

Redes Neuronales

Nicolás Kossacoff

Noviembre 2024

1. Introducción

Los perceptrones simples, como hemos explorado previamente, presentan limitaciones al abordar problemas no lineales. Como consecuencia de estas limitaciones surgen las redes neuronales, las cuales combinan múltiples neuronas con el objetivo de construir fronteras de decisión capaces de capturar patrones o relaciones no lineales.

En este artículo vamos a estudiar las redes neuronales, definidas como un conjunto de perceptrones (neuronas) interconectados. Veremos cómo estas redes son capaces de capturar relaciones no lineales, superando las limitaciones de los perceptrones simples.

Terminología. *Las redes neuronales se suelen organizar por **capas**. Los valores de entradas (usualmente se los llama **unidades de entrada**) se encuentran en la **capa de entrada**, las neuronas (también se las llama **unidades ocultas**) se encuentran en las **capas ocultas** y los resultados se encuentran en la **capa de salida**.*

*Las redes neuronales también se suelen llamar **perceptrones multi-capas (MLP)**. Además, las redes neuronales con una sola capa oculta se suelen llamar **redes neuronales superficiales**, mientras que las redes neuronales con dos o más capas ocultas se suelen llamar **redes neuronales profundas**.*

*Las redes neuronales que se ejecutan desde la izquierda a la derecha, y no de manera cíclica, se conocen como redes **feed-forward**. Por último, si los elementos de una capa se conecta con todos los elementos de otra capa, las redes neuronales son **fully-connected** (completamente conectadas).*

2. Redes Neuronales Superficiales

2.1. Definición

Sea $X \in \mathbb{R}^p$, es decir, un vector con p features. Si nuestra red neuronal tiene N neuronas en su capa oculta, entonces los pesos asociados a los valores de entrada se

encuentran en la siguiente matriz de dimensión $p \times N$:

$$W = \begin{pmatrix} w_{11} & \dots & w_{1N} \\ \vdots & \ddots & \vdots \\ w_{p1} & \dots & w_{pN} \end{pmatrix}$$

Si realizamos el producto matricial entre X y W obtenemos:

$$X \cdot W = \left(\sum_{i=1}^p w_{i1}x_i, \dots, \sum_{i=1}^p w_{iN}x_i \right)_{1 \times N}$$

donde cada elemento es el producto interno que calculan las neuronas antes de aplicar la función de activación. Si a cada elemento le aplicamos la función de activación $h(\bullet)$ obtenemos el siguiente vector:

$$Z = \left(h_1 \left(\sum_{i=1}^p w_{i1}x_i \right), \dots, h_N \left(\sum_{i=1}^p w_{iN}x_i \right) \right) = (z_1, \dots, z_N)$$

donde z_j es la salida de la j -ésima neurona oculta.

Ahora, si tenemos M neuronas de salida, la matriz de pesos asociada, W' , tiene dimensión $N \times M$. El producto entre esta matriz y Z nos devuelve el siguiente vector:

$$Z \cdot W' = \left(\sum_{i=1}^N w'_{i1}z_i, \dots, \sum_{i=1}^N w'_{iM}z_i \right)_{1 \times M}$$

Finalmente, aplicamos una función de activación de salida $f(\bullet)$, lo que nos devuelve:

$$Y = \left(f \left(\sum_{i=1}^N w'_{i1}h_i \right), \dots, f \left(\sum_{i=1}^N w'_{iM}h_i \right) \right) = (y_1, \dots, y_M)$$

donde cada elemento representa el resultado de la j -ésima neurona de salida.

Ejemplo. En la [Figura 1](#) podemos observar una red neuronal superficial con tres unidades de entrada ($p = 3$), tres neuronas en la capa oculta ($N = 3$) y dos neuronas en la capa de salida ($M = 2$). La matriz de pesos W_1 tiene dimensión 3×3 y la matriz de pesos W_2 tiene dimensión 3×2 .

2.1.1. Caso multidimensional

En esta sección definimos a X como un vector en \mathbb{R}^p , es decir, como una única observación. Podemos aumentar la dimensión de X para que deje de ser un vector y pase a ser una matriz, donde ahora cada fila representa una observación distinta.

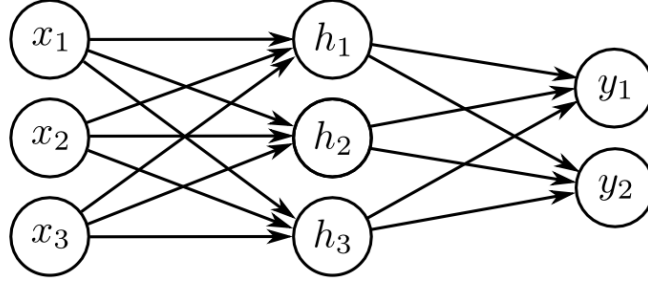


Figura 1: Price, S. (2024). *Understanding Deep Learning* [Figura 3.11, p.35].

