Sistemas de Inteligencia Artificial
Redes Neuronales Convolucionales

Centro de Inteligencia Computacional

2022

Redes Neuronales Convolucionales

Redes Neuronales Convolucionales

- **CNN**, Convolutional Neural Networks, o, **ConvNet**.
- Basadas alrededor de la operación de convolución.
- Aplicadas a procesamiento de imágenes.
- Tremendamente exitosas y uno de los motores detrás del auge de DL.

Las CNN fueron propuestas x Fukushima y LeCun en 1989, e iniciaron la tercer hola de las redes neuronales con el paper de Krizhevsky 2012

Redes Neuronales Convolucionales

Convolución Matemática

- Es una operación lineal
- Suponiendo que $x(t)$ es una señal unidimensional en función del tiempo y $w(t)$ es un **núcleo** de convolución,
- $s(t) = \int x(k)w(t-k)dk$
- $s(t) = (x * w)(t) =< x(k), w^*(t-k) >$, donde $*$ es el conjugado.
- (Correlación $s(t) = \int x(k)w(t+k)dk$ (+ positivo))

Convolución Discreta

- $s[n] = \sum_k x[k]w[n-k]$

Redes Neuronales Convolucionales

Convolución

- La convolución matemática cumple un rol primordial en el procesamiento de señales digitales. Por ejemplo, cualquier filtro digital lineal puede representarse por una igualdad basada en convoluciones discretas.

CCDE - Constant Coefficient Difference Equation

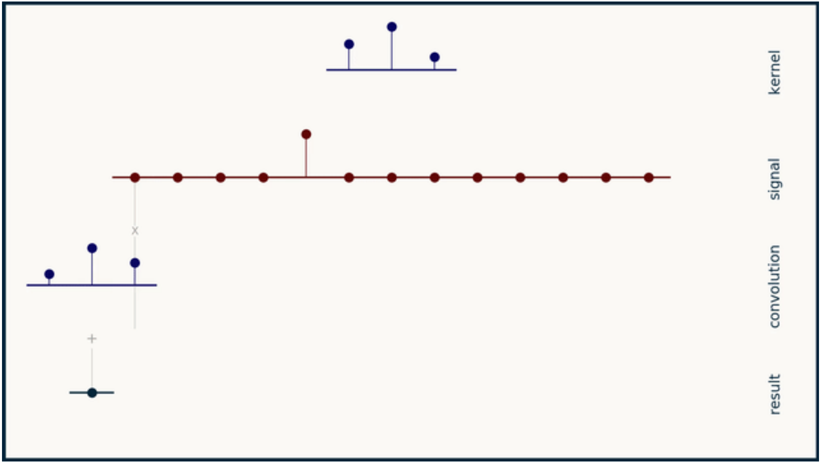
- $\sum_{k=0}^{N-1} w_1[k]y[n-k] = \sum_{k=0}^{M-1} w_2[k]x[n-k]$
- $Y(z) = H(z)X(z)$
- w_1, w_2 son kernels de convolución, x, y son señales discretas.
- X, Y son los estados del sistema y H es la función de transferencia.

