la cantidad de muestras, ya que si se entre en un mismo ambiente la red neuronal solo servirá para esas mismas circunstancias de iluminación o ambiente. A su vez es muy importante también en este caso considerar la distancia a la que se censaba cada muestra.

Código [Anexo 1] ENTRENAMIENTO

```
%% MLP
clear all; clc
%% Entrenamiento
close all; clear all; clc
sigmoide = @(v) 1./(1+exp(-v)); %Para poder hacer calculos matriciales con ella
xd =[23 19 20 21 22 18 19 20 19 17 22 19 22 24 25 23 26 26 27 28 27 54 47 65 39 48 49
51 53 47 55 51 65 69 45 52 54 56 62 53 49 54 32 52 39 47 43 46 42 45 57 57 82  60 74
63 68 65 60 46 58 51 65 53 67 52 66 53 63 45 52 52 59 60 56 63 66 72 87 65 111 107 142
```

```
143 103 125 130 120 138 117 96  120 110 107 96  101 136 145 143 123 71  56  60  50  60
55  56  63  60  57  60  65  60  39  35  51  58  60  62  54  65  72  82  80  78  57  56
51  52  48  60  65  57  51  52  54  55  53  52  59 120 106 117 125 101 112 123 110 127
102 130 91  83  100 88  115 119 100 85  109;
     40 28 31 34 37 30 31 34 30 28 36 30 35 38 40 33 40 42 43 46 42 72 75 90 56 70 70
71 76 63 77 71 89 91 60 71 72 71 84 73 49 63 39 62 51 56 53 56 50 51 78 79 105 87 100
86 85 90 89 57 69 69 73 60 77 65 78 74 74 63 67 62 85 78 78 83 80 85 87 90 155 169 194
193 156 162 172 140 183 167 150 176 176 150 138 105 165 192 177 164 120 100 110 105
123 100 111 116 110 107 110 160 100 77  86  107 102 111 100 108 90  100 117 114 104 77
82  76  77  68  70  97  82  70  82  79  78  79  73  72 160 152 169 177 150 163 172 167
187 143 175 141 142 147 139 163 164 150 118 144;
     16 11 12 13 15 12 13 14 12 11 14 13 13 14 15 12 16 17 16 18 16 30 30 36 21 27 28
34 30 24 30 28 36 36 23 29 29 28 33 29 16 21 12 20 17 17 17 17 16 16 28 29 37  32  35
30 30 32 29 18 27 26 28 23 29 24 28 28 26 23 25 24 32 29 29 32 31 32 33 33 50  57  62
61  53  50  54  40  55  52  49  57  60  57  47  47  52  61  52  54  35  30  30  30  34
29  33  35  35  33  32  34  32  22  22  28  30  35  32  32  32  35  41  40  37  27  28
26  20  23  23  30  27  26  27  25  26  26  24  24 52  40  51  55  47  50  55  50  57
42  50  38  40  44  40  50  52  44  35  42];
d=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
[~,nK] = size(xd); %Datos de entrenaimiento
x1 = xd(1,:);
x2 = xd(2,:);
x3 = xd(3,:);
x1 = normalize(x1,nK);
x2 = normalize(x2,nK);
x3 = normalize(x3,nK);
xd = [x1; x2; x3];
[xd,D] = permutation(xd,d);
nP=3; %numero de entradas
nS=8; %numero de salidas
nO=7; %capa oculta
eta=.8; %aprendisaje
wO=rand(nP+1,nO); %Inicialisamos pesos aleatorios de capa oculta
wS=rand(nO+1,nS); %Inicialisamos pesos aleatorios de capa de salida
```

```
eVector=[];
for i=1:10000 %iteraciones del MLP
    eA=0;
    for k=1:nK  %para recopilacion de datos de entreniamiento
        %% Porpagacion hacia delante
        xO=[1; xd(:,k)]; % agregamos al dato k el bias
        vO=wO'*xO;      % se calcula la salida de las neuronas de la clapa oculta
        yO= tanh(vO);   % se calucla la funcion de activacion de las salida de la capa oculta
        xS=[1; yO];     % ahora las salida son la entrada de la capa de salida
        vS=wS'*xS;      % se calucla la salida de las neuronas  de la capa de salida
        yS=sigmoide(vS); % se calulca la funcion de activacion de la capa de salida
        e=d(:,k)-yS;    %Se calucla el error al deseado
        eA=eA+e;
        %% Propagacion hacia atras
        deltaS=e.*(yS.*(1-yS));
        deltaO=((ones(nO,1)-yO).*(ones(nO,1)+yO)).*(wS(2:end,:)*deltaS);
        %% Reecalculo de los pesos
        wS=wS+eta*(deltaS*xS')';
        wO=wO+eta*(deltaO*xO')';

    end
    eVector=[eVector eA];
end
% Plotear evoluvion el error por iteracion
for sal=1:nS
figure
plot(eVector(sal,:));
str = sprintf('Erro en la salida: %d', sal);
title(str);
end
%% Funciones
function [x] = normalize(x,n)
for k = 1:n
    x(k) = (x(k)-mean(x))/(max(x)-min(x));
end
end
function [x,D] = permutation(x,D)
[~,n] = size(x);
xrp = zeros(3,n);
Drp = zeros(8,n);
rp = randperm(n);
for k = 1:n
    xrp(:,k) = x(:,rp(k));
    Drp(:,k) = D(:,rp(k));
end
x = xrp;
D = Drp;
end

%keep wS wO;
```

