

Práctica 2: Perceptrón multicapa para problemas de clasificación

Convocatoria de febrero (curso académico 2016/2017)

Asignatura: Introducción a los modelos computacionales
4º Grado Ingeniería Informática (Universidad de Córdoba)

17 de octubre de 2016

Resumen

Esta práctica sirve para familiarizar al alumno con la adaptación de un modelo computacional de red neuronal a problemas de clasificación, en concreto, con la adaptación del perceptrón multicapa programado en la práctica anterior. Por otro lado, también se implementará la versión *off-line* del algoritmo de entrenamiento. El alumno deberá programar estas modificaciones y comprobar el efecto de distintos parámetros sobre una base de datos real de clasificación de dígitos escritos a mano, con el objetivo de obtener los mejores resultados posibles. La entrega se hará utilizando la tarea en Moodle habilitada al efecto. Se deberá subir, en un único fichero comprimido, todos los entregables indicados en este guión. El día tope para la entrega es el **6 de noviembre de 2016**. En caso de que dos alumnos entreguen prácticas copiadas, no se puntuarán ninguna de las dos.

1. Introducción

El trabajo a realizar en la práctica consiste en adaptar el algoritmo de retropropagación realizado en la práctica anterior a problemas de clasificación. En concreto, se dotará un significado probabilístico a dicho algoritmo mediante dos elementos:

- Utilización de la función de activación *softmax* en la capa de salida.
- Utilización de la función de error basada en entropía cruzada.

Además, se implementará la versión *off-line* del algoritmo.

El alumno deberá desarrollar un programa capaz de realizar el entrenamiento con las modificaciones anteriormente mencionadas. Este programa se utilizará para entrenar modelos que predigan de la forma más correcta posible un conjunto de bases de datos disponible en Moodle y se realizará un análisis de los resultados obtenidos. **Este análisis influirá en gran medida en la calificación de la práctica.**

En el enunciado de esta práctica, se proporcionan valores orientativos para todos los parámetros del algoritmo. Sin embargo, se valorará positivamente si el alumno encuentra otros valores para estos parámetros que le ayuden a mejorar los resultados obtenidos. La única condición es que no se puede modificar el número máximo de iteraciones del bucle externo (establecido a 1000 iteraciones para los problemas XOR e *iris*, y 300 iteraciones para la base de datos *digits*).

La sección 2 describe una serie de pautas generales a la hora de implementar las modificaciones del algoritmo de retropropagación. La sección 3 explica los experimentos a realizar una vez implementadas dichas modificaciones. Finalmente, la sección 4 especifica los ficheros a entregar para esta práctica.

2. Implementación del algoritmo de retropropagación

Se deben de seguir las indicaciones aportadas en las diapositivas de clase para añadir las siguientes características al algoritmo implementado en la práctica anterior:

- *Incorporación de la función softmax*: Se incorporará la posibilidad de que las neuronas de capa de salida sean de tipo *softmax*, quedando su salida definida de la siguiente forma:

$$net_j^H = w_{j0} + \sum_{i=1}^{n_H-1} w_{ji} out_i^{H-1}, \quad (1)$$

$$o_j = out_j^H = \frac{\exp(net_j^H)}{\sum_{l=1}^{n_H} \exp(net_l^H)}. \quad (2)$$

- *Utilización de la función de error basada en entropía cruzada*: Se dará la posibilidad de utilizar la entropía cruzada como función de error, es decir:

$$L = -\frac{1}{N} \sum_{p=1}^N \left(\frac{1}{k} \sum_{o=1}^k d_{po} \ln(o_{po}) \right), \quad (3)$$

donde N es el número de patrones de la base de datos considerada, k es el número de salidas, d_{po} es 1 si el patrón p pertenece a la clase o (y 0 en caso contrario) y o_{po} es el valor de probabilidad obtenido por el modelo para el patrón p y la clase o .

