Convoluciones

Visión por Computador, curso 2024-2025

Silvia Martín Suazo, silvia.martin@u-tad.com

26 de septiembre de 2024

U-tad | Centro Universitario de Tecnología y Arte Digital

Transformaciones basadas en ventanas

Características de las transformaciones basadas en ventanas

A diferencia de los casos vistos anteriormente, las transformaciones basadas en ventanas modifican el valor de cada píxel dependiendo del nivel de intensidad de sus píxeles vecinos.

De esta manera se consigue que la transformación tenga un contexto espacial de la situación de cada píxel de la imagen.

Dentro de este conjunto de transformaciones se encuentran:

- Convolución 1-D
- Convolución 2-D
- Filtrado paso alto
- Filtrado de paso bajo
- Filtrado no lineal

Operación de convolución

Sean $f(x)$ y $g(y)$ dos funciones discretas tales que $x \in \{0, \dots, a-1\}, y \in \{0, \dots, b-1\}$.

La operación de convolución $s(r) = f * g$ con $r \in \{0, \dots, s-1\}; s = a + b - 1$

f	2	1	4	0	1

g	0	1	-1

s = f*g			

Operación de convolución

Sean $f(x)$ y $g(y)$ dos funciones discretas tales que $x \in \{0, \dots, a-1\}, y \in \{0, \dots, b-1\}$.

Matemáticamente, la operación de convolución de f con g se define a través de la siguiente función:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{1}$$

donde el filtro se invierte a $h[-n]$

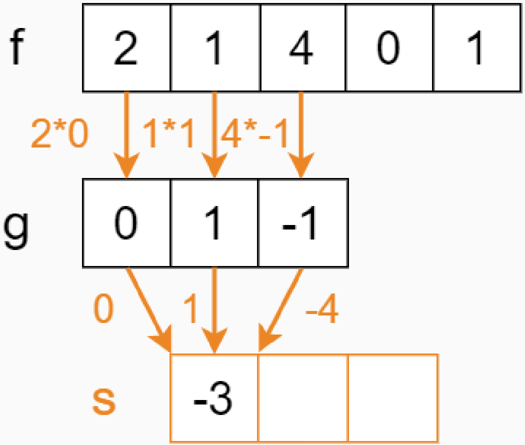
Operación de convolución

$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$

f	2	1	4	0	1
g	0	1	-1		
s = f*g					

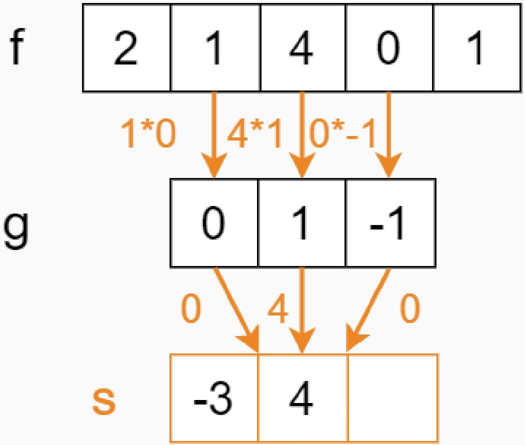
Operación de convolución

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$



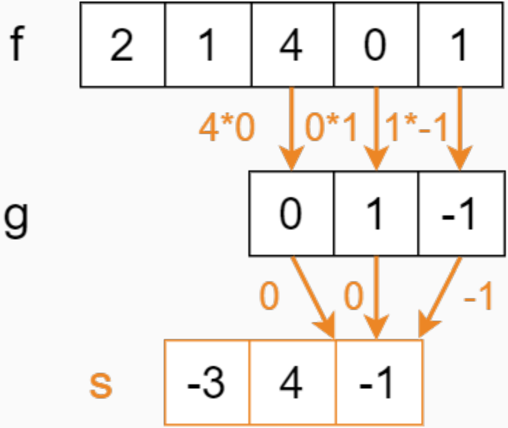
Operación de convolución

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$



Operación de convolución

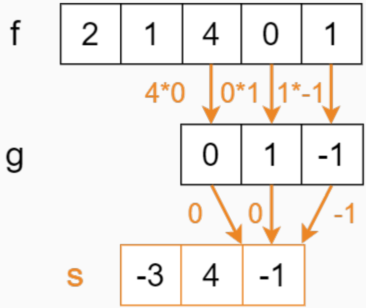
$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$



Convolución 2-D

Operación de convolución

Como anteriormente se ha visto, la operación de convolución permite operar entre dos vectores obteniendo como resultado otro vector, resultado de la convolución.



