

REDES NEURONALES CONVOLUCIONALES



“Las computadoras han sido capaces de derrotar al mejor jugador de ajedrez humano, pero solo recientemente de manera equiparable a las habilidades promedio del ser humano para reconocer objetos o palabras” Ian Goodfellow

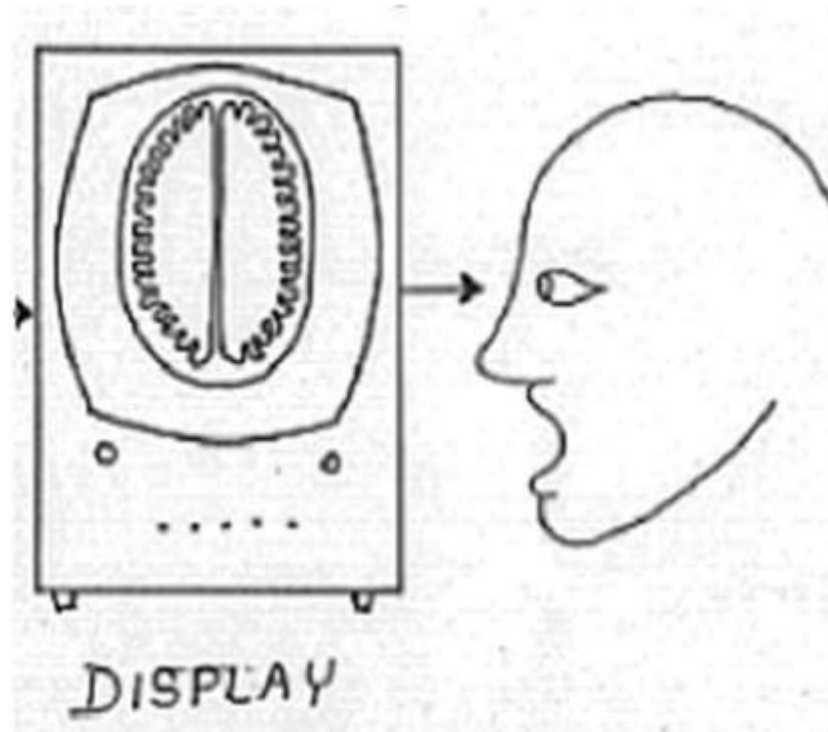
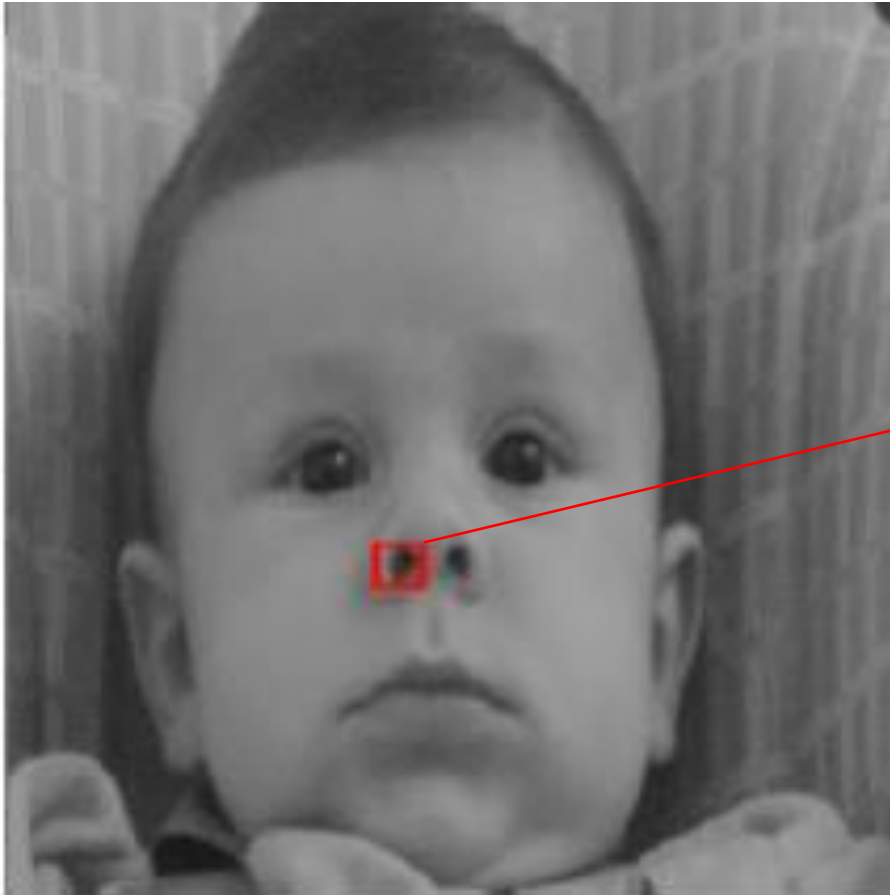


Imagen Digital 2Df(x,y)



99	71	61	51	49	40	35	53	86	99
93	74	53	56	48	46	48	72	85	102
101	69	57	53	54	52	64	82	88	101
107	82	64	63	59	60	61	90	93	100
114	93	78	69	72	85	85	94	95	99
117	106	94	92	97	101	100	108	105	99
116	114	109	106	105	108	108	102	107	110
115	113	109	114	111	111	113	108	111	115
110	113	111	109	106	108	108	110	120	122
103	107	106	108	109	114	120	124	124	132

Imagen Digital 2D



Una imagen...

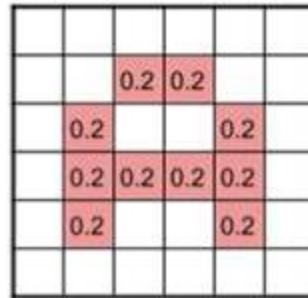
		0.6	0.6		
	0.6			0.6	
	0.6	0.6	0.6	0.6	
	0.6			0.6	

...es una matriz de pixeles.

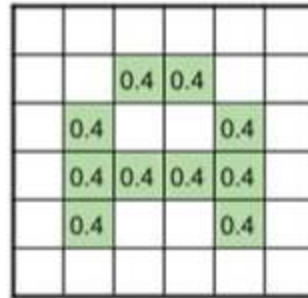
El valor de los pixeles va de 0 a 255 pero se normaliza para la red neuronal de 0 a 1

Tamaño $6*6=36$

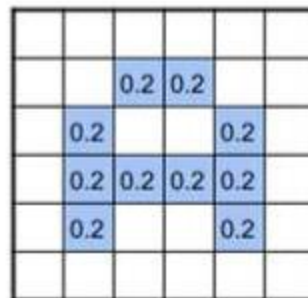
Imagen Digital 2D



Si la imagen es a color, estará compuesta de tres canales: rojo, verde, azul.



¿ Cómo obtener características ?



Filtros

- Matriz de coeficientes conocida **filtro**, **máscara** o **núcleo (*kernel*)** de **convolución**.
- **Idea intuitiva:** se pasa el filtro en cada píxel de la imagen, aplicando los coeficientes según donde caigan.

Máscara de convolución con pesos W 's

-1/4	-1/4
-1/4	-1/4

Aunque este filtro es par se utilizan
Filtros impares- Simetría de la imagen

Imagen de entrada, **A**

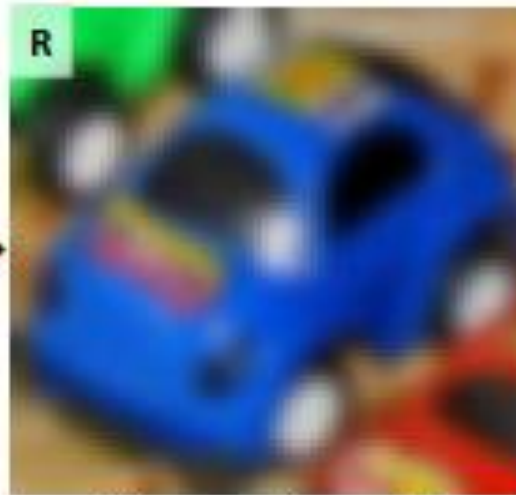
92	78	82
45	80	130
39	115	154

Imagen de salida, **R**

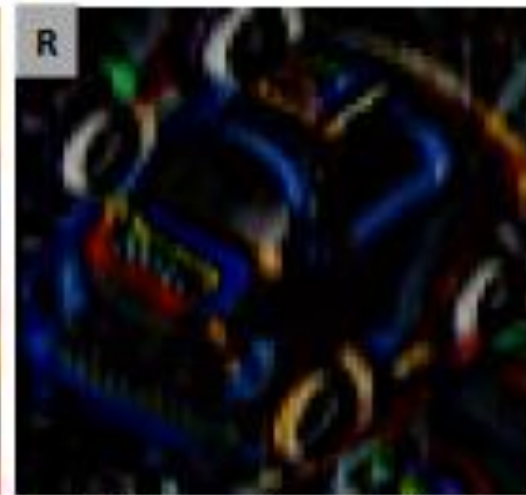
Σ

Filtros y convolución.

