

REDES DE NEURONAS ARTIFICIALES

Curso 2021-22

PRÁCTICA I. PROBLEMA DE REGRESIÓN

Resistencia a la compresión del hormigón

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

Se trabajará con el dominio real de regresión conocido como “Concrete Compressive Strength (Resistencia a la compresión del hormigón)”. El hormigón es uno de los materiales más utilizados en la ingeniería civil. Para los expertos en este campo es muy importante conocer la resistencia de dicho material a la fuerza de compresión. Además, es de gran interés conocer dicha resistencia en función de variables como la edad del hormigón o los ingredientes que lo componen, evitando así el tener que realizar ensayos con el propio material antes de su uso.

Para abordar el problema se dispone de un conjunto de datos que contiene 1030 instancias con 8 variables de entrada: la edad del hormigón en días y las cantidades de 7 componentes (cemento, escoria de alto horno, cenizas volantes, agua, superplastificante, agregado grueso y agregado fino), medidas en kg por m³ de mezcla. La variable de salida o variable a predecir es la resistencia que se observó en el hormigón a la fuerza de compresión, medida en MPa (megapascuales). En el siguiente enlace <http://sci2s.ugr.es/keel/dataset.php?cod=44> puede encontrarse una breve descripción del conjunto de datos.

Los datos que serán utilizados para la realización de la práctica podrán descargarse en la sección Practica I en Aula Global.

Trabajo a realizar

1. Preparación de datos

Antes de realizar el aprendizaje de las redes, hay que realizar un preproceso 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:

$$VarNorm_i = (VarOrig_i - ValorMin_i) / (ValorMax_i - ValorMin_i)$$

- **Aleatorización:** Para que el entrenamiento de las redes se realice en condiciones adecuadas es importante desordenar o ‘aleatorizar’ los datos.
- **Separación de tres conjuntos de datos:**
 - Datos de **entrenamiento (70% del total de datos)** para realizar el aprendizaje de la red.
 - Datos de **validación (15% 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 en Perceptron Multicapa).

- Datos de **test (15% 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 las clases de teoría. 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 siguiente tabla). El error de entrenamiento debe ir decreciendo o permanecer constante. El error de validación debe ir decreciendo, aunque pudiera aumentar, lo que significa que se está produciendo sobreaprendizaje.

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

Tabla 1

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 (es decir, el modelo) 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). Nuestro modelo final entrenado será el que corresponda con el menor error de validación. Esto podrá conseguirse de cualquiera de las siguientes formas:
 - Se repite el entrenamiento con los mismos valores aleatorios iniciales pero solo el número de ciclos que minimizaban el error de validación.
 - Se entrena la red una sola vez pero en cada ciclo guardamos el modelo si el error de validación es menor que el obtenido hasta este momento. Cuando lleguemos al número de ciclos máximo tendremos guardado el mejor modelo.

- 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, así como un **fichero comprimido** que contendrá los ficheros correspondientes a los datos utilizados, los resultados y los modelos entrenados.

El plazo para la entrega de la práctica finalizará **el 31 de octubre**. En caso de que haya que modificar esta fecha se avisará con suficiente antelación.

4.1 Normas de entrega de la 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.

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

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

- Una tabla que contenga los resultados obtenidos para todos los experimentos realizados (hiperparámetros utilizados, errores de entrenamiento, validación y test)
- Una gráfica que muestre la salida obtenida por la red y la salida deseada para los datos de test, que correspondan al mejor experimento en cada caso (ADALINE y PM). Estas salidas deberán “desnormalizarse” para que el rango de valores sea el original. Es decir, los datos estarán dados en MPa. Para desnormalizar estos valores, 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 en formato ZIP contendrá todos los ficheros que se indican a continuación.

- Ficheros de datos de entrenamiento, validación y test utilizados
- Para la parte del ADALINE:
 - Código fuente del ADALINE en el lenguaje elegido.
 - Fichero que contenga el modelo (los pesos y el umbral) 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.