



ESCUELA POLITÉCNICA  
SUPERIOR DE CÓRDOBA  
Universidad de Córdoba

EP  
SC

UNIVERSIDAD DE CÓRDOBA  
ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA

INGENIERÍA INFORMÁTICA  
ESPECIALIDAD: COMPUTACIÓN  
CUARTO CURSO. PRIMER CUATRIMESTRE

INTRODUCCIÓN A LOS MODELOS  
COMPUTACIONALES.

## Práctica 2: Implementación del perceptrón multicapa para problemas de clasificación.

*Antonio Ariza García*

DNI

i62argaa@uco.es

Curso académico 2020-2021  
Córdoba, 2 de marzo de 2021

# Índice

Índice de figuras	II
Índice de tablas	II
Índice de algoritmos	III
1. Introduccion a los modelos de redes neuronales	1
2. Implementación	1
3. Resultados Prácticos	2
3.1. Descripción de los distintos parámetros . . . . .	3
3.2. Utilizando función error entropía cruzada y la función softmax en la capa de salida (modo offline) . . . . .	4
4. Pruebas modificando el Factor de decremento	6

## Índice de tablas

1.	Errores Base de datos XOR en distintas arquitecturas . . . . .	4
2.	Errores Base de datos Seno en distintas arquitecturas . . . . .	5
3.	Errores Base de datos Quake en distintas arquitecturas . . . . .	5
4.	Errores Base de datos Parkinsons en distintas arquitecturas . .	6
5.	Variaciones Factor decremento para base datos XOR . . . . .	6
6.	Variaciones Factor decremento para base datos Seno . . . . .	6
7.	Variaciones Factor decremento para base datos Quake . . . . .	7
8.	Variaciones Factor decremento para base datos Parkinsons . .	7

## List of Algorithms

1. weightAdjustment . . . . . 2

# 1. Introduccion a los modelos de redes neuronales

Esta práctica consiste en la adaptación de la práctica anterior. En concreto se dotará un significado probabilístico a dicho algoritmo con 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 offline del algoritmo.

## 2. Implementación

Resumen de las modificaciones a realizar.

- 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:

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

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

- Utilización de la función de error basada en entropía cruzada:

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

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

- Modo de funcionamiento: Además del funcionamiento online (práctica anterior), en esta práctica también podrá trabajar en modo offline. 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.

---

**Algorithm 1** weightAdjustment
 

---

```

1:  $\eta \leftarrow dEta$ 
2: for all capa  $i$  de 1 a  $H$  do do
3:   for all neurona  $j$  de la capa  $i$  do do
4:     for all neurona  $k$  de la capa  $i-1$  do do
5:        $w_{jk}^i \leftarrow w_{jk}^i - \eta * \Delta w_{jk}^i - \mu * \eta * ultimo \Delta w_{jk}^i$ 
6:     end for
7:    $w_{j0}^i \leftarrow w_{j0}^i - \eta * \Delta w_{j0}^i - \mu * \eta * ultimo \Delta w_{j0}^i$ 
8:   end for
9:    $\eta \leftarrow \eta dDecremento^{-(nofLayer-i)}$ 
10: end for
  
```

---

### 3. Resultados Prácticos

Probaremos distintas configuraciones de la red neuronal y ejecutaremos cada configuración con cinco semillas (1, 2, 3, 4 y 5). A partir de los resultados obtenidos, se obtendrá la media y la desviación típica del error, el entrenamiento va a guiarse utilizando la función de entropía cruzada o el MSE.

Para valorar como funciona el algoritmo implementado en esta práctica, emplearemos un total de 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 train y para test.
- Base de datos divorce: contiene 127 patrones de entrenamiento y 43 patrones de test. La base de datos contiene la respuesta a una serie de preguntas de un conjunto de encuestas en las que se pretende predecir si se va a producir un divorcio en la pareja. Las respuestas a las preguntas de la encuesta se proporcionan en escala de Likert con valores de 0 a

4. Todas las variables de entrada se consideran numéricas. La base de datos tiene un total de 54 preguntas (por lo tanto, 54 variables de entrada) y dos categorías (0 no hay divorcio, 1 hay divorcio).

