

## CONVOLUCIÓN 2D

## KERAS Y PYTHON

## REQUISITOS:

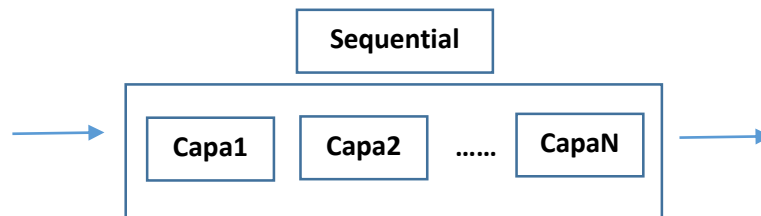
Tener instalado python , numpy , tensorflow 2.0 y Keras .

## KERAS

API desarrollado en Python que puede correr sobre TensorFlow , CNTK o Theano. Su uso es para realizar desarrollo en corto tiempo , las operaciones que se realizan pueden ser ejecutados en dispositivos CPU o GPU. La versión de Keras 2.3.0 soporta tensorflow 2.0 .

Lo primero que se debe realizar es crear un modelo para poder diseñar la arquitectura de la red neuronal.

La clase Sequential permite representar un conjunto de capas de manera secuencial .



## CONVOLUCIÓN EN 2 DIMENSIONES

La convolución es una operación utilizada en el área de procesamiento de señales y su extensión de 2D aplicada en imágenes juega un rol importante a la hora de poder resaltar, extraer, atenuar características de una imagen en base a la definición de un kernel.

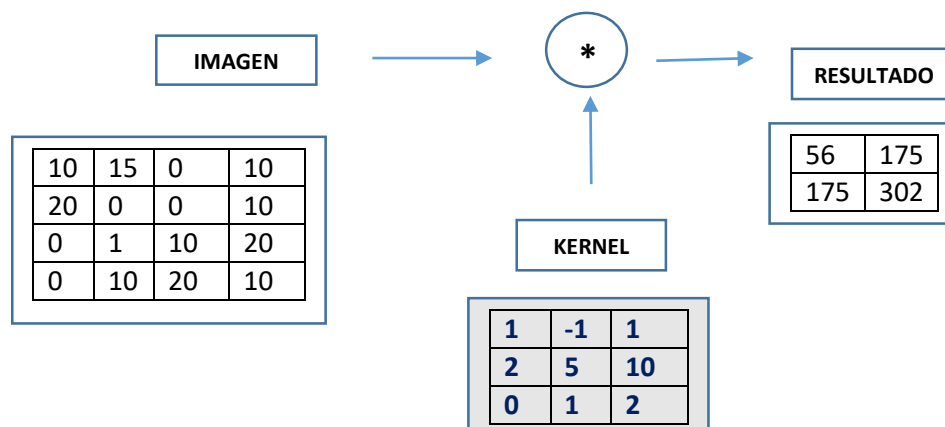
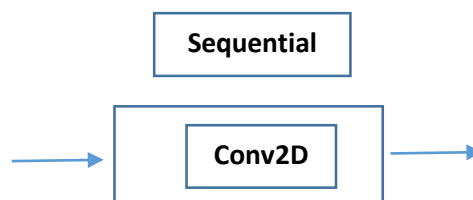
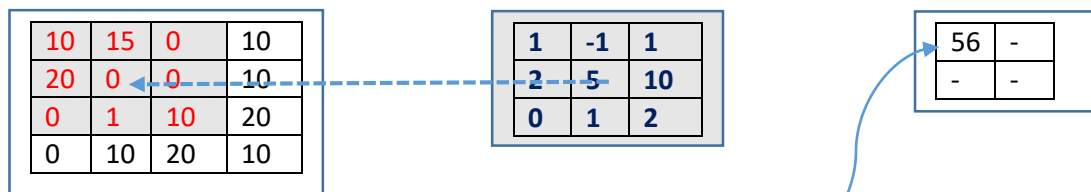


Figura N°1 Convolución 2D en Deep Learning

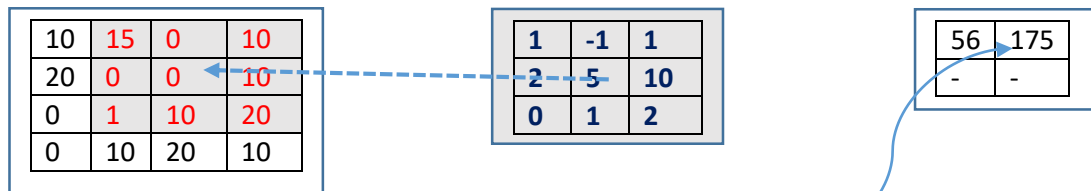


Según la operación de convolución utilizada en el área de Deep Learning , el resultado de la figura N°1 viene dado por las siguientes operaciones :

