

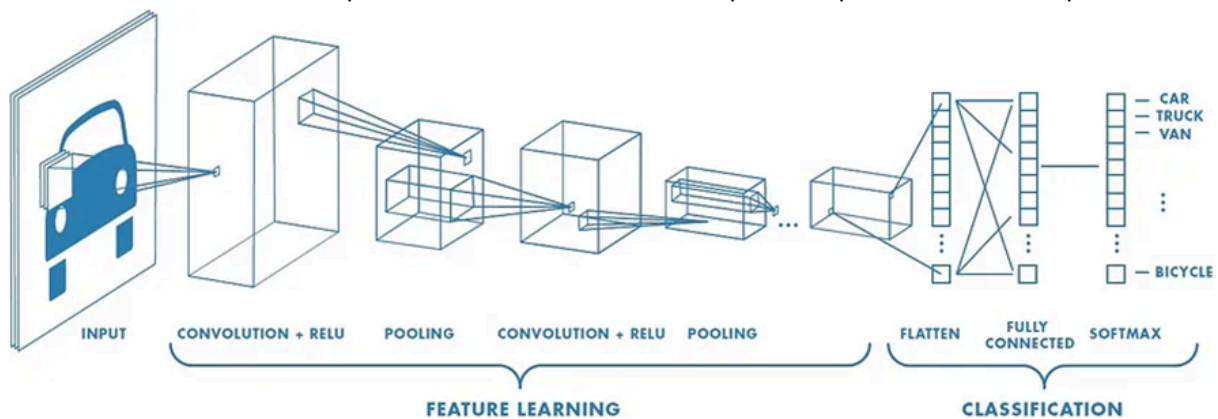
Redes Neuronales Convolucionales

Esta guía paso a paso desglosará las Redes Neuronales Convolucionales (CNNs) en términos sencillos. Cubriremos la configuración de tu entorno, la construcción de un modelo CNN y su entrenamiento para clasificar imágenes.

¿Qué Son las Redes Neuronales Convolucionales (CNNs)?

Una red neuronal convolucional es un tipo de red neuronal artificial utilizada en el reconocimiento y procesamiento de imágenes, inspirada en los procesos biológicos de la corteza visual de los animales. Las CNNs están compuestas por neuronas con pesos y sesgos aprendibles.

Las CNNs utilizan una técnica llamada convolución en lugar de la multiplicación matricial general en al menos una de sus capas. La convolución es un tipo de operación lineal especializada.



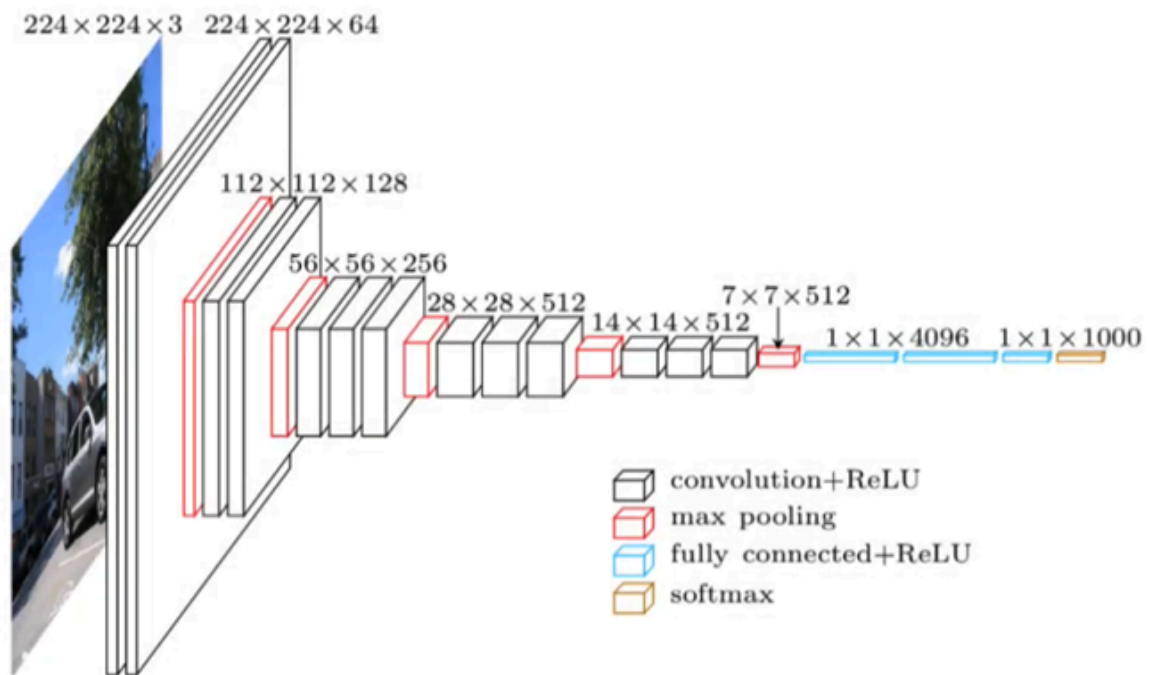
La convolución es una operación que nos permite extraer características de una imagen, para lo cual hace uso de un filtro o kernel, el cual está diseñado específicamente para extraer las características que resulten de mayor interés en la imagen.

