

Redes Neuronales Convolucionales

Neuronas

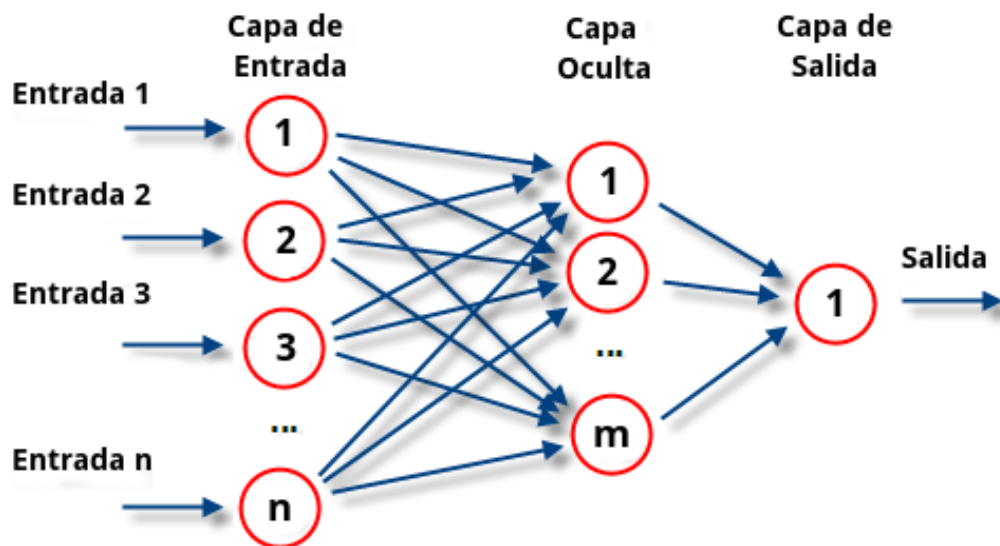
Bajo el marco teórico de las redes neuronales, una neurona es una función matemática que modela el funcionamiento de una neurona biológica. Una neurona posee un conjunto de entradas, pesos, y funciones de activación (umbrales).

En sí, las neuronas de una RNC intentan replicar matemáticamente a aquellas neuronas que tenemos en la cabeza.

Perceptrones multicapa

Una red neuronal convolucional es una variación de un **perceptrón multicapa**. Debido a esto, comparten ciertas características, lo que hace conveniente saber qué es y cómo funciona un perceptrón.

Los perceptrones multicapas son redes neuronales donde todas las neuronas (exceptuando aquellas pertenecientes a la capa de entrada) se encuentran conectadas entre sí.



Perceptrón multicapa, todas las neuronas están conectadas entre sí.

Capas de una red neuronal

Las capas de una red neuronal son divisiones en las que se agrupan un conjunto de neuronas.

Por ejemplo, la **capa de entrada** es aquella capa que recibe los datos de entrada. Cada parámetro de entrada es una neurona.

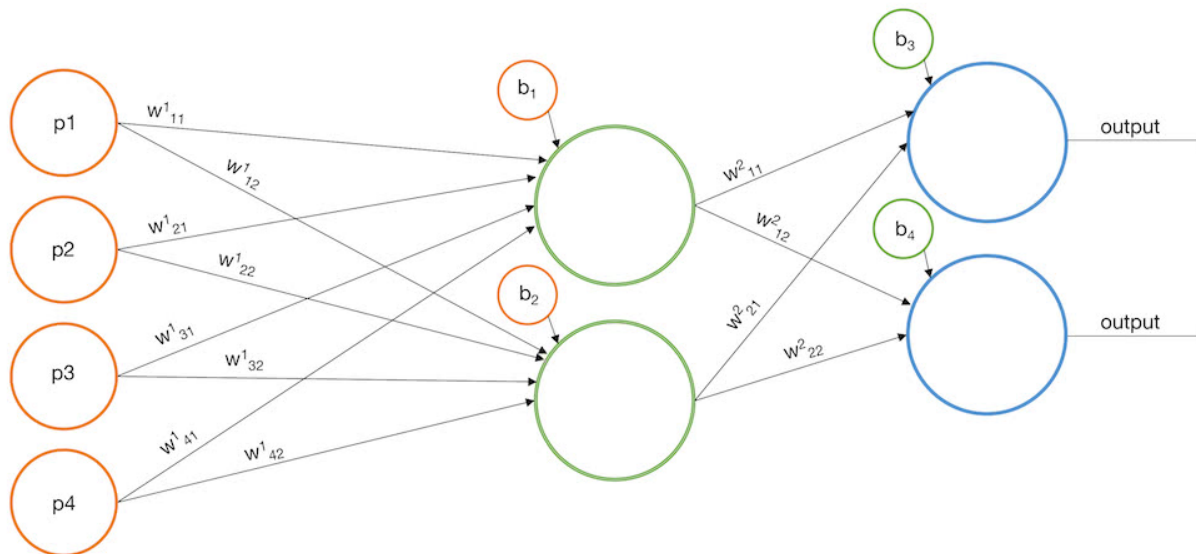
La **capa oculta** es aquella que está entre medio. Son un grupo de neuronas en donde las entradas que reciben las mismas provienen de otras neuronas, y sus salidas se dirigen hacia neuronas también.

La **capa de salida** es la capa final, donde las neuronas de esta capa reciben el valor que fue calculado por toda la red.

Conexiones

Las neuronas están conectadas entre sí. Cada conexión entre neuronas tiene asignado **un peso**. Los pesos regulan qué tan importante es para una neurona esa conexión.

A su vez, cada neurona tiene una “conexión” extra, que no proviene de ninguna otra neurona, conocida como el **umbral de activación**, que regula si la neurona se excita o no ante la entrada de los valores que vienen dados por las conexiones con pesos.



