

Práctica 1. Identificación y control neuronal (I)

Sistemas de Control Inteligente (GIC, GII, GSI)

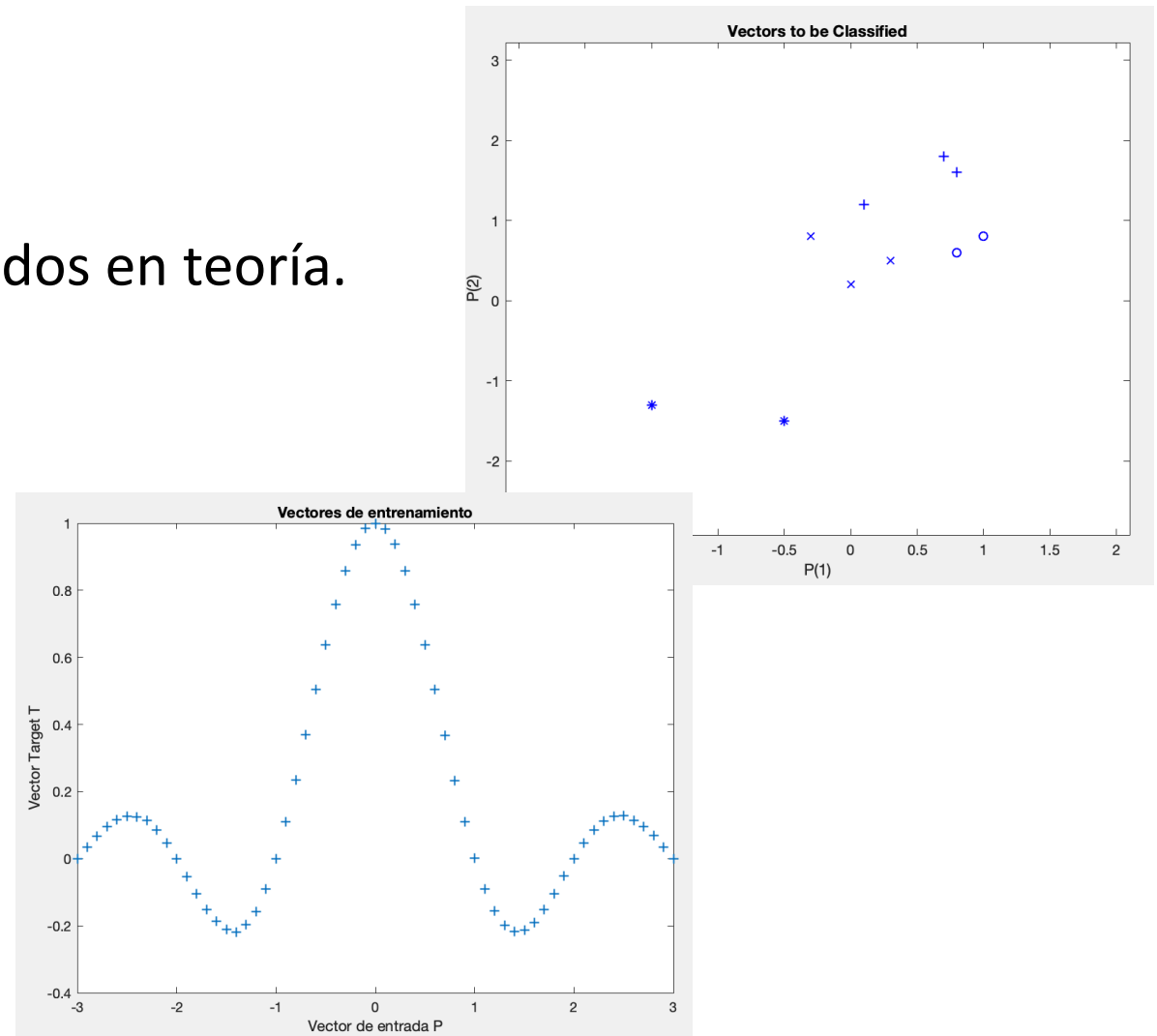
Dpto. de Electrónica.

Universidad de Alcalá

José Luis Martín, Daniel Pizarro, Luis Miguel Bergasa, Ángel Llamazares, David Gualda.

