Aprendizaje profundo

Capas convolucionales

Gibran Fuentes-Pineda

Septiembre 2023

Sesgo inductivo relacional

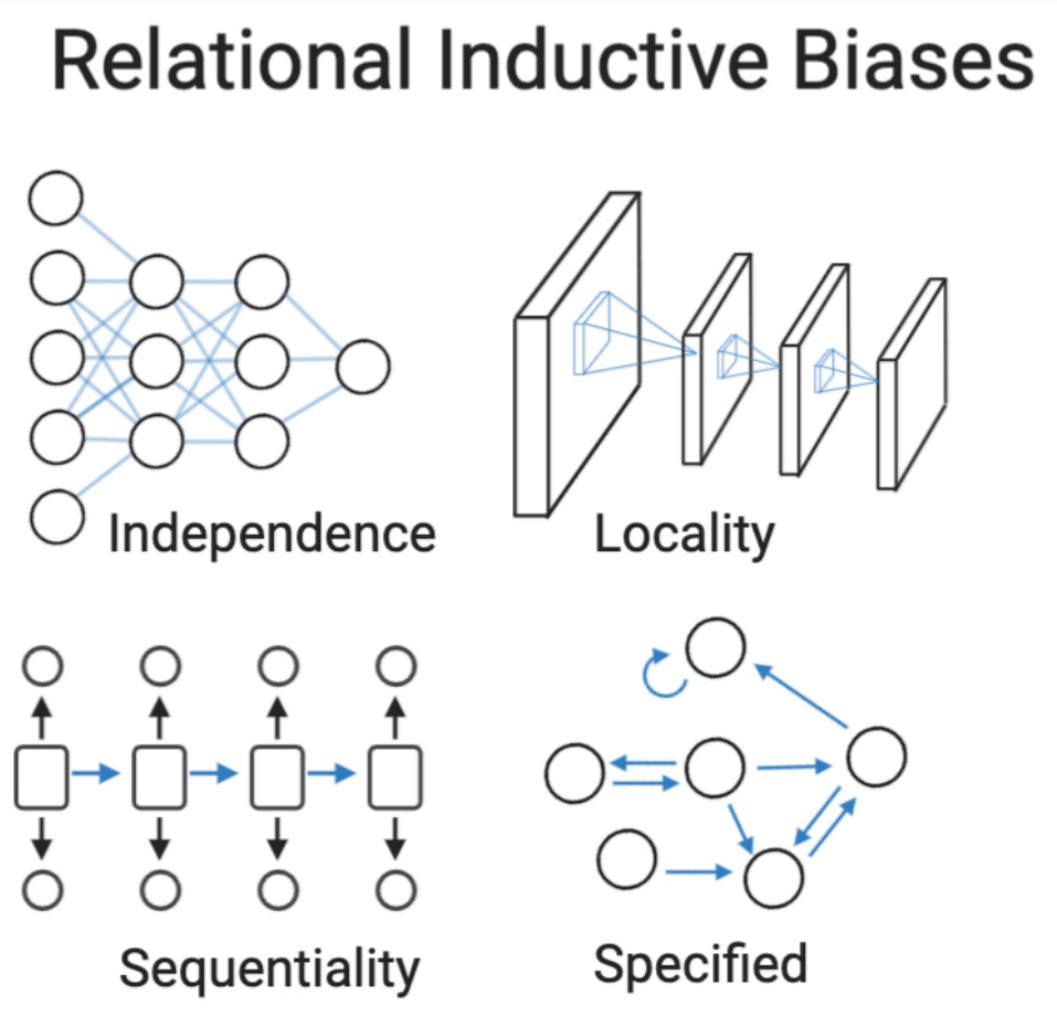
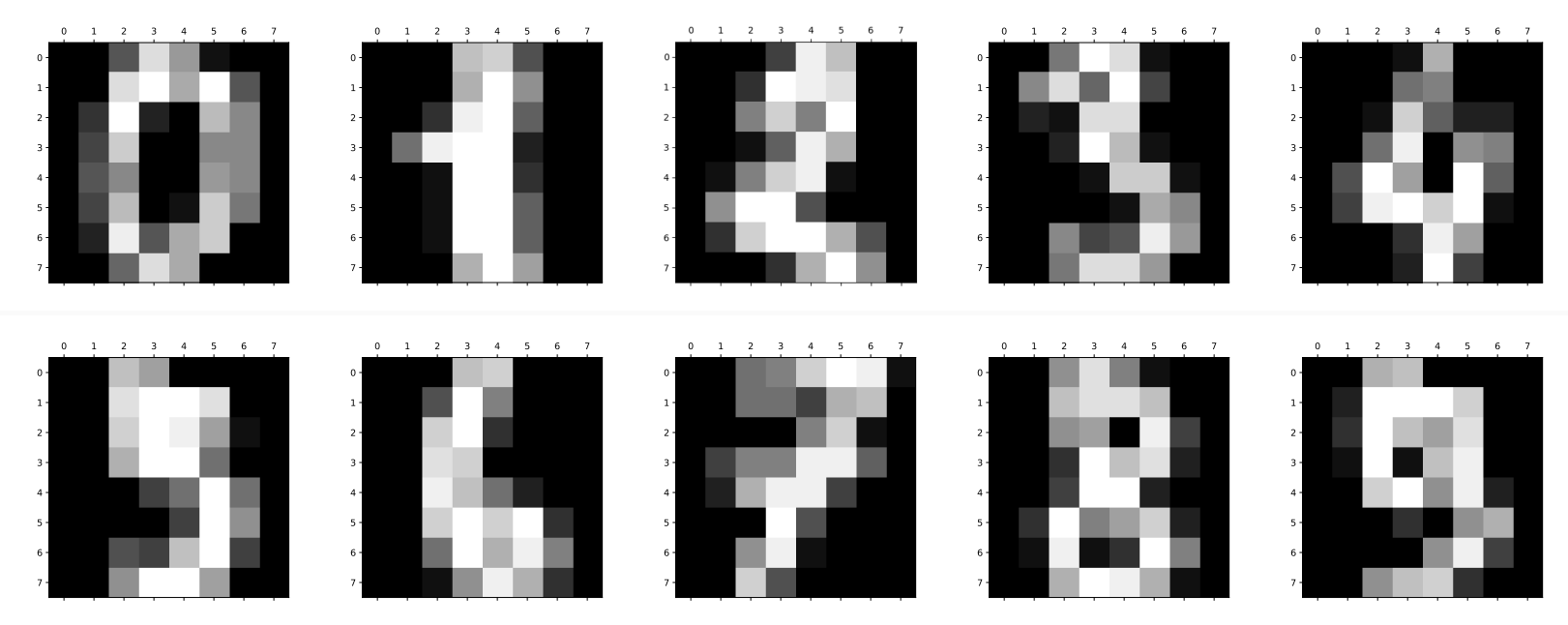


Imagen tomada de Sam Finlayson. Induction, Inductive Biases, and Infusing Knowledge into Learned Representations. Junio 22, 2022.

