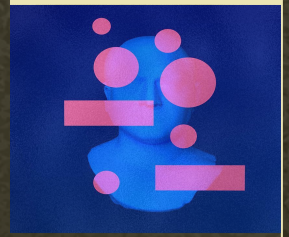
INTRODUCCIÓN AL MACHINE LEARNING APLICADO AL AUDIO

Teoría e Implementación

Diciembre 2023, Universidad de Chile

Profesor: P.h.D Rodolfo Lobo C.

Clase 09



INTRODUCCIÓN AL MACHINE LEARNING APLICADO AL AUDIO

Teoría e Implementación

Diciembre 2023, Universidad de Chile

Profesor: P.h.D Rodolfo Lobo C.

MODELOS AVANZADOS

Redes Recurrentes, Redes Convolucionales, GANs

Objetivos de la Clase

Al finalizar la clase tú aprenderás:

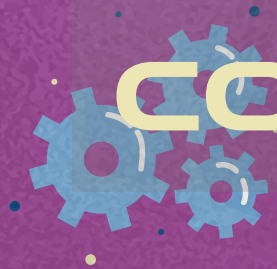
Objetivos de Aprendizaje:

- Conocer la arquitectura de una red convolucional estándar.
- Entender cómo se aplica la operación de convolución.
- Entender cómo se aplica la operación de convolución.

¿Cómo lo Lograremos?:

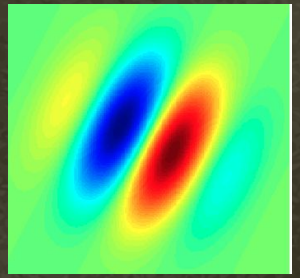
- A través de la revisión de conceptos de la literatura.
- Ejemplos en Notebooks de Google Colab.
- Algunos ejemplos de cálculos a mano!.

¿Qué son las redes convolucionales?



Evolución histórica

Cell Study Hubel and Torsten



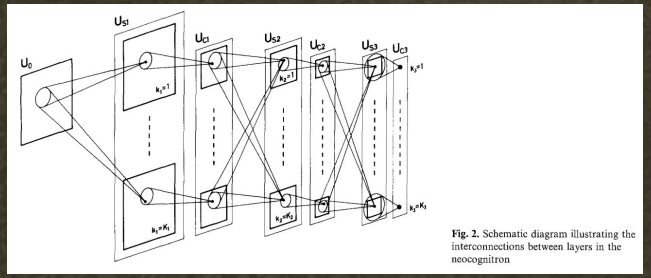
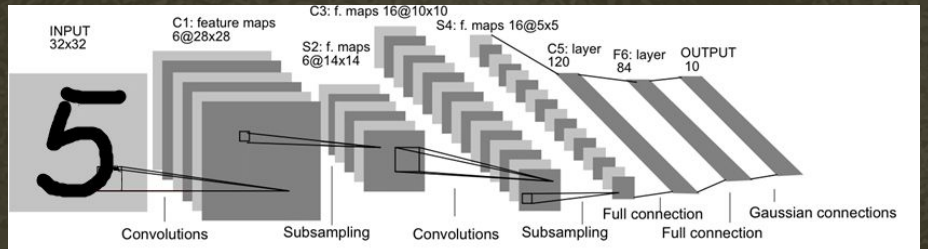


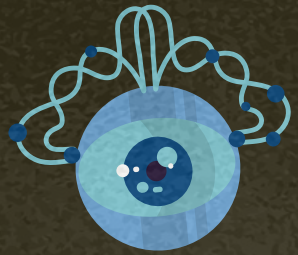
Fig. 2. Schematic diagram illustrating the interconnections between layers in the neocognitron

Self-supervised vision model: neocognitrón



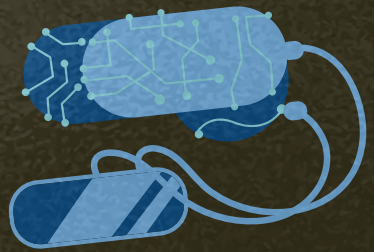
An early (Le-Net5) Convolutional Neural Network design, LeNet-5, used for recognition of digits

Lenet-5: gradient descent + conv network based on neocognitron ideas.

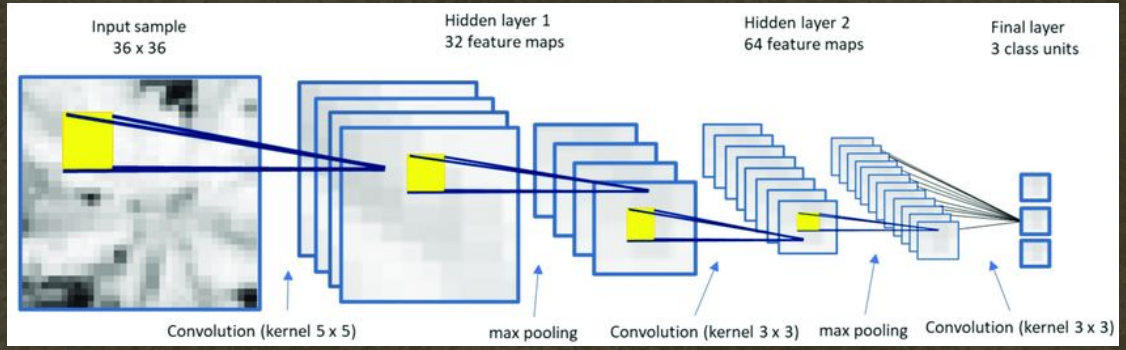


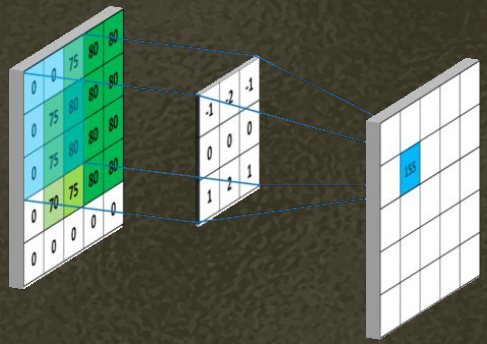
¿Qué es una red convolucional?

Es una red neuronal que utiliza una operación llamada “convolución” a los valores de entrada. Se basa en el funcionamiento de células neuronales del córtex visual (Hubel, Wiesel 1950 - 1960). Comúnmente se utilizan en problemas asociados a imágenes.

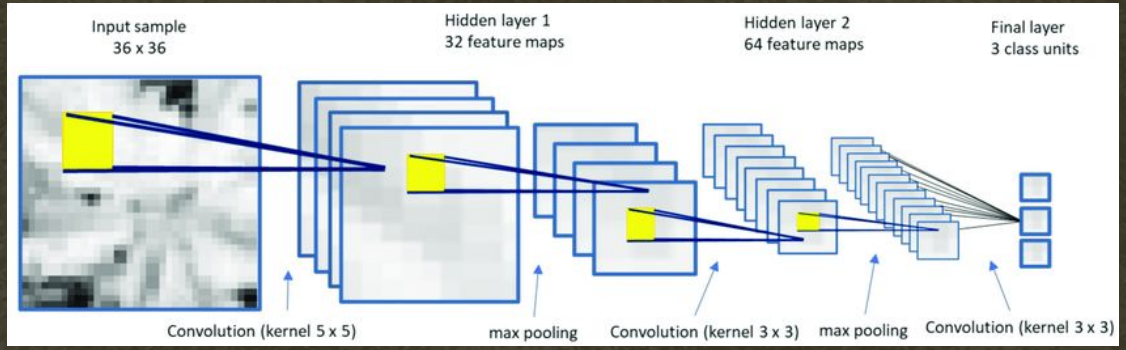


¿ Qué es una red convolucional?