- *Modo de funcionamiento*: El algoritmo podrá trabajar en modo *off-line* o *batch*. Esto es, por cada patrón de entrenamiento (bucle interno), calcularemos el error y acumularemos el cambio, pero no ajustaremos los pesos de la red. Una vez procesados todos los patrones de entrenamiento (y acumulados todos los cambios), entonces ajustaremos los pesos, comprobaremos la condición de parada del bucle externo y volveremos a empezar por el primer patrón, si la condición no se cumple.
- El resto de características del algoritmo (utilización de ficheros de *entrenamiento* y un fichero de *test*, *condición de parada*, *copias de los pesos* y *semillas para los números aleatorios*) se mantendrán similares a como se implementaron en la práctica anterior.

3. Experimentos a realizar

Probaremos distintas configuraciones de la red neuronal y ejecutaremos cada configuración con cinco semillas (10, 20, 30, 40 y 50). A partir de los resultados obtenidos, se obtendrá la media y la desviación típica del error. Aunque el entrenamiento va a guiarse utilizando la función de entropía cruzada o el *MSE*, el programa deberá mostrar el porcentaje de patrones bien clasificados, ya que en problemas de clasificación es ésta medida de rendimiento la que nos interesa. El porcentaje de patrones bien clasificados se puede expresar de la siguiente forma:

$$CCR = 100 \times \frac{1}{N} \sum_{p=1}^N (I(y_p = y_p^*)) \quad (4)$$

donde N es el número de patrones de la base de datos considerada, y_p es la clase deseada para el patrón p (es decir, el índice del valor máximo del vector \mathbf{d}_p , $y_p = \arg \max_o d_{po}$) e y_p^* es la clase obtenida para el patrón p (es decir, el índice del valor máximo del vector \mathbf{o}_p o la neurona de salida que obtiene la máxima probabilidad para el patrón p , $y_p^* = \arg \max_o o_{po}$).

Para valorar cómo funciona el algoritmo implementado en esta práctica, emplearemos tres bases de datos:

- *Problema XOR*: esta base de datos representa el problema de clasificación no lineal del XOR. Se utilizará el mismo fichero para entrenamiento y para *test*. Como puede verse, se ha adaptado dicho fichero a la codificación 1-de- k , encontrándonos en este caso con dos salidas en lugar de una.
- *Base de datos iris*: *iris* contiene 112 patrones de entrenamiento y 38 patrones de test. Las clases son tres especies distintas de la flor *iris*, de manera que para cada flor se extraen cuatro medidas o variables de entrada (longitud y ancho de los pétalos y los sépalos, en cm). Las tres especies a distinguir son *iris setosa*, *iris virginica* e *iris versicolor*.
- *Base de datos digits*: esta base de datos está compuesta por 1274 patrones de entrenamiento y 319 patrones de test. Está formada por un conjunto de dígitos (del 0 al 9) escritos a mano por 80 personas distintas, y ajustados a una rejilla cuadrada de 16×16 píxeles. Aunque las imágenes originales estaban en escala de grises, éstas fueron binarizadas, con un valor de umbral fijo¹. Cada uno de los píxeles forman parte de las variables de entrada (con un total de $16 \times 16 = 256$ variables de entrada) y las clases se corresponden con el dígito escrito (0, 1, ..., 9, con un total de 10 clases). La figura 1 presenta los 1274 patrones del conjunto de entrenamiento, mientras que los dígitos del conjunto de test están en la figura 2. Además, todos los dígitos están colgados en la plataforma Moodle en los ficheros `train.digits.tar.gz` y `test.digits.tar.gz`.

