

THE PERCEPTRON

El Perceptron es una de las arquitecturas ANN más simples, se basa en una neurona artificial ligeramente diferente denominada una unidad lógica de umbral (TLU) o, a veces, una unidad de umbral lineal (LTU). Las entradas y salidas son números (en lugar de valores binarios de encendido / apagado) y cada conexión de entrada está asociada con un peso.

La función de paso más común utilizada en Perceptrons es la función de paso Heaviside:

$$\text{heaviside}(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases} \quad \text{sgn}(z) = \begin{cases} -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ +1 & \text{if } z > 0 \end{cases}$$

Se puede utilizar una sola TLU para una clasificación binaria lineal simple. Calcula una combinación lineal de las entradas y, si el resultado supera un umbral, genera la clase positiva. De lo contrario, genera la clase negativa (como una regresión logística o un clasificador lineal de SVM).

Un Perceptron se compone simplemente de una sola capa de TLU. Cuando todas las neuronas de una capa están conectadas a todas las neuronas de la capa anterior, la capa se denomina capa completamente conectada o capa densa.

¿Cómo se entrena el Perceptron?

La regla de Hebb dice que "Las células que se disparan juntas, se conectan juntas". Los perceptrones se entrenan utilizando una variante de esta regla que tiene en cuenta el error cometido por la red cuando hace una predicción; La regla de aprendizaje del Perceptron refuerza las conexiones que ayudan a reducir el error.

Más específicamente, el Perceptron recibe una instancia de entrenamiento a la vez, y para cada instancia hace sus predicciones. Por cada neurona de salida que produjo una predicción incorrecta, refuerza los pesos de conexión de las entradas que habrían contribuido a la predicción correcta. La regla se muestra en la siguiente ecuación:

$$w_{i,j}^{(\text{next step})} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

En esta ecuación:

- $w_{i,j}$ es el peso de conexión entre la i -ésima neurona de entrada y la j -ésima neurona de salida.
- x_i es el i -ésimo valor de entrada de la instancia de entrenamiento actual.
- y_j es la salida de la j -ésima neurona de salida para la instancia de entrenamiento actual.
- \hat{y}_j es la salida de destino de la j -ésima neurona de salida para la instancia de entrenamiento actual.
- η es la tasa de aprendizaje.

El límite de decisión de cada neurona de salida es lineal, por lo que los perceptrones son incapaces de aprender patrones complejos (al igual que los clasificadores de regresión logística). Sin embargo, si las instancias de entrenamiento son linealmente separables, Rosenblatt demostró que este algoritmo convergería en una solución, esto se llama teorema de convergencia del perceptrón.