uc3m

Campus de Colmenarejo

Escuela Politécnica Superior Campus Colmenarejo
Universidad Carlos III de Madrid

Problema de regresión
Resistencia a la compresión del hormigón

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Luis Cabrero García
Tutor: José María Valls Ferrán

Curso 4º - Gr80

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

La presente práctica trata de abordar mediante el punto de vista de las redes de neuronas supervisadas la resolución de un problema real: la predicción de la resistencia del hormigón a la fuerza de compresión. Para ello disponemos de ocho variables de entrada: la edad del hormigón en días y siete variables que representan los componentes que forman el hormigón (cemento, escoria de alto horno, cenizas volantes, agua, superplastificante, agregado grueso y agregado fino). Estos datos han sido tomados del KEEL[1].

1.1 Motivación

La utilidad de los modelos de regresión que vamos a aplicar de forma práctica es reseñable: nos permiten determinar el valor de la resistencia a la compresión del hormigón a partir de sus características, lo que tiene bastante interés a la hora de saber predecir la resistencia del material en construcciones reales. Si somos capaces de mirar un poco más alla somos capaces de ver que estos modelos de regresión lineal (Adaline) y no lineal (Perceptrón multicapa) son capaces de estimar salidas en función de los datos de entrada pero no solo en este caso sino en problemas de otra índole. Esto nos da una primera idea de la potencia de estos modelos.

1.2 Objetivos

Los objetivos de la práctica son: procesar los datos del KEEL[1], implementar el aprendizaje del modelo Adaline en el lenguaje de programación deseado (en mi caso PHP), grabar en distintos ficheros los resultados del aprendizaje de la red, las salidas obtenidas y error cometido para el conjunto de test, y posteriormente realizar experimentación con MLP y el modelo Adaline. Posteriormente se realizará una comparación entre ambos modelos para ver en que casos un modelo puede aproximar mejor que el otro.

1.3 Estructura de la memoria

La memoria dispone de tres índices, uno para consultar cada una de las secciones y otros dos para figuras y tablas. En la memoria se puede encontrar un apartado general para el modelo Adaline y otro para el Perceptrón multicapa. Dentro de los respectivos apartados de cada modelo, se explica la base teórica de cada uno de ellos, se explica la implementación en el caso de Adaline, y en ambos se explica los distintos experimentos realizados y se razona el motivo por el que se consideran dichos experimentos como ne-

cesarios y suficientes. En las conclusiones se explican las diferencias entre ambos modelos y se comparan los resultados obtenidos en la experimentación. Al final de la memoria se indican las referencias bibliográficas consultadas para llevar a cabo el siguiente trabajo.

CAPÍTULO 2
Modelo Adaline

En este capítulo se va a explicar en primer lugar el modelo Adaline, se va a indicar cuales son sus ventajas e inconvenientes, se explicará la implementación que se ha llevado a cabo en PHP y el formato que se ha elegido para devolver las salidas del modelo. También se expermientará con distintos ciclos de aprendizaje y distintas razones de aprendizaje, y se verá el comportamiento de la red en todos los casos posibles, para ello se representará cada posible situación con un experimento para abordar todos los casos posibles.

2.1 ADALINE: ADAptive LInear NEuron

Es un tipo de red de neuronas artificiales desarrollada por Bernard Widrow y Ted Hoff. La red está compuesta por n neuronas(2.1), con m entradas. La salida de cada neurona se representa por la función de activación(2.1) y representa el impacto de cada peso sobre las entradas, sumando el umbral de la neurona (θ).

$$y = \sum_{i=1}^{n} w_i * x_i + \theta \tag{2.1}$$

En el caso que nos ocupa(2.2), teniendo ocho entradas, la salida que se obtiene es la estimación de la resistencia del hormigón a la compresión. Los pesos aplicables a cada una de las entradas, junto con el umbral, se inicializan de manera aleatoria entre *-0.5* y *0.5*.

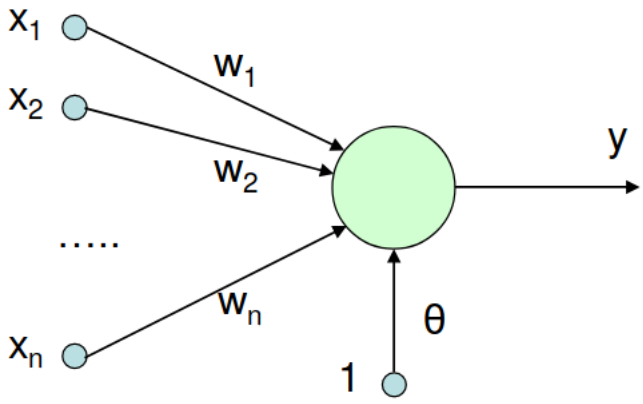

Figura 2.1: Neurona ADALINE [2]

$$ConcreteCompressiveStrengthEst = \sum_{i=1}^{8} w_i * x_i + \theta \tag{2.2}$$

El **aprendizaje** de la red, se realiza teniendo en cuenta la diferencia entre la salida deseada (adjunta como dato en los conjuntos de entrenamiento, test y validación) y la salida obtenida(2.1). Teniendo en cuenta esta diferencia se trata de minimizar el error obtenido mediante la modificación de los pesos y el umbral. Para hallar el error cuadrático medio(2.3) se opera con todos los errores cuadráticos obtenidos en cada uno de los N patrones.

