Implementación del Algoritmo de Back-Propagation Para una Red Neuronal de Clasificación Multi-Clase

Carlos Santana Esplá y Ricardo Cárdenes Pérez

November 2023

1 Introduction

Supongamos que tenemos una red neuronal FC compuesta por l capas. Cada una de las capas, ocultas o de salida, computará una combinación lineal con las salidas de la capa anterior, las cuales, para la capa k, se denotan con el vector $A^{[k-1]}$. Para ello, cada neurona i de la capa k contará con un vector de pesos $w_i = (w_{1i} \cdots w_{n(k-1),i}) \in \mathbb{R}^{n(k-1)}$, donde $n(k-1)$ representa el número de neuronas de la capa anterior, y un sesgo o bias $b_i^{[k]}$. Estos, en conjunto, actuarán como los coeficientes en dicha operación. A su salida, se le aplicará una no linealidad que será general para toda la capa, evitando que la red colapse en una única combinación lineal compuesta de las entradas. Así, tenemos que la salida de la neurona en cuestión para la capa arbitraria escogida no es más que $A_i^{[k]}$, donde

$$z_i^{[k]} = w_i^{[k]} \cdot A^{[k-1]} + b_i^{[k]} \tag{1}$$

$$A_i^{[k]} = g_k\left(z_i^{[k]}\right) \tag{2}$$

Siendo $g_k : \mathbb{R} \to \mathbb{R}$ la función de activación definida para dicha capa y $z_i^{[k]}$ la combinación lineal descrita. La notación utilizada para los pesos, algo que cobrará gran importancia en las siguientes secciones, es $w_{ji}^{[k]}$, donde los índices i y j representan la neurona de entrada en la capa k, y la de salida en la capa $k-1$, respectivamente.

Podemos calcular así para una muestra cualquiera $x \in \mathbb{R}^n$ su salida en la red como

$$A^{[l]} = g_l(w^{[l]} \cdot (g_{l-1}(w^{[l-1]} \cdot (\cdots(g_1(w^{[1]}x^t + b^{[1]}))) + b^{[l-1]})) + b^{[l]}) \tag{3}$$

Hemos definido así el feedfoward de nuestra red. Ahora bien, ¿cómo podemos optimizar esta red para que la salida se ajuste correctamente a la realidad de los datos?

2 Optimización de la red neuronal

Usaremos la función de error de Cross-Entropy, la cual resulta ser de las mejores opciones a la hora de desarrollar redes de clasificación, aunque otras como MSE nos traen al mismo resultado ajustando adecuadamente las constantes implicadas. La función se define, para una muestra $X \in \mathbb{R}^n$ y su etiqueta $Y \in \{0,1\}^C$, como

$$\mathcal{L}(X,Y) = -\sum_{i=1}^{C} y_i \cdot log(A_i^{[l]}) \tag{4}$$

donde $A_i^{[l]}$ es la salida de la red para la $i-$ésima clase de la muestra X. Ante la posibilidad de trabajar en un espacio muestral con múltiples clases, usamos la función softmax como función de activación para la capa de salida. Esto es, sea $z^{[l]}$ el vector que contiene los cálculos lineales realizados en dicha capa, la salida de la misma se define como

$$A^{[l]} = \frac{e^{z^{[l]}}}{\sum_{j=1}^{C} e^{z_j^{[l]}}} \tag{5}$$

Sustituyendo este vector de salida en la función de coste dada en la ecuacion (4), obtenemos que

$$\mathcal{L}(X,Y) = -\sum_{i=1}^{C} y_i \cdot log\left(\frac{e^{z_i^{[l]}}}{\sum_{j=1}^{C} e^{z_j^{[l]}}}\right) \tag{6}$$

Y al aplicar las propiedades básicas del logaritmo

$$\mathcal{L}(X,Y) = -\sum_{i=1}^{C} y_i \left(log\left(e^{z_i^{[l]}}\right) - log\left(\sum_{j=1}^{C} e^{z_j^{[l]}}\right)\right) \tag{7}$$

