APRENDIZAJE AUTOMÁTICO

Entregable T5: Redes Neuronales Multicapa

Victoria Beltrán Domínguez

vicbeldo@inf.upv.es

Grupo computación: 4CO11

Universitat Politécnica de Valencia

En este trabajo, se pide realizar los siguientes ejercicios:
1)Escribir el algoritmo BackProp (batch e incremental) con momentum.

ALGORITMO BACKPROP BATCH CON MOMENTUM

Entrada: Topología, pesos iniciales θ_{ij}^{l} (0), 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, factor de aprendizaje ρ, condiciones de convergencia, N datos de entrenamiento S, inercia o momentum 0 ≤ v < 1.

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

k=1

Mientras no se cumplan las condiciones de convergencia

Para 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, inicializar $\Delta\theta_{ij}^{l}(k) = 0$

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

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

Para 1 ≤ i ≤ M_l:

si l=0: $s_i^0 = x_i$

sino: Calcular ϕ_i^l y $s_i^l = g(\phi_i^l)$

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

Para cada nodo (1 ≤ i ≤ M_l) calcular:

si l == L: $\delta_i^l = g'(\phi_i^l)(t_{ni} - s_i^L)$

sino $\delta_i^l = g'(\phi_i^l)(\Sigma_r \delta_r^{l+1} \theta_{ri}^{l+1})$

para cada peso θ_{ij}^{l} (0 ≤ j ≤ M_{l-1}):

calcular: $\Delta\theta_{ij}^{l}(k) = \Delta\theta_{ij}^{l}(k) + \rho\, \delta_i^l s_j^{l-1}$

Para 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, actualizar pesos:

$$\Delta\theta_{ij}^{l}(k) = \frac{1}{N}\Delta\theta_{ij}^{l}(k) + v\Delta\theta_{ij}^{l}(k-1)$$

$$\theta_{ij}^{l} = \theta_{ij}^{l} + \Delta\theta_{ij}^{l}(k)$$

k=k+1

ALGORITMO BACKPROP INCREMENTAL CON MOMENTUM

Entrada: Topología, pesos iniciales $\theta^l_{ij}(0)$, 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, factor de aprendizaje ρ, condiciones de convergencia, N datos de entrenamiento S, inercia o momentum 0 ≤ v < 1.

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

k=1

Mientras no se cumplan las condiciones de convergencia

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

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

Para 1 ≤ i ≤ M_l:

si l=0: $s^0_i = x_i$

sino: Calcular ϕ^l_i y $s^l_i = g(\phi^l_i)$

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

Para cada nodo (1 ≤ i ≤ M_l) calcular:

si l == L: $\delta^l_i = g'(\phi^l_i)(t_{ni} - s^L_i)$

sino $\delta^l_i = g'(\phi^l_i)(\Sigma_r \delta^{l+1}_r \theta^{l+1}_{ri})$

para cada peso $\theta^l_{ij}(k)$ (0 ≤ j ≤ M_{l-1}) calcular: $\Delta\theta^l_{ij}(k)= \rho\ \delta^l_i s^{l-1}_j$

Para 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, actualizar pesos:

$$\Delta\theta^l_{ij}(k)= \frac{1}{N}\Delta\theta^l_{ij}(k)+ v\Delta\theta^l_{ij}(k-1)$$

$$\theta^l_{ij} = \theta^l_{ij}+ \Delta\theta^l_{ij}(k)$$

k=k+1

2)Escribir el algoritmo BackProp (batch e incremental) con amortiguamiento.

ALGORITMO BACKPROP BATCH CON AMORTIGUAMIENTO

Entrada: Topología, pesos iniciales $\theta_{ij}^{l}(0)$, 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-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.

k=1

Mientras no se cumplan las condiciones de convergencia

Para 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, inicializar $\Delta\theta_{ij}^{l}(k) = 0$

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

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

Para 1 ≤ i ≤ M_l:

si l=0: $s_i^0 = x_i$

sino: Calcular ϕ_i^l y $s_i^l = g(\phi_i^l)$

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

Para cada nodo (1 ≤ i ≤ M_l) calcular:

si l == L: $\delta_i^l = g'(\phi_i^l)(t_{ni} - s_i^L)$

sino $\delta_i^l = g'(\phi_i^l)(\Sigma_r \delta_r^{l+1}\theta_{ri}^{l+1})$

para cada peso θ_{ij}^{l} (k) (0 ≤ j ≤ M_{l-1})

calcular: $\Delta\theta_{ij}^{l}(k) = \Delta\theta_{ij}^{l}(k) + \rho\,\delta_i^l s_j^{l-1}$