- Base de datos noMNIST: la base de datos que se utilizará está compuesta por 900 patrones de entrenamiento y 300 patrones de test. Está formada por un conjunto de letras (de la a a la f) escritas con diferentes tipografías o simbologías.

Crearemos una tabla para cada base de datos, en la que se compare la media y desviación típica de cuatro medidas.

- Error entrenamiento y de test: Se utilizará la función que el usuario haya elegido para ajustar los pesos (MSE o entropía cruzada)..
- CCR de entrenamiento y de test.

### 3.1. Descripción de los distintos parámetros

A continuación se explicarán cada parámetro que podrá recibir el programa:

- 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.
- 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 4 neuronas.
- Argumento e: Indica el valor del parámetro eta ( $\eta$ ). Por defecto, utilizar  $\eta = 0,7$ .



- Argumento m: Indica el valor del parámetro  $\mu$ . Por defecto, utilizar  $\mu = 1$ .
- Argumento v: Indica el ratio de patrones de entrenamiento a utilizar como patrones de validación. Por defecto, utilizar  $v = 0;0$ .
- Argumento d: Indica el valor del factor de decremento a utilizar por cada una de las capas. Por defecto, utilizar  $F = 1$ . item Argumento o: Booleano que indica si se va a utilizar la versión online. Si no se especifica utilizaremos la versión offline.
- 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.
- Argumento w: Indica el nombre del fichero en el que se almacenarán la configuración y el valor de los pesos del modelo entrenado.

### 3.2. Utilizando función error entropía cruzada y la función softmax en la capa de salida (modo offline)

En esta prueba no emplearemos factor de decremento ( $F=1$ ) ni conjunto de validación ( $v= 0;0$ ). En la práctica anterior la mejor arquitectura fue con dos capas ocultas y 100 neuronas en cada una.

Arquitecturas	Train error	$\sigma(\text{Train error})$	Test error	$\sigma(\text{Test error})$
{n:100:100:k}	0.001701	0.0001286	0.001701	0.0001286

Tabla 1: Errores Base de datos XOR en distintas arquitecturas

Para la BD Seno la mejor arquitectura sería con dos capas ocultas de 64 neuronas cada una, porque aunque la formada por dos capas ocultas y 100 neuronas tenga un error muy parecido, no es así para la desviación típica, pues presenta 5 veces más por lo que puede dar errores mayores.

Arquitecturas	Train error	$\sigma$ (Train error)	Test error	$\sigma$ (Test error)
{n:2:k}	0.0297282	3.08475e-05	0.0365526	0.000360624
{n:4:k}	0.029529	0.000411965	0.0363783	0.000223319
{n:8:k}	0.0294088	0.000534317	0.03617	0.000366612
{n:32:k}	0.0290472	0.000360613	0.0360009	0.000806839
{n:64:k}	0.0280052	0.000322043	0.0352727	0.000584671
{n:100:k}	0.0285983	0.000986158	0.036288	0.000428383
{n:2:2:k}	0.0297237	4.24326e-05	0.036248	0.000170634
{n:4:4:k}	0.0297821	3.06686e-05	0.0362179	0.000150539
{n:8:8:k}	0.0299407	0.000118453	0.036485	0.000630194
{n:32:32:k}	0.0296178	0.00064876	0.0361732	0.000585751
<b>{n:64:64:k}</b>	<b>0.0269406</b>	<b>0.00080941</b>	<b>0.034039</b>	<b>0.000834419</b>
{n:100:100:k}	0.0259549	0.00480641	0.0340658	0.004816921

Tabla 2: Errores Base de datos Seno en distintas arquitecturas

En el caso de Quake tenemos 32 neuronas y con una sola capa oculta.