Redes Neuronales Convolucionales

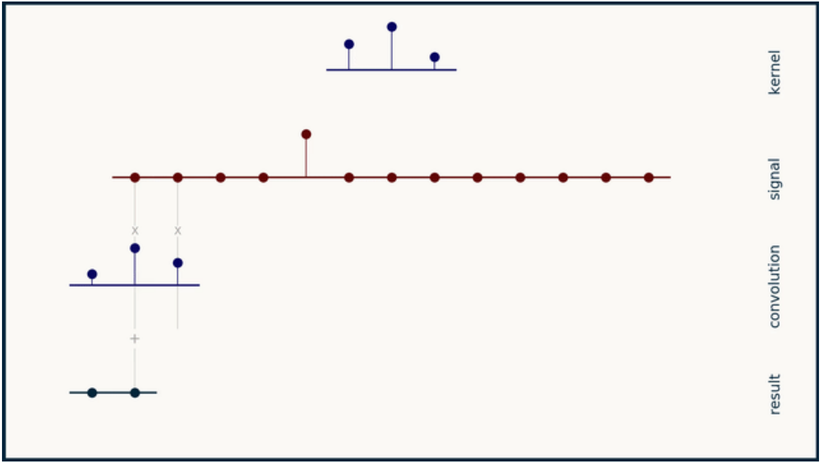
Convolución

- $x = [1]$
- $y = [1, 3, 2]$

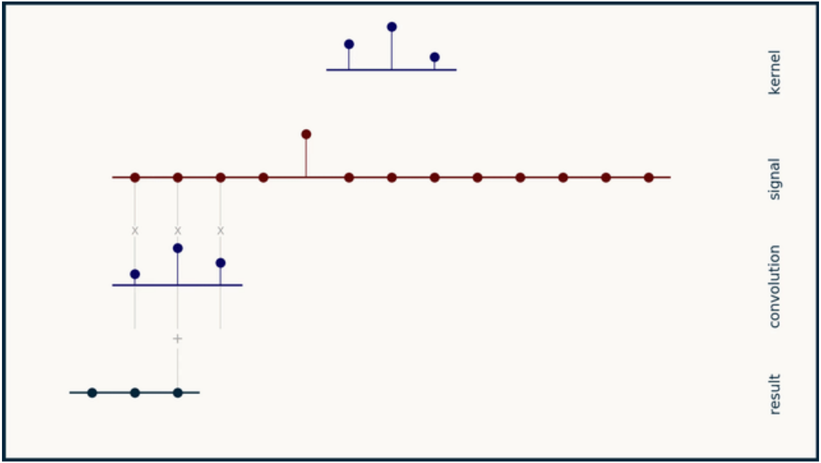
Convolution



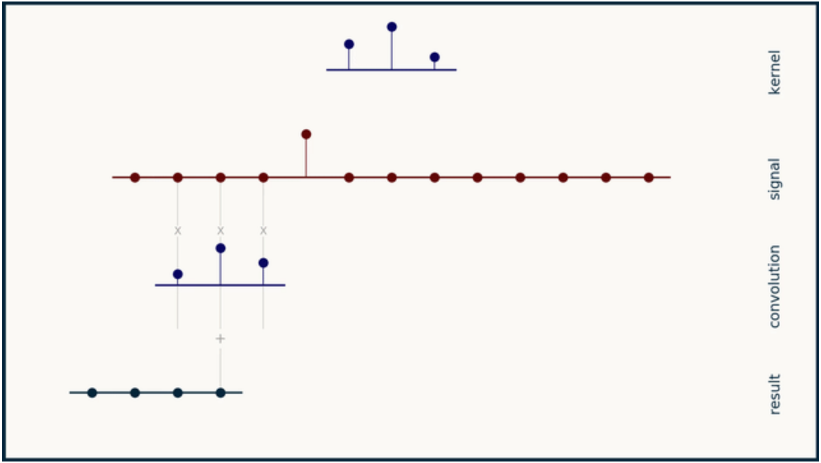
Convolution



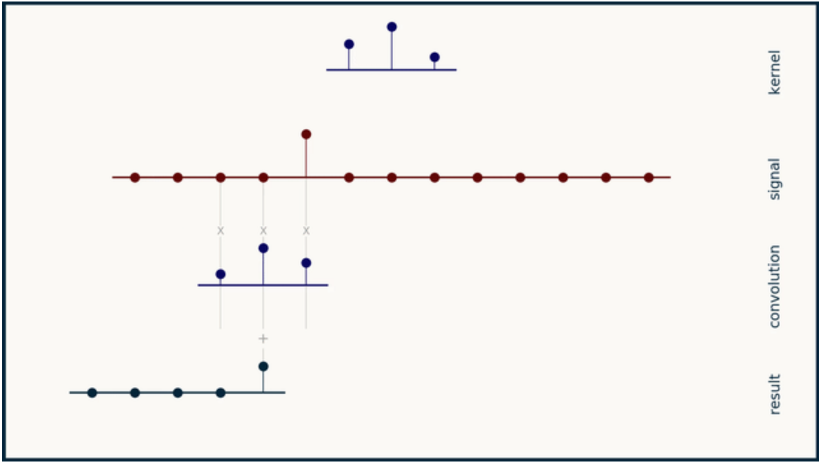
Convolution



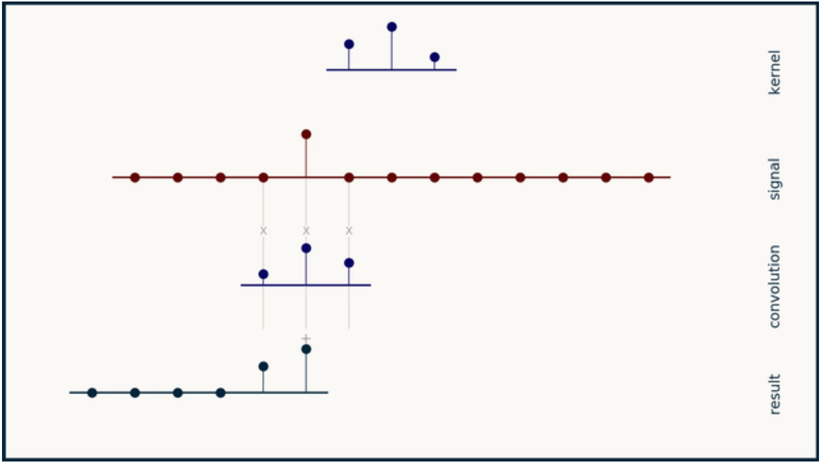
Convolution



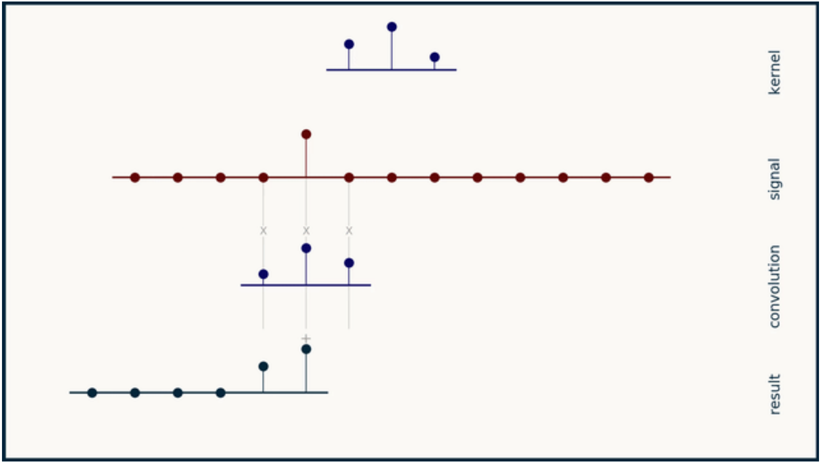
Convolution



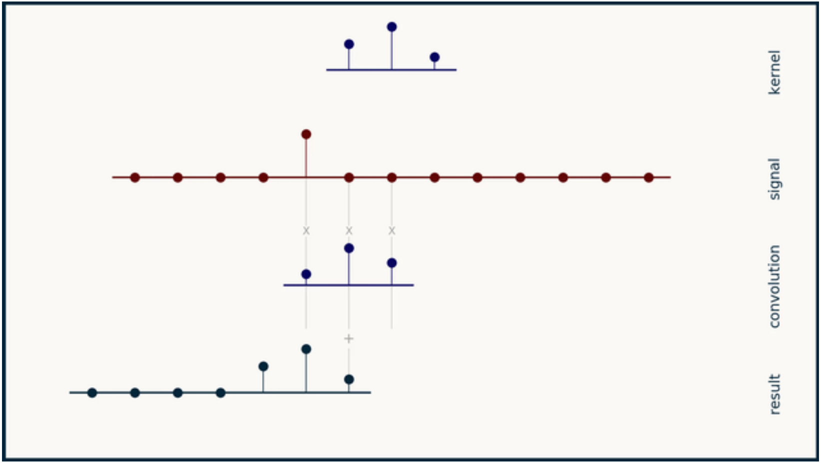
Convolution



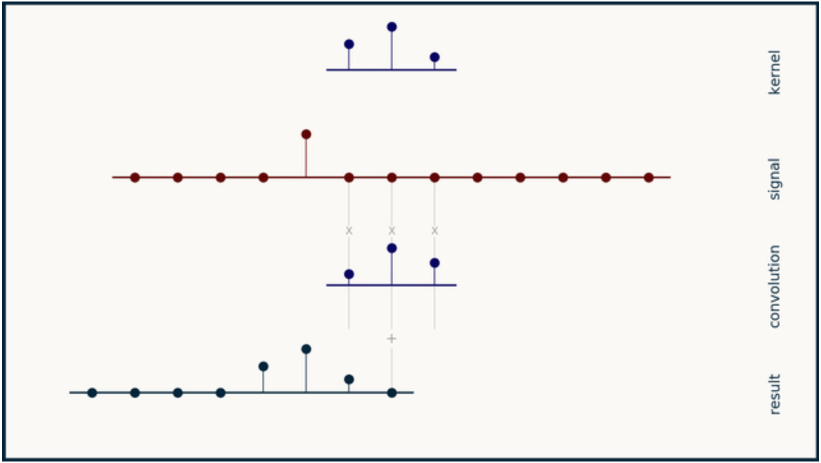
Convolution



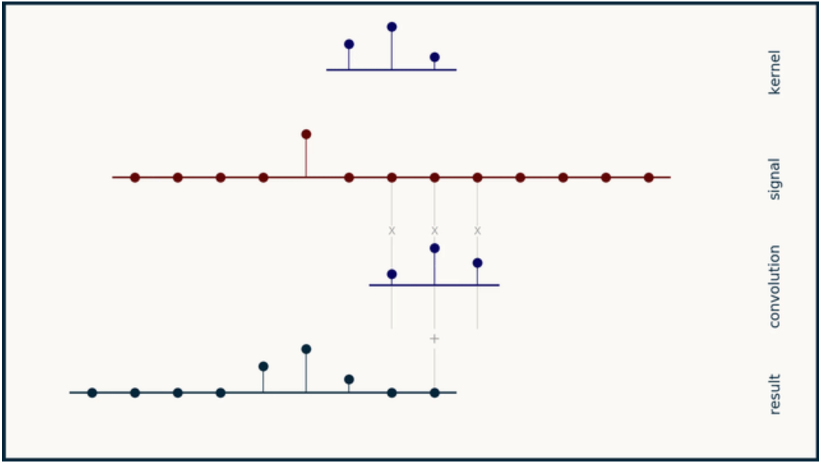
Convolution



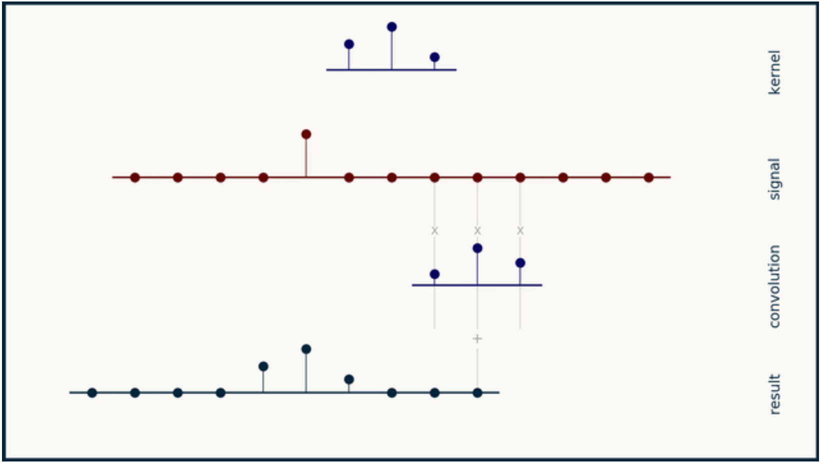