El algoritmo de retropropagación (Back-Propagation) se fundamenta en el cálculo de las derivadas del error con respecto a los diferentes pesos de la red. Estas derivadas se utilizan en técnicas de descenso del gradiente para actualizar los pesos, uno por uno, epoch tras epoch (cada ciclo de entrenamiento). Por consiguiente, procederemos a calcular las derivadas correspondientes a los pesos de la capa de salida. Para lograr esto, aplicaremos la regla de la cadena. Empezamos derivando la función de error con respecto a $z_i^{[k]}$, donde lo primero será aplicar la regla de la derivada para la suma de funciones. Así

$$\frac{\partial \mathcal{L}}{\partial z_i^{[l]}} = -\frac{\partial}{\partial z_i^{[l]}}\left(y_i \cdot log\left(e^{z_i^{[l]}}\right)\right) + \sum_{k=1}^{C} \frac{\partial}{\partial z_i^{[l]}}\left(y_k \cdot log\left(\sum_{j=1}^{C} e^{z_j^{[l]}}\right)\right) \tag{8}$$

Veáse que en el primer término de la derivada nos hemos quedado únicamente con el término del sumatorio original que depende del parámetro respecto al cual estamos derivando. Centrándonos en dicho término, vemos que

$$-\frac{\partial}{\partial z_i^{[l]}}\left(y_i \cdot log\left(e^{z_i^{[l]}}\right)\right) = -\frac{\partial}{\partial z_i^{[l]}}\left(y_i z_i^{[l]}\right) = -y_i \tag{9}$$

Por otra parte,

$$\sum_{k=1}^{C}\frac{\partial}{\partial z_i^{[l]}}\left(y_k \cdot log\left(\sum_{j=1}^{C} e^{z_j^{[l]}}\right)\right) = \sum_{k=1}^{C} y_k \frac{e^{z_i^{[l]}}}{\sum_{j=1}^{C} e^{z_j^{[l]}}} = \frac{e^{z_i^{[l]}}}{\sum_{j=1}^{C} e^{z_j^{[l]}}}\sum_{k=1}^{C} y_k \tag{10}$$

Hemos de tener en cuenta que las etiquetas de las muestras vienen dadas en codificación one-hot, y por tanto $\sum_k y_k = 1$. Bastaría así con aplicar esta propiedad junto con la igualdad (5) para obtener que

$$\sum_{k=1}^{C}\frac{\partial}{\partial z_i^{[l]}}\left(y_k \cdot log\left(\sum_{j=1}^{C} e^{z_j^{[l]}}\right)\right) = A_i^{[l]} \tag{11}$$

Podemos ahora sustituir los resultados (9) y (11) en la expresión (8), de forma que

$$\frac{\partial \mathcal{L}}{\partial z_i^{[l]}} = A_i^{[l]} - y_i \tag{12}$$

Expresado en forma matricial,

$$\frac{\partial \mathcal{L}}{\partial z^{[l]}} = A^{[l]} - Y \tag{13}$$

Teniendo esto claro, uno podría calcular las derivadas del error con respecto a los pesos de la última capa con tan solo aplicar la regla de la cadena, donde se tiene que, por definición de $z^{[l]}$,

$$\frac{\partial \mathcal{L}}{\partial w_{ji}^{[l]}} = \frac{\partial \mathcal{L}}{\partial z_i^{[l]}}\frac{\partial z_i^{[l]}}{\partial w_{ji}^{[l]}} \tag{14}$$

Teniendo en cuenta la ecuación (1), vemos que para una capa $k \in \{1, ..., l\}$,

$$\frac{\partial z_i^{[k]}}{\partial w_{ji}^{[l]}} = \frac{\partial}{\partial w_{ji}^{[l]}}\left(w_i^{[k]} \cdot A^{[k-1]} + b^{[k]}\right) = \frac{\partial}{\partial w_{ji}^{[l]}}\left(w_{ji}^{[k]} A_j^{[k-1]}\right) = A_j^{[k-1]} \tag{15}$$

