

PRÁCTICA 3 DE NEUROCOMPUTACIÓN

Curso 2013-2014

Otras aplicaciones de las Redes Neuronales Artificiales

PARTE 2

Series temporales

En esta parte usaremos las redes neuronales desarrolladas hasta el momento para predecir series temporales. Tenemos dos ficheros de series temporales (p3serie1.txt, p3serie2.txt), donde cada fila es un tiempo (los tiempos están ordenados), y en cada fila hay un solo valor.

En estas dos series usaremos las redes neuronales para predecir el comportamiento de la serie temporal.

Hay dos formas de abordar el problema, se deberá elegir una de ellas:

Opción 1: clasificación del comportamiento de la serie en el siguiente punto.

- **Crea una función *adapta-fichero-serie* que tome como argumentos:**

- Nombre del fichero con la serie original
- Nombre del fichero que se quiere crear
- Número de puntos anteriores (N_p) que se quieren usar en la predicción actual.

Dicha función deberá leer el fichero original y crear el fichero nuevo en el formato que hemos usado hasta ahora para definir problemas de clasificación.

Como ejemplo, supón que el fichero de la serie original contiene sólo 7 puntos (7 filas):

0.5

0.3

0.6

0.7

0.1

0.1

0.5

Si $N_p=1$, nuestra función deberá crear el siguiente fichero que contiene 6 puntos:

```
1 2
0.5 0 1
0.3 1 0
0.6 1 0
0.7 0 1
0.1 0 1
0.1 1 0
```

Recuerda que en la primera línea, en este caso, 1 es el número de atributos, y 2 el número de clases. La primera clase, (1 0), representa que el valor de la serie aumenta, y la segunda, (0 1), que no aumenta.

Si N_p fuera 2, nuestra función deberá crear el fichero siguiente con 5 puntos:

```
2 2
0.5 0.3 1 0
0.3 0.6 1 0
0.6 0.7 0 1
0.7 0.1 0 1
0.1 0.1 1 0
```

En general, el número de puntos en el fichero creado es igual al del fichero original menos N_p .

- Cambia el código de tu red neuronal para que el conjunto de entrenamiento esté formado por los primeros datos del fichero de entrada a la red, y el conjunto de test por el restante. Esto lo haremos porque ahora sí que hay un orden temporal en el fichero: los primeros datos representan el pasado, lo que usamos para construir el modelo. El test representa datos futuros, lo que queremos predecir.
- Entrena tu red con p3serie1.txt usando un tamaño de entrenamiento de 100 patrones, y un tamaño de test de 300 patrones. ¿Es capaz de predecir adecuadamente tu red? ¿con cuántas neuronas? Prueba $N_p=1$, $N_p=2$ y $N_p=5$.

- Haz lo mismo pero usando ahora un tamaño de entrenamiento de 200 patrones, y un conjunto de test con el resto de patrones. ¿Ves alguna diferencia?

Prueba $N_p=1$, $N_p=2$ y $N_p=5$.

Para entender lo que está pasando, se recomienda visualizar la serie.

- Repetir los mismos apartados anteriores con los datos p3serie2.txt

¿Qué nivel de error comete tiene tu red? ¿Hay el mismo error a lo largo de todo el test, o los errores se concentran en alguna zona?

Opción 2:

En esta opción usaremos la red neuronal para realizar regresión, y no clasificación como hasta ahora. El objetivo de la red es entonces predecir el valor numérico en el siguiente punto en el tiempo.

Crea una función *adapta-fichero-serie* que tome como argumentos:

- Nombre del fichero con la serie original
- Nombre del fichero que se quiere crear
- Número de puntos anteriores (N_p) que se quieren usar en la predicción actual.

Dicha función deberá leer el fichero original y crear el fichero nuevo en un formato extendido.