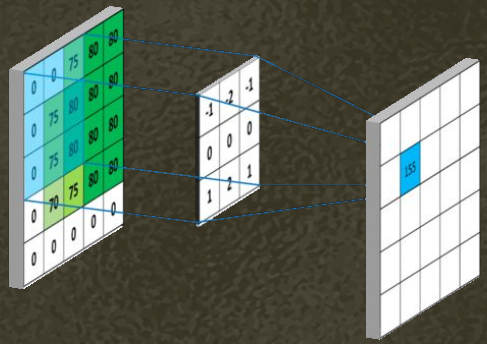




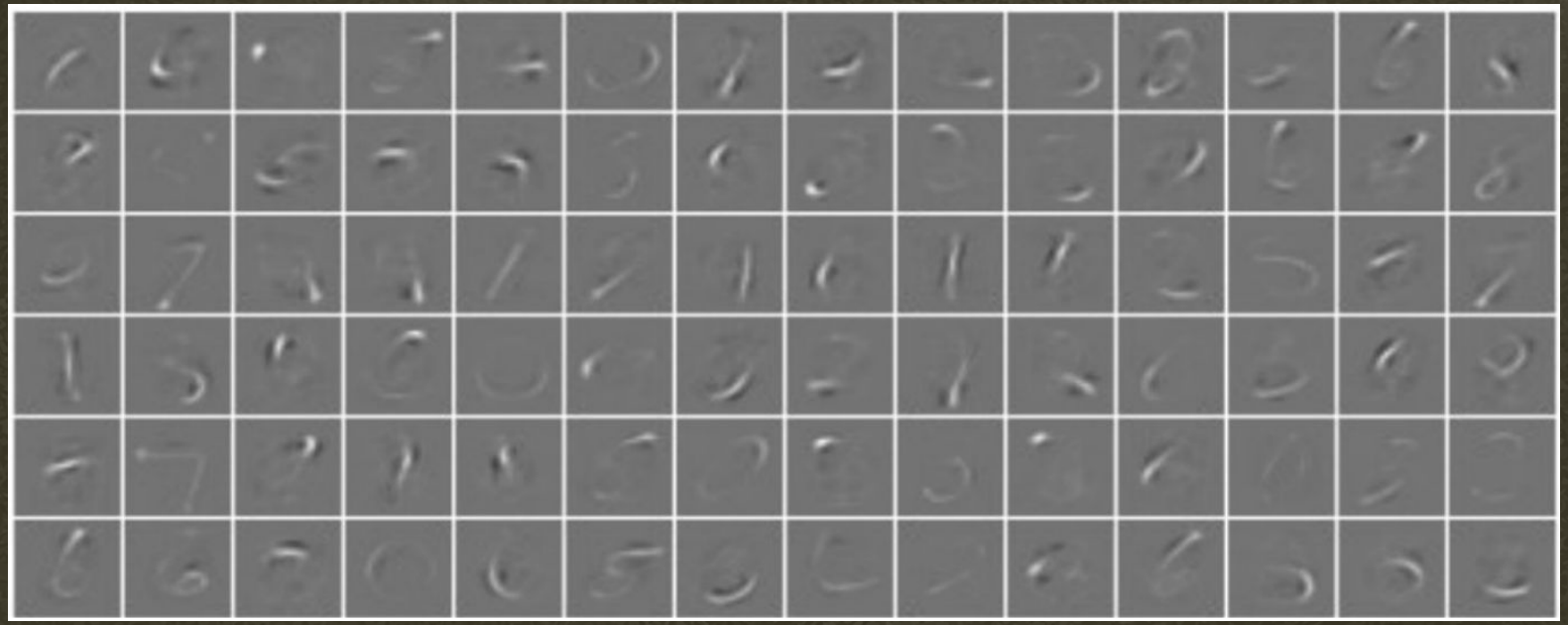
¿ Qué es una red convolucional?



- 3D
- Imágenes
- Kernels/Filtros



¿Qué aprende este tipo de redes?



Parámetros de una Red Convolucional CNN

La estructura y funcionalidad de una red convolucional puede dividirse en 4 partes fundamentales, cuando es utilizada particularmente con información visual:

- La capa de entrada que recibe los píxeles de la imagen.
- La capa convolucional que determina qué salidas están conectadas a regiones locales de la entrada (reconocer relaciones y estructuras dentro de las imágenes), a través de productos escalares entre los filtros y las sucesivas capas de entrada. Aplicar una función de activación no lineal, generalmente RelU, componente a componente.
- Aplicar Pooling, que se puede definir de manera simple como una reducción de dimensión del espacio vectorial en el cual se encuentra una salida de una capa convolucional, reduciendo el número de parámetros.
- Y finalmente, una red totalmente conectada, que permite calcular scores o clases desde las funciones de activación. En el caso de clasificar, se utilizan funciones como la softmax y funciones de pérdida como la entropía.

Parámetros de una Red Convolucional CNN

Ejemplo de como se ve la arquitectura de una red convolucional utilizando tensorflow-keras [LINK EJEMPLO]

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 32, 32, 3)]       0

 conv2d_4 (Conv2D)           (None, 30, 30, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_3 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_6 (Conv2D)           (None, 4, 4, 128)         73856

 flatten_1 (Flatten)         (None, 2048)              0

 dense (Dense)               (None, 10)                20490

=================================================================
Total params: 113,738
Trainable params: 113,738
Non-trainable params: 0
_________________________________________________________________
```

Parámetros de una Red Convolucional CNN

Ejemplo de como se ve la arquitectura de una red convolucional utilizando tensorflow-keras [LINK EJEMPLO]

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 32, 32, 3)]       0

 conv2d_4 (Conv2D)           (None, 30, 30, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_3 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_6 (Conv2D)           (None, 4, 4, 128)         73856

 flatten_1 (Flatten)         (None, 2048)              0

 dense (Dense)               (None, 10)                20490

=================================================================
Total params: 113,738
Trainable params: 113,738
Non-trainable params: 0
_________________________________________________________________
```

(image_height, image_width, image_channels)

Parámetros de una Red Convolucional CNN

Ejemplo de como se ve la arquitectura de una red convolucional utilizando tensorflow-keras [LINK EJEMPLO]

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 32, 32, 3)]       0

 conv2d_4 (Conv2D)           (None, 30, 30, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_3 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_6 (Conv2D)           (None, 4, 4, 128)         73856

 flatten_1 (Flatten)         (None, 2048)              0

 dense (Dense)               (None, 10)                20490

=================================================================
Total params: 113,738
Trainable params: 113,738
Non-trainable params: 0
_________________________________________________________________
```

(image_height, image_width, image_channels)

Tensor de rango 3

Parámetros de una Red Convolucional CNN

Ejemplo de como se ve la arquitectura de una red convolucional utilizando tensorflow-keras [LINK EJEMPLO]

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 32, 32, 3)]       0

 conv2d_4 (Conv2D)           (None, 30, 30, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_3 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_6 (Conv2D)           (None, 4, 4, 128)         73856

 flatten_1 (Flatten)         (None, 2048)              0

 dense (Dense)               (None, 10)                20490

=================================================================
Total params: 113,738
Trainable params: 113,738
Non-trainable params: 0
_________________________________________________________________
```

