

Redes convolucionales

Jocelyn Dunstan Escudero

jdunstan@uc.cl

Departamento de Ciencia de la Computación
& Instituto de Matemática Computacional
Pontificia Universidad Católica de Chile



29 de octubre de 2025

Objetivos

- Distinguir los elementos básicos de una red convolucional.
- Entender el concepto de convolución.
- Calcular dimensiones de la salida de una capa convolucional.



Recomendación: Cursos de Domingo Mery

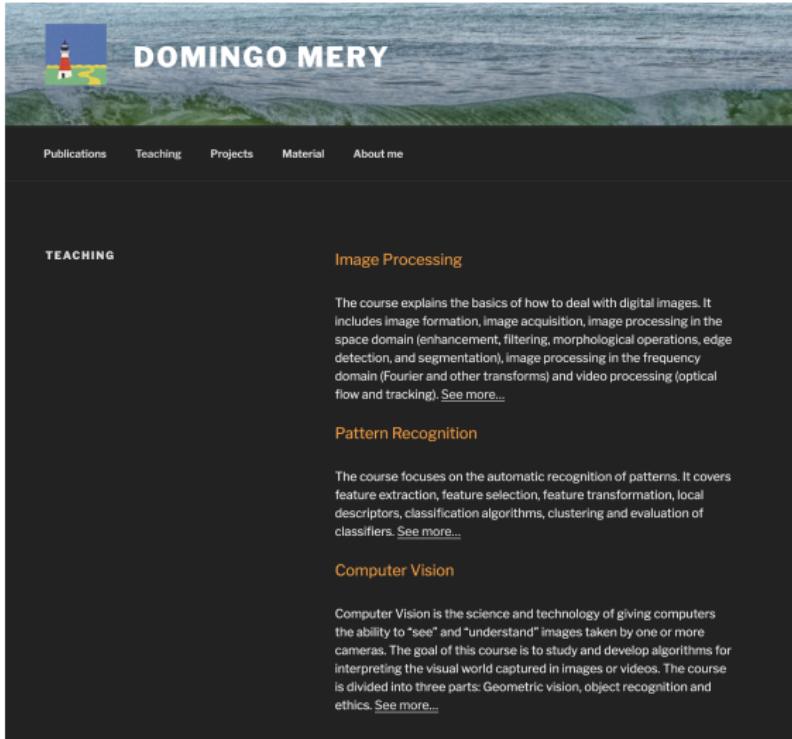
A screenshot of a website for "DOMINGO MERY". The header features a small logo with a lighthouse and the text "DOMINGO MERY" over a background image of ocean waves. Below the header is a navigation bar with links: Publications, Teaching, Projects, Material, and About me. The main content area is titled "TEACHING" and contains three sections: "Image Processing", "Pattern Recognition", and "Computer Vision".

Image Processing

The course explains the basics of how to deal with digital images. It includes image formation, image acquisition, image processing in the space domain (enhancement, filtering, morphological operations, edge detection, and segmentation), image processing in the frequency domain (Fourier and other transforms) and video processing (optical flow and tracking). [See more...](#)

Pattern Recognition

The course focuses on the automatic recognition of patterns. It covers feature extraction, feature selection, feature transformation, local descriptors, classification algorithms, clustering and evaluation of classifiers. [See more...](#)

Computer Vision

Computer Vision is the science and technology of giving computers the ability to "see" and "understand" images taken by one or more cameras. The goal of this course is to study and develop algorithms for interpreting the visual world captured in images or videos. The course is divided into three parts: Geometric vision, object recognition and ethics. [See more...](#)



<https://domingomery.ing.uc.cl/teaching/>

6.874, 6.802, 20.390, 20.490, HST.506
Deep Learning in the Life Sciences

Lecture 3: Convolutional Neural Networks

Prof. Manolis Kellis

Slides credit: **6.S191**, Dana Erlich, Param Vir Singh,
David Gifford, Alexander Amini, Ava Soleimany,
@TessFerrandez's totally awesome **Coursera** Notes,
and many more outstanding online resources



