Red ADALINE

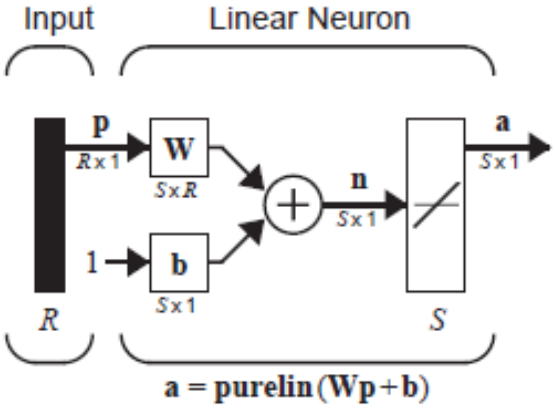
Alumno: *Osornio Gutiérrez Juan Damián*

Introducción.

La Red Adaline fue desarrollada en el 1960 por Bernard Widrow y su estudiante Marcian Hoff de la universidad de Stanford.

ADALINE proviene de Adaptive Lineal Element (Elemento Lineal Adaptativo), pero antes de que se le diera este nombre esta red sufrió un cambio ya que primeramente se llamaba Adaptive Lineal Neuron (Neurona Linear Adaptiva), dicho cambio se dio por que la Red Adaline es un dispositivo que consta de un único elemento de procesamiento, como tal no es técnicamente considerada una red neuronal.

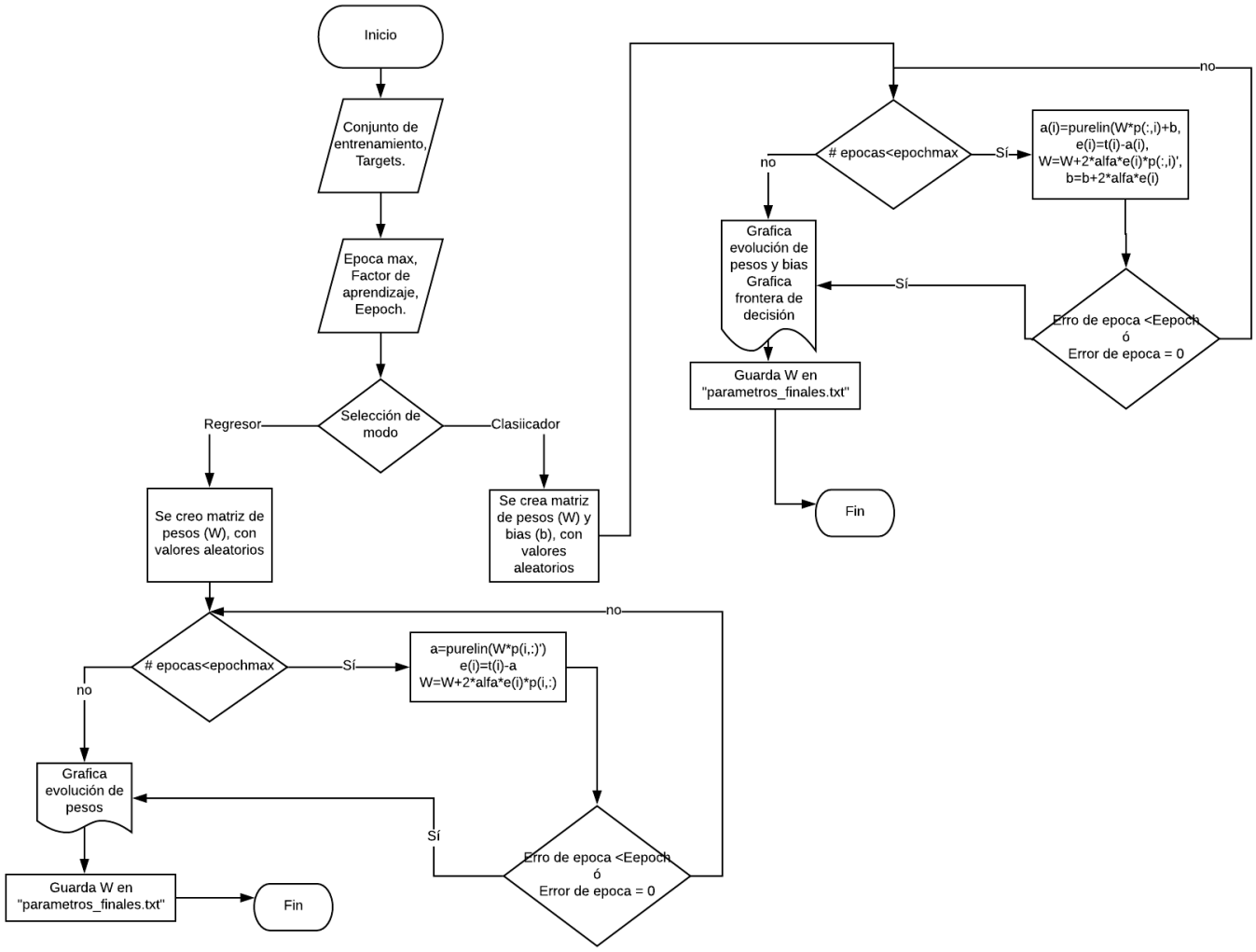
La red Adaline tiene una diferencia significativa con relación al Perceptrón ya que esta emplea una regla de aprendizaje más fuerte que la del perceptrón que es la llamada regla LMS (Least Mean Square) mínimo cuadrado medio. El perceptrón se basa en diferenciar de una clase de otra, separa por decirlo así manzana de naranjas, en cambio una red Adaline tiene una salida lineal y toma valores continuos que pueden ir de valores negativos hasta positivos.



