Redes de base radial

Vamos a probar redes de respuesta radial para nuestro problema de estimar precios de casas. Usaremos la toolbox Netlab. Vamos a plantearnos dos estrategias de ajuste: una, la especial de respuesta radial, con agrupamiento para la capa oculta y regresión lineal en la salida; otra que empieza igual la capa oculta, pero que ajusta después todos los pesos, por uno de los algoritmos que ya hemos visto.

1. En las prácticas anteriores, en la parte de Netlab, está explicado como cargamos y preparamos los datos. Búscalo y repítelo. La instrucción con la que cargamos los datos es como:

   ```
   [ptot,ttot,nent,nsal,ndata]=datread('casas.dat');
   ```

   Aquí ptot son las variables de entrada, ttot las de salida, nent el número de variables de entrada, nsal el número de salida y ndata la cantidad de datos. Separa una parte para prueba y llámale p y t al resto.

2. Segundo paso ajustando como regresión lineal la capa de salida. *Atención aquí, porque se usa internamente la función kmeans de netlab. Si le habías cambiado el nombre, regrésalo.*

   ```
   radial1=rbf(nent,noc,nsal,'gaussian','linear',alfa,beta);
   radial1.mask = rbfprior('gaussian', nent, noc, nsal, alfa,
       alfa);
   params(2,18)=0;
   params(2,14) = 5;   %5 iteraciones de ajuste para la capa
       oculta
   params(2,5)=1; %Necesario cuando son muchas ocultas
   params(1,1) = 1;   %Con mensajes
   ```

```
radial1 = rbftrain(radial1, params, p,t);%Ajuste
[radial1, numpardeter, logev] = evidence(radial1, p, t, 2);
```

En la primera instrucción noc es la cantidad de procesadores ocultos, que irá variando y elegimos capa oculta gaussiana y la de salida lineal; alfa y beta son los inversos de la varianza supuesta de los pesos y del ruido en los datos. Como regla general empezaremos con alfa pequeña (en el rango de la centésima) y beta grande (decenas). Las instrucciones posteriores las recalcularán.
La siguiente instrucción nos prepara la red para el cálculo de las probabilidades.
La penúltima instrucción es para ajustar los pesos.
Todo esto hay que repetirlo variando noc (no puede ser menos de 2), hasta encontrar una evidencia máxima (logev).

3. Segundo paso ajustando todos los pesos.
   Vamos a crear una red radial. Prepararemos una macro que vaya añadiendo procesadores de respuesta radial hasta llegar a la evidencia máxima. Los comandos para inicializar la red son:

```
radial2=rbf(nent,noc,nsal,'gaussian','linear',alfa,beta);
[radial2.mask, apriori] = rbfprior('gaussian', nent, noc,
    nsal, alfa, alfa);
%Esta inicialización especial es por el ajuste en dos fases
    que se hace
radial2 = netinit(radial2, apriori);
params(14) = 5;   %5 iteraciones de ajuste
params(5)=1; %Necesario cuando son muchas ocultas
params(1) = -1;   %Sin mensajes
radial2 = rbfsetbf(radial2, params, p);%Fase de la capa
    oculta
```

   En la primera instrucción noc es la cantidad de procesadores ocultos, que irá variando y elegimos capa oculta gaussiana y la de salida lineal; alfa y beta son los inversos de la varianza supuesta de los pesos y del ruido en los datos. Como regla general empezaremos con alfa pequeña (en el rango de la centésima) y beta grande (decenas). Las instrucciones posteriores las recalcularán.
   La siguiente instrucción nos prepara las probabilidades a priori iniciales para los pesos de la capa de salida.

Después inicializamos los pesos en la capa de salida.
Las últimas instrucciones son para ajustar los pesos de la capa oculta.
Las varianzas se ajustan a la distancia entre los centros.
Los comandos para ajustarla y recalcular su evidencia son:

```
params(1) = 1;
params(2) = 1.0e-5;% Precision para pesos
params(3) = 1.0e-5;% Precision para el error
params(14) = 50;% Ciclos de optimización
params(18)=0;
radial2=netopt(radial2,params,p,t,'scg');
[radial2, numpardeter, logev] = evidence(radial2, p, t, 2);
```

Primero se ajustan un poco los pesos (con gradiente conjugado escalado) y luego se recalcula la evidencia en dos iteraciones; numpardeter es la cantidad de pesos bien determinada y logev es el logaritmo de la evidencia.
Este grupo de comandos conviene repetirlo varias veces (desde un par hasta 5) para estabilizar el cálculo.
Tanto este ciclo de 4 ó 5 vueltas como la inicialización anterior hay que repetirlo variando noc (no puede ser menos de 2), hasta encontrar una evidencia máxima (logev). Borra params cada vez, para que no te meta valores de una parte en otra.

4. Obtenemos las respuestas de las redes finales para los datos de prueba y una estimación del error con

   ```
   [sr0, error] = netevfwd(netpak(radial), radial, p, t, ppru);
   ```

   Visualizamos su comparación con los resultados correctos con los plot que hemos usado en otras prácticas.
   ¿Qué tal impresión da? ¿Cómo comparan las posibilidades de optimizar para la salida todos los pesos o sólo los de la segunda capa?

5. Comparar con el perceptrón en tamaño y precisión.

6. Repetir con otra partición de la muestra en el paso 1 y valorar la sensibilidad a los datos