<http://mit6874.github.io>

<https://mit6874.github.io/assets/sp2021/slides/103.pdf>

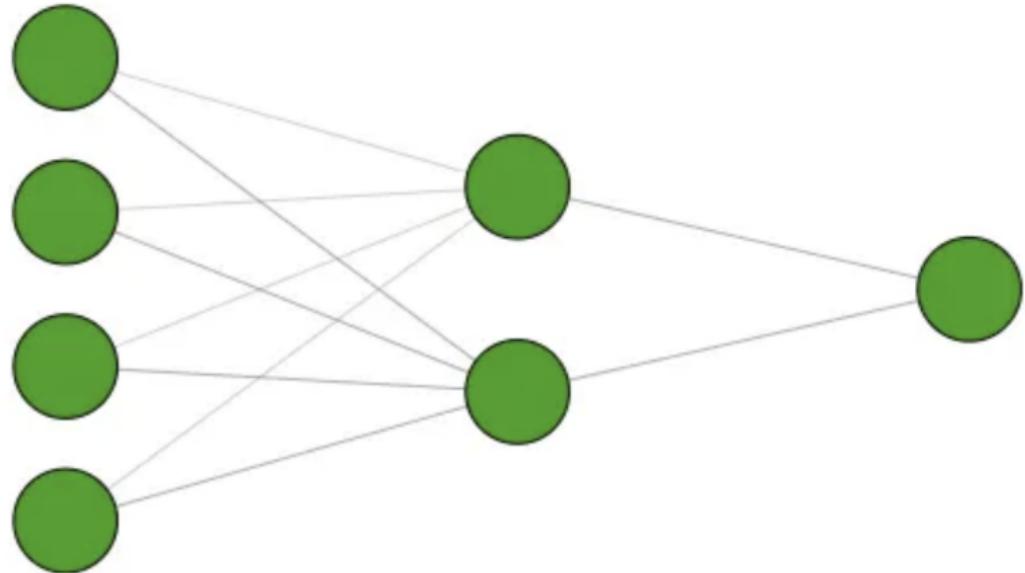
<https://www.youtube.com/watch?v=r5nXYc2wYvI&list=PLypixJdtIca5sxV7aE3-PS9fYX3vUdIOX&index=5>



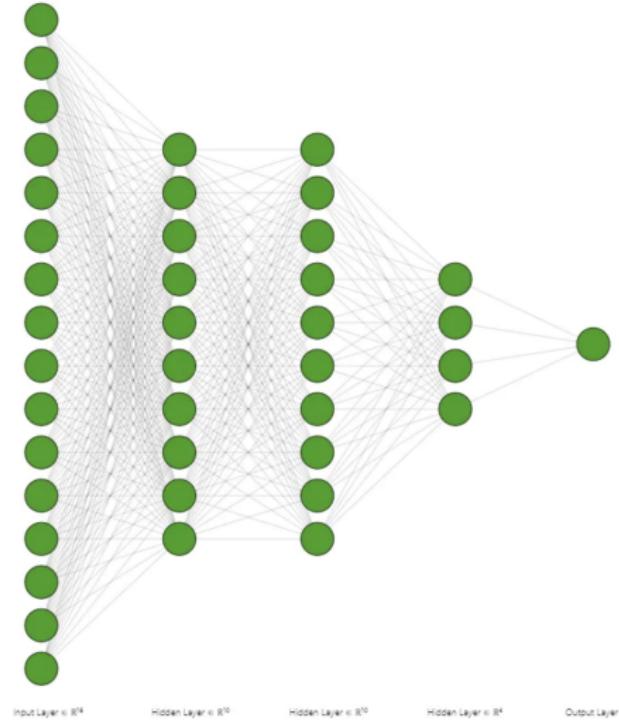
**Hasta ahora hemos visto
solo redes fully connected**



¿Cuántos parámetros tengo que aprender en esta red?



¿Y en esta?



Contando parámetros en una red *fully connected*



¿Qué vemos cuando vemos?



