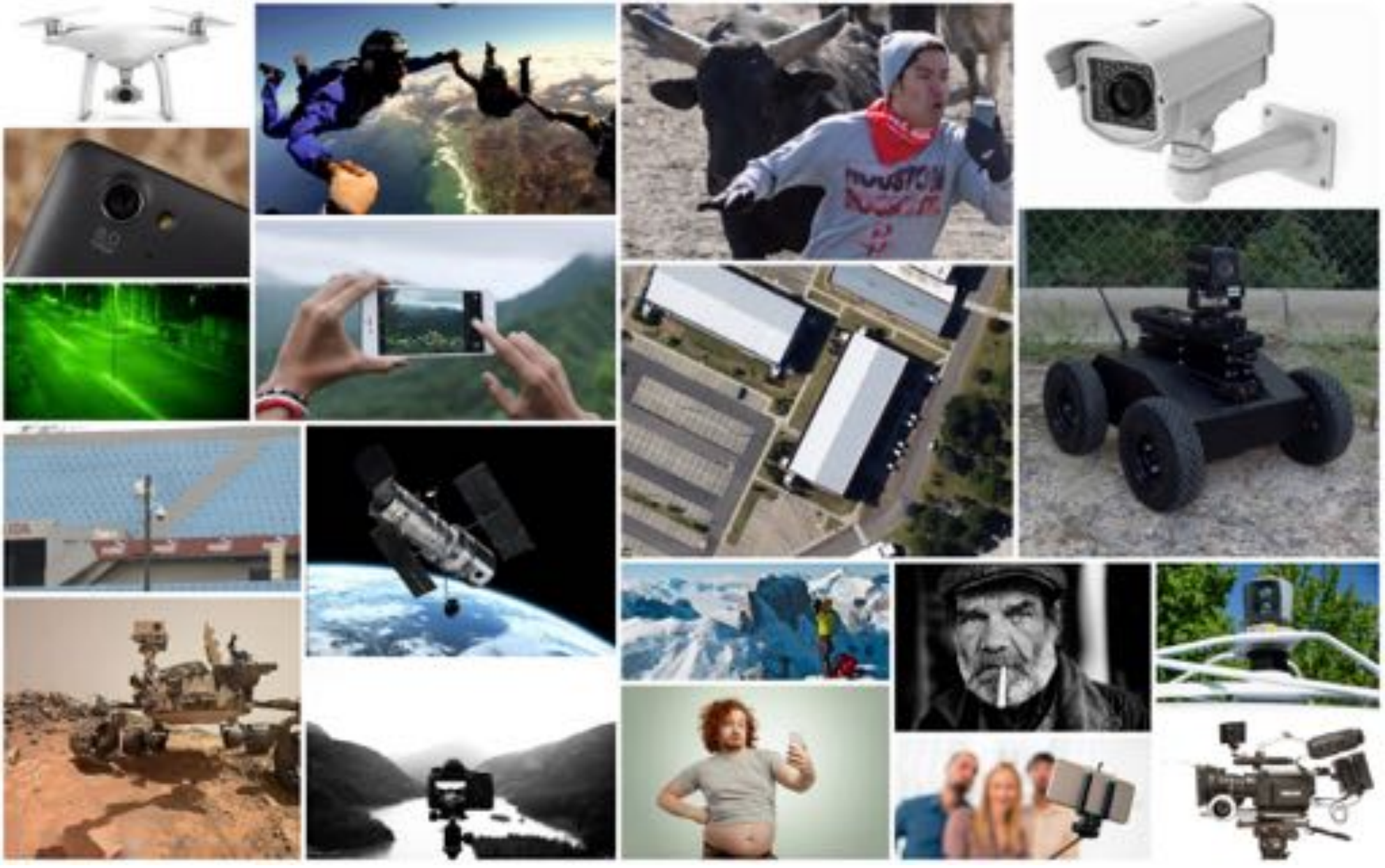
Procesamiento de Imágenes y Visión Artificial (con un toque de Aprendizaje Automático)

Departamento de Procesamiento de Señales
Instituto de Ingeniería Eléctrica

Aprendizaje Profundo para Visión Artificial

Mauricio Delbracio - mdelbrafing.edu.uy

Universo de imágenes y videos



Masividad de datos

Visión Artificial - Ejemplos

Viola and Jones. “Robust real-time face detection” IJCV 2004

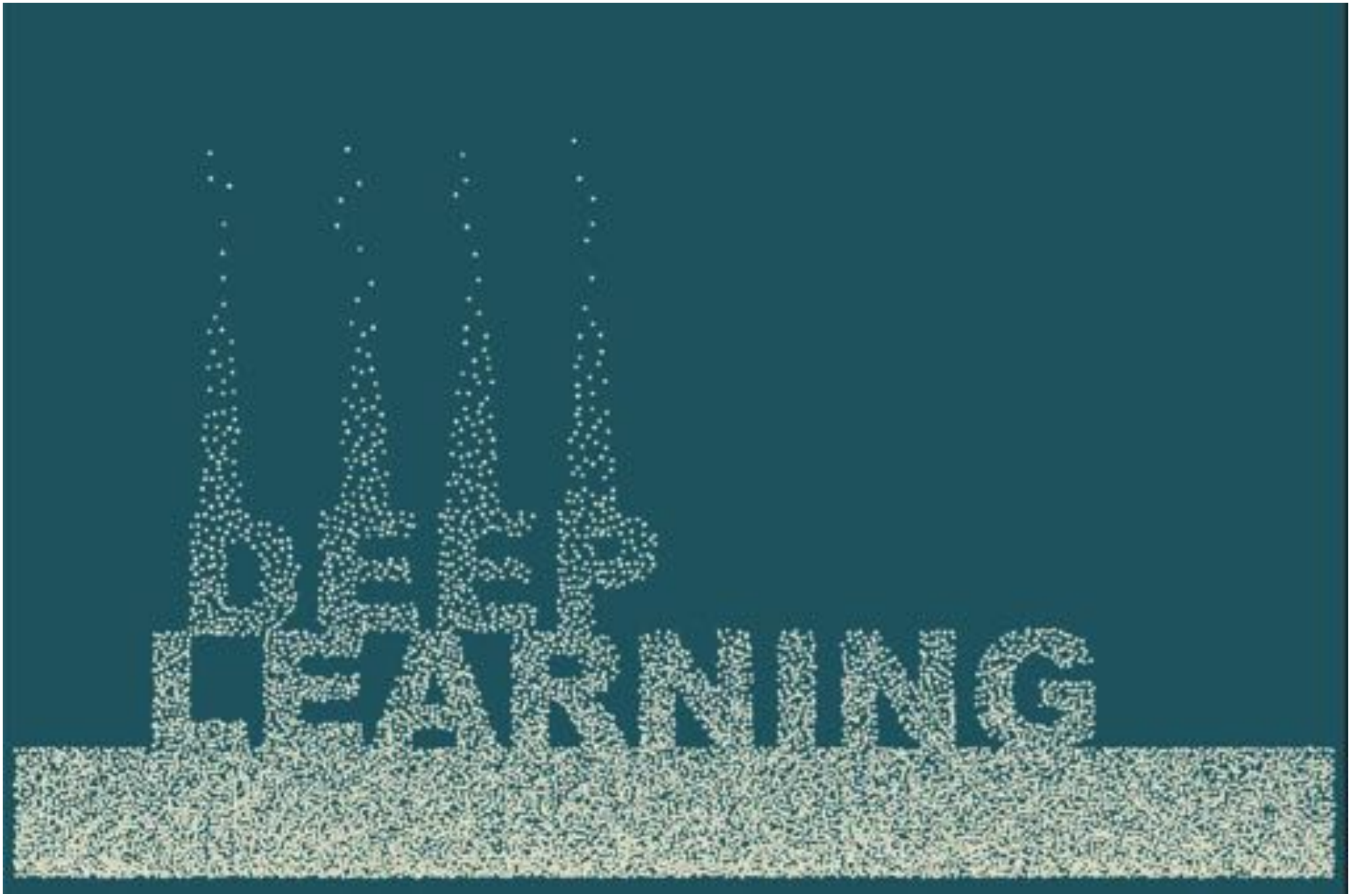
Visión Artificial - Ejemplos

Farabet, et al. “Learning hierarchical features for scene labeling” PAMI 2013

Visión Artificial - Ejemplos

Cao, et al. “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields” arXiv 2016

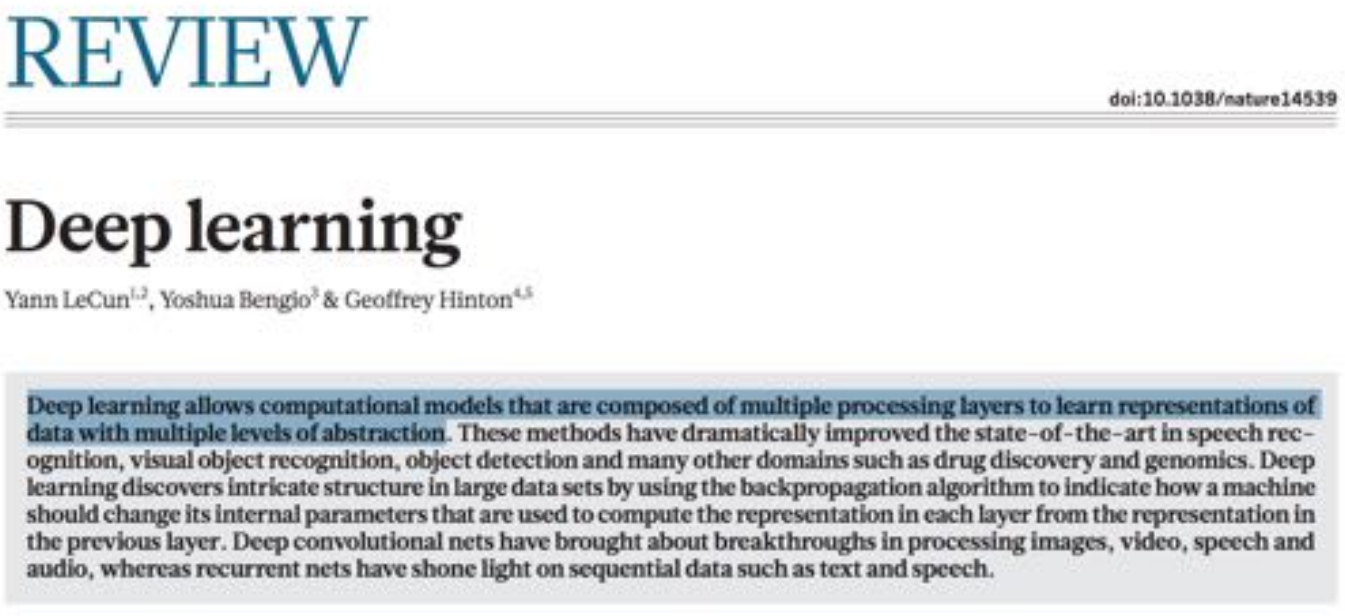
¿Qué es el aprendizaje profundo?



Aprendizaje Profundo (*deep learning*)

*"Deep learning allows computational models that are composed of multiple processing layers to **learn representations** of data with **multiple levels of abstraction**."*

– Yann LeCun, Yoshua Bengio and Geoffrey Hinton, Deep Learning, *Nature*, 2015



REVIEW

doi:10.1038/nature14539

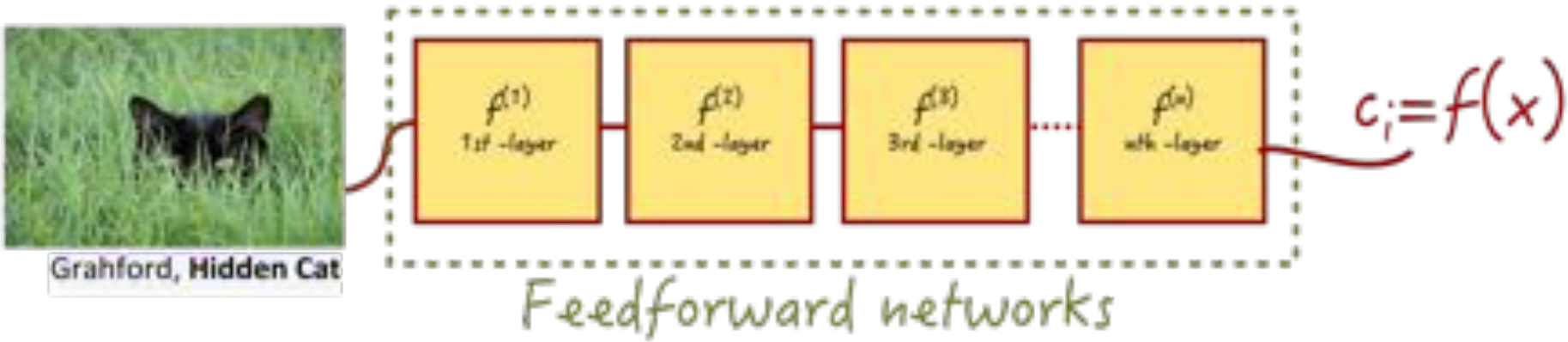
Deep learning

Yann LeCun[1,2], Yoshua Bengio[3] & Geoffrey Hinton[4,5]

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

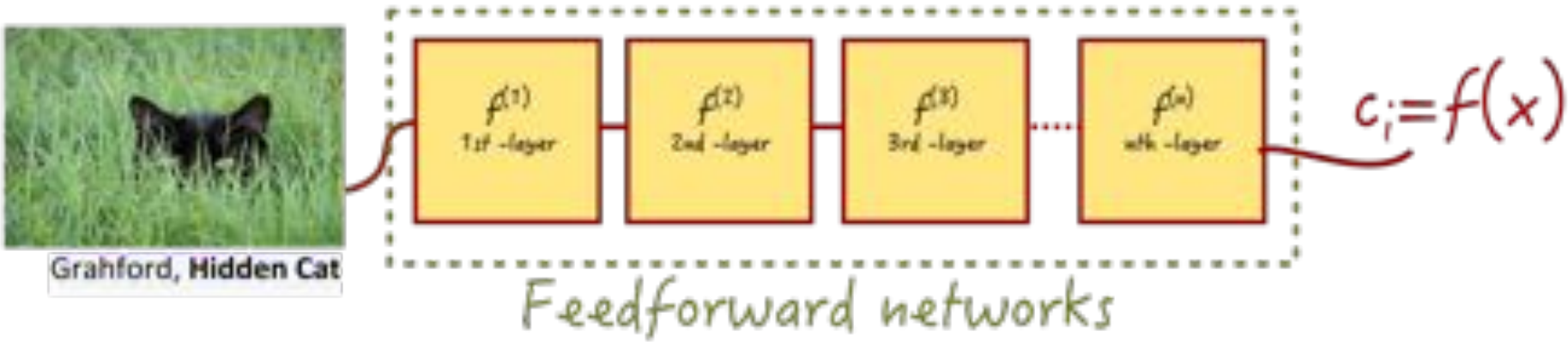
Aprendizaje Profundo (*deep learning*)

- Cascada de capas de procesamiento no lineal para extraer y transformar variables.



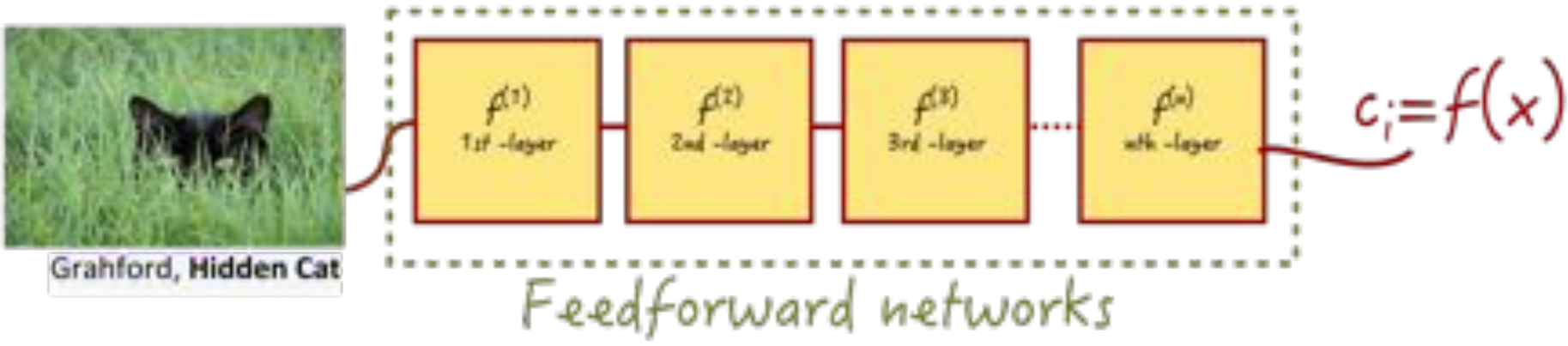
Aprendizaje Profundo (*deep learning*)

- Cascada de capas de procesamiento no lineal para extraer y transformar variables.
- Entrada a una capa = salida de capa anterior:
  transforma representación en una de mayor nivel de abstracción



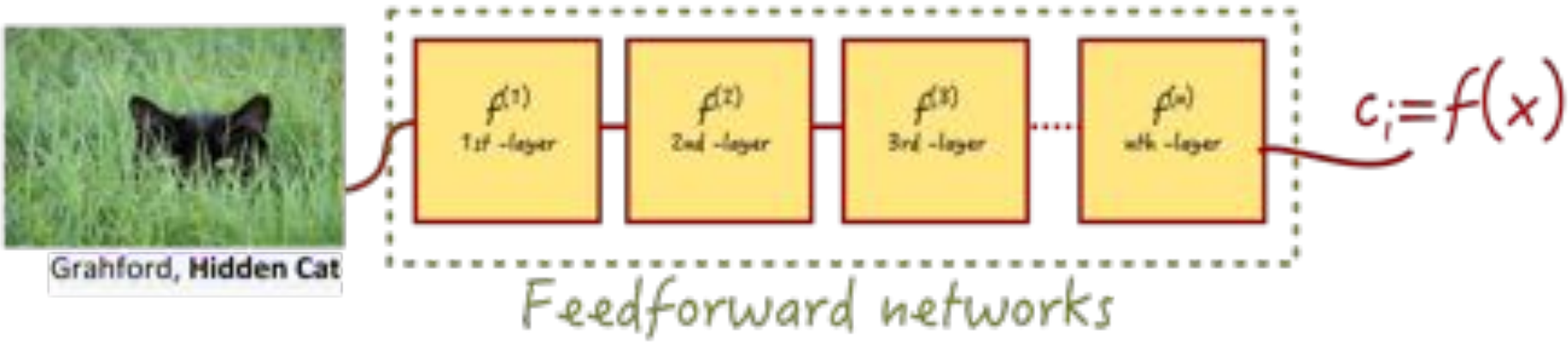
Aprendizaje Profundo (*deep learning*)

- Cascada de capas de procesamiento no lineal para extraer y transformar variables.
- Entrada a una capa = salida de capa anterior:
  transforma representación en una de mayor nivel de abstracción
- Múltiples niveles de representación se corresponden con diferentes niveles de abstracción (jerarquía de conceptos).



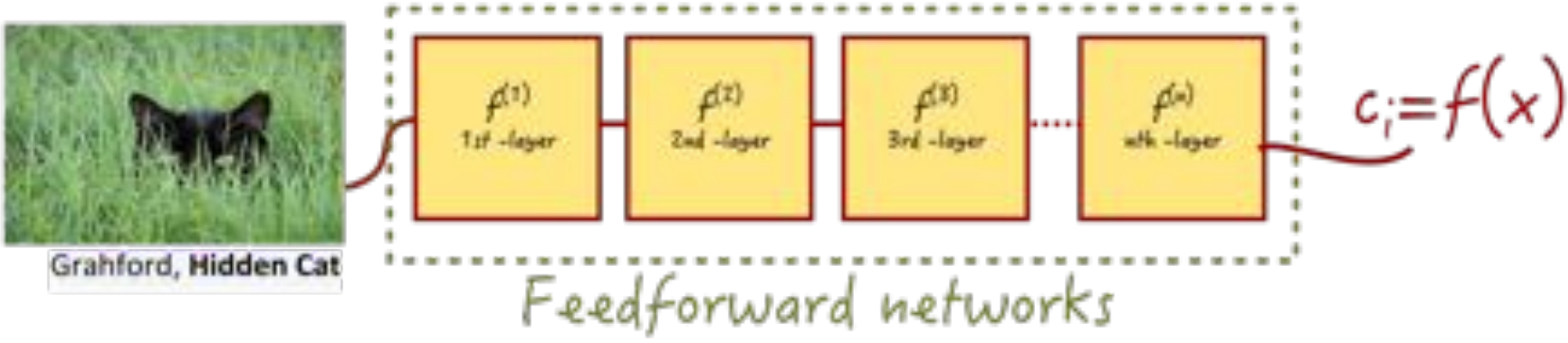
Aprendizaje Profundo (*deep learning*)

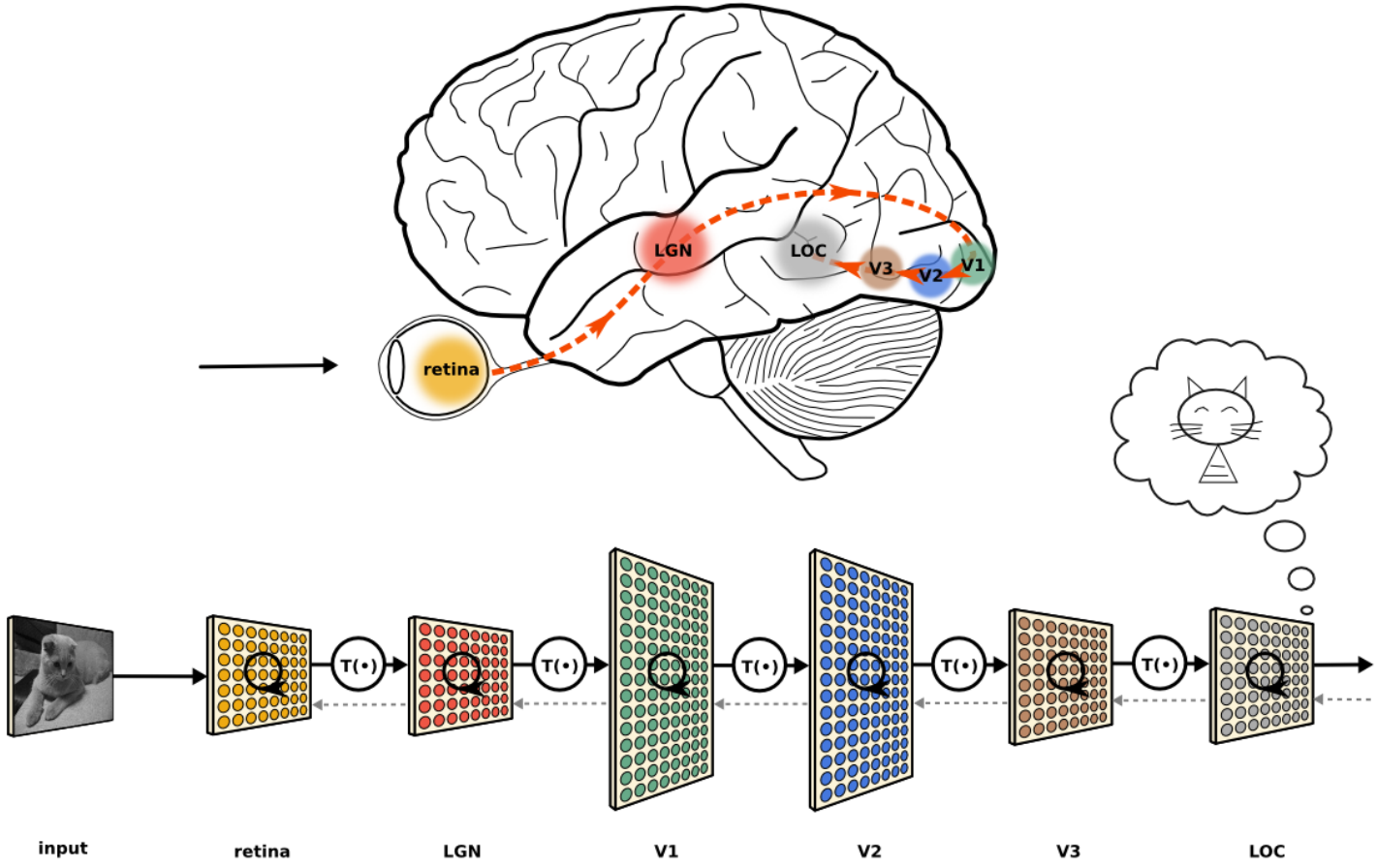
- Cascada de capas de procesamiento no lineal para extraer y transformar variables.
- Entrada a una capa = salida de capa anterior:
  transforma representación en una de mayor nivel de abstracción
- Múltiples niveles de representación se corresponden con diferentes niveles de abstracción (jerarquía de conceptos).
- **Imágenes:** Pixel → bordes → partes → objetos



Aprendizaje Profundo (*deep learning*)

- Cascada de capas de procesamiento no lineal para extraer y transformar variables.
- Entrada a una capa = salida de capa anterior:
  transforma representación en una de mayor nivel de abstracción
- Múltiples niveles de representación se corresponden con diferentes niveles de abstracción (jerarquía de conceptos).
- **Imágenes:** Pixel → bordes → partes → objetos
- Entrenamiento de punta a punta utilizando un conjunto de ejemplos llamado datos de entrenamiento

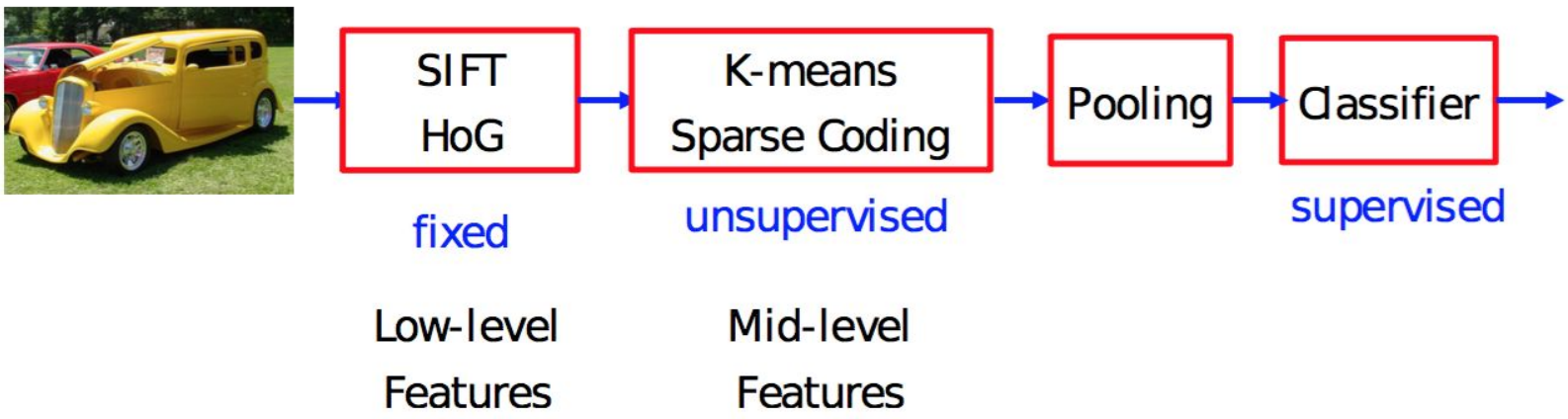




- Representaciones intermedias: Retina - LGN - V1 - V2 - V4 - PIT - AIT
- Del orden de 100.000 millones de neuronas
- Cerebro humano no funciona como las redes de *Deep Learning*

Aprendizaje **no** profundo

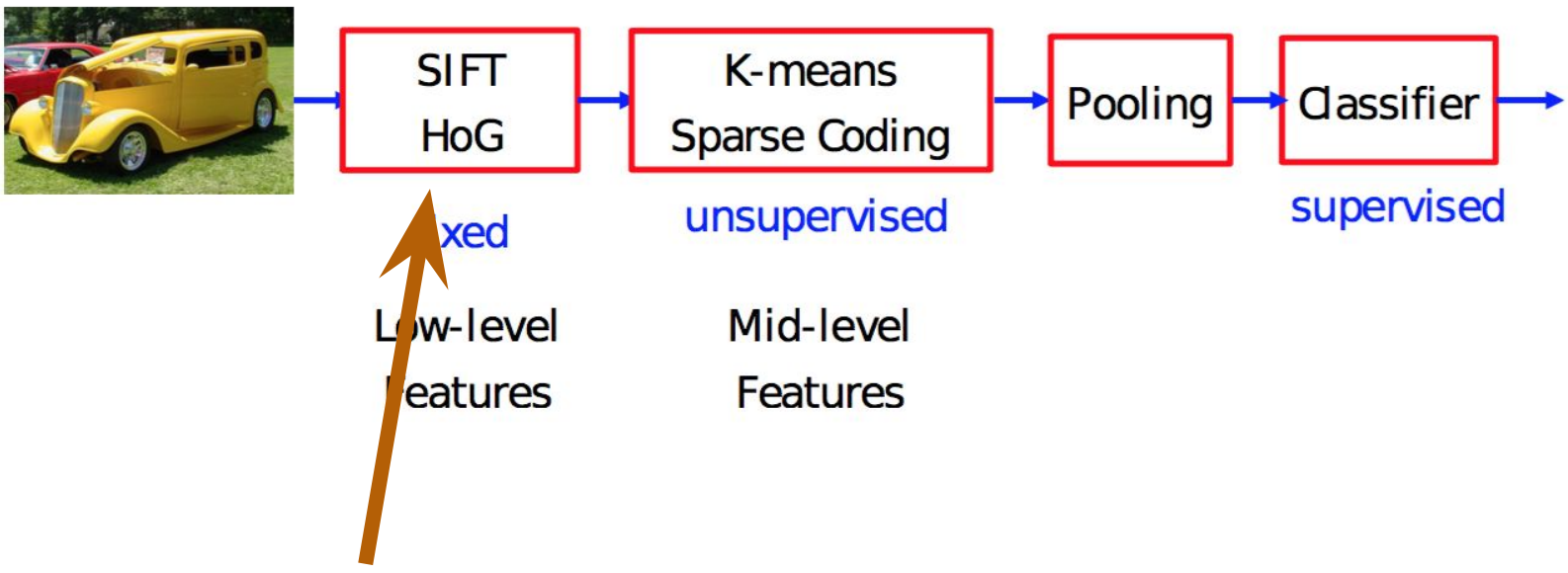
Reconocimiento de imágenes, principios del 2000 hasta el 2012:



Ejemplo tomado de charlas de Y. Lecun.

Aprendizaje **no** profundo

Reconocimiento de imágenes, principios del 2000 hasta el 2012:

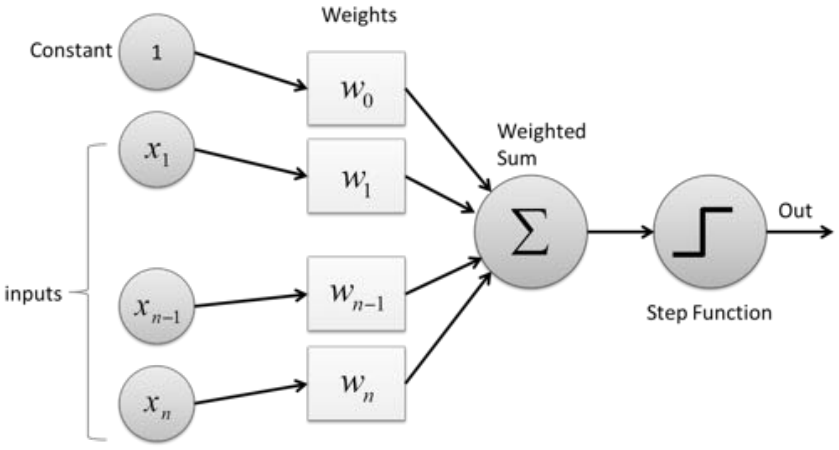


Ejemplo tomado de charlas de Y. Lecun.

