REDES DE NEURONAS
Curso 2008-09

PRÁCTICA 2

PROBLEMA DE CLASIFICACION

IONOSFERA

1. Introducción

En esta práctica se trabajará con un dominio real muy conocido en el campo del Aprendizaje Automático: el dominio de clasificación biclase 'IONOSFERA', compuesto por 351 instancias, con 34 atributos reales o enteros, y accesible en ***UCI Machine Learning Repository*** (http://archive.ics.uci.edu/ml/index.html). Los datos han sido suministrados por

Space Physics Group
Applied Physics Laboratory
Johns Hopkins University

Son datos obtenidos por un sistema de radar de 16 antenas de alta frecuencia situado en Goose Bay, Labrador. El objetivo es la detección de electrones libres en la ionosfera. Cuando se muestra una evidencia de estructura en la ionosfera y se detectan los electrones, la señal se etiqueta como "g" (good, buena). Cuando la señal atraviesa la ionosfera sin mostrar evidencia de estructura en ella, se etiqueta como "b" (bad, mala).

2. Trabajo a realizar

2.1 Obtención de los conjuntos de entrenamiento y de test

Disponemos del fichero ***ionosphere.data*** (***http://archive.ics.uci.edu/ml/machine-learning-databases/ionosphere/ionosphere.data*** que contiene todos los datos. Deberá separarse en los conjuntos de entrenamiento y test de la siguiente forma:

Entrenamiento: *ionosphere.train.data*: 80% de los datos totales
Test: *ionosphere.test.data*: el resto (20% de los datos totales)
Antes de hacer la separación deberán 'aleatorizarse' los datos.

2.2 Clasificación con Perceptron simple

Modificar el programa realizado en práctica 1 para obtener un perceptron simple que permita clasificar los datos. Al no ser estos datos linealmente separables, el perceptron no podrá converger y por tanto habrá que establecer otro criterio de parada.

Obtener el porcentaje de aciertos en el fichero de test.

2.3 Clasificación con LVQ

El algoritmo LVQ es un método de clasificación, muy similar a los mapas de Kohonen, aunque es supervisado. Determina la localización de una serie de protopipos (o centros, o centroides) que representen a cada una de las clases. Una vez localizados estos prototipos, se puede clasificar cada patrón de test asignándole la clase del prototipo más cercano.

El software disponible en *http://www.cis.hut.fi/research/lvq_pak/* es LVQ_PAK. Es muy sencillo de utilizar y está explicado en el siguiente documento:

http://et.evannai.inf.uc3m.es/docencia/rn-inf/documentacion/lvq_doc.pdf

Los pasos básicos serían los siguientes:

- **Inicializar los centros o prototipos**. Esto se hace de una vez para las dos clases, porque el programa se encarga de separar las clases. Se utilizará el programa *eveninit* (el mismo número de prototipos para cada clase) o *propinit* (número de prototipos proporcional a las instancias de cada clase).

- **Entrenar con LVQ** para ajustar estos prototipos de forma supervisada. Hay varios algoritmos de entrenamiento siendo el más apropiado *olvq1*

- **Ver la tasa de aciertos con el fichero de test**
  Programa *accuracy*

  Nos dice la tasa de aciertos por clase y total. Se puede utilizar con cualquier fichero que se quiera clasificar, en este caso nos interesa el fichero de test. Sólo nos dice la tasa, no genera un fichero con la clasificación.

- **Generar el fichero con los patrones clasificados**. Si usamos el fichero de test, ignora la clase a la que pertenece cada patrón y le asigna la clase según el prototipo más cercano.

  El Programa *classify* genera un nuevo fichero de datos con la nueva clasificación. Habría que compararlo con el fichero de test original para ver si cada patrón se ha clasificado correctamente o no. De todos modos, el porcentaje global lo da el programa *accuracy*.

2.4 Mapas autoorganizados de Kohonen

Al ser **SOM** (Self-Organizing Map) un algoritmo no supervisado, no tiene sentido clasificar los datos sino generar un mapa que permita sacar alguna conclusión sobre la organización de dichos datos.

Podrá utilizarse el SOM_PAK, que puede encontrase en
http://www.cis.hut.fi/research/som_pak/

El trabajo consiste en aprender a utilizar el software SOM_PAK generando varios mapas y observar los resultados. En la medida de lo posible se intentarán analizar los mapas obtenidos con las herramientas visuales proporcionadas por este software.

Los pasos básicos serán los siguientes:

- Inicializar el mapa con *randinit* (ver apartado 6.4.1 de la documentación)
- Entrenar el mapa con *vsom* (apartado 6.4.2). Es recomendable hacer el entrenamiento en dos fases, como se explica en dicho apartado.
- Evaluar el error de cuantización (quantization error) con *qerror* (apartado 6.4.3)
- Calibrar el mapa con *vcal* (apdo. 6.4.4), asignando a cada prototipo o centro la etiqueta o clase que le corresponde según una muestra de ejemplos etiquetados. En un caso real, el fichero de entrenamiento no estaría etiquetado (es un algoritmo no supervisado) y se podrían etiquetar los prototipos (calibrar el mapa) disponiendo de algunos ejemplos significativos que sí estuvieran etiquetados. En esta práctica podremos utilizar el conjunto de entrenamiento para calibrar el mapa.
- Obtener un gráfico umatrix con *umat* (apdo. 7.4, página 21)

Opcionalmente, podrán utilizarse otras herramientas de monitorización de los mapas, como *visual*, *sammon*, etc…

2.5 Documentación a entregar

Se entregará una memoria de la práctica que deberá contener, al menos, un capítulo de introducción, otro donde se describe la experimentación realizada con un resumen de los resultados obtenidos (gráficas y tablas) y un capítulo con las conclusiones obtenidas al interpretar los datos experimentales.

Se deberá adjuntar un CD donde se almacenen los ficheros más representativos de la práctica realizada: red o mapa o fichero de prototipos más adecuados y ficheros de resultados.

El plazo de entrega finalizará el día 12 de enero de 2009