Planteamiento del problema.

- El programa recibe un archivo *.txt con el dataset (sólo se puede usar este formato). El usuario elige entre el modo regresor o clasificador.
- El programa da el valor final de los parámetros de la red.
    - En ambos casos se debe presentar la gráfica de la evolución de los parámetros de la RNA.
    - En el modo clasificador debe dibujar la frontera de decisión.

Diagrama de flujo.



Resultado.

Para la compuerta AND.

Se reciben dos archivos .txt con el dataset, uno para los vectores de entrada y otro archivo para los targets.

Para los vectores de entrada:

P1	*P2*	*P3*	*P4*
0	*0*	*1*	*1*
0	*1*	*0*	*1*

Para los targets:

t1	*t2*	*t3*	*t4*
0	*0*	*0*	*1*

El programa pide los siguientes datos:

Ingrese el número máximo de épocas a realizar (epochmax):

50

Ingrese el valor al cuál se desea que llegue la señal de error (eepoch):

.01

Ingrese el factor de aprendizaje:

.3

Nos arroja los siguientes valores y la gráfica:

Los errores de las épocas:

2.5310

0.9261

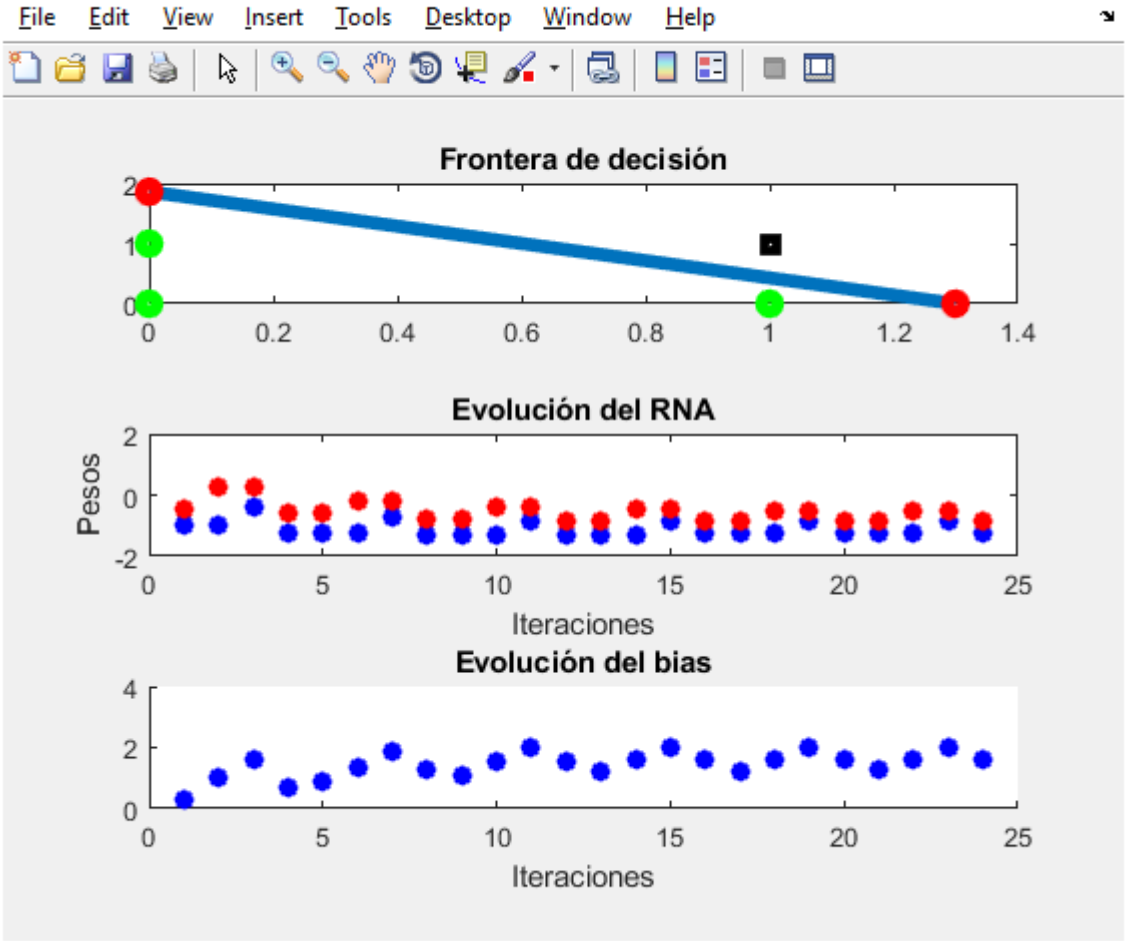
0.4020

0.1478

0.0440

0.0068

Se realizaron 6 épocas



Para la compuerta OR.

Para los vectores de entrada:

P1	*P2*	*P3*	*P4*
0	*0*	*1*	*1*
0	*1*	*0*	*1*

Para los targets:

t1	*t2*	*t3*	*t4*
0	*0*	*0*	*1*

Nos arroja los siguientes valores y la gráfica:

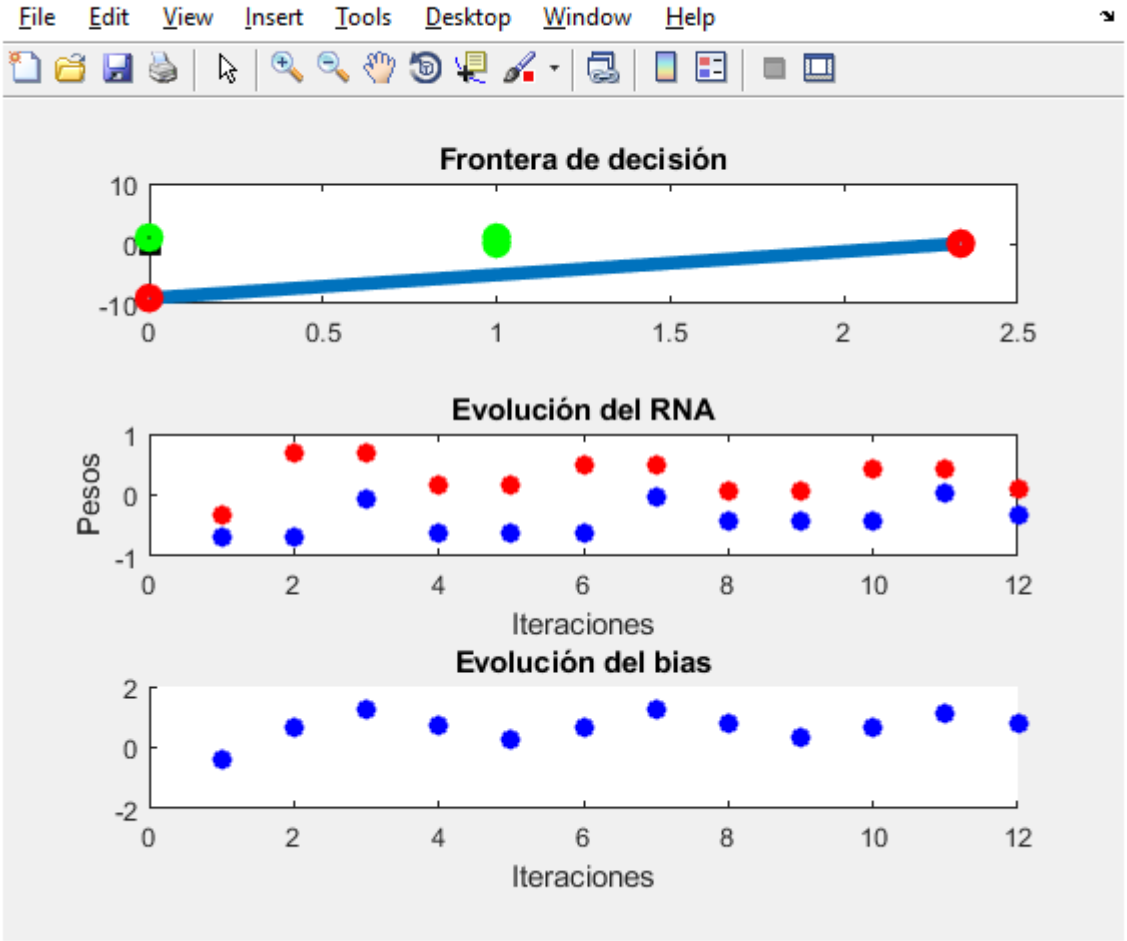
Se realizaron 3 épocas

Los errores de las épocas:

2.7915

0.1319

-0.0434



Otro ejemplo.

Volvemos a correr el programa, pero ahora seleccionamos el modo regresor, con el siguiente dataset:

t	*P1*	*P2*	*P3*
0	*0*	*0*	*0*
1	*0*	*0*	*1*
2	*0*	*1*	*0*
3	*0*	*1*	*1*
4	*1*	*0*	*0*
5	*1*	*0*	*1*
6	*1*	*1*	*0*
7	*1*	*1*	*1*

Nos arroja los siguientes valores:

Los errores de las epocas:

1.0643

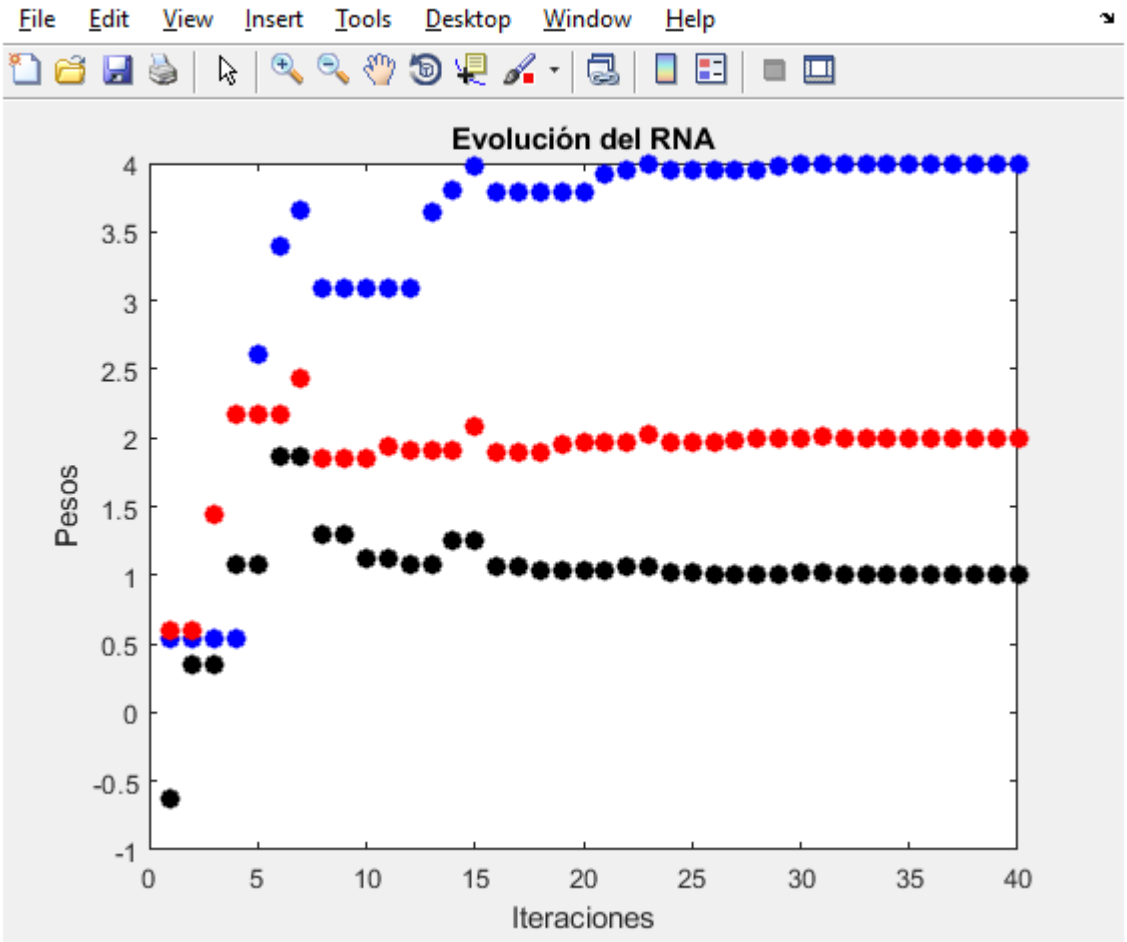
0.1204

0.0394

0.0104

0.0026

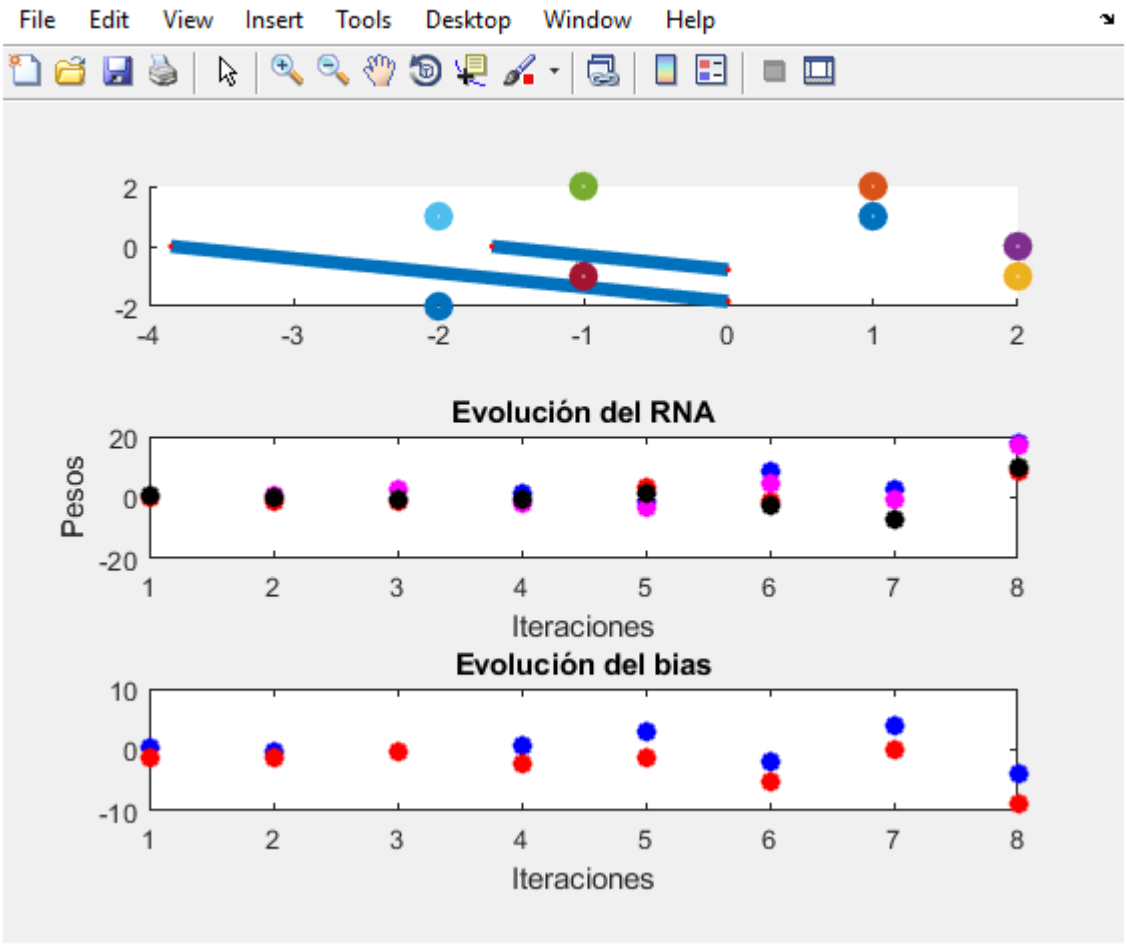
Se realizaron 5 épocas



Otro ejemplo de regla de aprendizaje, ahora se ingresarán las siguientes matrices:

P1	*P2*	*P3*	*P4*	*P5*	*P6*	*P7*	*P8*
1	*1*	*2*	*2*	*-1*	*-2*	*-1*	*-2*
1	*2*	*-1*	*0*	*2*	*1*	*-1*	*-2*

t1	*t2*	*t3*	*t4*	*t5*	*t6*	*t7*	*t8*
0	*0*	*0*	*0*	*1*	*1*	*1*	*1*
0	*0*	*1*	*1*	*0*	*0*	*1*	*1*



Y en el archivo “parámetro_finales.txt” se guardó el siguiente valor:

Parámetros finales W= [17.8719 8.2152] [17.2288 10.1493]

Discusión de resultados.

Como se puede observar en los ejemplos el programa funciona correctamente, aunque en el modo clasificador falla un poco a la hora de seleccionar la frontera de decisión, por ejemplo, en el caso de un operador lógico, AND, si los valores iniciales de los pesos y bias son negativos, casi siempre separa las clases de manera correcta. El caso contrario es con la OR, si se le dan valores iniciales negativos, no separa las clases.

Conclusiones.

La red Adaline se basan en el objetivo de evaluar de la forma más correcta la salida, para de esta manera poder minimizar el error medio cuadrático, por tanto, son muy similares al perceptron, lo único en lo que cambian es la transferencia en la salida.

Adaline tiene una considerable diferencia con respecto a perceptron, puesto que en la modificación de los pesos que se dan en la preparación, la Red Adaline tiene muy presente el valor de corrección de la salida estimada con relación a la esperada.