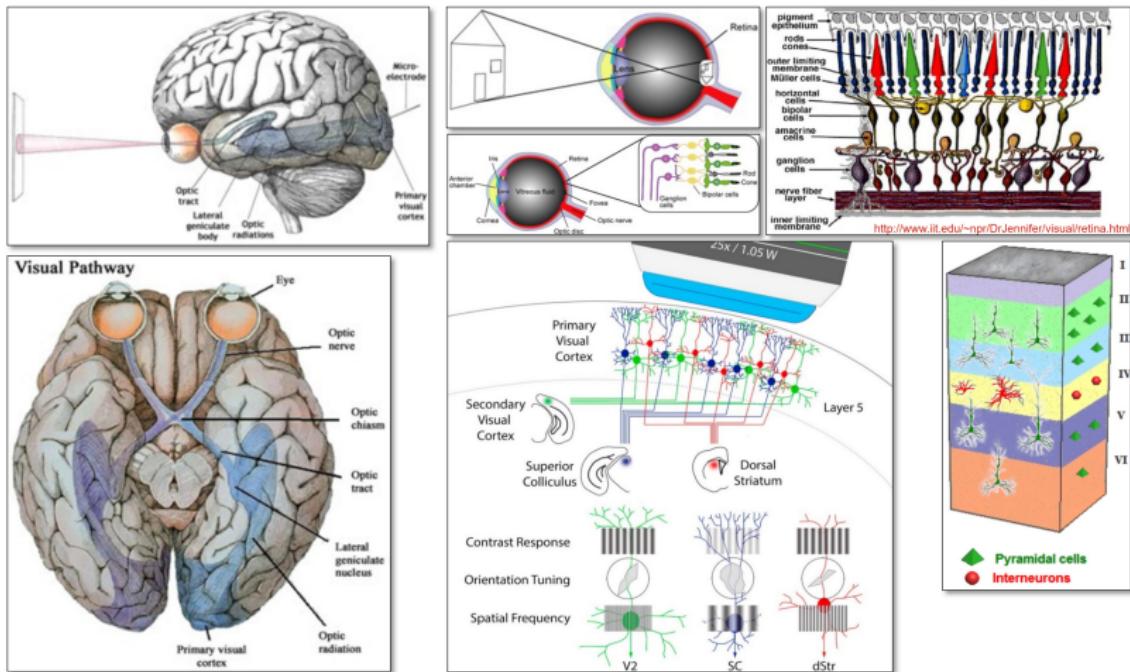
¿Qué vemos cuando vemos?



<https://mit6874.github.io/assets/sp2021/slides/103.pdf>



CNN inspiration in the 50s/60s: human/animal visual cortex

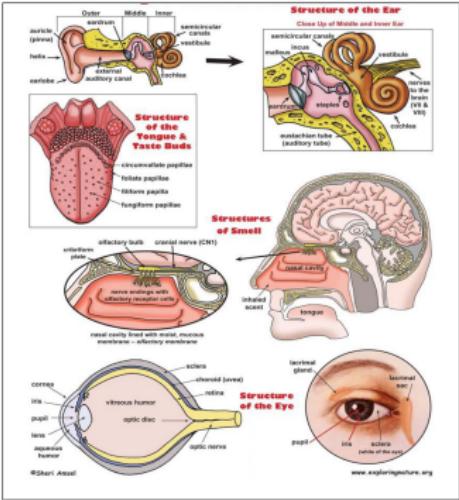


- Hubel/Wisell 1968 cat/monkey: (1) Receptive fields = local computation. (2) Simple cells = edge/orientation detectors. (3) Complex cells = position invariance/pooling
- Layers: pixels, edges (bands given slant, contrast edges), shapes, primitives, scenes
- Hierarchical abstractions, simple building blocks, local computation, learning, invariance

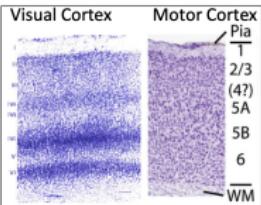


<https://youtu.be/I0Hayh06LJ4?si=4BKUxjJowyteSYu>

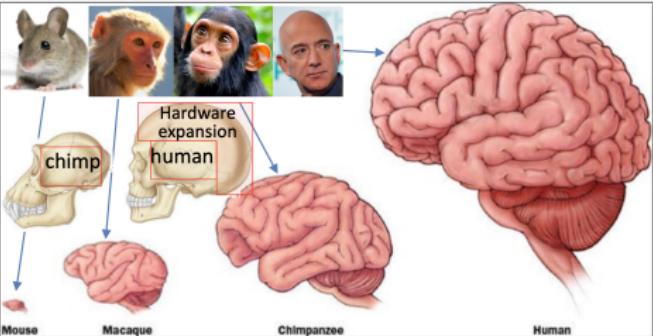
General “learning machine”, reused widely



- Hearing, taste, smell, sight, touch all re-use **similar learning architecture**



- Interchangeable circuitry**
- Auditory cortex learns to ‘see’ if sent visual signals
- Injury area tasks shift to uninjured areas



- Massive **recent expande** of human brain has re-used a relatively simple but general learning architecture