Funcionamiento de una CNN

Las CNNs aplican filtros (pequeños rectángulos) a una imagen de entrada para detectar características como bordes o formas. Estos filtros se desplazan sobre el ancho y alto de la imagen de entrada y calculan productos punto entre el filtro y la entrada para producir un mapa de activación.

Las CNNs utilizan una técnica llamada convolución. Se toman datos digitales de entrada y, progresivamente, a través del entrenamiento de múltiples capas convolucionales (conformadas por filtros que realizan diferentes operaciones sobre la imagen) se extraen progresivamente características cada vez más complejas de la imagen para lograr su reconocimiento.

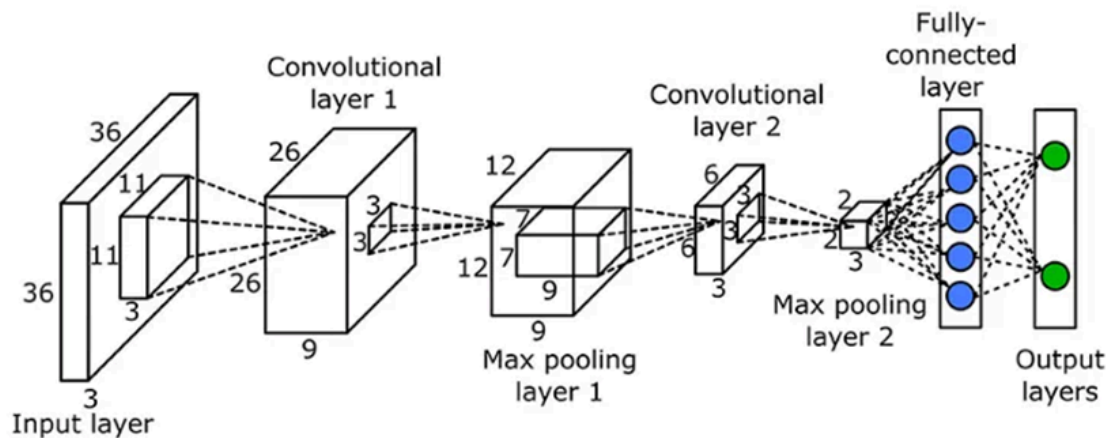
En las primeras capas algunos de ellos se encargarán por ejemplo de detectar tonalidades rojas, otros de tonalidades verdes, otros por ejemplo podrán detectar líneas curvas y otros rectas, y así.