$$E = \frac{1}{N} * \sum_{p=1}^{N} (d^p - y^p)^2 \tag{2.3}$$

Por último, la modificación de los pesos y umbral se realiza siguiendo la Regla Delta, que busca la minimización iterativa de la función de error(2.3), realizando un cambio a cada peso proporcional a la derivada del error, medida en el patrón actual, respecto del peso (2.4). El umbral también se modifica(2.5).

$$\Delta_p w_j = -\gamma \frac{\partial E^p}{\partial w_j} = \gamma * (d^p - y^p) * x_j \tag{2.4}$$

$$\Delta_p \theta = \gamma * (d^p - y^p) \tag{2.5}$$

siendo γ la razón o tasa de aprendizaje.

En este punto es importante resaltar que los datos deben **normalizarse**, **aleatorizarse** y **separados** en tres conjuntos de datos: **entrenamiento**, **validación** y **test**.

- **Datos de entrenamiento**: (70 % del total de datos) para realizar el aprendizaje de la red.
- **Datos de validación**: (15 % del total de datos) utilizados para elegir los valores óptimos de los parámetros de la red (razón de aprendizaje, número de ciclos, número de neuronas).
- **Datos de test**: (15 % del total de datos) para evaluar la capacidad de generalización de la red.

2.2 Implementación del modelo

El modelo se ha decidido implementar en PHP por estar el autor familiarizado con dicho lenguaje y por la tremenda facilidad que ofrece a la hora de hacer cambios de tipo. En el fichero adaline.php se define la clase Adaline, que tiene atributos que caracterizan a la red de una sola neurona. Se leen los parámetros pasados por terminal y se ejecuta el aprendizaje de la red. Posteriormente se pasa el conjunto de validación y por último se pasa el conjunto de test.

```
<?php

    //EJECUCION: php entrenamiento.php <tasa_aprendizaje> <num_ciclos>

    include 'adaline.php';
```

```
    $tasa_aprendizaje = $argv[1];
    $num_ciclos = $argv[2];
    $adaline = new Adaline($tasa_aprendizaje);
    $adaline->ejecutaradaline($num_ciclos);

?>
```

Especificación de los métodos implementados

- *construct*. Parámetro: tasa de aprendizaje. Inicializa pesos y umbral de manera aleatoria, inicializa la tasa de aprendizaje, inicializa los arrays necesarios.
- *aprendizaje*. Parámetros: fichero a utilizar y ciclo. Realiza el aprendizaje de la red utilizando los datos de entrenamiento provenientes de un fichero csv.
- *error*. Parámetros: fichero a utilizar y ciclo. Calcula el error producido en un ciclo de entrenamiento sin modificar los pesos ni el umbral.
- *errorvalidacion*. Parámetros: fichero a utilizar y ciclo. Calcula el error producido en un ciclo de validación sin modificar los pesos ni el umbral.
- *errortest*. Parámetro: fichero a utilizar. Calcula el error cuadrático medio sobre el conjunto de los datos de test.
- *resultados*. Parámetros: carpeta, ciclo y número de ciclos. Muestra en un fichero HTML las salidas obtenidas para el conjunto de entrenamiento y validación normalizadas y desnormalizadas.
- *mostrarerrores*. Parámetros: carpeta y número de ciclos. Muestra en un fichero HTML los errores cuadráticos medios por cada ciclo para entrenamiento y validación, el valor del error cuadrático medio para el conjunto de test, los pesos y umbral resultantes tras aprendizaje y las salidas desnormalizadas de test.

2.3 Experimentación realizada

CAPÍTULO 3
??? ???? ??????

????? ????????????? ????????????? ????????????? ????????????? ?????????????

3.1 ?? ???? ???? ? ?? ??

????? ???????El objetivo es obtener una red tal que y?????? ???????????? ????????????? ????????????? ?????????????

CAPÍTULO 4
Conclusiones

????? ????????????? ????????????? ????????????? ????????????? ?????????????

Bibliografía

[1] Conjunto de datos del hormigón, Knowledge Extraction based on Evolutionary Learning. Obtenido de http://sci2s.ugr.es/keel/dataset.php?cod=44.

[2] Primeros modelos computacionales, Inés M. Galván y José Mª Valls. Obtenido de http://ocw.uc3m.es/ingenieria-informatica/redes-de-neuronas-artificiales/transparencias/material-de-clase.-tema-2/view.

[3] Peceptrón Multicapa, Inés M. Galván y José Mª Valls. Obtenido de http://ocw.uc3m.es/ingenieria-informatica/redes-de-neuronas-artificiales/transparencias/material-de-clase.-tema-3/view.