

# PRÁCTICA 3 DE NEUROCOMPUTACIÓN

## Curso 2013-2014

### Otras aplicaciones de las Redes Neuronales Artificiales

1 de Abril de 2014. Entrega: 30 de Abril

## PARTE 1

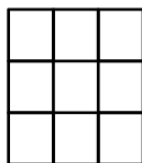
### 1. Autoencoders

Un autoencoder es una red cuya salida es igual a la entrada.

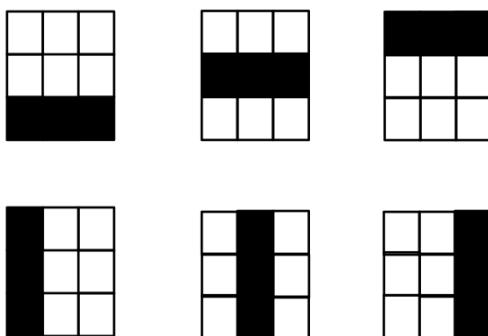
Sus aplicaciones más importantes son la eliminación de ruido y distorsiones en la entrada, y la compresión de datos.

Para comprimir los datos, la red debe capturar estructura interna del problema. Esto es lo que probaremos en la primera parte de la práctica.

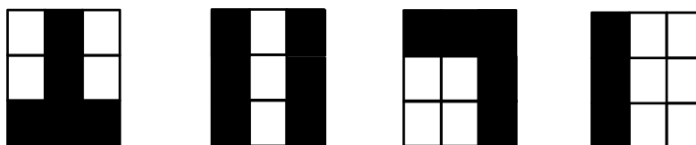
Consideremos imágenes en una rejilla de 3x3 píxeles:



Todas las posibles imágenes que va a mostrar dicha rejilla estarán construidas a partir de barras horizontales o verticales:



De esta forma, los siguientes 4 ejemplos son imágenes posibles en nuestra rejilla:



## 1.1 Creación del fichero de datos.

Implementa la siguiente función:

**construye\_bd\_autoencoder(int n, char \*nombre\_fichero)**

n es el tamaño de nuestra rejilla (en el ejemplo anterior, n=3).

nombre\_fichero es el nombre del fichero .txt que se creará en el mismo formato de las prácticas anteriores.

Primera línea de dicho fichero:

**n^2 n^2**

ya que con un ancho de rejilla=n habrá  $n^2$  píxeles, y en un problema de autoencoder el número de entradas es igual al número de salidas. Por ejemplo, si n=3 la primera línea del fichero será 9 9.

Siguientes líneas: habrá tantas como patrones. Cada patrón tendrá  $2n^2$  números: los primeros  $n^2$  correspondientes a las entradas, y los siguientes  $n^2$  a las salidas.

- El primer patrón corresponde a la imagen vacía. En el caso de n=3, será:

**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

- Los primeros n patrones corresponderán a las n barras horizontales

Por ejemplo, para la barra horizontal inferior y n=3, la codificación del patrón correspondiente será:

**0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1**

En negrita se representan las 9 entradas, y en letra normal la información correspondiente a las 9 salidas.

- Los siguientes n patrones corresponderán a las n barras verticales

Por ejemplo, en el caso de la barra vertical situada en el medio y n=3, el patrón correspondiente será:

**0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0**

- Los siguientes patrones corresponden a combinar uno cualquiera de esos patrones con otro diferente. Habrá entonces  $2n \cdot (2n-1)/2 = n \cdot (2n-1)$  patrones resultado de la combinación de dos barras.

Por ejemplo, en el caso de la imagen siguiente:



Su codificación en el fichero será:

**1 1 1 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1**

En total, en el fichero habrá entonces  $1 + n + n + n \cdot (2n-1) = 1 + n \cdot (2n+1)$  patrones. Para  $n=3$  habrá entonces 22 patrones.

Recuerda que tu implementación de **construye\_bd\_autoencoder** debe servir para cualquier  $n > 1$ .

## 1.2 Experimentos con el autoencoder.

**$n=3$ .**

Entrena una red neuronal del tipo backpropagation con este problema. Haz entrenamiento = test = toda la base de datos.

Internamente, codifica los 1 de la base de datos como +0.9 y los 0 de la base de datos como -0.9, como hacíamos en la práctica anterior.

En vez de error de clasificación, vamos a considerar el error de reconstrucción, es decir, número de píxeles de salida que la red reconstruye mal. Consideramos que una neurona de salida (que tiene función de activación bipolar) predice que el píxel está activado si su actividad es  $>0$ , y predice que el píxel no está activado en caso contrario.

Prueba con 1 neurona oculta, 2, 3 ... ¿A partir de qué número de neuronas la red aprende perfectamente el problema? (no comete ningún error en ningún píxel de ningún patrón).

Para ver si la red está aprendiendo, da gráficas del ECM en cada época de entrenamiento. Y del error de reconstrucción (% de píxeles mal reconstruidos) en función de la época de aprendizaje.

### **Codificación encontrada por las neuronas ocultas.**

Cuando hayas encontrado cuál es el número de neuronas ocultas necesarias para resolver el problema, intenta entender qué información están capturando, qué compresión ha encontrado la red neuronal.

Para ello puedes hacer varias cosas:

- Analizar la actividad de cada neurona oculta en función del patrón de entrada.
- Analizar los pesos entre la capa de entrada y cada neurona oculta: imprimiéndolos, dibujándolos, etc.

**Haz experimentos con  $n=5$ . ¿Sigues observando lo mismo?**