Capas de una CNN

1. Capas de Convolución.

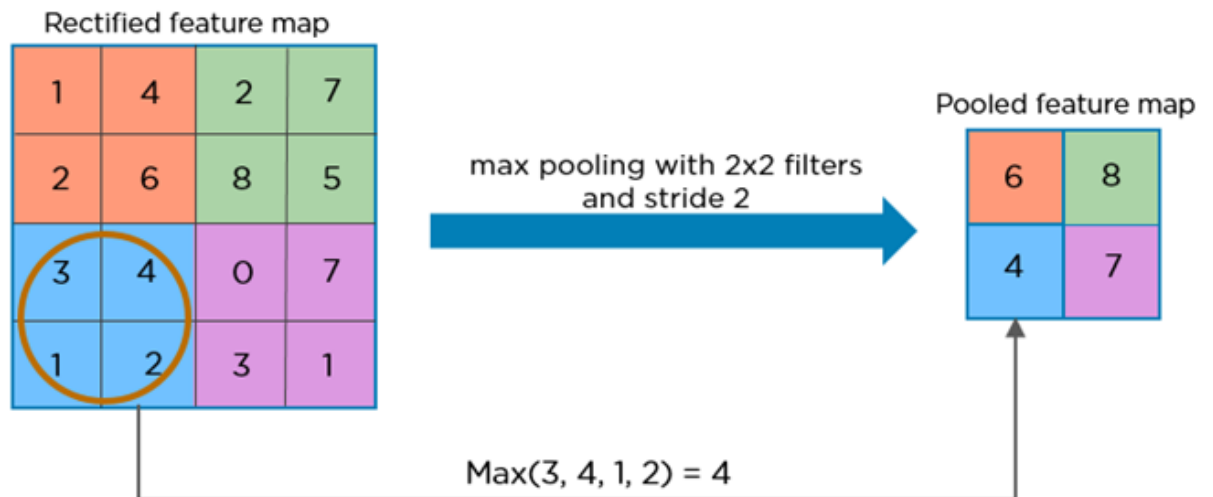
Las capas de convolución aplican una operación de convolución a la entrada, pasando un filtro sobre toda la imagen. Este filtro detecta características como bordes o curvas en la imagen. Varios filtros pueden detectar diferentes características.



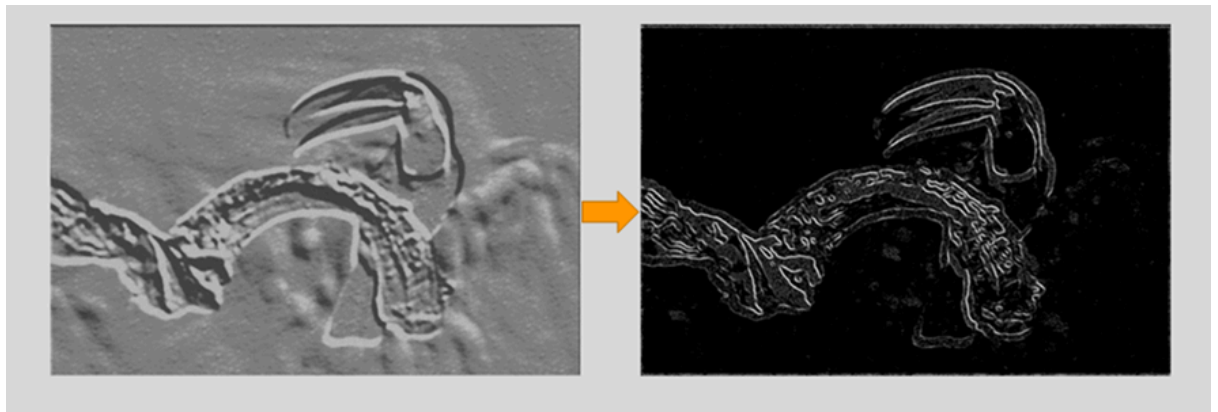
2. Capas de Pooling

Las capas de pooling se insertan entre las capas de convolución. Estas capas reducen la dimensionalidad de los mapas de características para disminuir el número de parámetros, controlar el sobreajuste y hacer que la red sea invariante a pequeñas traslaciones. Las

técnicas comunes de pooling incluyen max pooling y average pooling.