- Learning not fully-general, but **well-adapted to our world**
- Humans co-opted this circuitry to **many new applications**
- Modern tasks accessible to any homo sapiens (**70k years!**)
- ML still similar to animals: room for **architecture novelty!**



Algoritmos de Deep Learning del Curso

- Object Classification (CNN)
 - Object Detection (YOLO)
 - Tracking
 - Segmentation (UNet)
 - GAN
 - Anomaly Detection
 - Transformers
 - Facial Analysis
- 
- Face detection
 - Face recognition
 - Face restoration
 - Age recognition
 - Gender recognition
 - Expression recognition
 - Soft attributes
 - Face clustering
 - Pose estimation
 - Explainability

github.com/domingtonery/vision



Redes convolucionales para clasificación

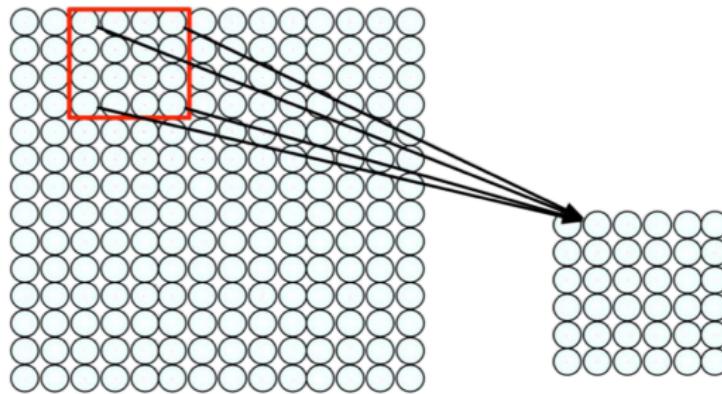


Típicos elementos en una red convolucional para clasificación

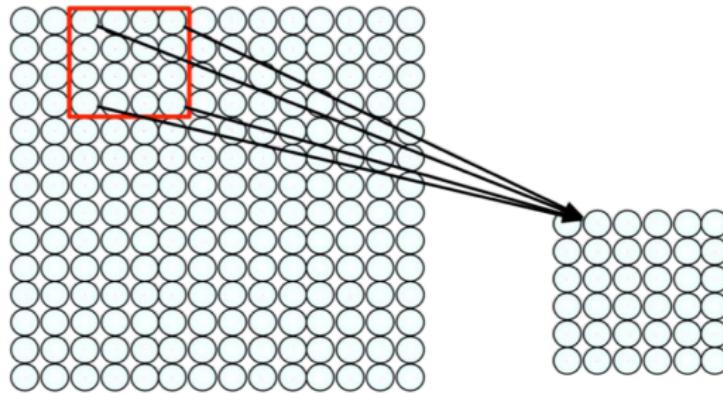
- Capa convolucional
- Uso de RELU como función de activación
- Capa de pooling (*max or mean*)
- Capa Fully connected y Softmax.



Idea general detrás de la convolución



Idea general detrás de la convolución



- Aplicar un set de pesos (filtros) moviéndose dentro de la imagen. Aplicar el filtro es hacer una suma ponderada
- Aplicar el sesgo o bias
- Aplicar la función de activación (en general no-lineal)



Filtros o kernels

Aplicar el filtro es multiplicar elemento a elemento y luego sumar

1	0	1
0	1	0
1	0	1

Filter / Kernel

1 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1 <small>$\times 1$</small>	0	0
0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1	0
0 <small>$\times 1$</small>	0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

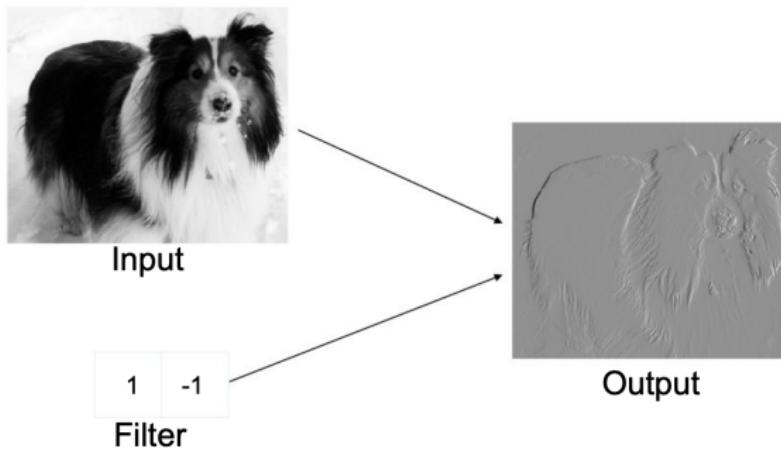
4		

Convolved Feature

<https://mit6874.github.io/assets/sp2021/slides/103.pdf>



Ejemplo de detección de bordes (*edges*)