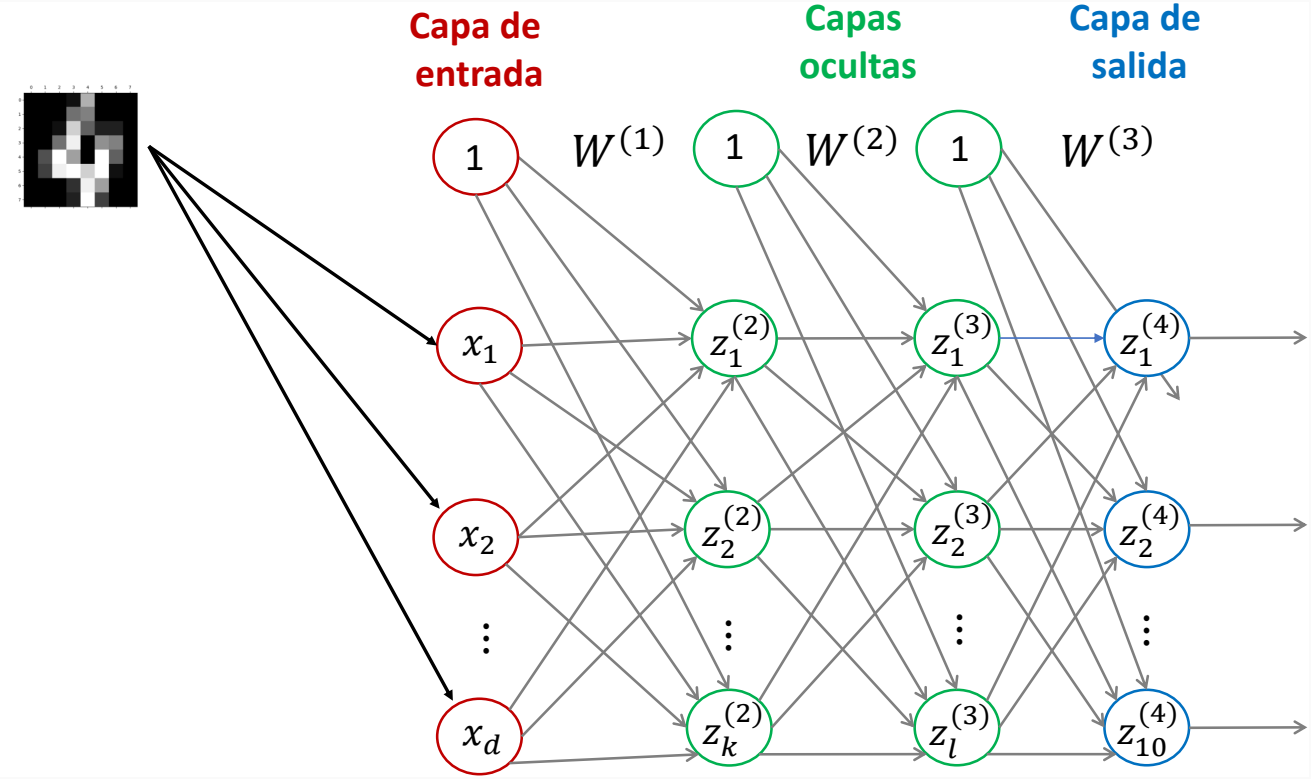
Clasificación de imágenes

- Por ejemplo, clasificar dígitos escritos a mano



Clasificación de imágenes con PMC

- Redes perceptrón multicapa tienen muchos parámetros
  - Por ej., si tenemos imágenes de 32×32 y 1 red con 1 sóla capa oculta con 10 neuronas tendríamos $(32 \times 32 \times 10) + 1$ pesos



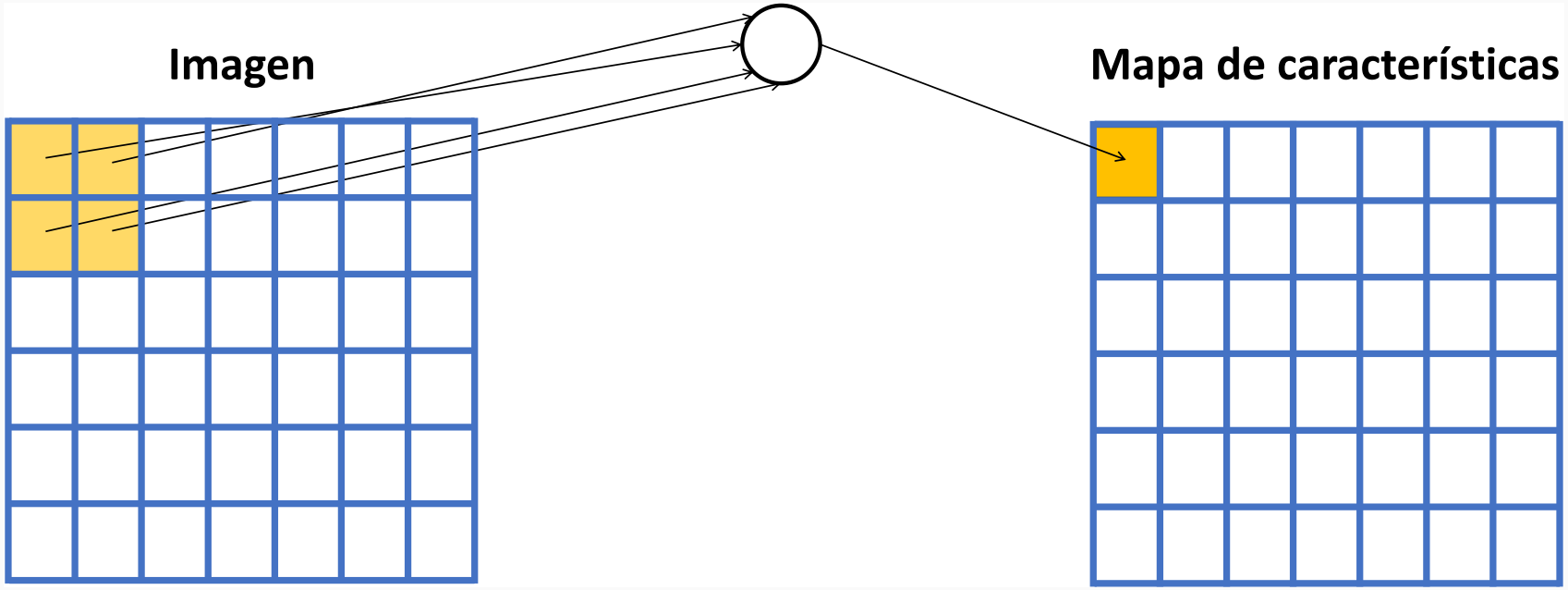
Capa convolucional

- Pueden verse como un caso especial de una capa densa con 2 variaciones
  1. Conectividad local (dispersa)
  2. Pesos compartidos