Pesos de las conexiones y sus umbrales de activación

Etapas de aprendizaje/entrenamiento

La etapa de aprendizaje consiste en alimentar la red neuronal con datos y calcular el error de la salida. Cuando detectamos un error, ajustamos ciertos parámetros (por ejemplo, los **pesos**) de la red para que la salida se acerque más al valor esperado.

Este proceso se repite una cierta cantidad de veces. **Es a este proceso el que conocemos con el nombre de época.**

Se dice que la red neuronal aprende cuando el error de la misma tiende a cero.

Etapas de validación

En la etapa de validación, se alimenta la red neuronal con un lote específico de datos, y se verifica que la salida dé el valor esperado. Si el resultado es cercano, significa que la red está aprendiendo.

La etapa de validación utiliza un set de datos diferente al de la etapa de entrenamiento. Esto es para minimizar el **overfitting**. El overfitting sucede cuando una red neuronal se “acostumbra” a un set de datos y solo puede predecir con exactitud ese set de datos. Es decir, es una pérdida de generalidad.

En la etapa de validación no se ajustan parámetros de la red neuronal (como los pesos). Lo que se verifica es que cada incremento que tengamos en la precisión con nuestro set de datos de entrenamiento también se vea reflejado en un set de datos con el que la red no estuvo expuesta (el set de validación). Si el incremento se ve reflejado en ambos lados, quiere decir que efectivamente la precisión está subiendo, y que no estamos produciendo overfitting.

Disminución del error

Para que nuestra red neuronal aprenda, necesitamos disminuir el error que se comete en cada predicción. El problema de aprendizaje en las redes neuronales se formula en términos de la minimización de la función de error.

Para minimizar esta función, se implementa una técnica llamada **Descenso del gradiente**. Esta técnica nos asegura hallar de forma rápida y eficiente los mínimos de la función, minimizando así el error.

Backpropagation

La propagación hacia atrás es un algoritmo en el que, luego de calcular una salida, el error de la misma se propaga hacia atrás, con el objetivo de modificar los parámetros que influyen en la misma.

Para lograr esto, se mide la contribución al error dada por cada parámetro. Se utilizan conceptos de cálculo diferencial: el descenso del gradiente, para ir hacia la dirección en que menor error haya, y las derivadas parciales, para ver cuánto contribuye cada parámetro en específico.

Es con este algoritmo que la red neuronal realmente mejora su aprendizaje.

Redes neuronales convolucionales

Es muy costoso tener todas las neuronas conectadas densamente (entre sí). Si analizamos una imagen de esta forma, sería muy ineficiente, y nuestra red neuronal podrá caer en overfitting.

Para evitar esto, se intenta conectar a las neuronas localmente. Esta es la diferencia de una red neuronal convolucional con un perceptrón.

En una imagen, cada píxel de entrada es una neurona. Las imágenes son representadas como matrices donde cada píxel es un valor.

Una convolución es la aplicación de un filtro (un filtro es una matriz de tamaño reducido) sobre una imagen. Al solapar el filtro sobre un área de la matriz, obtenemos un escalar como resultado. El filtro se mueve por toda la imagen, generando una nueva matriz que contiene los escalares como entrada.

Padding

El padding es una técnica que se usa para que, al aplicar el filtro y hacer una convolución, la matriz resultado sea del mismo tamaño que la imagen original. Consiste en agregar filas y columnas de ceros en la matriz de la imagen.

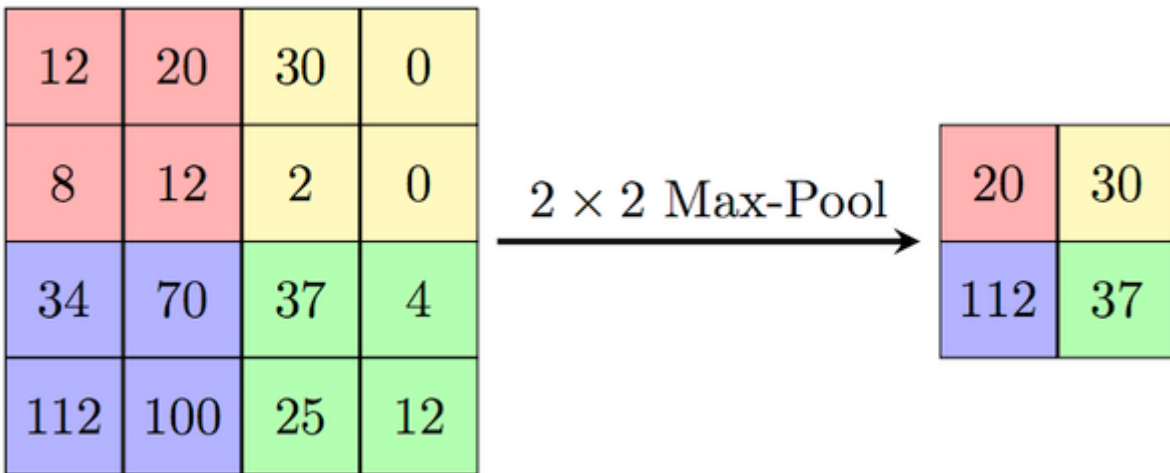
Por ejemplo, una convolución **same** (del mismo tamaño) utiliza padding.

Stride

El stride es el paso o salto que damos al aplicar el filtro. Se utiliza para saltar algunas áreas de la matriz, evitando así recorrerla en su completitud.

Pooling

Los pooling son un tipo de filtro que tratan de obtener un valor específico a medida que van recorriendo la matriz. Por ejemplo, el **máx-pooling** nos devuelve el valor máximo que se encuentra en esa sección de la matriz.



Hay otros como el average pooling o el min-pooling.