- Sobre una imagen se pueden aplicar **sucesivas operaciones** de convolución: $\dots M3 \otimes (M2 \otimes (M1 \otimes A))$



Máscara de media aplicada 4 veces



Máscara de media + máscara de resta

Filtros y convolución - Efectos

A. Imagen original



B. Suaviz. gauss. 40x40



Suma: $0,3A+0,7B$

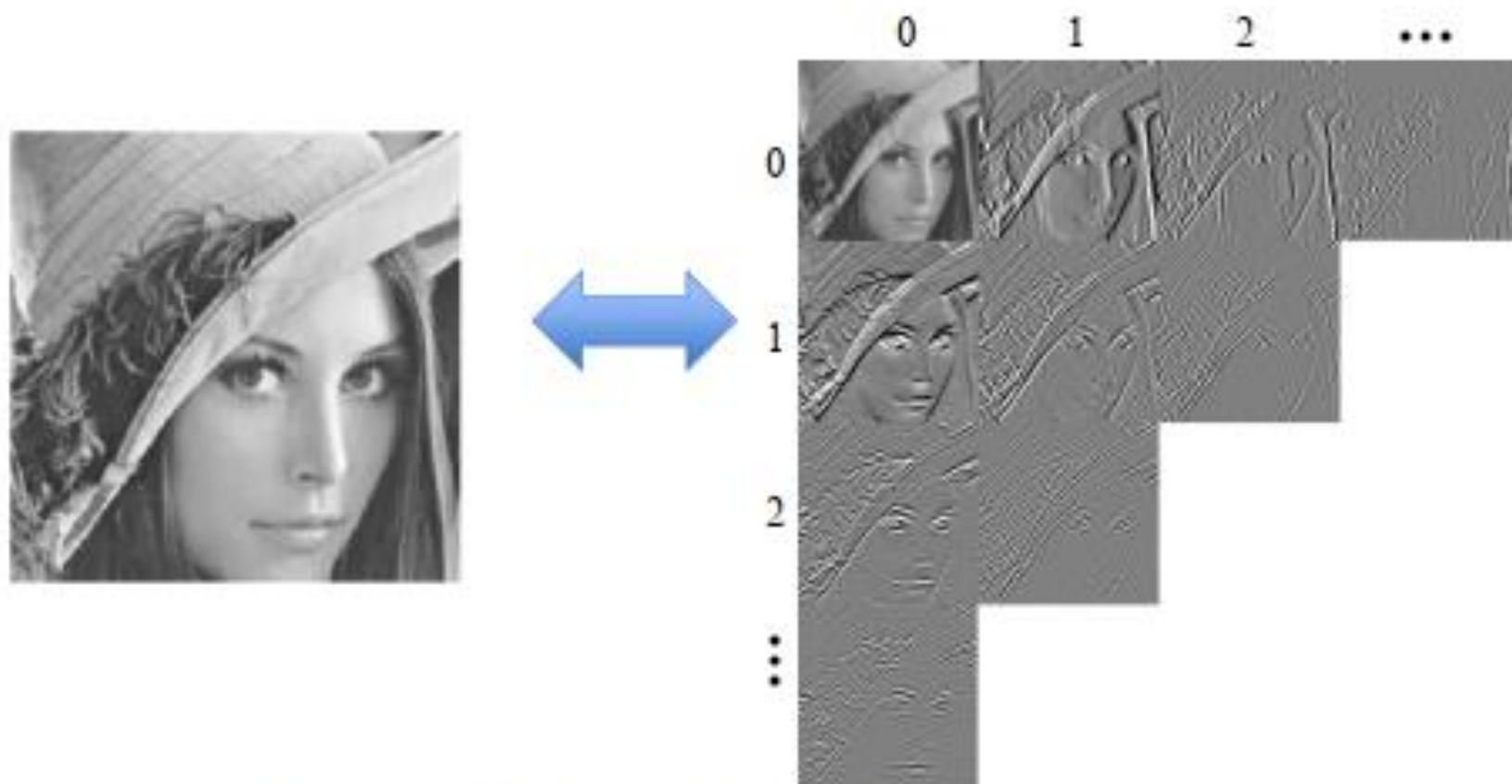


Cambios de luminosidad

- Invariancia a cambios de luminosidad, color de la fuente de luz, etc.



Descomposición en diferentes patrones



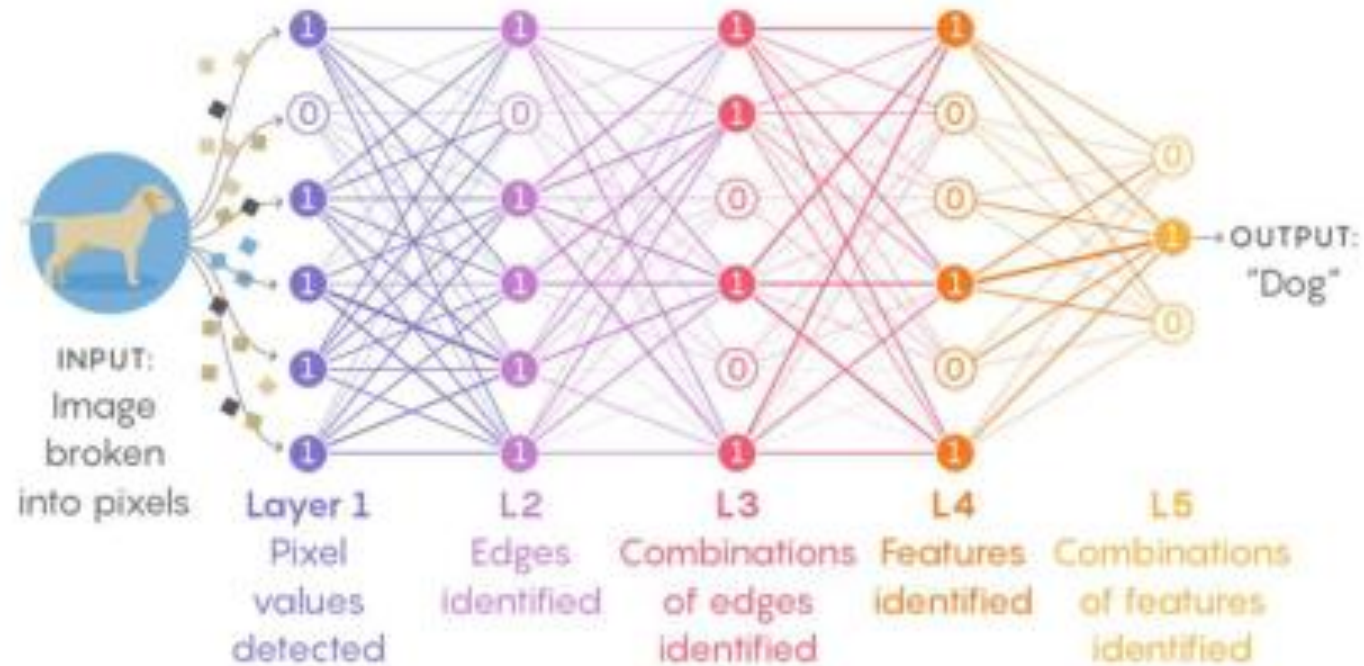
..¡para extraer características significativas!



- A los tres años, un niño ha visto cientos de millones de fotos del mundo real. Son muchos ejemplos de entrenamiento - Dígitos

Las redes neuronales convolucionales son diferentes. La idea es tomar la mayor cantidad de imágenes (dígitos) para su entrenamiento

Red neuronal convolucional (CNN)

