

# Introducción a los modelos computacionales

## Práctica 2. Perceptrón multicapa para problemas de clasificación

Pedro Antonio Gutiérrez  
pagutierrez@uco.es

Asignatura “Introducción a los modelos computacionales”  
4º Curso Grado en Ingeniería Informática  
Especialidad Computación  
Escuela Politécnica Superior  
(Universidad de Córdoba)

13 de octubre de 2020



- 1 Contenidos
- 2 Introducción
- 3 Consideraciones específicas



# Objetivos de la práctica

- Implementar la versión *off-line* del algoritmo de retropropagación básico para el perceptrón multicapa.
- Adaptar la formulación para problemas de clasificación mediante una interpretación probabilística de las salidas (función *softmax*).
- Utilizar una función de error probabilística para el entrenamiento de la red (entropía cruzada).
- Comprobar si estas modificaciones mejoran los resultados.



# Clasificación

- En la parte de teoría, hemos estudiado como adaptar el MLP a problemas de clasificación:
  - Representación de la etiqueta de clase usando codificación 1-de- $J$ .
  - Uso de múltiples neuronas en capa de salida y la función **softmax**.
  - Uso de la función de error **entropía cruzada** como alternativa al  $MSE$ , durante el entrenamiento.
  - Uso de la función de evaluación  $CCR$  para comprobar la bondad de los modelos.



# Resumen de modificaciones a realizar

- Debemos hacer que el programa saque información del *CCR*.
- Debemos incorporar la función *softmax* en la capa de salida, es decir, cambiar la forma en que se **propagan las entradas** (según la definición de la *softmax*) y la forma en que se **retropropaga el error** (según la nueva expresión de  $\delta_j^h$ ).
- Debemos incorporar la función de error  $L$  (entropía cruzada), haciendo que se calcule en las funciones que tienen que calcular un error y que se modifique la forma en que se **retropropaga el error** en los  $\delta_j^H$  (de la capa de salida).
- Debemos incorporar la versión *off-line* del algoritmo (práctica anterior).



# Cálculo de $\delta_j^h$

- Derivadas para neuronas de tipo sigmoide:

- Capa de salida:

- Error *MSE*:

$$\delta_j^H \leftarrow - (d_j - out_j^H) \cdot out_j^H \cdot (1 - out_j^H)$$

- Entropía cruzada:

$$\delta_j^H \leftarrow - (d_j / out_j^H) \cdot out_j^H \cdot (1 - out_j^H)$$

- Capas ocultas:

$$\delta_j^h \leftarrow \left( \sum_{i=1}^{n_{h+1}} w_{ij}^{h+1} \delta_i^{h+1} \right) \cdot out_j^h \cdot (1 - out_j^h)$$

- Derivadas para neuronas de tipo *softmax*:

- Capa de salida:

- Error *MSE*:

$$\delta_j^H \leftarrow - \sum_{i=1}^{n_H} ((d_i - out_i^H) \cdot out_j^H (I(i=j) - out_i^H))$$

- Entropía cruzada:

$$\delta_j^H \leftarrow - \sum_{i=1}^{n_H} ((d_i / out_i^H) \cdot out_j^H (I(i=j) - out_i^H))$$



# Ajuste de las derivadas para el modo *off-line*

- Cuando trabajamos en modo *off-line*, las derivadas se van acumulando para todos los patrones, haciendo que su magnitud pueda ser muy alta.
- Como el error usado es un error medio, deberíamos dividir el cambio realizado por el número de patrones de entrenamiento ( $N$ ).



# Ajuste de las derivadas para el modo *off-line*

ajustarPesosOffLine()

**Inicio**

① **Para**  $h$  de 1 a  $H$  // *Para cada capa ( $\Rightarrow \Rightarrow$ )*

① **Para**  $j$  de 1 a  $n_h$  // *Para cada neurona de la capa  $h$*

① **Para**  $i$  de 1 a  $n_{h-1}$  // *Para cada neurona de la capa  $h-1$*

$$w_{ji}^h \leftarrow w_{ji}^h - \frac{\eta \Delta w_{ji}^h}{N} - \frac{\mu (\eta \Delta w_{ji}^h (t-1))}{N}$$

**Fin Para**

②  $w_{j0}^h \leftarrow w_{j0}^h - \frac{\eta \Delta w_{j0}^h}{N} - \frac{\mu (\eta \Delta w_{j0}^h (t-1))}{N}$  // *Sesgo*

**Fin Para**

**Fin Para**

**Fin**





# Introducción a los modelos computacionales

## Práctica 2. Perceptrón multicapa para problemas de clasificación

Pedro Antonio Gutiérrez  
pagutierrez@uco.es

Asignatura “Introducción a los modelos computacionales”  
4º Curso Grado en Ingeniería Informática  
Especialidad Computación  
Escuela Politécnica Superior  
(Universidad de Córdoba)

13 de octubre de 2020