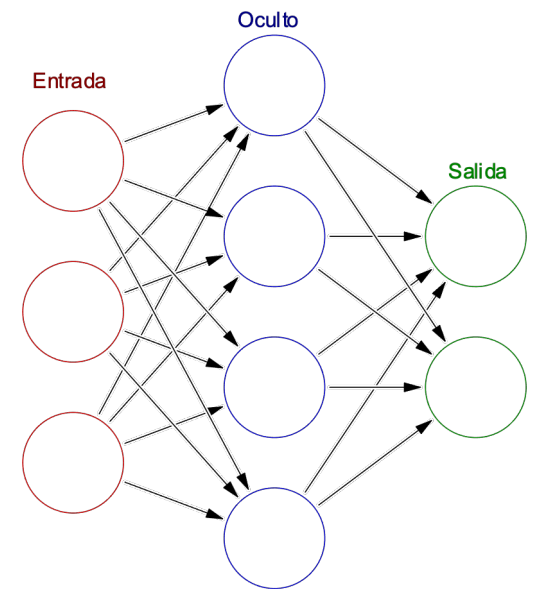
Objetivo

- Revisar los conceptos explicados en teoría.
- Aplicar las RRNN a tareas de:
 - Clasificación
 - Aproximación de funciones



Algunos conceptos previos

- ¿Qué es una red neuronal artificial?
- Conjunto de datos: entrenamiento / validación / test
- “Tipos” de redes (perceptron, fitnet, patternnet)
- Métodos de entrenamiento (trainlm, trainbr, trainbfg, trainrp, ...)



De en:User:Cburnett - File:Colored neural network uk.svg, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=78778791>

Deep Learning Toolbox / Neural Network Toolbox

- `net = fitnet(hiddenLayerSize,'trainrp');`
- `net.divideParam.trainRatio = 70/100;`
- `net.divideParam.valRatio = 15/100;`
- `net.divideParam.testRatio = 15/100;`
- `net = train(net,t,F);`
- `train(net,inputs,targets);`

Deep Learning Toolbox / Neural Network Toolbox

<code>simplefit_dataset</code>	- Simple fitting dataset.
<code>abalone_dataset</code>	- Abalone shell rings dataset.
<code>bodyfat_dataset</code>	- Body fat percentage dataset.
<code>building_dataset</code>	- Building energy dataset.
<code>chemical_dataset</code>	- Chemical sensor dataset.
<code>cho_dataset</code>	- Cholesterol dataset.
<code>engine_dataset</code>	- Engine behavior dataset.
<code>vinyl_dataset</code>	- Vinyl bromide dataset.

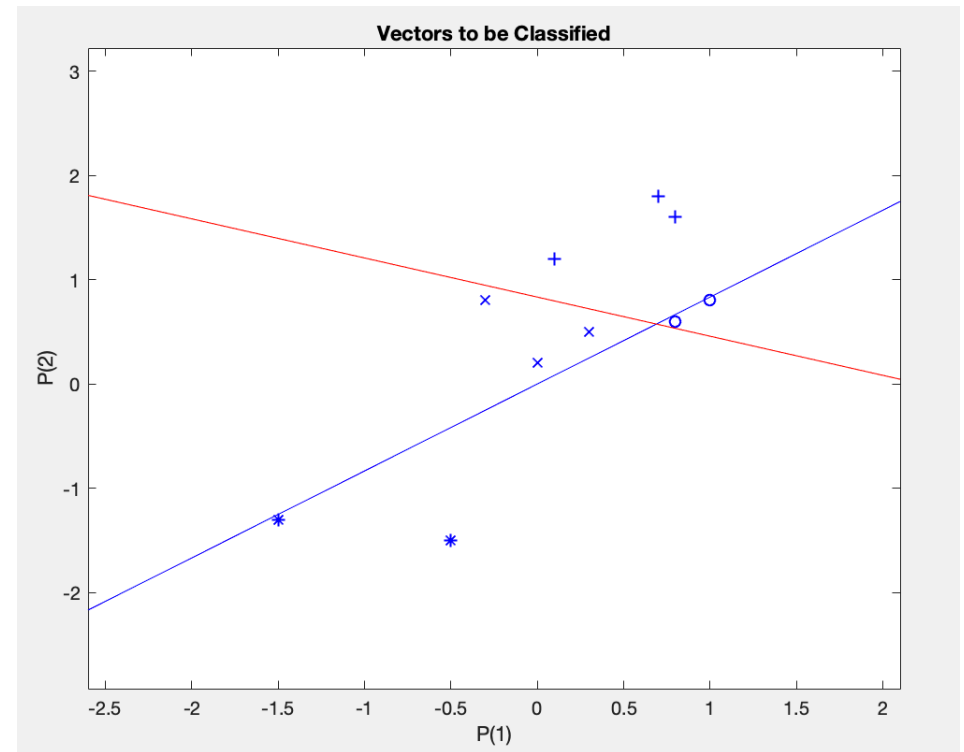
<code>simpleclass_dataset</code>	- Simple pattern recognition dataset.
<code>cancer_dataset</code>	- Breast cancer dataset.
<code>crab_dataset</code>	- Crab gender dataset.
<code>glass_dataset</code>	- Glass chemical dataset.
<code>iris_dataset</code>	- Iris flower dataset.
<code>ovarian_dataset</code>	- Ovarian cancer dataset.
<code>thyroid_dataset</code>	- Thyroid function dataset.
<code>wine_dataset</code>	- Italian wines dataset.