Como ejemplo, supón que el fichero de la serie original contiene sólo 7 puntos (7 filas):

0.5

0.3

0.6

0.7

0.1

0.1

0.5

Si $N_p=1$, nuestra función deberá crear el siguiente fichero que contiene 6 puntos:

```
1 1
0.5 0.3
0.3 0.6
0.6 0.7
0.7 0.1
0.1 0.1
0.1 0.5
```

En la primera línea, en este caso, 1 es el número de atributos, y 1 el número de puntos siguientes a predecir. Este último número será siempre 1. Es decir, la salida que se espera de la red es el valor de la serie en el siguiente instante temporal.

Si N_p fuera 2, nuestra función deberá crear el fichero siguiente con 5 puntos:

```
2 1
0.5 0.3 0.6
0.3 0.6 0.7
0.6 0.7 0.1
0.7 0.1 0.1
0.1 0.1 0.5
```

En general, el número de puntos en el fichero creado es igual al del fichero original menos N_p .

- Cambia el código de tu red neuronal para que se adapte al nuevo formato (lo único que cambia es que ya no hay clases sino un solo número a predecir), y para que el conjunto de entrenamiento esté formado por los primeros datos del fichero de entrada a la red, y el conjunto de test por el restante. Esto lo haremos porque ahora sí que hay un orden temporal en el fichero: los primeros datos representan el pasado, lo que usamos para construir el modelo. El test representa datos futuros, lo que queremos predecir.
- Cambia la configuración de tu red neuronal para que sólo haya una neurona de salida, que tendrá una función de transferencia lineal. Esta neurona será la encargada de predecir el valor siguiente. Cambia las funciones de entrenamiento y predicción de la red neuronal para tener en cuenta esto.

- Entrena tu red con p3serie1.txt usando un tamaño de entrenamiento de 100 patrones, y un tamaño de test de 300 patrones. ¿Es capaz de predecir adecuadamente tu red? ¿con cuántas neuronas ocultas? Prueba $N_p=1$, $N_p=2$ y $N_p=5$.

Computa los errores cuadráticos medios tanto en entrenamiento como en test. Compara dichos valores con los errores del modelo más básico para predicción de series temporales: el valor predicho es igual al último valor observado.

- Haz lo mismo pero usando ahora un tamaño de entrenamiento de 200 patrones, y un conjunto de test con el resto de patrones. ¿Ves alguna diferencia?

Prueba $N_p=1$, $N_p=2$ y $N_p=5$.

Para entender lo que está pasando, se recomienda visualizar la serie original.

- Implementa una función **predicción-recursiva que**, una vez entrenada la red, permita reconstruir la serie a partir de un punto determinado. Dicha función tomará como entrada la red ya entrenada, un entero que determinará el número de puntos futuros a predecir, y un vector de longitud N_p que determina el punto a partir del cuál queremos predecir la serie recursivamente.

Por ejemplo, si tenemos una red ya entrenada en la que hemos usado $N_p=5$, y queremos predecir qué va a pasar a partir del último punto en entrenamiento, por ejemplo [0.4 0.1 0.3 0.6 0.5], dicha función deberá:

Inicializar un vector V interno a esa función como:

```
V <- [0.4 0.1 0.3 0.6 0.5]
```

1. Predecir cuál es el punto siguiente a V . Supongamos que la predicción fuera 0.2
2. Desplazar dicho vector un valor a la izquierda para incluir ese 0.2 como último elemento. Por ejemplo, en la primera iteración daría [0.1 0.3 0.6 0.5 0.2].
3. Iterar desde 1 tantas veces como puntos futuros queramos predecir.

Haz predicciones recursivas a partir de los últimos puntos de entrenamiento y representa gráficamente la predicción comparándola con la serie real. ¿Cuántos puntos podemos predecir de manera fiable?

- Repetir los mismos apartados anteriores con los datos p3serie2.txt

¿Qué nivel de error comete tiene tu red? ¿Hay el mismo error a lo largo de todo el test, o los errores se concentran en alguna zona?