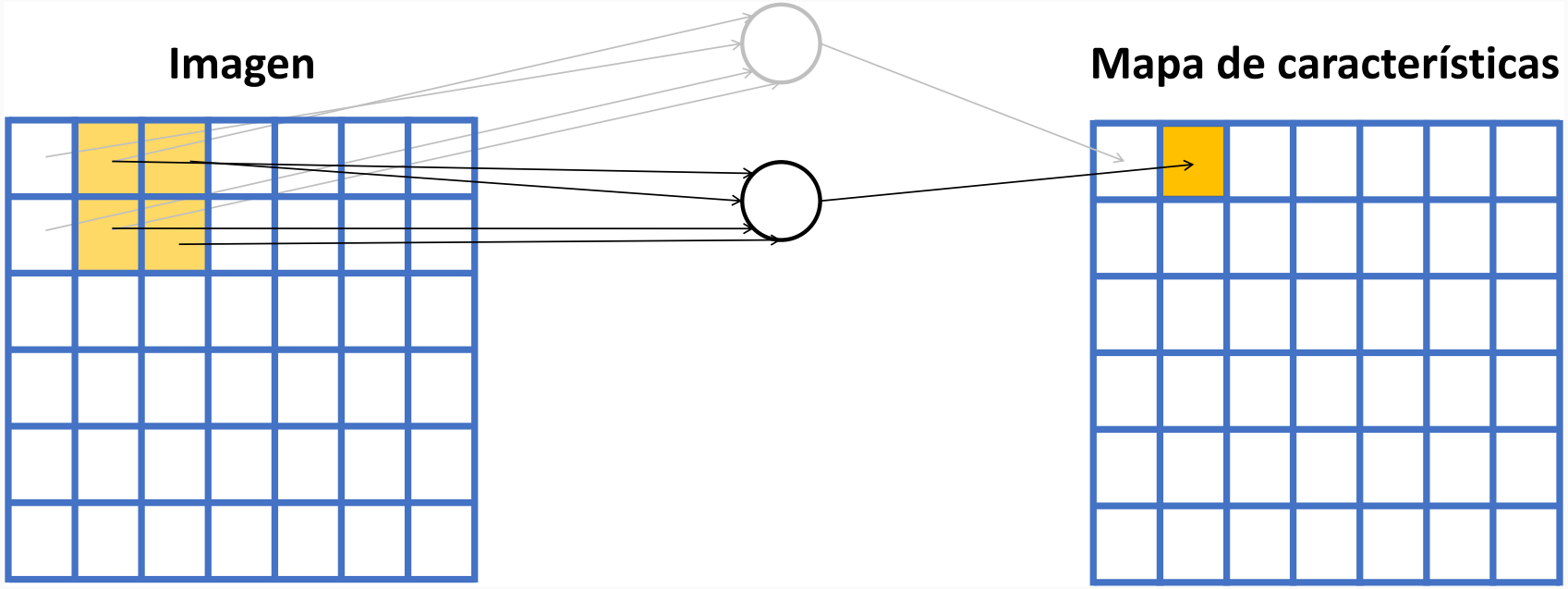
- Las representaciones obtenidas son más eficientes

- La operación es equivariante

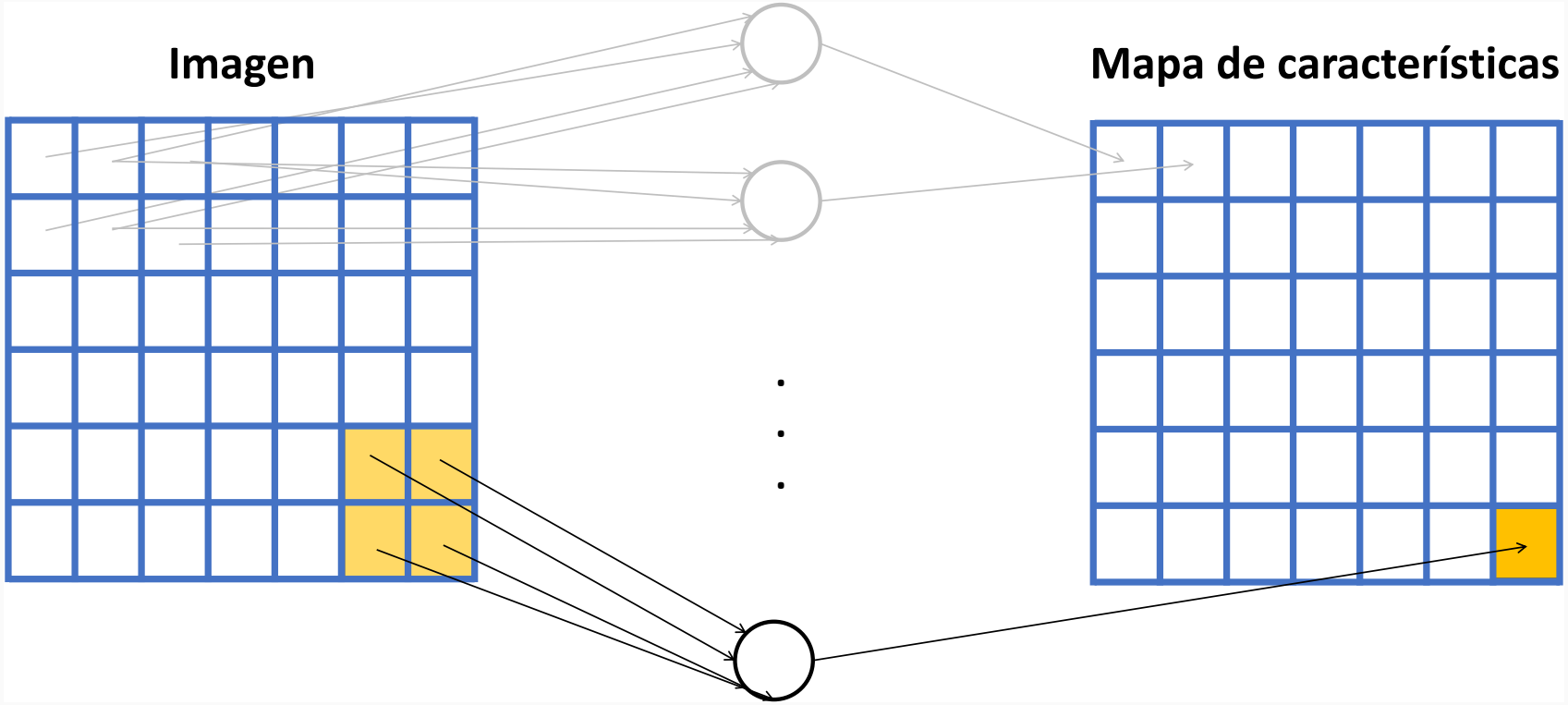
Conectividad local (1)