Hacia la convolución 2-D

Hasta ahora se ha visto cómo operar unidimensionalmente, sin embargo para poder aplicar convoluciones a imágenes se ha de poder operar en un espacio bidimensional.

f

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

g

0	1	-1
0	1	2
0	1	0

Convolución 2-D

f

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

g

0	1	-1
0	1	2
0	1	0

s

4	

Convolución 2-D

f

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

g

0	1	-1
0	1	2
0	1	0

s

4	8

Convolución 2-D

f

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

g

0	1	-1
0	1	2
0	1	0

s

4	8
5	

Convolución 2-D

f

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

g

0	1	-1
0	1	2
0	1	0

s

4	8
5	5

Convolución 2-D

f

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

g

0	1	-1
0	1	2
0	1	0

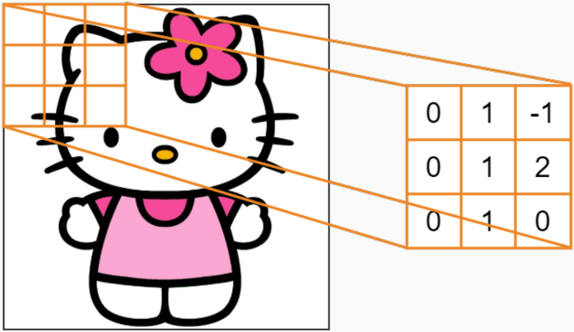
s

4	8
5	5

Convolución en el procesamiento de imágenes

Las convoluciones son la operación más típica para procesar imágenes.

La versatilidad que proporcionan consiguen que se puedan obtener objetivos muy diversos con su uso.



Aplicaciones que usan convolución

La convolución es la operación básica que permite realizar operaciones tales como:

- Eliminar ruido
- Resaltar estructuras
- Detección de puntos de interés
- Clasificación
- Segmentar elementos

Propiedades de la convolución

Propiedades

La operación de convolución cumple las siguientes propiedades matemáticas:

- Conmutativa

$$h * g = g * h \tag{2}$$

Propiedades

La operación de convolución cumple las siguientes propiedades matemáticas:

- Conmutativa
- Asociativa

$$h * (f * g) = (h * f) * g \tag{3}$$

Propiedades

La operación de convolución cumple las siguientes propiedades matemáticas:

- Conmutativa
- Asociativa
- Lineal

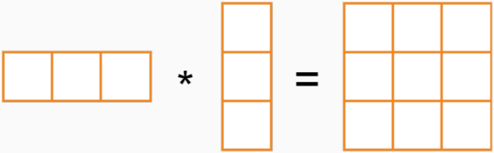
$$h * (f + g) = (h * f) + (h * g) \tag{4}$$

Propiedades

La operación de convolución cumple las siguientes propiedades matemáticas:

- Conmutativa
- Asociativa
- Lineal
- Separabilidad

Toda convolución con un filtro de dimensiones *mxn* tiene dos convoluciones de *mx*1 y 1*xn* que producen el mismo resultado.



Parámetros de la convolución

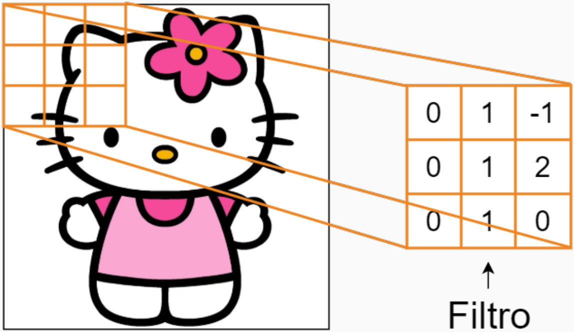
Parámetros de una convolución

La operación de convolución tiene distintos parámetros que cambian su comportamiento:

- Filtro o *kernel*
- *Padding*
- *Strides* o paso

Filtro

El filtro, también conocido como *kernel*, de una convolución corresponde con la matriz (en el caso bidimensional) que realiza la operación de convolución.

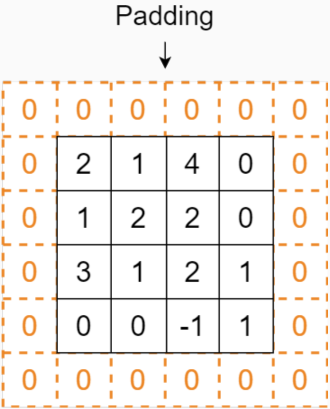


Pese a cumplirse la propiedad conmutativa, por convenio se define al filtro como la matriz más pequeña. Normalmente los filtros no suelen pasar de la decena de longitud en sus dimensiones.

Padding

A la hora de realizar una convolución, se puede definir el “marco de ceros” que rodea a la matriz que se está convolucionando.

Este parámetro se llama *padding* y sirve para mantener la dimensión de la matriz tras realizar la convolución.



Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

f

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

g

0	1	0
0	1	0
0	1	0

s

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (1, 1)

f

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

g

0	1	0
0	1	0
0	1	0

s

4			

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (1, 1)

f

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

g

0	1	0
0	1	0
0	1	0

s

4	8		

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (1, 1)

f

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

g

0	1	0
0	1	0
0	1	0

s

4	8	1	

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (1, 1)

f

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

g

0	1	0
0	1	0
0	1	0

s

4	8	1	3

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (1, 1)

f

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

g

0	1	0
0	1	0
0	1	0

s

4	8	1	3
3			

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (1, 1)

f

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

g

0	1	0
0	1	0
0	1	0

s

4	8	1	3
3	3	2	1
3	2	4	0
2	1	6	0

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (2, 2)

f

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

g

0	1	0
0	1	0
0	1	0

s

Cabe destacar que para strides mayores a $(1, 1)$ se produce una reducción de dimensiones.

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (2, 2)

f

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

g

0	1	0
0	1	0
0	1	0

s

4	

Cabe destacar que para strides mayores a $(1, 1)$ se produce una reducción de dimensiones.

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (2, 2)

f

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

g

0	1	0
0	1	0
0	1	0

s

4	1

Cabe destacar que para strides mayores a $(1, 1)$ se produce una reducción de dimensiones.

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (2, 2)

f

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

g

0	1	0
0	1	0
0	1	0

s

4	1
3	

Cabe destacar que para strides mayores a $(1, 1)$ se produce una reducción de dimensiones.

Strides

Los *strides* o pasos de una convolución corresponden con el número de casillas que se desplaza horizontal y verticalmente el filtro al realizar la convolución.

Stride = (2, 2)

f

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

g

0	1	0
0	1	0
0	1	0

s

4	1
3	4

Cabe destacar que para strides mayores a $(1, 1)$ se produce una reducción de dimensiones.

Dimensionalidad de salida

Una vez conocidos los parámetros de una convolución, podemos definir la fórmula para calcular las dimensiones que tendrá la imagen de salida tras operar con ella, a través de la siguiente fórmula:

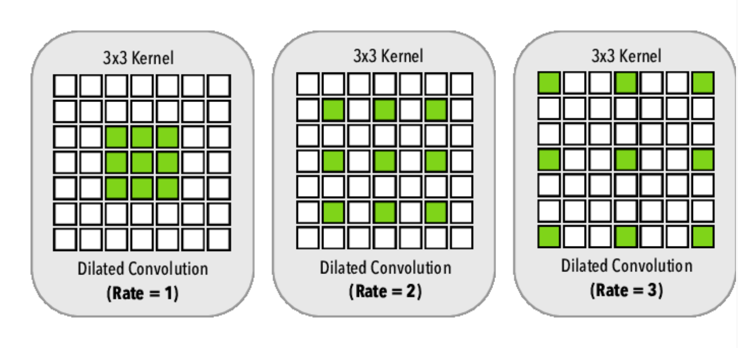
$$Dim_{out} = (\frac{Dim_{in} - Dim_{ker} + 2 * padding}{strides}) + 1 \tag{5}$$

donde *Dim* indica las dimensiones de cada elemento, siendo *out* la imagen de salida, *in* la imagen de entrada y *ker* en kernel.

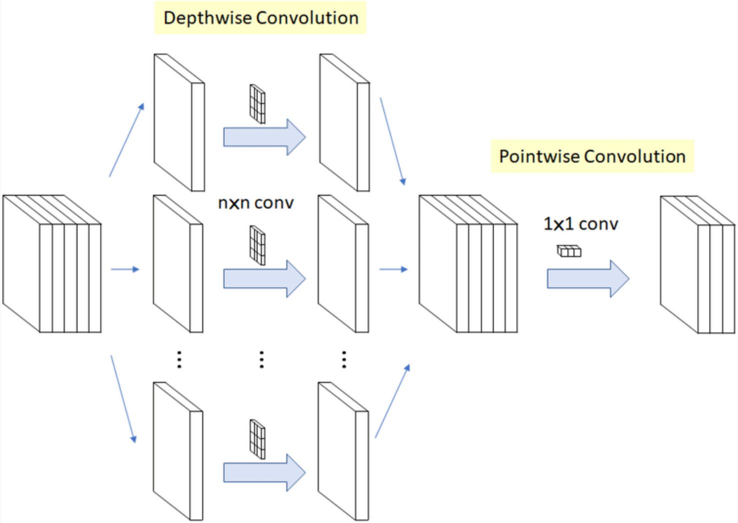
Otras convoluciones

- Convolución 3D: mismo procedimiento sobre datos 3D como videos.
- Convolución dilatada: introduce dilataciones en el kernel para aumentar el campo receptivo.
- Convolución profunda (Convolutional Depthwise Separable): se divide la convolución en convolución de canal (se aplican convoluciones a cada canal independientemente) y convolución punto a punto (1x1).

Otras convoluciones



Otras convoluciones



Notebook ejemplo de convoluciones

- 02.03-Introduccion_Convoluciones.ipynb

Referencias i