Concluimos así que para un peso $w_{ji}^{[k]}$, la deriva de la función de error con respecto a dicho peso no es más que

$$\frac{\partial \mathcal{L}}{\partial w_{ji}^{[l]}} = \frac{\partial \mathcal{L}}{\partial z_i^{[l]}} \frac{\partial z_i^{[l]}}{\partial w_{ji}^{[l]}} = \left(A_i^{[l]} - y_i\right) A_j^{[l-1]} \tag{16}$$

y, para toda capa $k \in \{1, ..., l\}$,

$$\frac{\partial \mathcal{L}}{\partial b_i^{[k]}} = \frac{\partial \mathcal{L}}{\partial z_i^{[k]}} \tag{17}$$

por lo que

$$\frac{\partial \mathcal{L}}{\partial b_i^{[l]}} = \left(A_i^{[l]} - y_i\right) \tag{18}$$

2.1 Derivadas para parámetros en capas ocultas

Veamos como se comportan las derivadas para la última capa oculta. Comenzaremos derivando con respecto a $z_i^{[l-1]}$ para luego aplicar la regla de la cadena y obtener la derivada respecto a cada uno de sus pesos. Teniendo en cuenta que estamos tratando de derivar la ecuación (6), que consiste en un sumatorio de C términos todos dependientes de $z_i^{[l-1]}$, por tratarse de una red FC, la derivada consistirá en un sumatorio, por la propiedad de la derivada de la suma de funciones, con las derivadas parciales con respecto a $z_i^{[l-1]}$ para cada uno de sus sumandos. Esto es,

$$\frac{\partial \mathcal{L}}{\partial z_i^{[l-1]}} = \sum_{n_l=1}^{C} \frac{\partial \mathcal{L}}{\partial A_{n_l}^{[l]}} \frac{\partial A_{n_l}^{[l]}}{\partial z_{n_l}^{[l]}} \frac{\partial z_{n_l}^{[l]}}{\partial A_i^{[l-1]}} \frac{\partial A_i^{[l-1]}}{\partial z_i^{[l-1]}} \tag{19}$$

Para simplificar los cálculos, supondremos que todas las capas ocultas de la red utilizan una función de activación sigmoide, $\sigma : \mathbb{R} \to \mathbb{R}$, la cual es derivable $\forall x \in \mathbb{R}$ y cumple que $\sigma'(x) = \sigma(x)(1 - \sigma(x))$. Gracias a esta suposición, sea $k \in \{1, ..., l-1\}$, sabemos que

$$A_i^{[k]} = \sigma(z_i^{[k]}) \tag{20}$$

Y, consecuentemente,

$$\frac{\partial A_i^{[k]}}{\partial z_i^{[k]}} = \sigma(z_i^{[k]}) \left(1 - \sigma(z_i^{[k]})\right) = A_i^{[k]} \left(1 - A_i^{[k]}\right) \tag{21}$$

Por tanto, al sustituir los resultados obtenidos en las ecuaciones (12) y (21) en la expresión (16), y observar que la derivada parcial de $z_{n_l}^{[l]}$ con respecto a $A_i^{[l-1]}$ no es más que su coeficiente $w_{i,n_l}^{[l]}$, obtenemos la siguiente expresión

$$\frac{\partial \mathcal{L}}{\partial z_i^{[l-1]}} = \sum_{n_l=1}^{C} \left(A_{n_l}^{[l]} - y_{n_l}\right) w_{i,n_l}^{[l]} A_i^{[l-1]} \left(1 - A_i^{[l-1]}\right) \tag{22}$$

Si ahora queremos la derivada con respecto a los pesos de una neurona i de la capa $l-1$, obtenemos, aplicando (14), que

$$\frac{\partial \mathcal{L}}{\partial w_{ji}^{[l-1]}} = \sum_{n_l=1}^{C} \left(A_{n_l}^{[l]} - y_{n_l}\right) w_{i,n_l}^{[l]} A_i^{[l-1]} \left(1 - A_i^{[l-1]}\right) A_j^{[l-2]} \tag{23}$$

El conjunto MNIST puede ser clasificado con un alto accuracy con una única capa oculta, por lo que ya tendríamos los cálculos necesarios para desarrollar el algoritmo. No obstante, para generalizar este algoritmo a cualquier red FC, necesitamos saber que ocurre con los pesos más allá de la penúltima capa.