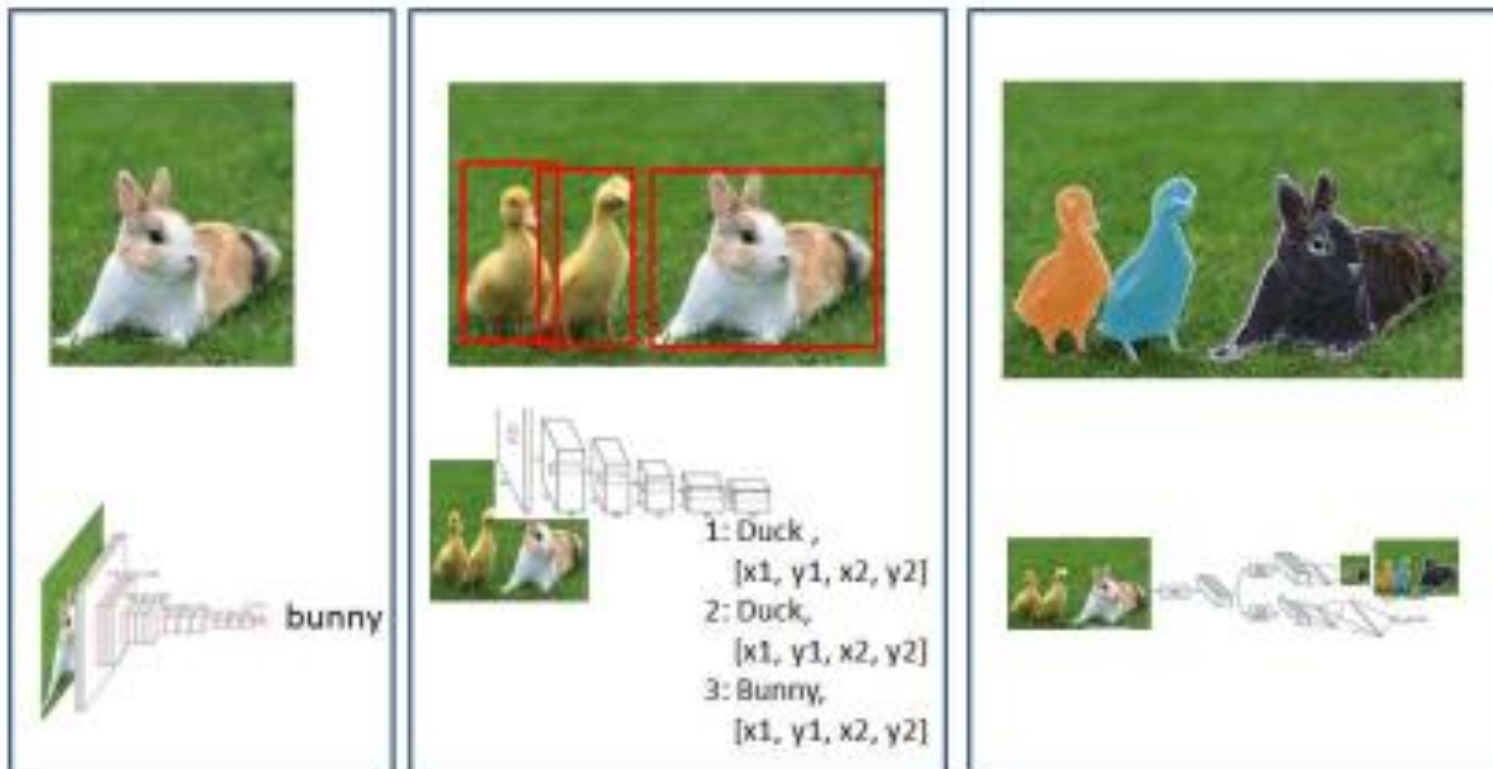


"En lugar de centrarme únicamente en mejores y mejores algoritmos, mi punto de vista fue proporcionar a los algoritmos datos de entrenamiento que un niño recibe a través de la experiencia tanto en cantidad como en calidad."

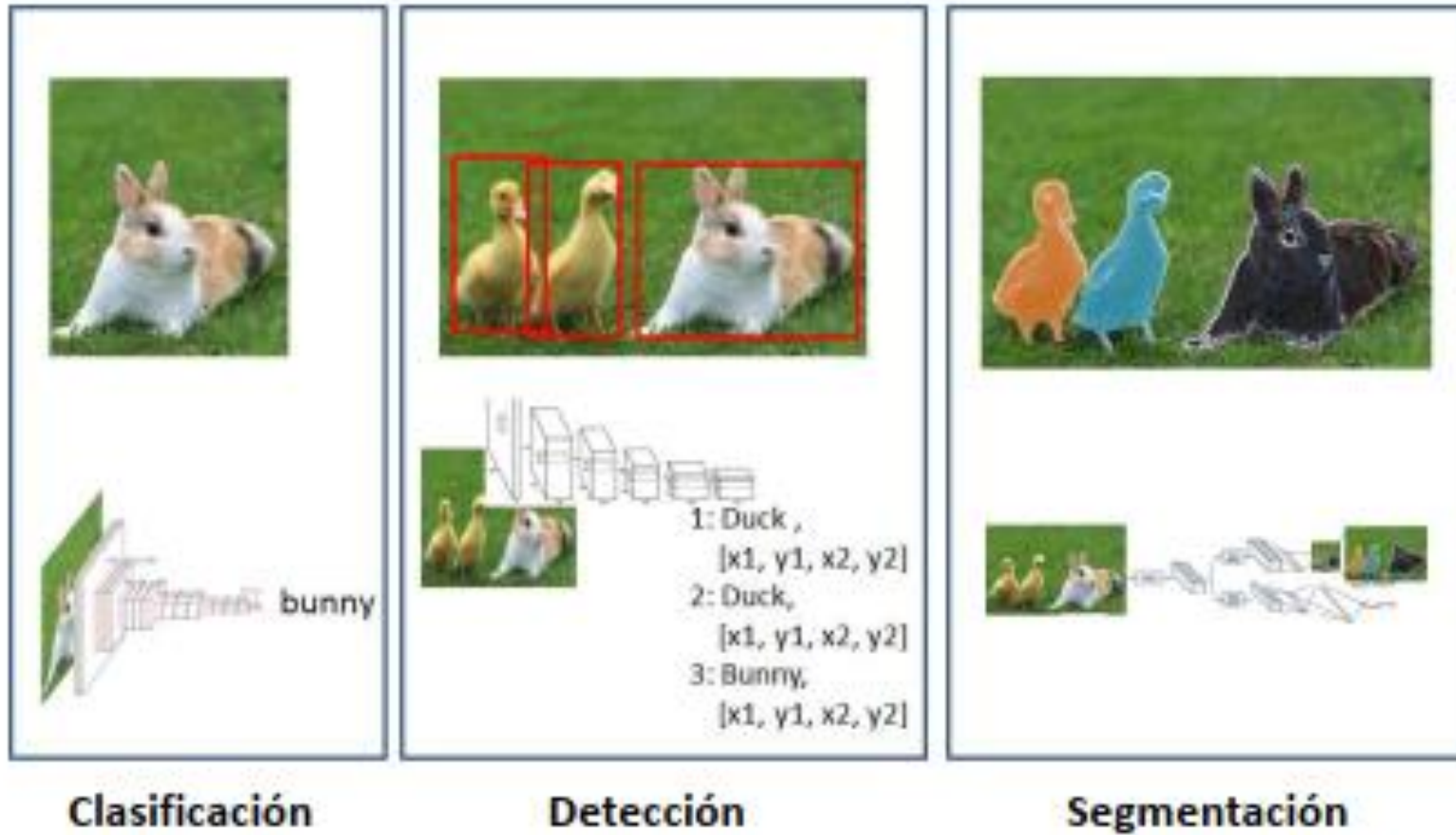
(Fei Fei Li, 2015)

1.1

Aplicaciones de CNN



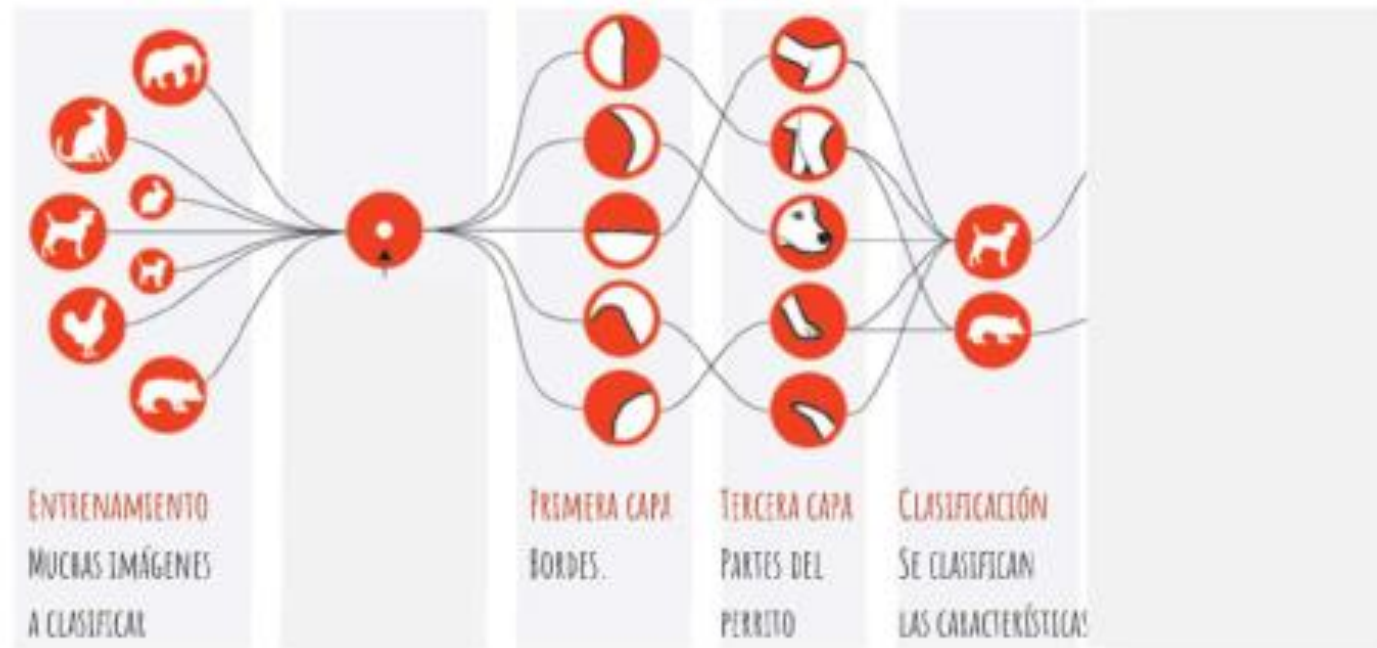
Aplicaciones de CNN



Funcionamiento (CNN)