Breve historia del (último) resurgimiento de las redes neuronales

Perceptrón (1957)

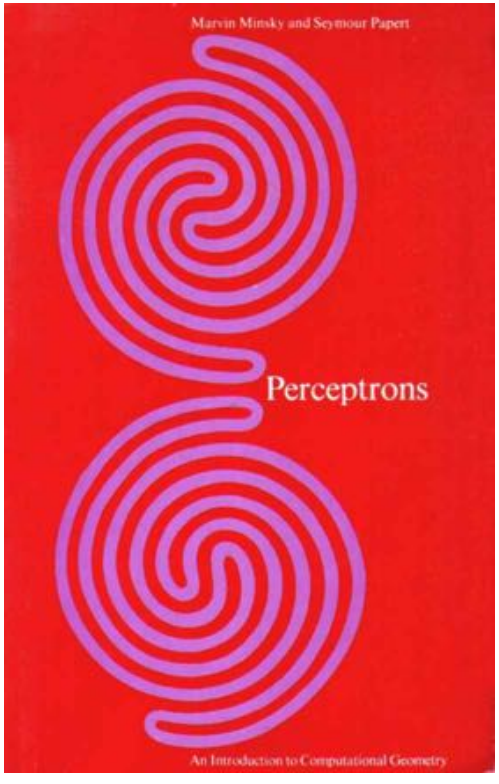
- Creada por Rosenblatt, 1957, en Universidad Cornell
- De las primeras "máquinas" capaces de "aprender"
- Una única capa (separa datos linealmente separables)
- Modelo actual es similar (pero multi-capa)

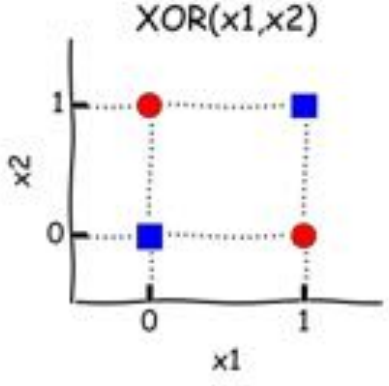


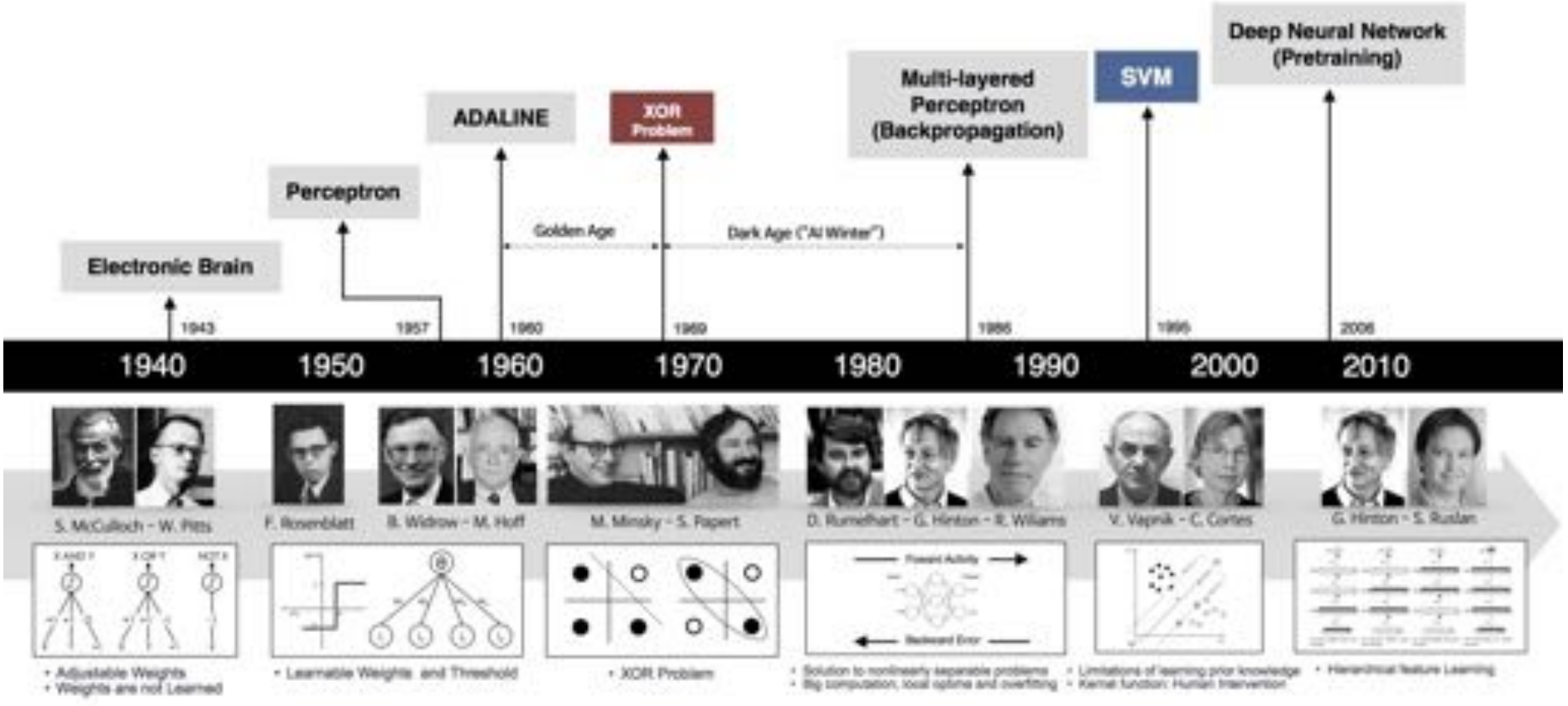
Rosenblatt, 1958

Marvin Minsky y Seymour Papert (1969)

- Una máquina sólo puede **reconocer** lo que sabe **representar**
- Perceptrones sólo pueden representar patrones linealmente separables
- No pueden representar XOR



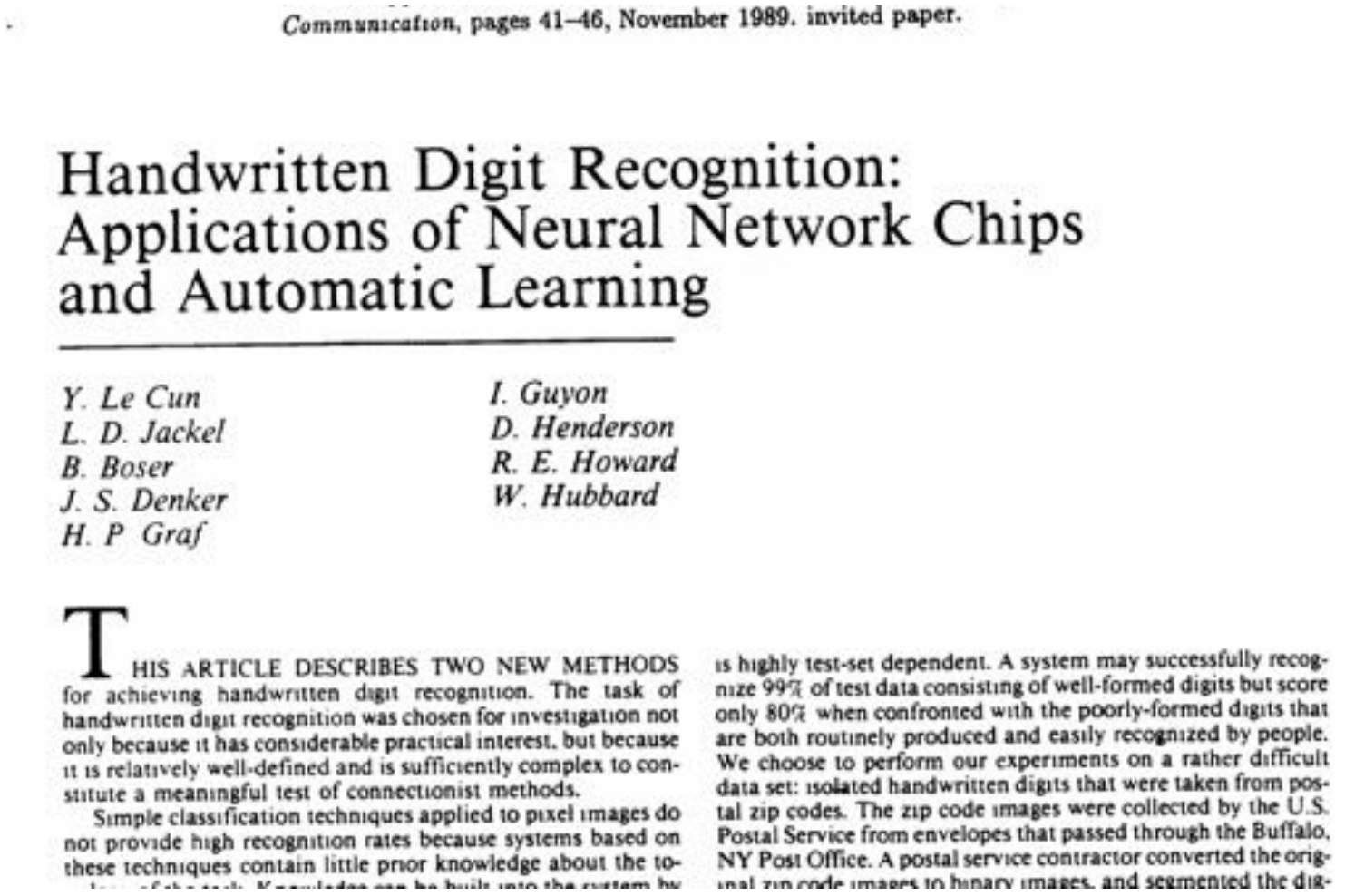




Andrew L. Beam

En los 90s

Primera red convolucional entrenada con backpropagation



Communication, pages 41–46, November 1989. invited paper.

Handwritten Digit Recognition: Applications of Neural Network Chips and Automatic Learning

Y. Le Cun
L. D. Jackel
B. Boser
J. S. Denker
H. P Graf

I. Guyon
D. Henderson
R. E. Howard
W. Hubbard

THIS ARTICLE DESCRIBES TWO NEW METHODS for achieving handwritten digit recognition. The task of handwritten digit recognition was chosen for investigation not only because it has considerable practical interest, but because it is relatively well-defined and is sufficiently complex to constitute a meaningful test of connectionist methods.

Simple classification techniques applied to pixel images do not provide high recognition rates because systems based on these techniques contain little prior knowledge about the to-

is highly test-set dependent. A system may successfully recognize 99% of test data consisting of well-formed digits but score only 80% when confronted with the poorly-formed digits that are both routinely produced and easily recognized by people. We choose to perform our experiments on a rather difficult data set: isolated handwritten digits that were taken from postal zip codes. The zip code images were collected by the U.S. Postal Service from envelopes that passed through the Buffalo, NY Post Office. A postal service contractor converted the orig-

Le Cun, Y., L. Jackel, B. Boser, J. Denker, H. Graf, I. Guyon, D. Henderson, R. Howard, and W. Hubbard. IEEE Com. Mag. 1989

En los 90s

- En los '90 resolvía problemas prácticos **serios** (reconocimiento de dígitos)
- En otros dominios era también competitiva, pero no se uso en producción

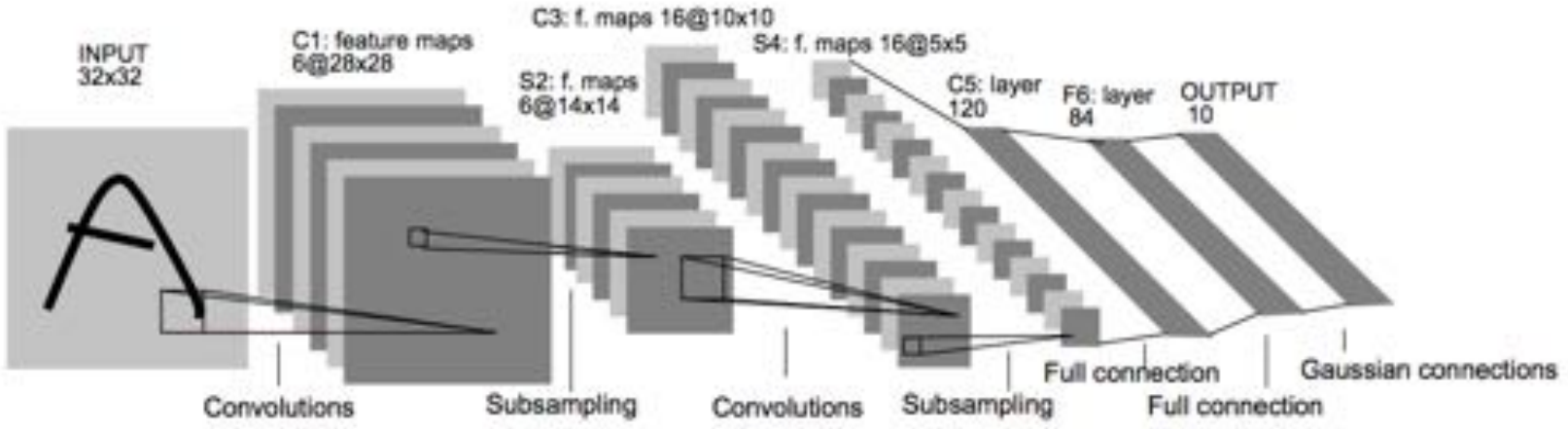


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeNet-5

2006 - Hinton et al.

Se consolida el término *Deep*

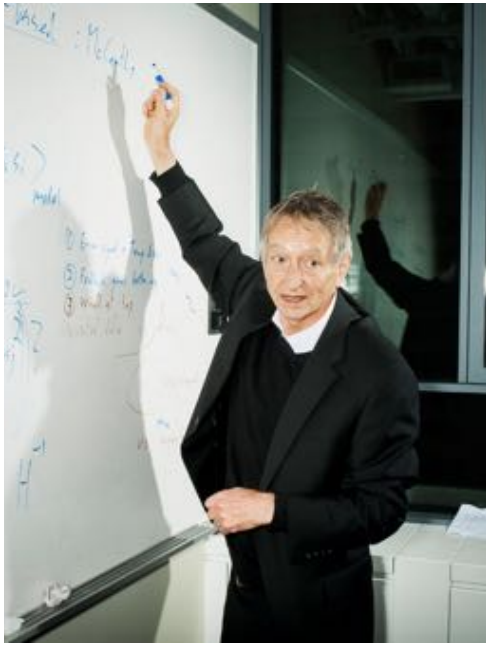
LETTER — Communicated by Yann Le Cun

A Fast Learning Algorithm for Deep Belief Nets

Geoffrey E. Hinton
hinton@cs.toronto.edu
Simon Osindero
osindero@cs.toronto.edu
Department of Computer Science, University of Toronto, Toronto, Canada M5S 3G4

Yee-Whye Teh
tehyw@comp.nus.edu.sg
Department of Computer Science, National University of Singapore, Singapore 117543

We show how to use "complementary priors" to eliminate the explaining-away effects that make inference difficult in densely connected belief nets that have many hidden layers. Using complementary priors, we derive a fast, greedy algorithm that can learn deep, directed belief networks one layer at a time, provided the top two layers form an undirected associative memory. The fast, greedy algorithm is used to initialize a slower learning procedure that fine-tunes the weights using a contrastive version of the wake-sleep algorithm. After fine-tuning, a network with three hidden layers forms a very good generative model of the joint distribution of handwritten digit images and their labels. This generative model gives better digit classification than the best discriminative learning algorithms. The low-dimensional manifolds on which the digits lie are modeled by long ravines in the free-energy landscape of the top-level associative memory, and it is easy to explore these ravines by using the directed connections to display what the associative memory has in mind.



Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh.
"A fast learning algorithm for deep belief nets." Neural computation, 2006.

Antes del 2012

- Resultados competitivos
- Trabajos con redes neuronales eran **minoría**

Face detection [Vaillant et al '93,'94 ; Osadchy et al, '03, '04, '07]

Antes del 2012

Principales razones de su falta de popularidad:

- Demasiados parámetros para aprender (millones).
  "Yo conozco el problema, mis *features* son mejores!"

Antes del 2012

Principales razones de su falta de popularidad:

- Demasiados parámetros para aprender (millones).
  "Yo conozco el problema, mis *features* son mejores!"
- ¿Optimización **no convexa**? Ufff

Antes del 2012

Principales razones de su falta de popularidad:

- Demasiados parámetros para aprender (millones).
  "Yo conozco el problema, mis *features* son mejores!"
- ¿Optimización **no convexa**? Ufff
- Modelo de caja-negra: **no se puede interpretar**.

Antes del 2012

Principales razones de su falta de popularidad:

- Demasiados parámetros para aprender (millones).
  "Yo conozco el problema, mis *features* son mejores!"
- ¿Optimización **no convexa**? Ufff
- Modelo de caja-negra: **no se puede interpretar**.
- Dificultad para obtener/reproducir resultados, falta de **bibliotecas de código sólidas.**

Primer *break-through*: Reconocimiento de voz

- Nuevo método en base a redes profundas produce excelente resultados
- Incluido en Android en 2012.

Deep Belief Networks for phone recognition

Abdel-rahman Mohamed, George Dahl, and Geoffrey Hinton
Department of Computer Science
University of Toronto
{asamir,gdahl,hinton}@cs.toronto.edu

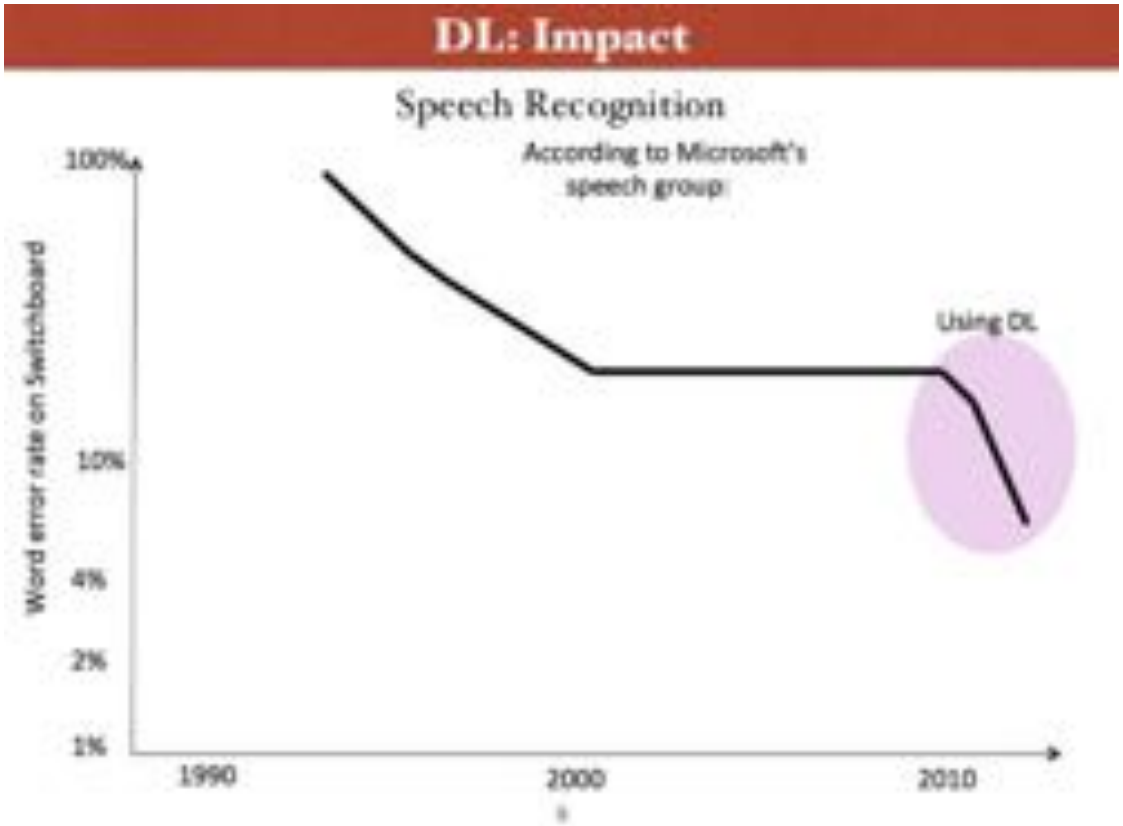
Abstract

Hidden Markov Models (HMMs) have been the state-of-the-art techniques for acoustic modeling despite their unrealistic independence assumptions and the very limited representational capacity of their hidden states. There are many proposals in the research community for deeper models that are capable of modeling the many types of variability present in the speech generation process. Deep Belief Networks (DBNs) have recently proved to be very effective for a variety of machine learning problems and this paper applies DBNs to acoustic modeling. On the standard TIMIT corpus, DBNs consistently outperform other techniques and the best DBN achieves a phone error rate (PER) of 23.0% on the TIMIT core test set.

Mohamed, Abdel-rahman, George Dahl, and Geoffrey Hinton.
"Deep belief networks for phone recognition." NIPS, 2009.

Primer *break-through*: Reconocimiento de voz

- Nuevo método en base a redes profundas produce excelente resultados
- Incluido en Android en 2012.



Mohamed, Abdel-rahman, George Dahl, and Geoffrey Hinton. "Deep belief networks for phone recognition." NIPS, 2009.

ImageNet Large Scale Visual Recognition Competition

Competencia de reconocimiento de objetos

- 1000 categorías de imágenes; 1.2 millones de imágenes de entrenamiento
- Respuesta correcta entre las primeras 5 imágenes.

ImageNet Large Scale Visual Recognition Competition

Competencia de reconocimiento de objetos

- 1000 categorías de imágenes; 1.2 millones de imágenes de entrenamiento
- Respuesta correcta entre las primeras 5 imágenes.

2012 Teams	%error
Supervision (Toronto)	15.3
ISI (Tokyo)	26.1
VGG (Oxford)	26.9
XRCE/INRIA	27.0
UvA (Amsterdam)	29.6
INRIA/LEAR	33.4

Técnicas que usan DL - Otros métodos

Krizhevsky, A., Sutskever, I. and Hinton, G.E.,
"Imagenet classification with deep convolutional neural networks." NIPS, 2012

ImageNet Large Scale Visual Recognition Competition

Competencia de reconocimiento de objetos

- 1000 categorías de imágenes; 1.2 millones de imágenes de entrenamiento
- Respuesta correcta entre las primeras 5 imágenes.

2012 Teams	%error
Supervision (Toronto)	15.3
ISI (Tokyo)	26.1
VGG (Oxford)	26.9
XRCE/INRIA	27.0
UvA (Amsterdam)	29.6
INRIA/LEAR	33.4

2013 Teams	%error
Clarifai (NYU spinoff)	11.7
NUS (singapore)	12.9
Zeiler-Fergus (NYU)	13.5
A. Howard	13.5
OverFeat (NYU)	14.1
UvA (Amsterdam)	14.2
Adobe	15.2
VGG (Oxford)	15.2
VGG (Oxford)	23.0

Técnicas que usan DL - Otros métodos

Krizhevsky, A., Sutskever, I. and Hinton, G.E.,
"Imagenet classification with deep convolutional neural networks." NIPS, 2012

ImageNet Large Scale Visual Recognition Competition

Competencia de reconocimiento de objetos

- 1000 categorías de imágenes; 1.2 millones de imágenes de entrenamiento
- Respuesta correcta entre las primeras 5 imágenes.

2012 Teams	%error
Supervision (Toronto)	15.3
ISI (Tokyo)	26.1
VGG (Oxford)	26.9
XRCE/INRIA	27.0
UvA (Amsterdam)	29.6
INRIA/LEAR	33.4

2013 Teams	%error
Clarifai (NYU spinoff)	11.7
NUS (singapore)	12.9
Zeiler-Fergus (NYU)	13.5
A. Howard	13.5
OverFeat (NYU)	14.1
UvA (Amsterdam)	14.2
Adobe	15.2
VGG (Oxford)	15.2
VGG (Oxford)	23.0

2014 Teams	%error
GoogLeNet	6.6
VGG (Oxford)	7.3
MSRA	8.0
A. Howard	8.1
DeeperVision	9.5
NUS-BST	9.7
TTIC-ECP	10.2
XYZ	11.2
UvA	12.1

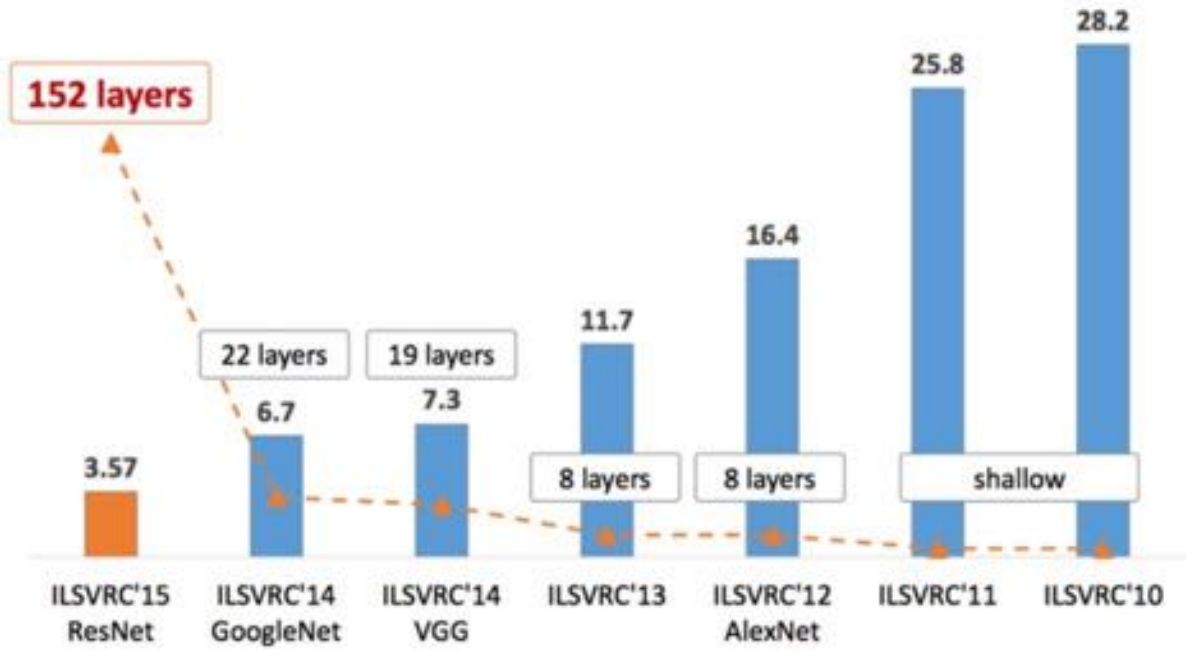
Técnicas que usan DL - Otros métodos

Krizhevsky, A., Sutskever, I. and Hinton, G.E.,
"Imagenet classification with deep convolutional neural networks." NIPS, 2012

ImageNet Large Scale Visual Recognition Competition

Competencia de reconocimiento de objetos

- 1000 categorías de imágenes; 1.2 millones de imágenes de entrenamiento
- Respuesta correcta entre las primeras 5 imágenes.



Krizhevsky, A., Sutskever, I. and Hinton, G.E.,
"Imagenet classification with deep convolutional neural networks." NIPS, 2012

Redes Neuronales Prealimentadas

- **Redes neuronales *feedforward*** o perceptrones multicapa
  base del Aprendizaje Profundo.

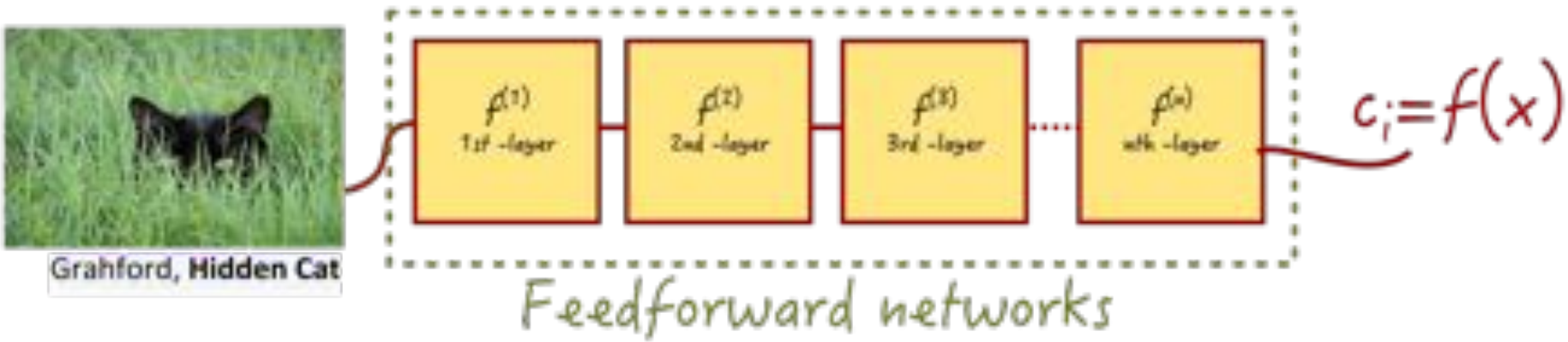
Redes Neuronales Prealimentadas

- **Redes neuronales *feedforward*** o perceptrones multicapa
  base del Aprendizaje Profundo.

- Mapeo $\hat{y} = f(\mathbf{x}; \theta)$ formado por capas (imágenes en categorías)

$$f(\mathbf{x}) := f^{(n)}\left(f^{(n-1)}\left(\cdots f^{(2)}\left(f^{(1)}\left(\mathbf{x}\right)\right)\cdots\right)\right),$$

  donde $f^{(1)}$ es la primera capa, $f^{(2)}$ la segunda ...



Redes Neuronales Prealimentadas

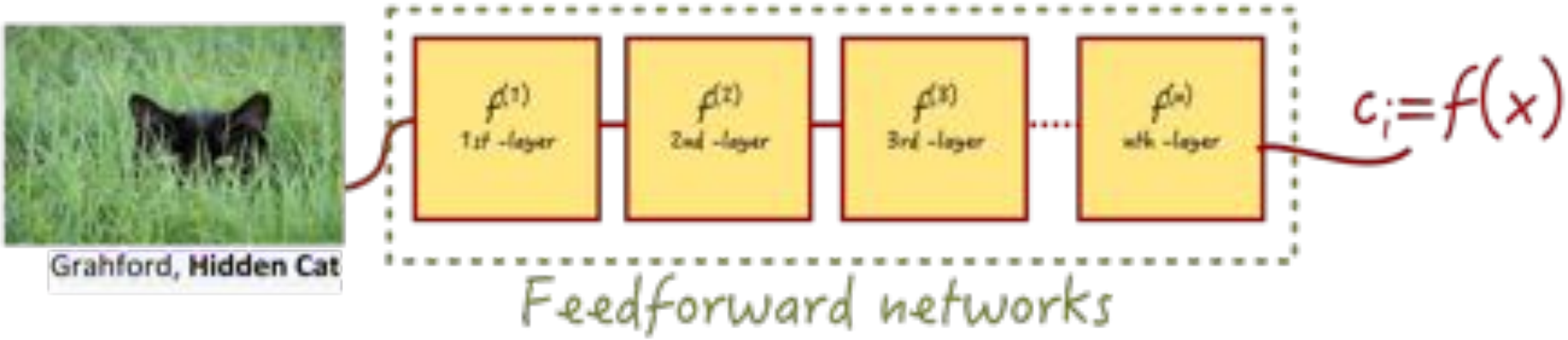
- **Redes neuronales *feedforward*** o perceptrones multicapa
  base del Aprendizaje Profundo.

- Mapeo $\hat{y} = f(\mathbf{x}; \theta)$ formado por capas (imágenes en categorías)

$$f(\mathbf{x}) := f^{(n)}\left(f^{(n-1)}\left(\cdots f^{(2)}\left(f^{(1)}\left(\mathbf{x}\right)\right)\cdots\right)\right),$$

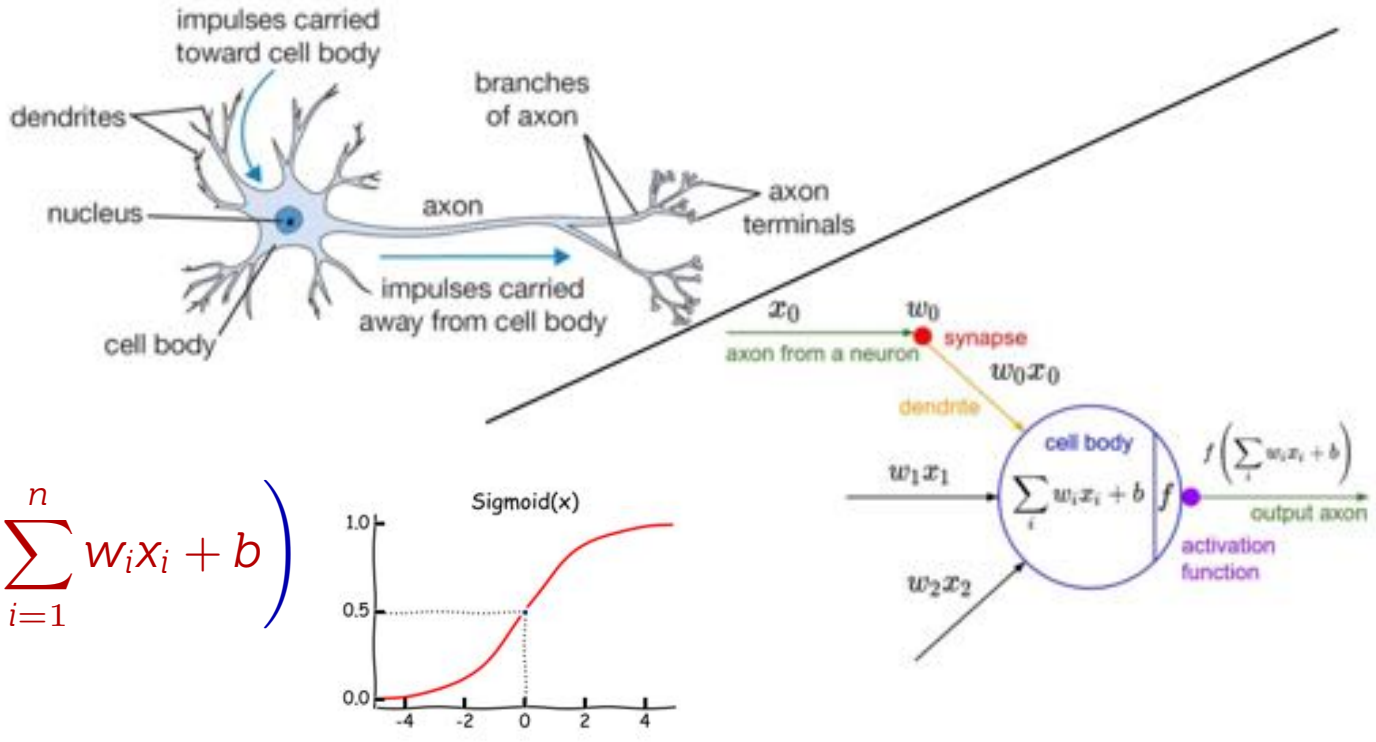
  donde $f^{(1)}$ es la primera capa, $f^{(2)}$ la segunda ...

