



Universidad
de Alcalá

SISTEMAS DE CONTROL INTELIGENTE

Práctica 1.
Identificación y control neuronal

Jorge Revenga Martín de Vidales
Ángel Salgado Aldao

Grado en Ingeniería Informática
Universidad de Alcalá

30 de octubre de 2023

Índice

I	2
1. Ejercicio 1. Perceptrón.	2
1.1. Código	2
1.2. Preguntas	2
1.3. Ejecución	3
2. Ejercicio 2. Aproximación de funciones.	4
2.1. Solución	4
2.1.1. Ejecución	4

Parte I

1. Ejercicio 1. Perceptrón.

Se desea clasificar un conjunto de datos pertenecientes a cuatro clases diferentes. Los datos y las clases a las que pertenecen con los que se muestra a continuación:

x_1	x_0	Clase
0.1	1.2	2
0.7	1.8	2
0.8	1.6	2
0.8	0.6	0
1.0	0.8	0
0.3	0.5	3
0.0	0.2	3
-0.3	0.8	3
-0.5	-1.5	1
-1.5	-1.3	1

Se desea diseñar un clasificador neuronal mediante un perceptrón simple que clasifique estos datos. Diseñe el clasificador, visualice los parámetros de la red y dibuje los datos junto con las superficies que los separan.

1.1. Código

```

1  P = [0.1 0.7 0.8 0.8 1.0 0.3 0 -0.3 -0.5 -1.5; 1.2 1.8 1.6 0.6 0.8 0.5 0.2 0.8 -1.5 -1.3];
2  T = [1 1 1 0 0 1 1 1 0 0; 0 0 0 0 0 1 1 1 1 1];
3
4  net = perceptron;
5  net = train(net, P, T);
6  plotpv(P, T);
7  plotpc(net.iw{1,1}, net.b{1});

```

1.2. Preguntas

¿Consigue la red separar los datos? Sí, los clasifica en las 4 clases correctamente

¿Cuántas neuronas tiene la capa de salida?, ¿por qué? Tiene 4 neuronas, cada una genera una salida que representa la probabilidad de pertenencia a una clase particular, como tenemos 4 clases, tenemos 4 neuronas

¿Qué ocurre si se incorpora al conjunto un nuevo dato: [0.0 -1.5] de la clase 3? Las líneas que dividen los datos en las clases dejan de aplicarse a todos los datos, su pendiente ha cambiado pero el nuevo dato no está separado en el mismo espacio que el resto de datos de clase 3. Está con los datos de clase 1

1.3. Ejecución

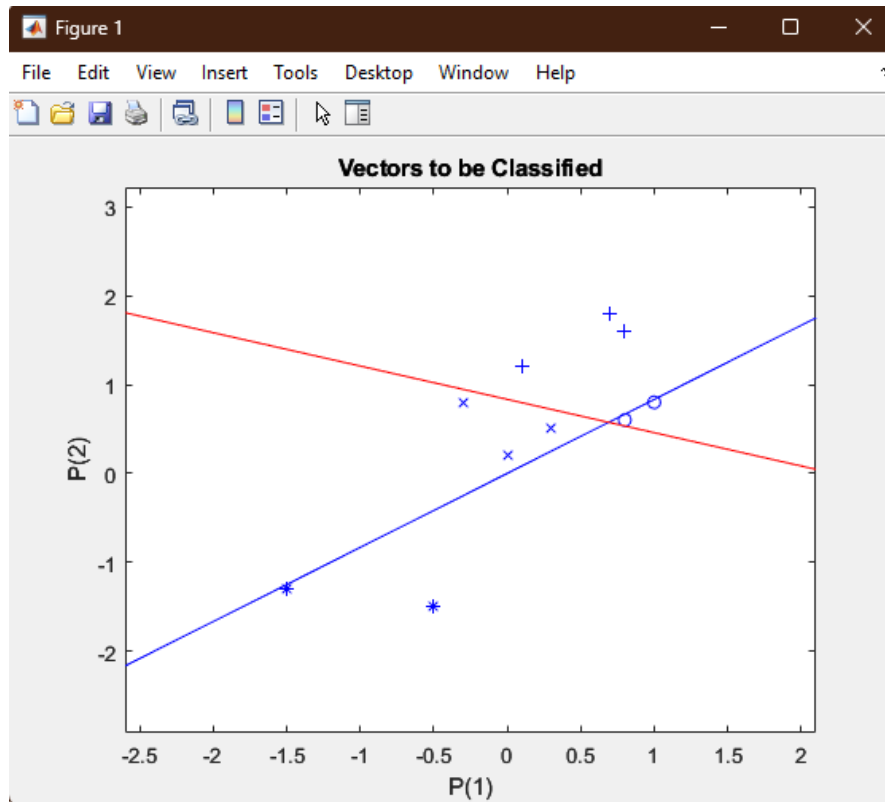


Figura 1: Ejecución ejercicio 1.