(Goodfellow 2016)



Filtro para (*sharpen*)



Original



Sharpen

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix}$$

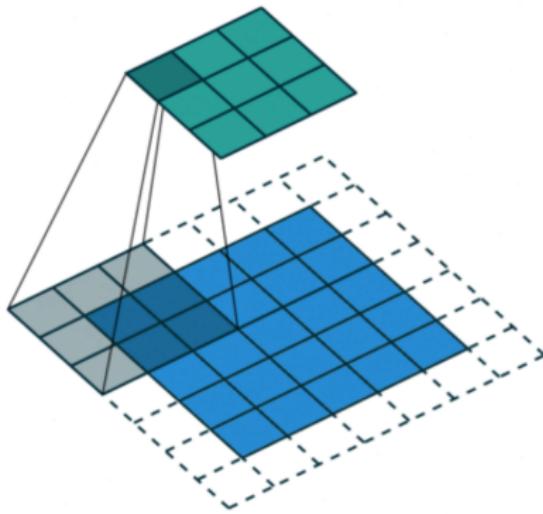
<https://mit6874.github.io/assets/sp2021/slides/l03.pdf>



En aprendizaje profundo, la red aprende los valores para los filtros, y se aplica el mismo filtro sobre toda la imagen



Convolución de 3x3 en una imagen de 5x5 usando un *padding* de 1

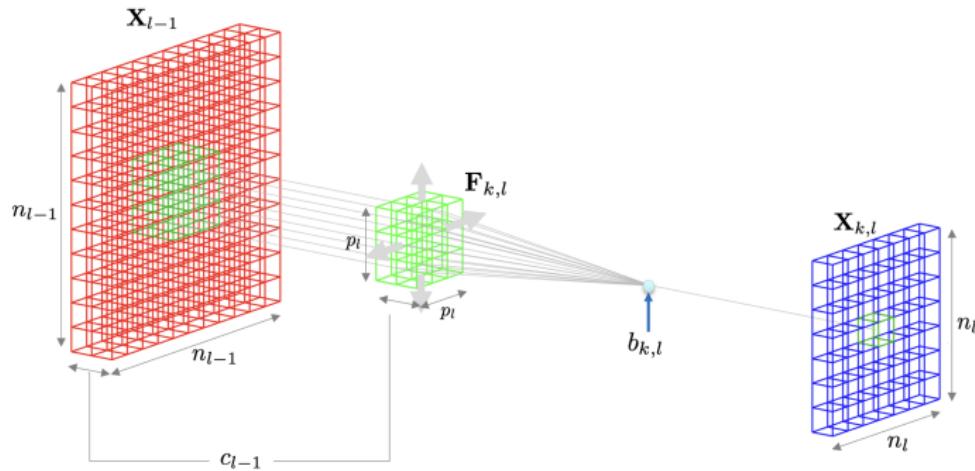


Stride es el tamaño del paso con que el filtro recorre la imagen.

<https://mit6874.github.io/assets/sp2021/slides/103.pdf>



Convolución en una imagen RGB

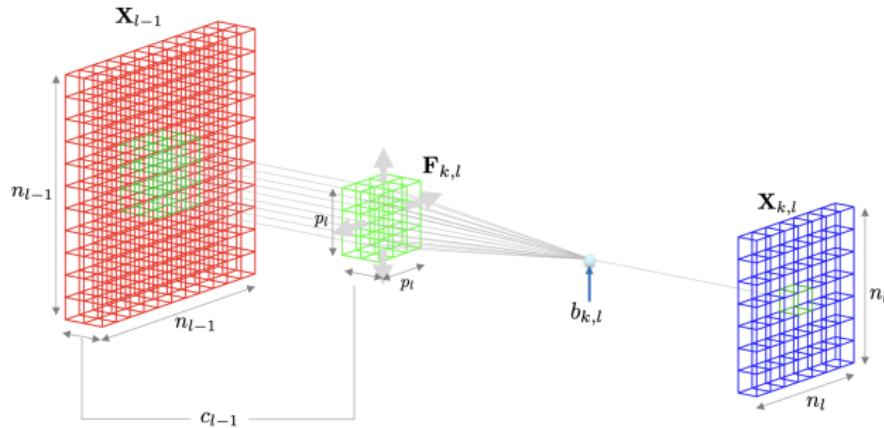


https://github.com/domingomery/vision/blob/master/clases/Cap03_DeepLearning/presentations/CV03_