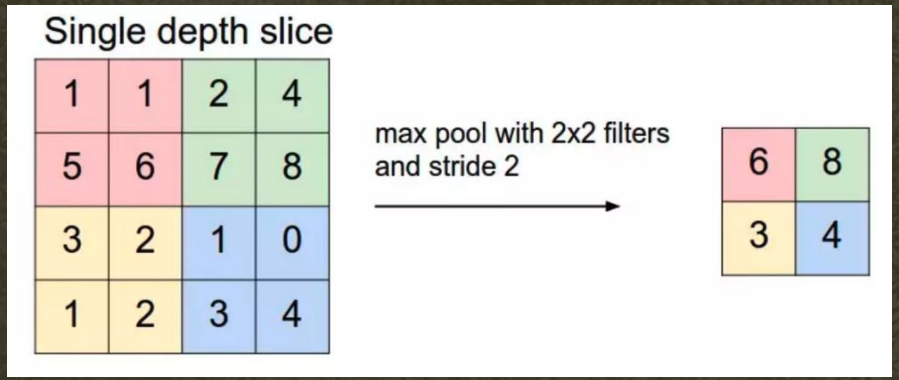
(image_height, image_width, image_channels)

Tensor de rango 3

El número de canales del filtro debe coincidir con el número de canales de la imagen. Por ejemplo, en el caso de la imagen de CIFAR, es de 32x32x3 y eventualmente podríamos usar un filtro 5x5x3, donde el último 3 representa el número de canales. En el caso de imágenes representa los canales RGB (red-green-blue)

Capa de Pooling

- Se utiliza generalmente después de una capa convolucional
- Se utiliza para disminuir el ancho y alto en las dimensiones de las capas internas.
- Reduce el número de parámetros, evita el overfit y disminuye el tiempo de cómputo.
- Una de las formas más utilizadas es el max pooling, ejemplo:

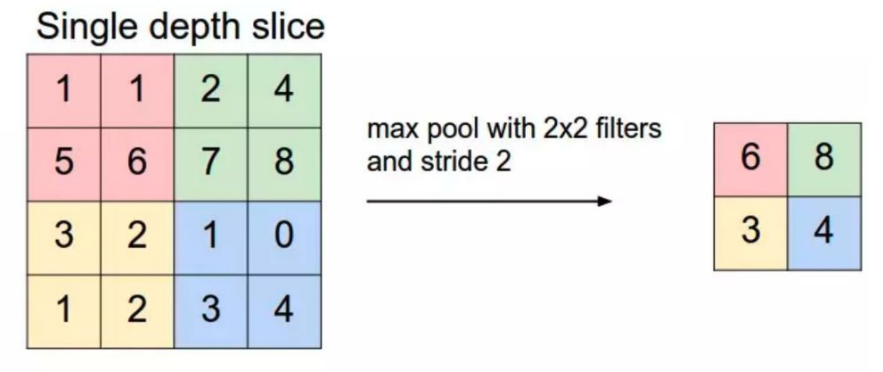


W o H (ancho o alto pueden ser sustituidos en esta expresión, F es el tamaño del filtro y S el tamaño del paso)

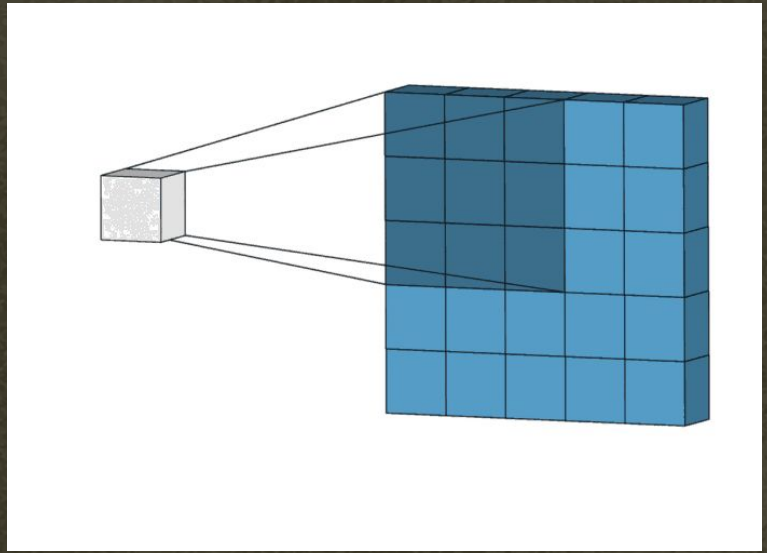
$$\text{Outputdim} = \frac{W - F}{S} + 1$$

Capa de Pooling

- Se utiliza generalmente después de una capa convolucional
- Se utiliza para disminuir el ancho y alto en las dimensiones de las capas internas.
- Reduce el número de parámetros, evita el overfit y disminuye el tiempo de cómputo.
- Una de las formas más utilizadas es el max pooling, ejemplo:



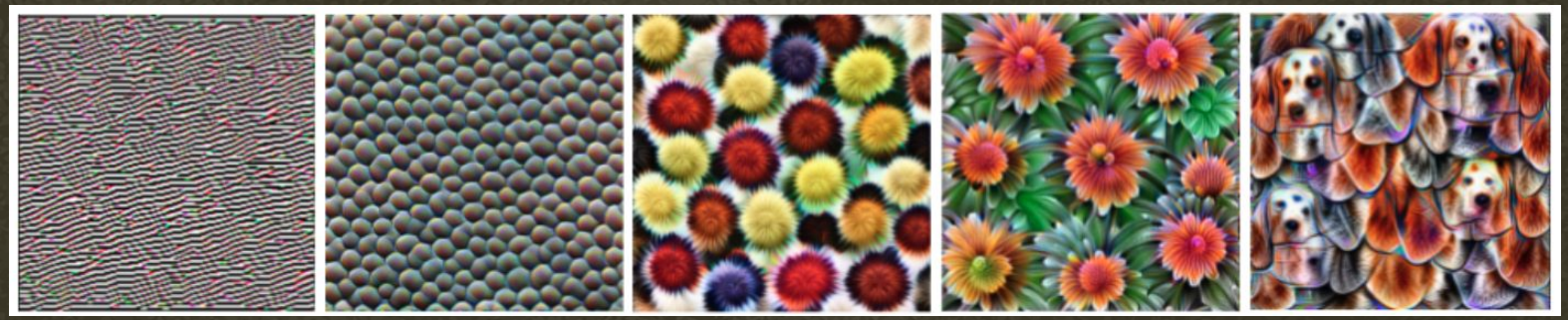
Ejemplo de Convolución



Capa Totalmente Conectada (Fully Connected)

- También es llamada de capa densa.
- Recibe la salida de todas las neuronas de una capa previa.
- Las capas de convolución y pooling se utilizan para extraer información relevante de las imágenes.
- Esta capa final, permite clasificar información de alto nivel de abstracción obtenida por las capas convolucionales.