Convolution



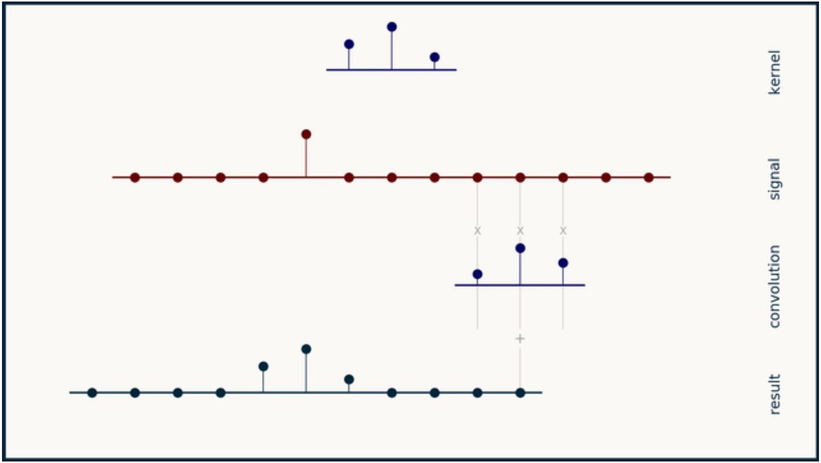
Convolution



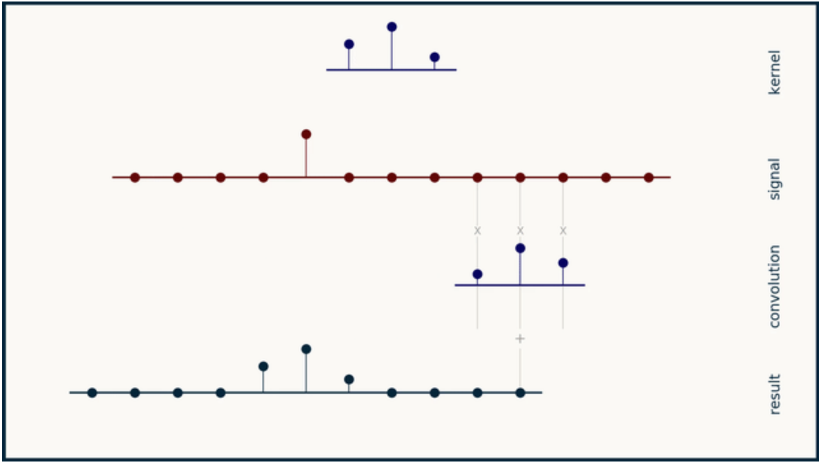
Convolution



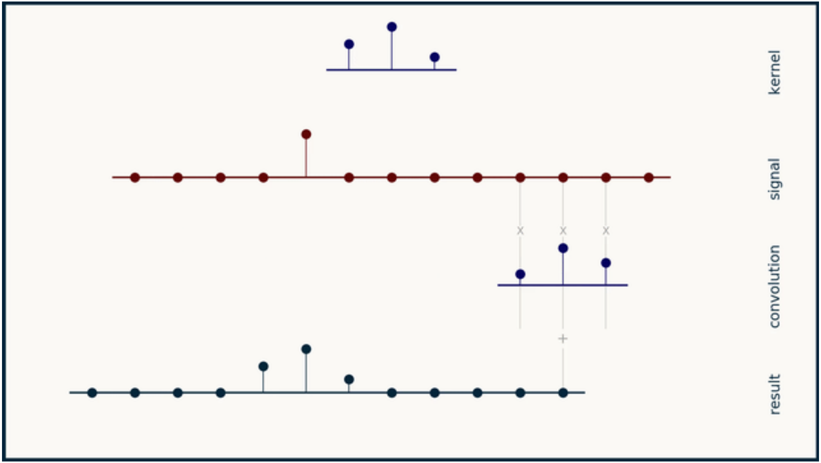
Convolution



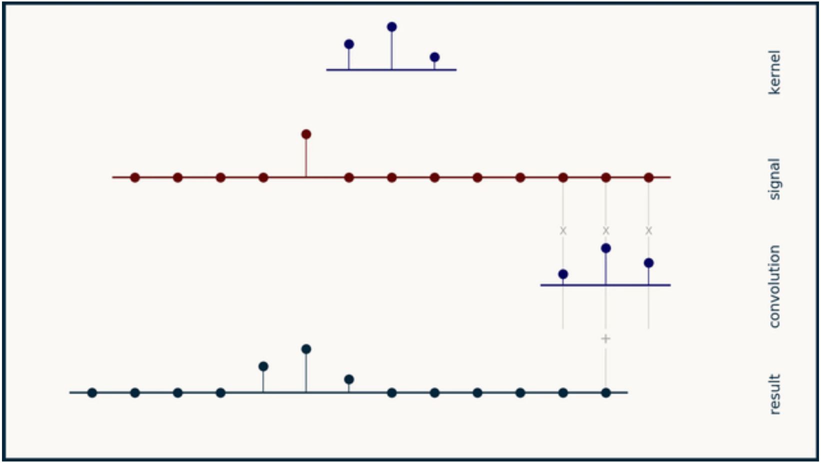
Convolution



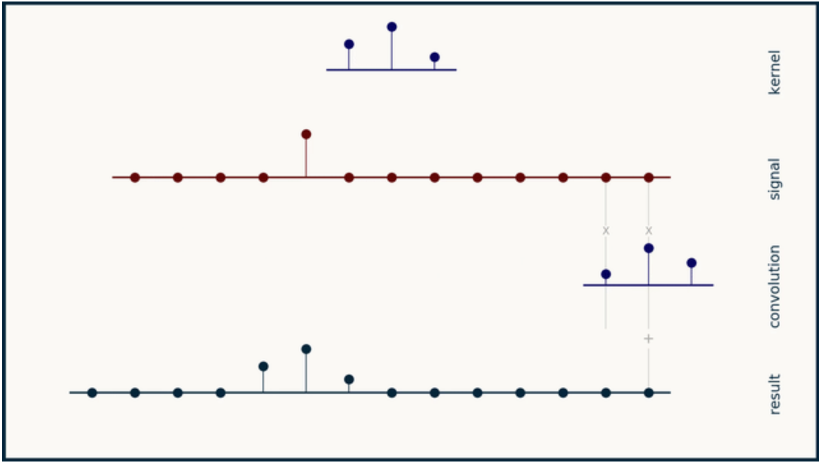
Convolution



Convolution



Convolution



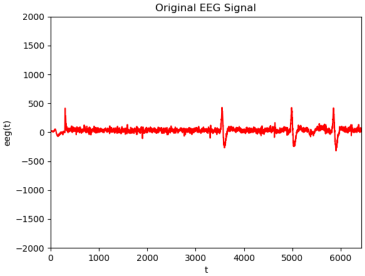
Redes Neuronales Convolucionales

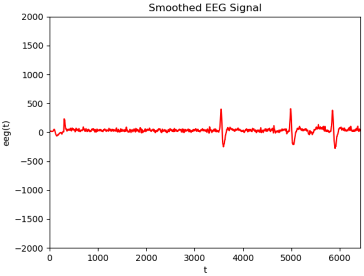
Moving Average

```
windowlength = 10
avgeeg = np.convolve(signal,
     np.ones((windowlength,))/windowlength,
           mode='same')
```