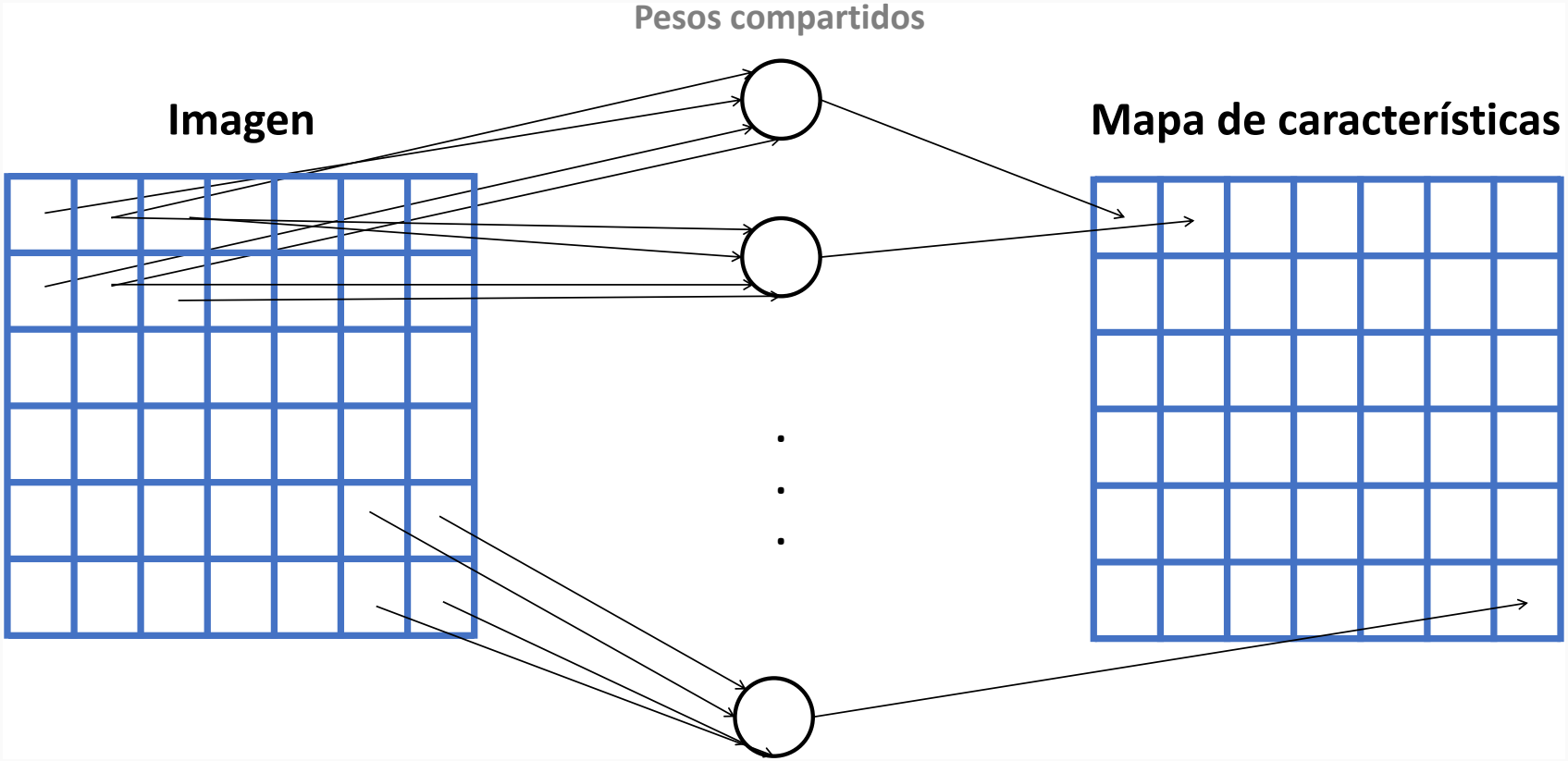
Conectividad local (2)



Conectividad local (3)



Pesos compartidos



Operación de convolución

- Convolución en 1 dimensión
  - Continua $s(t) = (x * k)(t) = \int_m x(m)k(t-m)dm$
  - Discreta $s[i] = (x * k)[i] = \sum_{m=-\infty}^{\infty} x[m]k[i-m]$
- Convolución en 2 dimensiones (discreta)
  - $S[i,j] = (I * K)[i,j] = \sum_m \sum_n I[m,n]K[i-m, j-n]$
  - $S[i,j] = (K * I)[i,j] = \sum_m \sum_n I[i-m, j-n]K[m,n]$
- Correlación cruzada (discreta)
  - $S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)$

