Existen Variantes de este modelo: GRU

"Gated Recurrent Unit". Tarea!

Notebook: Implementación.

Redes Convolucionales 30/11/2023

Ideas generales (Valerio Velardo)

- Se usan generalmente en tratamiento / Procesamiento de imágenes
- En problemas relacionados a imágenes performan mejor
- Utilizan una cantidad menor de parámetros comparadas con las redes totalmente conectadas.

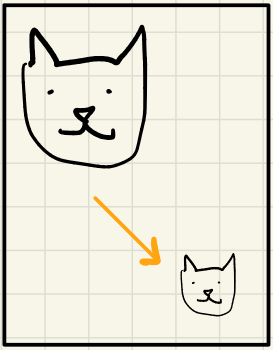
Recordando los conceptos relacionados a data estructurada / No-estructurada; pues habíamos definido a las imágenes dentro del grupo "no estructurado".

Sin embargo, en cierta medida, las imágenes tienen estructuras definidas:

i) Bordes, formas

ii) invarianza a la traslación, es decir un objeto identificable dentro de una imagen puede encontrarse en otro lugar de la misma y sigue siendo el mismo objeto.

iii) Invarianza a la escala, análogamente a las ideas anteriores, las formas persisten ante diferentes escalas y el objeto no pierde su naturaleza.



- La red convolucional intenta imitar el sistema de visión humana.
- David Hubel y Torsten Wiesel 1959 describieron el funcionamiento de células "simples" y "complejas" del cortex visual.
- Las células simples respondían a orillas o esquinas y barras con orientaciones particulares
- Las células complejas también responden a los mismos estímulos de las células simples pero además capturan barras o esquinas que pueden estar trasladadas en diferentes partes de la imagen (invarianza espacial)
- El año 1980 el profesor Dr. Fumihiko Fukushima propone un modelo inspirado en los trabajos de Hubel y Torsten. (Neocognitron)

- En el año 1998, Yann Le Cunn León Bottou, Yoshua Bengio y Patricia Haffner publicaron:

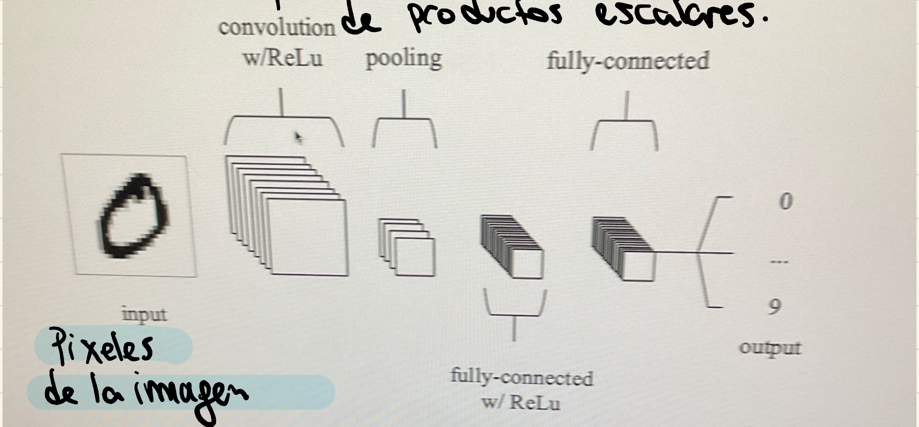
"Gradient-Based Learning Applied to document Recognition".

Paper citado más de 60K veces.

Yann LeCun VP & Chief en Meta

Arquitectura del Modelo

Convolución: Determina la salida de las neuronas conectadas a regiones locales de la entrada, a través de productos escalares.



Pixeles de la imagen

Fig. 2: An simple CNN architecture, comprised of just five layers

ReLU : función de activación componente a componente.

Pooling : Capa de "Compresión" de la información a lo largo de la dimensión espacial de la imagen.

Neuronas "Fully - Connected" : Transformación de la información para ser posteriormente interpretada como clases o targets (misma interpretación de antes).

Ejemplos y dimensiones (Notebooks).