Redes Neuronales Convolucionales

Aplicando un moving average de tamaño 10.





Redes Neuronales Convolucionales

Padding

- valid
- same
- full

```
windowlength = 10
avgeeg = np.convolve(signal,
     np.ones((windowlength,))/windowlength,
           mode='same')
```

Convolución en Señales

Padding

Señal discreta [1,2,-5,4,2,-1], con un kernel [-1,2,-1]:

Valid	Same	Full
[8, -16, 11, 1]	[0, 8, -16, 11, 1, -4]	[-1,0, 8, -16, 11, 1, -4,1]

Convolución en Señales

Padding

Señal discreta [1,2,-5,4,2,-1], con un kernel [-1,2,-1]:

Valid	Same	Full
[8, -16, 11, 1]	[0, 8, -16, 11, 1, -4]	[-1,0, 8, -16, 11, 1, -4,1]

Convolución en Señales

Padding

Señal discreta [1,2,-5,4,2,-1], con un kernel [-1,2,-1]:

Valid	Same	Full
[8, -16, 11, 1]	[0, 8, -16, 11, 1, -4]	[-1,0, 8, -16, 11, 1, -4,1]

Convolución en Señales

Padding

Señal discreta [1,2,-5,4,2,-1], con un kernel [-1,2,-1]:

Valid	Same	Full
[8, -16, 11, 1]	[0, 8, -16, 11, 1, -4]	[-1,0, 8, -16, 11, 1, -4,1]

Redes Neuronales Convolucionales

Convolución bidimensional

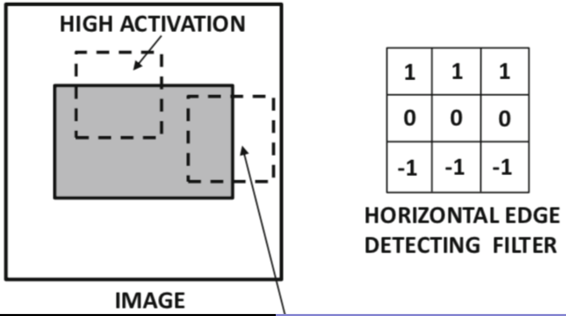
$$\begin{aligned} S(i,j) = (I * k)(i,j) &= \sum_m \sum_n I(m,n) K(i-m, j-n) \\ &= \sum_m \sum_n I(i-m, j-n) K(m,n) \end{aligned}$$