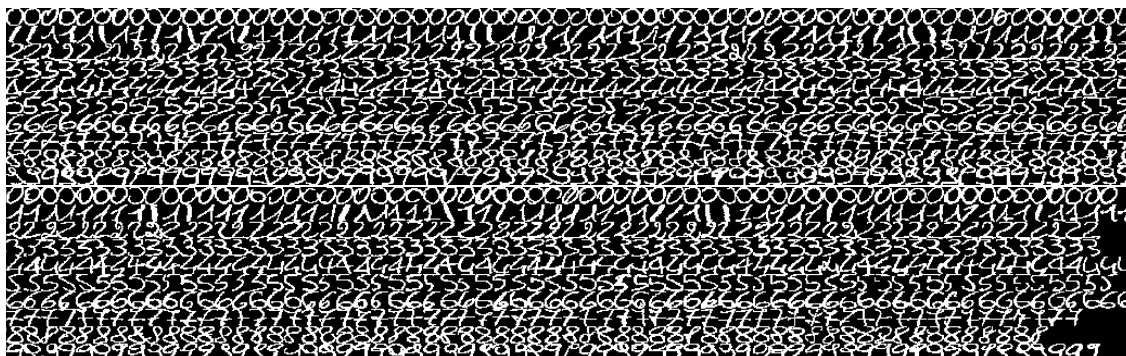


Figura 1: Dígitos del conjunto de entrenamiento.

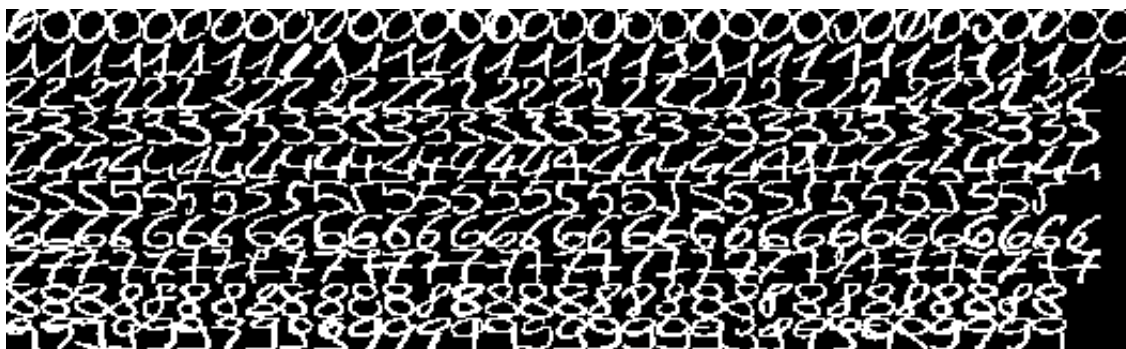


Figura 2: Dígitos del conjunto de test.

¹Para más información, consultar <http://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+Digit>

Se deberá construir **una tabla para cada base de datos**, en la que se compare la media y desviación típica de cuatro medidas:

- Error de entrenamiento y de *test*. Se utilizará la medida que se haya elegido para ajustar los pesos (*MSE* o entropía cruzada).
- *CCR* de entrenamiento y de *test*.

Se deberá utilizar siempre sesgo y factor de momento. Se deben probar, al menos, las siguientes configuraciones:

- *Arquitectura de la red*: Para esta primera prueba, utiliza la función de error entropía cruzada y la función de activación *softmax* en la capa de salida, con el algoritmo configurado como *off-line*.
 - Para el problema XOR utilizar la arquitectura que resultó mejor en la práctica anterior.
 - Para los problemas *digits* e *iris*, se deberán probar 8 arquitecturas (una o dos capas ocultas con 5, 10, 50 o 75 neuronas).
- Una vez decidida la mejor arquitectura, probar las siguientes combinaciones (con el algoritmo como *off-line*):
 - Función de error *MSE* y función de activación *sigmoide* en la capa de salida.
 - Función de error *MSE* y función de activación *softmax* en la capa de salida.
 - Función de error entropía cruzada y función de activación *softmax* en la capa de salida.
 - **NOTA**: no probar la combinación de entropía cruzada y función de activación *sigmoide*, ya que llevará a malos resultados (explicar por qué).
- Una vez decidida la mejor combinación de las anteriores, comparar los resultados con el algoritmo *on-line* frente a los obtenidos con el algoritmo *off-line*.