- Utilizaremos en nuestros ejemplos la transformación Constant Q
- Es una transformación relacionada a la transformada de Fourier.
- A través de la transformada de Fourier podemos observar las frecuencias dominantes

en una porción de audio pequeña.

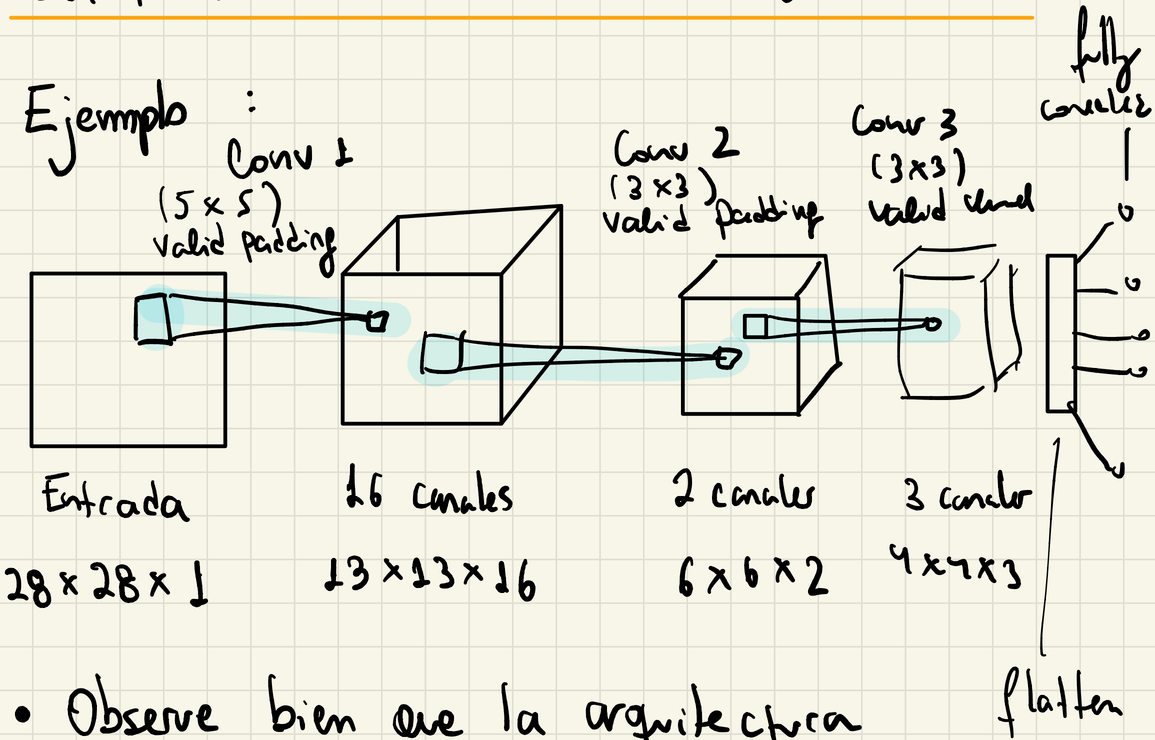
- Constant Q usa ventanas de tiempo dependientes de la frecuencia que queremos extraer, ventanas más grandes para las frecuencias graves y ventanas más pequeñas para las frecuencias altas. [Brown 1991]
- A modo de comparación con Fourier:

TABLE I. Comparison of variables in calculation of discrete Fourier transform (DFT) and of constant Q transform.

	Constant Q	DFT
Frequency	$(2^{1/24})^k \cdot f\text{min}$ exponential in k	$k\,\Delta f$ linear in k
Window	variable $= \mathrm{N}[\mathrm{k}] = \dfrac{SR\cdot Q}{f_k}$	constant $= N$
Resolution Δf	variable $= f\mathrm{k}/Q$	constant $= \mathrm{SR}/\mathrm{N}$
$\dfrac{f_k}{\Delta f_k}$	constant $= Q$	variable $= k$
Cycles in Window	constant $= Q$	variable $= \mathrm{k}$