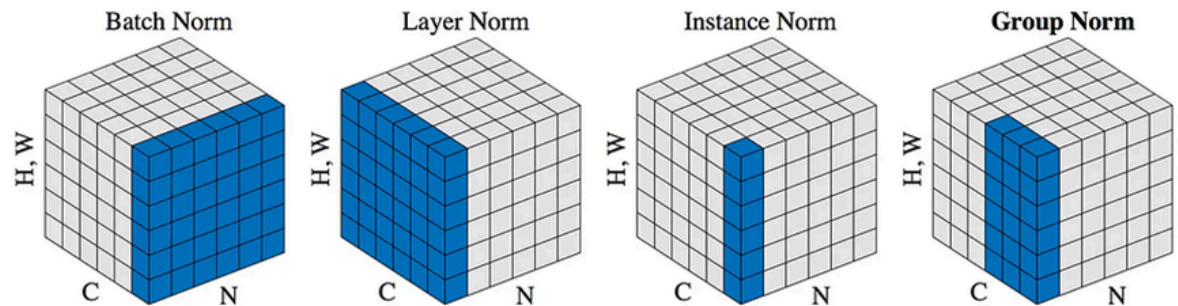
La capa de pooling se encarga de simplificar los datos y reducir su dimensionalidad, lo que hace que la red sea más eficiente desde el punto de vista computacional. No detecta características específicas como bordes o ojos por sí misma. En cambio, trabaja con los datos que han sido procesados por las capas de convolución, que son las que realmente detectan esos detalles.



El pooling aplica una operación estadística simple (como tomar el máximo o el promedio) sobre un bloque de píxeles en los mapas de características.

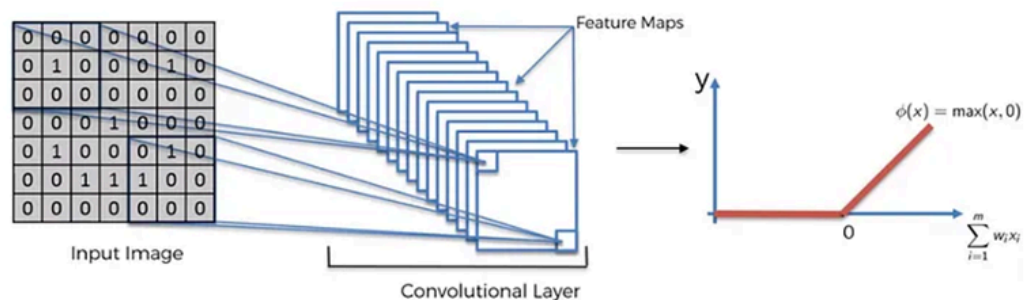
3. Capas de Normalización.

Las capas de normalización, como la normalización por lotes (Batch Normalization), aseguran que las activaciones de cada capa estén bien condicionadas y ayudan a prevenir el sobreajuste.



