

CONVOLUCIÓN 2D

KERAS Y PYTHON

REQUISITOS:

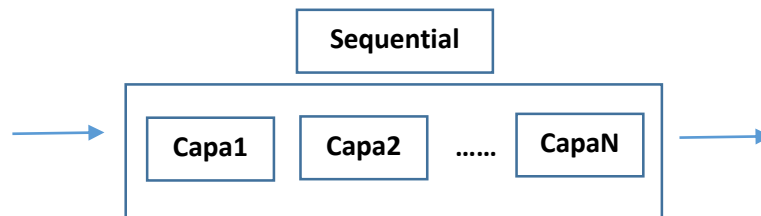
Tener instalado python , numpy , tensorflow 2.0 y Keras .

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.

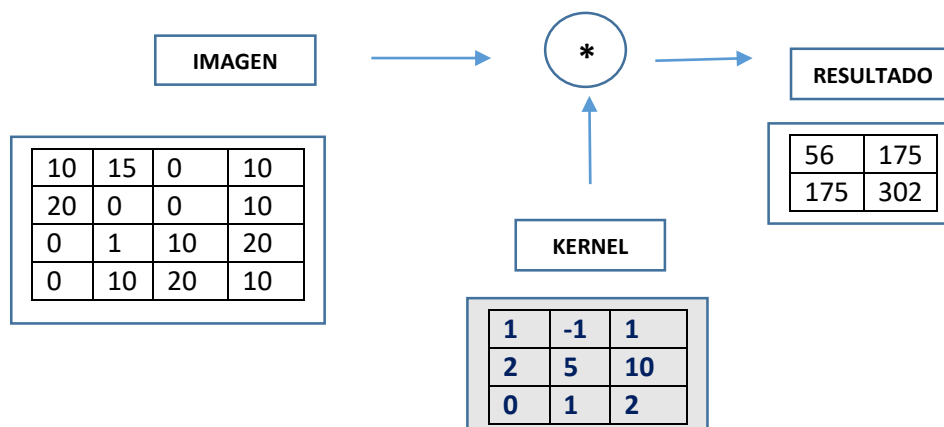
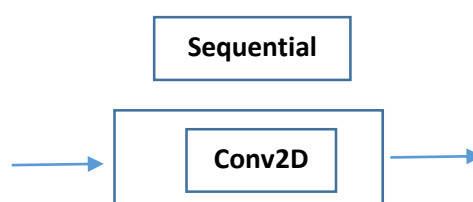
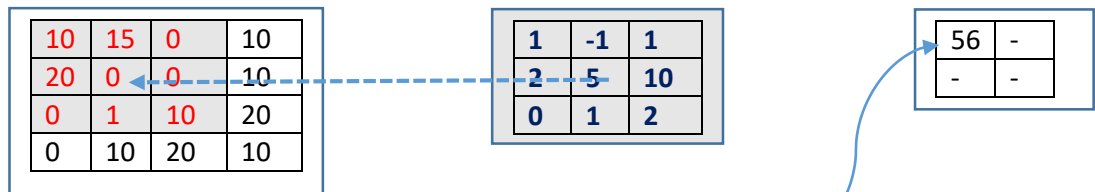


Figura N°1 Convolución 2D en Deep Learning

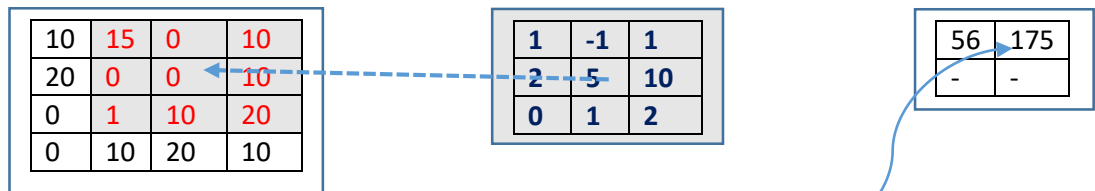


Según la operación de convolución utilizada en el área de Deep Learning , el resultado de la figura N°1 viene dado por las siguientes operaciones :

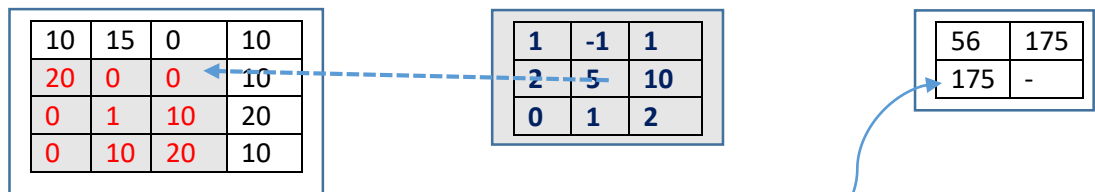
Resultados parciales:



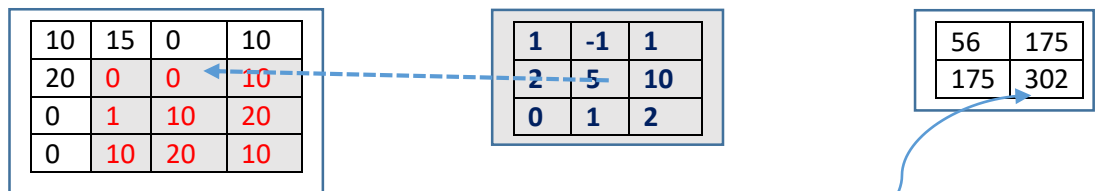
$$10*1 + 15*(-1) + 0*1 + 20*2 + 0*5 + 0*10 + 0*1 + 1*1 + 10*2 = 56$$



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



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



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

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