- Información fluye **sin** existir conexiones de realimentación.



Cada capa está formada por **neuronas**

- Cada elemento de una capa se llama **neurona** o unidad: Dada una entrada vectorial **x** calcula una salida unidimensional $h(\mathbf{x})$.
- Compuesta por: operación lineal + función de activación no lineal.

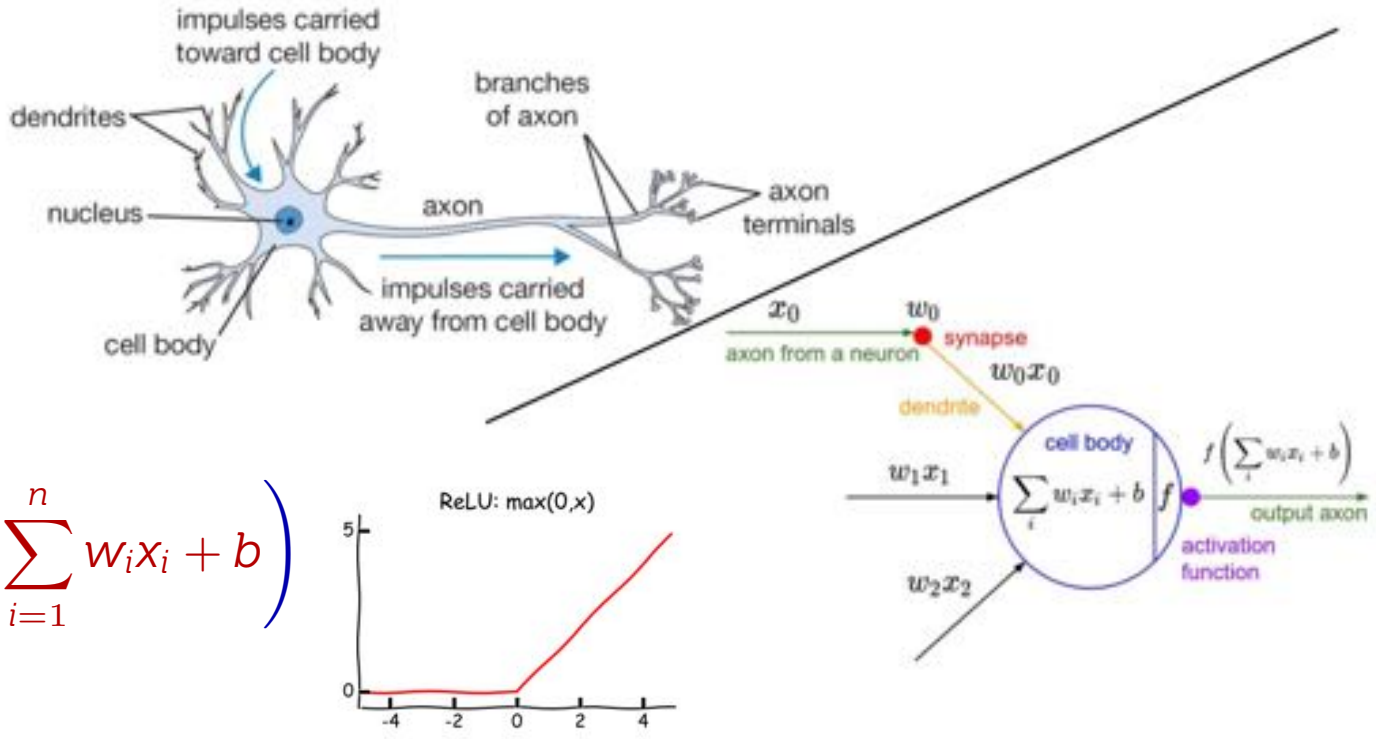


$$h(\mathbf{x}) = g\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

$g(\cdot)$: no-linealidad

Cada capa está formada por **neuronas**

- Cada elemento de una capa se llama **neurona** o unidad: Dada una entrada vectorial **x** calcula una salida unidimensional $h(\mathbf{x})$.
- Compuesta por: operación lineal + función de activación no lineal.



$$h(\mathbf{x}) = g\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

$g(\cdot)$: no-linealidad

Neuronas *artificiales*

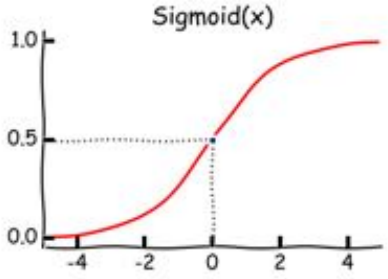
- Una neurona se compone (en general) de: una operación lineal + una función de activación no lineal:

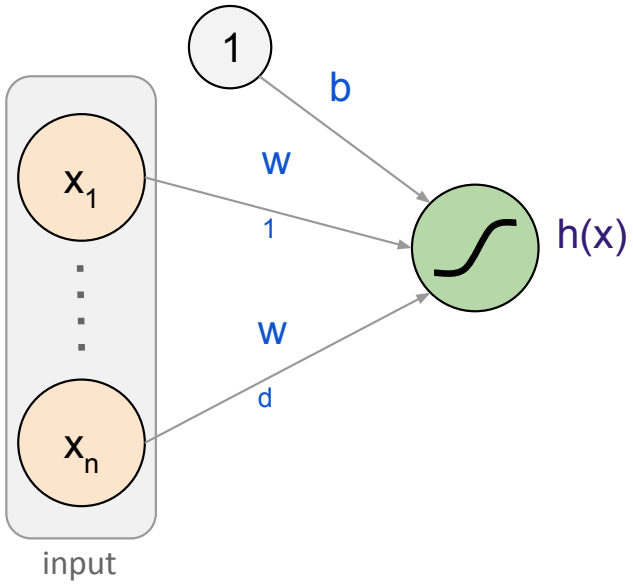
1. Pre-activación:

$$a(\mathbf{x}) = \sum_{i=1}^{n-1} w_i x_i + b = \mathbf{w}^T \mathbf{x} + b$$

2. Activación (salida):

$$h(\mathbf{x}) = g(a(\mathbf{x})) = g\left(\sum_{i=1}^{n-1} w_i x_i + b\right)$$





Neuronas *artificiales*

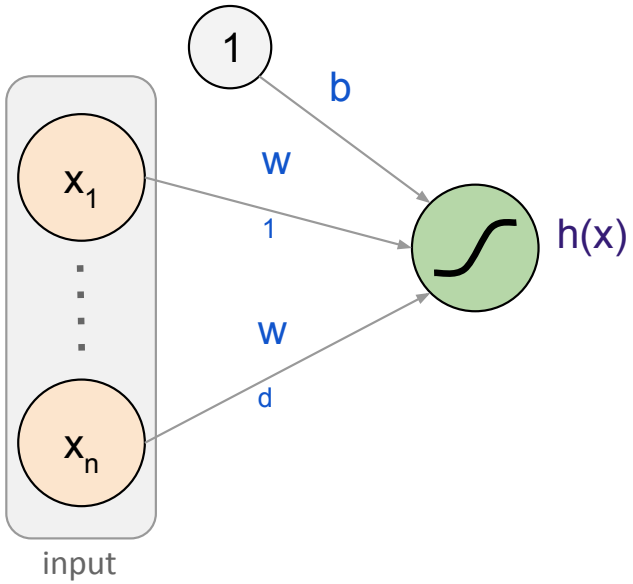
- Una neurona se compone (en general) de: una operación lineal + una función de activación no lineal:

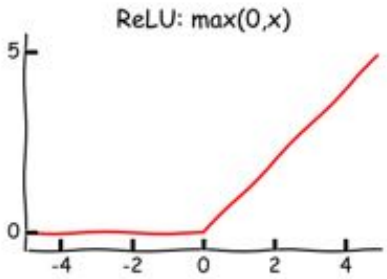
1. Pre-activación:

$$a(\mathbf{x}) = \sum_{i=1}^{n-1} w_i x_i + b = \mathbf{w}^T \mathbf{x} + b$$

2. Activación (salida):

$$h(\mathbf{x}) = g(a(\mathbf{x})) = g\left(\sum_{i=1}^{n-1} w_i x_i + b\right)$$





Red *feedforward* de una capa oculta

- Entrada: $\mathbf{x}$
- Pre-activación:

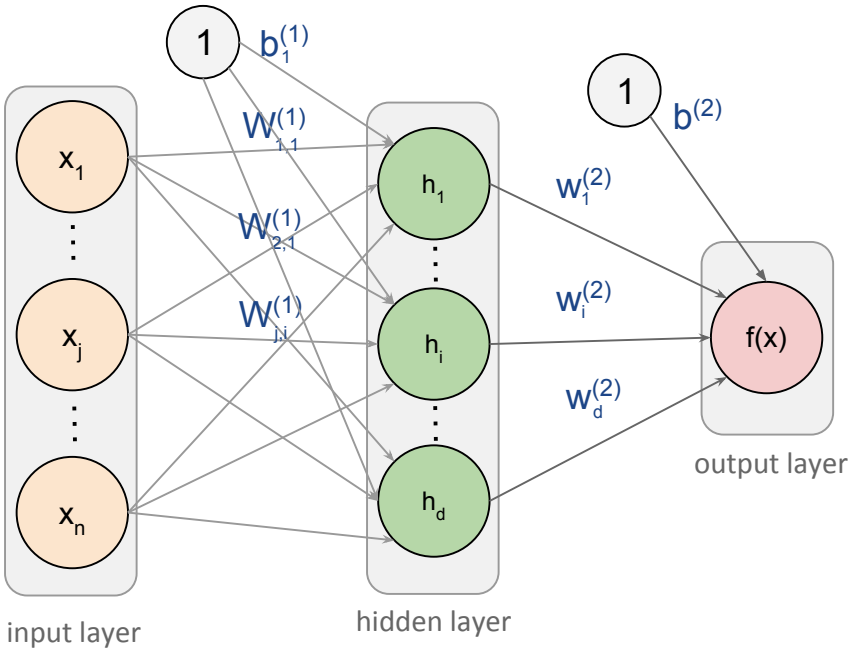
$$\mathbf{a} = \mathbf{W}^{(1)T}\mathbf{x} + \mathbf{b}^{(1)}$$

- Activación capa oculta:

$$\mathbf{h}(\mathbf{x}) = g(\mathbf{a}(\mathbf{x}))$$

- Capa de salida:

$$f(\mathbf{x}) = \mathbf{w}^{(2)T}\mathbf{h}^{(1)} + b^{(2)}$$



Arquitectura de red *deep feedforward*

Red con **L** capas ocultas:

- Entrada: $\mathbf{x}$
- Pre-activación capa (j):

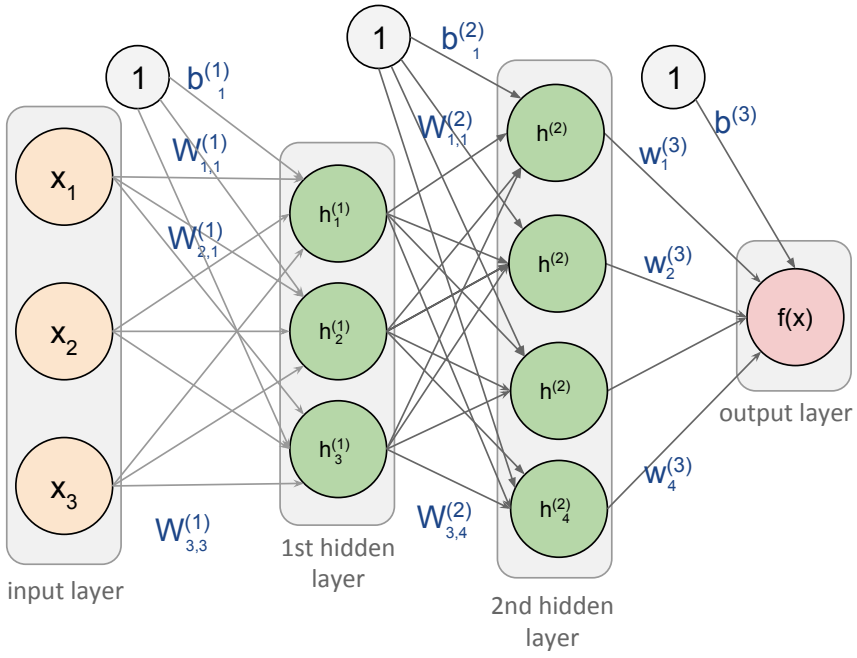
$$\mathbf{a}^{(j)}(\mathbf{x}) = \left(\mathbf{W}^{(j)T}\mathbf{h}^{(j-1)} + \mathbf{b}^{(j)}\right)$$

- Salida capa (j):

$$\mathbf{h}^{(j)}(\mathbf{x}) = g(\mathbf{a}^{(j)}(\mathbf{x}))$$

- Salida de la red (capa $L+1$):

$$f(\mathbf{x}) = \left(\mathbf{w}^{(L+1)T}\mathbf{h}^{(L)} + b^{(L+1)}\right)$$



- Capas totalmente conectadas - **"Fully-connected layers"**

Aprendizaje basado en Optimización

- Definimos arquitectura (capas, num. neuronas, no-linealidad)

Aprendizaje basado en Optimización

- Definimos arquitectura (capas, num. neuronas, no-linealidad)
- Tenemos que encontrar los parámetros de la red (i.e., θ)

Aprendizaje basado en Optimización

- Definimos arquitectura (capas, num. neuronas, no-linealidad)
- Tenemos que encontrar los parámetros de la red (i.e., θ)
- Dado un conjunto de datos de entrenamiento $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_1, y_1)\}$, se formula como una optimización, donde se busca minimizar un costo de ajuste L (*loss*)

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} L_i(f(\mathbf{x}_i; \theta), y_i)$$

Aprendizaje basado en Optimización

- Definimos arquitectura (capas, num. neuronas, no-linealidad)
- Tenemos que encontrar los parámetros de la red (i.e., θ)
- Dado un conjunto de datos de entrenamiento $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_1, y_1)\}$, se formula como una optimización, donde se busca minimizar un costo de ajuste L (*loss*)

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} L_i(f(\mathbf{x}_i; \theta), y_i)$$

- Se resuelve mediante descenso por gradiente (estocástico)

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t)$$

- Algoritmo de *Backpropagation* para calcular gradiente

Descenso por gradiente

Autor: desconocido

Teorema de Aproximación Universal

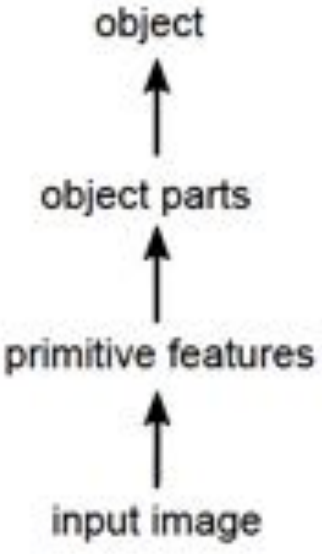
Teorema (Cybenko, 1989; Hornik 1991)

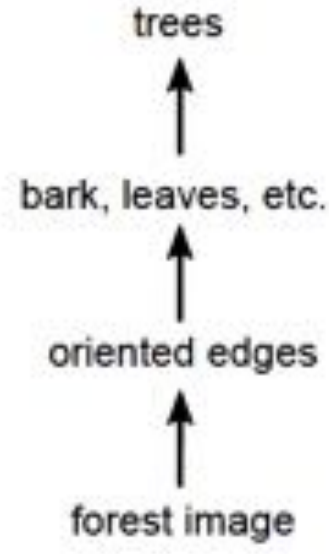
"Una red neuronal prealimentada con una única capa oculta y un número finito de neuronas, puede aproximar cualquier función continua en un espacio compacto de $\mathbb{R}^n$*".*

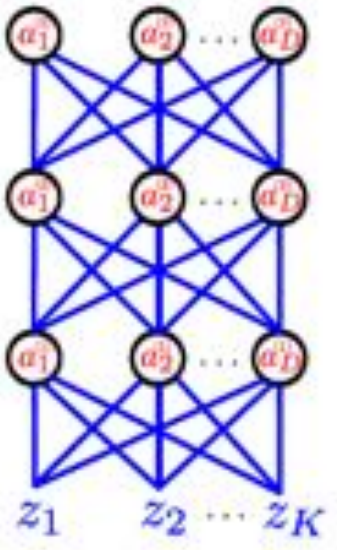
- Con los paramétros adecuados, se puede representar una gran variedad de funciones
- El teorema no habla de cómo aprender los parámetros
- George Cybenko en 1989 para función de activación sigmoide
- Kurt Hornik lo extiende en 1991, a funciones generales, lo importante es la arquitectura feedfoward no la función de activación

¿Por qué usar redes profundas?

Los datos (en general) tienen una organización jerárquica

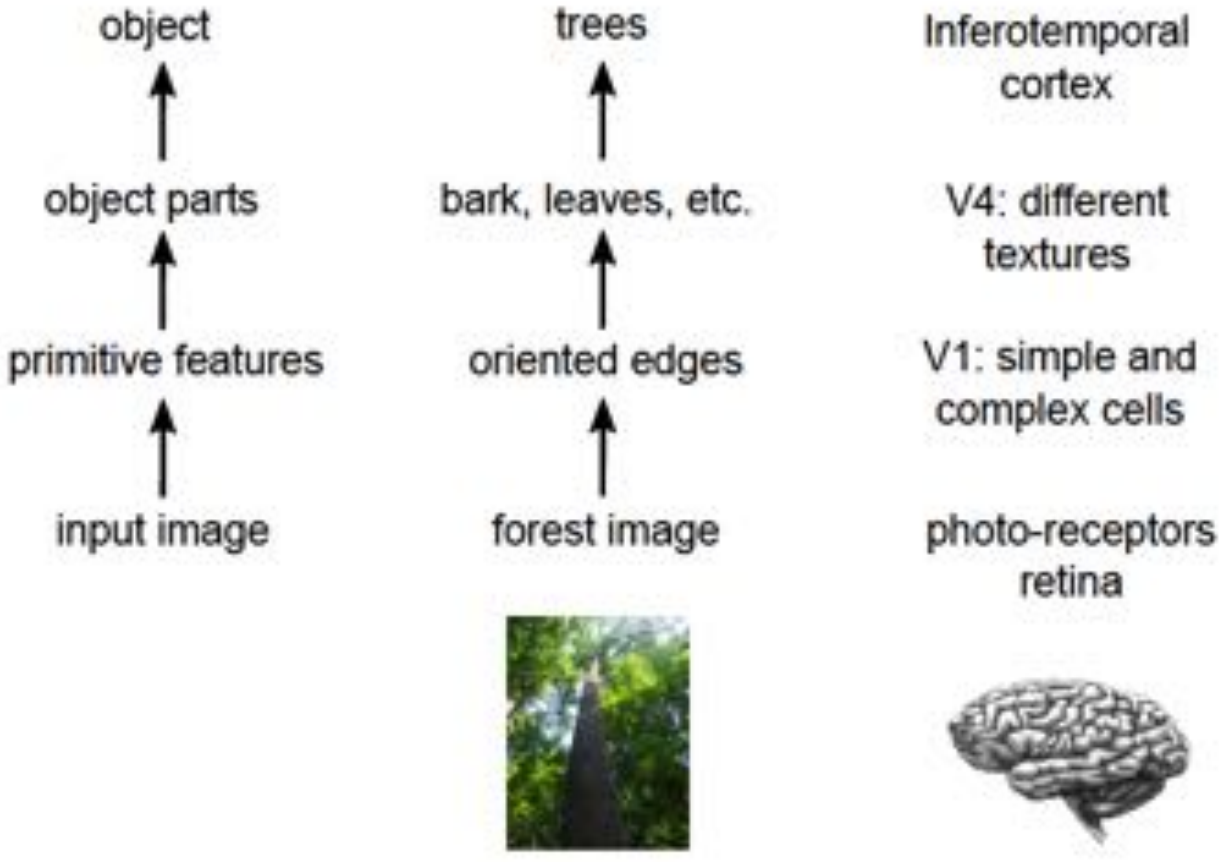






¿Por qué usar redes profundas?

Nuestra visión tiene una organización jerárquica



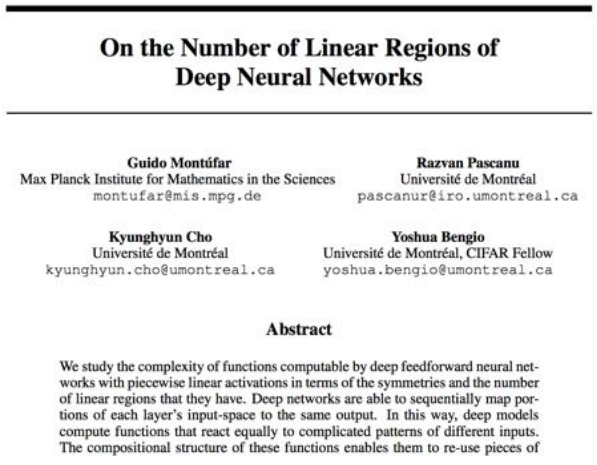
¿Por qué usar redes profundas?

Redes poco profundas ineficientes para representar funciones complejas

Montufar et al. [2014], *"Red neuronal (ReLu), con con d entradas, L capas, n unidades por capa oculta, puede calcular funciones con:*

$$O\left(\binom{n}{d}^{d(L-1)} n^d\right),$$

regiones lineales."



On the Number of Linear Regions of Deep Neural Networks

Guido Montúfar
Max Planck Institute for Mathematics in the Sciences
montufar@mis.mpg.de

Razvan Pascanu
Université de Montréal
pascanur@iro.umontreal.ca

Kyunghyun Cho
Université de Montréal
kyunghyun.cho@umontreal.ca

Yoshua Bengio
Université de Montréal, CIFAR Fellow
yoshua.bengio@umontreal.ca

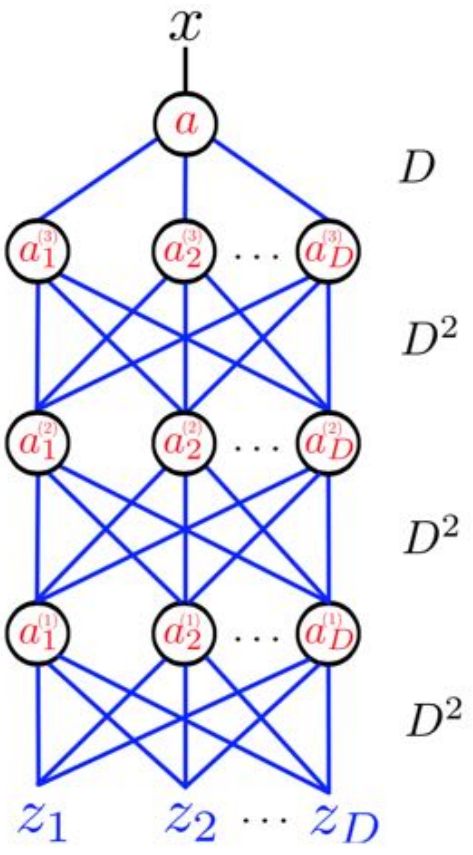
Abstract

We study the complexity of functions computable by deep feedforward neural networks with piecewise linear activations in terms of the symmetries and the number of linear regions that they have. Deep networks are able to sequentially map portions of each layer's input-space to the same output. In this way, deep models compute functions that react equally to complicated patterns of different inputs. The compositional structure of these functions enables them to re-use pieces of

- Número de regiones crece de manera exponencial con profundidad *L* y polinomial con *n*, (más rápido que red de una capa con *nL* neuronas).

¿Cuál es el problema de las redes *fully connected*?

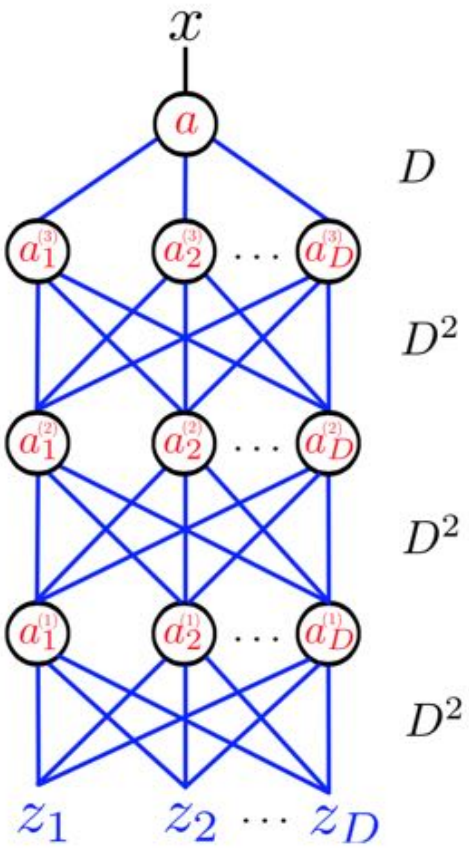
- ¿Cuántos parámetros tiene esta red?



Slides adaptadas de Richard E. Turner

¿Cuál es el problema de las redes *fully connected*?

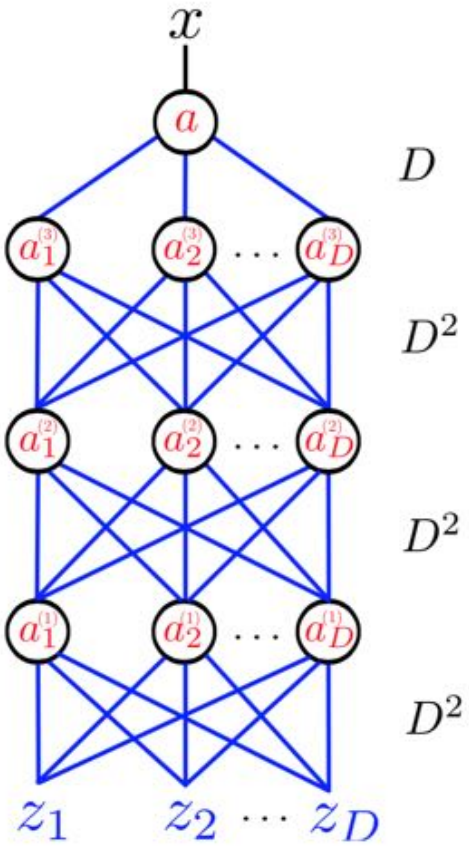
- ¿Cuántos parámetros tiene esta red?
  - $3D^2 + D$



Slides adaptadas de Richard E. Turner

¿Cuál es el problema de las redes *fully connected*?

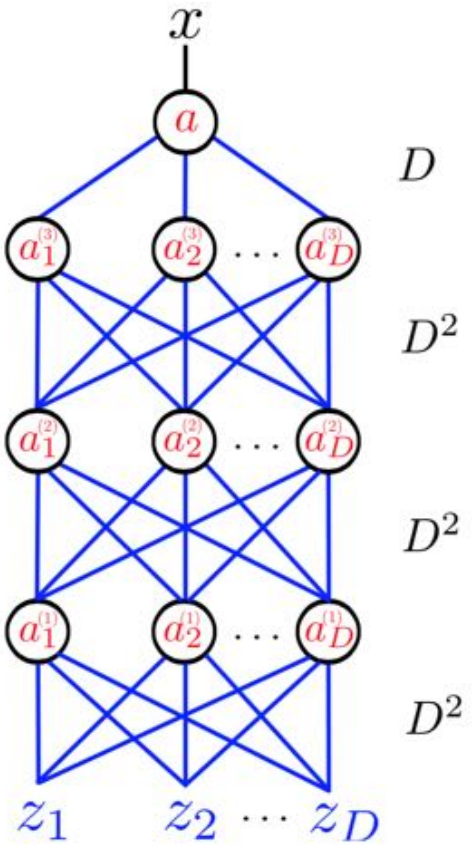
- ¿Cuántos parámetros tiene esta red?
  - $3D^2 + D$
- Si tenemos una imagen pequeña 32x32



Slides adaptadas de Richard E. Turner

¿Cuál es el problema de las redes *fully connected*?

- ¿Cuántos parámetros tiene esta red?
  - $3D^2 + D$
- Si tenemos una imagen pequeña 32x32
  - $3 \times (32^2)^2 + 32^2 \approx 3 \times 10^6$

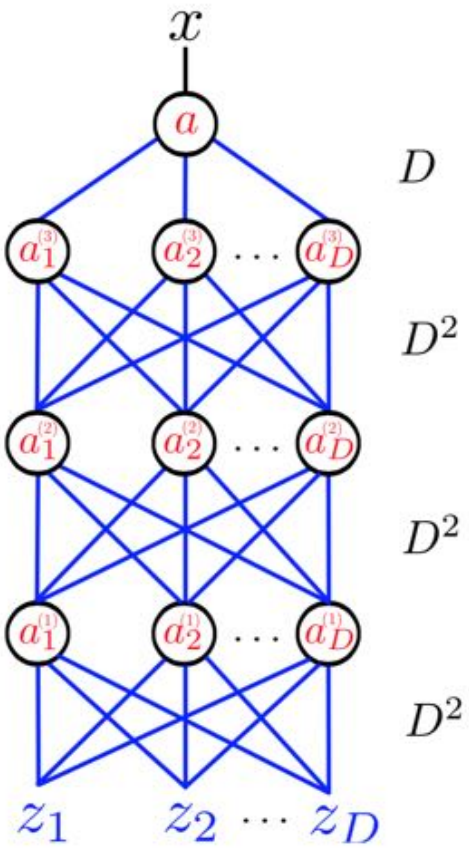


Slides adaptadas de Richard E. Turner

¿Cuál es el problema de las redes *fully connected*?

- ¿Cuántos parámetros tiene esta red?
  - $3D^2 + D$
- Si tenemos una imagen pequeña 32x32
  - $3 \times (32^2)^2 + 32^2 \approx 3 \times 10^6$

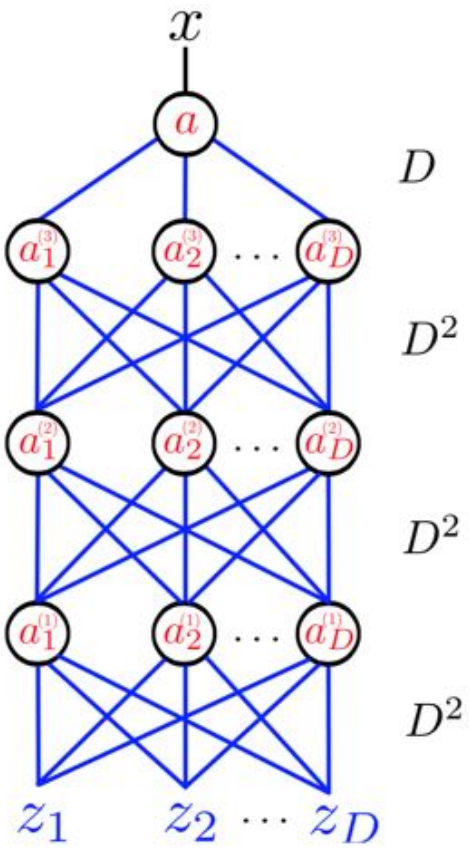
- **Difícil de entrenar**: sobreajuste, inicialización



Slides adaptadas de Richard E. Turner

¿Cuál es el problema de las redes *fully connected*?

- ¿Cuántos parámetros tiene esta red?
  - $3D^2 + D$
- Si tenemos una imagen pequeña 32x32
  - $3 \times (32^2)^2 + 32^2 \approx 3 \times 10^6$



- **Difícil de entrenar**: sobreajuste, inicialización
- **Redes de Convolución:** permiten disminuir número de parámetros forzando invarianzas

Slides adaptadas de Richard E. Turner

Hubel y Wiesel - 1961

Base neurológica de la percepción visual

106

J. Physiol. (1962), **160**, *pp.* 106–154
With 2 plates and 20 text-figures
Printed in Great Britain

RECEPTIVE FIELDS, BINOCULAR INTERACTION AND FUNCTIONAL ARCHITECTURE IN THE CAT'S VISUAL CORTEX

BY D. H. HUBEL AND T. N. WIESEL
From the Neurophysiology Laboratory, Department of Pharmacology Harvard Medical School, Boston, Massachusetts, U.S.A.

