

```
sigmoid = @(x) 1./(1 + exp(-x));
sigmoid_derivative = @(x) sigmoid(x) .* (1 - sigmoid(x));

softmax = @(x) exp(x) ./ sum(exp(x))

softmax = function_handle with value:
    @(x)exp(x)./sum(exp(x))
```

Definimos los datos de entrada

```
X1 = [0.5; 0.8; 0.2; 0.3];
X2 = [0.6; 0.9; 0.2; 0.1];
X3 = [0.8; 0.1; 0.2; 0.3];

y1 = [1; 0; 0];
y2 = [0; 1; 0];
y3 = [0; 0; 1];
```

A continuación, definimos la red neuronal

```
% Crear una instancia de la red neuronal
addpath('neuralnet/matlab_net');
net = NeuralNetwork();

% Definir capa 1 con 2 entradas, 3 salidas, función de activación sigmoideal y su
derivada
layer1 = SequentialLayer(4, 10, sigmoid, sigmoid_derivative);
net = net.set(layer1);

% Definir capa 2 con 3 entradas (salidas de la capa anterior), 1 salida, función de
activación lineal y su derivada
layer2 = SequentialLayer(10, 3, softmax, false);
net = net.set(layer2);
```

Comprobamos la salida para los datos dados, para posteriormente comprobar que se está entrenando correctamente

```
% Realizar propagación hacia adelante
output = net.feedforward(X1);
disp('Salida después de la propagación hacia adelante X1:');
```

Salida después de la propagación hacia adelante X1:

```
disp(output);
```

```
0.0135
0.3417
0.6449
```

```
output = net.feedforward(X2);
disp('Salida después de la propagación hacia adelante X2:');
```

Salida después de la propagación hacia adelante X2:

```
disp(output);
```

```
0.0161
0.2737
0.7102
```

```
output = net.feedforward(X3);
disp('Salida después de la propagación hacia adelante X3:');
```

Salida después de la propagación hacia adelante X3:

```
disp(output);
```

```
0.0060
0.6463
0.3478
```

Entrenamos la red neuronal

```
for iter = 1:10000
    % Realizar retropropagación
    net = net.backpropagation(X1, y1, 0.1);
    net = net.backpropagation(X2, y2, 0.1);
    net = net.backpropagation(X3, y3, 0.1);
end
```

Comprobamos que se ha entrenado

```
output = net.feedforward(X1);
disp('Salida después de la propagación hacia atrás X1:');
```

Salida después de la propagación hacia atrás X1:

```
disp(output);
```

```
0.9981
0.0014
0.0006
```

```
output = net.feedforward(X2);
```

```
disp('Salida después de la propagación hacia atrás X2:');
```

Salida después de la propagación hacia atrás X2:

```
disp(output);
```

```
0.0016  
0.9984  
0.0000
```

```
output = net.feedforward(X3);  
disp('Salida después de la propagación hacia atrás X3:');
```

Salida después de la propagación hacia atrás X3:

```
disp(output);
```

```
0.0005  
0.0000  
0.9995
```