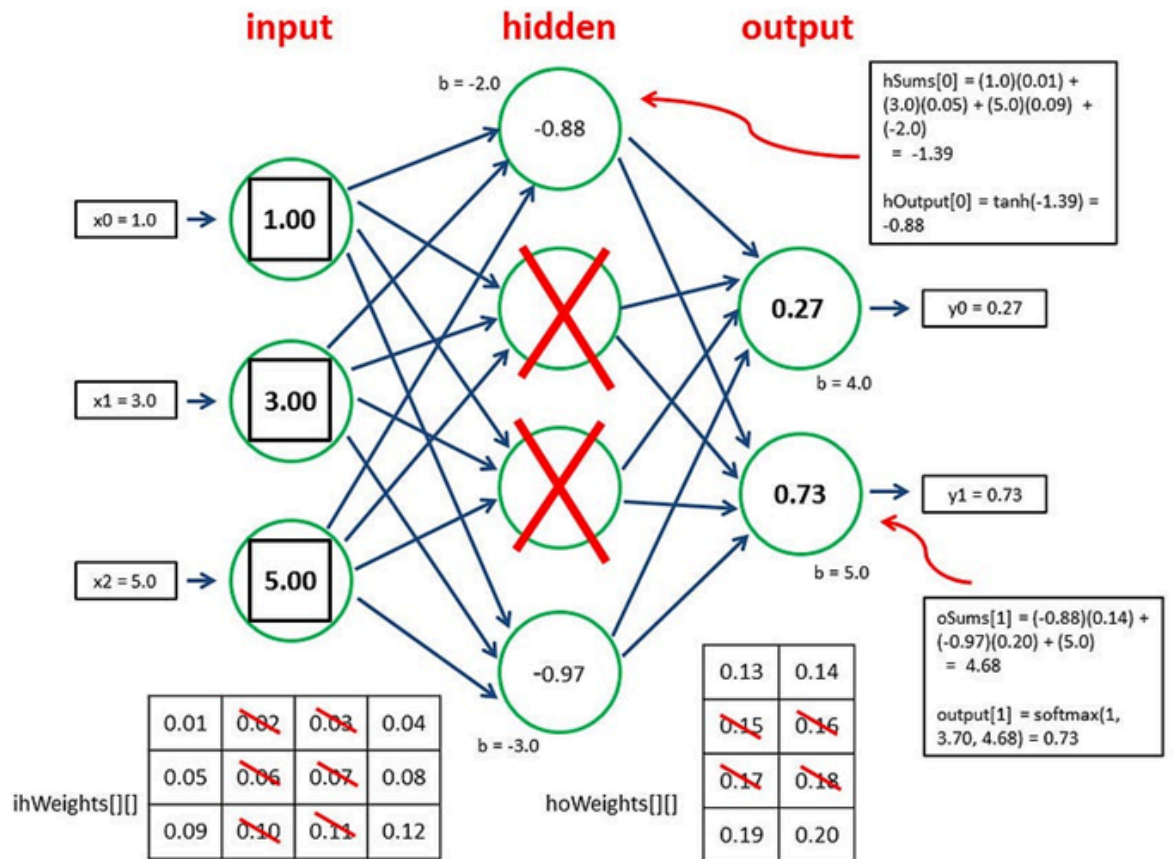
4. Capas de Activación

Estas capas aplican funciones de activación no lineales, como ReLU, para introducir no linealidad en el modelo, permitiendo que aprenda representaciones más complejas de los datos de entrada.



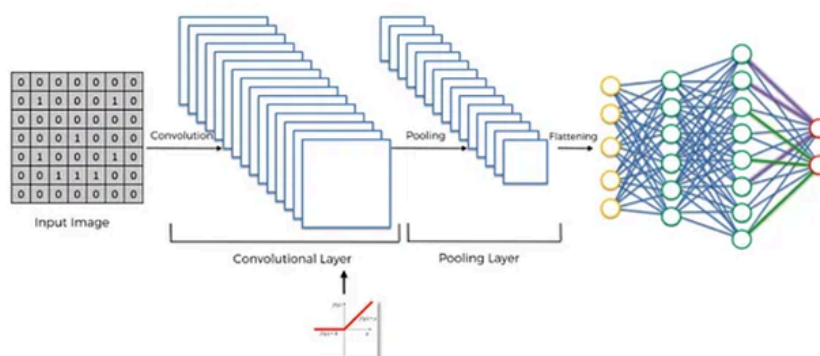
5. Capas de Dropout

Utilizadas para prevenir el sobreajuste, las capas de dropout eliminan aleatoriamente neuronas durante el entrenamiento, asegurando que el modelo no memorice los datos de entrenamiento sino que generalice a datos nuevos.



6. Capas Densas

Después de las capas convolucionales y de pooling, las capas densas combinan las características extraídas para hacer la predicción final. En una CNN, la capa densa usualmente es la capa final y produce las predicciones de salida.



El entrenamiento

Lo interesante de las Redes Convolucionales es que no es necesario diseñar manualmente cada uno de los filtros a utilizar. En su lugar, durante el entrenamiento, la misma red “aprende” de forma automática, y con la ayuda de una gran cantidad de imágenes de entrenamiento, a ajustar iterativamente estos coeficientes para progresivamente ir detectando características cada vez más complejas en el proceso de entrada.

Lo que sí elegimos es las capas y las configuraciones.

Cómo elegir las capas y sus configuraciones:

Sí, es un tema complejo. Pero va una idea general para tener una mejor aproximación al entendimiento de estas redes.