(*Received* 31 *July* 1961)

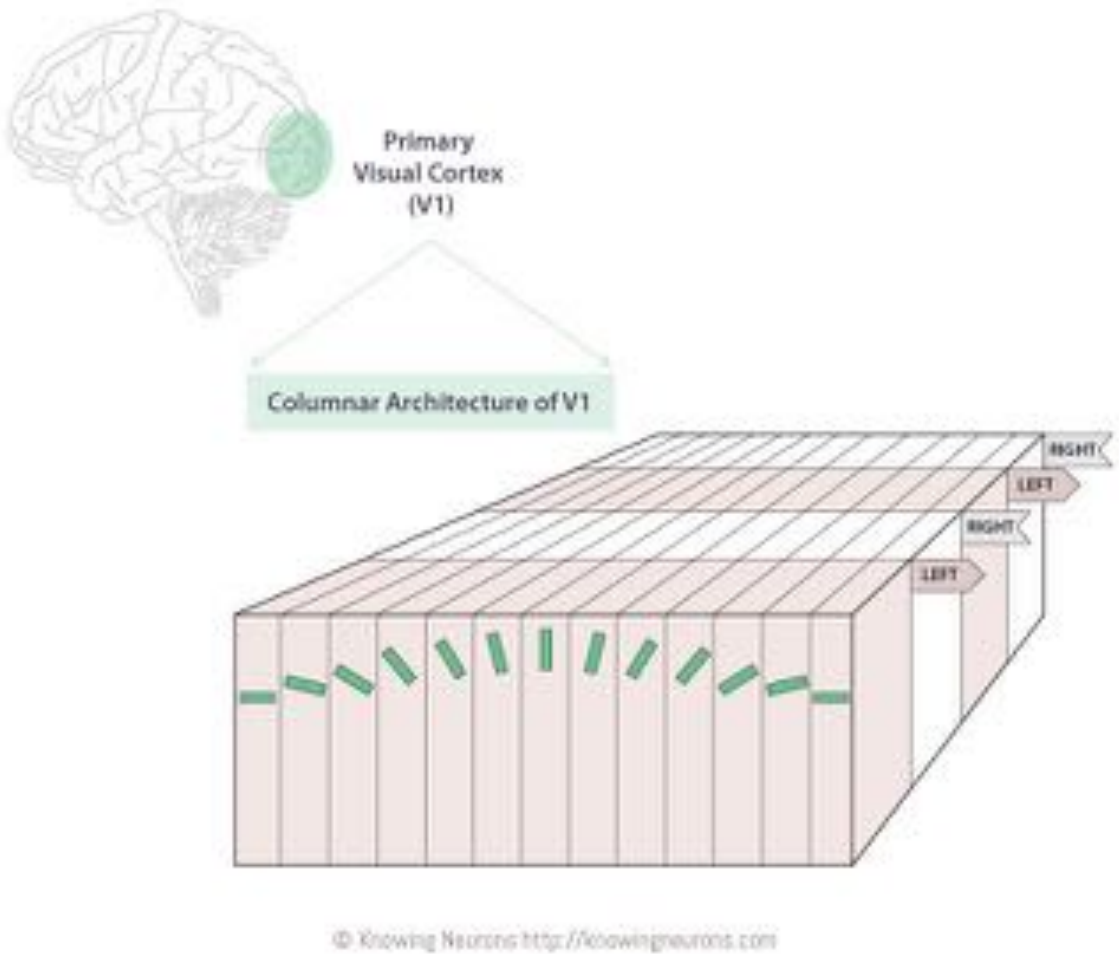
What chiefly distinguishes cerebral cortex from other parts of the central nervous system is the great diversity of its cell types and interconnexions. It would be astonishing if such a structure did not profoundly modify the response patterns of fibres coming into it. In the cat's visual cortex, the receptive field arrangements of single cells suggest that there is indeed a degree of complexity far exceeding anything yet seen at lower levels in the visual system.

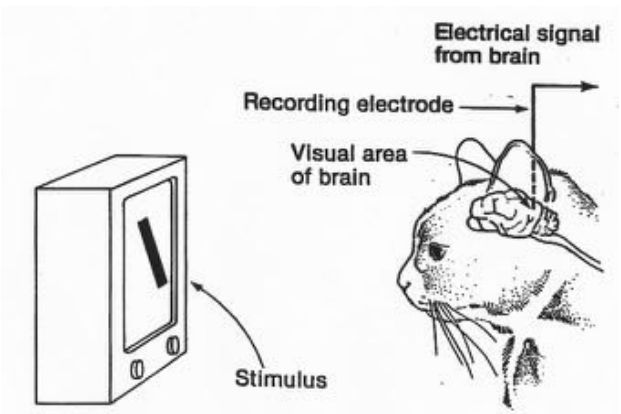
Hubel, David H., and Torsten N. Wiesel. The Journal of physiology, 1962
"Receptive fields, binocular interaction and functional architecture in the cat's visual cortex."

Hubel y Wiesel - 1961

Base neurológica de la percepción visual

El experimento del gato





Hubel, David H., and Torsten N. Wiesel. The Journal of physiology, 1962
"Receptive fields, binocular interaction and functional architecture in the cat's visual cortex."

Hubel y Wiesel - 1961

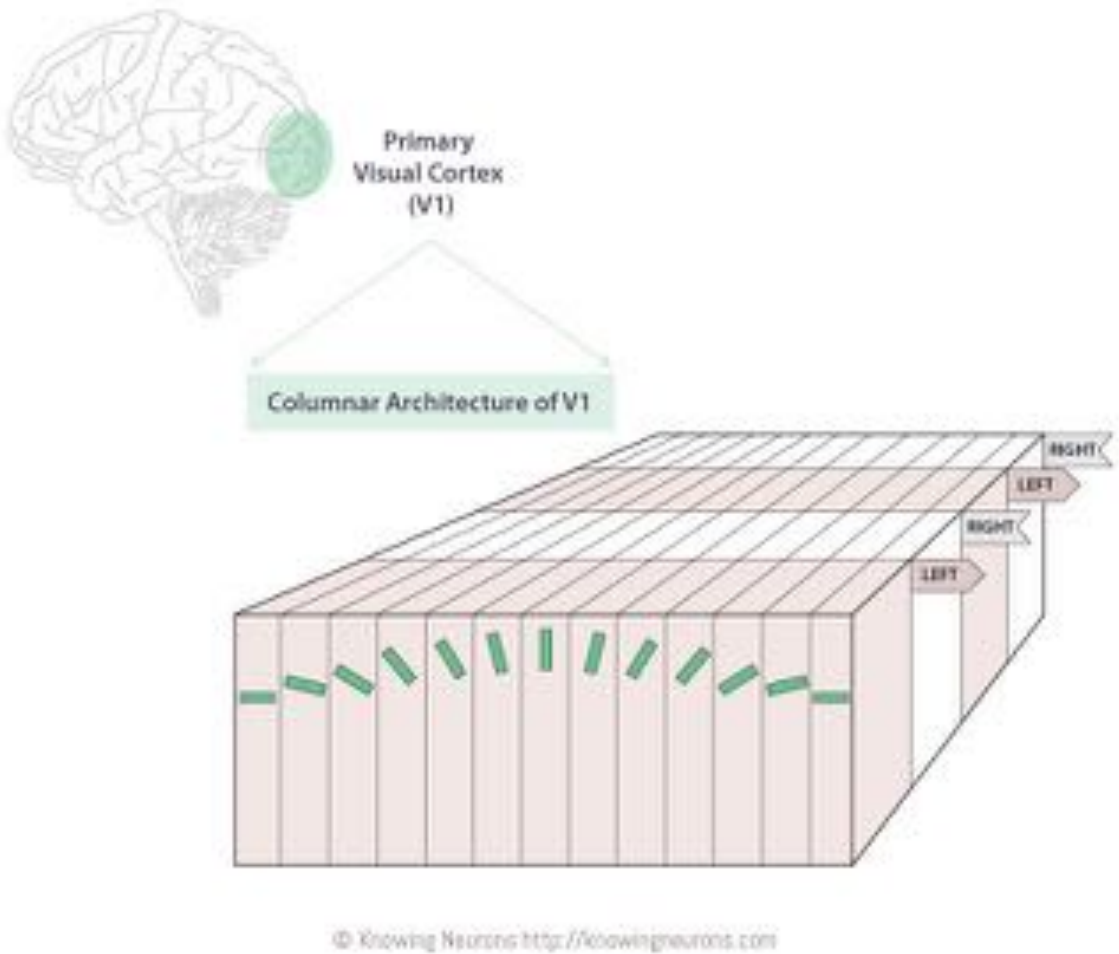
Base neurológica de la percepción visual

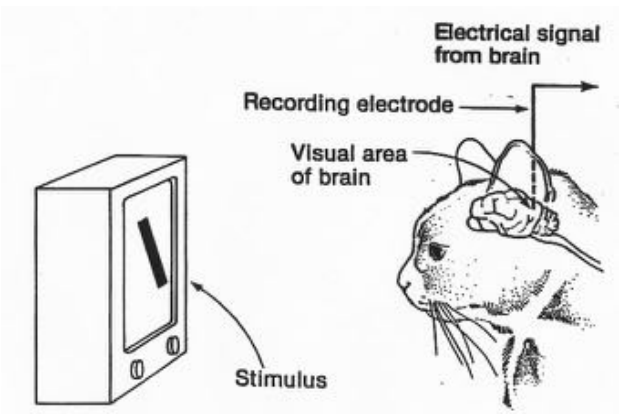
Hubel, David H., and Torsten N. Wiesel. The Journal of physiology, 1962
"Receptive fields, binocular interaction and functional architecture in the cat's visual cortex."

Hubel y Wiesel - 1961

Base neurológica de la percepción visual

El experimento del gato





Hubel, David H., and Torsten N. Wiesel. The Journal of physiology, 1962
"Receptive fields, binocular interaction and functional architecture in the cat's visual cortex."

Redes neuronales de convolución (CNN)

Problemas de visión son muy difíciles: se requieren invarianzas a distintas transformaciones (punto de vista, iluminación,..)

Dos grandes caminos:

1. **Aprender** las invarianzas a partir de un (enorme) conjunto de un entrenamiento (*let the data talk*)
2. **Construir** las invarianzas imponiendo un modelo en la representación

Francesco Peri, Hidden Cat - Penny

Christina Gandolfo, Cat in the box

Matteo, hiding

Grahford, Hidden Cat

Redes Convolucionales: Motivación

1. Estadísticas en imágenes son invariantes a traslaciones
   - Imponer invarianza a traslación en el modelo (en lugar de aprenderla)
   - Baja el número de parámetros: Se comparten pesos

Redes Convolucionales: Motivación

1. Estadísticas en imágenes son invariantes a traslaciones
   - Imponer invarianza a traslación en el modelo (en lugar de aprenderla)
   - Baja el número de parámetros: Se comparten pesos

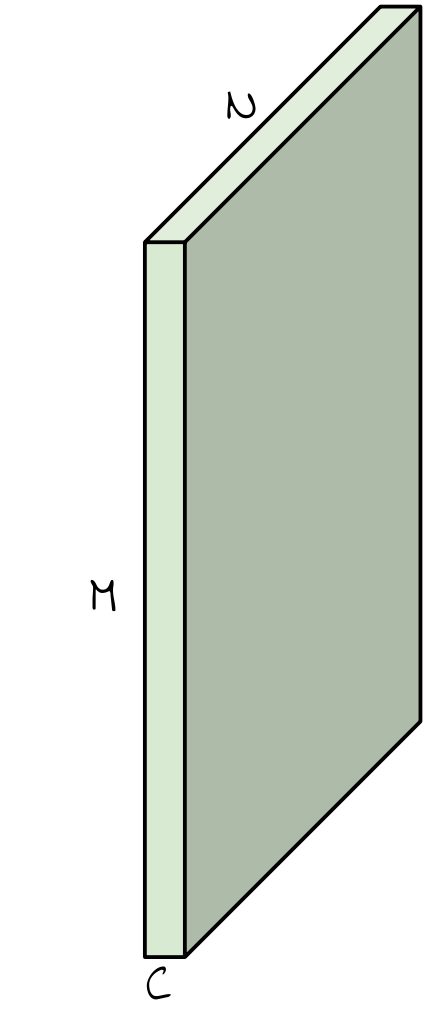
2. Características de bajo nivel son locales (detector de blobs, bordes)
   - Imponer localidad en el modelo: conectividad local (soporte del filtro)
   - Baja el número de parámetros: Núcleos pequeños

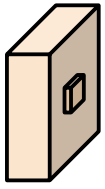
Redes Convolucionales: Motivación

1. Estadísticas en imágenes son invariantes a traslaciones
   - Imponer invarianza a traslación en el modelo (en lugar de aprenderla)
   - Baja el número de parámetros: Se comparten pesos
2. Características de bajo nivel son locales (detector de blobs, bordes)
   - Imponer localidad en el modelo: conectividad local (soporte del filtro)
   - Baja el número de parámetros: Núcleos pequeños
3. Se espera que características de alto nivel sean gruesas (biología)
   - Se puede submuestrear a medida que aumenta la profundidad en la red
   - Baja (aún más) el número de parámetros

Capa de Convolución

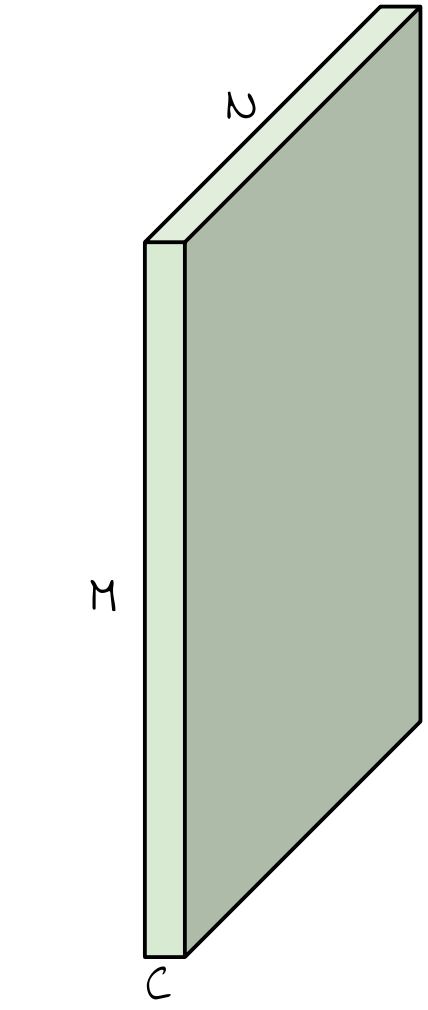
Capa de Convolución





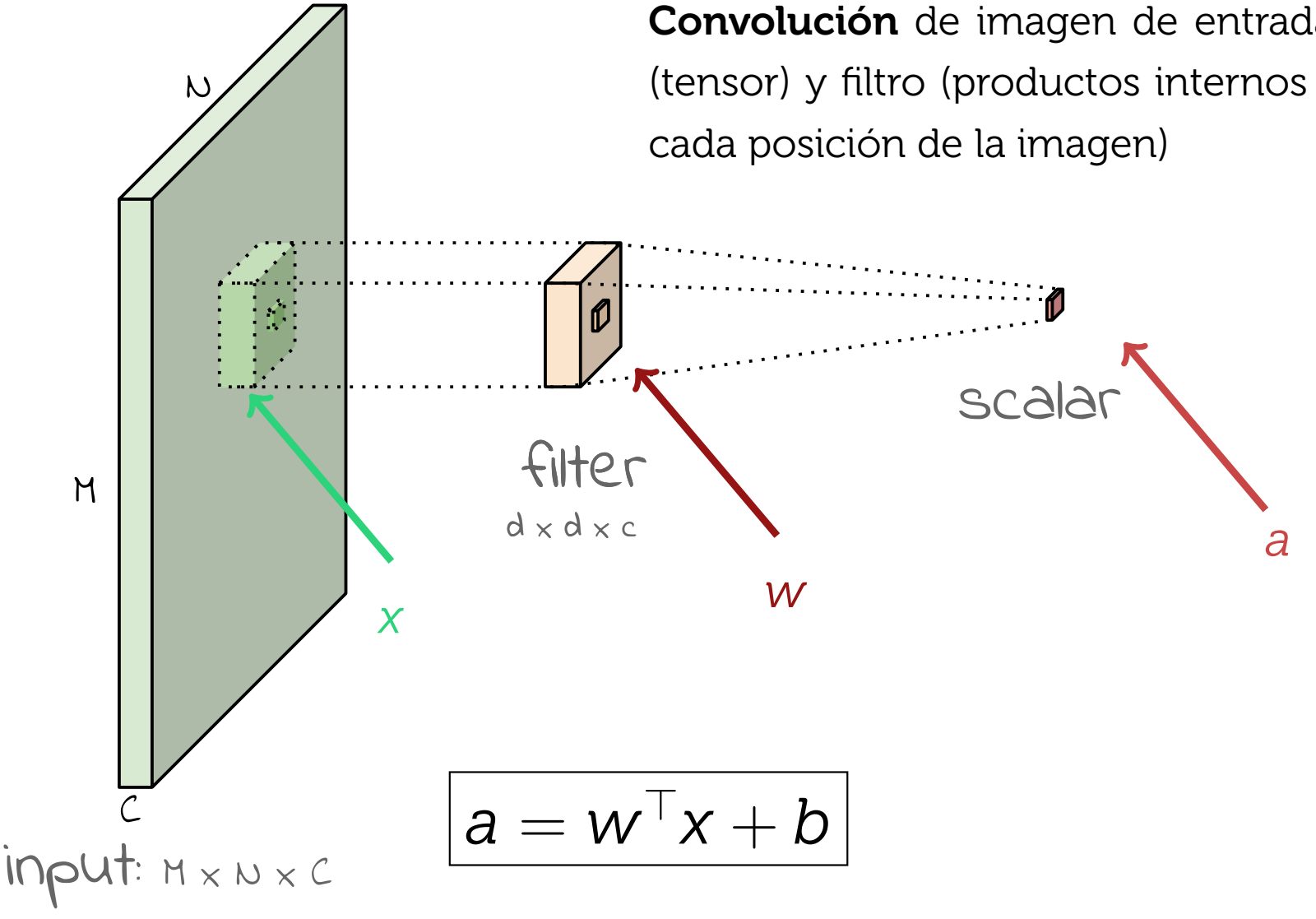
Convolución de imagen de entrada (tensor) y filtro (productos internos en cada posición de la imagen)

Capa de Convolución



Convolución de imagen de entrada (tensor) y filtro (productos internos en cada posición de la imagen)

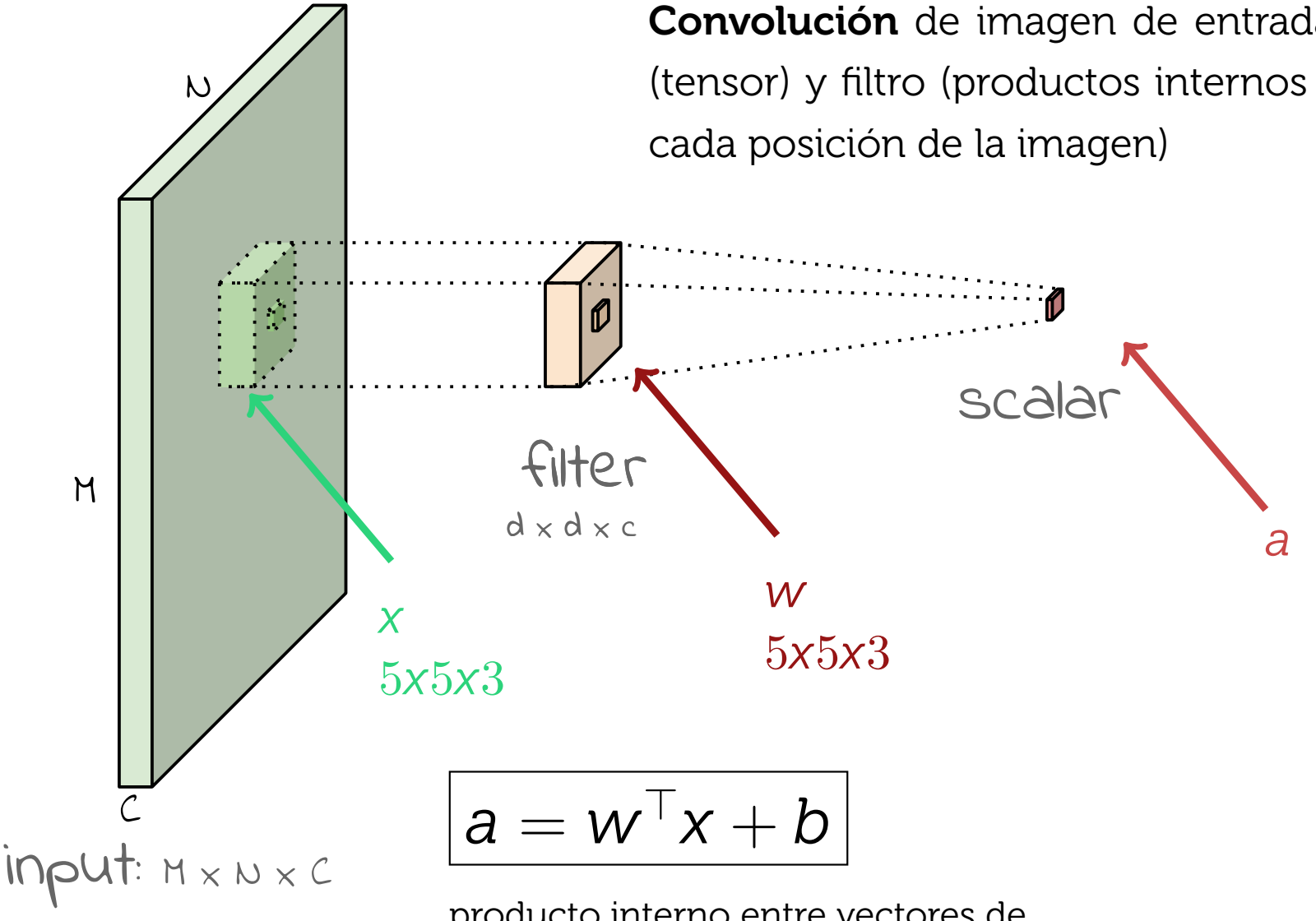
Capa de Convolución



Convolución de imagen de entrada (tensor) y filtro (productos internos en cada posición de la imagen)

$$a = w^\top x + b$$

Capa de Convolución

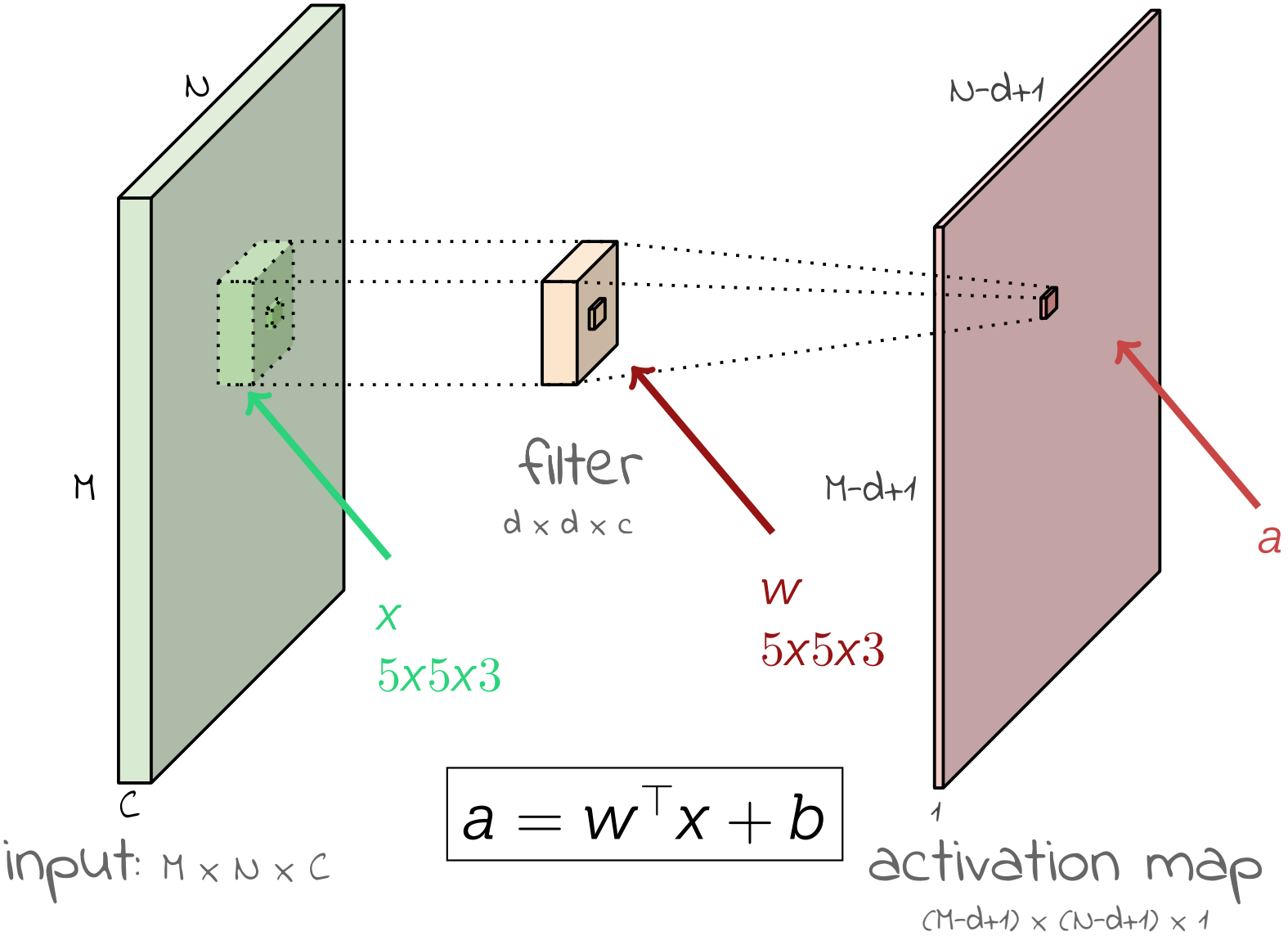


Convolución de imagen de entrada (tensor) y filtro (productos internos en cada posición de la imagen)

$$a = w^\top x + b$$

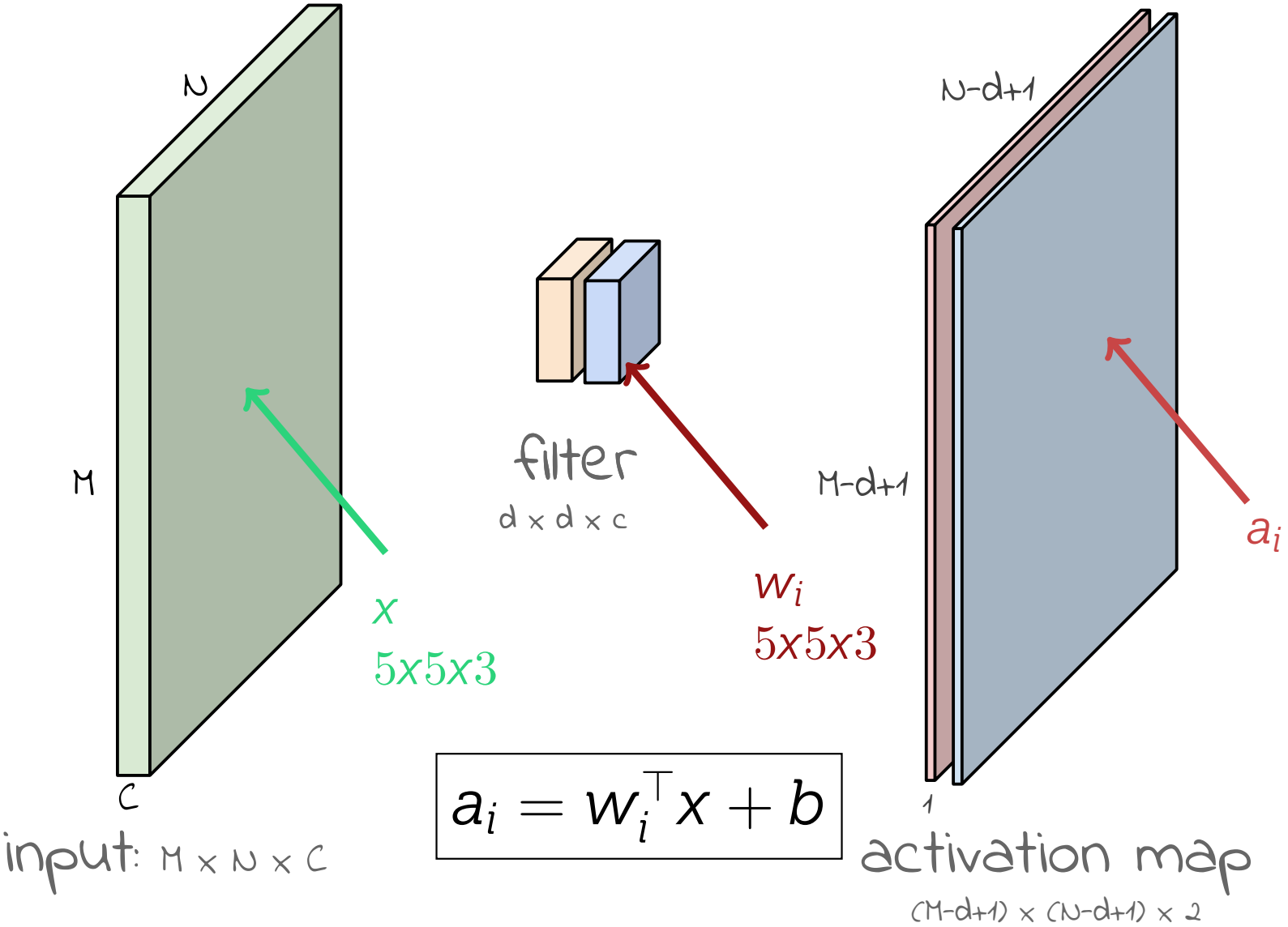
producto interno entre vectores de dimensión 75 $(5 \times 5 \times 3$ $)$+ bias $\rightarrow$ **escalar**