Bordes Texturas Patrones Partes Objetos



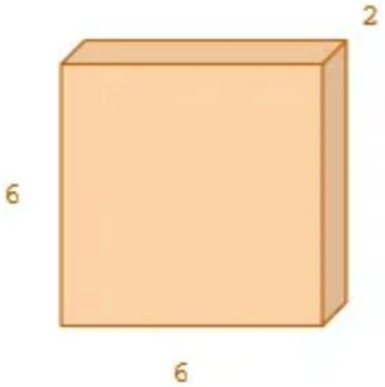
Aspectos Importantes

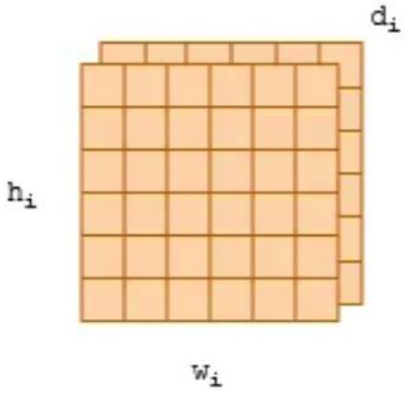
- Es posible utilizar más de 1 filtro en cada capa convolucional. Por ejemplo, podemos usar 7 filtros 5x5x3 lo que daria por salida un objeto de tamaño H x W x 6. Donde H y W, o las nuevas dimensiones tras la convolución pueden ser calculadas a través de una fórmula.
- La fórmula es dada de forma general por:

$$\text{Outputdim} = \frac{N - F + 2P}{S} + 1$$

Para una entrada N x N (cuadrada)

Ejemplo 1 de Convolución

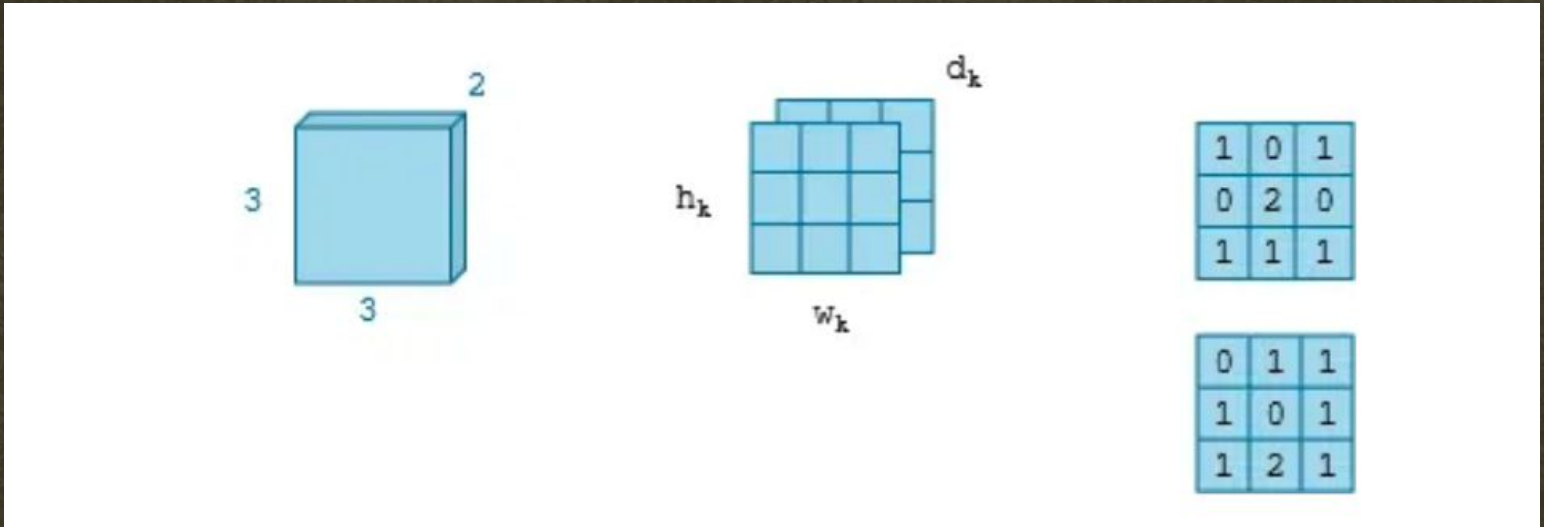




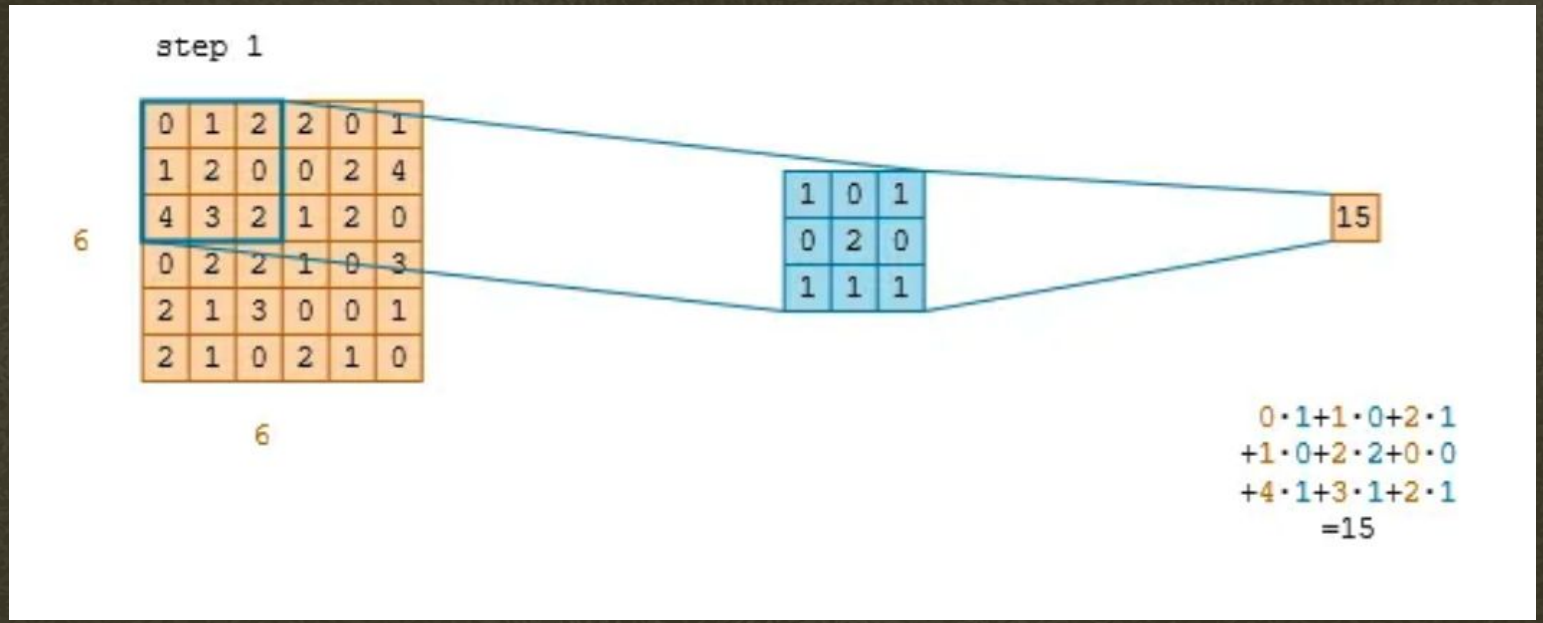
1	2	0	2	1	2
0	1	2	1	3	0
1	2	1	1	0	0
2	1	4	0	0	1
2	1	0	2	1	1
2	3	1	0	1	0