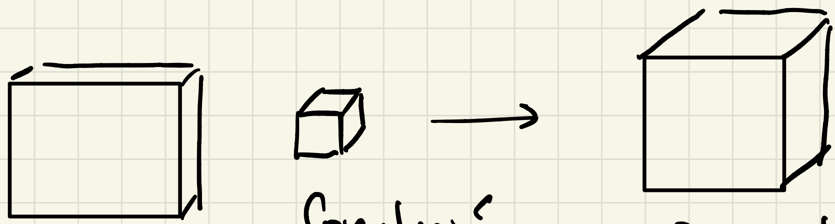
Calculando dimensiones en redes Convolucionales

Ejemplo :



- Observe bien que la arquitectura de la red esta definida por las entradas y salidas generadas por cada convolución.
- Destacado en azul vemos las capas convolucionales y el resto son las Entradas/Salidas de cada capa.
- Multiples filtros/kernels convolucionales son aplicados a la entrada. Cada kernel genera un mapa de caracteristica

- Una capa convolucional es la concatenación de filtros/kernels convolucionales.
- Por lo tanto es más intuitivo pensar la capa convolucional como un bloque



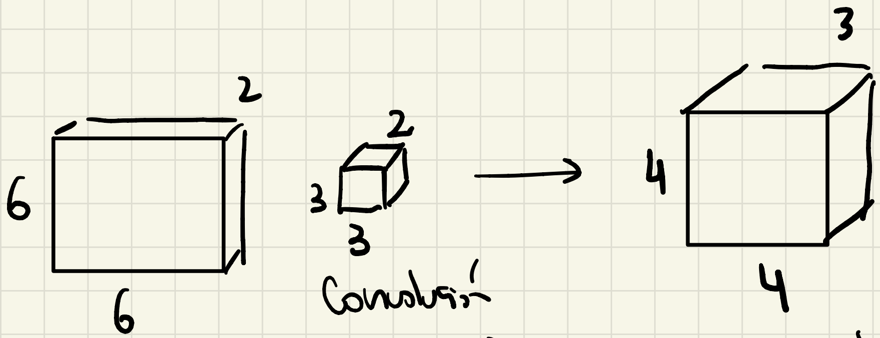
2 canales
$(6 \times 6 \times 2)$

Convolución
(3×3) kernel
padding válido

3 canales
$(4 \times 4 \times 3)$

- Las dimensiones de entrada, salida y kernel son posibles de inferir.
- Si las dimensiones de 2 de los tres son conocidas + los parámetros de Padding y Stride.

Inferencia de las dimensiones



Convolución
$(3 \times 3 \times 2)$

2 canales
$(4 \times 4 \times 3)$

2 canales
$(6 \times 6 \times 2)$

Entrada		Kernel	Salida
$W_i = 6$	(Width : Ancho)	$W_k = 3$	$W_o = 4$
$h_i = 6$	(Height : Alto)	$h_k = 3$	$h_o = 4$
$d_i = 2$	(depth : Profundidad)	$d_i = 2$	$d_o = 3$

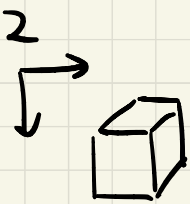
También se usa depth ≡ channels ≡ feature maps

- **Zero Padding**: Técnica que adiciona ceros extras en los bordes del arreglo, mientras se aplica el kernel/filtro.

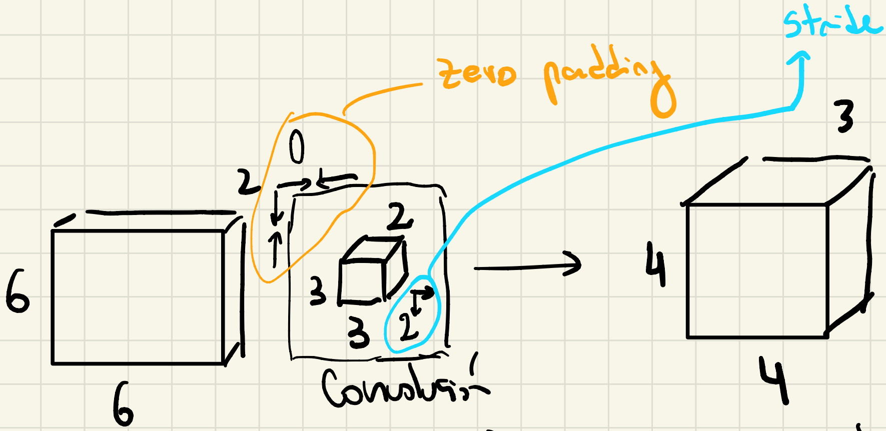
Valid padding = 0

- **Stride**: Paso de tamaño s que dará el kernel para aplicar la convolución.