Capa de Convolución



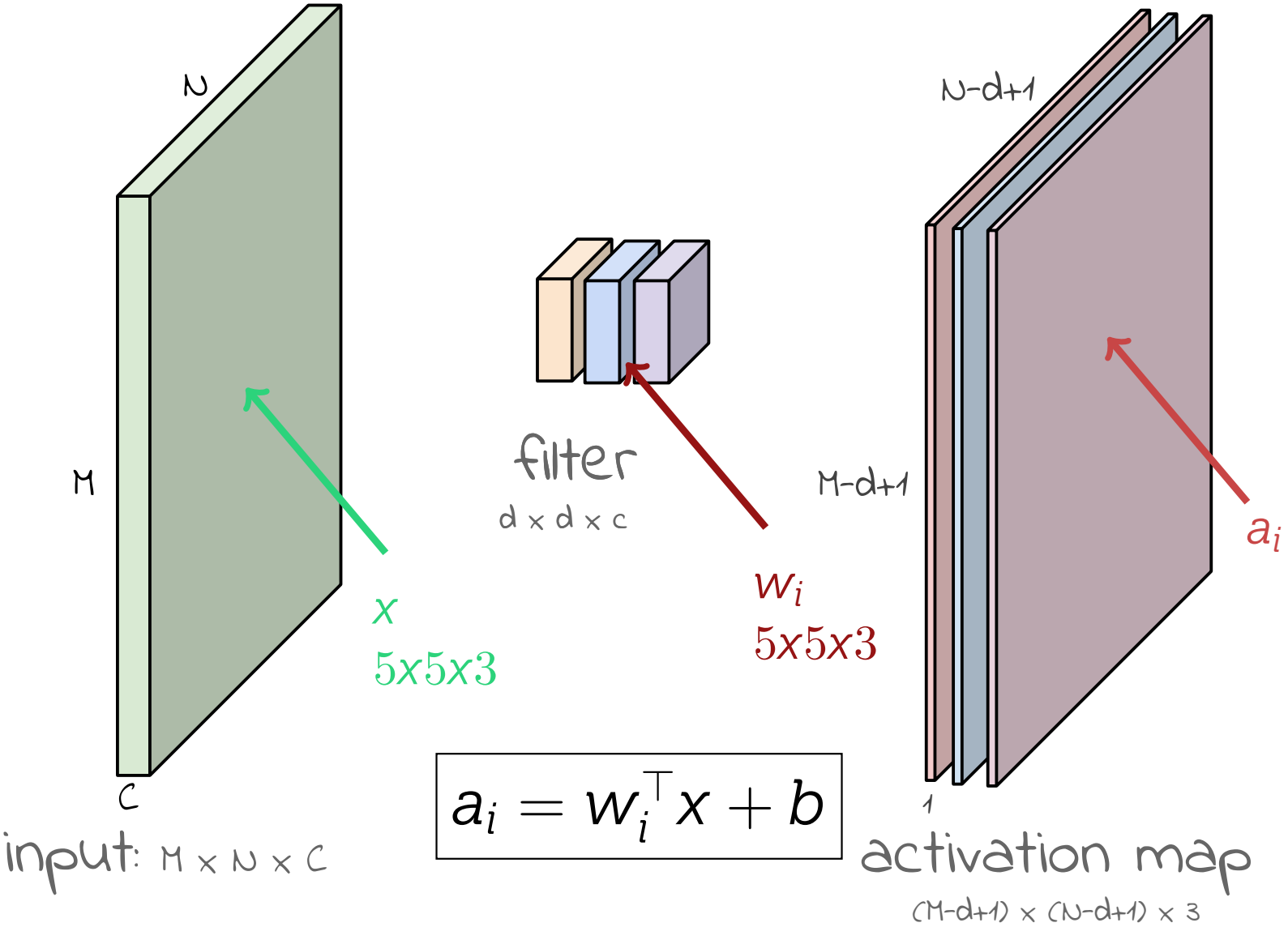
$$a = w^\top x + b$$

Capa de Convolución



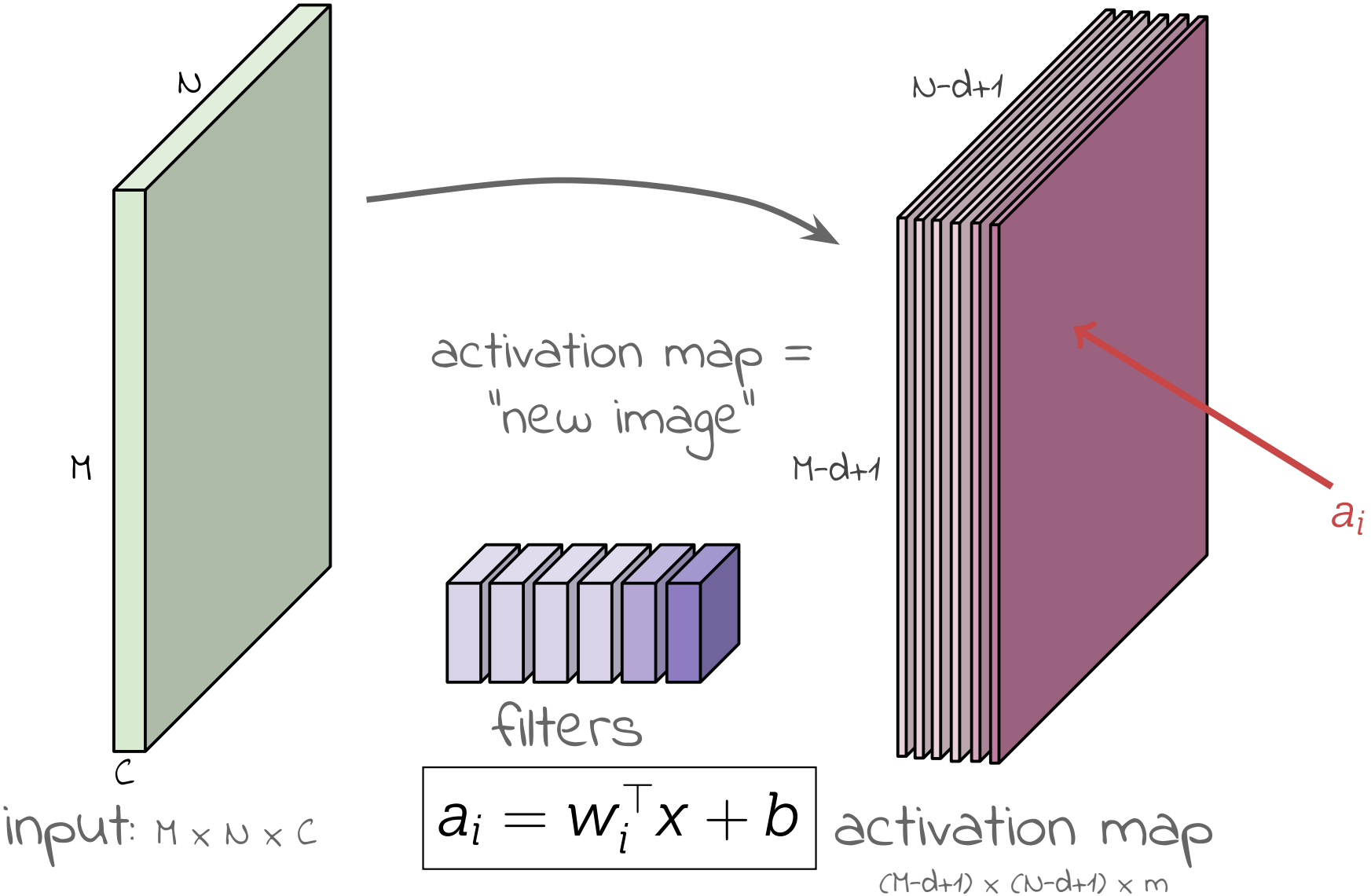
$$a_i = w_i^\top x + b$$

Capa de Convolución

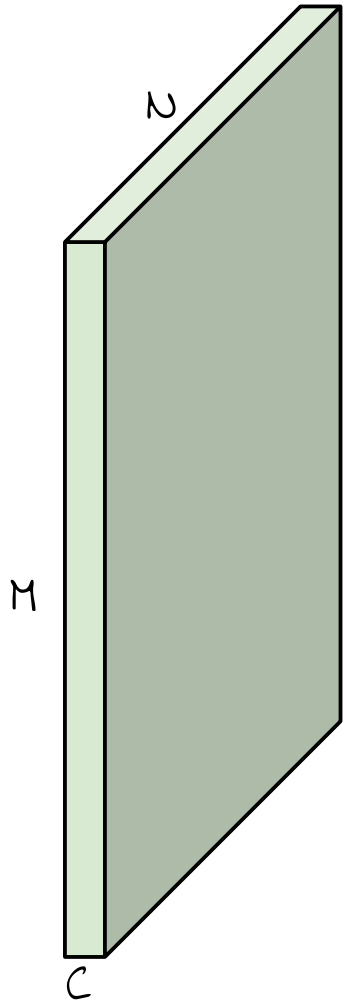


$$a_i = w_i^\top x + b$$

Capa de Convolución

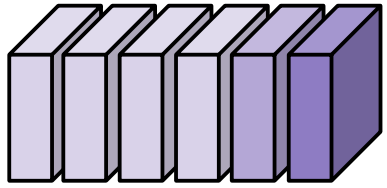


Capa de Convolución



input: $M \times N \times C$

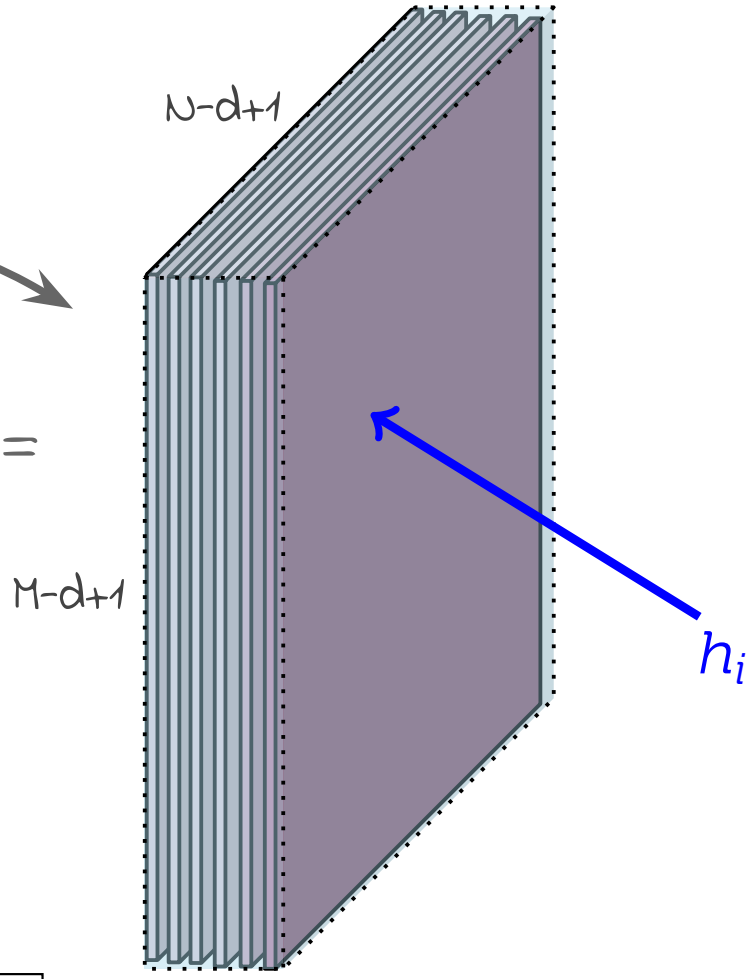
activation map = "new image"



filters

$$a_i = w_i^\top x + b$$

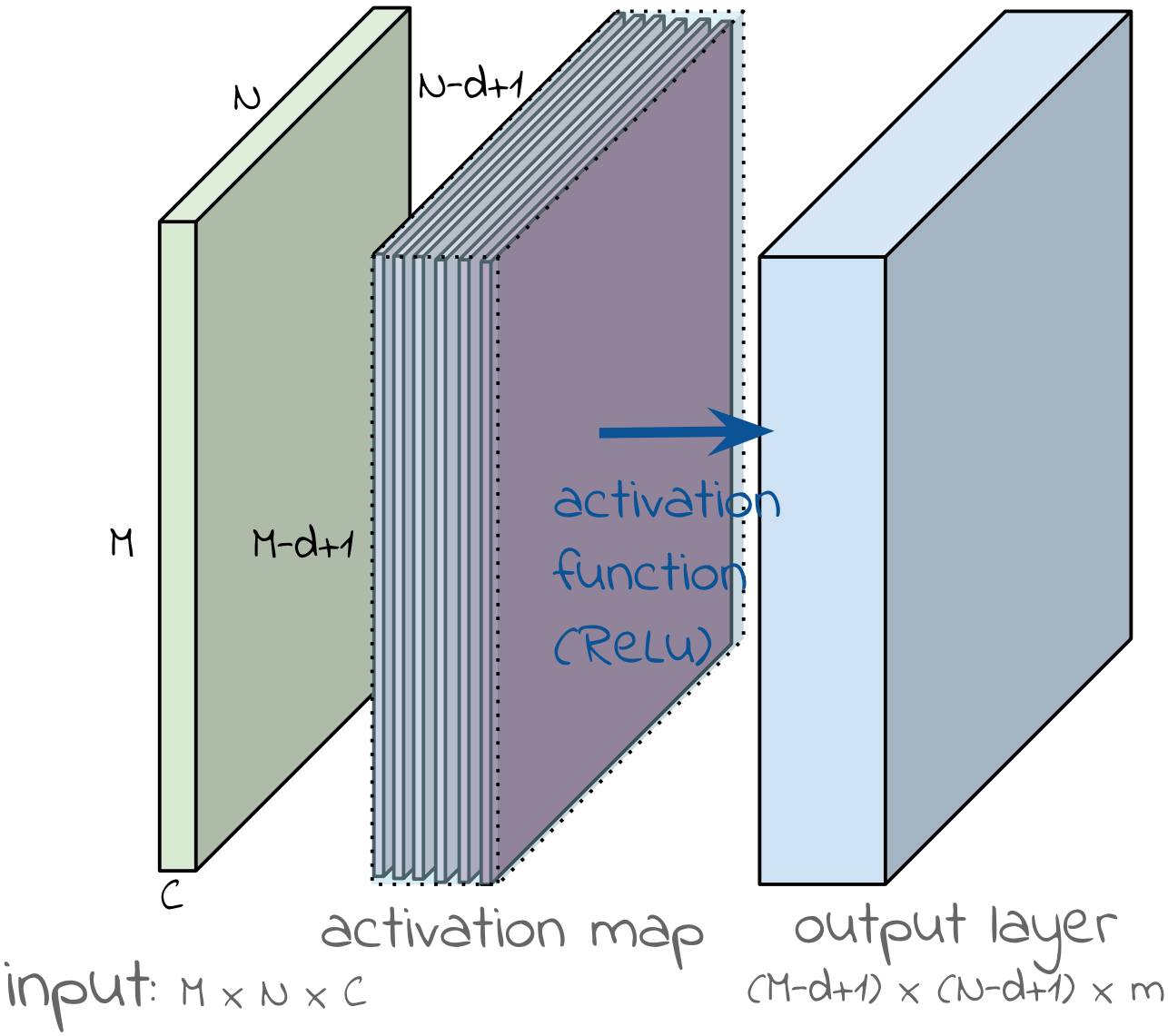
$$h_i = g(a_i)$$



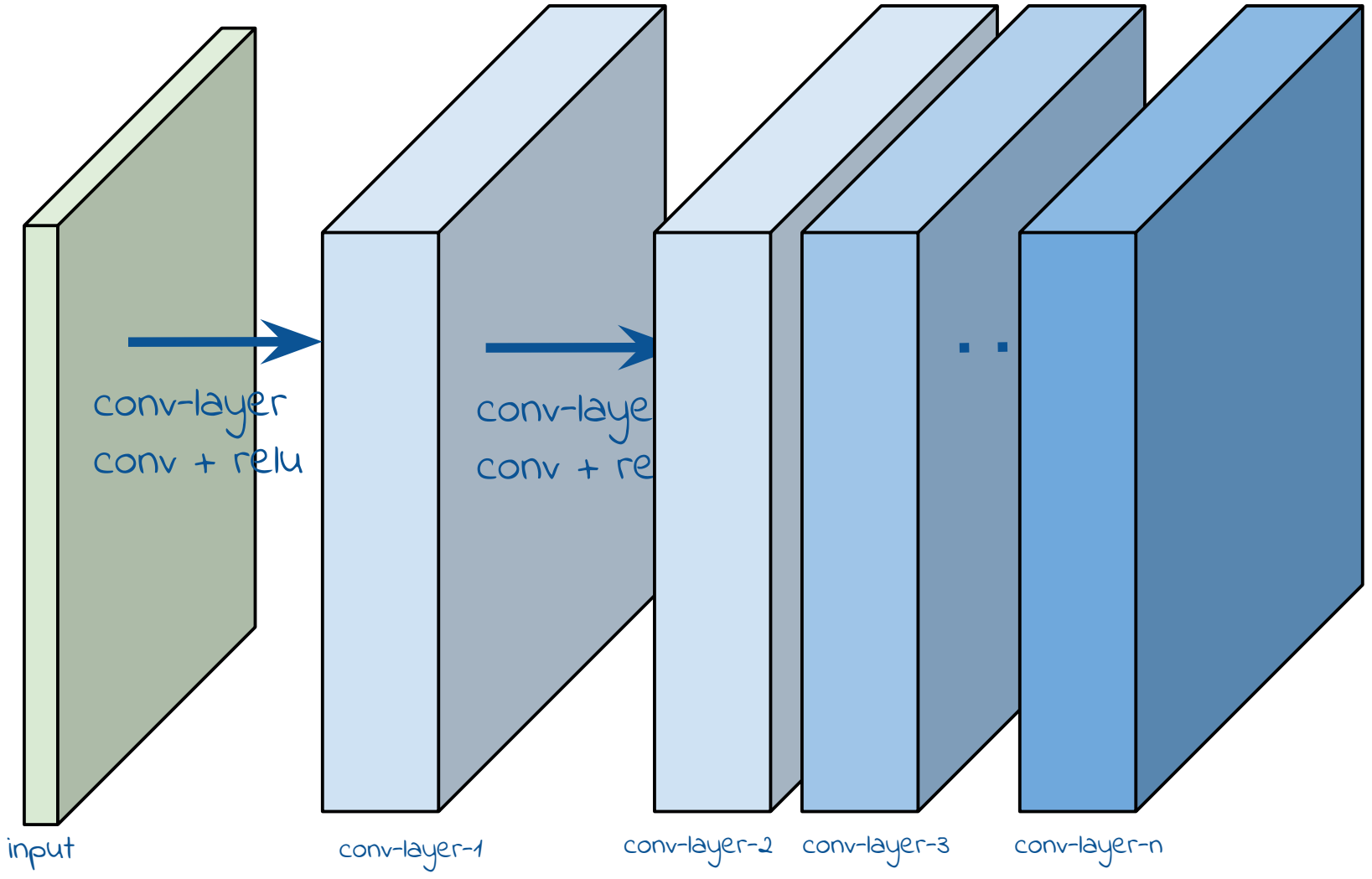
activation map

$(M-d+1) \times (N-d+1) \times m$

Capa de Convolución: Convolución + Activación



Capa de Convolución: Convolución + Activación

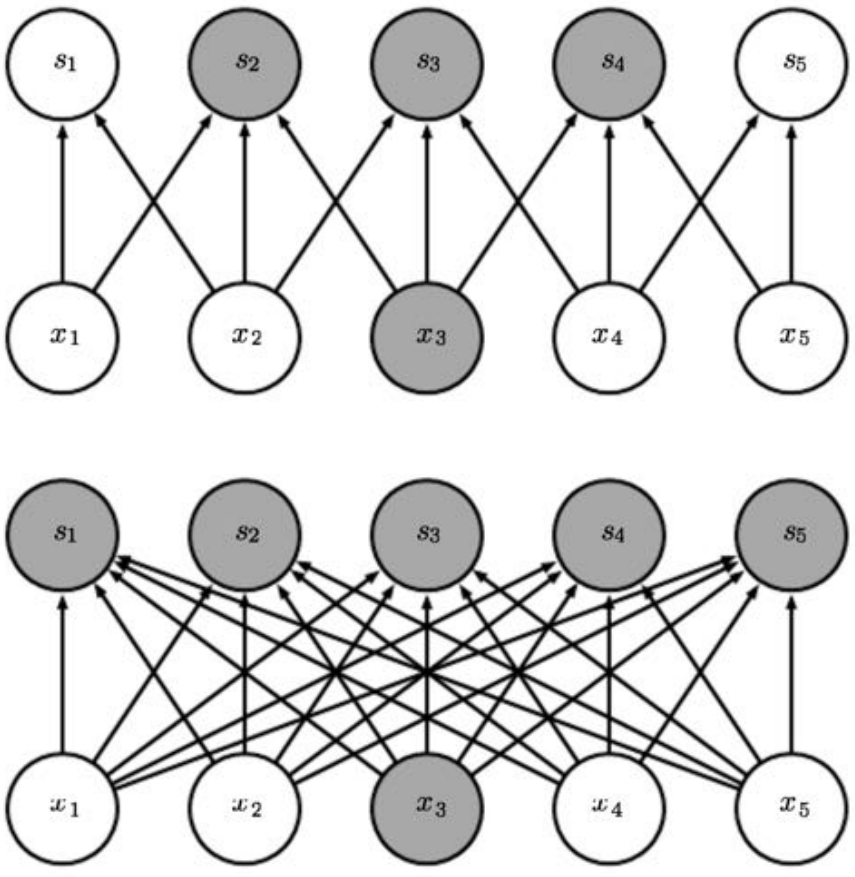


Capas de Convolución: Observaciones

- **Representaciones:** Capas de convolución permiten manejar datos de diferente tamaño (e.g., imágenes), sin necesidad de cambiar la red
- **Pesos compartidos:** Pesos de las capas de convolución (filtros) se reutilizan en varios elementos de la entrada.
- **Conexiones locales:** La conectividad de un elemento en la salida está dada por el soporte del filtro (pequeño).
- **Equivarianza:** Si trasladamos la entrada, la salida se trasladada.

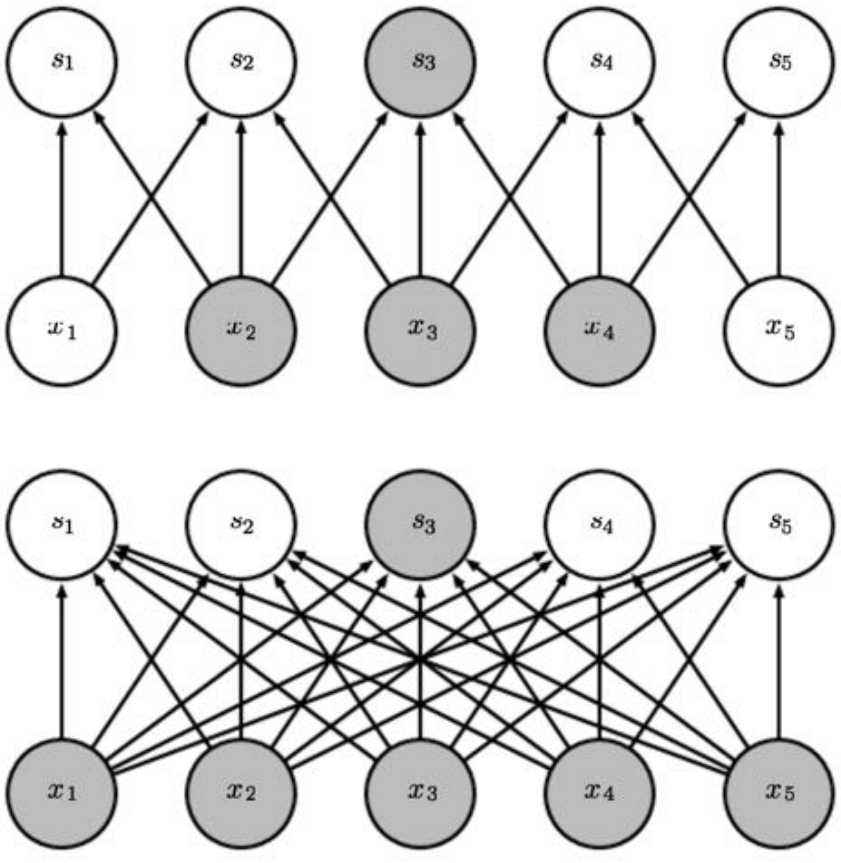
Capa de Convolución: Observaciones

Campo Receptivo (*receptive field*): Varias capas de convolución con filtros pequeños $\rightarrow$ Aumenta el campo receptivo).



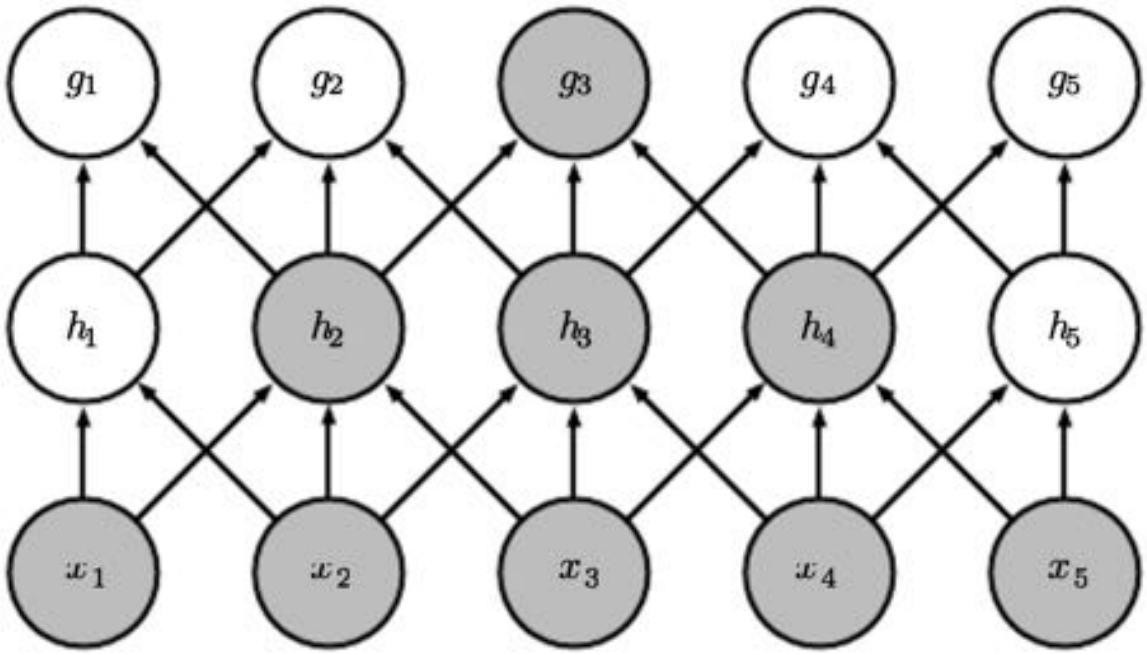
Capa de Convolución: Observaciones

Campo Receptivo (*receptive field*): Varias capas de convolución con filtros pequeños $\rightarrow$ Aumenta el campo receptivo).



Capa de Convolución: Observaciones

Campo Receptivo (*receptive field*): Varias capas de convolución con filtros pequeños $\rightarrow$ Aumenta el campo receptivo).

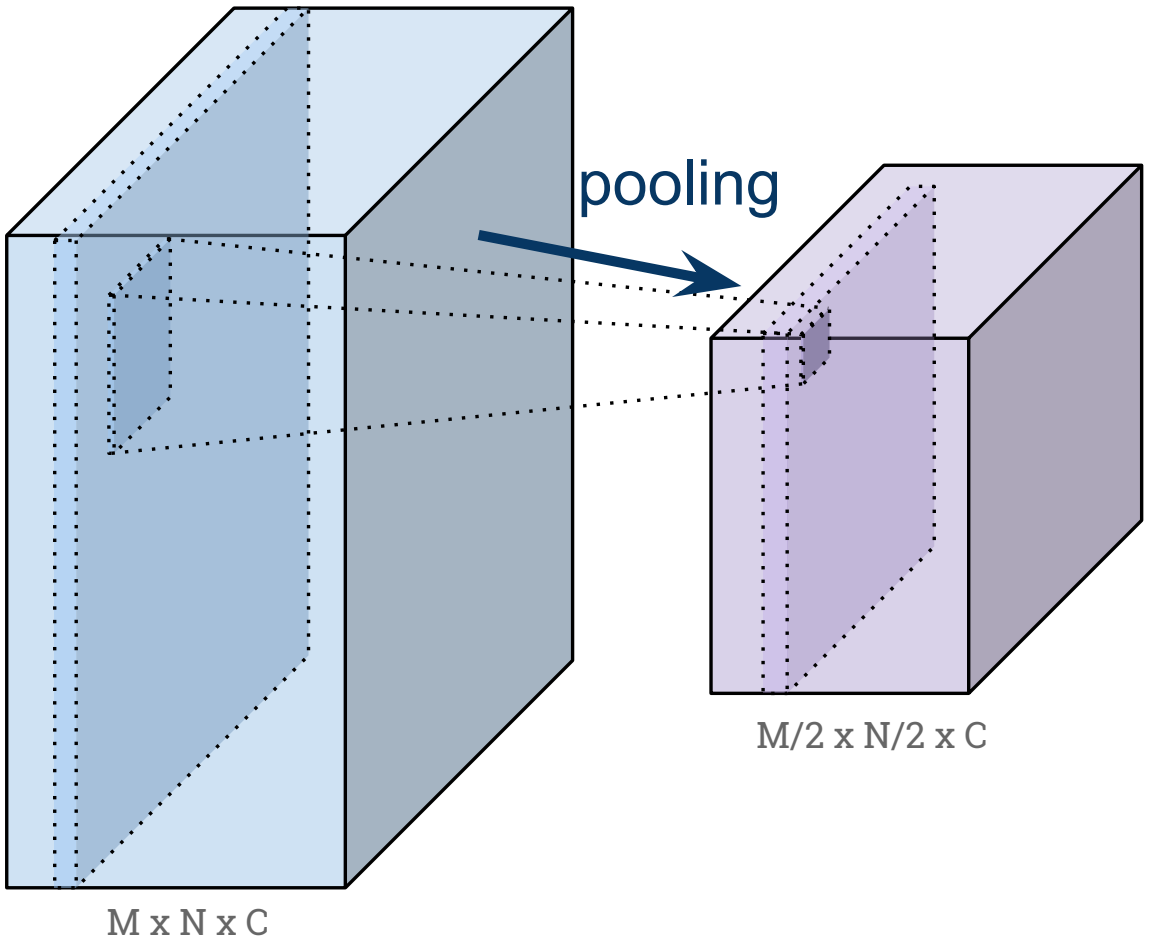


campo receptivo

Capa de *Pooling*

Capa de *Pooling*

- Comprime (sub-muestreo) de la representación
- Opera en cada mapa de activación (canal) por separado



Capa de *Pooling*

- Comprime (sub-muestreo) de la representación
- Opera en cada mapa de activación (canal) por separado

Capa de *Pooling*

- Comprime (sub-muestreo) de la representación
- Opera en cada mapa de activación (canal) por separado

Lenet-5 - 1998

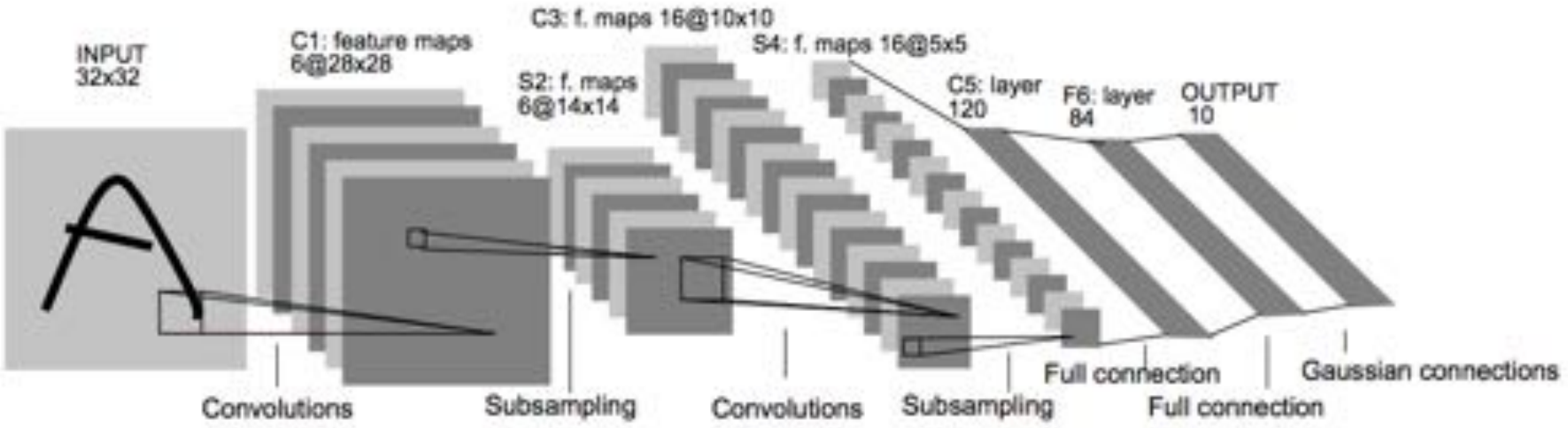


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.
"Gradient-based learning applied to document recognition." Proc. IEEE (1998).