0	1	2	2	0	1
1	2	0	0	2	4
4	3	2	1	2	0
0	2	2	1	0	3
2	1	3	0	0	1
2	1	0	2	1	0

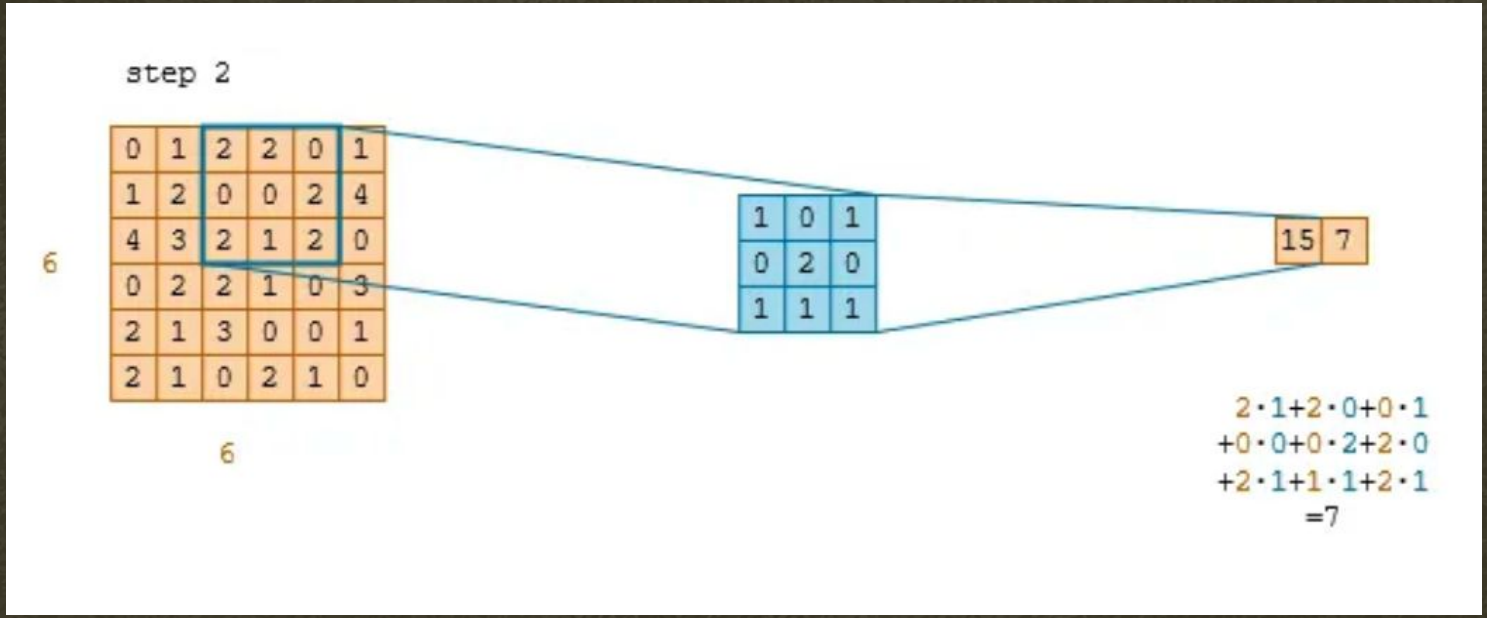
Ejemplo 1 de Convolución



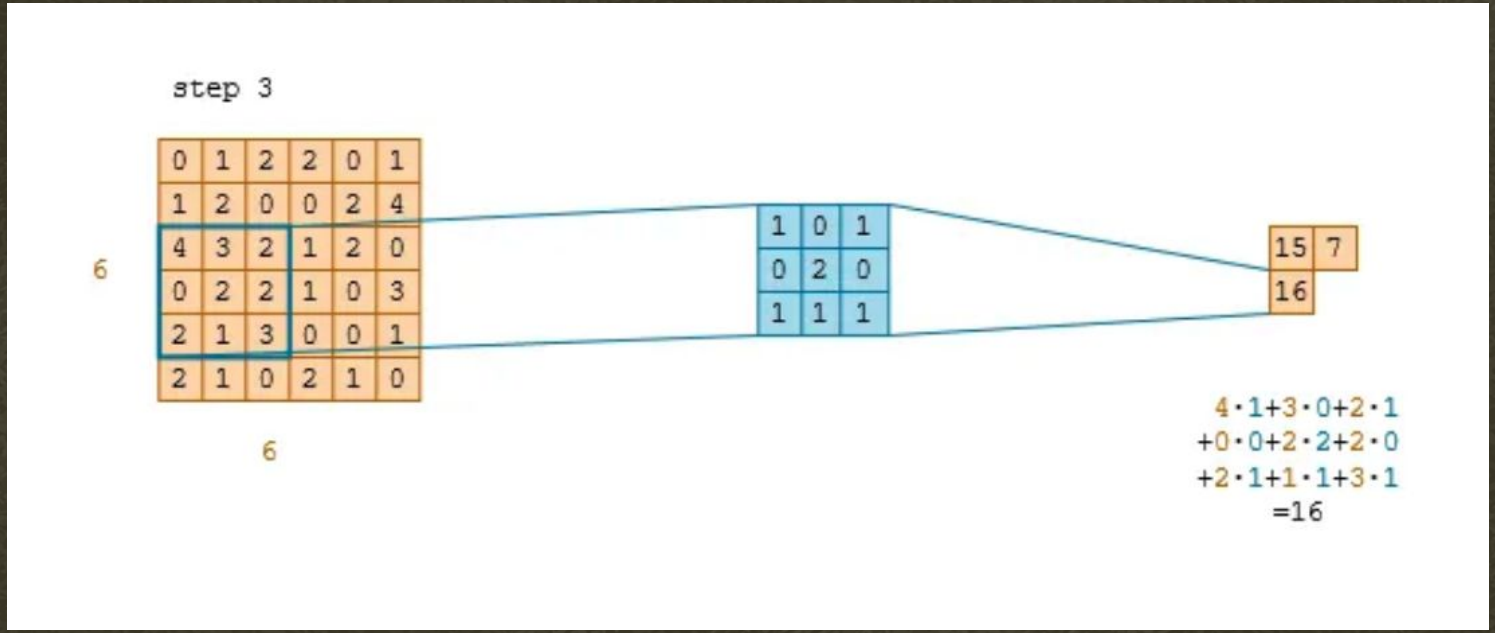
Ejemplo 1 de Convolución



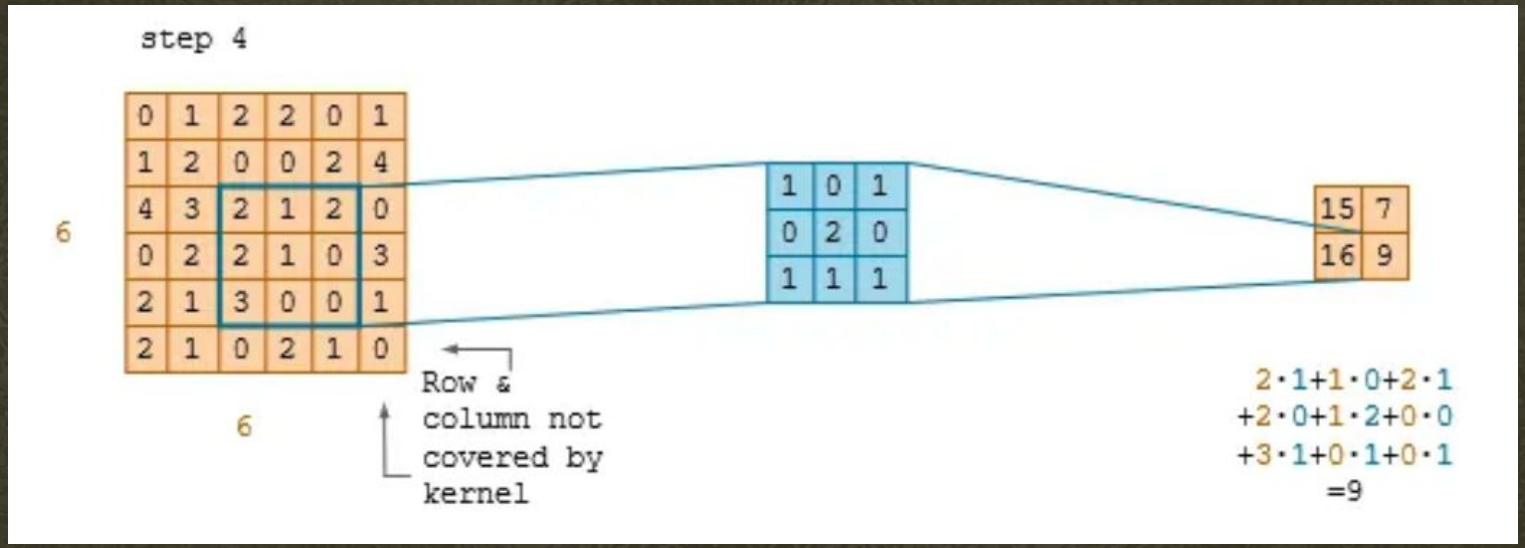
Ejemplo 1 de Convolución



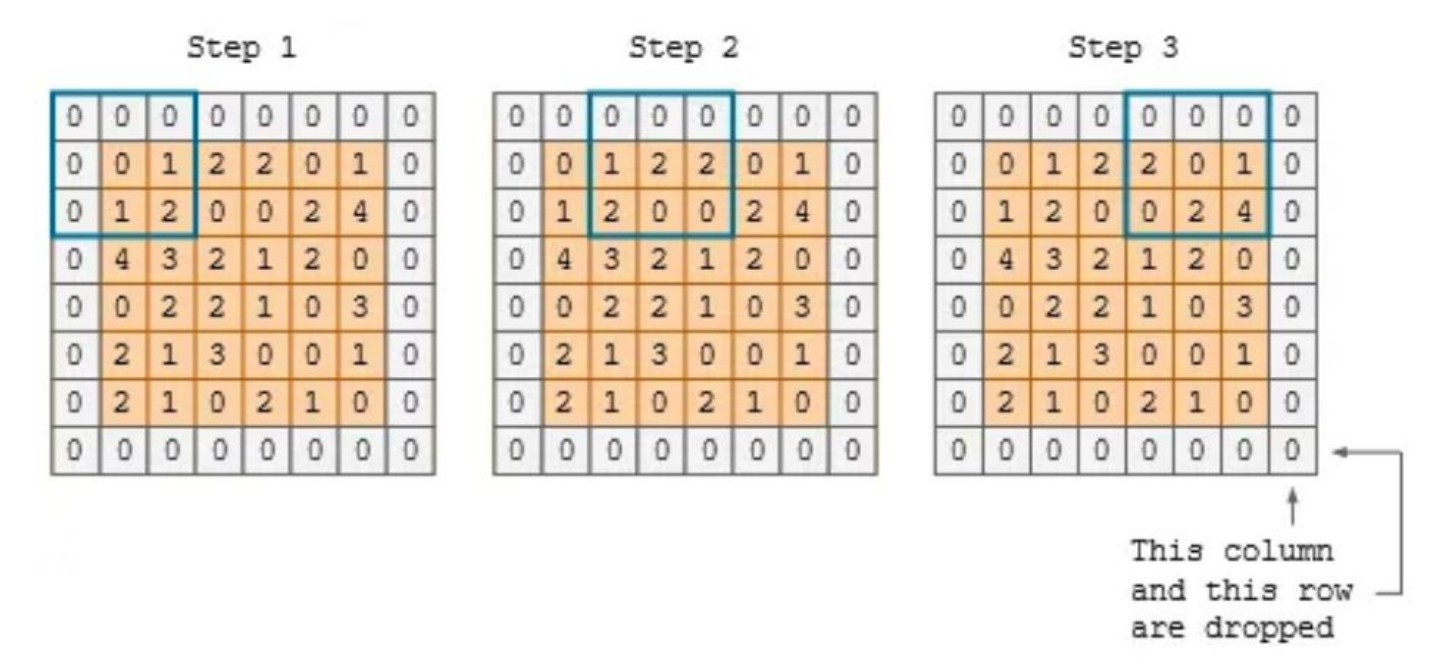