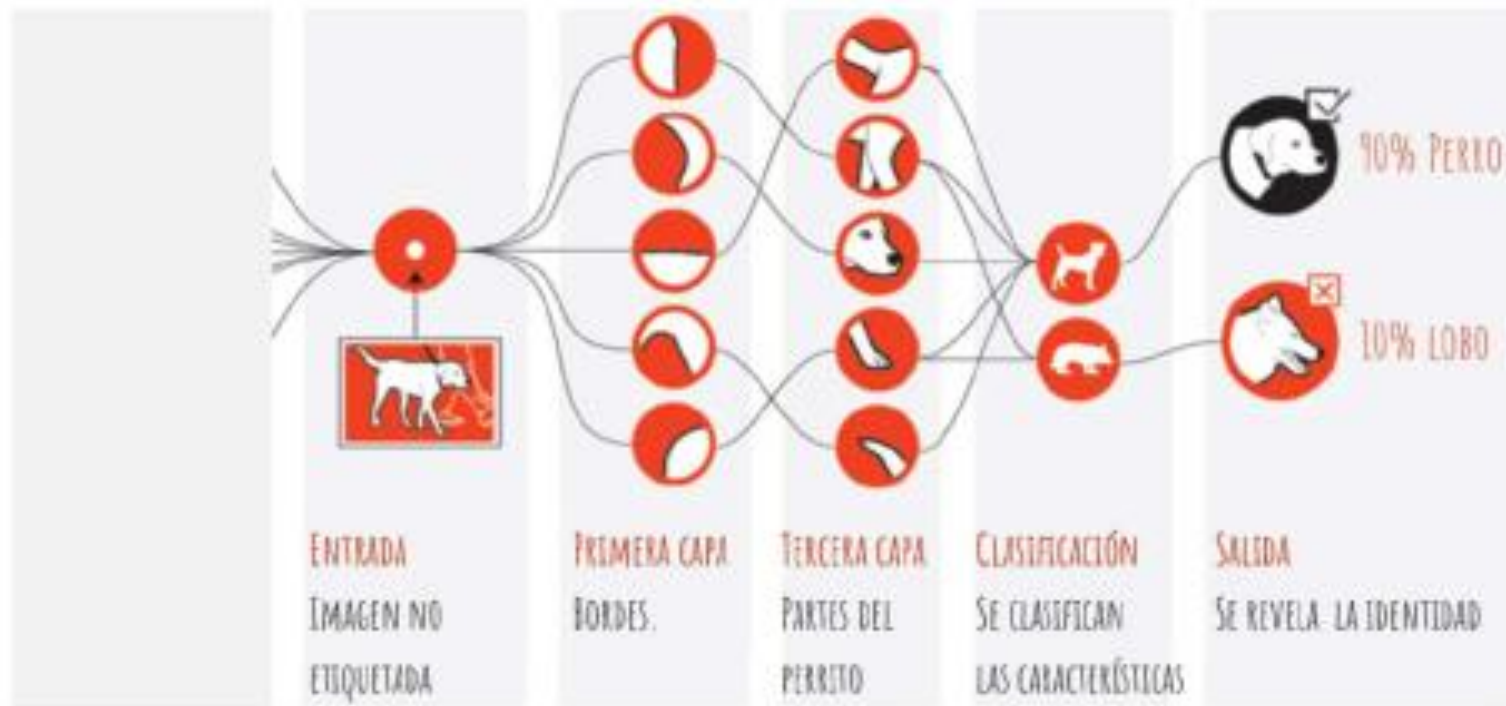
- CNNs son usadas para reconocer, detectar y segmentar objetos y escenas.
- Su uso se ha incrementado por 3 características importantes:
 - Eliminan la necesidad de extraer características (features) manualmente, éstas se aprenden directamente en la CNN.
 - Obtiene altos resultados de reconocimiento state-of-the-art.
 - CNNs se pueden reentrenar para nuevas tareas de reconocimiento, permitiendo el re-uso de redes preexistentes.

¿Qué es una Red neuronal convolucional? (CNN)



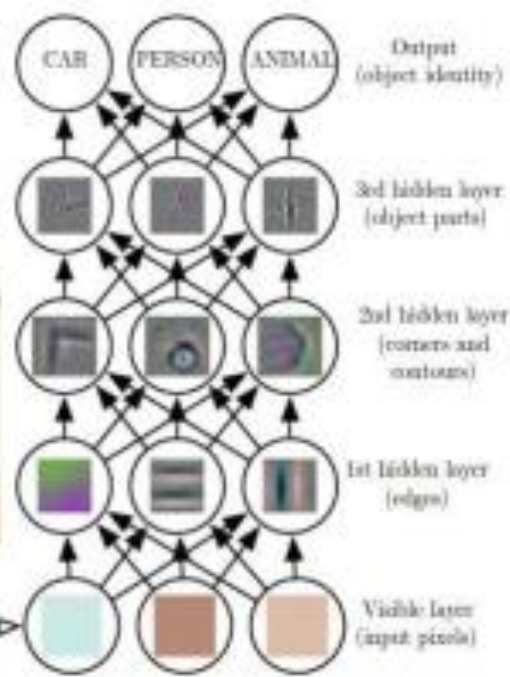
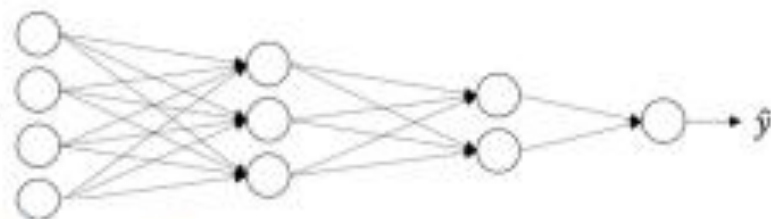
1. Entrenamiento

¿Qué es una Red neuronal convolucional? (CNN)

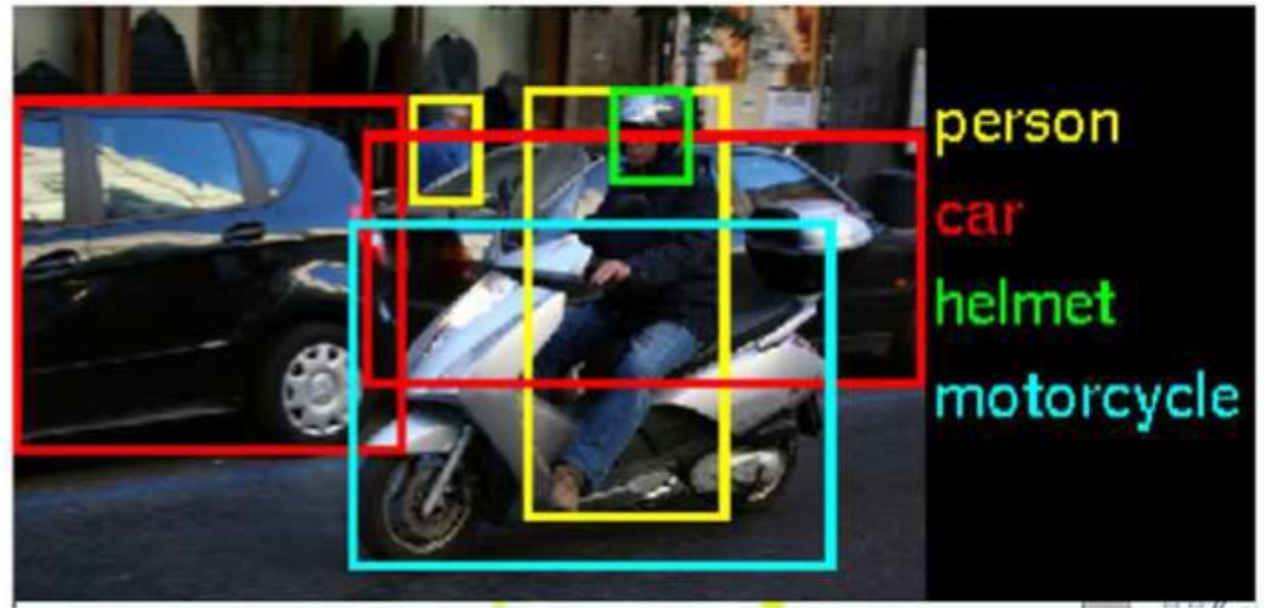
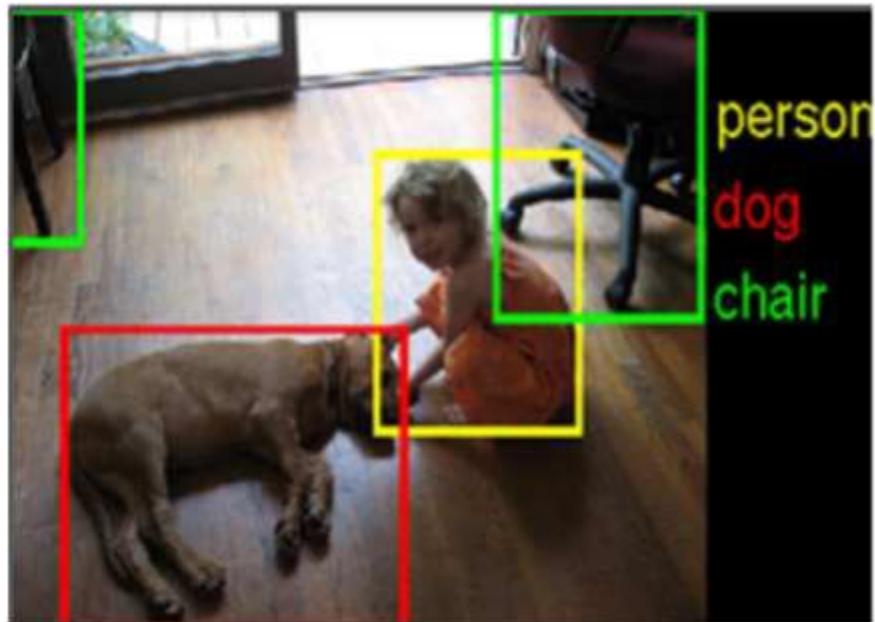


2. Prueba

Red neuronal convolucional



En imágenes



En video



Reconocimiento de acciones



Clasificación



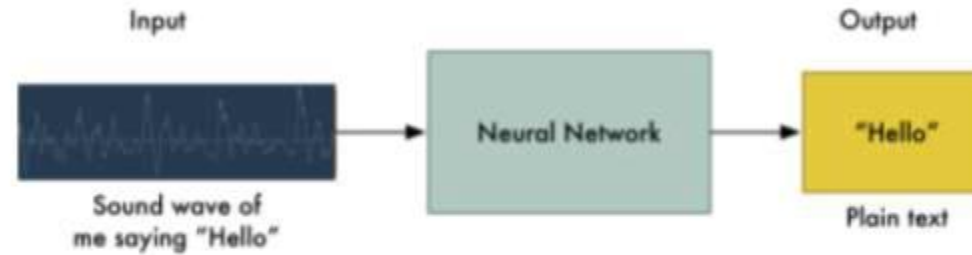
A woman is playing frisbee with a black dog on the grass.