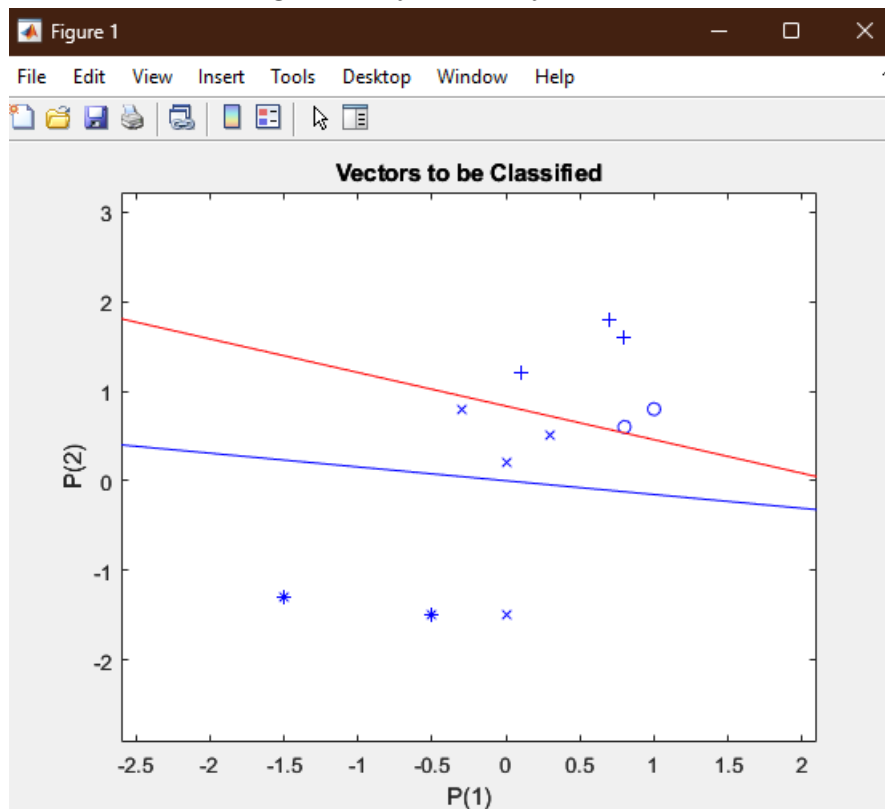


Figura 2: Ejecución ejercicio 1 con el nuevo dato.

2. Ejercicio 2. Aproximación de funciones.

Una de las aplicaciones inmediatas de las redes neuronales es la aproximación de funciones. Para ello, Matlab dispone de una red optimizada, `fitnet`, con la que se trabajará en este ejercicio. El objetivo en este caso es aproximar la función $f = \text{sinc}(t)$ tal y como se muestra a continuación:

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % APROXIMACIÓN DE FUNCIONES
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4  clear all; close all;
5  % DEFINICIÓN DE LOS VECTORES DE ENTRADA-SALIDA
6  % =====
7  t = -3:.1:3; % eje de tiempo
8  F=sinc(t)+.001*randn(size(t)); % función que se desea aproximar
9  plot(t,F,'+');
10 title('Vectores de entrenamiento');
11 xlabel('Vector de entrada P');
12 ylabel('Vector Target T');
13 % DISEÑO DE LA RED
14 % =====
15 hiddenLayerSize = 4;
16 net = fitnet(hiddenLayerSize,'trainrp');
17 net.divideParam.trainRatio = 70/100;
18 net.divideParam.valRatio = 15/100;
19 net.divideParam.testRatio = 15/100;
20 net = train(net,t,F);
21 Y=net(t);
22 plot(t,F,'+'); hold on;
23 plot(t,Y,'-r'); hold off;
24 title('Vectores de entrenamiento');
25 xlabel('Vector de entrada P');
26 ylabel('Vector Target T');

```

Estudie los efectos sobre la solución final de modificar el método de entrenamiento (consulte la ayuda de Matlab y pruebe 4 métodos diferentes) y el número de neuronas de la capa oculta.

2.1. Solución

Para poder visualizar el código en funcionamiento primero debemos definir la función $\text{sinc}(t)$. Lo haremos en un archivo llamado `sinc.m`. En nuestro caso hemos definido la siguiente función $\text{sinc}(t)$:

```

1  function y = sinc(t)
2      y = sin(pi * t) ./ (pi * t);
3  end

```

2.1.1. Ejecución

Para modificar el método de entrenamiento basta con cambiar el segundo parámetro de la función `fitnet()` por otra función de entrenamiento. Vamos a estudiar las funciones `'traingd'` (descenso de gradiente), `'traingdm'` (descenso de gradiente con momento), `'trainlm'` (Levenberg-Marquardt), `'trainbr'` (Broyden-Fletcher-Goldfarb-Shanno).