Código:

```
clear;
clc;

p = double(dataset('File', 'entrada2.txt'));
t = double(dataset('File' , 'targets2.txt'));
[x,y] = size(p);
[x2,y2] = size(t);

fprintf('¿Qué modo desea, regresión o clasificación?\n');
fprintf('1.Regresor \n2.Clasificador\n')
D=input('');
clc;
    fprintf('Ingrese el número máximo de épocas a realizar
(epochmax):\n');
    emax = input('');
    fprintf('Ingrese el valor al cuál se desea que llegue la señal de
error (eepoch):\n');
    epoch = input('');
    fprintf('Ingrese el factor de aprendizaje:\n');
    alfa = input('');

if D==1
    W=2*rand(1,y)-1;
    j=1;

    for Epoca=1:emax
        sumaerror=0;
       for i=1:x
           a=purelin(W*p(i,:)');
           e(i)=t(i)-a;
           W=W+2*alfa*e(i)*p(i,:);
```

```
      plot(j,W(1),'b*','LineWidth',5)
      title('Evolución del RNA');
      ylabel('Pesos');
      xlabel('Iteraciones');
      hold on
      plot(j,W(2),'r*','LineWidth',5)
      hold on
      plot(j,W(3),'k*','LineWidth',5)


      j=j+1;
        end
        sumaerror=sum(e,2);
        Error(Epoca)=sumaerror/x;

  if (Error(Epoca)<epoch || Error(Epoca)==0)
      break;
  end

    end

fid = fopen('parametros_finales.txt','w');
fprintf(fid, 'Parametros finales\n W=[ ');
fprintf(fid, '\n%.4f  \n', W.');
fprintf(fid,']');
fclose(fid);

end


    clases=0;
 for i2 = 1:y2
        if i2+1>y2
            break;
         end
     if t(:,i2)==t(:,i2+1)
         clases=clases+1;
         if i2>y2
             break;
         end
     end
 end
 S=log2(clases);


if D==2
   j=1;

   if x2==1
   W=-rand(1,x);
   b=-rand(1);

for Epoca= 1:emax
```

```
sumaerror=0;
    for i=1:y
        a(i)=purelin(W*p(:,i)+b);
        e(i)=t(i)-a(i);
        W=W+2*alfa*e(i)*p(:,i)';
        b=b+2*alfa*e(i);


    subplot(3,1,2)
     plot(j,W(1),'b*','LineWidth',5)
     title('Evolución del RNA');
     ylabel('Pesos');
     xlabel('Iteraciones');
     hold on
     plot(j,W(2),'r*','LineWidth',5)

     subplot(3,1,3)
     hold on
     plot(j,b,'b*','LineWidth',5)
     title('Evolución del bias');
     xlabel('Iteraciones');

     j=j+1;
    end
         sumaerror=sum(e);
         Error(Epoca)=sumaerror/S;