CNN.pptx



Convolución en una imagen RGB



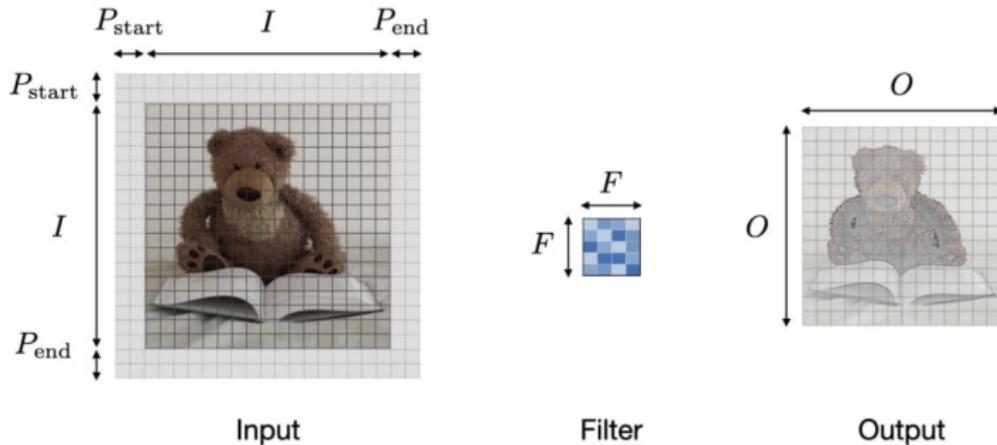
$$\mathbf{X}_{k,l} = \mathbf{X}_{l-1} * \mathbf{F}_{k,l} + b_{k,l} \quad \text{for } k = 1 \dots m_l$$

https://github.com/domingomery/vision/blob/master/clases/Cap03_DeepLearning/presentations/CV03_CNN.pptx



Dimensiones de salida de la capa de convolución

$$O = \frac{I - F + P_{\text{start}} + P_{\text{end}}}{S} + 1$$



<https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks>

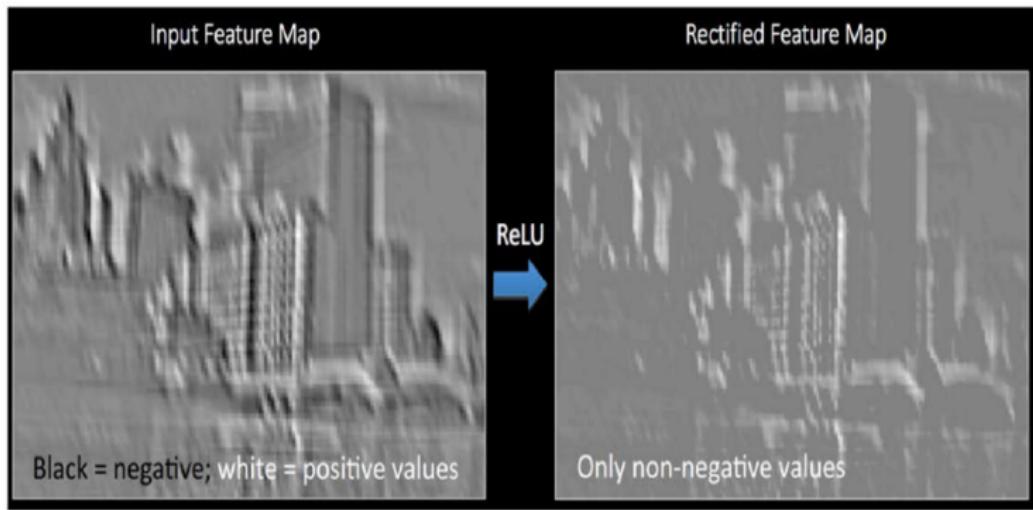


Típicos elementos en una red convolucional para clasificación

- Capa convolucional
- Uso de RELU como función de activación
- Capa de pooling (*max or mean*)
- Capa Fully connected y Softmax.



