REDES DE NEURONAS ARTIFICIALES
Curso 2020-21

PRÁCTICA I. PROBLEMA DE REGRESIÓN
Predicción del precio medio de la vivienda en California

Introducción

El objetivo de esta práctica es abordar un problema real de regresión utilizando dos modelos de redes de neuronas supervisados:

- El modelo lineal Adaline
- El modelo no-lineal Perceptron Multicapa

El problema planteado en esta práctica consiste en predecir el precio medio de la vivienda para diferentes distritos de California a partir de información derivada del censo nacional de 1990. En la Tabla 1 se muestra una breve descripción de la información disponible.

Tabla 1. Descripción de los datos

Atributo	Descripción
longitude	Indica cuán al oeste se encuentra una casa.
latitude	Indica cuán al norte se encuentra una casa.
housingMedianAge	Antigüedad media (mediana) de una casa dentro de un distrito
totalRooms	Cantidad total de habitaciones en las casas de un distrito
totalBedrooms	Cantidad total de camas en las casas de un distrito
population	Cantidad total de residentes en un distrito
households	Cantidad total de grupos familiares (conjunto de personas que reside en una misma casa) en un distrito
medianIncome	Ingreso medio (mediana) de los grupos familiares de un distrito (medido en decenas de miles de dólares estadounidenses)
medianHouseValue	Precio medio (mediana) de una vivienda (medido en dólares estadounidenses)

Se trata de predecir el precio medio de una casa (medianHouseValue) en función del resto de los 8 atributos o variables disponibles. Se dispone un conjunto de 17000 instancias, conjunto disponible en aula global.

Trabajo a realizar

1. Preparación de datos

Antes de realizar el aprendizaje de las redes, hay que realizar un preprocesamiento de los datos disponibles:

- **Normalización:** Es recomendable normalizar las variables de entrada y salida en el rango [0,1]. Para la normalización se calculará el valor mínimo y máximo de cada variable i y se aplicará la siguiente transformación lineal:

$$VarNormal_i = (VarOriginal_i - ValorMin_i)/(ValorMax_i - ValorMin_i)$$

- **Aleatorización**: Para que el entrenamiento de las redes se realice en condiciones adecuadas es importante desordenar o 'aleatorizar' los datos.

- **Separación en tres conjuntos de datos.**
    - Datos de **entrenamiento** (**60% del total de datos**) para realizar el aprendizaje de la red.
    - Datos de **validación** (**20% del total de datos**) que serán utilizados para elegir los valores óptimos de los hiperparámetros de la red (razón de aprendizaje, número de ciclos, número de neuronas).
    - Datos de **test** (**20% del total de datos**) para evaluar la capacidad de generalización de la red.

2. Desarrollo y experimentación con Adaline

Debido a la sencillez de esta red, no se utilizará un simulador sino que se desarrollará un programa que realice el aprendizaje del Adaline, explicado en el Tema 2. Dicho programa se puede desarrollar en el lenguaje de programación que los alumnos decidan.

Para verificar que el programa funciona correctamente, es decir que simula el aprendizaje del Adaline, es aconsejable mostrar en pantalla el error medio a lo largo de los ciclos de aprendizaje (como se muestra en la Tabla 2). El error de entrenamiento debe ir decreciendo o permanecer constante. El error de validación debe ir decreciendo, aunque podría aumentar, lo que significa que se está produciendo sobreaprendizaje.

Tabla 2. Información mostrada en pantalla

Ciclo	Error medio de entrenamiento	Error medio de validación
1	2.1	2.8
2	1.5	1.9
3	0.9	1.7
…	…	…

Además de realizar el aprendizaje de la red, el programa debe:

- Calcular el error sobre el conjunto de test una vez finalizado el aprendizaje.
- Guardar en fichero las salidas de la red para todos los patrones de test.
- Guardar en fichero los pesos y el umbral de la red una vez finalizado el aprendizaje.

Con los datos procesados y el programa desarrollado, se realizarán **diferentes experimentos cambiando el valor de la razón de aprendizaje**, con el objetivo de encontrar el valor más adecuado para el problema dado. El valor óptimo se elegirá utilizando el error de validación. El número de ciclos de aprendizaje más adecuado para cada razón de aprendizaje hay que ajustarlo a cada experimento para conseguir la estabilización del error de entrenamiento y validación. Se obtendrá de la siguiente manera:

- Se entrena la red durante un número de ciclos máximo.
- Se obtiene el número de ciclos que minimiza el error de validación (número de ciclos óptimo)
- A continuación se vuelve a entrenar la red con el conjunto de entrenamiento durante el número de ciclos óptimo para obtener el modelo final. También puede programarse para que se guarde en todo momento el modelo (pesos y umbral) que va obteniendo el menor error de validación, así como el número de ciclos realizados. De esta forma, al finalizar el número máximo de ciclos ya tendremos el modelo y número de ciclos óptimo.
- Con este modelo final entrenado, se utilizará el conjunto de test para comprobar la capacidad de generalización de la red.