AlexNet - 2012

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

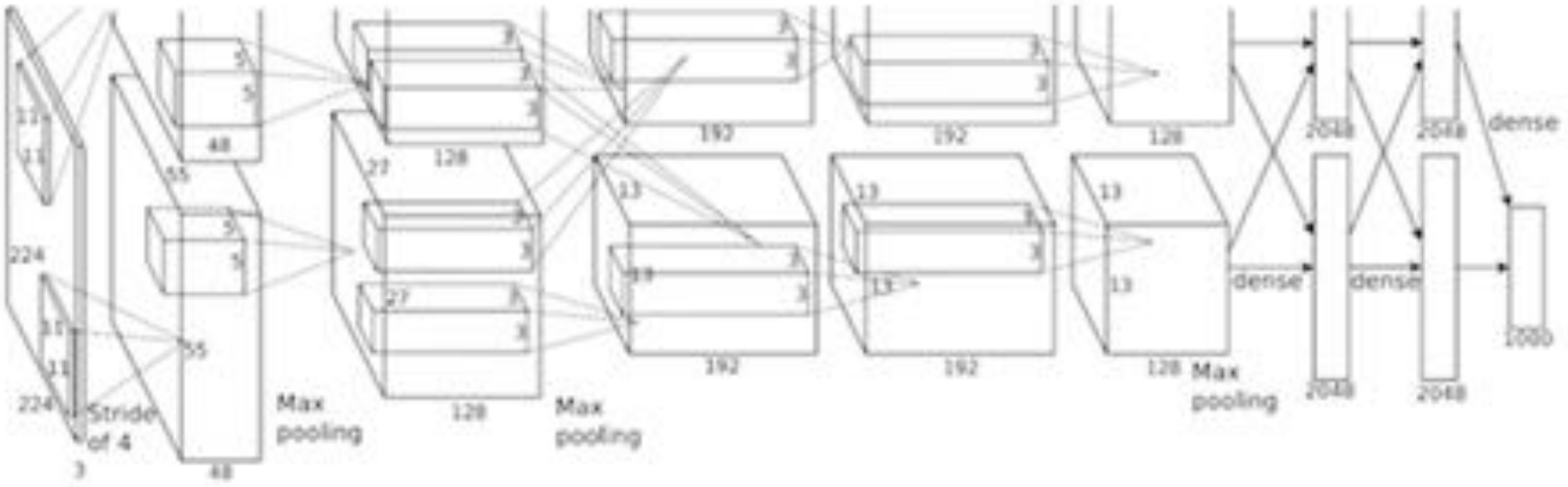
Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

Krizhevsky, Alex, Ilya Sutskever, Geoffrey E. Hinton.
"Imagenet classification with deep convolutional neural networks.", NIPS 2012. (13k citas)

AlexNet - 2012

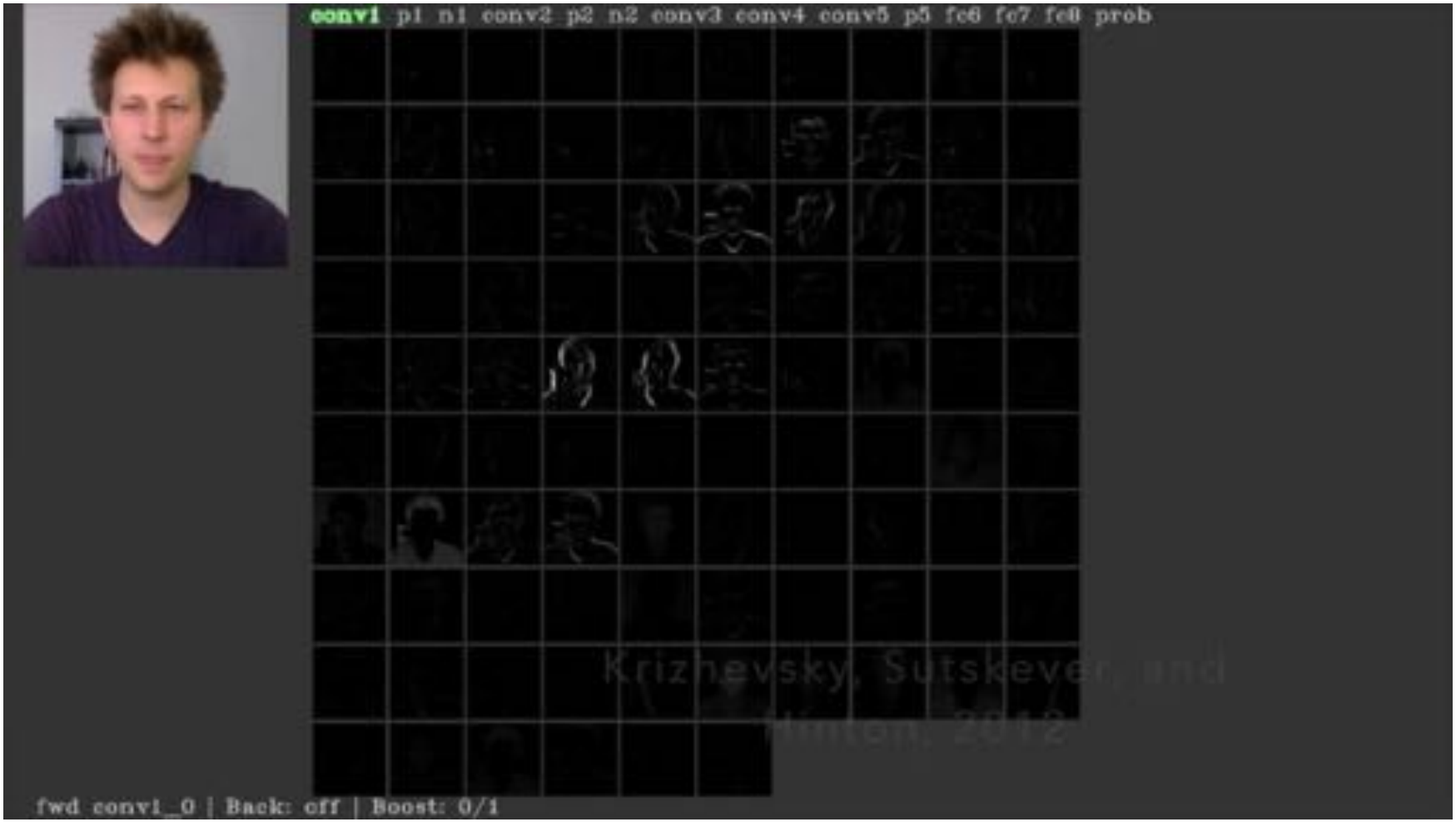


AlexNet architecture (May look weird because there are two different "streams". This is because the training process was so computationally expensive that they had to split the training onto 2 GPUs)

Krizhevsky, Alex, Ilya Sutskever, Geoffrey E. Hinton.
"Imagenet classification with deep convolutional neural networks.", NIPS 2012. (13k citas)

Capas de Convolución - Mapas de Activación

Primera capa: Información de bajo nivel (bordes)

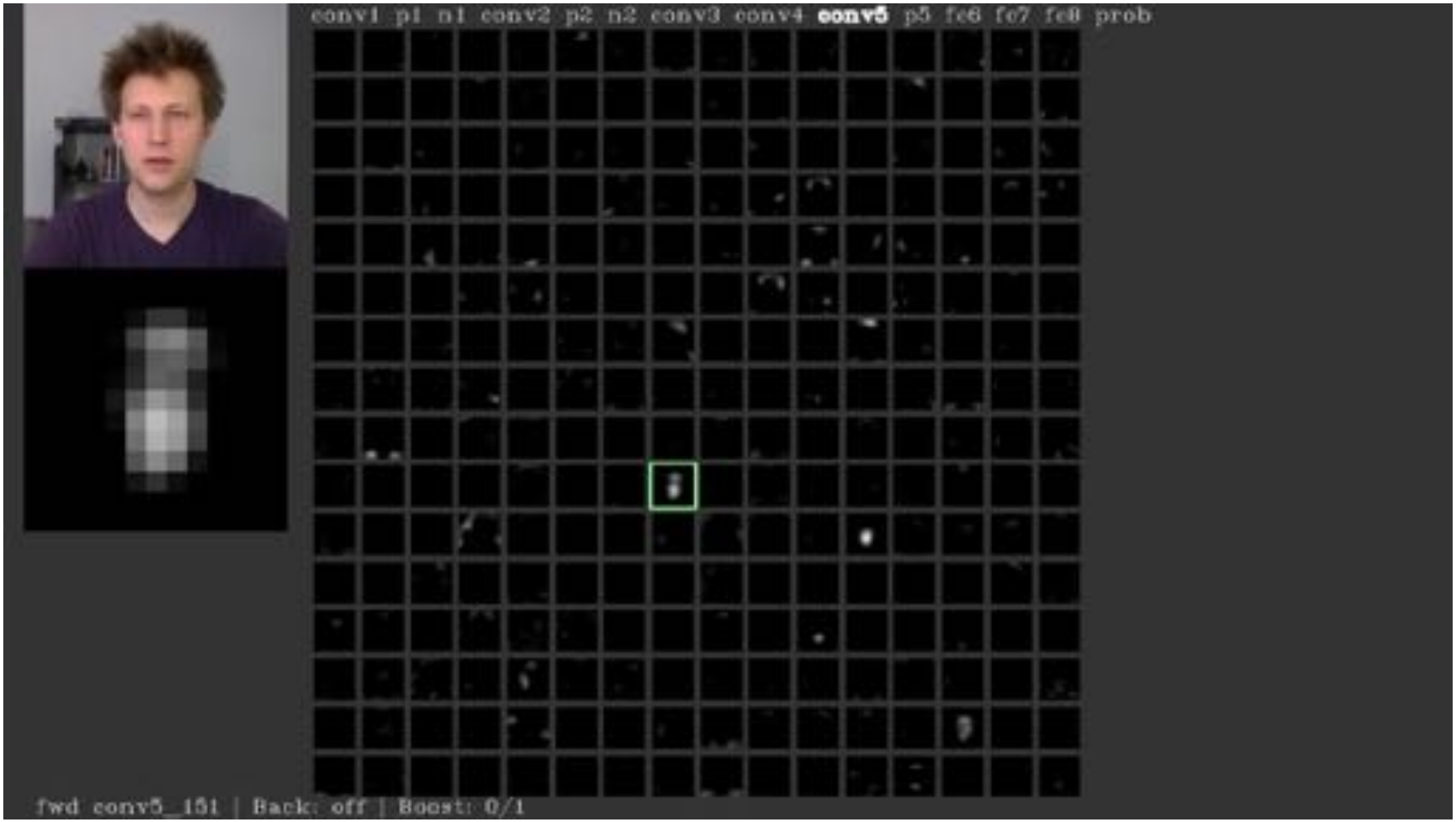


AlexNet - conv1 - 96 filtros

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. and Lipson, H.,
"Understanding neural networks through deep visualization." ICML DL Workshop , 2015

Capas de Convolución - Mapas de Activación

Capas intermedias: Información de alto nivel (caras)



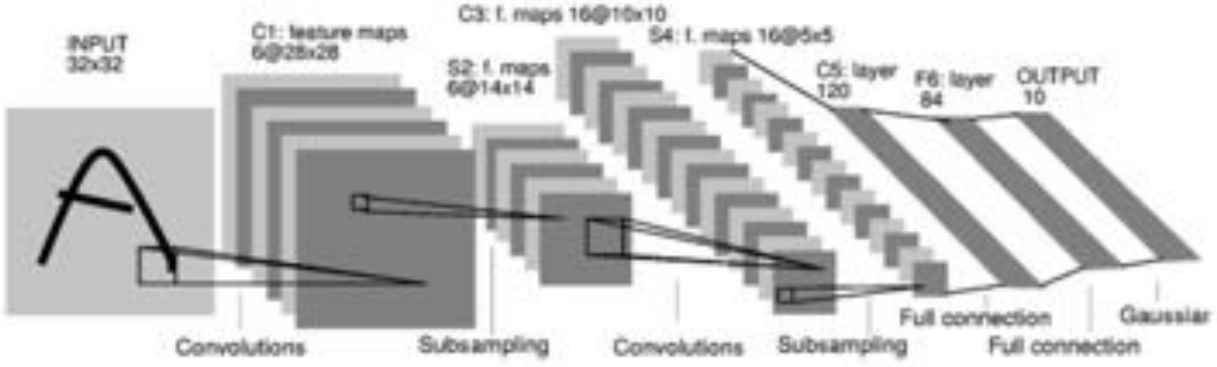
AlexNet - conv5 - 256 filtros

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. and Lipson, H.,
"Understanding neural networks through deep visualization." ICML DL Workshop , 2015

Evolución de las arquitecturas

1998

LeCun et al.



# of transistors

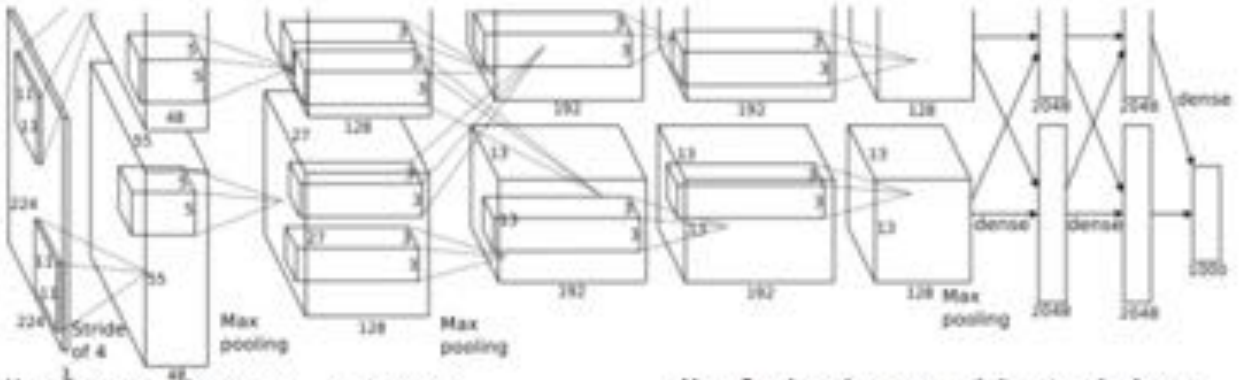
10^6

# of pixels used in training

10^7 NIST

2012

Krizhevsky et al.



# of transistors

10^9

GPUs

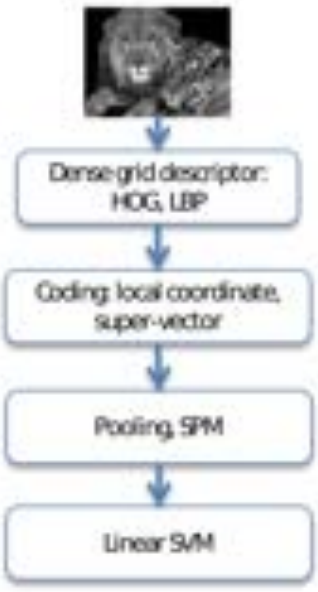
# of pixels used in training

10^{14} IMAGENET

Evolución de las arquitecturas

Year 2010

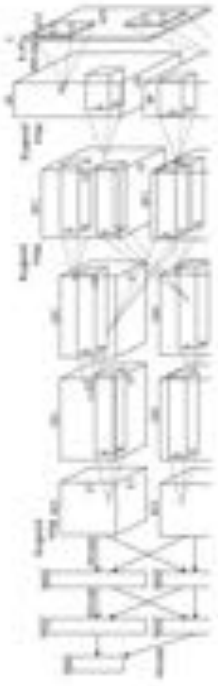
NEC-UIUC



[Lin CVPR 2011]

Year 2012

SuperVision



[Krizhevsky NIPS 2012]

Year 2014

GoogLeNet

[Szegedy arxiv 2014]

VGG

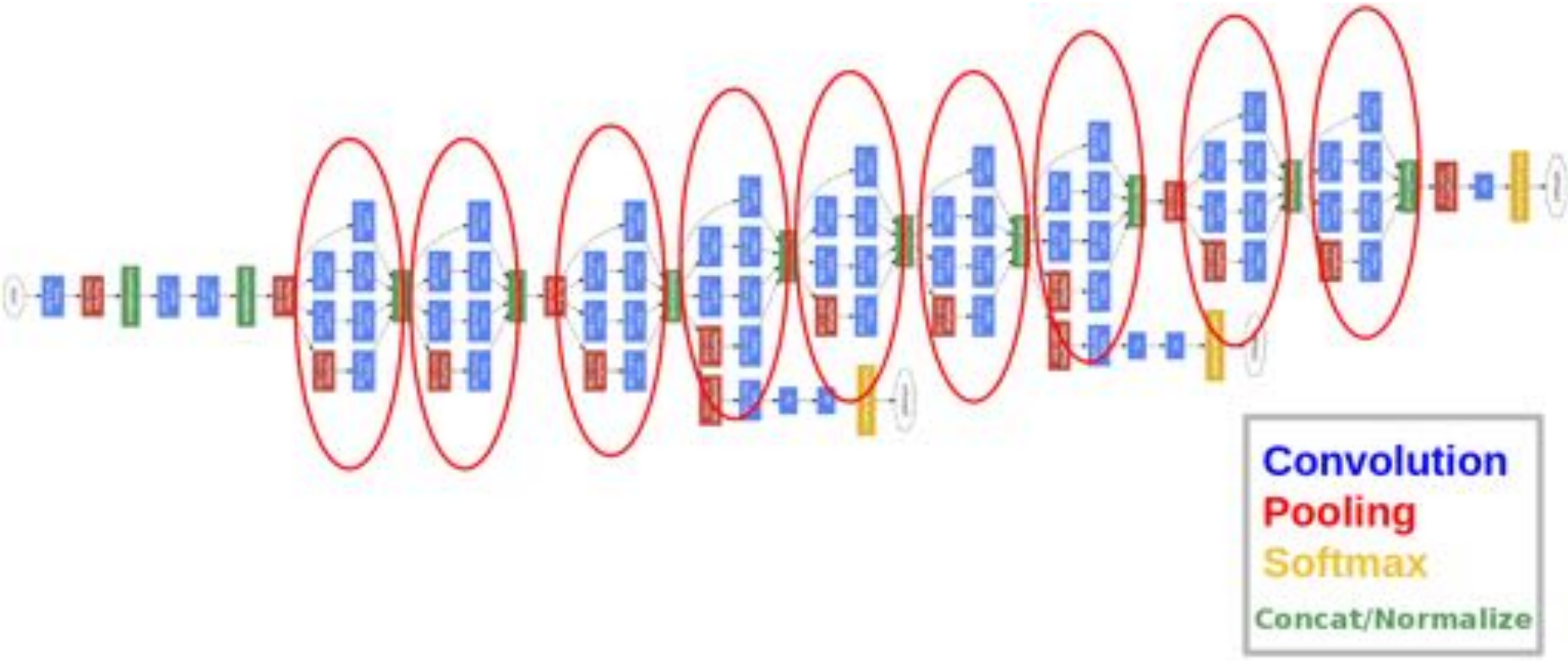
[Simonyan arxiv 2014]

Year 2015

Res-nets

[He et al 2015]

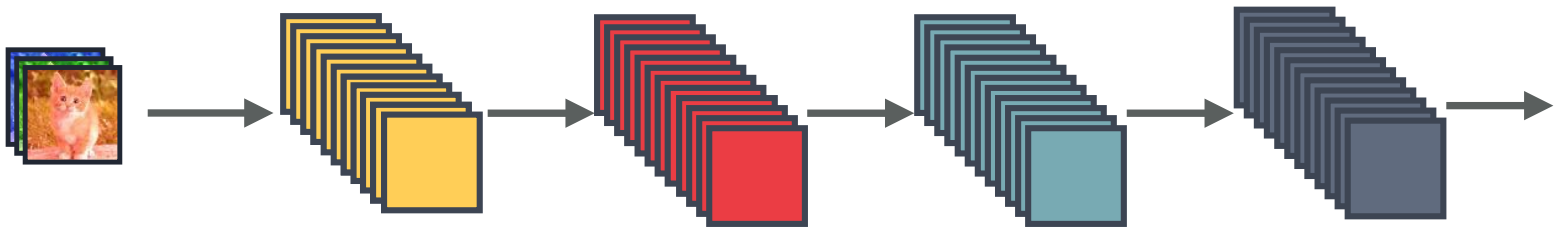
GoogleNet - Inception



C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich
"Going deeper with convolutions". CVPR 2015

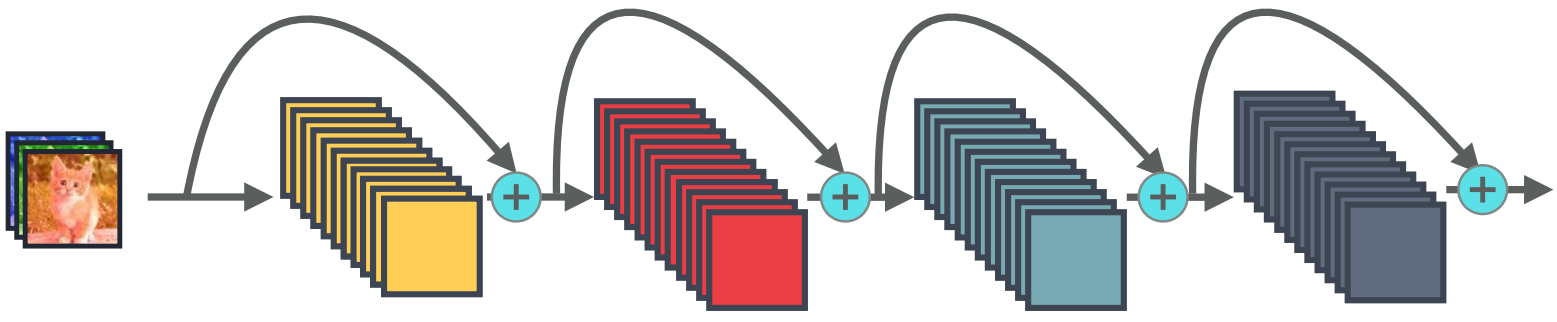
Conectividad en la Arquitectura

Conectividad Estándar



Conectividad en la Arquitectura

Conectividad Residual (*skip connections*)

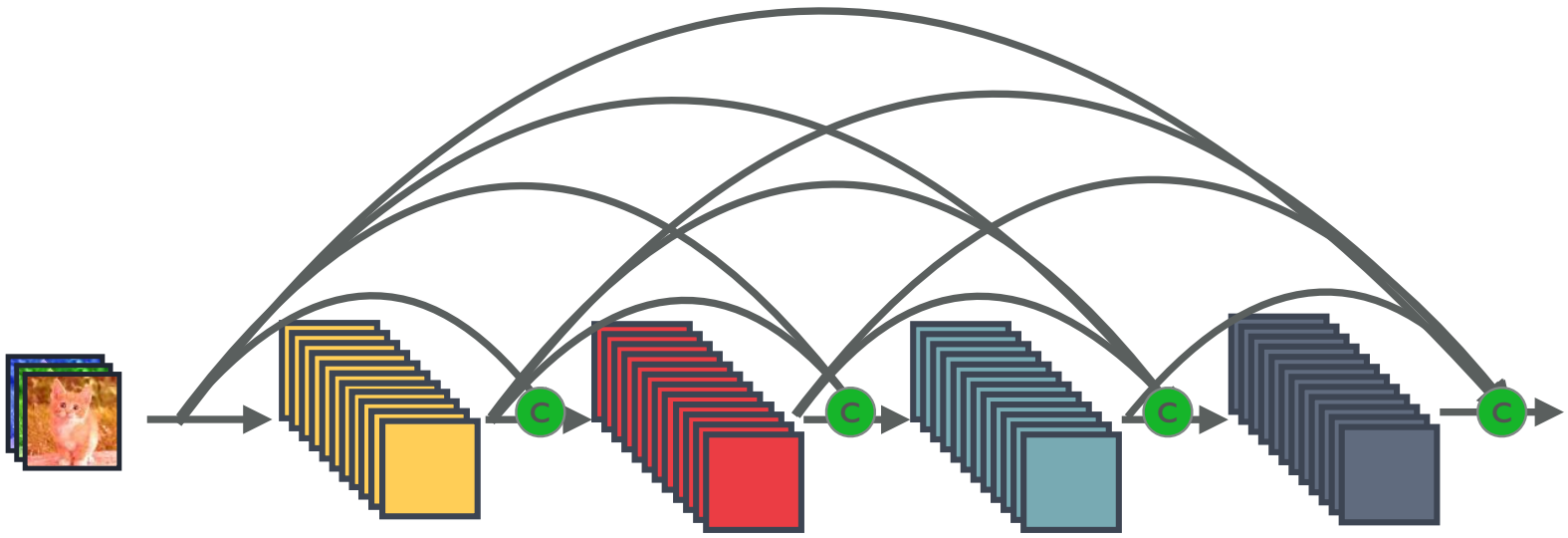


⊕ : Element-wise addition

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
"Deep residual learning for image recognition." CVPR 2016

Conectividad en la Arquitectura

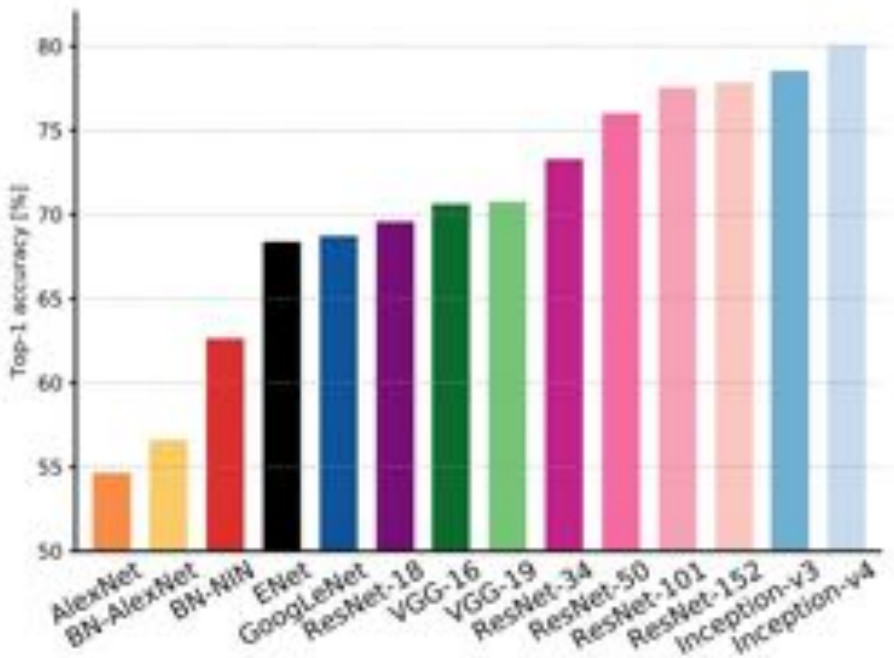
Conectividad Densa (*densenet*)

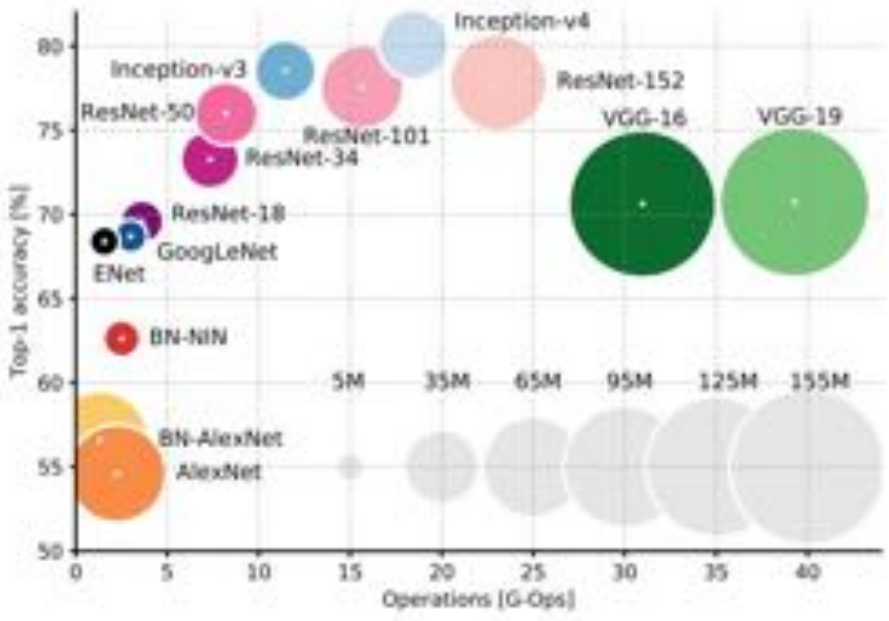


C : Channel-wise concatenation

Huang, Gao, Zhuang Liu, Kilian Q. Weinberger, and Laurens van der Maaten. "Densely connected convolutional networks." CVPR 2017

Evolución de las arquitecturas

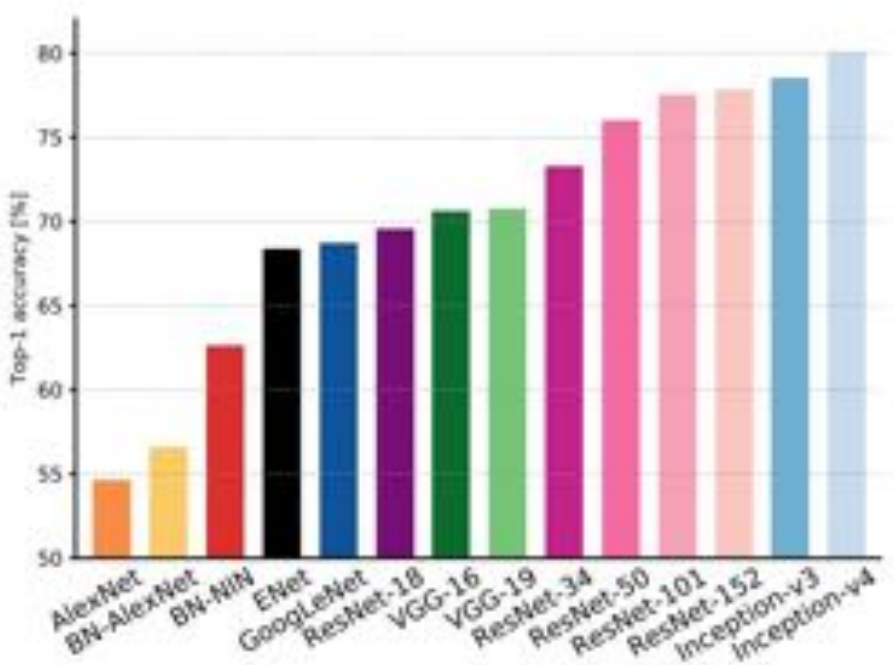


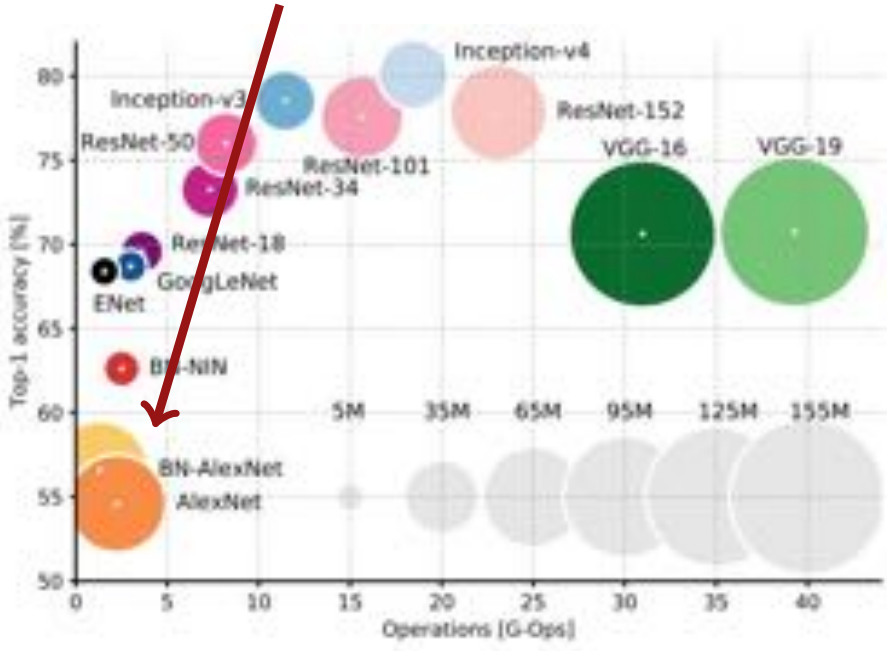


Canziani, A., Paszke, A. and Culurciello, E.,
An analysis of deep neural network models for practical applications. arXiv preprint, 2016

Evolución de las arquitecturas

AlexNet: "Pequeña" pero mucha memoria, baja precisión

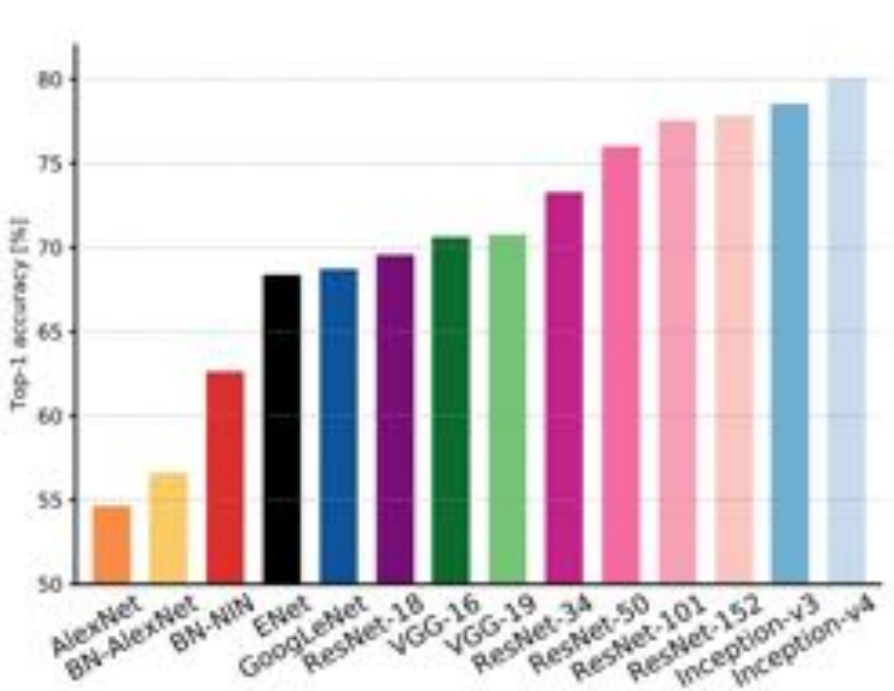


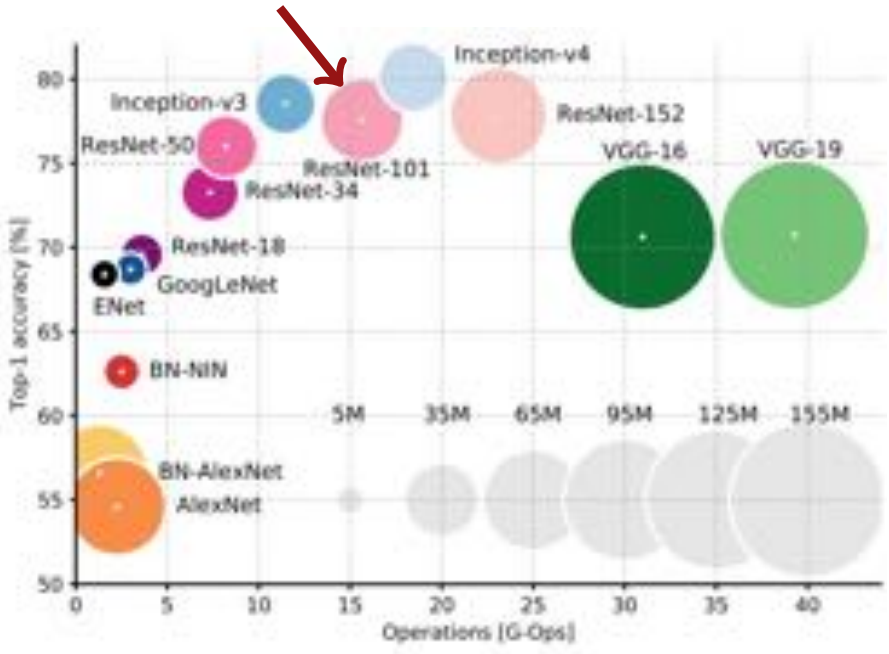


Canziani, A., Paszke, A. and Culurciello, E.,
An analysis of deep neural network models for practical applications. arXiv preprint, 2016