Código [Anexo 2] SENSADO EN LINEA

```
%% GENERALIZACION
for asd=1:10000000
delete(instrfind({'Port'},{'COM7'}));
pserial=serial('COM7','BaudRate',9600);
fopen(pserial);
data = fscanf(pserial,'%d %d %d',[3,1]);
for asd = 1:length(data)
    if  data(asd)== '.'
        indP=asd;
    elseif data(asd)== '-'
        indDiag=asd;
    elseif data(asd)== 'Q'
        bre=asd;
    end
```

```
end
for hjk=1:length(data)
    if data(hjk)=='R'
       R=str2double(data(hjk+1:indP-1));
    elseif  data(hjk)=='G'
       G=str2double(data(hjk+1:indDiag-1));
    elseif  data(hjk)=='B'
       B=str2double(data(hjk+1:bre-1));
    end

end

            xg=[R;G;B]; %entrada
            x0=[1;xg]; %entrada con bias
            v0=wO'*x0; %applicacion
            y0=tanh(v0);
            xS=[1; y0]; % salida
            vS=wS'*xS;
            yS=sigmoide(vS)';
            [~,ic]=max(yS);
            if ic==1
                disp('BLANCO');
            elseif ic==2
                disp('           VERDE');
            elseif ic==3
                disp('                              AZUL');
            elseif ic==4
                disp('                                        CIAN');
            elseif ic==5
                disp('                                                  NEGRO');
            elseif ic==6
                disp('                                                         ROJO');
            elseif ic==7
                disp('
AMARILLO');
            elseif ic==8
                disp('
MORADO');
            end


end%NO TOCAR

fclose(pserial);
delete(pserial);
clear canal_serie;
```

Código [Anexo 2] PRUEBA

```
for i = 1:20:255
    for j=1 :20:255
        for k=1:20:255
xg=[i;j;k]; %entrada
            x0=[1;xg]; %entrada con bias
            v0=wO'*x0; %applicacion
            y0=tanh(v0);
            xS=[1; y0]; % salida
```

```
            vS=wS'*xS;
            yS=sigmoide(vS)';
            [~,ic]=max(yS);
            if ic==1
                disp('BLANCO');
            elseif ic==2
                disp('           VERDE');
            elseif ic==3
                disp('                              AZUL');
            elseif ic==4
                disp('                                        CIAN');
            elseif ic==5
                disp('                                                  NEGRO');
            elseif ic==6
                disp('                                                         ROJO');
            elseif ic==7
                disp('
AMARILLO');
            elseif ic==8
                disp('
MORADO');
            end
        end
    end
end
```

Código [Anexo 4]ArudinoIDE y Sensor

```
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8
int frequency = 0;
void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);

  // Setting frequency-scaling to 20%
  digitalWrite(S0,HIGH);
```

```
  digitalWrite(S1,LOW);


  Serial.begin(9600);
}
void loop() {
  // Setting red filtered photodiodes to be read
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  // Reading the output frequency
  frequency = pulseIn(sensorOut, LOW);
  // Printing the value on the serial monitor
  Serial.print("R");//printing name
  Serial.print(frequency);//printing RED color frequency
  Serial.print(".");
  delay(100);
  // Setting Green filtered photodiodes to be read
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  // Reading the output frequency
  frequency = pulseIn(sensorOut, LOW);
  // Printing the value on the serial monitor
  Serial.print("G");//printing name
  Serial.print(frequency);//printing RED color frequency
  Serial.print("-");
  delay(100);
  // Setting Blue filtered photodiodes to be read
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  // Reading the output frequency
  frequency = pulseIn(sensorOut, LOW);
  // Printing the value on the serial monitor
  Serial.print("B");//printing name
  Serial.print(frequency);//printing RED color frequency
```

Omar Baruch Morón López
Smart Systems

viernes, 27 de marzo de 2020
Robotics student of the University of Guadalajara

```
  Serial.println("Q");
  delay(100);
}
```

ANEXO SET DE ENTRENAMIENTO