Operación de convolución

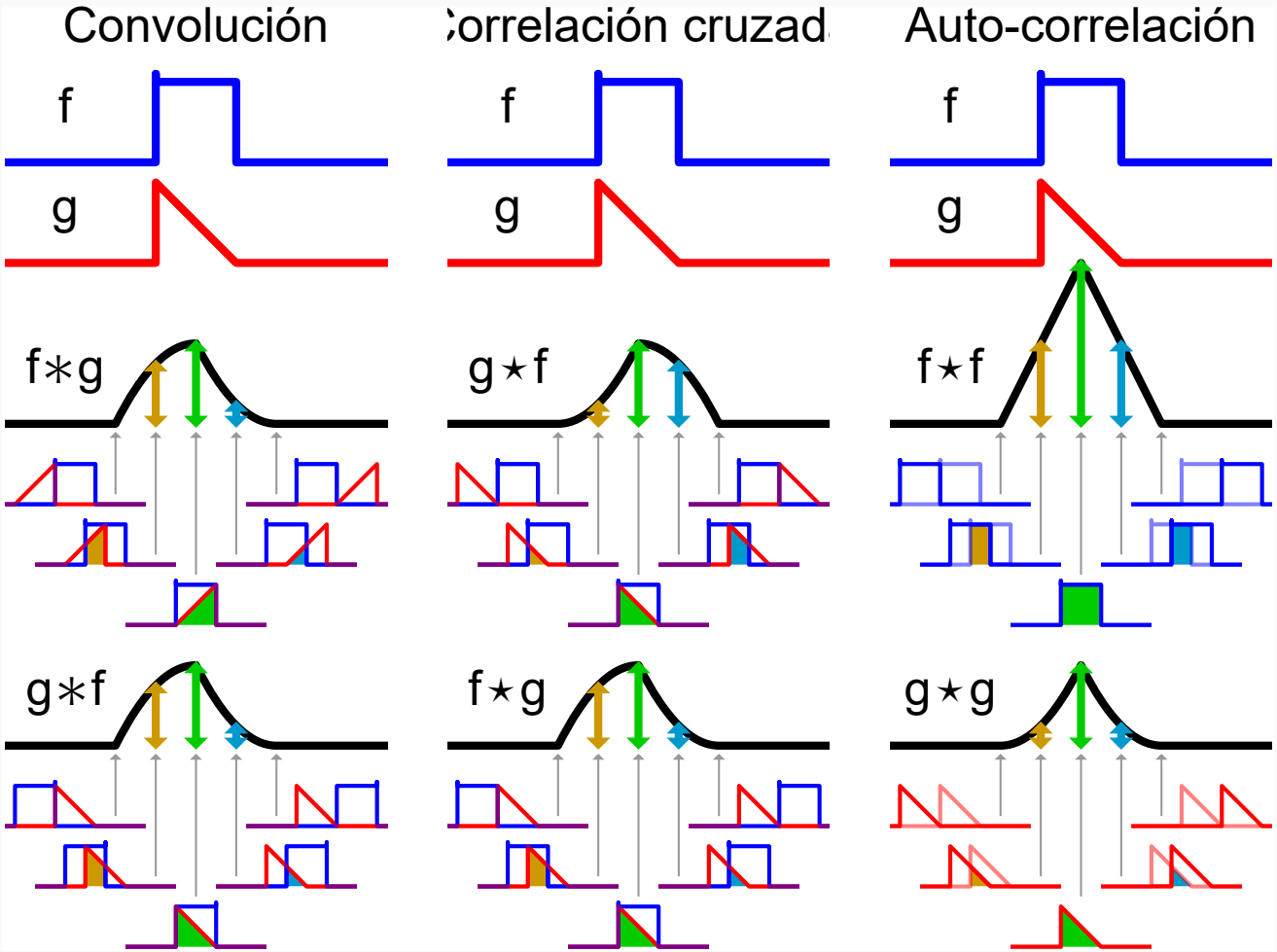
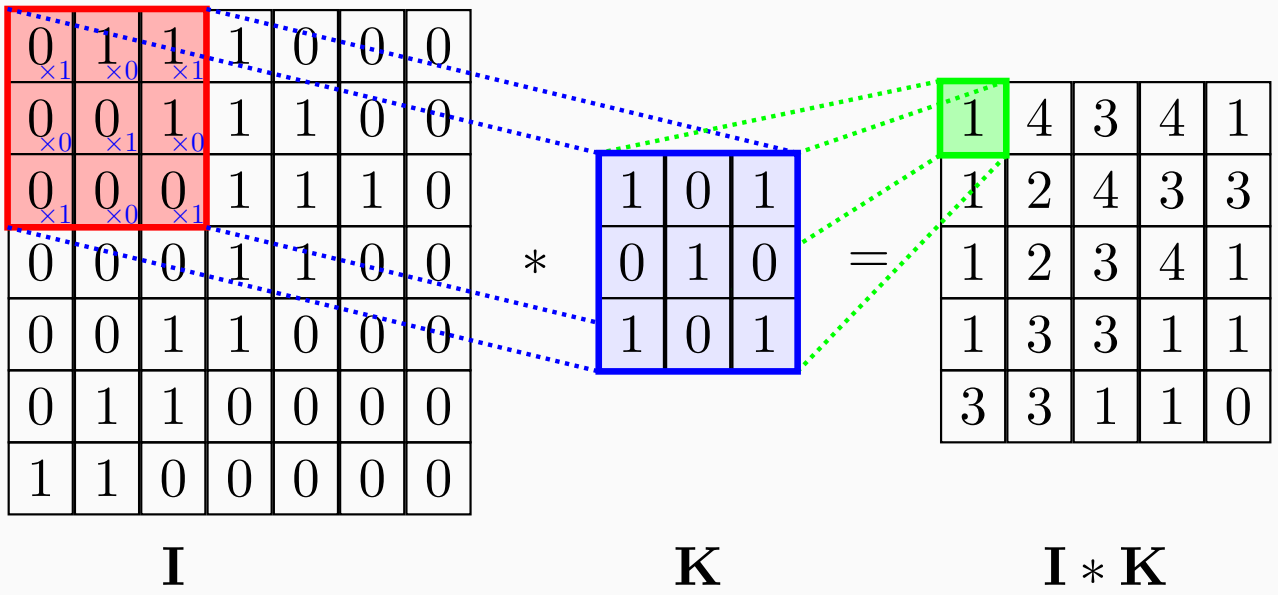
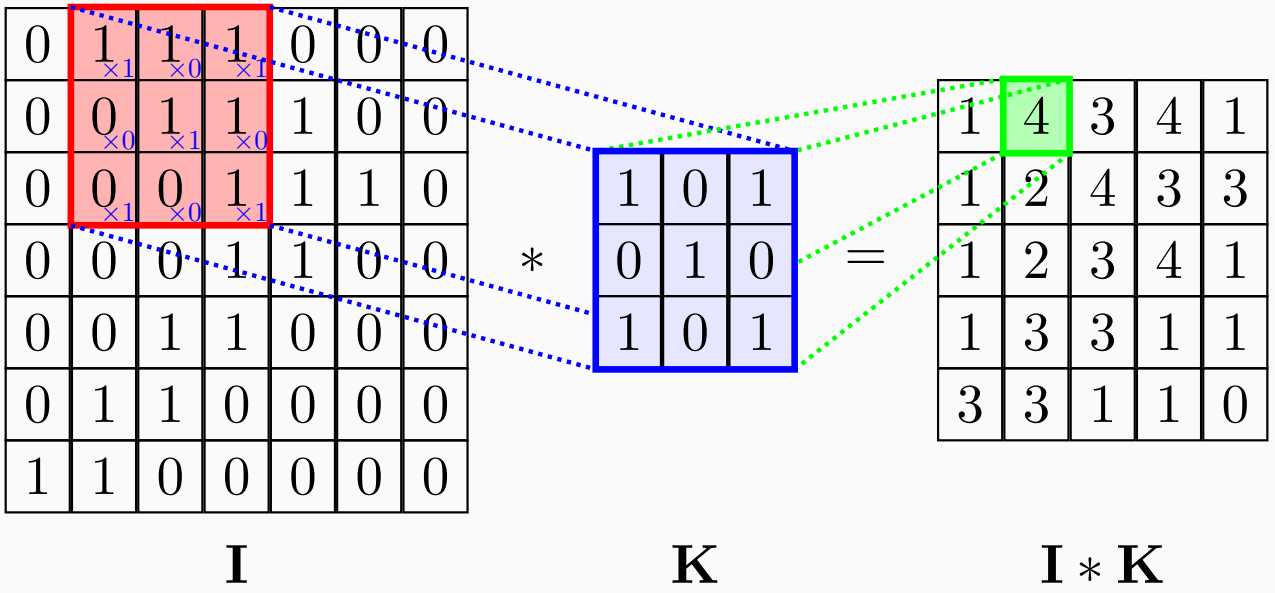