NOTA: Dependiendo de la función de error, puede ser necesario adaptar los valores de la tasa de aprendizaje (η) y del factor de momento (μ).

Como valor orientativo, se muestra a continuación el *CCR* de entrenamiento y de generalización obtenido por una regresión logística lineal utilizando Weka en las dos bases de datos:

- *Problema XOR*: $CCR_{\text{entrenamiento}} = CCR_{\text{test}} = 50 \%$.
- *Base de datos iris*: $CCR_{\text{entrenamiento}} = 83,0357 \%$; $CCR_{\text{test}} = 92,1053 \%$.
- *Base de datos digits*: $CCR_{\text{entrenamiento}} = 94,3485 \%$; $CCR_{\text{test}} = 73,0408 \%$.

El alumno debería ser capaz de superar estos porcentajes con algunas de las configuraciones y semillas.

3.1. Formato de los ficheros

Los ficheros que contienen las bases de datos tendrán el mismo formato que en la práctica anterior.

4. Entregables

Los ficheros a entregar serán los siguientes:

- Memoria de la práctica en un fichero pdf que describa el programa generado, incluya las tablas de resultados y analice estos resultados.
- Fichero ejecutable de la práctica y código fuente.

4.1. Memoria de la práctica

La memoria de la práctica deberá incluir, al menos, el siguiente contenido:

- Portada con el número de práctica, título de la práctica, asignatura, titulación, escuela, universidad, curso académico, nombre, DNI y correo electrónico del alumno.
- Índice del contenido de la memoria con numeración de las páginas.
- Descripción de las adaptaciones aplicadas a los modelos de red utilizados para tener en cuenta problemas de clasificación (**máximo 1 carilla**).
- Descripción en pseudocódigo de aquellas funciones que haya sido necesario modificar respecto a las implementaciones de la práctica anterior (**máximo 3 carillas**).
- Experimentos y análisis de resultados:
 - Breve descripción de las bases de datos utilizadas.
 - Breve descripción de los valores de los parámetros considerados.
 - Resultados obtenidos, según el formato especificado en la sección anterior.
 - Análisis de resultados. El análisis deberá estar orientado a justificar los resultados obtenidos, en lugar de realizar un análisis meramente descriptivo de las tablas. Tener en cuenta que esta parte es decisiva en la nota de la práctica. Se valorará la inclusión de los siguientes elementos de comparación:
 - Matriz de confusión en test del mejor modelo de red neuronal obtenido para la base de datos *digits*. Analizar los errores cometidos, **incluyendo las imágenes de aquellos dígitos en los que el modelo de red se equivoca**, para comprobar visualmente si son confusos.
 - Gráficas de convergencia: reflejan, en el eje x , el número de iteración del algoritmo y, en el eje y , el valor del *CCR* de entrenamiento y/o el valor del *CCR* de *test*.
- Referencias bibliográficas u otro tipo de material distinto del proporcionado en la asignatura que se haya consultado para realizar la práctica (en caso de haberlo hecho).

Aunque lo importante es el contenido, se valorará también la presentación, incluyendo formato, estilo y estructuración del documento. La presencia de demasiadas faltas ortográficas puede disminuir la nota obtenida.

4.2. Ejecutable y código fuente

Junto con la memoria, se deberá incluir el fichero ejecutable preparado para funcionar en las máquinas de la UCO (en concreto, probar por `ssh` en `ts.uco.es`). Además se incluirá todo el código fuente necesario. El fichero ejecutable deberá tener las siguientes características:

- Su nombre será `practica2`.
- El programa a desarrollar recibe once argumentos por la línea de comandos (que pueden aparecer en cualquier orden)². Los ocho primeros no han cambiado respecto a la práctica anterior. Los tres últimos incorporan las modificaciones realizadas en esta práctica:
 - Argumento `t`: Indica el nombre del fichero que contiene los datos de entrenamiento a utilizar. Sin este argumento, el programa no puede funcionar.
 - Argumento `T`: Indica el nombre del fichero que contiene los datos de *test* a utilizar. Si no se especifica este argumento, utilizar los datos de entrenamiento como *test*.