Filtrado por Núcleos de Convolución

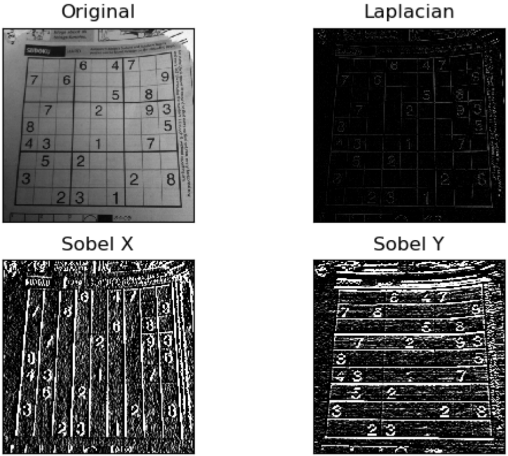
La operación de filtrado básica en Visión por Computadora es mediante la aplicación de una operación de convolución bidimensional con diferentes Kernels de convolución.

Redes Neuronales Convolucionales

- En Visión por Computadora o Análisis y Tratamiento de imágenes, es necesario aplicar 'filtros' a las imágenes para resaltar ciertas características.
- Por ejemplo, para detectar bordes horizontales o verticales.



Redes Neuronales Convolucionales



Redes Neuronales Convolucionales

Filtros

Los kernels actúan como filtros, como por ejemplo un filtro gaussiano (una curva de Gauss bidimensional)
`https://twitter.com/i/status/1303489896519139328`

O también pueden ser filtros para detectar bordes verticales u horizontales.

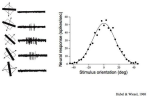
$$\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

Redes Neuronales Convolucionales

Biomimetismo

La idea es de 1989, y busca plantear una biomimesis del propio esquema de la corteza visual. En ese momento no pudo ser implementada de manera eficiente porque el hardware requerido no estaba disponible.

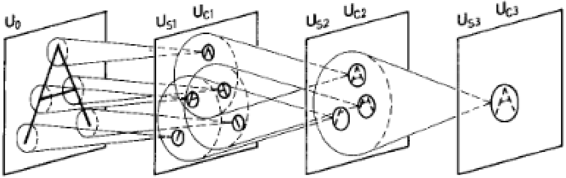
La visión de redes neuronales es que los filtros tradicionales que se usan en visión por computadora son versiones "shallow" de redes neuronales convolucionales, y en la profundidad reside justamente uno de los éxitos de esta alterantiva.



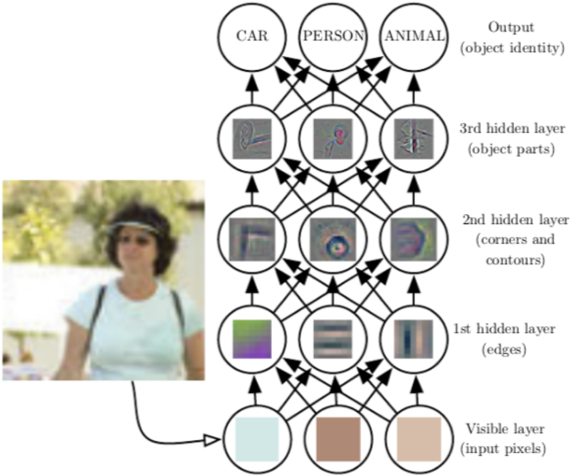
Redes Neuronales Convolucionales

Biomimetismo

Neocognitron [7].



Redes Neuronales Convolucionales

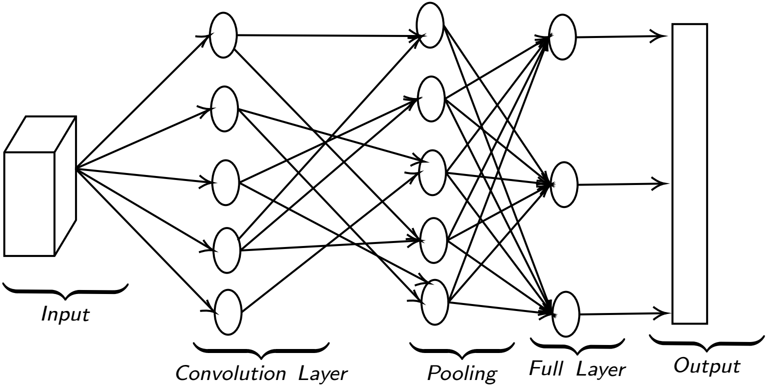


Redes Neuronales Convolucionales

La idea.

- Modificar una MLP de forma que realice una operación de convolución, de una capa a la otra.
- Imitar la misma idea de lo que se observa en la corteza visual, mediante la utilización de la convolución.