Ejemplo 1 de Convolución



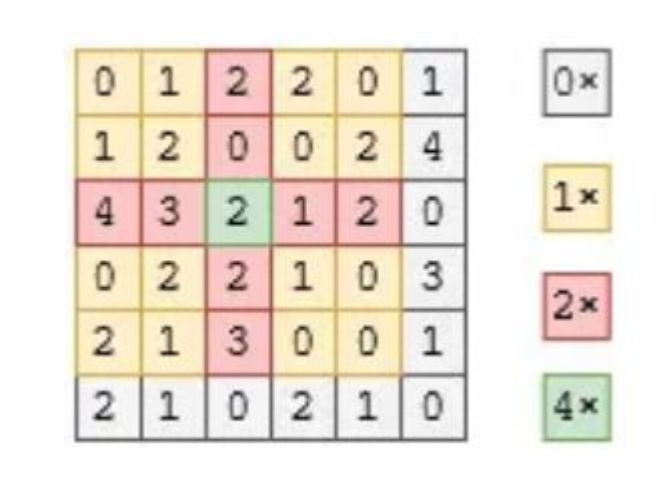
Ejemplo 1 de Convolución



Ejemplo 1 de Convolución Padding

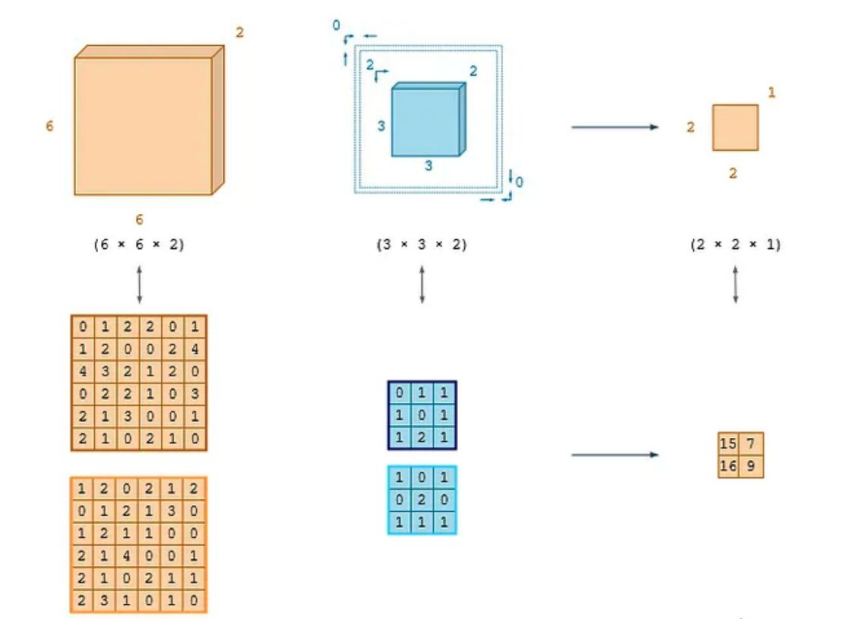


Ejemplo 1 de Convolución: representacion



Cantidad de veces que aparece una posición del arreglo bajo el cálculo de convolución