Después de cada capa convolucional aplicaremos la función de activación (usualmente ReLU)



<https://mit6874.github.io/assets/sp2021/slides/l03.pdf>

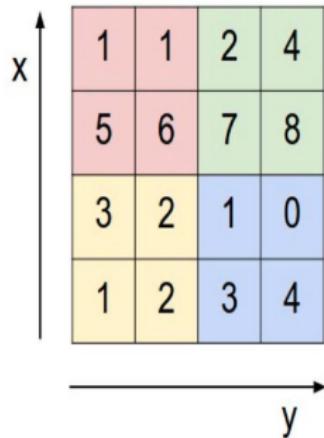


Típicos elementos en una red convolucional para clasificación

- Capa convolucional
- Uso de RELU como función de activación
- Capa de pooling (*max or mean*)
- Capa Fully connected y Softmax.



Pooling



max pool with 2x2 filters
and stride 2

```
tf.keras.layers.Max  
Pool2D(  
    pool_size=(2, 2),  
    strides=2  
)
```

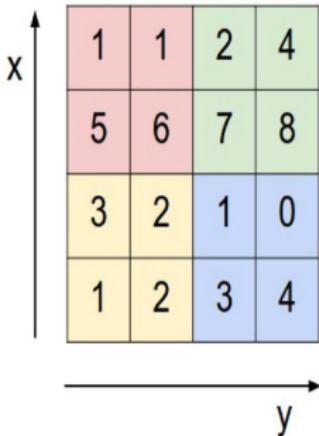
6	8
3	4

- 1) Reduced dimensionality
- 2) Spatial invariance

Max Pooling, average pooling

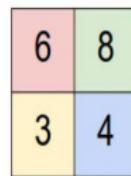


Pooling



max pool with 2x2 filters
and stride 2

```
tf.keras.layers.Max  
Pool2D(  
    pool_size=(2, 2),  
    strides=2  
)
```



- 1) Reduced dimensionality
- 2) Spatial invariance

Max Pooling, average pooling

¿Necesito aprender parámetros para esta capa?

<https://mit6874.github.io/assets/sp2021/slides/103.pdf>

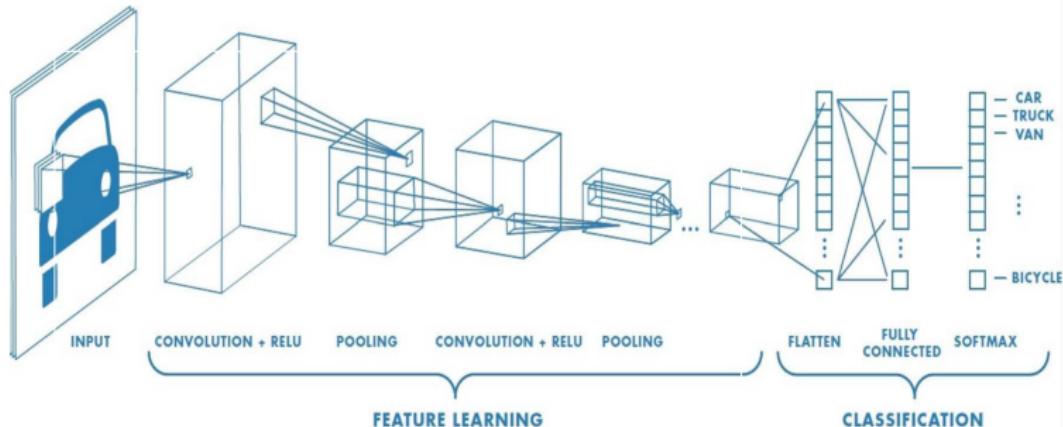


Típicos elementos en una red convolucional para clasificación

- Capa convolucional
- Uso de RELU como función de activación
- Capa de pooling (*max or mean*)
- Capa Fully connected y Softmax.