Redes Neuronales Convolucionales



Redes Neuronales Convolucionales

$$W^{p,q} = [w_{ijk}^{(p,q)}]$$

i : height

j : width, q : Layer

k : depth, p : Filter

$$H^q = [h_{ijk}^q]$$

H : *inputlayer*

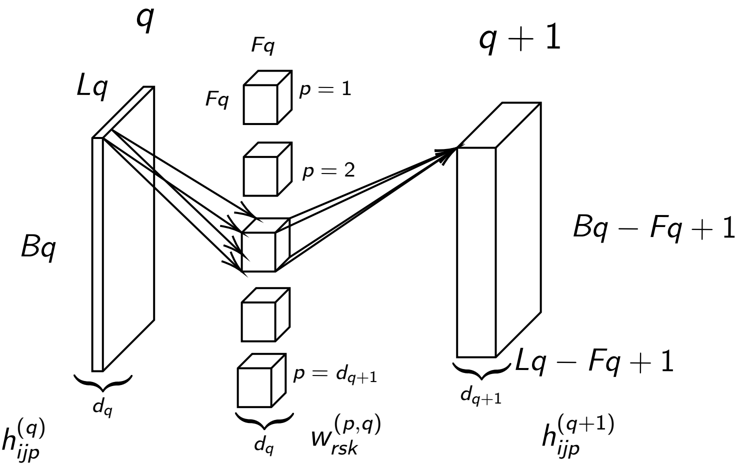
$$h_{ijp}^{(q+1)} = \sum_{r=1}^{Fq} \sum_{s=1}^{Fq} \sum_{k=1}^{dq} w_{rsk}^{(p,q)} h_{i+r-1,j+s-1,k}^q$$

$$\forall i \in \{1, ..., Lq - Fq + 1\}$$

$$\forall j \in \{1, ..., Bq - Fq + 1\}$$

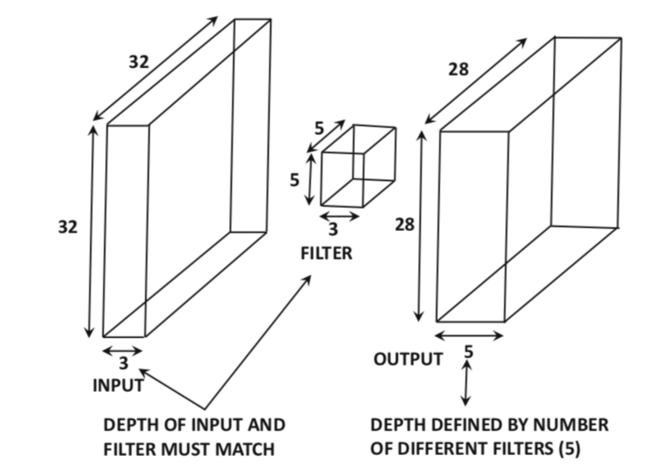
$$\forall p \in \{1, ..., d_{q+1}\}$$

Redes Neuronales Convolucionales



Redes Neuronales Convolucionales

Feature Maps



Redes Neuronales Convolucionales

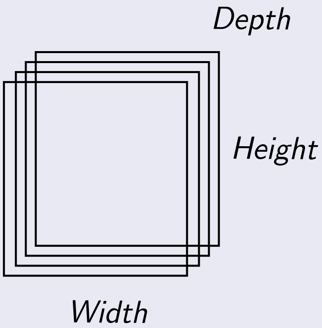
Capa ReLU

$$f(x) = x^{+} = max(0, x) \tag{1}$$

$$f(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases}$$

Redes Neuronales Convolucionales

Feature Maps



Características

- Stride
- Border Effects

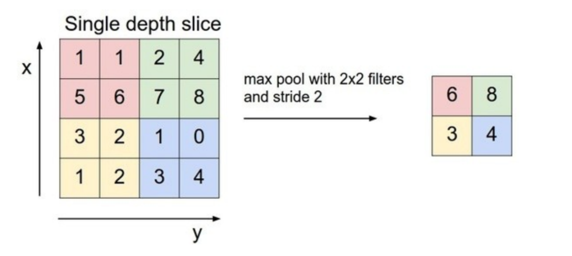
Redes Neuronales Convolucionales

Pooling

- Pooling es similar a un downsampling.
- MaxPooling es la operación más usada. Dada una región de 2x2, obtener el valor máximo.
- Aportan a la invarianza translacional: no importa dónde se de el máximo, se puede filtrar en la capa siguiente.
- Puede extenderse a hacer otras características invariantes, depende como se agrupen los filtros, y los poolings siguientes.

Redes Neuronales Convolucionales

Pooling



Redes Neuronales Convolucionales

Fully Connected Layer

- Al final de la Red hay una (o más capas) que se encargan de hacer la clasificación final.
- Softmax: $\sigma(\vec{V})_i = \frac{e^{V_i}}{\sum_{j=1}^{K} e^{V_i}}$

Redes Neuronales Convolucionales

Backpropagation

- Mantener la identificación de la influencia al momento de retropropagar los errores
- En las capas de convolución, como los pesos son compartidos, es necesario acumular todas las variaciones a la hora de actualizar los pesos. Es decir, lo mismo que ocurre con los en las épocas en el tiempo, ahora hay que hacerlo en el espacio.

Redes Neuronales Convolucionales

Backpropagation

1. Supongamos una red para identificar perros, botes y Gokus.
2. Red inicializada al azar, igual que siempre.
3. Calculamos los valores de salida, capa a capa.
4. En las capas *Fully Connected* es exactamente igual que el perceptrón multicapa.
5. En las capas de Pooling, solo se actualizan la contribución de los pesos de la salida ganadora (asumiendo una función identidad).
6. En las capas convolucionales, hay que mantener los pesos de los filtros en la relación espacial.

