UMBRAL CON FUNCIONES DE SALIDA

1. Función o regla de propagación:

$$Net_j : \mathbb{R} \longrightarrow \mathbb{R} \qquad \text{Campo escalar}$$

$$(w_1, w_2, \ldots, w_n) \longrightarrow Net_j\,(w_1, w_2, \ldots, w_n) = \sum_j y_i w_{ij}$$

2. Función o regla de activación

Esta función que llamaremos $g : \mathbb{R} \longrightarrow \mathbb{R}$ produce un nuevo estado de activación en una neurona a partir del estado (a_i) que existía y la combinación de las entradas con los pesos de las conexiones (Net_j).

Nota: En la mayoría de los casos, g es la función identidad, por lo que el estado de activación de una neurona en $t+1$ coincidirá con el **Net** de la misma en t. En este caso, el parámetro que se le pasa a la función de salida f de la neurona será directamente el Net. **El estado de activación anterior no se tiene en cuenta.**

$$f : \mathbb{R} \longrightarrow \mathbb{R}$$

> Consideramos únicamente la función f, que denominaremos indistintamente 'de transferencia', 'de salida' o 'de activación'.

Luis Fernando Alvarez V. Leonardo Alvarez V. Hernando Alvarez R.
lfalvare@utp.edu.co lalvarez@utp.edu.co healvare@utp.edu.co

3. Funciones de salida basadas en umbral

A continuación se presentan algunas funciones de salida clásicas que incorporan explícitamente el umbral θ_i.

3.1. Función escalón binaria con valores en $\{0, 1\}$

En este caso, la neurona adopta únicamente dos estados posibles: apagada (0) o encendida (1). La actualización se define como

$$y_i(t+1) = \begin{cases} 1, & \text{si } Net_i > \theta_i, \\ y_i(t), & \text{si } Net_i = \theta_i, \\ 0, & \text{si } Net_i < \theta_i. \end{cases}$$

Cuando la entrada neta supera el umbral θ_i, la neurona se activa (salida 1); si la entrada neta está por debajo del umbral, la neurona permanece inactiva (salida 0). En el caso límite $Net_i = \theta_i$, se mantiene el estado anterior $y_i(t)$.

3.2. Función escalón bipolar con valores en $\{-1, 1\}$

En algunos modelos se prefiere que la neurona tenga dos estados simétricos: uno positivo y otro negativo. La regla de actualización queda entonces

$$y_i(t+1) = \begin{cases} +1, & \text{si } Net_i > \theta_i, \\ y_i(t), & \text{si } Net_i = \theta_i, \\ -1, & \text{si } Net_i < \theta_i. \end{cases}$$

Esta versión es útil cuando se desea distinguir entre una respuesta «excitadora» (+1) y una «inhibidora» (−1) alrededor del umbral.

3.3. Función lineal a trozos (lineal saturada)

Otra opción consiste en hacer que la salida sea lineal en un intervalo central y se *sature* a valores máximos y mínimos prefijados. Sean $b < B$ dos constantes que fijan esos límites. La función de salida se puede describir como

$$y_i(t+1) = \begin{cases} b, & \text{si } Net_i \leq b + \theta_i, \\ Net_i - \theta_i, & \text{si } b + \theta_i < Net_i < B + \theta_i, \\ B, & \text{si } Net_i \geq B + \theta_i. \end{cases}$$

Así, cuando la entrada neta es muy pequeña, la salida se mantiene en b; cuando la entrada neta es muy grande, la salida se fija en B; y entre esos valores la salida crece de manera lineal respecto a Net_i desplazada por el umbral θ_i.

3.4. Función sigmoidal logística con umbral

Una función de salida muy habitual en redes neuronales clásicas es la sigmoidal logística desplazada por el umbral θ_i:

$$y_i(t+1) = \frac{1}{1 + e^{-(Net_i - \theta_i)}}.$$

En este caso, la salida de la neurona toma valores continuos en el intervalo $(0, 1)$. El umbral θ_i desplaza horizontalmente la curva sigmoidal: al aumentar θ_i, se requiere mayor entrada neta Net_i para obtener el mismo nivel de salida.

3.5. Funciones de salida comúnmente utilizadas en redes neuronales

Además de las funciones anteriores, en la práctica de la inteligencia artificial moderna se emplean con frecuencia las siguientes funciones de salida (a veces sin escribir explícitamente el umbral, que se incorpora como un peso adicional o un sesgo):

Función sigmoidal logística. Es la misma que la descrita antes,

$$f(z) = \frac{1}{1 + e^{-z}},$$

y se utiliza cuando se quieren salidas en $(0, 1)$, por ejemplo en problemas de clasificación binaria. El umbral puede incorporarse escribiendo $z = Net_i - \theta_i$.

Función tangente hiperbólica. En lugar de limitarse a $(0, 1)$, la función tangente hiperbólica produce valores en $(-1, 1)$:

$$f(z) = \tanh(z).$$

De nuevo, puede considerarse $z = Net_i - \theta_i$ para interpretar explícitamente el umbral. Esta función es útil cuando se prefieren salidas centradas en cero.

Función ReLU (Rectified Linear Unit). En redes profundas es habitual usar la función ReLU, que anula las entradas negativas y mantiene las positivas:

$$f(z) = \max\{0, z\}.$$

Si se introduce un umbral θ_i, puede escribirse $f(Net_i - \theta_i) = \max\{0, Net_i - \theta_i\}$, de modo que la neurona solo se activa cuando la entrada neta supera el umbral.

Función softmax. Cuando se manejan varias neuronas de salida en un problema de clasificación multiclase, se utiliza con frecuencia la función softmax. Si $Net_1, Net_2, \ldots, Net_K$ son las entradas netas de K neuronas de salida, la función softmax define

$$y_k = \frac{e^{Net_k}}{\displaystyle\sum_{j=1}^{K} e^{Net_j}}, \qquad k = 1, \ldots, K.$$

Cada y_k puede interpretarse como una probabilidad asociada a la clase k. En este caso el umbral puede incorporarse como parte de las entradas netas Net_k (por ejemplo, $Net_k = \sum_j y_j w_{jk} - \theta_k$).

En resumen, el umbral θ_i desplaza el argumento de la función de salida y determina a partir de qué valor de la entrada neta la neurona modifica su comportamiento. La elección concreta de la función de salida (escalón, lineal a trozos, sigmoidal, tanh, ReLU, softmax, etc.) depende del tipo de problema a resolver y de las propiedades deseadas para la red neuronal.