²Para procesar la secuencia de entrada, se recomienda utilizar la función `getopt()` de la librería `libc`

- Argumento *i*: Indica el número de iteraciones del bucle externo a realizar. Si no se especifica, utilizar 1000 iteraciones.
- Argumento *l*: Indica el número de capas ocultas del modelo de red neuronal. Si no se especifica, utilizar 1 capa oculta.
- Argumento *h*: Indica el número de neuronas a introducir en cada una de las capas ocultas. Si no se especifica, utilizar 5 neuronas.
- Argumento *e*: Indica el valor del parámetro *eta* (η). Por defecto, utilizar $\eta = 0,1$.
- Argumento *m*: Indica el valor del parámetro *mu* (μ). Por defecto, utilizar $\mu = 0,9$.
- Argumento *b*: Booleano que indica si se va a utilizar sesgo en las neuronas. Si no se especifica, no utilizaremos sesgo.
- Argumento *o*: Booleano que indica si se va a utilizar la versión *on-line*. Si no se especifica, utilizaremos la versión *off-line*.
- Argumento *f*: Indica la función de error que se va a utilizar durante el aprendizaje (0 para el error *MSE* y 1 para la entropía cruzada). Por defecto, utilizar el error *MSE*.
- Argumento *s*: Booleano que indica si vamos utilizar la función *softmax* en la capa de salida. Si no se especifica, utilizaremos la función sigmoide.

■ Un ejemplo de ejecución se puede ver en la siguiente salida:

```

1 i02gupep@NEWTS:~/imc/practica2/Debug$ ./practica2 -t ../train_xor.dat -T ../
  test_xor.dat -i 1000 -b -l 1 -h 10 -e 0.1 -m 0.9 -f 1 -s
2 Fichero de entrenamiento: ../train_xor.dat
3 Fichero de test: ../test_xor.dat
4 Número de iteraciones del bucle externo: 1000
5 Utilizamos sesgo
6 1 capas ocultas de 10 neuronas
7 Valor de eta: 0.1
8 Valor de mu: 0.9
9 Función de error: Entropía cruzada
10 Utilizamos softmax en la capa de salida
11 Algoritmo Off-line
12 *****
13 SEMILLA 10
14 *****
15 Iteración 1          Error de entrenamiento: 0.376804
16 Iteración 2          Error de entrenamiento: 0.373111
17 Iteración 3          Error de entrenamiento: 0.370409
18 ...
19 Iteración 1000       Error de entrenamiento: 0.00842754
20 PESOS DE LA RED
21 =====
22 Capa 1
23 -----
24 0.154169 0.038461 0.037247
25 -2.241612 2.056153 -2.219186
26 2.080109 2.267694 -2.104531
27 -2.080150 -1.910036 -2.118657
28 1.619104 1.433154 1.607444
29 0.403391 -1.320108 -0.420115
30 0.617576 0.516169 -0.096045
31 1.715708 1.898087 -1.704513
32 -2.512375 -2.350109 -2.536862
33 0.819229 -0.982442 -0.569460
34 Capa 2
35 -----
36 0.127535 2.441292 -2.448816 -2.338967 0.467336 1.358839 -0.494895 -1.763285
  -2.310780 0.285596 1.199232
37 -0.120020 -1.936437 2.652796 1.314135 -1.855409 -1.415333 0.374982 1.729195
  2.860525 -1.229694 0.726225
38 Salida Esperada Vs Salida Obtenida (test)
39 =====