Arquitecturas	Train error	$\sigma$ (Train error)	Test error	$\sigma$ (Test error)
{n:2:k}	0.0301022	7.06987e-05	0.027198	5.85502e-05
{n:4:k}	0.029967	5.33303e-05	0.0270686	8.04283e-05
{n:8:k}	0.029916	9.88792e-05	0.027052	7.48808e-05
<b>{n:32:k}</b>	<b>0.0297653</b>	<b>2.91166e-05</b>	<b>0.026971</b>	<b>2.59938e-05</b>
{n:64:k}	0.0297662	3.47861e-05	0.0270388	3.04479e-05
{n:100:k}	0.0297446	5.52488e-05	0.0270129	3.55988e-05
{n:2:2:k}	0.030148	4.05299e-05	0.0272508	4.38807e-05
{n:4:4:k}	0.0301119	3.07848e-05	0.0272228	3.27004e-05
{n:8:8:k}	0.0300999	7.20564e-05	0.0271879	8.2355e-05
{n:32:32:k}	0.0298273	8.00825e-05	0.0269784	4.39127e-05
{n:64:64:k}	0.0295095	7.63962e-05	0.0270458	5.78271e-05
{n:100:100:k}	0.0294047	1.42461e-05	0.0271243	1.80462e-05

Tabla 3: Errores Base de datos Quake en distintas arquitecturas

Para BD Parkinsons la mejor arquitectura está compuesta por dos capas ocultas y 64 neuronas en cada una.

Arquitecturas	Train error	$\sigma(\text{Train error})$	Test error	$\sigma(\text{Test error})$
{n:2:k}	0.0343992	9.38514e-06	0.0371213	9.13403e-05
{n:4:k}	0.0257571	0.000242337	0.0255326	8.04128e-05
{n:8:k}	0.0207401	0.00149025	0.0216091	0.0015686
{n:32:k}	0.0154767	0.000518124	0.0173277	0.000980353
{n:64:k}	0.0147414	0.000632241	0.0164462	0.000540946
{n:100:k}	0.0133494	0.000380026	0.0155019	0.000628582
{n:2:2:k}	0.0315812	0.000894328	0.032942	0.000926471
{n:4:4:k}	0.0234318	0.0014714	0.0239546	0.00123368
{n:8:8:k}	0.0146101	0.00204368	0.0154939	0.00246286
{n:32:32:k}	0.0090358	0.00131826	0.0112496	0.00153763
<b>{n:64:64:k}</b>	<b>0.00777673</b>	<b>0.00121271</b>	<b>0.0101671</b>	<b>0.00159433</b>
{n:100:100:k}	0.00878881	0.00585147	0.010591	0.00506543

Tabla 4: Errores Base de datos Parkinsons en distintas arquitecturas

## 4. Pruebas modificando el Factor de decremento

Una vez elegida la mejor arquitectura para cada una de las bases de datos, probaremos modificando el factor de decremento con valores  $\{1,2\}$ .

Para la mejor arquitectura de la base de datos XOR.

Arquitecturas	F	Train error	$\sigma(\text{Train error})$	Test error	$\sigma(\text{Test error})$
{n:100:100:k}	1	0.0017019	0.00012862	0.0017019	0.00012862
{n:100:100:k}	2	0.0145757	0.00814214	0.0145757	0.00814214

Tabla 5: Variaciones Factor decremento para base datos XOR

Para la mejor arquitectura de la base de datos Seno.

Arquitecturas	F	Train error	$\sigma(\text{Train error})$	Test error	$\sigma(\text{Test error})$
{n:64:64:k}	1	0.0269406	0.00080941	0.0340395	0.00083442
{n:64:64:k}	2	0.0284589	0.00039619	0.0360283	0.00110174

Tabla 6: Variaciones Factor decremento para base datos Seno

Para la mejor arquitectura de la base de datos Quake.

Arquitecturas	F	Train error	$\sigma$ (Train error)	Test error	$\sigma$ (Test error)
{n:32:k}	1	0.0297653	2.91166e-05	0.026971	2.59938e-05
{n:32:k}	2	0.0298116	2.73964e-05	0.0269728	2.02602e-05

Tabla 7: Variaciones Factor decremento para base datos Quake

Para la mejor arquitectura de la base de datos Parkinson.

Arquitecturas	F	Train error	$\sigma$ (Train error)	Test error	$\sigma$ (Test error)
{n:64:64:k}	1	0.0077767	0.00121271	0.0101671	0.00159433
{n:64:64:k}	2	0.0130969	0.0015374	0.0130969	0.00153748

Tabla 8: Variaciones Factor decremento para base datos Parkinsons

Aquí podemos comprobar como el factor de decremento apenas cambia los resultados, pero empeora nuestro modelo en todas las bases de datos.