Veamos así que sucede para la capa $l-2$. Para ello, hemos de tener en cuenta que, dado un peso $w_{ji}^{[l-2]}$, su valor únicamente pondera a la salida de la neurona j de la capa $l-3$, como entrada a la neurona i de la capa $l-2$. Ahora bien, al tratarse de una red neuronal FC, todas las neuronas de la capa $l-1$ estarán conectadas con todas las de la capa $l-2$. Por tanto, el peso $w_{ji}^{[l-2]}$ afectará a todas las salidas de esta $l-1$. Es por esta razón por la que aparece un nuevo sumatorio en la expresión, ya que todas las $z_i^{[l]}$ consisten en una suma ponderada de las distintas coordenadas del vector $A^{[l-1]}$, las cuales dependen todas de dicho peso y, por tanto, actúara la regla de la derivada para la suma de funciones sobre ellas. Esto es,

$$\frac{\partial}{\partial w_{ji}^{[l-2]}} \left(w_i^{[l]} \cdot A^{[l-1]}\right) = \sum_{n_{l-1}=1}^{n(l-1)} \frac{\partial}{\partial w_{ji}^{[l-2]}} \left(w_{n_{l-1},i}^{[l]} A_{n_{l-1}}^{[l-1]}\right) \tag{24}$$

Así, cobra más sentido el siguiente resultado

$$\frac{\partial \mathcal{L}}{\partial z_i^{[l-2]}} = \sum_{n_l=1}^{C} \frac{\partial \mathcal{L}}{\partial A_{n_l}^{[l]}} \frac{\partial A_{n_l}^{[l]}}{\partial z_{n_l}^{[l]}} \sum_{n_{l-1}=1}^{n(l-1)} \frac{\partial z_{n_l}^{[l]}}{\partial A_{n_{l-1}}^{[l-1]}} \frac{\partial A_{n_{l-1}}^{[l-1]}}{\partial z_{n_{l-1}}^{[l-1]}} \frac{\partial z_{n_{l-1}}^{[l-1]}}{\partial A_i^{[l-2]}} \frac{\partial A_i^{[l-2]}}{\partial z_i^{[l-2]}} \tag{25}$$

pues no es más que

$$\frac{\partial \mathcal{L}}{\partial z_i^{[l-2]}} = \sum_{n_l=1}^{C} \frac{\partial \mathcal{L}}{\partial A_{n_l}^{[l]}} \frac{\partial A_{n_l}^{[l]}}{\partial z_{n_l}^{[l]}} \sum_{n_{l-1}=1}^{n(l-1)} \frac{\partial}{\partial w_{ji}^{[l-2]}} \left(w_{n_{l-1},i}^{[l]} A_{n_{l-1}}^{[l-1]}\right) \tag{26}$$

2.2 Generalización de las derivadas

Uno podría seguir calculando las derivadas para capas inferiores, y observaría un patrón que se repite. Existirían $l-k$ sumatorios anidados en la expresión, uno por cada capa comprendida entre la capa k que se esté evaluando y la de salida, incluyendo a esta. Los $l-k-1$ sumatorios correspondientes a las capas $\{l, l-1, ..., k+2\}$ computarían la derivada del cálculo lineal que hace cada una de las neuronas de dicha capa con respecto a la salida de la capa anterior, multiplicado por su sumatorio. Esto, en conjunto, supone la derivada de la

función de error con respecto al cálculo lineal realizado en la capa $k+1$, tras aplicar sucesivas veces la regla de la cadena. Es decir,

$$S_{l:(k+2)} = \sum_{j=1}^{C} \frac{\partial \mathcal{L}}{\partial A_j^{[l]}} \frac{\partial A_j^{[l]}}{\partial z_j^{[l]}} \sum_{n_1=1}^{n(l-1)} \frac{\partial z_j^{[l]}}{\partial A_{n_1}^{[l-1]}} \frac{\partial A_{n_1}^{[l-1]}}{\partial z_{n_1}^{[l-1]}} \cdots \sum_{n_{k+2}=1}^{n(k+2)} \frac{\partial z_{n_{k+3}}^{[k+3]}}{\partial A_{n_{k+2}}^{[k+2]}} \frac{\partial A_{n_{k+2}}^{[k+2]}}{\partial z_{n_{k+2}}^{[k+2]}} \tag{27}$$