Descripción

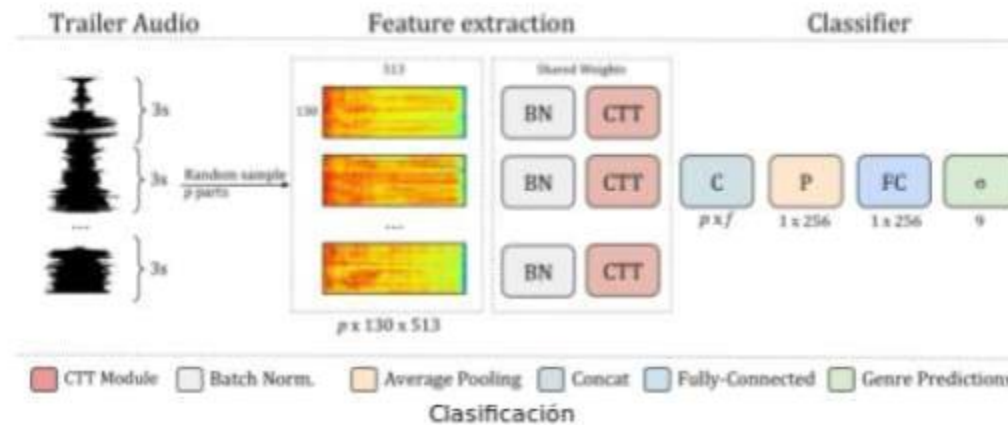
En audio



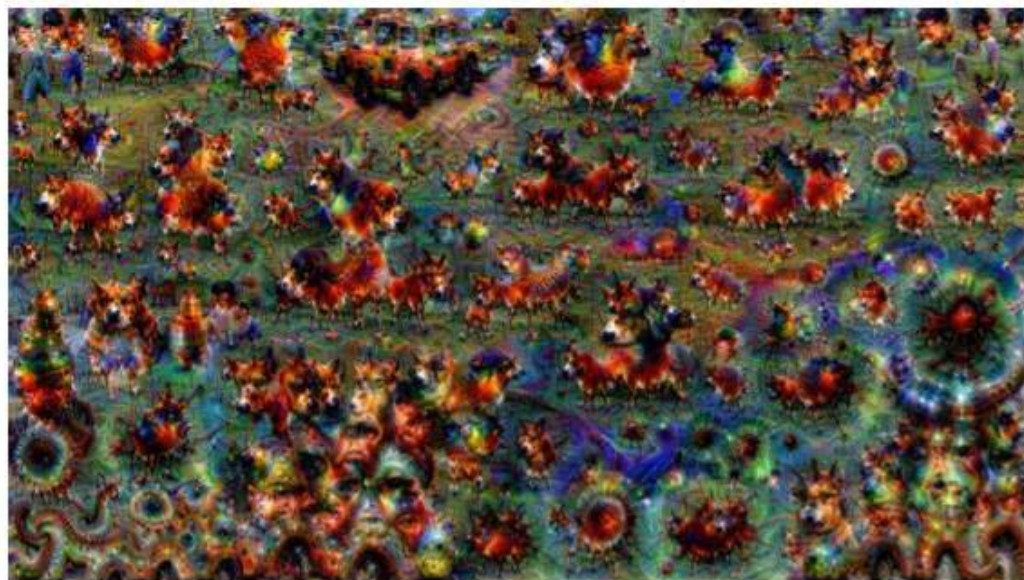
Reconocimiento acústico de escenas



Reconocimiento y síntesis de discurso



Crear conocimiento



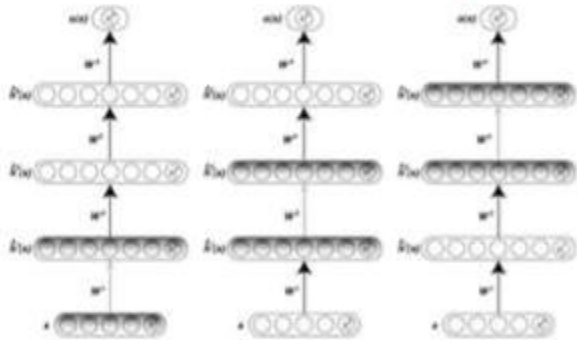
Original image is CC0 public domain
Starry Night and Tree Roots by Van Gogh are in the public domain
Bokeh image is in the public domain
Stylized images copyright Justin Johnson, 2017;

Gatys et al, "Image Style Transfer using Convolutional Neural Networks", CVPR 2016
Gatys et al, "Controlling Perceptual Factors in Neural Style Transfer", CVPR 2017

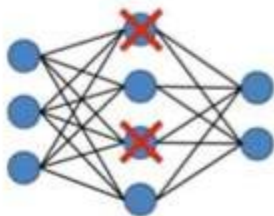
¿Por qué su rápida expansión?

Algorithms

Unsupervised pre-training



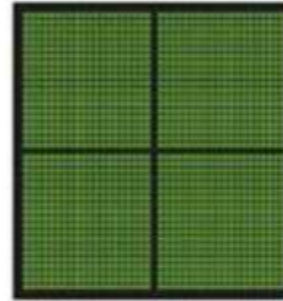
Supervised training for deeper models



Parallel computing



CPU
MULTIPLE CORES



GPU
THOUSANDS OF CORES



NVIDIA® CUDA® 코어 5760
메모리 클럭 7.0 Gbp
표준 메모리 설정 12288 MB

Big Data



IMAGENET Large Scale Visual Recognition Challenge

Steel drum

The Image Classification Challenge:

1,000 object classes

1,431,167 images



Output:

Scale

T-shirt

Steel drum

Drumstick

Mud turtle



Output:

Scale

T-shirt

Giant panda

Drumstick

Mud turtle



ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- Tarea1: Clasificación
- Tarea 2: Clasificación con localización
- Tarea 3: Clasificación Fina (Nueva)
 - En mas de 100+ categoría de perros.
 - Para cada categoría de perros predecir si el perro especificado dentro de una caja (bounding box) en la imagen de prueba es de una categoría particular