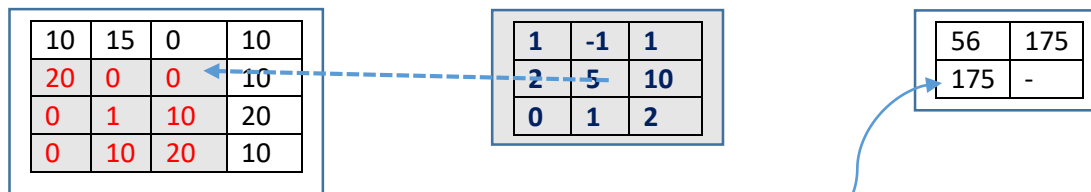
Resultados parciales:



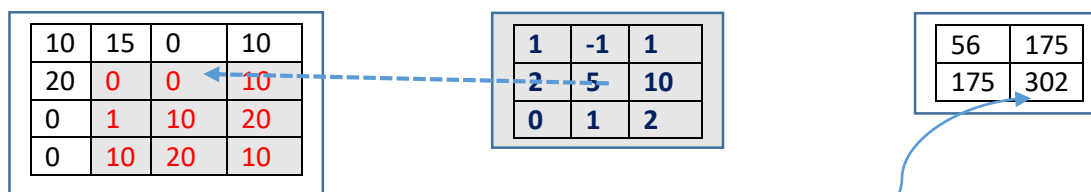
$$10*1 + 15*(-1) + 0*1 + 20*2 + 0*5 + 0*10 + 0*1 + 1*1 + 10*2 = 56$$



$$15*1 + 0*(-1) + 10*1 + 0*2 + 0*5 + 0*10 + 1*1 + 10*1 + 20*2 = 175$$



$$20*1 + 0*(-1) + 0*1 + 0*2 + 1*5 + 10*10 + 0*1 + 10*1 + 20*2 = 175$$



$$0*1 + 0*(-1) + 10*1 + 1*2 + 10*5 + 20*10 + 10*1 + 20*1 + 10*2 = 302$$

NF\_input=4

NC\_input=4

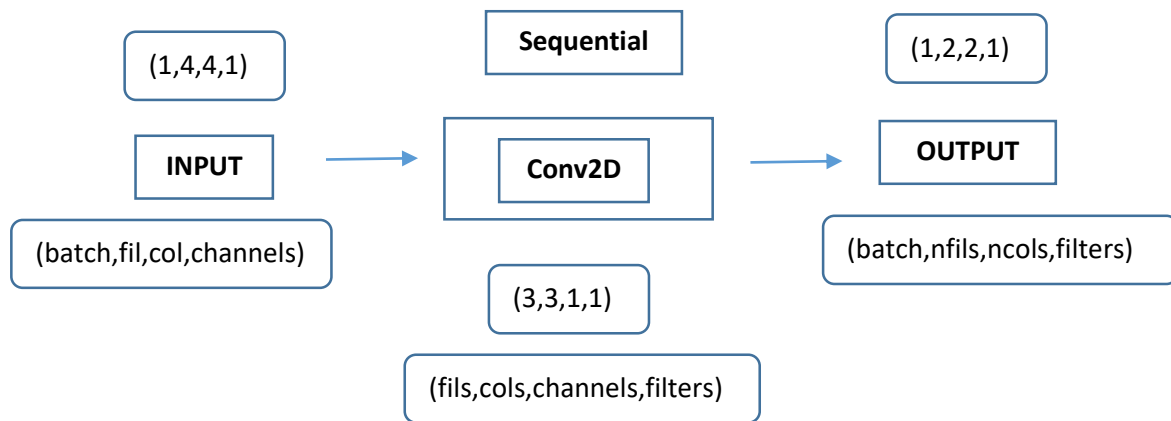
NF\_kernel=3

NC\_kernel=3

Las nuevas dimensiones de la imagen de salida es:

**NF\_output=NF\_input - NF\_kernel +1**

**NC\_output=NC\_input - NC\_kernel +1**



Existen dos parámetros de suma importancia denominados **strides** y **padding**.

El parámetro **<strides>** define el paso en que se mueve el kernel sobre la imagen de entrada . **strides =1** indica que cada paso en sentido horizontal y vertical es de una posición.

El parámetro **<padding>** define si se realizar algún relleno sobre la imagen de entrada , se puede apreciar a la hora de definir una capa de Conv2D el parámetro **padding** puede asumir el string **"valid"** o **"same"** donde **"valid"** indica que las operaciones se realizaran sobre datos validos de la imagen y el string **"same"** indica que se realizara un relleno de números 0 sobre la imagen de entrada para poder brindar una salida con las mismas dimensiones .

La fórmula anterior es cierta cuando el parámetro **strides=1** y **padding="valid"**.

Para mayor información consultar en los siguientes links:

<https://arxiv.org/pdf/1603.07285.pdf>

[https://www.google.com/search?q=strides+and+padding+convolutional+network&rlz=1C1CHBF\\_esPE837PE837&oq=strides+and+padding+convolutional+network&aqs=chrome..69j57j33l3.5525j0j7&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=strides+and+padding+convolutional+network&rlz=1C1CHBF_esPE837PE837&oq=strides+and+padding+convolutional+network&aqs=chrome..69j57j33l3.5525j0j7&sourceid=chrome&ie=UTF-8)