El sumatorio de la capa $k+1$ calcula la derivada de cada $z_{n_{k+2}}^{[k+2]}$ con respecto al $z_i^{[k]}$ que utiliza el peso $w_{ji}^{[k]}$ que nos intersea, que se reduce a lo siguiente

$$\frac{\partial \mathcal{L}}{\partial z_i^{[k]}} = S_{l:(k+2)} \cdot \sum_{n_{k+1}=1}^{n(k+1)} \frac{\partial z_{n_{k+2}}^{[k+2]}}{\partial A_{n_{k+1}}^{[k+1]}} \frac{\partial A_{n_{k+1}}^{[k+1]}}{\partial z_{n_{k+1}}^{[k+1]}} \frac{\partial z_{n_{k+1}}^{[k+1]}}{\partial A_i^{[k]}} \frac{\partial A_i^{[k]}}{\partial z_i^{[k]}} \tag{28}$$

Finalmente, sea la k la capa en la que se encuentra un peso respecto al cual queremos calcular la derivada, tenemos que

$$\frac{\partial \mathcal{L}}{\partial z_i^{[k]}} = \sum_{j=1}^{C} \frac{\partial \mathcal{L}}{\partial A_j^{[l]}} \frac{\partial A_j^{[l]}}{\partial z_j^{[l]}} \sum_{n_1=1}^{n(l-1)} \left(\cdots \sum_{n_{k+1}=1}^{n(k+1)} \frac{\partial z_{n_{k+2}}^{[k+2]}}{\partial A_{n_{k+1}}^{[k+1]}} \frac{\partial A_{n_{k+1}}^{[k+1]}}{\partial z_{n_{k+1}}^{[k+1]}} \frac{\partial z_{n_{k+1}}^{[k+1]}}{\partial A_i^{[k]}} \frac{\partial A_i^{[k]}}{\partial z_i^{[k]}} \right) \tag{29}$$

Y, consecuentemente, la derivada del error con respecto a dicho peso no es más que

$$\frac{\partial \mathcal{L}}{\partial w_{ji}^{[k]}} = \sum_{j=1}^{C} \frac{\partial \mathcal{L}}{\partial A_j^{[l]}} \frac{\partial A_j^{[l]}}{\partial z_j^{[l]}} \sum_{n_1=1}^{n(l-1)} \left(\cdots \sum_{n_{k+1}=1}^{n(k+1)} \frac{\partial z_{n_{k+2}}^{[k+2]}}{\partial A_{n_{k+1}}^{[k+1]}} \frac{\partial A_{n_{k+1}}^{[k+1]}}{\partial z_{n_{k+1}}^{[k+1]}} \frac{\partial z_{n_{k+1}}^{[k+1]}}{\partial A_i^{[k]}} \frac{\partial A_i^{[k]}}{\partial z_i^{[k]}} A_j^{[k-1]} \right) \tag{30}$$

3 Notación matricial para el Back-Propagation

A la hora de implementar estas derivadas para que puedan calcularse de forma sencilla en un algoritmo de aprendizaje, se hace impresindible buscar una manera de emcapsular los cálculos en alguna estructura que nos permita reducir la notación planteada. Para los pesos de la primera capa, por ejemplo, cuyas derivadas vienen expresadas de forma genérica en la expresión (16), uno tenría que tener en cuenta las ixj combinaciones de subíndices posibles, a fin de calcular todas las derivadas necesarias para la corrección de errores en la capa de salida. Esto podría resolverse cómodamente haciendo uso del producto de matrices.