Si ahora $X \in \mathbb{R}^{n \times p}$, entonces al multiplicar X por W_1 y aplicar la función de activación $h(\bullet)$ obtenemos la matriz Z de dimensión $n \times N$:

$$Z = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \ddots & \vdots \\ w_{n1} & \dots & w_{np} \end{pmatrix} \cdot \begin{pmatrix} w_{11} & \dots & w_{1N} \\ \vdots & \ddots & \vdots \\ w_{p1} & \dots & w_{pN} \end{pmatrix} = \begin{pmatrix} z_{11} & \dots & z_{1N} \\ \vdots & \ddots & \vdots \\ z_{p1} & \dots & z_{pN} \end{pmatrix}$$

donde en la i -ésima fila tenemos la salida de las N neuronas para la i -ésima observación.

Por último, tomamos la matriz Z y la multiplicamos por W' y le aplicamos la función de activación de salida $f(\bullet)$. Esto nos devuelve la matriz de resultados Y de dimensión $n \times M$:

$$Y = \begin{pmatrix} z_{11} & \dots & z_{1p} \\ \vdots & \ddots & \vdots \\ z_{n1} & \dots & z_{np} \end{pmatrix} \cdot \begin{pmatrix} w'_{11} & \dots & w'_{1N} \\ \vdots & \ddots & \vdots \\ w'_{p1} & \dots & w'_{pN} \end{pmatrix} = \begin{pmatrix} y_{11} & \dots & y_{1N} \\ \vdots & \ddots & \vdots \\ y_{p1} & \dots & y_{pN} \end{pmatrix}$$

donde en la i -ésima fila tenemos el resultado de las M neuronas finales para la i -ésima observación. Por ejemplo, si la función f fuese la función Softmax, la i -ésima fila tendría las probabilidades de la i -ésima observación de pertenecer a cada una de las M posibles clases.

Comentario. *Notar que la cantidad de parámetros de nuestro modelo, i.e., la cantidad de pesos, no se ve afectada por la cantidad de observaciones que tiene nuestra muestra. La cantidad de parámetros se ve afectada únicamente por la cantidad de unidades en las capas.*

2.2. Teorema de Aproximación Universal

La cantidad de neuronas dentro de la capa oculta es una medida de cuanta capacidad tiene nuestra red neuronal. Cuantas más neuronas tengamos, mejor podrá nuestro modelo aproximar funciones complejas.

El Teorema de Aproximación Universal nos dice que una red neuronal con una única capa oculta es capaz de aproximar, con una precisión arbitraria, cualquier función, siempre y cuando sus funciones de activación sea no lineales.

3. Redes Neuronales Profundas

3.1. Definición

Las redes neuronales profundas se caracterizan por tener dos o más capas ocultas. La principal diferencia con las redes neuronales superficiales es que los resultados de una capa oculta son utilizados como valores de entradas de la siguiente, como se puede observar en la [Figura 2](#).

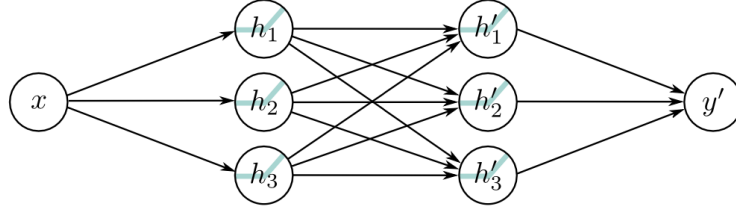


Figura 2: Price, S. (2024). *Understanding Deep Learning* [Figura 4.4, p.45].

Sea $X \in \mathbb{R}^p$ un vector de entrada. Este ejemplo se puede extender fácilmente a más dimensiones como en la [Sección 2.1.1](#).

Si la primera capa oculta tiene N neuronas, entonces la matriz de pesos asociada tiene dimensión $p \times N$. Al igual que antes, el producto interno de X con W nos devuelve el siguiente vector:

$$X \cdot W = \left(\sum_{i=1}^p w_{i1}x_i, \dots, \sum_{i=1}^p w_{iN}x_i \right)_{1 \times N}$$

Aplicando la función de activación $h(\bullet)$ para cada elemento del vector obtenemos:

$$Z = (z_1, \dots, z_N)$$

Si la segunda capa oculta tiene M neuronas, entonces la matriz de pesos asociada, W' , tiene dimensión $N \times M$. Multiplicamos al vector Z por la matriz de pesos W' para obtener:

$$Z \cdot W = \left(\sum_{i=1}^N w'_{i1}z_i, \dots, \sum_{i=1}^N w'_{iM}z_i \right)_{1 \times M}$$

Aplicando la función de activación $h(\bullet)$ para cada elemento obtenemos:

$$Z' = (z'_1, \dots, z'_N)$$

Finalmente, si la capa de salida tiene S neuronas, la matriz de pesos asociada, W'' , tiene dimensión $M \times S$. Multiplicando Z' por W'' y aplicando la función de activación de salida obtenemos el resultado de nuestra red:

$$Y = (y_1, \dots, y_S)$$