Imagen tomada de Wikipedia (Convolution)

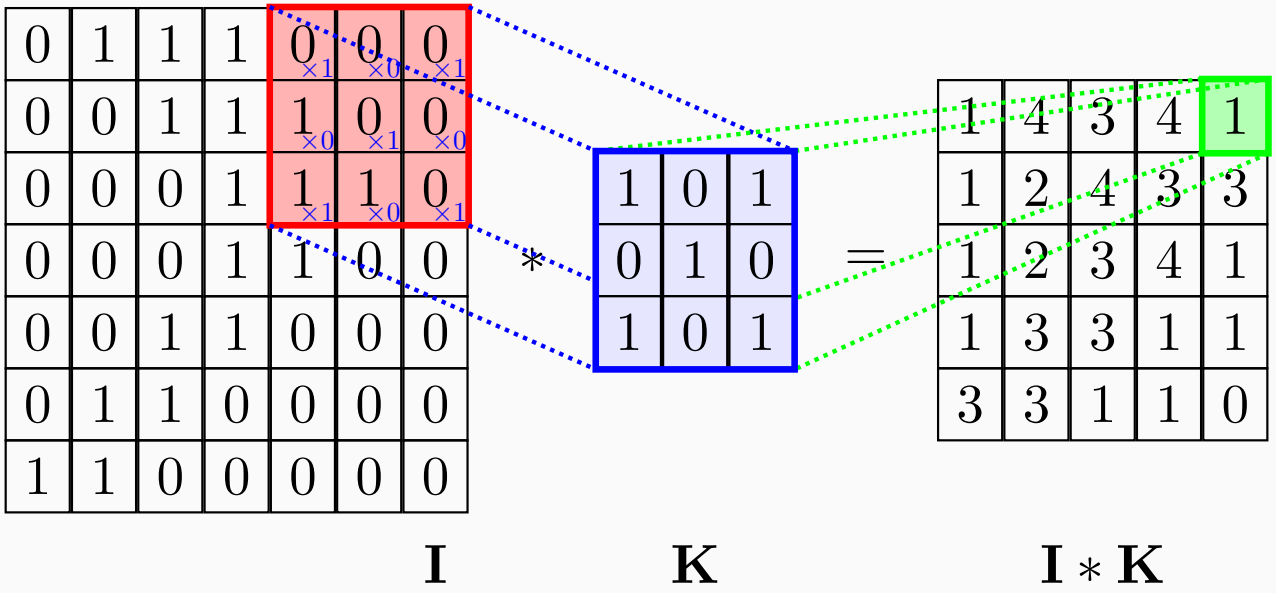
Convolución de una imagen con un filtro (1)



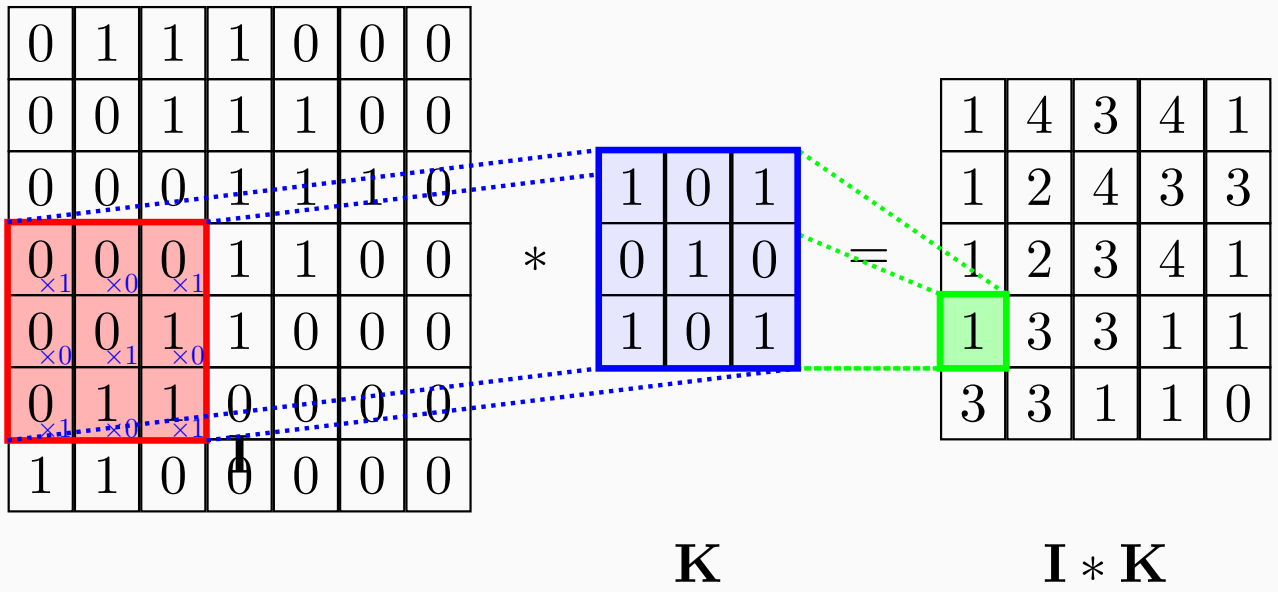
Convolución de una imagen con un filtro (2)



Convolución de una imagen con un filtro (3)



Convolución de una imagen con un filtro (4)



Capa convolucional

- Compuesta de filtros distintos
- La salida (mapas de activaciones) es un volumen