Por ser la matriz de pesos de la capa de salida, $W^{[l]}$, de dimensión $n(l)$ x $n(l-1)$, tenemos que $\frac{\partial \mathcal{L}}{\partial W^{[l]}}$, que no es más que la derivada parcial del error con respecto a dicha matriz, es también una matriz de orden $n(l)$ x $n(l-1)$. En

ella, cada entrada d_{ji} nos indica la derivada del error con respecto al peso $w_{ji}^{[l]}$.

$$\frac{\partial \mathcal{L}}{\partial W^{[l]}} = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial w_{11}^{[l]}} & \frac{\partial \mathcal{L}}{\partial w_{12}^{[l]}} & \cdots & \frac{\partial \mathcal{L}}{\partial w_{1,n(l-1)}^{[l]}} \\ \frac{\partial \mathcal{L}}{\partial w_{21}^{[l]}} & \frac{\partial \mathcal{L}}{\partial w_{22}^{[l]}} & \cdots & \frac{\partial \mathcal{L}}{\partial w_{2,n(l-1)}^{[l]}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{L}}{\partial w_{n(l),1}^{[l]}} & \frac{\partial \mathcal{L}}{\partial w_{n(l),2}^{[l]}} & \cdots & \frac{\partial \mathcal{L}}{\partial w_{n(l),n(l-1)}^{[l]}} \end{pmatrix} \tag{31}$$

Aplicando las propiedades de matrices, y apoyándonos en la ecuación (16), podemos expresar dicha derivada en la siguiente forma matricial

$$\frac{\partial \mathcal{L}}{\partial W^{[l]}} = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial z_1^{[l]}} \\ \frac{\partial \mathcal{L}}{\partial z_2^{[l]}} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial z_{n(l)}^{[l]}} \end{pmatrix} \cdot \begin{pmatrix} A_1^{[l-1]} & A_2^{[l-1]} & \cdots & A_{n(l-1)}^{[l-1]} \end{pmatrix} = \frac{\partial \mathcal{L}}{\partial z^{[l]}} \cdot (A^{[l-1]})^t \tag{32}$$

Hemos conseguido no solo reducir la notación, sino simplificar la implementación del cálculo, lo cual se hace imprescindible para capas más profundas. Veamos que ocurre para la penúltima capa del modelo. Recordemos la expresión de la derivada del error respecto a sus pesos, dada en la ecuación (23)

$$\frac{\partial \mathcal{L}}{\partial w_{ji}^{[l-1]}} = \sum_{n_l=1}^{C} \left(A_{n_l}^{[l]} - y_{n_l} \right) w_{i,n_l}^{[l]} A_i^{[l-1]} \left(1 - A_i^{[l-1]} \right) A_j^{[l-2]}$$

Podemos reordenar el segundo miembro, extrayendo del sumatorio aquellos términos que no dependen de su índice. Así,

$$\frac{\partial \mathcal{L}}{\partial w_{ji}^{[l-1]}} = A_i^{[l-1]} \left(1 - A_i^{[l-1]} \right) A_j^{[l-2]} \sum_{n_l=1}^{C} \left(A_{n_l}^{[l]} - y_{n_l} \right) w_{i,n_l}^{[l]} \tag{33}$$

Y nos bastaría con hacer uso del producto matricial para llegar al siguiente resultado

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_{ji}^{[l-1]}} &= A_i^{[l-1]} \left(1 - A_i^{[l-1]} \right) A_j^{[l-2]} w_i^{[l]} \left(A^{[l]} - y \right) \\ &= A_i^{[l-1]} \left(1 - A_i^{[l-1]} \right) w_i^{[l]} \left(A^{[l]} - y \right) A_j^{[l-2]} \end{aligned} \tag{34}$$

Aplicando la misma estrategia seguida para los pesos de la primera capa, obtenemos la siguiente forma matricial para la derivada con respecto a los pesos de la capa $l-1$

$$\frac{\partial \mathcal{L}}{\partial W^{[l-1]}} = \left(\left(A^{[l-1]} \odot \left(1 - A^{[l-1]} \right) \right) \odot \left(W^{[l]} \right) \left(A^{[l]} - y \right) \right) \left(A^{[l-2]} \right)^t \tag{35}$$

y

$$\frac{\partial \mathcal{L}}{\partial b^{[l-1]}} = \left(\left(A^{[l-1]} \odot \left(1 - A^{[l-1]} \right) \right) \odot \left(W^{[l]} \right) \left(A^{[l]} - y \right) \right) \tag{36}$$