3. Experimentación con el Perceptron Multicapa

Para el uso de Perceptron Multicapa (PM) se va a utilizar el **simulador SNNS** bajo el lenguaje de programación R. El paquete **RSNNS** permite un uso fácil de dicho simulador bajo el entorno de R. Se facilitará el script básico a utilizar para entrenar el PM, así como para calcular su error en diferentes conjuntos de datos o realizar un gráfico que muestre la evolución de los errores a lo largo de las iteraciones.

En la experimentación con el PM se realizarán diferentes pruebas, **cambiando el número de neuronas ocultas y la razón de aprendizaje,** con el objetivo de encontrar los valores más adecuados para el problema que se pretende resolver. Los valores óptimos se elegirán utilizando el error de validación. El número de experimentos a realizar siempre depende del problema, aunque se sugiere realizar como mínimo **tres experimentos cambiando el número de neuronas ocultas y otros tres cambiando la razón de aprendizaje**. Igual que con Adaline, el número de ciclos de aprendizaje más adecuado para cada configuración hay que ajustarlo a cada experimento para conseguir minimizar el error de validación. El script que se facilitará ya vendrá preparado para obtener el número de ciclos más adecuado.

Una vez finalizado el aprendizaje, se utilizará el conjunto de test para comprobar la capacidad de generalización de la red.

NOTA IMPORTANTE: Para poder comparar los resultados del PM con los resultados del Adaline, no hay que olvidar que los errores que se estén computando sean los mismos y no expresiones diferentes. Se recomienda utilizar el error medio absoluto (MAE) o el error cuadrático medio (MSE)

4. Entrega de la práctica

Se entregará una memoria de la práctica en PDF y un fichero comprimido que contendrá el trabajo realizado.

La entrega de la práctica se realizará **el 27 de octubre**. En caso de que haya que modificar esta fecha se avisará con suficiente antelación.

4.1 Normas entrega Memoria

La memoria deberá contener, al menos, un capítulo de introducción, un capítulo explicando cómo se ha realizado el preproceso, un capítulo para cada uno de los modelos (Adaline y Perceptron Multicapa), con una descripción en cada caso de la experimentación realizada, los resultados obtenidos y un análisis de los resultados. Y, finalmente, un capítulo con la comparación de ambos modelos y las conclusiones obtenidas al interpretar y comparar los resultados experimentales.

La memoria deberá tener un máximo de 8 páginas, con formato standard: margen normal, interlineado sencillo, letra arial, tamaño 11.

Los resultados que deben mostrarse en la memoria para cada modelo (Adaline y PM) son los siguientes:

- *Evolución de los errores a lo largo del aprendizaje solo para algunos (o algún) de los experimentos (los más significativos)*

- *Una tabla que contenga los resultados obtenidos para todos los experimentos realizados (hiperparámetros utilizados, ciclos, errores de entrenamiento, validación y test)*
- *Para el mejor experimento en cada caso (ADALINE y PM) una gráfica que muestre la salida obtenida por la red y la salida deseada para los datos de test. Estas salidas deberán "desnormalizarse" para que el rango de valores sea el original. Para ello, serán necesarios los valores máximo y mínimo que se utilizaron para hacer la normalización de la salida.*

4.2 Norma entrega fichero comprimido

El fichero comprimido contendrá todos los ficheros que se indican a continuación.

- Ficheros de datos de entrenamiento, validación y test utilizados
- Para la parte del ADALINE:
  - Programa fuente del ADALINE
  - Fichero que contenga los pesos y el umbral obtenidos para el mejor experimento.
  - Ficheros que contengan las salidas deseadas y las salidas del ADALINE desnormalizadas del mejor experimento para el conjunto de test.
  - Fichero que contenga la evolución del error en entrenamiento y validación del mejor experimento.
- Para la parte del Perceptron Multicapa:
  - Fichero que contenga la red obtenida para el mejor experimento.
  - Ficheros que contengan las salidas deseadas y las salidas del Perceptrón Multicapa desnormalizadas del mejor experimento para el conjunto de test.
  - Fichero que contenga la evolución del error en entrenamiento y validación del mejor experimento.