▪ The ImageNet Large Scale Visual Recognition Challenge



1,461,406 imágenes de alta resolución

Múltiples anotaciones

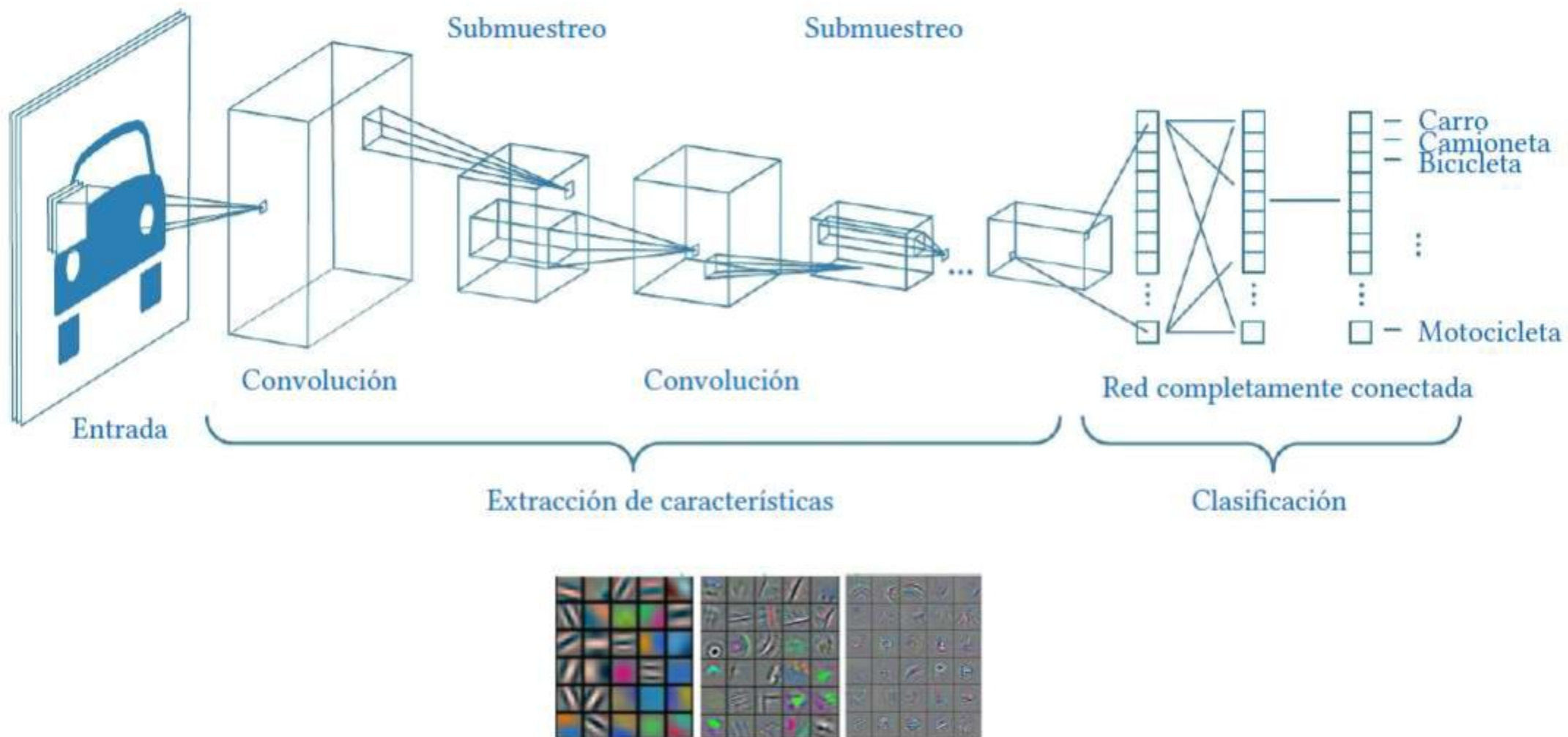
Jerarquía de 100 clases de objetos en diferentes dimensiones

Competencia desde 2010

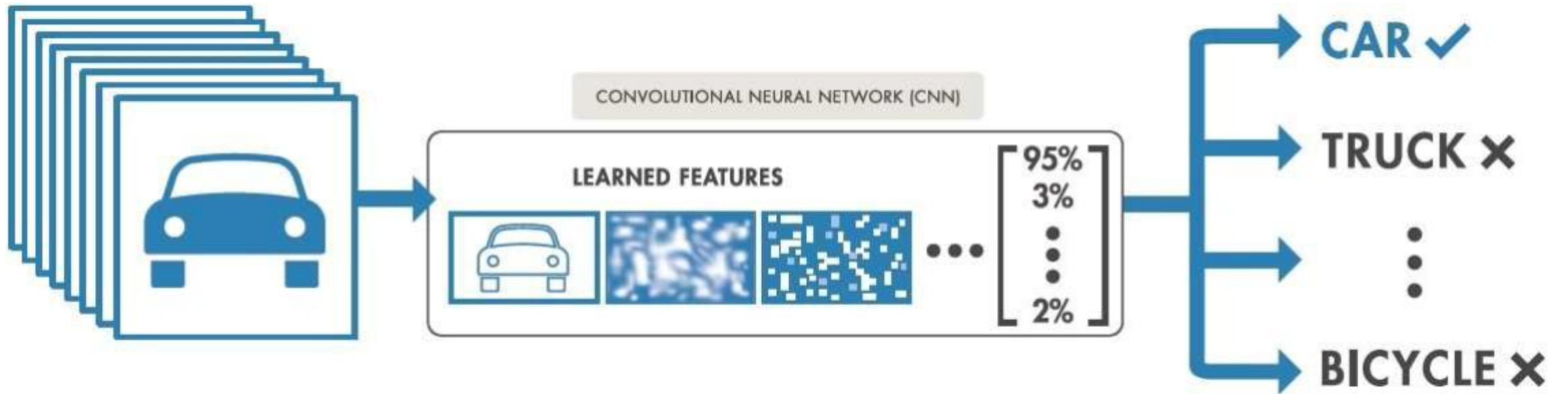
Los equipos de investigación evalúan sus algoritmos

[Figures from www.mvdlia.com]

Arquitectura general de una CNN

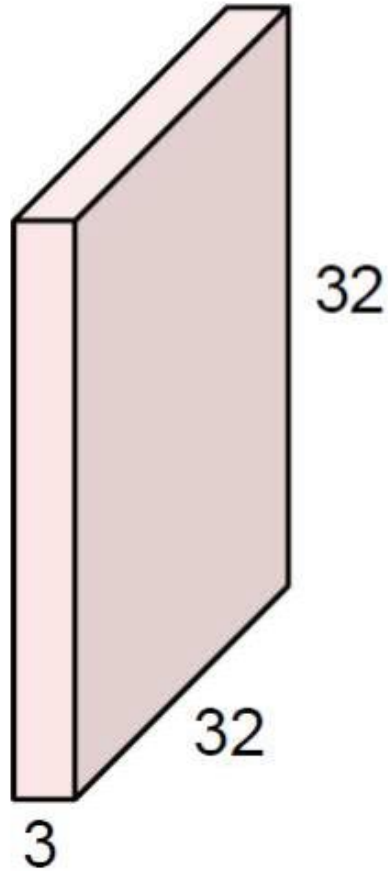


Arquitectura general de una CNN



Convolution Layer

32x32x3 image



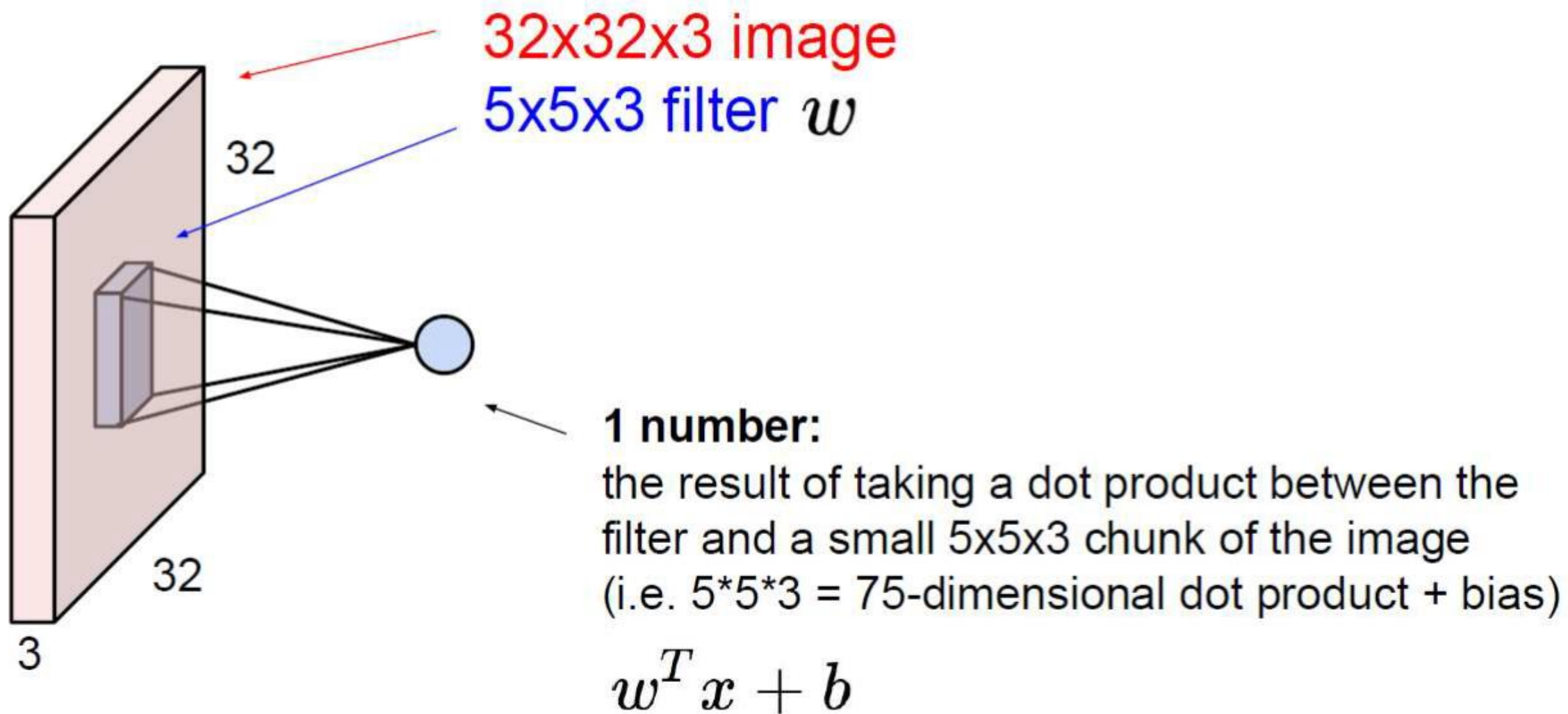
Filters always extend the full depth of the input volume

