Trabajo T5

# Ejercicio 1.1

## Algoritmo batch

Entrada: Topología, pesos iniciales θli j, 1 ≤ l ≤ L, 1 ≤ i ≤ Ml, 0 ≤ j ≤ Ml−1, factor de aprendizaje ρ, condiciones de convergencia, N datos de entrenamiento S, momentum 0 ≤ v < 1.

Salidas: Pesos de las conexiones que minimizan el error cuadrático medio de S.

Mientras no se cumplan las condiciones de convergencia

Para 1 ≤ l ≤ L, 1 ≤ i ≤ Ml, 0 ≤ j ≤ Ml−1, inicializar ∆θlij = 0

Para cada muestra de entrenamiento (x, t) ∈ S

Desde la capa de entrada a la de salida (l = 0, . . ., L):

Para 1 ≤ i ≤ Ml si l = 0 entonces s0i = xi sino calcular φli y sli = g(φli )

Desde la capa de salida a la de entrada (l = L, . . ., 1),

Para cada nodo (1 ≤ i ≤ Ml)

Calcular δli = si (l == L) then g’(φli ) (tni − sLi ) , else g’(φli ) ()

Para cada peso θlij (0 ≤ j ≤ Ml−1) calcular: ∆θlij = v∆θlij + ρ δli sl-1j

Para 1 ≤ l ≤ L, 1 ≤ i ≤ Ml , 0 ≤ j ≤ Ml−1, actualizar pesos: θlij = θlij + (1/N) ∆ θlij

## Algoritmo incremental

Entrada: Topología, pesos iniciales θli j, 1 ≤ l ≤ L, 1 ≤ i ≤ Ml, 0 ≤ j ≤ Ml−1, factor de aprendizaje ρ, condiciones de convergencia, N datos de entrenamiento S, momentum 0 ≤ v < 1.

Salidas: Pesos de las conexiones que minimizan el error cuadrático medio de S.

Mientras no se cumplan las condiciones de convergencia

Para 1 ≤ l ≤ L, 1 ≤ i ≤ Ml, 0 ≤ j ≤ Ml−1, inicializar ∆θlij = 0

Para cada muestra de entrenamiento (x, t) ∈ S

Desde la capa de entrada a la de salida (l = 0, . . ., L):

Para 1 ≤ i ≤ Ml si l = 0 entonces s0i = xi sino calcular φli y sli = g(φli )

Desde la capa de salida a la de entrada (l = L, . . ., 1),

Para cada nodo (1 ≤ i ≤ Ml)

Calcular δli = si (l == L) then g’(φli ) (tni − sLi ) , else g’(φli ) ()

Para cada peso θlij (0 ≤ j ≤ Ml−1) calcular: ∆θlij = ρ δli sl-1j

Para 1 ≤ l ≤ L, 1 ≤ i ≤ Ml , 0 ≤ j ≤ Ml−1, actualizar pesos: θlij = vθlij + (1/N) ∆ θlij

# Ejercicio 1.2

## Algoritmo batch

Entrada: Topología, pesos iniciales θli j, 1 ≤ l ≤ L, 1 ≤ i ≤ Ml, 0 ≤ j ≤ Ml−1, factor de aprendizaje ρ, condiciones de convergencia, N datos de entrenamiento S, factor de regularización λ.

Salidas: Pesos de las conexiones que minimizan el error cuadrático medio de S.

Mientras no se cumplan las condiciones de convergencia

Para 1 ≤ l ≤ L, 1 ≤ i ≤ Ml, 0 ≤ j ≤ Ml−1, inicializar ∆θlij = 0

Para cada muestra de entrenamiento (x, t) ∈ S

Desde la capa de entrada a la de salida (l = 0, . . ., L):

Para 1 ≤ i ≤ Ml si l = 0 entonces s0i = xi sino calcular φli y sli = g(φli )

Desde la capa de salida a la de entrada (l = L, . . ., 1),

Para cada nodo (1 ≤ i ≤ Ml)

Calcular δli = si (l == L) then g’(φli ) (tni − sLi ) , else g’(φli ) ()

Para cada peso θlij (0 ≤ j ≤ Ml−1) calcular: ∆θlij = ∆θlij - ρ λ δli sl-1j

Para 1 ≤ l ≤ L, 1 ≤ i ≤ Ml , 0 ≤ j ≤ Ml−1, actualizar pesos: θlij = θlij + (1/N) ∆ θlij

## Algoritmo incremental

Entrada: Topología, pesos iniciales θli j, 1 ≤ l ≤ L, 1 ≤ i ≤ Ml, 0 ≤ j ≤ Ml−1, factor de aprendizaje ρ, condiciones de convergencia, N datos de entrenamiento S, factor de regularización λ.

Salidas: Pesos de las conexiones que minimizan el error cuadrático medio de S.

Mientras no se cumplan las condiciones de convergencia

Para 1 ≤ l ≤ L, 1 ≤ i ≤ Ml, 0 ≤ j ≤ Ml−1, inicializar ∆θlij = 0

Para cada muestra de entrenamiento (x, t) ∈ S

Desde la capa de entrada a la de salida (l = 0, . . ., L):

Para 1 ≤ i ≤ Ml si l = 0 entonces s0i = xi sino calcular φli y sli = g(φli )

Desde la capa de salida a la de entrada (l = L, . . ., 1),

Para cada nodo (1 ≤ i ≤ Ml)

Calcular δli = si (l == L) then g’(φli ) (tni − sLi ) , else g’(φli ) ()

Para cada peso θlij (0 ≤ j ≤ Ml−1) calcular: ∆θlij = ρ δli sl-1j - p λ∆θlij

Para 1 ≤ l ≤ L, 1 ≤ i ≤ Ml , 0 ≤ j ≤ Ml−1, actualizar pesos: θlij = θlij + (1/N) ∆ θlij

# Ejercicio 2