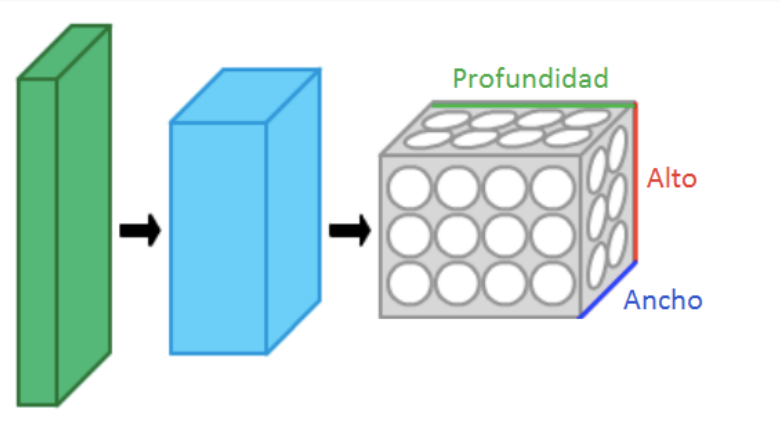
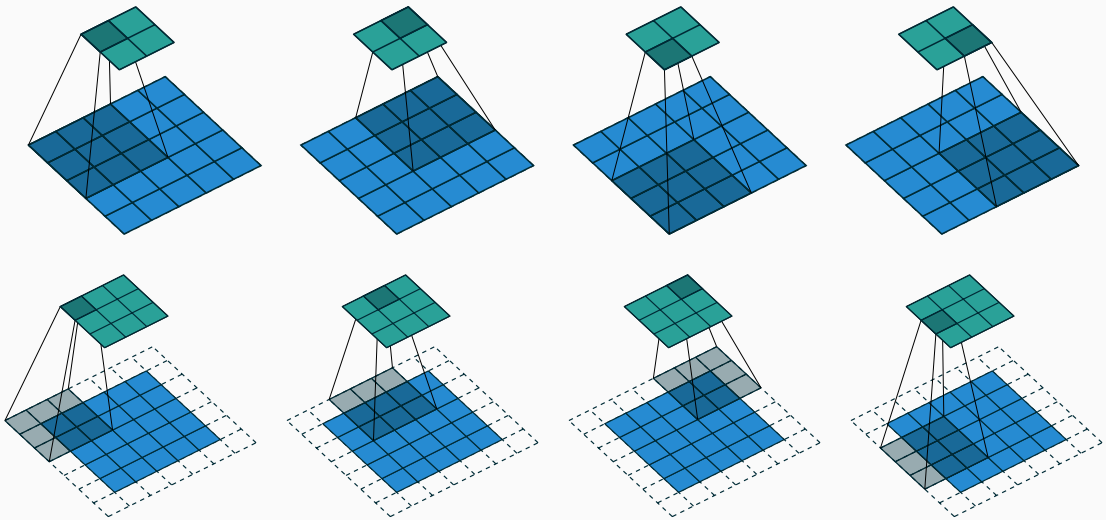


Imagen tomada de Wikipedia (Convolutional Neural Network)

Capa convolucional: hiperparámetros

- Número de filtros $N_{filtros}$, tamaño de cada filtro $F_H \times F_W$, desplazamiento S (*stride*) y relleno (*padding*)



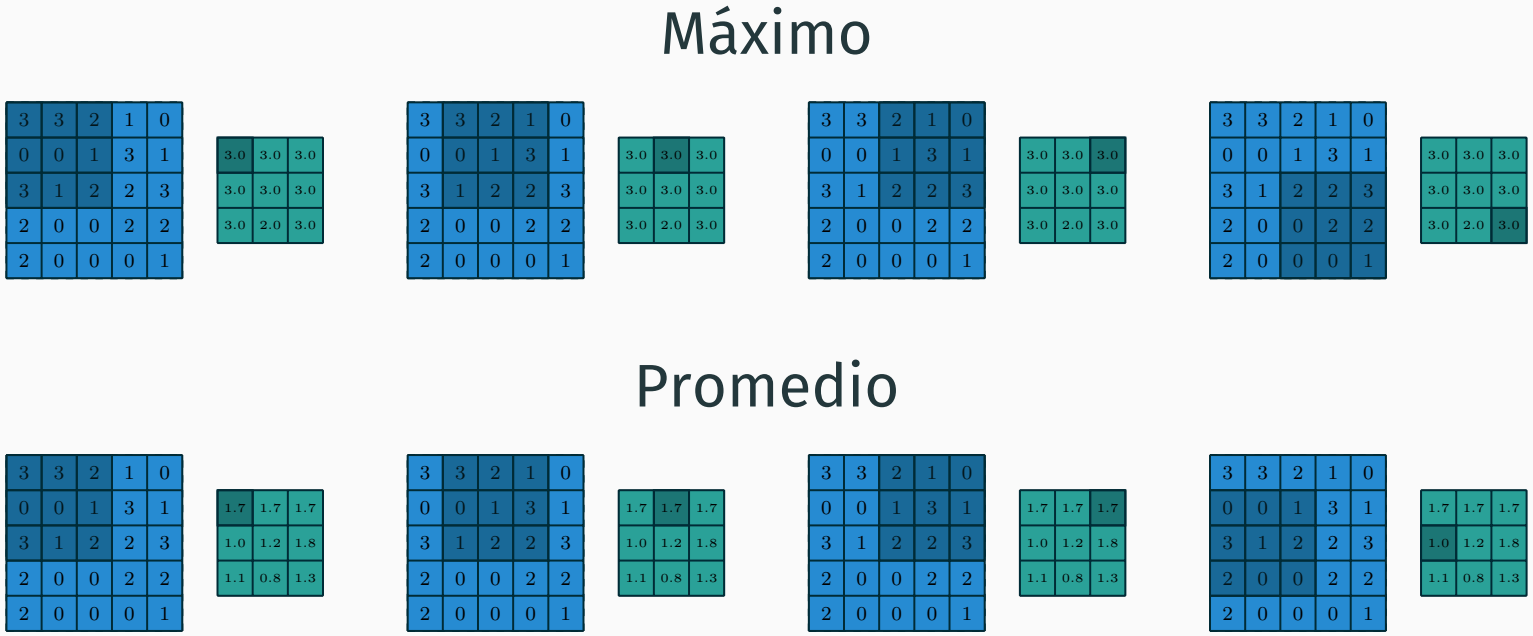
Imágenes creadas por Dumoulin and Visin. A guide to convolution arithmetic for deep learning, 2018.

$$H^{\{\ell\}} = \frac{H^{\{\ell-1\}} + 2P^{\{\ell-1\}} - F_H^{\{\ell\}}}{S^{\{\ell\}}} + 1$$

$$W^{\{\ell\}} = \frac{W^{\{\ell-1\}} + 2P^{\{\ell-1\}} - F_W^{\{\ell\}}}{S^{\{\ell\}}} + 1$$

Capa de submuestreo o decimación

- Capa para reducir dimensiones tomando el valor máximo (*max-pooling*) o el promedio (*average pooling*) de un grupo de activaciones usualmente contiguas



Imágenes creadas por Dumoulin and Visin. A guide to convolution arithmetic for deep learning, 2018.