Hay 3 aspectos a tener en cuenta:

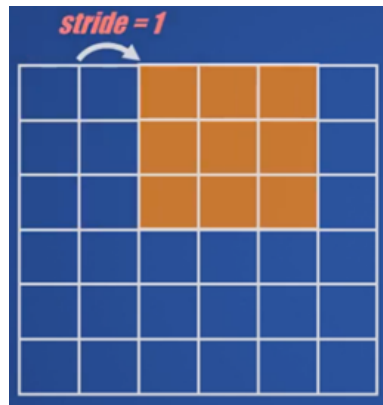
1. **Capas de Convolución:** Encuentran patrones en las imágenes.
2. **Max Pooling:** Simplifica y reduce el tamaño de la imagen.
3. **Flatten:** Convierte la información en un formato que las capas densas puedan usar.
4. **Capas Densas:** Realizan la clasificación final basándose en la información procesada.

1. Capa de Convolución (Conv2D)

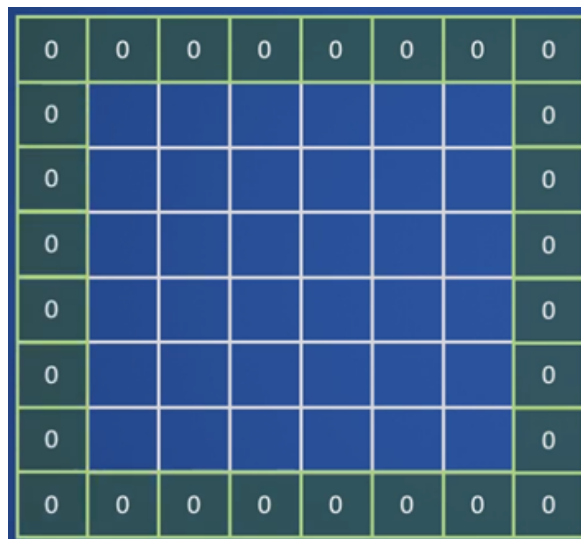
Propósito: La capa de convolución sirve para encontrar patrones en las imágenes, como bordes y texturas. Imagina que estás buscando patrones específicos en una foto, como el borde de un objeto. La capa de convolución usa pequeños filtros que "exploran" la imagen para encontrar estos patrones.

Configuración:

- **Número de Filtros:**
 - **Qué es:** Son como los diferentes "ojos" que usan distintos enfoques para mirar la imagen.
 - **Ejemplo:** Si tienes 32 filtros, la capa puede encontrar 32 tipos diferentes de patrones, como bordes verticales, horizontales, texturas, etc.
 - **Cómo afecta:** Más filtros pueden detectar más patrones, pero también hacen que el modelo sea más complejo y tarde más en entrenar.
- **Tamaño del Filtro:**
 - **Qué es:** Es el tamaño del área que el filtro "mira" a la vez.
 - **Ejemplo:** Un filtro de 3x3 examina un bloque de 3 píxeles por 3 píxeles. Si el filtro es 5x5, examina un área más grande.
 - **Cómo afecta:** Filtros más grandes pueden capturar patrones más grandes, pero también requieren más cálculos.
- **Stride (Paso):**
 - **Qué es:** Es cuánto se mueve el filtro cada vez que lo aplica.



- **Ejemplo:** Si el stride es 1, el filtro se mueve un píxel a la vez. Si el stride es 2, el filtro salta dos píxeles cada vez.
- **Cómo afecta:** Un stride mayor reduce el tamaño de la imagen más rápidamente, haciendo que el modelo sea más eficiente pero con menos detalle.
- **Padding:**
 - **Qué es:** Recordemos que la convolución genera una imagen resultante de un menor tamaño que la imagen original. Pues el padding se usa para lograr que la imagen resultante y la original tengan el mismo tamaño. Así, el padding consiste simplemente en agregar píxeles con valor igual a cero a los bordes de la imagen original.