A veces se escribe en notación:



"Los pasos dados en la dirección "derecha" y "abajo" son iguales a $S = 2$"



Convolución

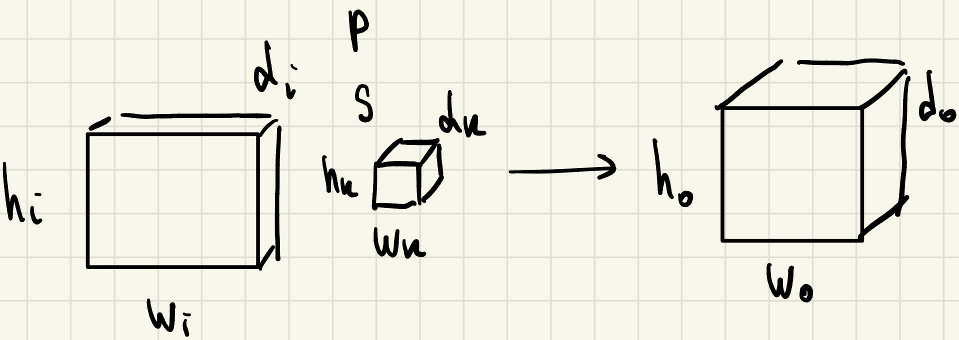
$(3 \times 3 \times 2)$

2 canales

$(4 \times 4 \times 3)$

2 canales

$(6 \times 6 \times 2)$



d_i Canales

$h_i \times w_i \times d_i$

$h_k \times w_k \times d_k$

valid padding p

d_o canales

$h_o \times w_o \times d_o$

Calculemos (Ejemplo Diego)

Fórmula :

$$w_o = \left\lfloor \frac{w_i - w_k + 2p_w}{s_w} \right\rfloor + 1 \qquad (1)$$

$$h_o = \left\lfloor \frac{h_i - h_k + 2p_h}{s_h} \right\rfloor + 1 \qquad (2)$$

$\lfloor \ \rfloor$: floor function

- Si el tamaño de los pasos verticales y horizontales son iguales (Kernel cuadrático) Entonces la dimensión de ancho y alto son iguales, se reduce a :

$$o = \left\lfloor \frac{\overset{\text{Canales (2)}}{i} - \overset{\text{filtros (3)}}{k} + 2\overset{\text{Padding (0)}}{p}}{\underset{\text{paso (2)}}{s}} \right\rfloor + 1 \qquad (3)$$

En nuestro ejemplo:

$$o = \left\lfloor \frac{6 - 3 + 2 \cdot 0}{2} \right\rfloor + 1 = \left\lfloor \frac{3}{2} \right\rfloor + 1$$

$$= \lfloor 1.5 \rfloor + 1$$

$$= 1 + 1$$

$$= 2$$

- Es decir, de la operación convolución nuestra salida será 2×2 para cada canal.

- La profundidad del arreglo de entrada es igual a la profundidad del arreglo de kernels.

- Como esas dimensiones deben coincidir, si quisieramos agregar más kernels estámos obligados a usar una 4 dimensión.

- La cuarta dimensión es llamada de "número de instancias".

Es decir tendremos:

Forma = instancias × ancho × alto × profundidad

Ejemplo

$(1 \times 6 \times 6 \times 2) \rightarrow (3 \times 3 \times 3 \times 2)$

$m_i \times h_i \times w_i \times d_i \qquad m_k \times h_k \times w_k \times d_k$

$\downarrow$

$(1 \times 2 \times 2 \times 3)$

$m_o \times h_o \times w_o \times d_o$

(Imágenes en la diapo)

3 instancias de 2 filtros 3×3 $\quad (3 \times 3 \times 3 \times 2)$

$\downarrow$

instancias