Evolución de las arquitecturas

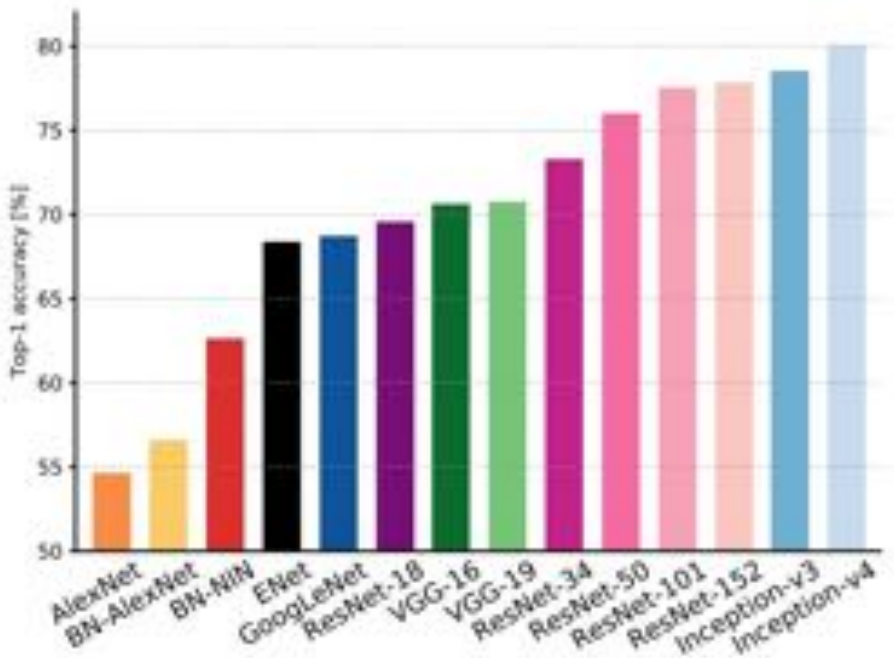
ResNet: Alta precisión, eficiencia media/buena



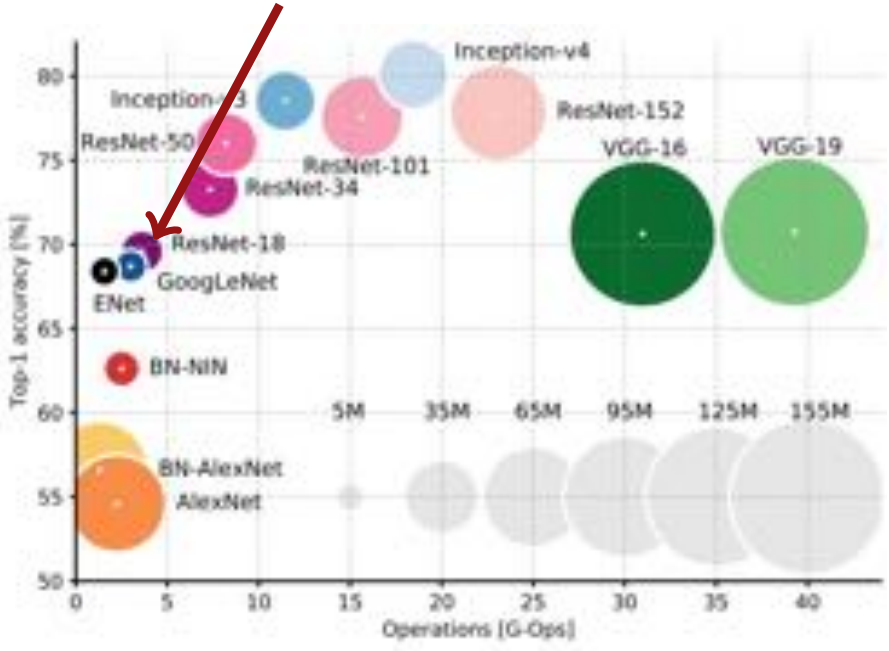


Canziani, A., Paszke, A. and Culurciello, E.,
An analysis of deep neural network models for practical applications. arXiv preprint, 2016

Evolución de las arquitecturas



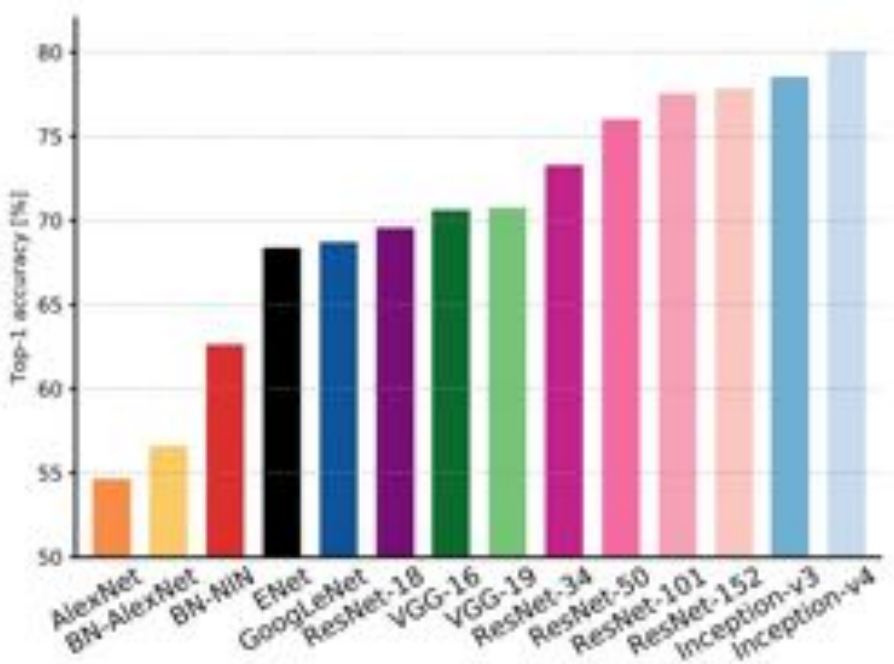
GoogLeNet: Media precisión, excelente eficiencia

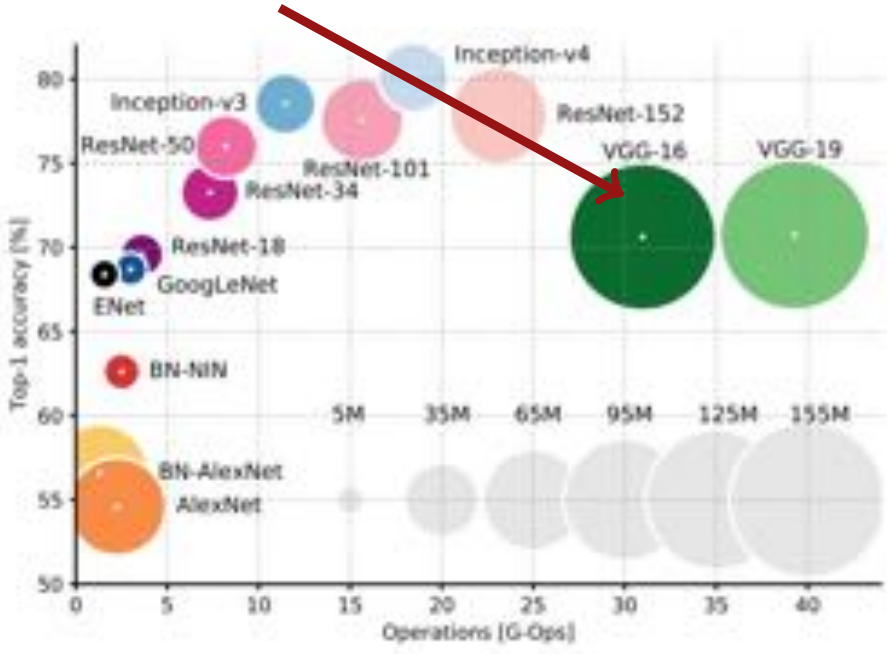


Canziani, A., Paszke, A. and Culurciello, E.,
An analysis of deep neural network models for practical applications. arXiv preprint, 2016

Evolución de las arquitecturas

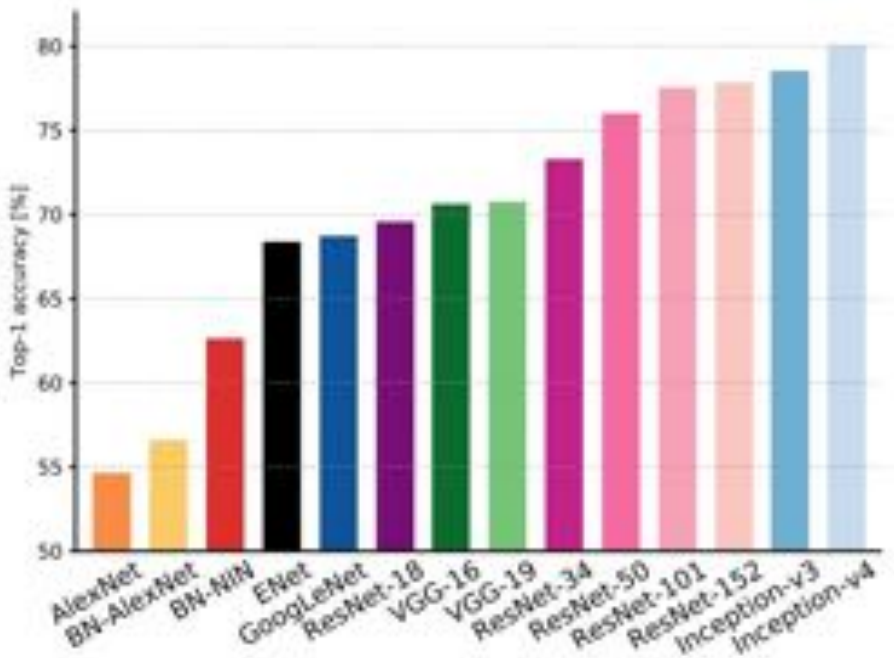
VGG: Mucha memoria, baja eficiencia, media precisión



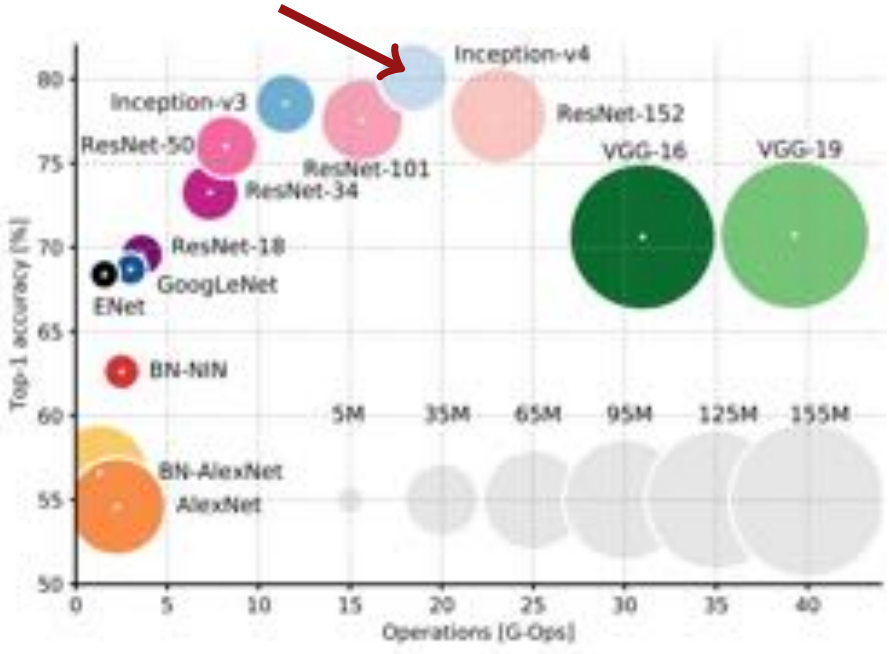


Canziani, A., Paszke, A. and Culurciello, E.,
An analysis of deep neural network models for practical applications. arXiv preprint, 2016

Evolución de las arquitecturas



Inception-v4:: Excelente precisión, media eficiencia



Canziani, A., Paszke, A. and Culurciello, E.,
An analysis of deep neural network models for practical applications. arXiv preprint, 2016

Frameworks para *Deep Learning* (Código Abierto)

theano

Caffe2

PYTORCH

TensorFlow

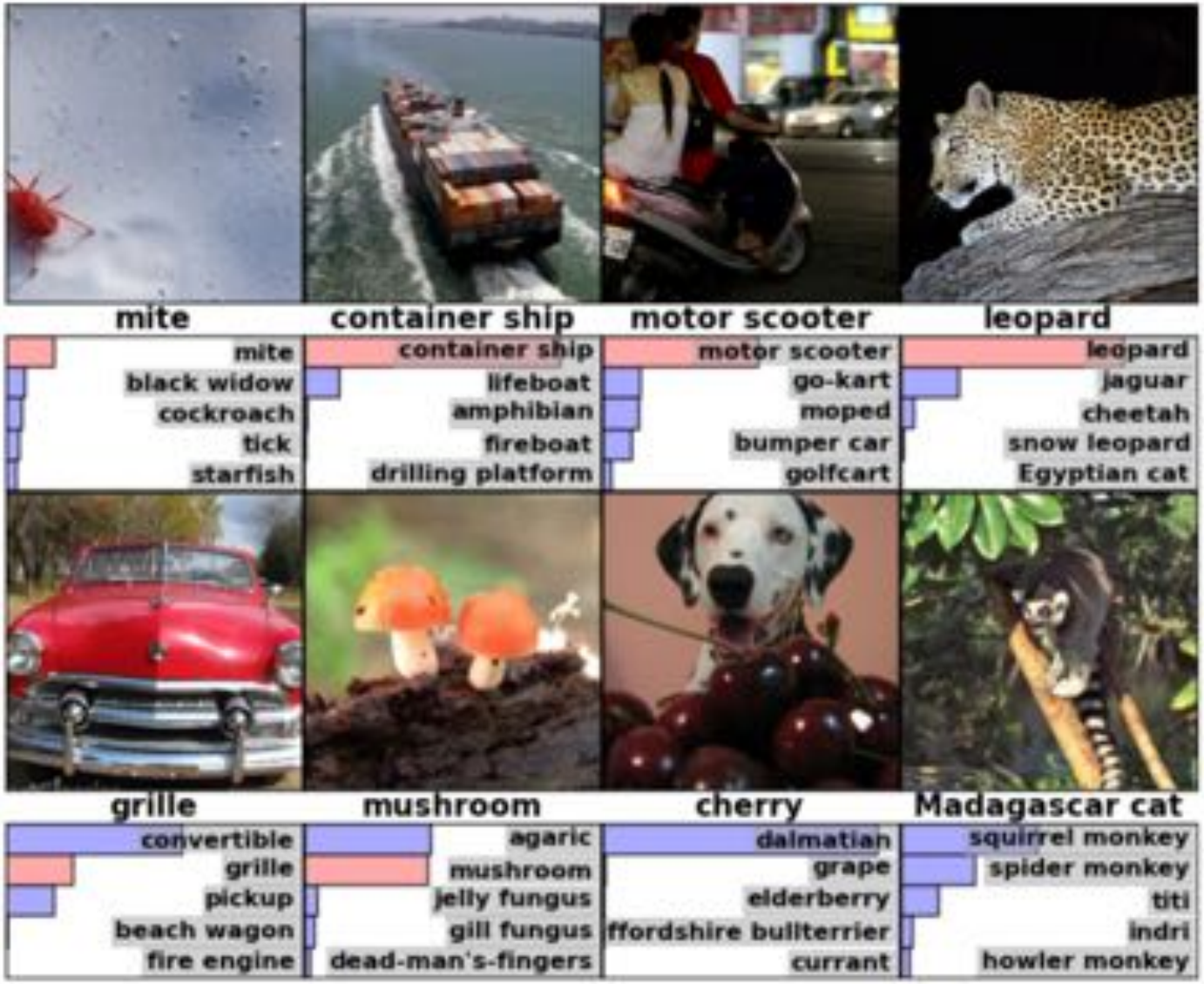
GitHub

torch

Caffe

Aplicaciones

Clasificación



Krizhevsky, Alex, Ilya Sutskever, Geoffrey E. Hinton.
"Imagenet classification with deep convolutional neural networks.", NIPS 2012. (13k citas)

Visión Artificial

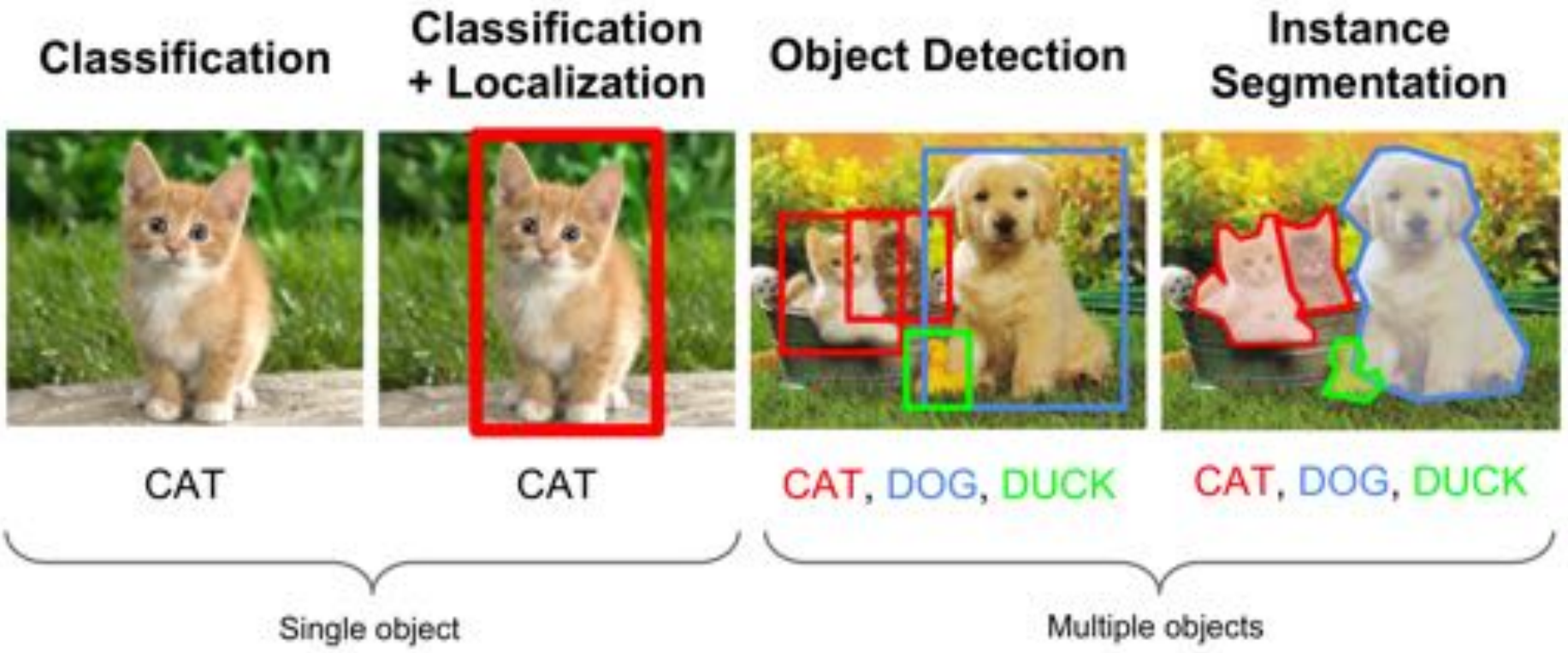
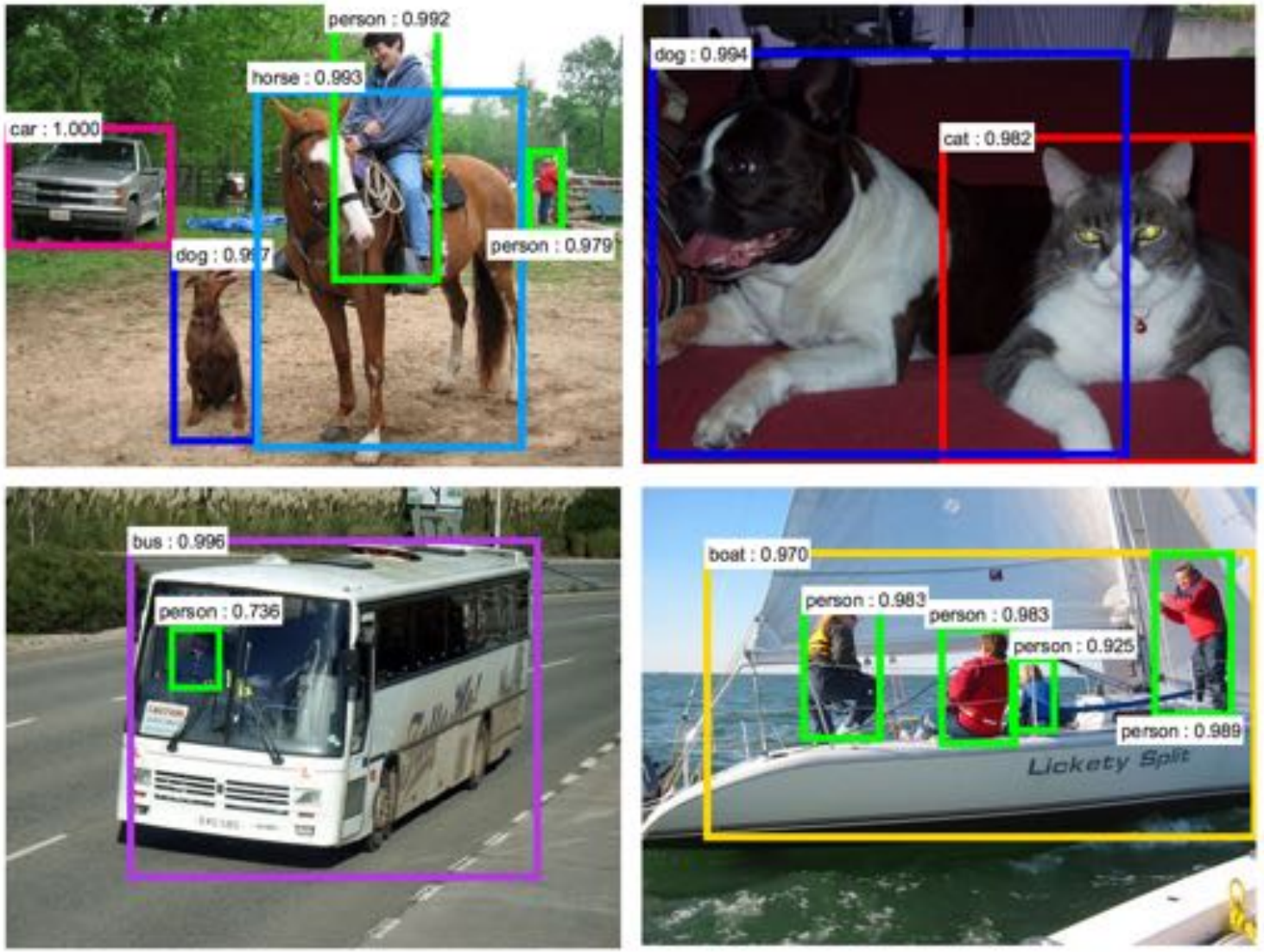


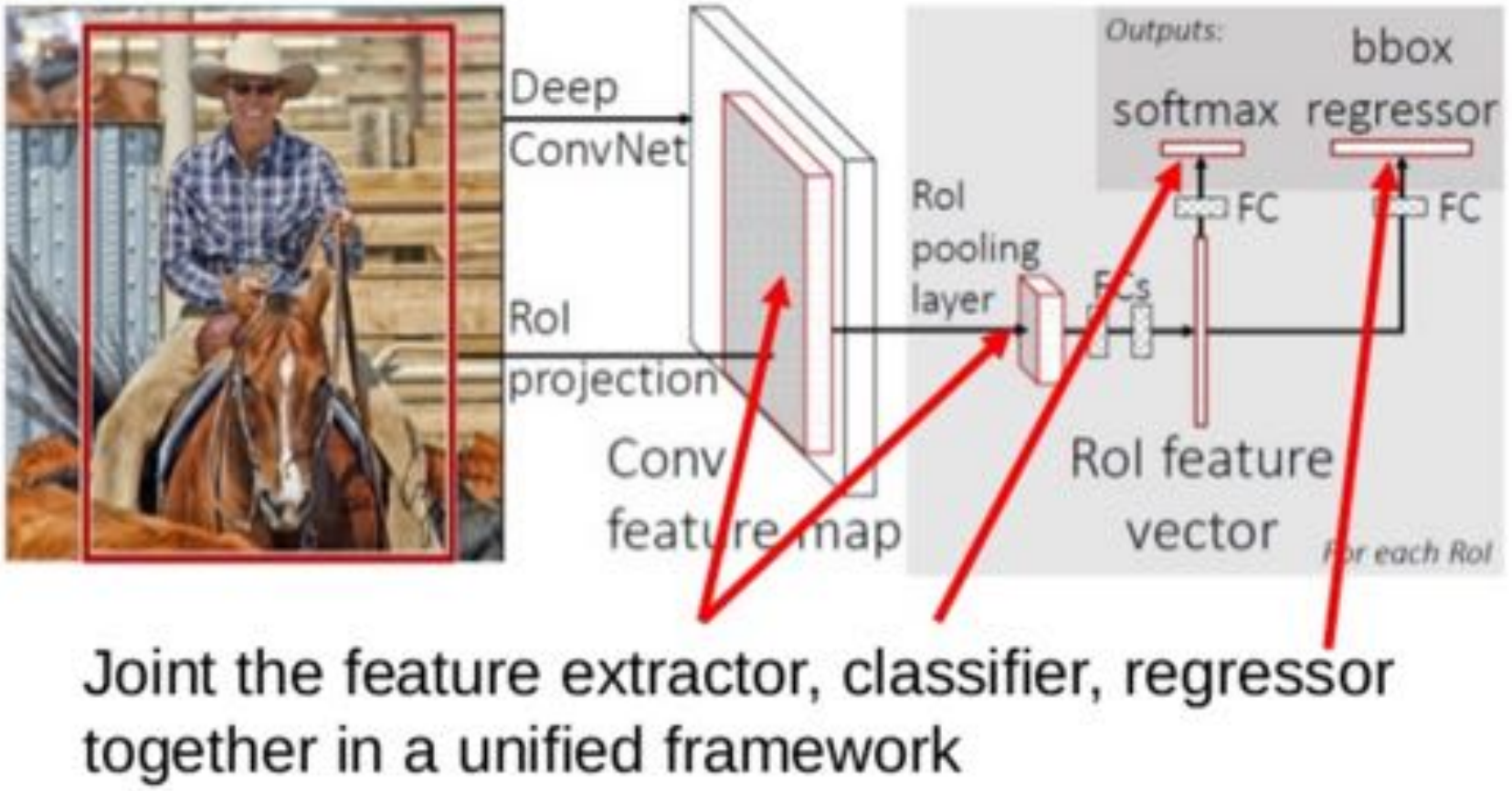
Imagen tomada de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Detección



Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun.
"Faster R-CNN: Towards real-time object detection with region proposal networks." NIPS 2015 (1.3k citas)

Detección (Fast/Faster R-CNN)



Girshick, R., "Fast R-CNN". CVPR 2015

"Faster R-CNN: Towards real-time object detection with region proposal networks." NIPS 2015

Segmentación

Figure 4. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick , "Mask R-CNN.", ICCV 2017

Detección de Pose

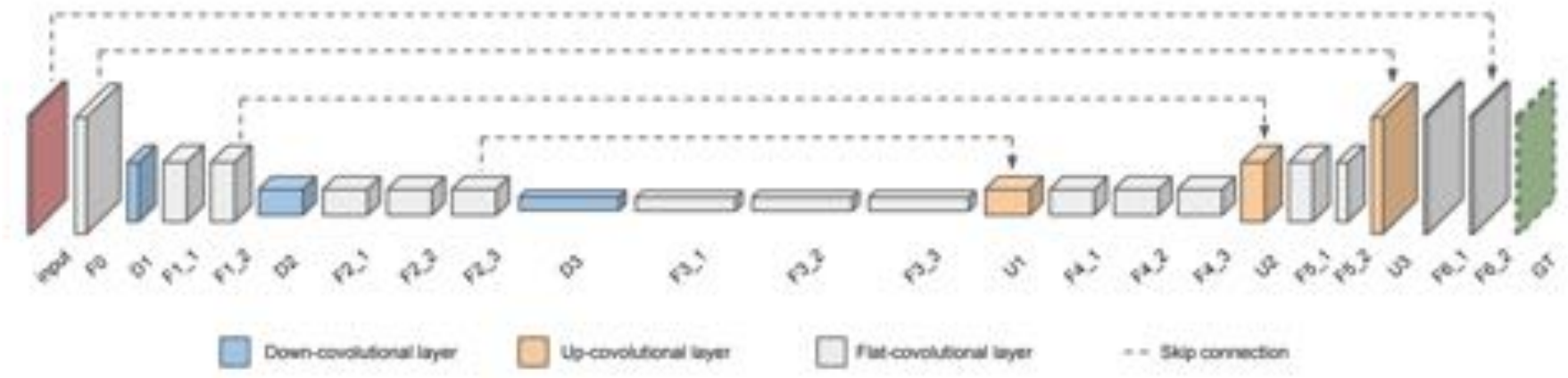
Zhe Cao and Tomas Simon and Shih-En Wei and Yaser Sheikh
"Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields.", CVPR 2017

Restauración: deblurring, superresolution

Su, Shuochen, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang "Deep video deblurring.", CVPR 2017

Dong, Chao, Chen Change Loy, Kaiming He, Xiaoou Tang. "Learning a deep convolutional network for image super-resolution.", ECCV 2014

Deep Video Deblurring



Su, Shuochen, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang "Deep video deblurring.", CVPR 2017

Deep Video Deblurring

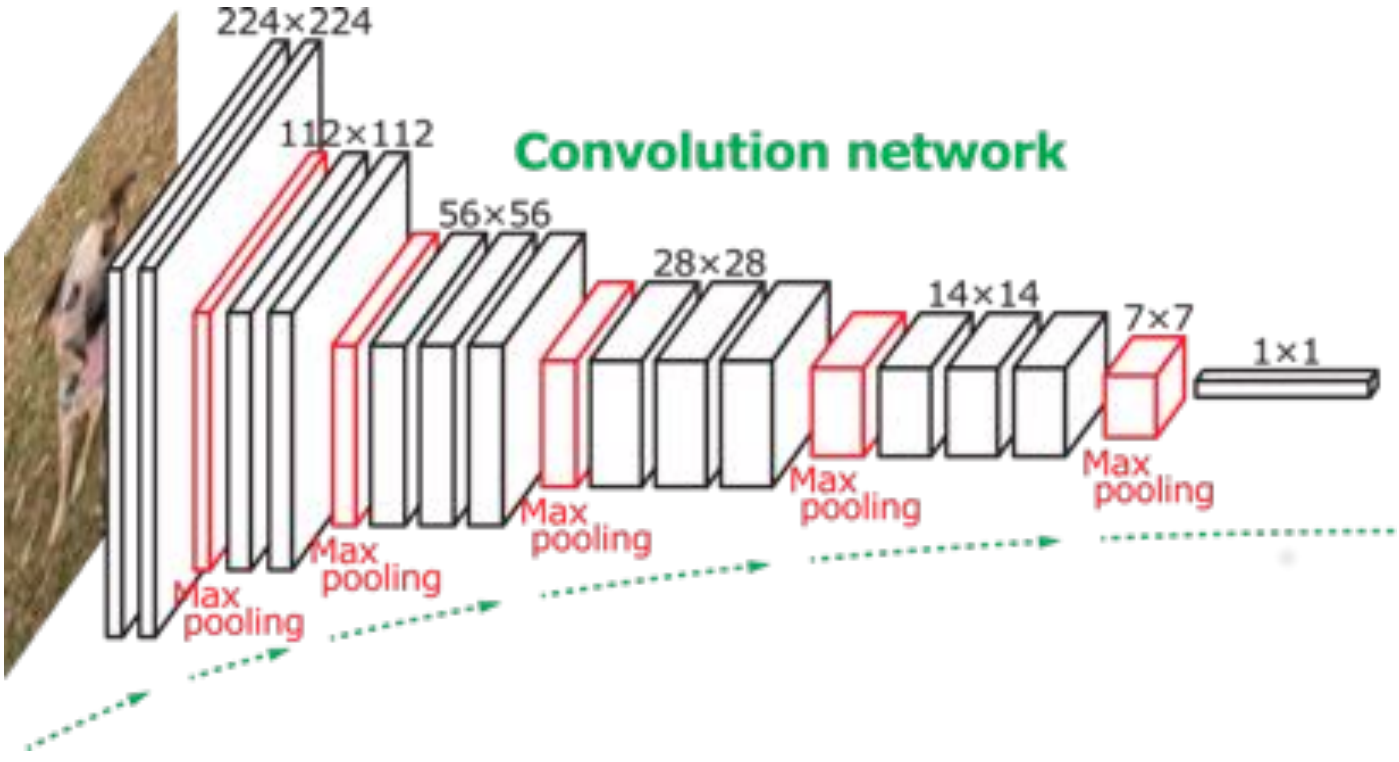
Su, Shuochen, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang "Deep video deblurring.", CVPR 2017