- **Ejemplo:** Con padding='same', añades píxeles alrededor de la imagen para que el tamaño de la imagen no cambie después de aplicar el filtro. Con padding='valid', no añades píxeles, y la imagen se hace un poco más pequeña.
- **Cómo afecta:** padding='same' mantiene el tamaño de la imagen, mientras que padding='valid' lo reduce.

2. Max Pooling (MaxPooling2D)

Propósito: El pooling simplifica la imagen al reducir su tamaño, lo que ayuda al modelo a enfocarse en las características más importantes y hace que el entrenamiento sea más rápido. El pooling es

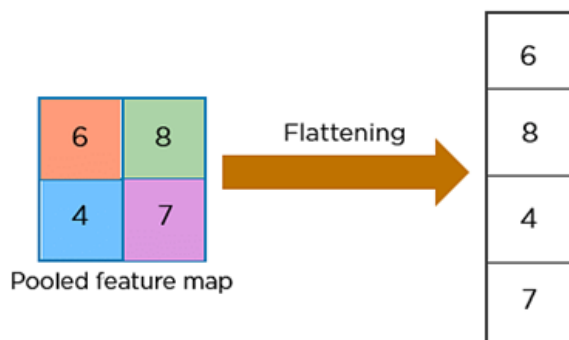
una operación que permite analizar el contenido de una imagen por regiones (o bloques) para extraer la información más representativa de las mismas. En el caso del max-pooling la imagen es dividida en regiones del mismo tamaño, y para cada región se extrae simplemente el valor máximo que corresponderá a un pixel en la imagen resultante.

Configuración:

- **Tamaño del Pool:**
 - **Qué es:** Es el tamaño del bloque que se examina para reducir la imagen.
 - **Ejemplo:** Si usas un pool de 2x2, el modelo toma el valor máximo de cada bloque de 2 píxeles por 2 píxeles y descarta el resto.
 - **Cómo afecta:** Esto reduce la cantidad de datos y hace que el modelo sea menos sensible a pequeñas variaciones en la imagen.

3. Flatten

Propósito: Después de usar capas de convolución y pooling, la imagen procesada es una especie de "cubierta" de información. La capa de flatten convierte esta "cubierta" en una sola línea de datos (vector) para que pueda ser utilizada por las capas densas para hacer la clasificación.



Configuración:

- No necesita parámetros específicos; simplemente toma todos los datos de las capas anteriores y los convierte en una forma que las capas densas puedan manejar.

4. Capa Densa (Dense)

Propósito: Las capas densas hacen la clasificación final al combinar toda la información extraída y procesada para decidir a qué categoría pertenece la imagen.

Configuración:

- **Número de Neuronas:**
 - **Qué es:** Son las "unidades de procesamiento" que combinan la información para hacer la clasificación.
 - **Ejemplo:** Si estás clasificando entre 10 tipos de animales, la última capa densa tendría 10 neuronas, una para cada tipo de animal.
- **Función de Activación:**

- **ReLU (Rectified Linear Unit):**
 - **Qué es:** Transforma los datos para introducir complejidad no lineal. Por ejemplo, convierte valores negativos en cero y deja los valores positivos como están.
 - **Cómo afecta:** Permite que el modelo aprenda características complejas y no lineales.
- **Softmax:**
 - **Qué es:** Convierte los valores finales en probabilidades que suman 1.
 - **Cómo afecta:** Da una probabilidad para cada clase, lo que es útil para decidir a qué categoría pertenece la imagen (por ejemplo, 0.2 para perro, 0.8 para gato).

Conclusión

Las Redes Neuronales Convolucionales (CNNs) son una poderosa arquitectura de aprendizaje profundo bien adaptada para tareas de clasificación de imágenes y reconocimiento de objetos. Con su capacidad para extraer automáticamente características relevantes, manejar imágenes ruidosas y aprovechar modelos preentrenados, las CNNs han demostrado un rendimiento de vanguardia en una variedad de tareas de visión por computadora. A pesar de sus limitaciones, como el alto costo computacional y la propensión al sobreajuste, siguen siendo una herramienta popular para muchas tareas de visión por computadora y probablemente continuarán siendo un área clave de investigación y desarrollo en los próximos años.