

Práctica 2: Perceptrón multicapa

Curso 2020–2021

Laboratorio de Neurocomputación
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Fecha de inicio: Lunes 15/03/2021 (2462) / Martes 16/03/2021 (2461)
Fecha de entrega: Domingo 11/04/2021 (2462) / Lunes 12/04/2021 (2461)

1. Algoritmo de retropropagación

Las redes neuronales implementadas en la práctica anterior, el Perceptrón y el Adaline, contienen una única capa y solo pueden resolver problemas linealmente separables. La forma de abordar otros problemas consiste en añadir más capas (denominadas capas ocultas) junto con funciones de activación no lineales. El problema que surge a continuación es el entrenamiento de estas redes multicapa. La solución es el algoritmo de retropropagación, que tiene tres fases:

1. Propagación hacia delante: Se introduce el patrón de entrada de la red como estímulo y se propagan las activaciones a través de las capas hasta producir una salida.
2. Retropropagación del error: La señal de salida generada se compara con la salida deseada y se calcula el error. Dicho error se propaga hacia atrás desde la capa de salida.
3. Ajuste de pesos: Los pesos de la red se actualizan para minimizar la salida generada por la red y la salida deseada.

Estas fases anterior, iteradas múltiples veces para cada patrón de entrenamiento, fuerza a que las neuronas de la red aprendan a reconocer durante el entrenamiento ciertas características que se utilizan después para clasificar patrones que no han sido observados por la red, es decir, permite que la red generalice.

Implementar el algoritmo de retropropagación para perceptrones multicapa con funciones de activación sigmoide bipolar en las capas ocultas y capa de salida:

$$f(x) = \frac{2}{1 + \exp(-x)} - 1, f'(x) = \frac{1}{2}(1 + f(x))(1 - f(x))$$

Para esta práctica, se tendrá en cuenta lo siguiente:

- El número de neuronas de la capa de salida coincidirá con el número de clases y la clase predicha será la que consiga una mayor activación.
- Debe ser posible configurar el número de neuronas de la capa oculta, la tasa de aprendizaje y deberá guardar el error cuadrático medio de cada época, la tasa de aciertos, la matriz de confusión, etc.
- También deben estar soportados los tres modos de lectura de ficheros de la anterior práctica.
- Solo es necesario implementar el algoritmo de retropropagación para una capa oculta. No obstante, los alumnos que lo deseen, podrán ampliar esta práctica añadiendo más capas ocultas y generalizando en algoritmo de retropropagación.

1.1. Memoria

- Describe el algoritmo de retropropagación implementado.

1.2. Código

El código debe ir acompañado con un **Makefile** que permita la compilación y ejecución. Tendrá, al menos, los siguientes objetivos:

- **compilar**: Compilación de todo, incluir aunque no haga nada.
- **ayuda_retro**: Explicación de los argumentos necesario para ejecutar el programa.

2. Problemas reales

Estudiad la aplicación de perceptrones multicapa a los problemas reales 1, 2, 3 y 5. Se deberán probar diferentes configuraciones de red neuronal así como parámetros para obtener los mejores resultados.

2.1. Memoria

- Describid todos los experimentos llevados a cabo y los resultados obtenidos.

2.2. Código

En el **Makefile** habrá, al menos, el siguiente objetivo:

- **ejecuta_problema_real2**: Ejecución concreta para el problema 2 (modo 1) y con un buen resultado.
- Includ el fichero **prediccion_problema_real2.txt** en la carpeta raíz de vuestra entrega. Deberá leerse en modo 1 y dar un buen resultado.

3. Problemas reales y normalización

En los problemas reales 4 y 6 no es posible obtener buenos resultados tal cual vienen los datos. Determina el motivo de este problema (puedes calcular la media y la desviación estándar de los valores de los atributos y reflexionar sobre la adecuación de los valores). Transformad los valores de los atributos de entrenamiento de manera adecuada y recordad que a los datos de test se les aplica la misma transformación que se ha aplicado a los datos de entrenamiento.

3.1. Memoria

- Explicad la normalización aplicada a los datos, cómo se ha hecho y porqué se ha hecho.

3.2. Código

El código debe ir acompañado con un **Makefile** que permita la compilación y ejecución. Tendrá, al menos, los siguientes objetivos:

- **ejecuta_problema_real6**: Ejecución concreta para el problema 6 con la siguiente configuración:
 - Una capa oculta con 20 neuronas
 - Tasa de aprendizaje: 0.1
 - Normalización de atributos
 - Porcentaje de entrenamiento: 70 %
 - Épocas: 5000 (sin posibilidad de terminar antes)

4. Entrega y memoria

- Contestar a las cuestiones planteadas en los diferentes apartados en una memoria. Incluir los resultados en gráficas que muestren los resultados.
- La entrega en Moodle se hará con el siguiente formato:

Practica2-ParejaXX-Apellido1-Apellido2.zip