



UNIVERSIDAD DE BURGOS

COMPUTACIÓN NEURONAL Y EVOLUTIVA

P2: Self-organizing Maps

Diseñar y entrenar un mapa auto-organizado con una rejilla bidimensional para el agrupamiento de un único conjunto de datos de muestra de tres dimensiones.

Estudiando el efecto que tienen una serie de parámetros durante el entrenamiento de este.

Estudiantes:

DAVID MIGUEL LOZANO
JAVIER MARTÍNEZ RIBERAS

Profesor de la asignatura:

ÁLVARO HERRERO COSÍO

1º semestre 2016

Índice

A. Introduction	2
B. Descripción del conjunto de datos	2
C. Descripción del procedimiento	3
D. Estudio	3
E. Conclusiones	4

A. Introduction

El objetivo de la práctica es diseñar y entrenar un mapa auto-organizado para una rejilla bidimensional con un único conjunto de datos de muestra de tres dimensiones.

Se realizará un estudio sobre el impacto que supone variar los siguientes parámetros en el entrenamiento del mapa para el mismo conjunto de datos:

1. Dimensiones de la topología.
2. Tamaño del vecindario.
3. Función de topología.
4. Función de distancia entre neuronas.

B. Descripción del conjunto de datos

El conjunto de datos utilizado para esta práctica ha sido generado aleatoriamente mediante la función `nngen`. Se ha intentado conseguir un conjunto en donde se diferencien 7 clústeres próximos entre ellos pero que sean linealmente separables.

Los datos han sido generados de la siguiente manera:

```
nngenc([0 5; 0 5; 0 5], 7, 200, 0.30);
```

El primer parámetro (**bounds**) se corresponde con los rangos del espacio en donde generar los datos. Nuestros datos estarán en un espacio de tres dimensiones en el rango $[0, 5]$ en cada dimensión.

El segundo parámetro (**clusters**) es el número de clústeres a generar. Generaremos 7 clústeres linealmente diferenciables.

El tercer parámetro (**points**) es el número de puntos por clúster. Nuestros clústeres tendrán 200 puntos cada uno.

Por último se indica la desviación estándar de los clústeres (**std_dev**). Asignamos una desviación de un 30 % para que los clústeres no estén muy localizados y su agrupamiento no sea tan trivial.

Una vez generados los datos deseados, los hemos guardado en un fichero CSV para su posterior explotación en el estudio.

C. Descripción del procedimiento

Para automatizar el estudio lo máximo posible, se ha realizado un script que entrena un mapa auto-organizado variando los parámetros de dimensiones de la topología, tamaño del vecindario, función de topología y función de distancia entre neuronas.

Por cada configuración, se generan las siguientes figuras y se guardan como imágenes png:

- **plotsomnd**: en esta representación se puede observar la topología del SOM (*self-organizing map*), las conexiones directas entre neuronas y la distancia entre estas. Las neuronas se dibujan como hexágonos azules, las conexiones mediante líneas rojas y las distancias se representan mediante el coloreado de las parcelas, correspondiendo el color amarillo a distancias cortas y el negro a distancias grandes. [2]
- **plotsomhits**: en esta representación se visualiza el número de datos para los que se está activando cada neurona (el número de vectores de entrada que clasifica). Se dibuja la topología del SOM y en cada parcela, correspondiente a una neurona, se muestra el número total así como un hexágono azul de tamaño relativo a este. [1]
- **plotsompos**: muestra un gráfico en 2D en donde se visualizan la posición de las neuronas (puntos azules), las conexiones entre estas (líneas rojas) y las posiciones de los datos (puntos verdes). Permite visualizar cómo el SOM clasifica el espacio de entrada según evoluciona el entrenamiento. [3]

Los comandos **plotsomnd** y **plotsomhits** están limitados a topologías **hextop** o **gridtop**, no pudiendo ser utilizados con **randtop**.

**Todo el código generado en esta práctica se encuentra disponible en el siguiente repositorio: [P2_SelfOrganizingMap](#).*

D. Estudio

TODO describir parámetros modificados (valores y rangos).

TODO se realizaron pruebas para acotar las configuraciones de los parámetros a estudiar.

E. Conclusiones

TODO

Referencias

- [1] Matlab documentation. plotsomhits command, 2016. URL <http://www.mathworks.com/help/nnet/ref/plotsomhits.html>. [Online; accessed 08-October-2016].
- [2] Matlab documentation. plotsomnd command, 2016. URL <http://www.mathworks.com/help/nnet/ref/plotsomnd.html>. [Online; accessed 08-October-2016].
- [3] Matlab documentation. plotsompos command, 2016. URL www.mathworks.com/help/nnet/ref/plotsompos.html. [Online; accessed 08-October-2016].