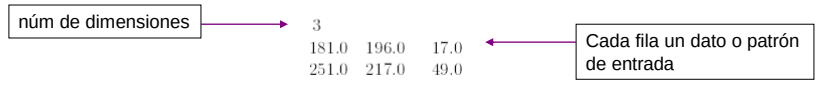
uc3m | Universidad Carlos III de Madrid

OPENCOURSEWARE
REDES DE NEURONAS ARTIFICIALES
Inés M. Galván – José M. Valls

Manual de uso de SOM-PAK

SOM-pak

- The Self-Organizing Map Program Package: Paquete de software de dominio público
- Conjunto de programas, cada uno con una funcionalidad, la cual se describe brevemente en este documento
- Se puede utilizar en windows y linux.
- Descarga de la dirección http://www.cis.hut.fi/research/som_pak/
- **Hay que descargarse directamente los ejecutables** que se encuentran en la carpeta "binaries_windows/" (windows). **No es necesario realizar instalación.**
- En ese enlace puede encontrarse un manuel completo (som_doc)
- Formato de los ficheros de datos:



SOM-pak

- Parámetros de los programas:

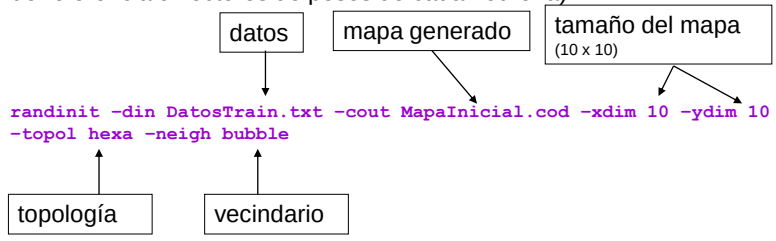
-din	Name of the input data file.
-dout	Name of the output data file.
-cin	Name of the file from which the reference vectors are read.
-cout	Name of the file to which the reference vectors are stored.
-rlen	Running length (number of steps) in training.
-alpha	Initial learning rate parameter. Decreases linearly to zero during training.
-radius	Initial radius of the training area in som-algorithm. Decreases linearly to one during training.
-xdim	Number of units in the x-direction.
-ydim	Number of units in the y-direction.
-topol	The topology type used in the map. Possible choices are hexagonal lattice (*hexa*) and rectangular lattice (*rect*).
-neigh	The neighborhood function type used. Possible choices are step function (*bubble*) and Gaussian (*gaussian*).
-plane	The component plane of the reference vectors that is displayed in the conversion routines.

SOM-pak

- Diferentes programas:
  - Inicialización de las neuronas (vectores de referencia): randinit
  - Entrenamiento: vsom
  - Cálculo del error de cuantización: qerror
  - Calibración del mapa: vcal
  - Cálculo de la neurona ganadora para un conjunto de patrones: Visual
  - Visualización: sammon, umat

Uso de los programas

- **Primer paso: Inicialización aleatoria del mapa (randinit)**
  Dado el tamaño del mapa, su topología y la función de vecindad, se inicializan aleatoriamente la posición de las neuronas del mapa (vectores de referencia o vectores de pesos de cada neurona)



Uso de los programas

- Segundo paso: Entrenamiento del mapa (vsom)
  Se utiliza el algoritmo de entrenamiento SOM de Kohonen. Es recomendable hacerlo en dos fases:
  - Primera fase: se ajustan las posiciones o pesos de la neuronas (aprendizaje) utilizando un conjunto de entrenamiento especificado en -din.

  El radio del vecindario (radius) y la tasa de aprendizaje (alpha) decrecen a lo largo del aprendizaje, por lo que deben tener un valor inicial grande. “radius” no debería superar el tamaño del mapa. “rlen” indica el máximo número de ciclos a realizar

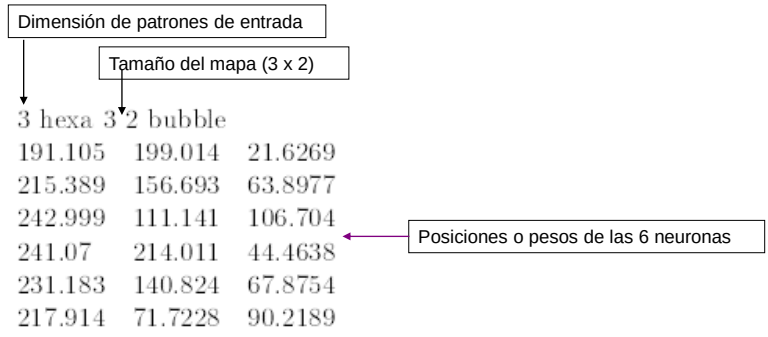
```
vsom -din DatosTrain.txt -cin MapaInicial.cod -cout MapaEntrenado.cod
-rlen 1000 -alpha 0.05 -radius 10
```

  - Segunda fase: Se produce un ajuste fino de las posiciones de las neuronas partiendo del mapa ya entrenado. Para ello es conveniente disminuir los valores iniciales de alpha y radius

```
vsom -din DatosTrain.txt -cin MapaEntrenado.cod -cout MapaEntrenado.cod
-rlen 10000 -alpha 0.02 -radius 3
```

