

Profesor responsable: Patricio García Báez

Fecha tope de corrección: 3 de noviembre

Objetivo: Dominio del SNNS y capacidad representación del conocimiento en las RN monocapas

Utilizando el neurosimulador SNNS se pretende diseñar y entrenar varias redes neuronales feed-forward capaces de aprender a **transformar valores** numéricos entre **diferentes tipos de codificaciones** y la codificación utilizada para las **etiquetas de códigos de barras**, tan comunes en los productos de consumo. Para ello habrán de diseñarse los ficheros correspondientes de patrones de entrenamiento así como los que almacenen las diferentes redes en sí.



Figura 1: Código de barras

Utilizaremos los **números del 0 al 9**, los cuales los codificaremos de cuatro diferentes maneras:

1. Utilizando **una entrada o neurona de salida** asociada a **cada uno de estos números** (un total de **10 entradas o salidas**), en dicha codificación ha de estar activo un único elemento, que indica el número a representar.
2. En **modo binario**, esto es, mediante **cuatro elementos**, representando respectivamente las diferentes potencias de dos de la codificación binaria del número a representar. El elemento que este activo indica un uno en la codificación binaria y el inactivo un cero.
3. En **modo decimal**, con **un solo elemento**, de forma que la magnitud de éste indique el valor que se quiere representar, una neurona por tanto.
4. En el formato utilizado por **las etiquetas de códigos de barras** EAN ó UPC (<http://www.milk.com/barcode/>), representando cada número por dos franjas con distinto grosor y espaciado, donde dichas franjas se codifican en **siete elementos** correspondientes a los valores (negro/blanco = 1/0 ó 0/1) indicados en la figura 2.

#	Codificación
0	0001101
1	0011001
2	0010011
3	0111101
4	0100011
5	0110001
6	0101111
7	0111011
8	0110111
9	0001011

Figura 2: Código de cada dígito en etiquetas de barras

Se pretende por tanto generar y entrenar **siete tipos de codificadores diferentes**:

1. Conversor de código 1 al 4
2. Conversor de código 4 al 1
3. Conversor de código 2 al 4
4. Conversor de código 4 al 2
5. Conversor de código 3 al 4
6. Conversor de código 4 al 3
7. Conversor de código 4 al 4 (con una o más neuronas ocultas)

Ha de tratarse de generar arquitecturas neuronales que utilicen el número mínimo posible de neuronas que resuelvan bien los problemas, y ratios de aprendizaje que la hagan converger en el número mínimo de ciclos posibles.

Con los resultados de los entrenamientos anteriores se confeccionará una **tabla de resultados** en la que se indique **tipo de patrón, mejor arquitectura de RN** encontrada, **mejor error** obtenido (tanto su valor **SSE** como en **porcentaje de aciertos**), valor del **ratio de aprendizaje óptimo** y **número de ciclos** empleado para el aprendizaje, así como las **observaciones** que se quieran hacer constar. Dicha tabla será la que finalmente se entregue como justificación de la práctica.