Image Captioning

Slide tomada de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

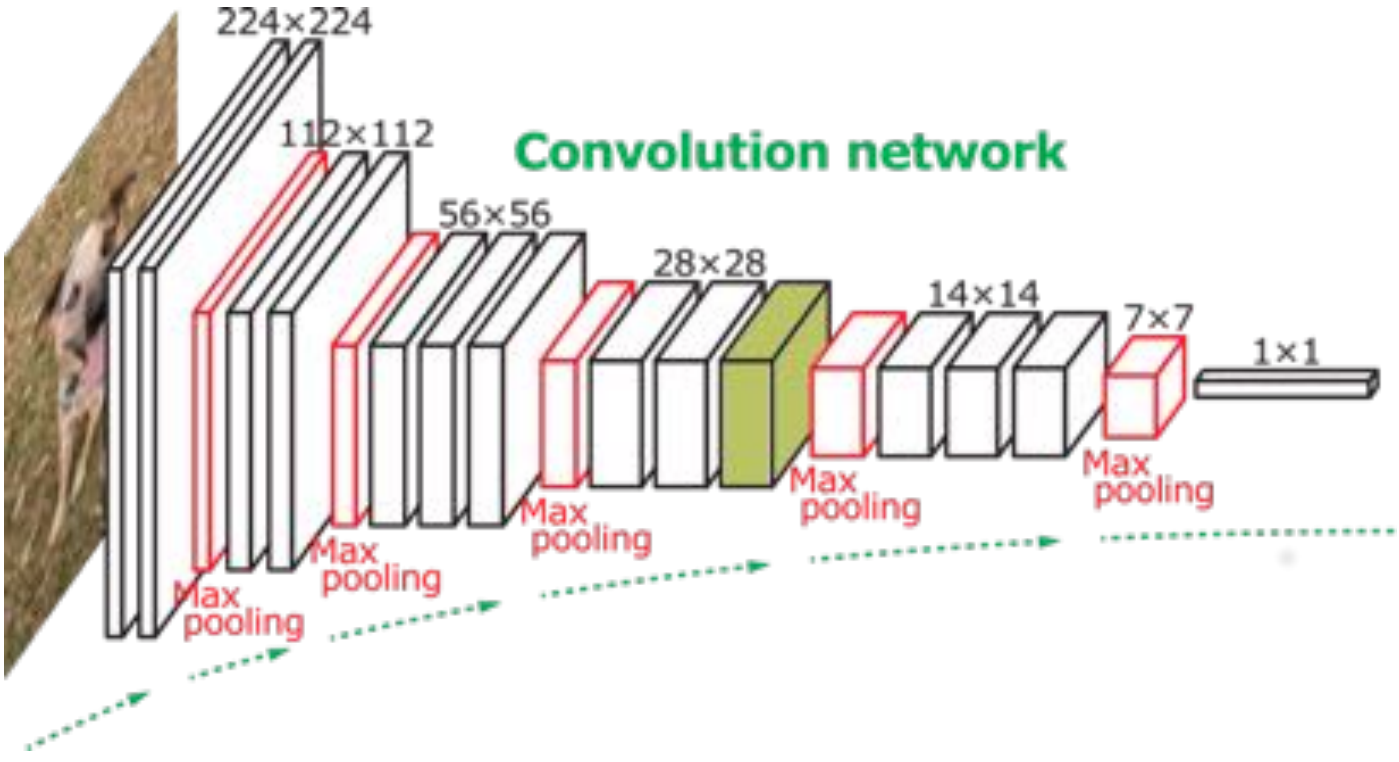
Extracción de Features

- Extraer **features** utilizando red CNN preentrenada (e.g., VGG-16).
- La red se "corta" en una capa intermedia: mapa de *features*

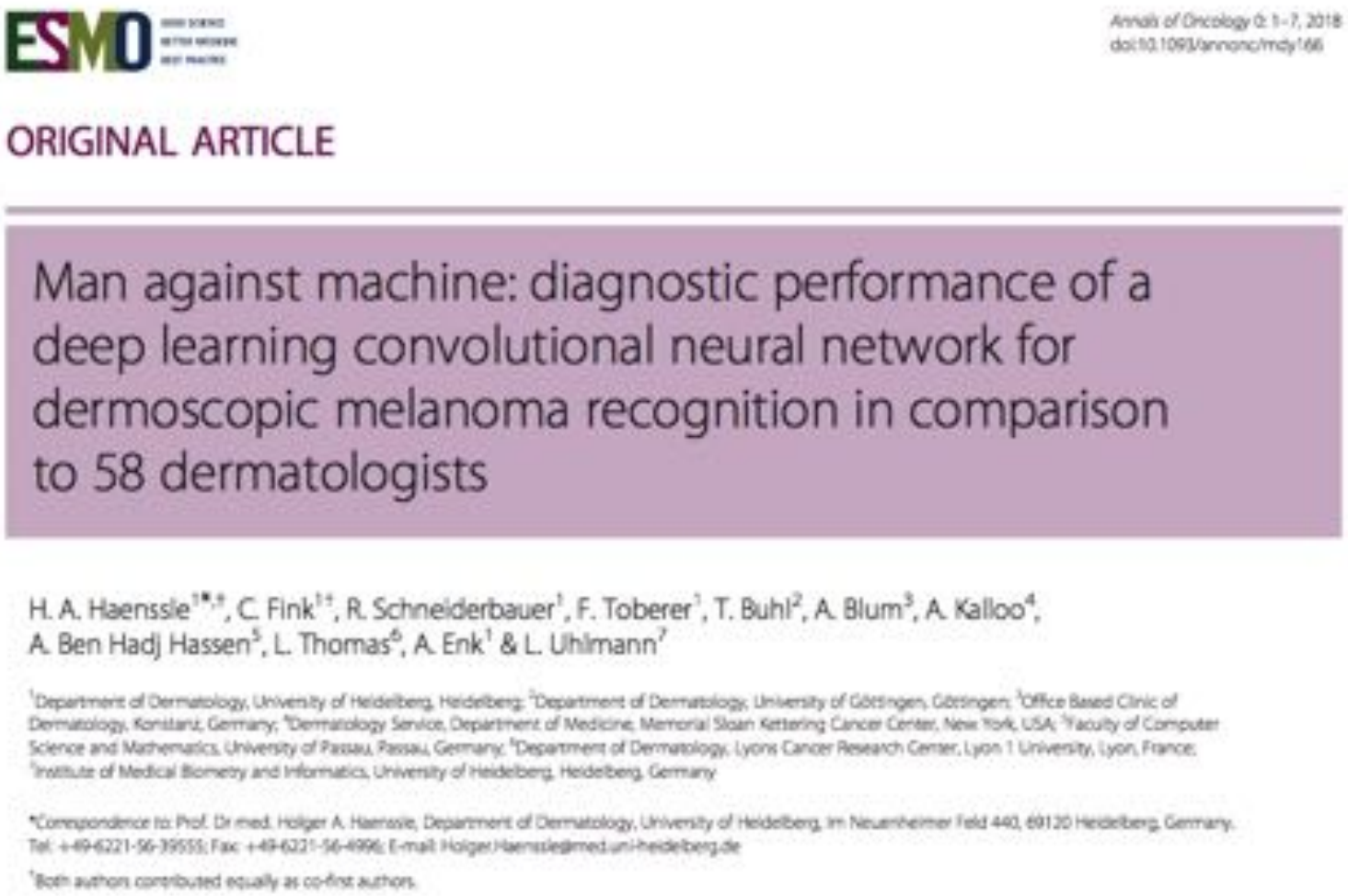


Extracción de Features

- Extraer **features** utilizando red CNN preentrenada (e.g., VGG-16).
- La red se "corta" en una capa intermedia: mapa de *features*



Detección de cancer de piel



Annals of Oncology 0: 1–7, 2018
doi:10.1093/annonc/mdy166

ORIGINAL ARTICLE

Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists

H. A. Haenssle[1*†], C. Fink[1†], R. Schneiderbauer[1], F. Toberer[1], T. Buhl[2], A. Blum[3], A. Kalloo[4], A. Ben Hadj Hassen[5], L. Thomas[6], A. Enk[1] & L. Uhlmann[7]

[1]Department of Dermatology, University of Heidelberg, Heidelberg; [2]Department of Dermatology, University of Göttingen, Göttingen; [3]Office Based Clinic of Dermatology, Konstanz, Germany; [4]Dermatology Service, Department of Medicine, Memorial Sloan Kettering Cancer Center, New York, USA; [5]Faculty of Computer Science and Mathematics, University of Passau, Passau, Germany; [6]Department of Dermatology, Lyons Cancer Research Center, Lyon 1 University, Lyon, France; [7]Institute of Medical Biometry and Informatics, University of Heidelberg, Heidelberg, Germany

*Correspondence to: Prof. Dr med. Holger A. Haenssle, Department of Dermatology, University of Heidelberg, Im Neuenheimer Feld 440, 69120 Heidelberg, Germany. Tel: +49-6221-56-39555; Fax: +49-6221-56-4996; E-mail: Holger.Haenssle@med.uni-heidelberg.de

†Both authors contributed equally as co-first authors.

Haenssle, H.A., et al., Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists
Annals of Oncology, May 2018

Detección de cancer de piel

Abstract

Background

Deep learning convolutional neural networks (CNN) may facilitate melanoma detection, but data comparing a CNN's diagnostic performance to larger groups of dermatologists are lacking.

Methods

Google's Inception v4 CNN architecture was trained and validated using dermoscopic images and corresponding diagnoses. In a comparative cross-sectional reader study a 100-image test-set was used (level-I: dermoscopy only; level-II: dermoscopy plus clinical information and images). Main outcome measures were sensitivity, specificity and area under the curve (AUC) of receiver operating characteristics (ROC) for diagnostic classification (dichotomous) of lesions by the CNN versus an international group of 58 dermatologists during level-I or -II of the reader study. Secondary end points included the dermatologists' diagnostic performance in their management decisions and differences in the diagnostic performance of dermatologists during level-I and -II of the reader study. Additionally, the CNN's performance was compared with the top-five algorithms of the 2016 International Symposium on Biomedical Imaging (ISBI) challenge.

Haenssle, H.A., et al., Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists *Annals of Oncology*, May 2018

Detección de cancer de piel

Results

In level-I dermatologists achieved a mean (±standard deviation) sensitivity and specificity for lesion classification of 86.6% (±9.3%) and 71.3% (±11.2%), respectively. More clinical information (level-II) improved the sensitivity to 88.9% (±9.6%, $P = 0.19$) and specificity to 75.7% (±11.7%, $P < 0.05$). The CNN ROC curve revealed a higher specificity of 82.5% when compared with dermatologists in level-I (71.3%, $P < 0.01$) and level-II (75.7%, $P < 0.01$) at their sensitivities of 86.6% and 88.9%, respectively. The CNN ROC AUC was greater than the mean ROC area of dermatologists (0.86 versus 0.79, $P < 0.01$). The CNN scored results close to the top three algorithms of the ISBI 2016 challenge.

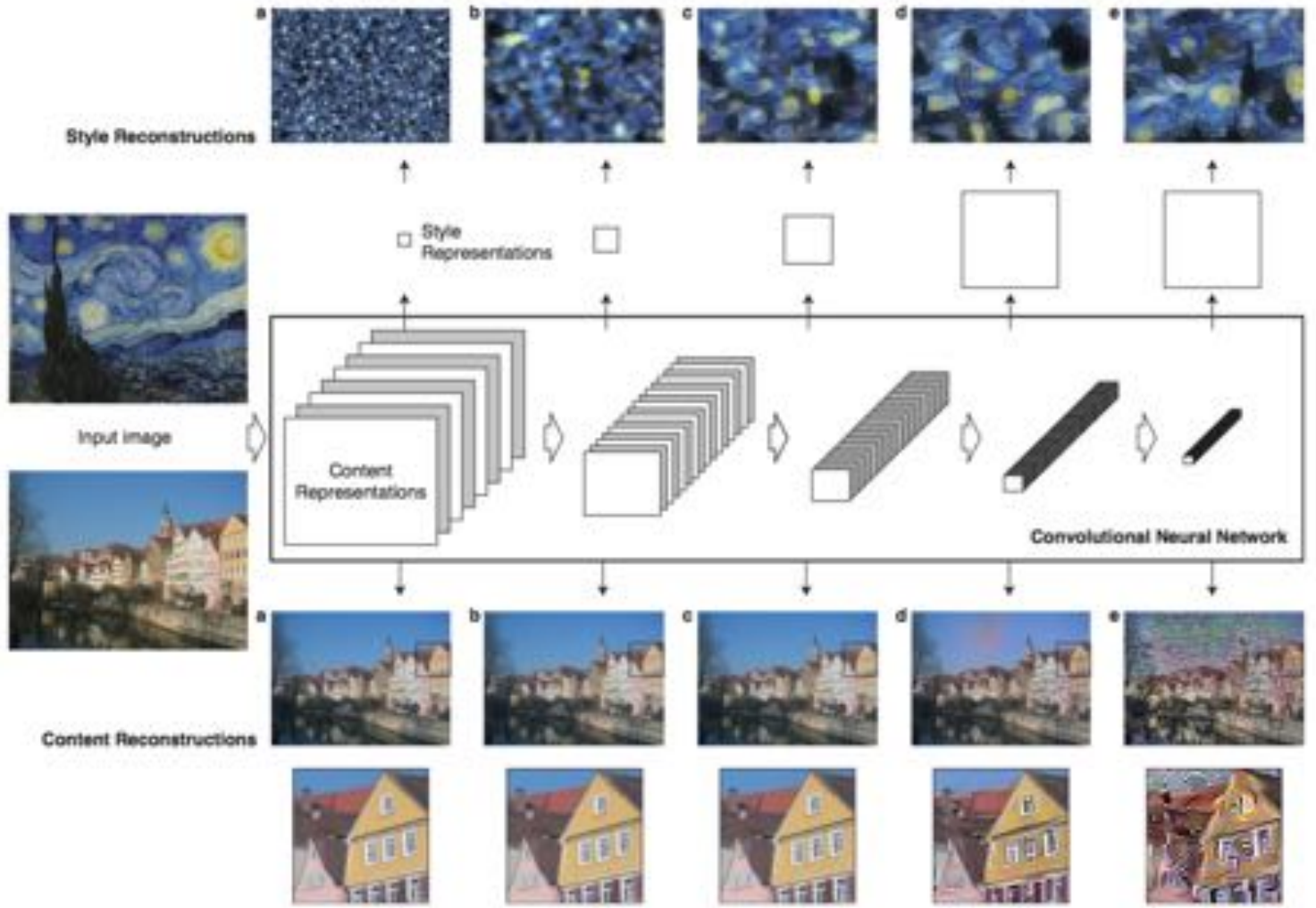
Conclusions

For the first time we compared a CNN's diagnostic performance with a large international group of 58 dermatologists, including 30 experts. Most dermatologists were outperformed by the CNN. Irrespective of any physicians' experience, they may benefit from assistance by a CNN's image classification.

Haenssle, H.A., et al., Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists *Annals of Oncology*, May 2018

Síntesis de Textura, Copa de Estilo

- Separar textura de estilo en imágenes



Gatys et al. 2015

Síntesis de Textura, Copa de Estilo

- Separar textura de estilo en imágenes

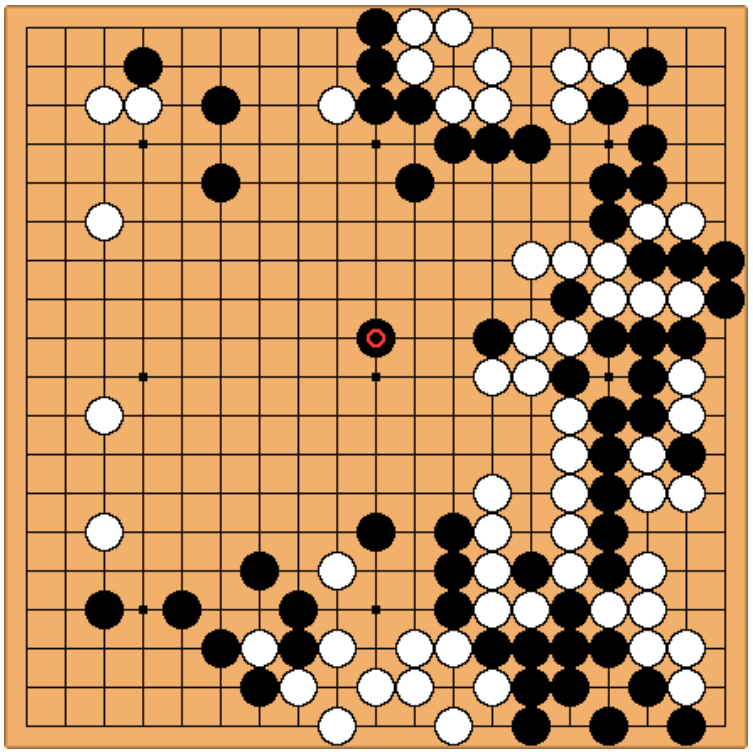
Gatys et al. 2015

Reinforcement Learning

- Aprendizaje por refuerzos (*reinforcement learning*)
- DeepMind: Atari games y Alpha Go

Reinforcement Learning

- Aprendizaje por refuerzos (*reinforcement learning*)
- DeepMind: Atari games y Alpha Go



Generative Adversarial Networks

- Entrenamiento en base a una competencia entre dos jugadores
- **Generator Network:** Tratar de generar muestras realistas que confundan al discriminador
- **Discriminator Network:** Tratar de distinguir entre muestras reales o sintéticas (generadas por el generador)

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., "Generative adversarial nets. In Advances in neural information processing systems", NIPS 2014

Generative Adversarial Networks

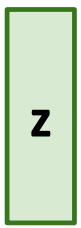
- Entrenamiento en base a una competencia entre dos jugadores
- **Generator Network:** Tratar de generar muestras realistas que confundan al discriminador
- **Discriminator Network:** Tratar de distinguir entre muestras reales o sintéticas (generadas por el generador)

real images
(training set)

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., "Generative adversarial nets. In Advances in neural information processing systems", NIPS 2014

Generative Adversarial Networks

- Entrenamiento en base a una competencia entre dos jugadores
- **Generator Network:** Tratar de generar muestras realistas que confundan al discriminador
- **Discriminator Network:** Tratar de distinguir entre muestras reales o sintéticas (generadas por el generador)



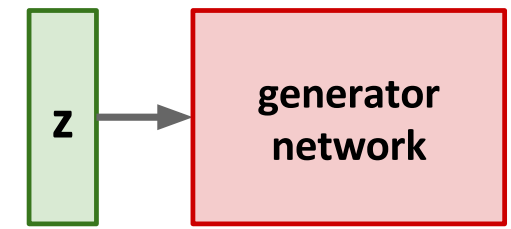
random noise

real images (training set)

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., "Generative adversarial nets. In Advances in neural information processing systems", NIPS 2014

Generative Adversarial Networks

- Entrenamiento en base a una competencia entre dos jugadores
- **Generator Network:** Tratar de generar muestras realistas que confundan al discriminador
- **Discriminator Network:** Tratar de distinguir entre muestras reales o sintéticas (generadas por el generador)



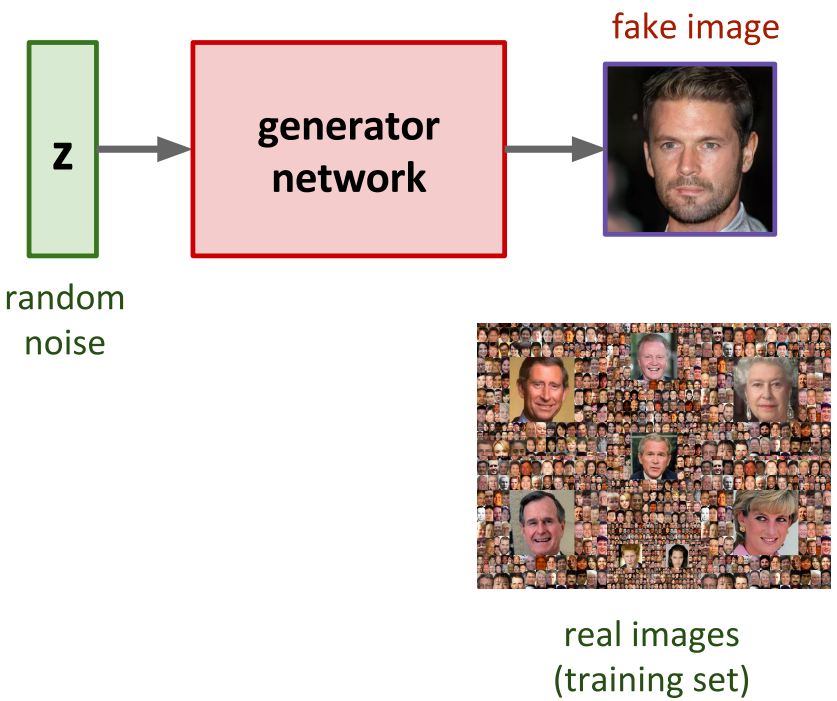
random noise

real images (training set)

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., "Generative adversarial nets. In Advances in neural information processing systems", NIPS 2014

Generative Adversarial Networks

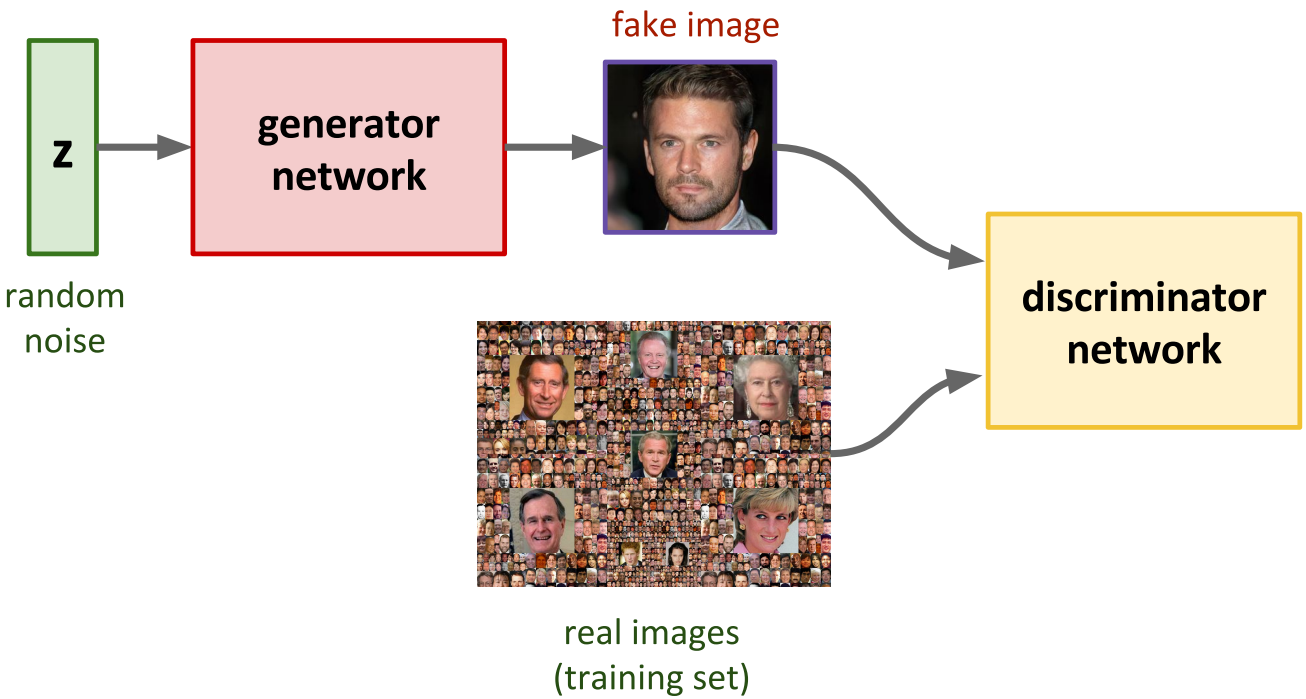
- Entrenamiento en base a una competencia entre dos jugadores
- **Generator Network:** Tratar de generar muestras realistas que confundan al discriminador
- **Discriminator Network:** Tratar de distinguir entre muestras reales o sintéticas (generadas por el generador)



Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., "Generative adversarial nets. In Advances in neural information processing systems", NIPS 2014

Generative Adversarial Networks

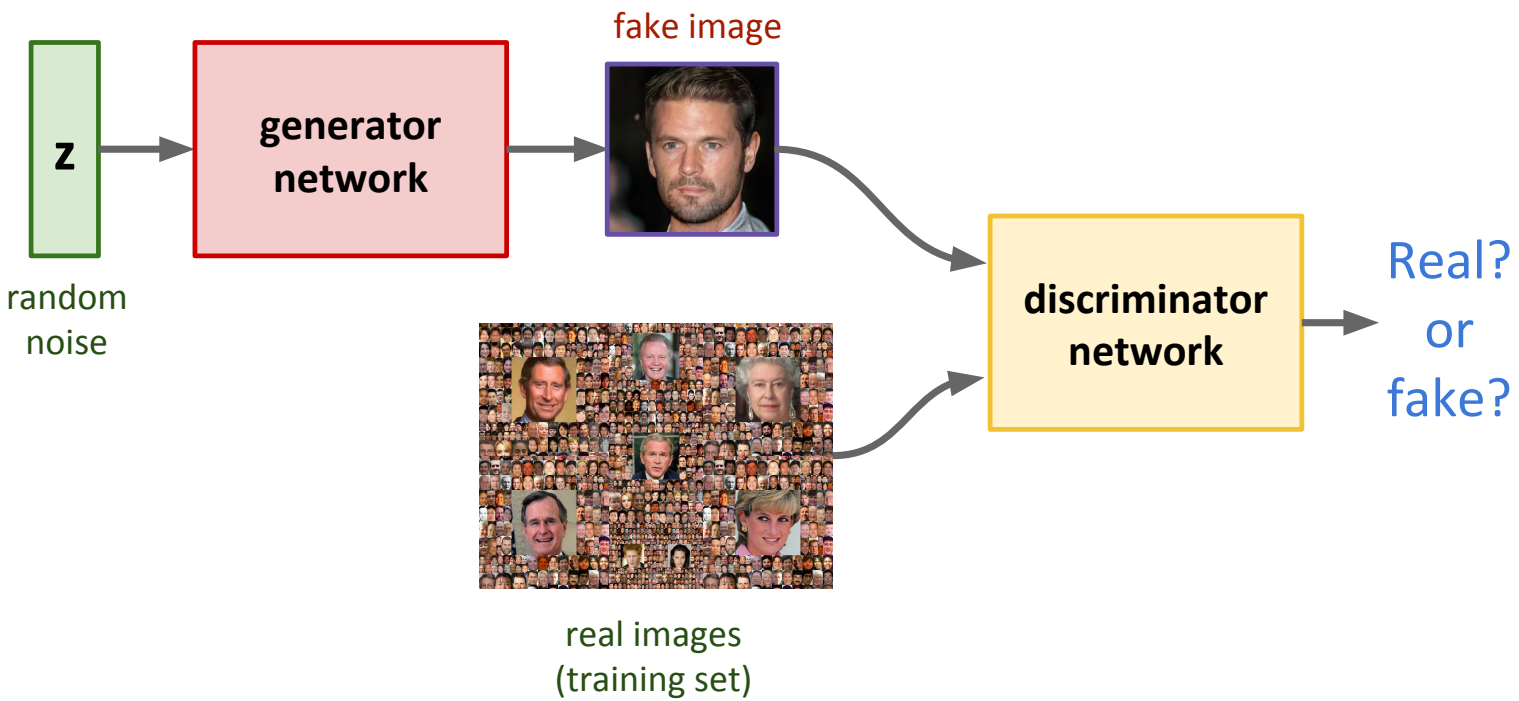
- Entrenamiento en base a una competencia entre dos jugadores
- **Generator Network:** Tratar de generar muestras realistas que confundan al discriminador
- **Discriminator Network:** Tratar de distinguir entre muestras reales o sintéticas (generadas por el generador)



Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., "Generative adversarial nets. In Advances in neural information processing systems", NIPS 2014

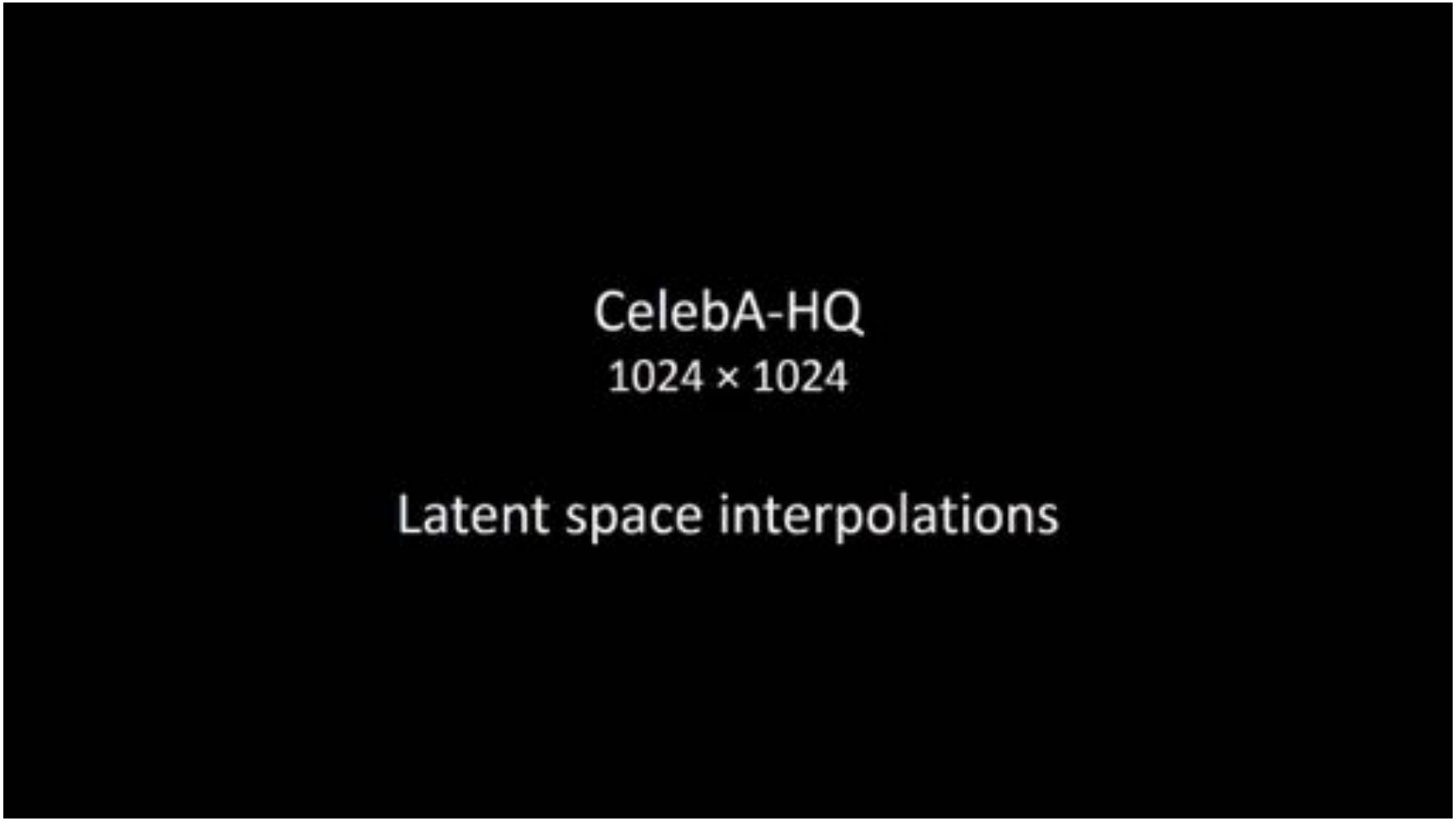
Generative Adversarial Networks

- Entrenamiento en base a una competencia entre dos jugadores
- **Generator Network:** Tratar de generar muestras realistas que confundan al discriminador
- **Discriminator Network:** Tratar de distinguir entre muestras reales o sintéticas (generadas por el generador)



Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., "Generative adversarial nets. In Advances in neural information processing systems", NIPS 2014

Generative Adversarial Networks



Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen
"Progressive Growing of GANs for Improved Quality, Stability, and Variation", Submitted to ICLR 2018

Discusión: lo que sabemos

- El **Aprendizaje Profundo** existe y **funciona** :)
- En casi todos los problemas de **Visión** es estado del arte
- **Clave:** Hardware + Masividad de datos + Astucias de entrenamiento/arquitectura
- Técnicas de diseño/entrenamiento que funcionan en muchas ocasiones (ADAM, *skip connections*, *batch normalization*,...)
- Tecnología relativamente madura: Varias bibliotecas abiertas, con gran soporte y buena documentación (e.g., Tensorflow, Pytorch)

Discusión: lo que nos gustaría saber

- **Teoría:** Falta entender más en profundidad.
- Capacidad de los modelos (¿sobreajuste?)
- ¿Cómo entrenar? ¿Cómo evitar mínimos locales, puntos silla?
- Aprendizaje no supervisado, por refuerzo
- Transferencia de conocimiento / Aprendizaje Activo
- *Deep learning* versus Biología
- ¿Cómo armar arquitecturas que funcionen "siempre"?
- ¿Cuántos datos es *big data*? ¿*One shot* learning?
- Mejorar entendimiento de representaciones.
- Modelos generativos