Softmax entrega la probabilidad de cada clase



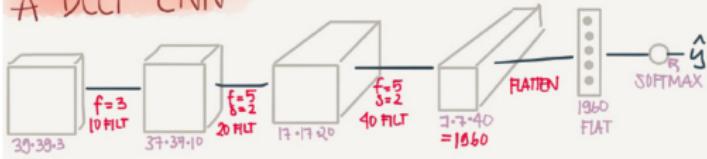
$$\text{softmax}(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

<https://mit6874.github.io/assets/sp2021/slides/l03.pdf>



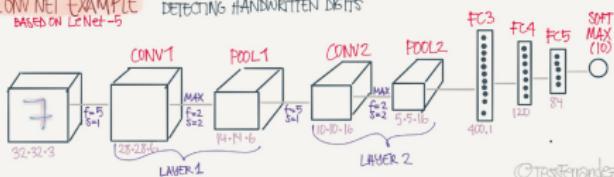
Resumen

A DEEP CNN

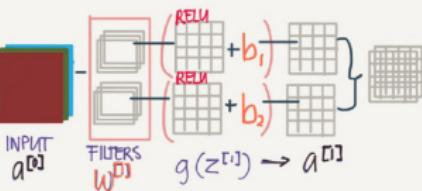


A LOT OF THE WORK IS FIGURING OUT HYPERPARAMS
= # FILTERS, STRIDE, PADDING ETC
TYPICALLY SIZE → TREND DOWN
FILTERS → TREND UP

CONV NET EXAMPLE BASED ON LeNet-5



ONE CONV. NET LAYER

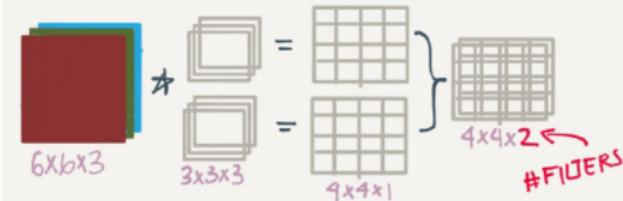


NOTE IT DOESN'T MATTER HOW BIG THE INPUT IS - THE LEARNABLE PARAMS W & b ONLY DEPEND ON THE # OF FILTERS AND THEIR SIZES.

$$W = 3 \cdot 3 \cdot 3 \cdot 2 = 54 \quad \{ 56 \text{ PARAMS TO LEARN}$$
$$b = 2$$

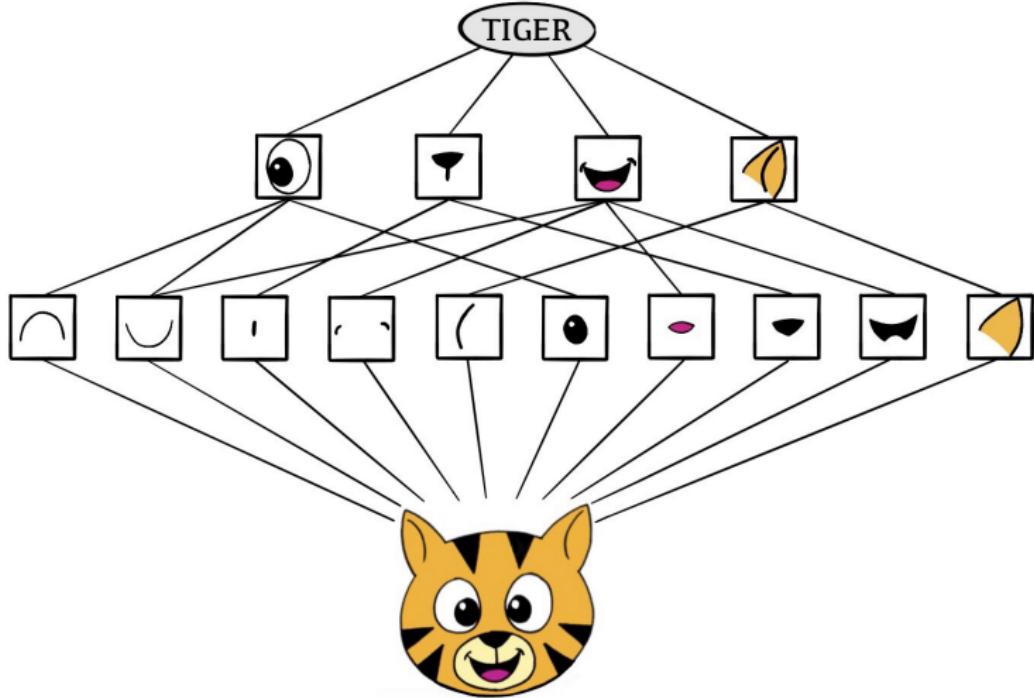
MULTIPLE FILTERS

DETECTING MULTIPLE FEATURES AT A TIME



<https://mit6874.github.io/assets/sp2021/slides/103.pdf>

Distintas capas, distinto nivel de abstracción



<https://www.statlearning.com/>