  if (Error(Epoca)<epoch || Error(Epoca)==0)
      break;
  end

end

for r = 1:y
    if t(r)==1
        subplot(3,1,1)
        plot(p(1,r),p(2,r),'go','LineWidth',5)
        title('Frontera de decisión');
        hold on
    end
    if t(r)==0
        subplot(3,1,1)
         plot(p(1,r),p(2,r),'ks','LineWidth',5)
         hold on
    end
end

P1=[0 -b/W(1)];
P2=[-b/W(2) 0];
subplot(3,1,1)
line(P1,P2, 'LineWidth',5)
hold on
plot(P1,P2,'ro','LineWidth',5)

   end
```

```
if x2~=1
    W=2*rand(x)-1;
    b=-rand(x,1);

for Epoca= 1:emax
    for i=1:y
        a(:,i)=purelin(W*p(:,i)+b);
        e(:,i)=t(:,i)-a(:,i);

        W=W+2*alfa*e(:,i)*p(:,i)';
        b=b+2*alfa*e(:,i);

     subplot(3,1,2)
     plot(j,W(1,1),'b*','LineWidth',5)
     title('Evolución del RNA');
     ylabel('Pesos');
     xlabel('Iteraciones');
     hold on
     plot(j,W(1,2),'r*','LineWidth',5)
     hold on
     plot(j,W(2,1),'m*','LineWidth',5)
     hold on
     plot(j,W(2,2),'k*','LineWidth',5)


     subplot(3,1,3)
     plot(j,b(1,1),'b*','LineWidth',5)
     title('Evolución del bias');
     xlabel('Iteraciones');
     hold on
     plot(j,b(2,1),'r*','LineWidth',5)
     j=j+1;

    end
sumaerror=sum(sum(e),2);

Error(Epoca)=sumaerror/S;
if (Error(Epoca)<epoch || Error(Epoca)==0)
      break;
end

end



W2=inv(W);

for linea=1:S
    P1=[0 (-b(linea)*W2(2,:))];
    P2=[(-b(linea)*W2(1,:)) 0];

    subplot(3,1,1)
```

```
    line(P1(1:2),P2(2:3), 'LineWidth',5)
    hold on
    plot(P1(1:2),P2(2:3),'r.','LineWidth',5)
end

for punto=1:y
    subplot(3,1,1)
    plot(p(1,punto),p(2,punto),'o','LineWidth',5)
    hold on
end
end

fid = fopen('parametros_finales.txt','w');
fprintf(fid, 'Parametros finales\n W= ');
fprintf(fid, '\n[%.4f  %.4f] \n', W.');
fclose(fid);


end

fprintf('Los errores de las epocas:\n');
fprintf('%.4f\n',Error(:));
fprintf('Se realizaron %d épocas\n',Epoca);
```

Referencias.

Neural Network Design. [Martin T. Hagan, Howard B, Demuth, Mark Beale – PWS Publishing Company]

Redes neuronales. Algoritmos, aplicaciones y tecnicas de programación, James A. Freeman, David M. Skapura, ADDISON-WESLEY, 1993

http://www.varpa.org/~mgpenedo/cursos/scx/archivospdf/Tema3-0.pdf