5x5x3 filter



Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Convolution Layer



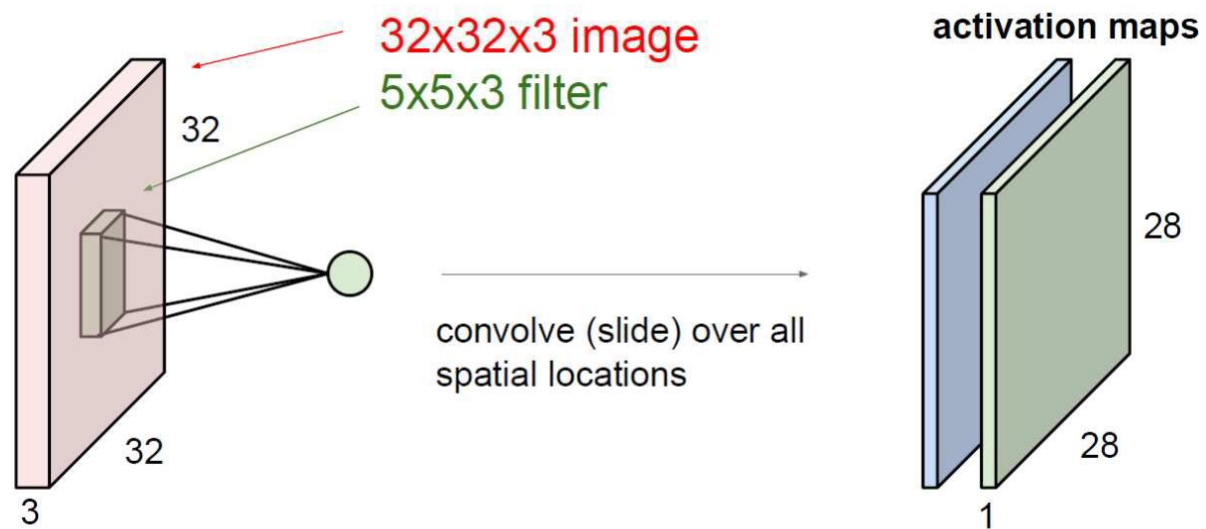
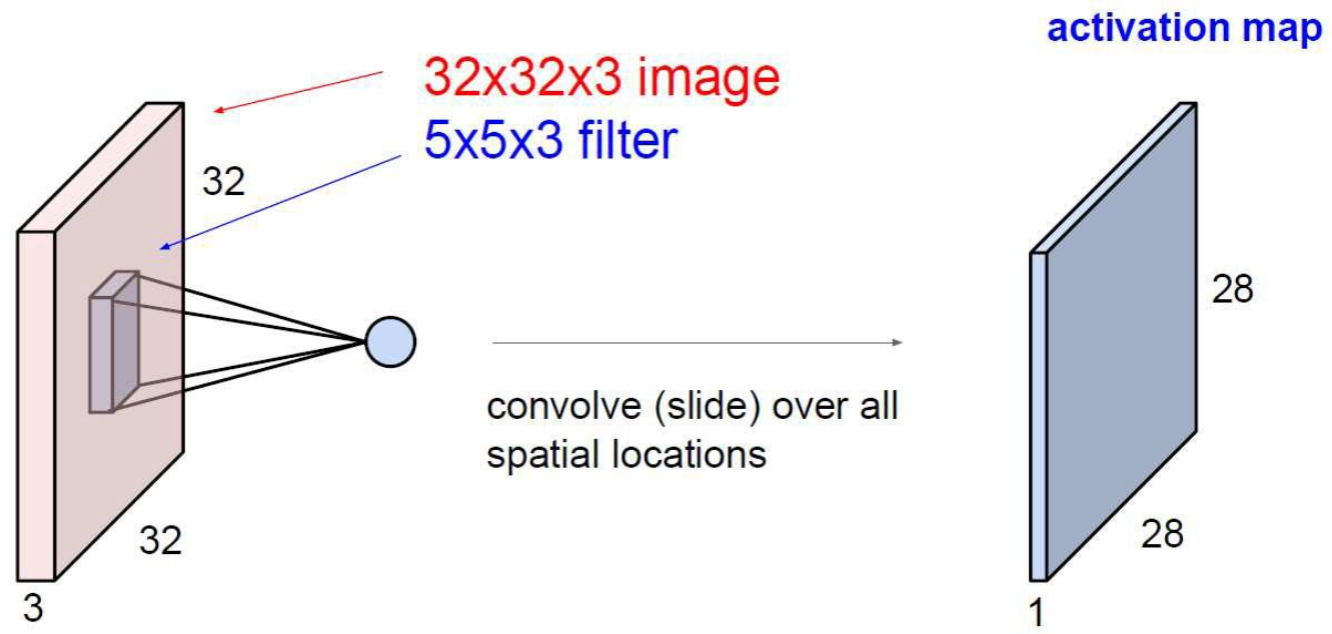
Tamaño final de la operacion