Redes Neuronales Convolucionales

Backpropagation - Transposición basdada en un tensor

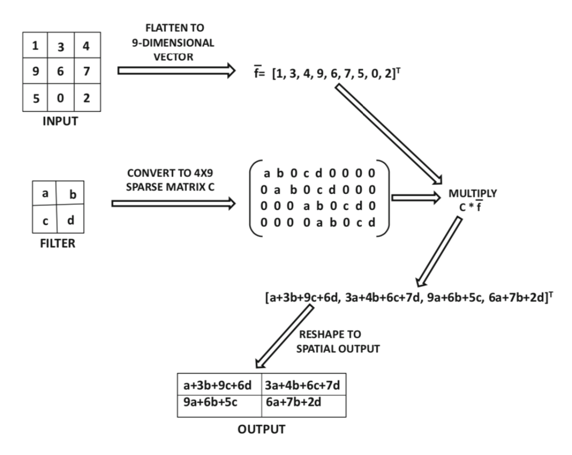
$$z_{q+1} = W^T z_q$$
$$g_q = W g_{q+1}$$

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \qquad \begin{pmatrix} i & h & g \\ f & e & d \\ c & b & a \end{pmatrix}$$

$$w_{ijk}^{(p,q)} = u_{rsp}^{(k,q+1)}$$
$$r = Fq - i + 1$$
$$s = Fq - j + 1$$

Redes Neuronales Convolucionales

Sparse Matrix Representation



Redes Neuronales Convolucionales

Arquitecturas y Redes

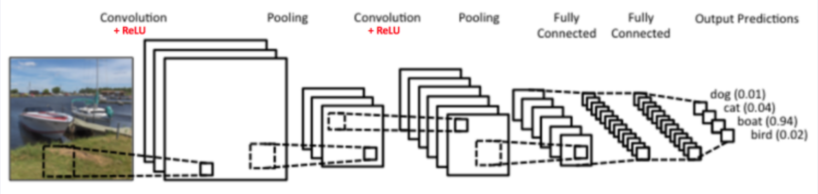
- Las diferentes combinaciones de capas de convolución, reLU, pooling y fully connected, permiten armar arquitecturas profundas y complejas que intentan solucionar diferentes problemas.
- Como siempre, la elección de la arquitectura correcta no escapa a las generalidades de las redes neuronales. Un poco arte, educated guesses y prueba y error.

Codificación

- (CR) 2P(CR) 2P(CR) 2PF

Redes Neuronales Convolucionales

Arquitecturas - LeNet 89



Redes Neuronales Convolucionales

¿ Por qué funcionan ?

- Matrices esparsas son mucho más eficientes: menos cómputos.
- *Parameter sharing: tied weights*
- *Equivariant* $g(f(x)) = f(g(x))$ Invariante a las traslaciones
- Jerarquías Espaciales: Ingeniería de Características Jerárquica

Redes Neuronales Convolucionales

Regularización

- Weight Dropout

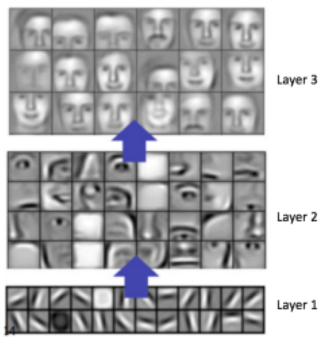
Redes Neuronales Convolucionales

Transfer Learning

- Pretrained FC7 Features

Redes Neuronales Convolucionales

Feature Learning



Redes Neuronales Convolucionales

Inteligible Property, Propiedad de Legibilidad

- *Saliency Maps* $\frac{\partial \psi}{\partial h}$.
- Shapley Value

Redes Neuronales Convolucionales

Bloques de construcción

- LexNet
- AlexNet
- UNet: segmentación de imágenes

CNN para diferentes dominios

Dimensions	Singlechannel	Multichannel
1-D	Audio Waveforms	Electroencephalography Motion Capture Markers
2-D	Spectrograms	Color Image
3-D	Computer Tomography	Video

Redes Neuronales Convolucionales

Playground

https://www.cs.ryerson.ca/~aharley/vis/conv/flat.html

Referencias

- Fukushima Neocognitron [3].
- LeCun 1989, LeNet Architecture [6].
- Libro de Aggarwal Neural Networks and Deep Learning [1].
- Guía para entender la aritmética de las convoluciones [2, 8]
- Estupendo resumen de CNN desde una perspectiva super variada [4]
- AlexNet 2012 [5].
- https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Referencias I

[1] Charu C Aggarwal et al. *Neural networks and deep learning*. Springer, 2018.

[2] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. mar 2016.

[3] Kunihiko Fukushima and Nobuaki Wake. Handwritten alphanumeric character recognition by the neocognitron. *IEEE transactions on Neural Networks*, 2(3):355–365, 1991.

[4] Shengli Jiang and Victor M. Zavala. Convolutional Neural Nets: Foundations, Computations, and New Applications. jan 2021.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Referencias II

[6] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[7] Grace W. Lindsay. Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of Cognitive Neuroscience*, pages 1–15, Feb 2020.

[8] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, nov 2017.