Para 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, actualizar pesos:

$$\Delta\theta_{ij}^{l}(k) = \frac{1}{N}\Delta\theta_{ij}^{l}(k) + \rho\lambda\Delta\theta_{ij}^{l}(k-1)$$

$$\theta_{ij}^{l} = \theta_{ij}^{l} + \Delta\theta_{ij}^{l}(k)$$

k=k+1

ALGORITMO BACKPROP INCREMENTAL CON AMORTIGUAMIENTO

Entrada: Topología, pesos iniciales $\theta_{ij}^{l}(0)$, 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-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.

k=1

Mientras no se cumplan las condiciones de convergencia

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

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

Para 1 ≤ i ≤ M_l:

si l=0: $s_i^0 = x_i$

sino: Calcular ϕ_i^l y $s_i^l = g\left(\phi_i^l\right)$

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

Para cada nodo (1 ≤ i ≤ M_l) calcular:

si l == L: $\delta_i^l = g'\left(\phi_i^l\right)(t_{ni} - s_i^L)$

sino $\delta_i^l = g'\left(\phi_i^l\right)(\Sigma_r \delta_r^{l+1}\, \theta_{ri}^{l+1})$

para cada peso $\theta_{ij}^{l}(k)$ (0 ≤ j ≤ M_{l-1})

calcular: $\Delta\theta_{ij}^{l}(k) = \rho\ \delta_i^l s_j^{l-1}$

Para 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, actualizar pesos:

$$\Delta\theta_{ij}^{l}(k) = \frac{1}{N}\Delta\theta_{ij}^{l}(k) + \rho\lambda\Delta\theta_{ij}^{l}(k-1)$$

$$\theta_{ij}^{l} = \theta_{ij}^{l} + \Delta\theta_{ij}^{l}(k)$$

k=k+1

3)Escribir el algoritmo BackProp (batch e incremental) para clasificación.

Para escribir el algoritmo BackProp para clasificación, debemos sustituir la fórmula del error cuadrático por la de la entropía cruzada y derivar para obtener los nuevos parámetros. Las fórmulas de la entropía cruzada son :

$$\mathrm{q_S}(\Theta) = -\frac{1}{N}\sum_{n=1}^{N} q_n(\Theta) \qquad q_n(\Theta) = \sum_{i=1}^{M_L} t_{ni} \log s_i^L \ (x_n; \Theta)$$

Como queremos que la entropía cruzada sea mínima, aplicaremos descenso por gradiente para encontrar las ecuaciones para las capas ocultas y la capa de salida.

Entonces, dada una topología de un perceptrón multicapa con L capas y un conjunto de entrenamiento:

$$S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}, \qquad x_N \epsilon \mathbb{R}^{M_0}, t_n \epsilon \mathbb{R}^{M_L}$$

obtener Θ que minimice la entropía cruzada.

$$\Delta\theta_{ij}^l = -p\frac{\partial \mathrm{q_S}(\Theta)}{\partial \theta_{ij}^l} = \frac{1}{N}\sum_{n=1}^{N} -p\frac{\partial \mathrm{q_n}(\Theta)}{\partial \theta_{ij}^l} = \frac{1}{N}\sum_{n=1}^{N} \Delta_n\theta_{ij}^l$$

Así, debemos calcular para cada muestra:

$$\Delta_n\theta_{ij}^l = -p\frac{\partial \mathrm{q_n}(\Theta)}{\partial \theta_{ij}^l}, 1 \le \mathrm{i} \le M_L, 0 \le \mathrm{j} \le M_{l-1}, 1 \le \mathrm{l} \le L, 1 \le \mathrm{n} \le N$$

Para simplificar pero sin pérdida de generalidad, en lo que sigue se asume L=2.Para las ecuaciones de la capa de salida, la actualización de pesos θ_{ij}^2 para una muestra genérica:

$$q(\Theta) = q_n(\Theta) = \sum_{i=1}^{M_2} t_{ni} \log s_i^L \qquad s_i^2 = g(\phi_i^2)$$

$$\phi_i^2 = \sum_{m=0}^{M_1} \theta_{im}^2 s_m^1$$

Entonces, si empezamos a derivar:

$$\frac{\partial q}{\partial \theta_{ij}^2} = \frac{\partial q}{d s_i^2} \cdot \frac{d s_i^2}{d \phi_i^2} \cdot \frac{\partial \phi_i^2}{\partial \theta_{ij}^2} = \frac{t_{ni}}{s_i^2} g'(\phi_i^2) s_m^1$$

Así, para la capa de salida (en este caso l=2), las ecuaciones serán:

$$\delta_i^2 = \frac{t_{ni}}{s_i^2} g'(\phi_i^2)$$

$$\Delta_n \theta_{ij}^2 = p \delta_i^2 s_j^1,\ 1 \leq \mathrm{i} \leq M_2,\ 0 \leq \mathrm{j} \leq M_1$$