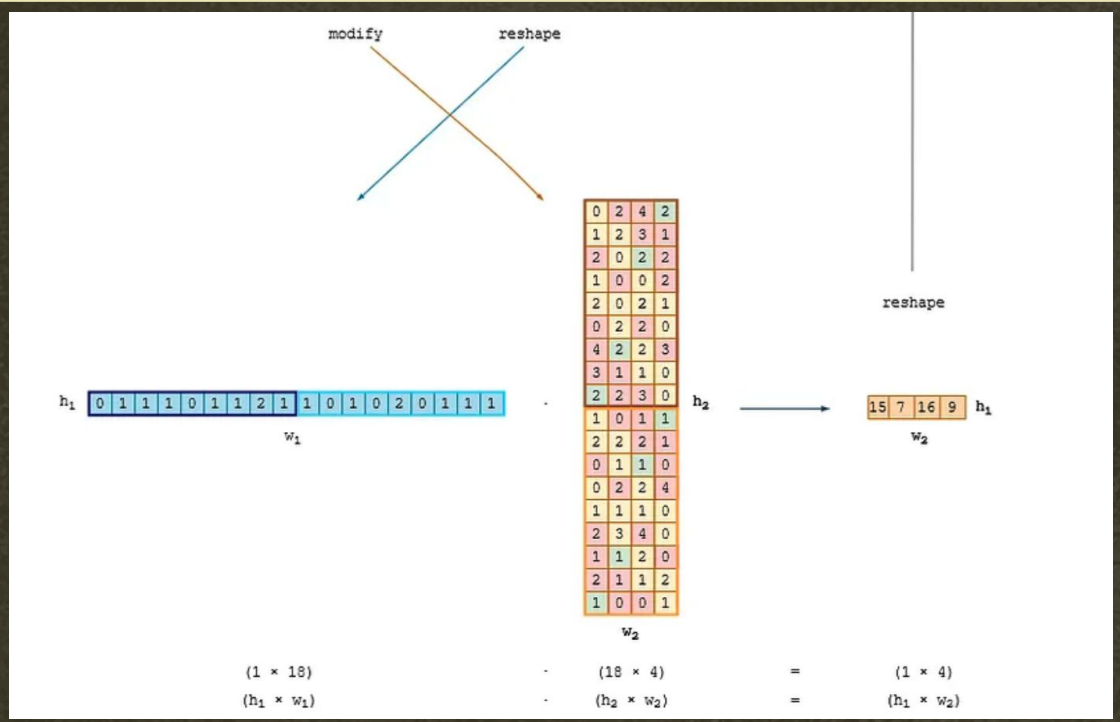
Ejemplo 1 de Convolución: representacion



Teniendo en cuenta que estamos dando pasos de a 2, es posible representar la operación de convolución como producto de matrices!

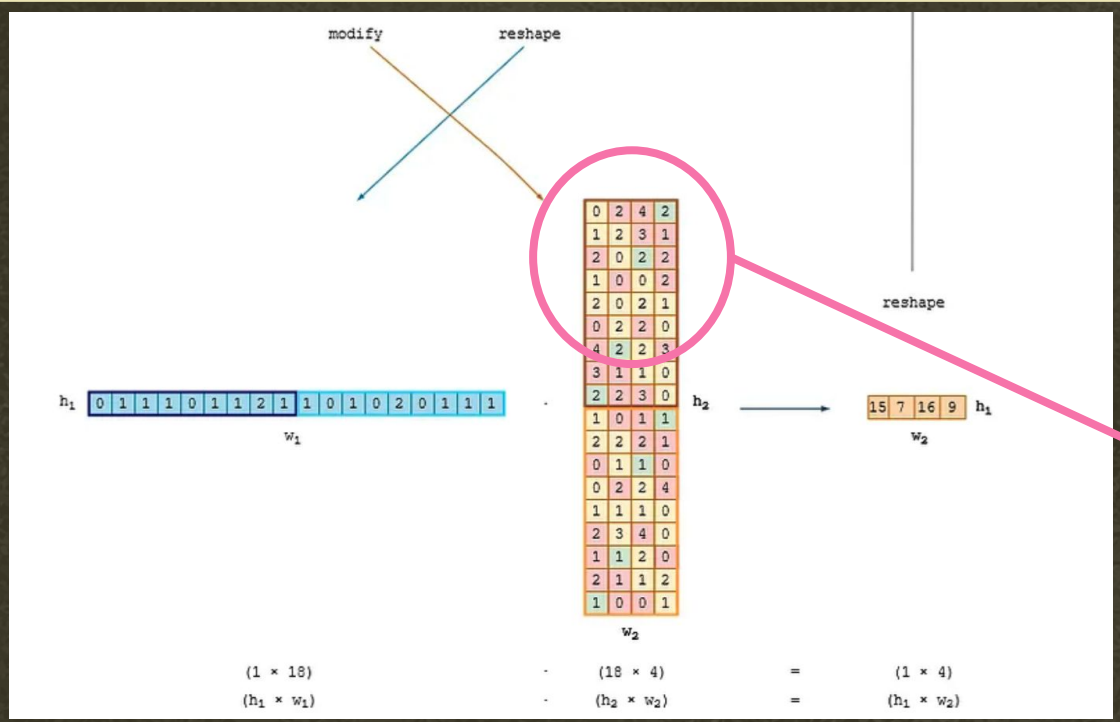
Vemos que tenemos 2 filtros de 3x3

Ejemplo 1 de Convolución: representacion



Teniendo en cuenta que estamos dando pasos de a 2, es posible representar la operación de convolución como producto de matrices!

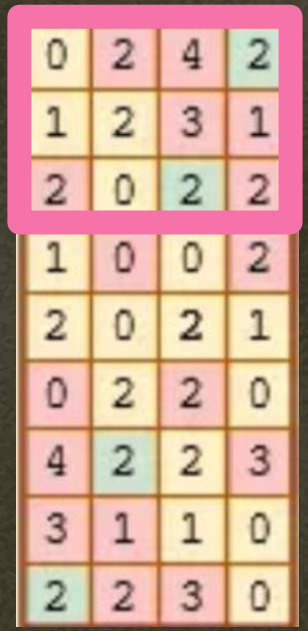
Ejemplo 1 de Convolución: representacion



Los colores representan la frecuencia con que aparecen los pixeles en el recorrido de los kernels.
La matriz modificada del centro se transformó para hacer coincidir las dimensiones con los kernels.

Los valores recorrido por el kernel Por fila, se colocan de forma vertical. En el primer canal Alcanzamos a recorrer en la primera fila los valores: 0 - 1 - 2 Y 2 - 2 - 0. Luego pasamos a la fila 3, con valores: 4 - 3 - 2 etc.