```

```

40 | 1 -- 0.980786 \ 0 -- 0.0192142 \
41 | 0 -- 0.0143244 \ 1 -- 0.985676 \
42 | 1 -- 0.985325 \ 0 -- 0.0146748 \
43 | 0 -- 0.0186319 \ 1 -- 0.981368 \
44 | Finalizamos => CCR de test final: 100
45 | *****
46 | SEMILLA 20
47 | *****
48 | ...
49 | HEMOS TERMINADO TODAS LAS SEMILLAS
50 | INFORME FINAL
51 | *****
52 | Error de entrenamiento (Media +- DT): 0.00989727 +- 0.00136428
53 | Error de test (Media +- DT): 0.00989727 +- 0.00136428
54 | CCR de entrenamiento (Media +- DT): 100 +- 0
55 | CCR de test (Media +- DT): 100 +- 0
56 |
57 | i02gupep@NEWTS:~/imc/practica2/Debug$ ./practica2 -t ../train_iris.dat -T ../
    test_iris.dat -i 300 -b -l 1 -h 10 -e 0.7 -m 1 -f 0 -o -s
58 | Fichero de entrenamiento: ../train_iris.dat
59 | Fichero de test: ../test_iriss.dat
60 | Número de iteraciones del bucle externo: 300
61 | Utilizamos sesgo
62 | 1 capas ocultas de 10 neuronas
63 | Valor de eta: 0.70
64 | Valor de mu: 1.0
65 | Función de error: MSE
66 | Utilizamos softmax en la capa de salida
67 | Algoritmo On-line
68 | *****
69 | SEMILLA 10
70 | *****
71 | Iteración 1          Error de entrenamiento: 0.232661
72 | Iteración 2          Error de entrenamiento: 0.195423
73 | Iteración 3          Error de entrenamiento: 0.126582
74 | ...
75 | Iteración 300        Error de entrenamiento: 0.00600283
76 | ...
77 | HEMOS TERMINADO TODAS LAS SEMILLAS
78 | INFORME FINAL
79 | *****
80 | Error de entrenamiento (Media +- DT): 0.00600592 +- 8.85296e-06
81 | Error de test (Media +- DT): 0.0350755 +- 1.02903e-05
82 | CCR de entrenamiento (Media +- DT): 99.1071 +- 0
83 | CCR de test (Media +- DT): 94.7368 +- 0
84 |
85 | i02gupep@NEWTS:~/imc/practica2/Debug$ ./practica2 -t ../train_digits.dat -T ../
    test_digits.dat -i 300 -b -l 1 -h 10 -e 0.7 -m 1 -f 1 -s
86 | Fichero de entrenamiento: ../train_digits.dat
87 | Fichero de test: ../test_digits.dat
88 | Número de iteraciones del bucle externo: 300
89 | Utilizamos sesgo
90 | 1 capas ocultas de 10 neuronas
91 | Valor de eta: 0.70
92 | Valor de mu: 1.0
93 | Función de error: Entropía cruzada
94 | Utilizamos softmax en la capa de salida
95 | Algoritmo Off-line
96 | *****
97 | SEMILLA 10
98 | *****
99 | Iteración 1          Error de entrenamiento: 0.246855
100 | Iteración 2          Error de entrenamiento: 0.236523
101 | Iteración 3          Error de entrenamiento: 0.23092
102 | ...
103 | Iteración 300        Error de entrenamiento: 0.0229645
104 | ....

```

```
105 | Finalizamos => CCR de test final: 86.2069
106 | ...
107 | HEMOS TERMINADO TODAS LAS SEMILLAS
108 | INFORME FINAL
109 | *****
110 | Error de entrenamiento (Media +- DT): 0.0230395 +- 0.00416873
111 | Error de test (Media +- DT): 0.0497216 +- 0.00771004
112 | CCR de entrenamiento (Media +- DT): 94.8352 +- 1.54951
113 | CCR de test (Media +- DT): 84.8903 +- 2.64514
```