Para las ecuaciones de las capas ocultas, procederemos de la misma forma para calcular la actualización de pesos θ_{ij}^2 para una muestra genérica:

$$q(\Theta) = q_n(\Theta) = \sum_{i=1}^{M_2} t_{ni} \log s_i^2 \qquad s_i^2 = g(\phi_i^2)$$

$$\phi_i^2 = \sum_{m=0}^{M_1} \theta_{im}^2 s_m^1 \qquad s_m^1 = g(\phi_m^1) \qquad \phi_m^1 = \sum_{k=0}^{M_0} \theta_{km}^1 x_k$$

$$\frac{\partial q}{\partial \theta_{ij}^1} = \sum_{r=1}^{M_2} \frac{\partial q}{\partial s_r^2} \cdot \frac{\partial s_r^2}{\partial \theta_{ij}^1} = \sum_{r=1}^{M_2} \frac{\partial q}{\partial s_r^2} \cdot \frac{d s_r^2}{d \phi_r^2} \cdot \frac{\partial \phi_r^2}{\partial s_i^1} \cdot \frac{d s_i^1}{d \phi_i^1} \cdot \frac{\partial \phi_i^1}{\partial \theta_{ij}^1}$$

Resolviendo las derivadas obtenemos los siguientes resultados:

$$\sum_{r=1}^{M_2} \delta_r^2\, \theta_{ri}^2\, g'(\phi_i^1) x_j = g'(\phi_i^1) x_j \sum_{r=1}^{M_2} \delta_r^2\, \theta_{ri}^2$$

Entonces, el error en el nodo i de la capa oculta l se obtiene como:

$$\delta_i^l = g'(\phi_i^l) \sum_{r=1}^{M_l} \delta_r^{l+1}\, \theta_{ri}^{l+1}$$

Resultando así : $\Delta_n \theta_{ij}^l = p \delta_i^l x_j$

A partir de las ecuaciones obtenidas en el proceso, se procede a escribir el pseudocódigo de batch e incremental.

ALGORITMO BACKPROP BATCH PARA CLASIFICACIÓN

Entrada: Topología, pesos iniciales θ_{ij}^{l}, 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, factor de aprendizaje ρ, condiciones de convergencia, N datos de entrenamiento S.

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 ≤ M_l, 0 ≤ j ≤ M_{l-1}, inicializar $\Delta\theta_{ij}^{l} = 0$

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

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

Para 1 ≤ i ≤ M_l:

si l=0: $s_i^0 = x_i$

sino: Calcular ϕ_i^l y $s_i^l = g(\phi_i^l)$

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

Para cada nodo (1 ≤ i ≤ M_l) calcular:

si l == L: $\delta_i^l = g'(\phi_i^l)\,\dfrac{t_{ni}}{s_i^2}$

sino $\delta_i^l = g'(\phi_i^l)\sum_{r=1}^{M_l} \delta_r^{l+1}\,\theta_{ri}^{l+1}$

para cada peso θ_{ij}^{l} (0 ≤ j ≤ M_{l-1}) calcular: $\Delta\theta_{ij}^{l} = \Delta\theta_{ij}^{l} + \rho\,\delta_i^l s_j^{l-1}$

Para 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, actualizar pesos: $\theta_{ij}^{l} = \theta_{ij}^{l} + \frac{1}{N}\Delta\theta_{ij}^{l}$

ALGORITMO BACKPROP INCREMENTAL PARA CLASIFICACIÓN

Entrada: Topología, pesos iniciales θ_{ij}^{l}, 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, factor de aprendizaje ρ, condiciones de convergencia, N datos de entrenamiento S.

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

Mientras no se cumplan las condiciones de convergencia

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

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

Para 1 ≤ i ≤ M_l:

si l=0: $s_i^0 = x_i$

sino: Calcular ϕ_i^l y $s_i^l = g(\phi_i^l)$

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

Para cada nodo (1 ≤ i ≤ M_l) calcular:

si l == L: $\delta_i^l = g'(\phi_i^l)\ \frac{t_{ni}}{s_i^2}$

sino $\delta_i^l = g'(\phi_i^l) \sum_{r=1}^{M_l} \delta_r^{l+1}\ \theta_{ri}^{l+1}$

para cada peso θ_{ij}^l (0 ≤ j ≤ M_{l-1}) calcular: $\Delta\theta_{ij}^l = \rho\ \delta_i^l s_j^{l-1}$

Para 1 ≤ l ≤ L, 1 ≤ i ≤ M_l, 0 ≤ j ≤ M_{l-1}, actualizar pesos: $\theta_{ij}^l = \theta_{ij}^l + \frac{1}{N}\Delta\theta_{ij}^l$