Ejemplo 1 de Convolución: representacion

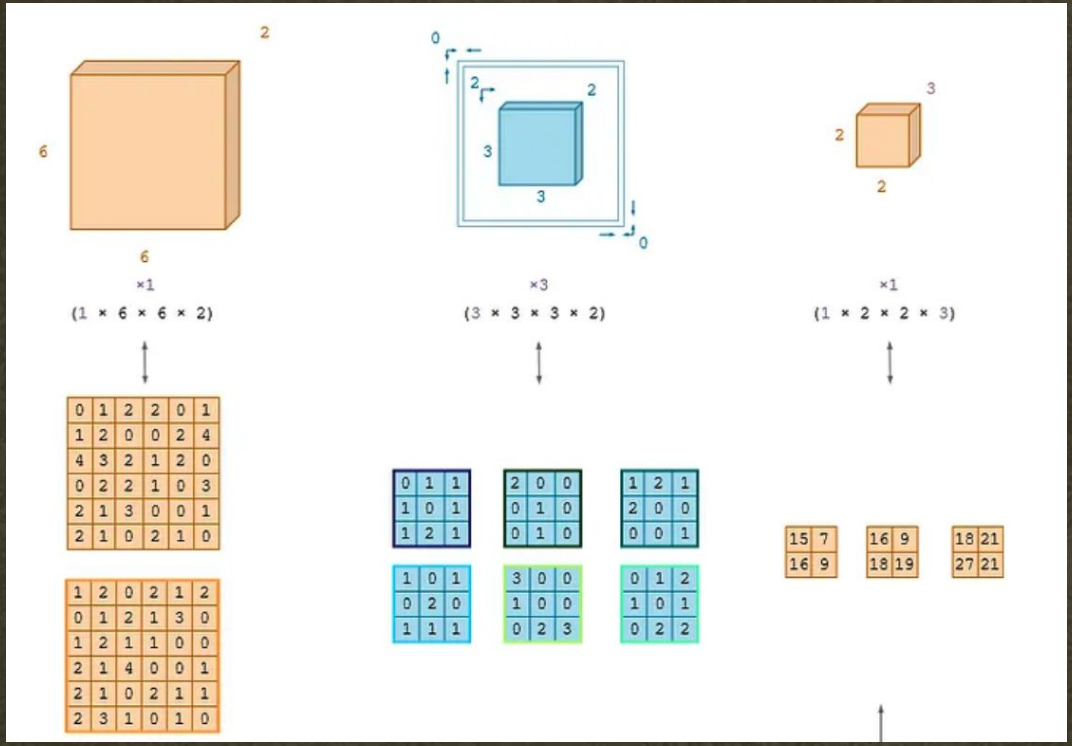


Primera fila de cada canal recorrida por los filtros

Ejemplo 1 de Convolución: representacion

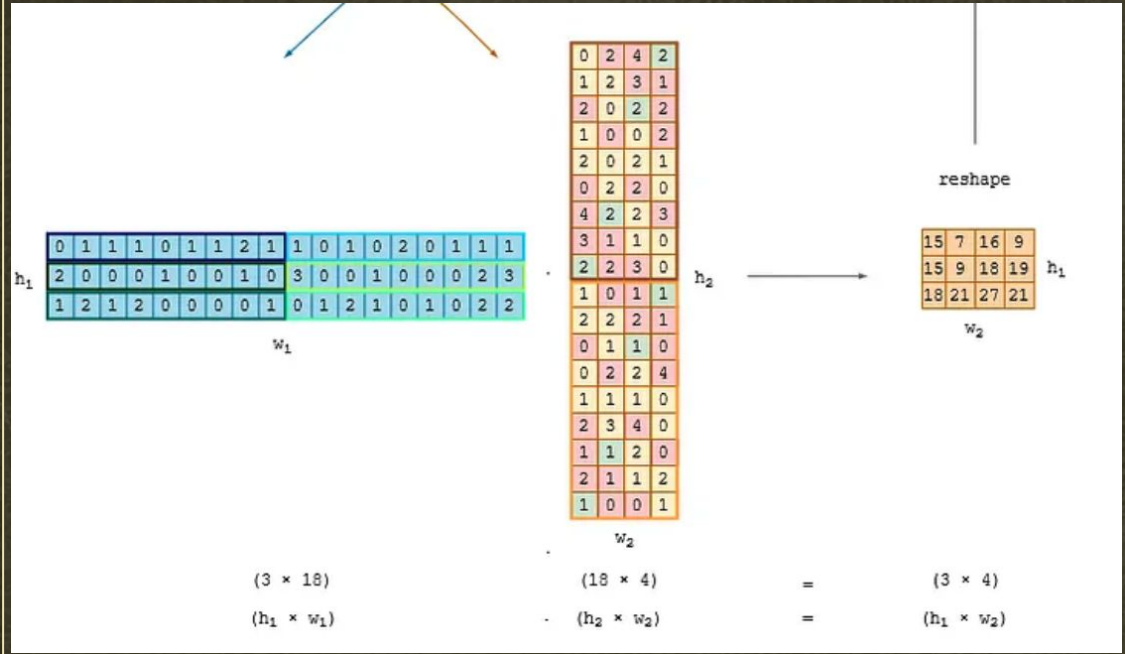
Segunda fila de cada canal recorrida por los filtros

Ejemplo 1 de Convolución: múltiples instancias



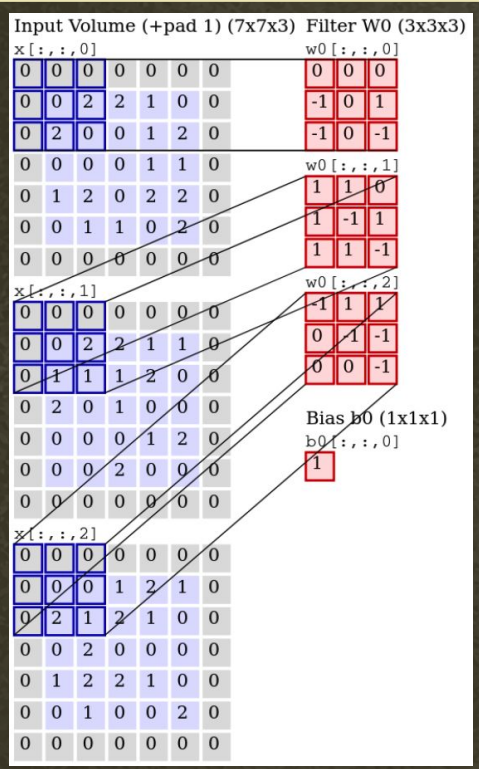
3 instancias de kernels de tamaño 3x3 con dos canales

Ejemplo 1 de Convolución: múltiples instancias



3 instancias de kernels de tamaño 3x3 con dos canales

Ejemplo de Convolución



Filtro 2

w1[:,:,0]

0	-1	1
1	-1	-1
-1	1	1

w1[:,:,1]

1	0	0
1	1	0
1	-1	0

w1[:,:,2]

1	-1	1
0	-1	0
-1	0	1

Bias b1 (1x1x1)

b1[:,:,0]

0

Bias

Salida

o[:,:,0]

4	-4	0
-3	-1	2
4	-1	1

o[:,:,1]

0	3	4
4	1	-1
1	5	-4

Tamaño de paso: 2 (Stride)

Ejemplo Tensorflow-Keras + CIFAR 10

NOTEBOOK LINK

Referencias principales

- Introduction to Probability for Data Science, Stanley H. Chan, 2021, Michigan Publishing. ISBN 978-1-60785-747-1
- Python Crash Course: a hands -on project-based introduction to programming.
- Second Edition. ISBN-13: 978-1593279288
- Deep Learning, Ian Good Fellow, Yoshua Bengio and Aaron Courville. MIT Press, 2016. ISBN, 0262035618, 9780262035613
- Neural Networks and Learning Machines, Haykin Simon, 2008. Third Edition
- ISBN 10: 0131471392 ISBN 13: 9780131471399.
- https://github.com/musikalkemist/AudioSignalProcessingForML
- Approaching (ALMOST) Any Machine Learning Problem, Abhishek Thakur, 2019.
- Deep Learning with Python, François Chollet, 2021, Second Edition, Manning Shelter Island.
- Python DataScience Handbook, Essential Tools for Working with Data. Jake VanderPlas O'Reilly, 2016, First Edition.
- https://www.coursera.org/specializations/machine-learning-introduction
- https://github.com/keunwoochoi/dl4mir
- https://christophm.github.io/interpretable-ml-book/cnn-features.html
- https://arxiv.org/pdf/1511.08458.pdf

Referencias de esta presentación

- https://christophm.github.io/interpretable-ml-book/cnn-features.html
- https://arxiv.org/pdf/1511.08458.pdf

Gracias!

¿Preguntas?

rodolfolobo@ug.uchile.cl