Capa de submuestreo: hiperparámetros

- Hiperparámetros: alto y ancho de la ventana que define el grupo (F_H, F_W), desplazamiento S, relleno (*padding*) P

- El alto $H^{\{\ell\}}$ y ancho $W^{\{\ell\}}$ de la salida de una capa de submuestreo está dado por

$$H^{\{\ell\}} = \frac{H^{\{\ell-1\}} + 2P - F_H}{S} + 1$$
$$W^{\{\ell\}} = \frac{W^{\{\ell-1\}} + 2P - F_W}{S} + 1$$

Submuestro global promedio

- Reduce las dimensiones de un volumen de características, tomando únicamente el promedio de cada mapa

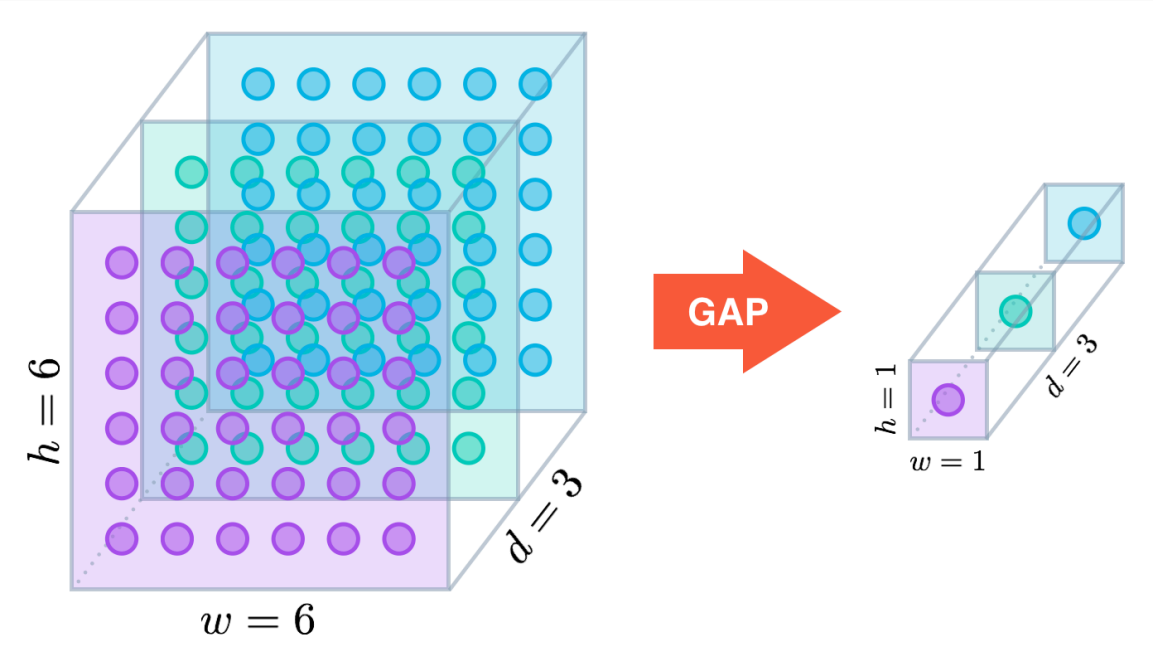


Imagen tomada de `https://alexisbcook.github.io/2017/global-average-pooling-layers-for-object-localization/`

Retro-propagación en capas convolucionales y de submuestreo

- Hacia adelante
  - Se aplican las operaciones de convolución con correspondiente activación y decimación
- Hacia atrás
  - En la convolución cada neurona actualiza los gradientes por separado y al final se suman para actualizar los pesos compartidos
  - En la decimación se actualiza sólo la neurona ganadora (*max-pooling*)

Convolución de 1×1

- Funciona como un tipo de decimación en profundidad

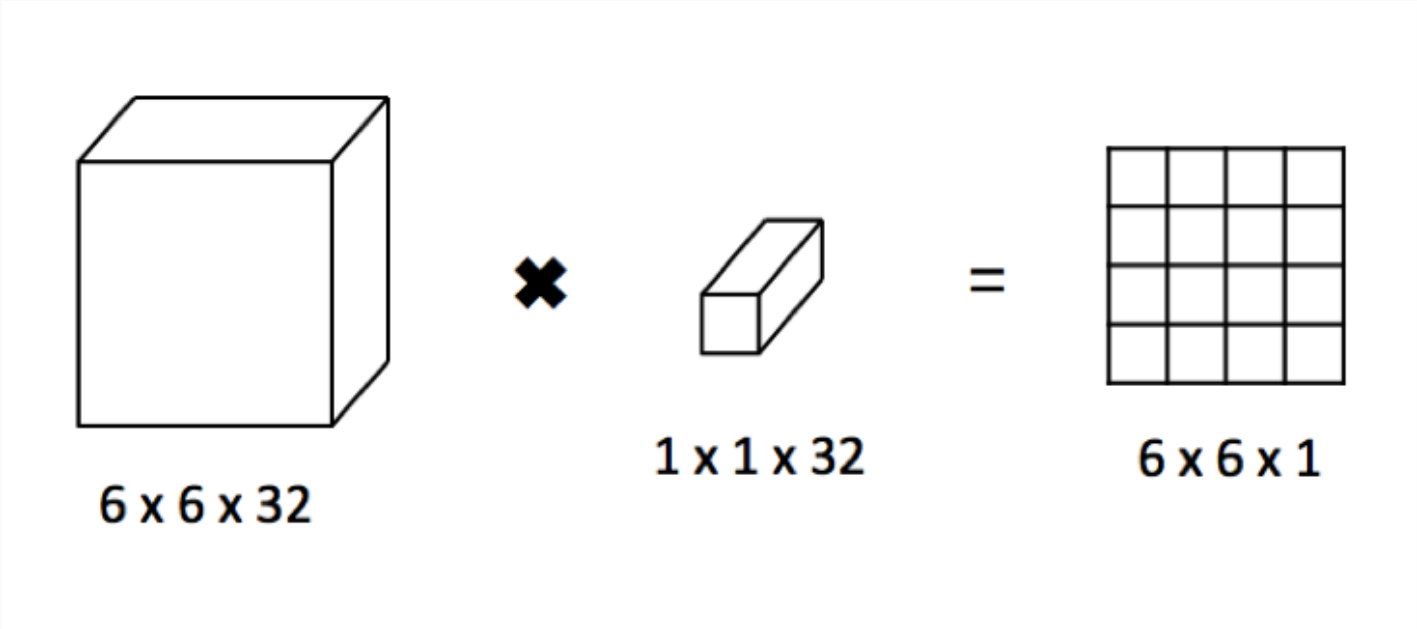


Imagen tomada de `https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524`