Ejemplo inicial: perceptron

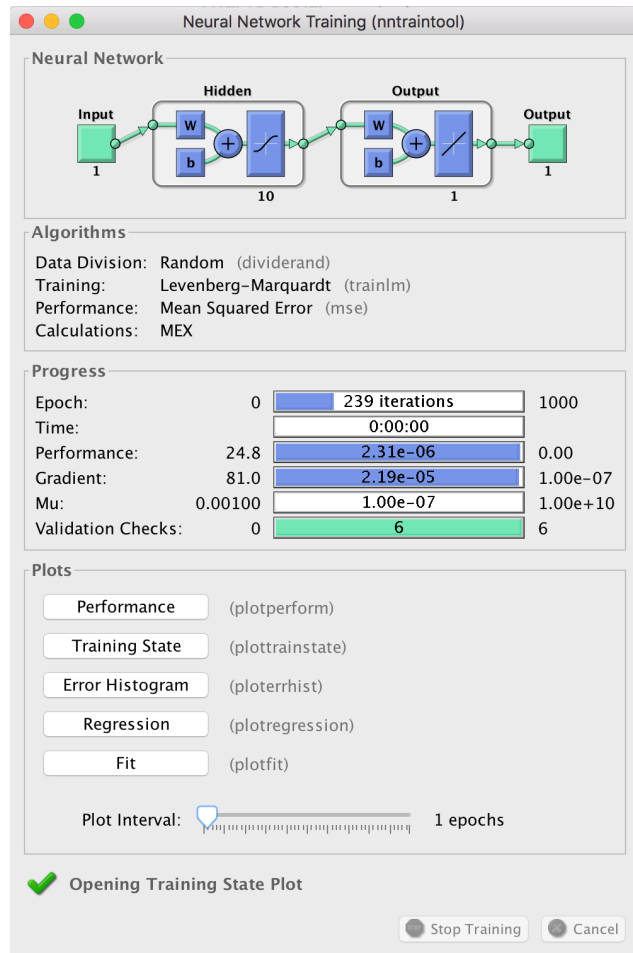
```
P=[0.1 0.7 0.8 0.8 1.0 0.3 0.0 -0.3 -0.5 -1.5;  
    1.2 1.8 1.6 0.6 0.8 0.5 0.2 0.8 -1.5 -1.3];  
T=[1 1 1 0 0 1 1 1 0 0;  
    0 0 0 0 1 1 1 1 1 1];
```

```
net = perceptron;  
net = train(net,P,T);
```

```
plotpv(P,T);  
plotpc(net.iw{1,1},net.b{1});
```



Aproximación de funciones y clasificación



← Esquema de la red

← Detalles de los algoritmos

← Progreso del proceso de
entrenamiento y test

← Gráficas con diversos
resultados