Aspecto de los mapas

- En los ficheros con extensión .cod se almacenan los mapas inicializados o ya entrenados. El formato es:



Uso de los programas

- **Tercer paso: Evaluación del error de cuantización (qerror)**

  qerror indica la distancia entre cada patrón de un conjunto de datos (especificado en –din) (entrenamiento, validación o test) y la neurona ganadora o vector de referencia más cercana del mapa (especificado –cin). Da una idea de como las neuronas se distribuyen en el espacio de entrada.

```
qerror -din Datos.txt -cin MapaEntrenado.cod
```

Qerror grandes: las neuronas están lejos de los patrones

Qerror pequeños: las neuronas están cerca de los patrones

Uso de los programas

- **Cuarto paso: Calibrado del mapa**

Si se quiere usar el SOM para clasificación supervisada, se puede calibrar el mapa con **vcal,** que consiste en asignar una clase a cada neurona. Para ello se utilizan las clases de los ejemplos de entrenamiento.

Ej: mapa sin etiquetar (**MapaEntrenado.cod**)

```
8 hexa 5 5 bubble
0.41112 0.631712 0.615787 0.0498632 0.013705 0.45574 0.134301 0.438569
0.353303 0.618678 0.609085 0.0372994 0.00640025 0.460865 0.131438 0.370228
0.242021 0.584911 0.558684 0.0395156 0.00970134 0.46229 0.132017 0.218037
0.167866 0.555057 0.446242 0.060497 0.0166111 0.42433 0.125583 0.113078
```

Calibrado del mapa:

```
vcal -din DatosTrain.txt -cin MapaEntrenado.cod -cout MapaCalibrado.cod
```

mapa calibrado o etiquetado (**MapaCalibrado.cod**)

Clase asignada a cada neurona

```
8 hexa 5 5 bubble
0.41112 0.631712 0.615787 0.0498632 0.013705 0.45574 0.134301 0.438569 0
0.353303 0.618678 0.609085 0.0372994 0.00640025 0.460865 0.131438 0.370228 1
0.242021 0.584911 0.558684 0.0395156 0.00970134 0.46229 0.132017 0.218037 0
0.167866 0.555057 0.446242 0.060497 0.0166111 0.42433 0.125583 0.113078 0
```


Uso de los programas

- **Quinto paso: Monitorización (visual, sammon, umat)**

Visual genera una lista con las neuronas más cercanas a cada patrón de un conjunto (especificado en –din) (entrenamiento, validación o test) y la distancia entre la neurona y el dato. Ej:

```
visual -din Datos.txt -cin MapaEntrenado.cod -dout SalidaVisual.txt
```

- Aspecto del fichero de salida **SalidaVisual.txt**

```
3 hexa 10 10 bubble
0 2 0.191139
2 2 0.193974
4 0 0.457788
0 2 0.22382
2 0 0.298035
1 3 0.215445
```

El primer dato del fichero pertenece a la célula (0,2) y está a una distancia de 0.191139
Cada línea se corresponde con la misma línea del fichero de datos

Uso de los programas

- **Uso de Visual en Clasificación supervisada**

  Si el mapa está calibrado (vcal), las neuronas tendrán asignada una clase o etiqueta. Entonces **Visual** clasificará cada patrón del conjunto de datos asignándole la etiqueta.

```
visual -din Datos.txt -cin MapaCalibrado.cod -dout SalidaVisualCal.txt
```

- Aspecto del fichero de salida **`SalidaVisualCal.txt`**

```
3 hexa 10 10 bubble
0 2 0.191139 1
2 2 0.193974 0
4 0 0.457788 0
0 2 0.22382 1
2 0 0.298035 0
1 3 0.215445 0
```

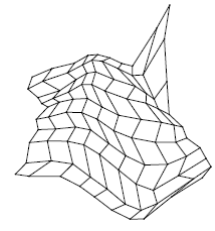
El primer dato del fichero pertenece a la célula (0,2), está a una distancia de 0.1911391 y pertenece a la clase 1 porque la célula (0,2) está etiquetada como 1.

Cada línea se corresponde con la misma línea del fichero de datos

Uso de los programas

- El programa **sammon** permite realizar una proyección del espacio n-dimensional a un plano 2D aproximando las distancias euclídeas de los vectores característicos en el espacio n-dimensional.
  - Genera un fichero de texto con las coordenadas 2D de los prototipos y si se da la opción –ps genera un postcript con la imagen

```
sammon -cin MapaEntrenado.cod -cout proyeccion.sam -rlen 100 -ps
```



Uso de los programas

- El programa **umat** genera un gráfico umatrix donde se visualizan las distancias entre los prototipos de las células con niveles de gris
- Si se usa el mapa calibrado etiqueta las células

```
umat -cin MapaEntrenado.cod > grMapa.ps
```

```
umat -cin MapaCalibrado.cod > grMapaCal.ps
```

Blanco: distancias pequeñas

Negro: distancias grandes