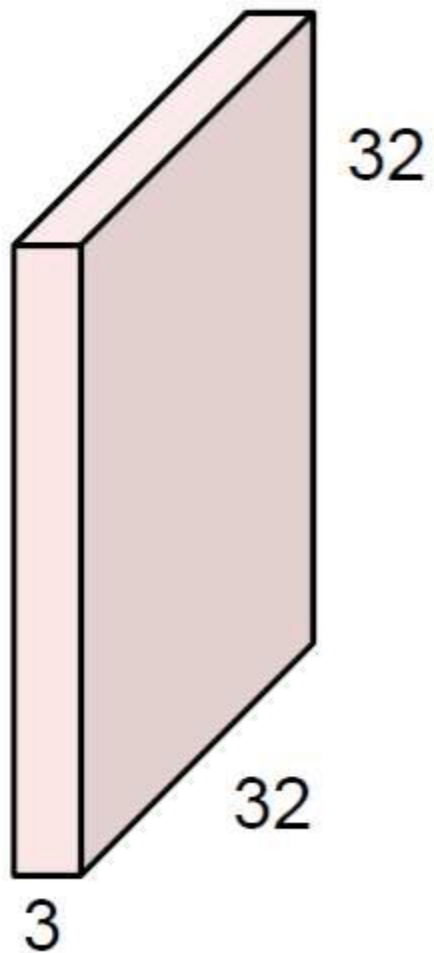
- $M-P+1$

- 7×7

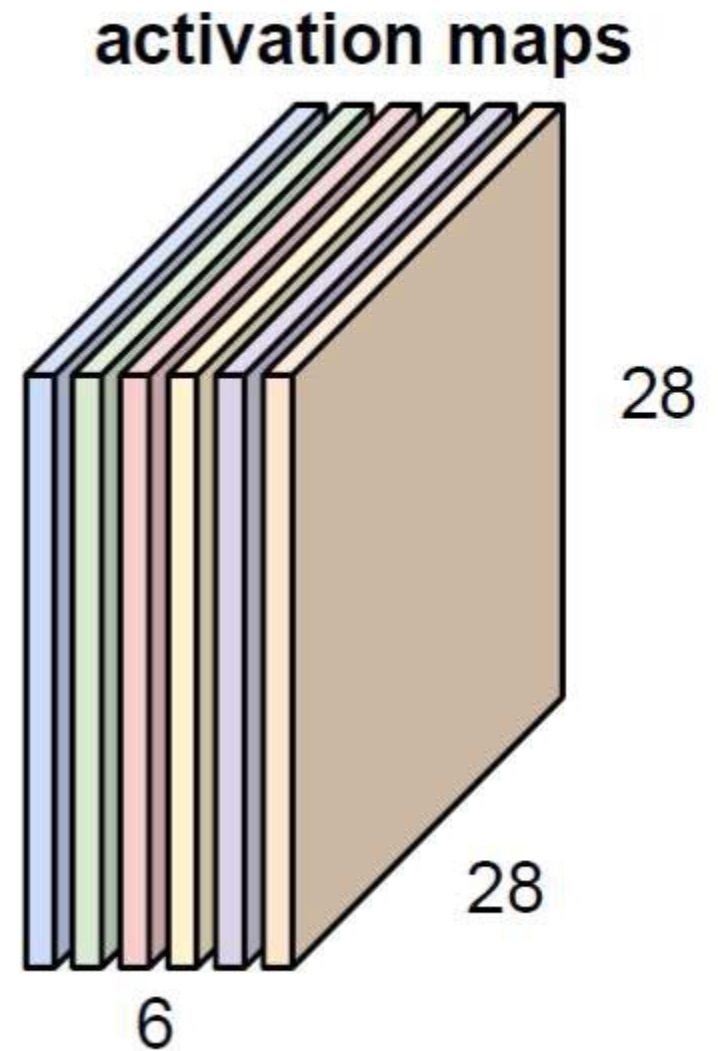
- 3×3

- $7-3+1=5 \rightarrow 5 \times 5$

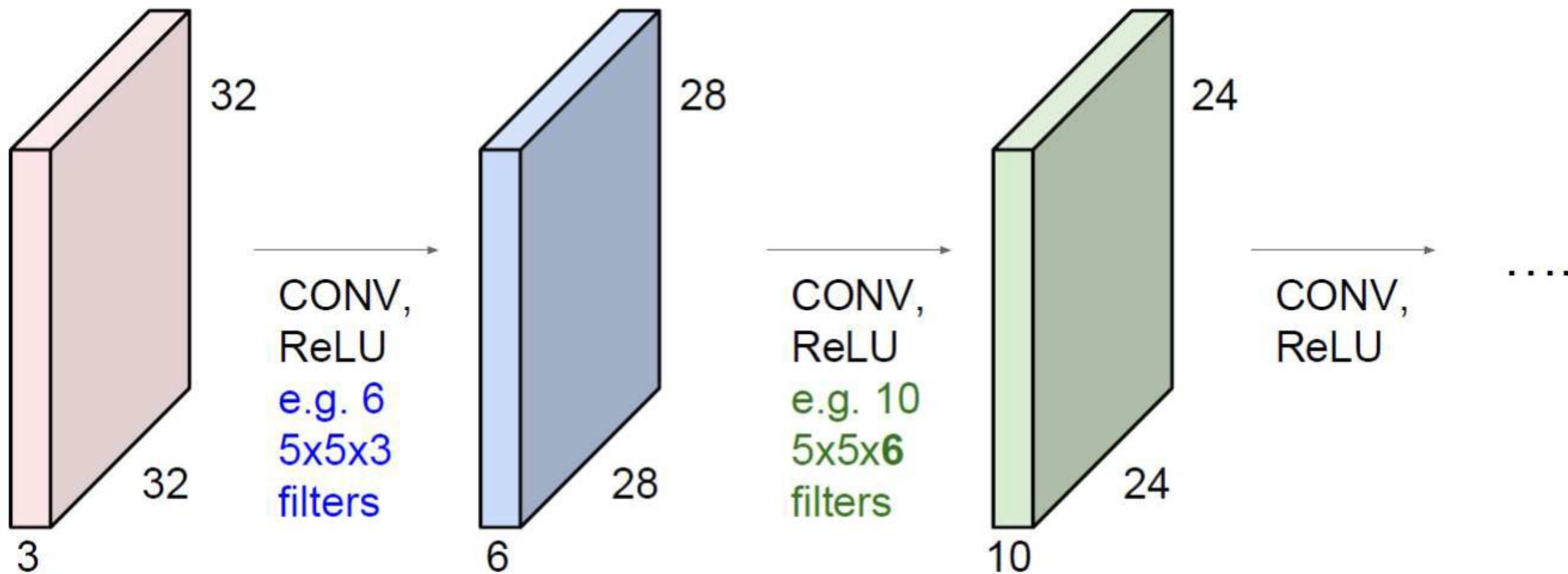




→
Convolution Layer

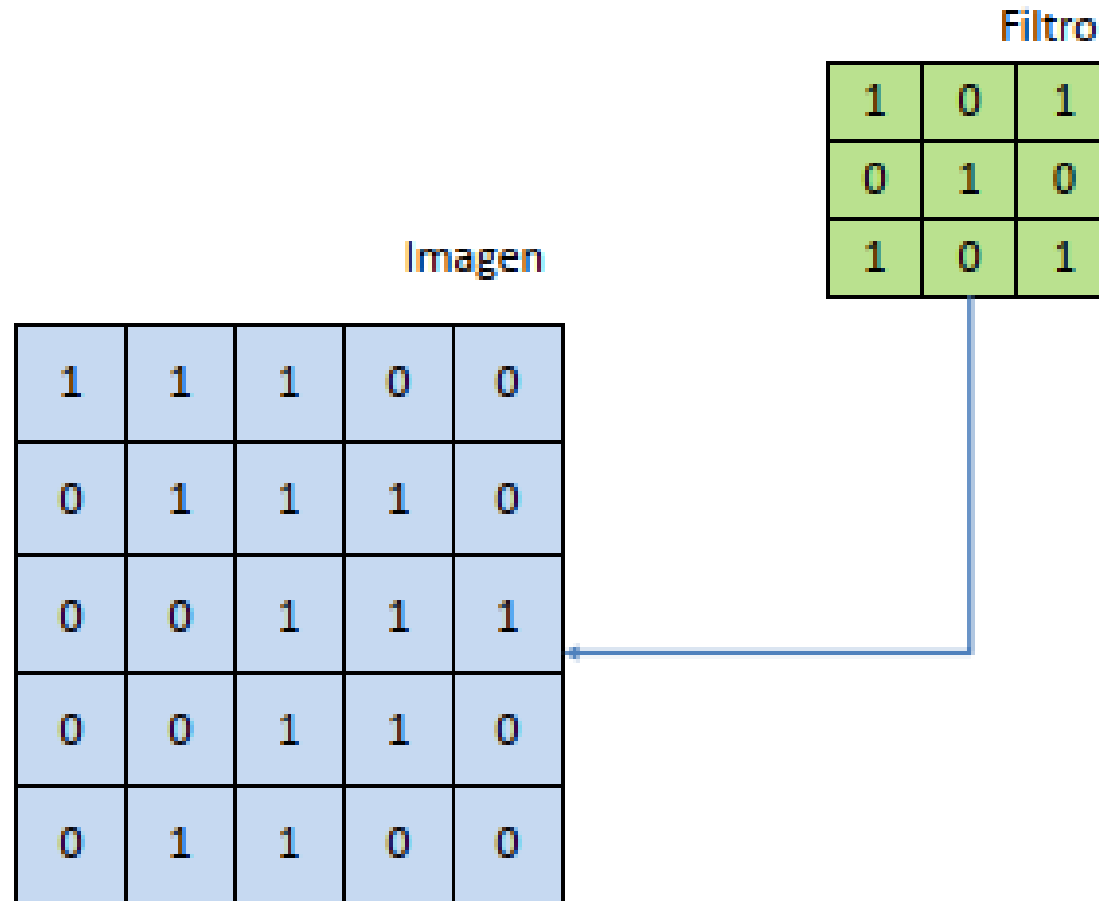


¡Muchas convoluciones en paralelo!
Obtener gran cantidad de características



Procesamiento multicanal

Algunos aspectos



Algunos aspectos-Paso (Stride)

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

Reduce cálculo computacional

Zero Padding

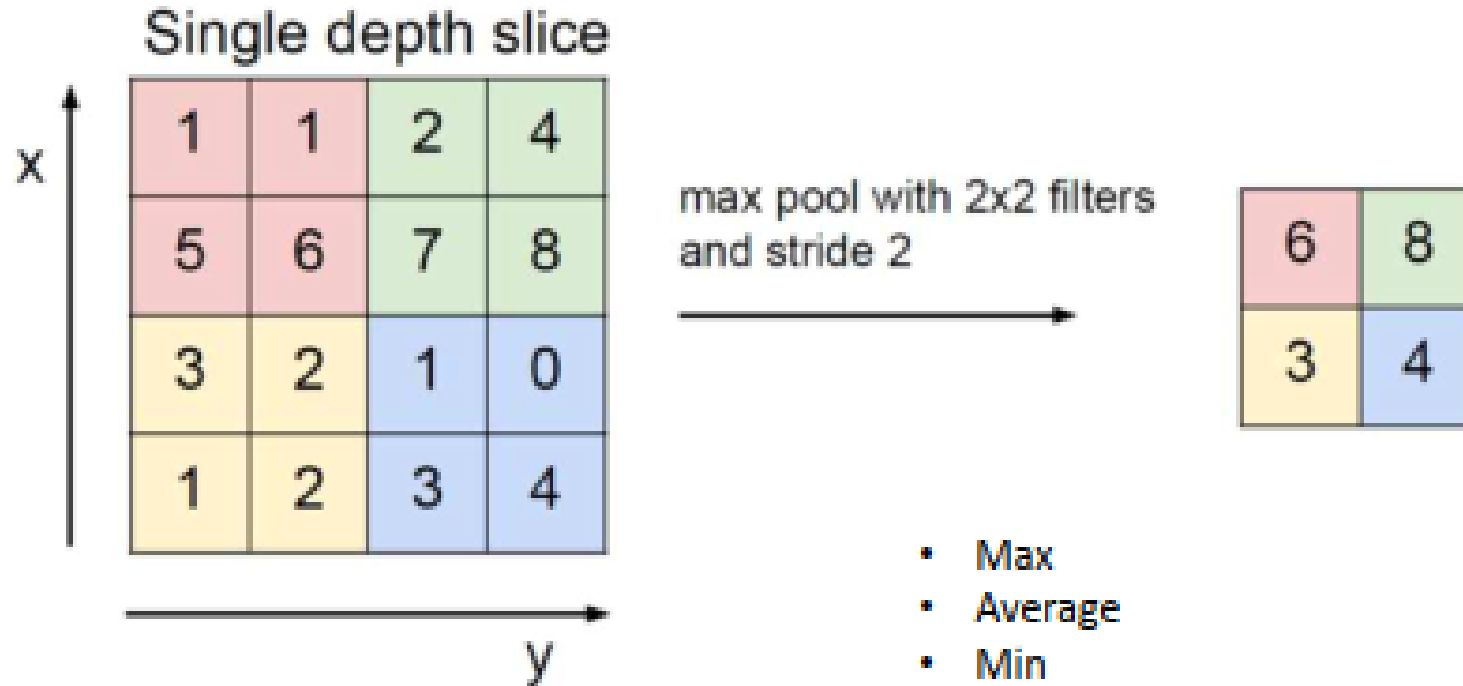
0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Kernel

0	-1	0
-1	5	-1
0	-1	0

114				

MAX POOLING



- Max
- Average
- Min
- Criterio de maxima energia
- Informacion redundante

Hiperparámetros

- Número de filtros K
- Tamaño del filtro 3x3, 5x5, 7x7
- Stride: 1,2 ..
- Pooling
- Padding (zero padding)

Valores Comunes

Cantidad de filtros por capa:

$K = 2^n$ ej: 32, 64, 128

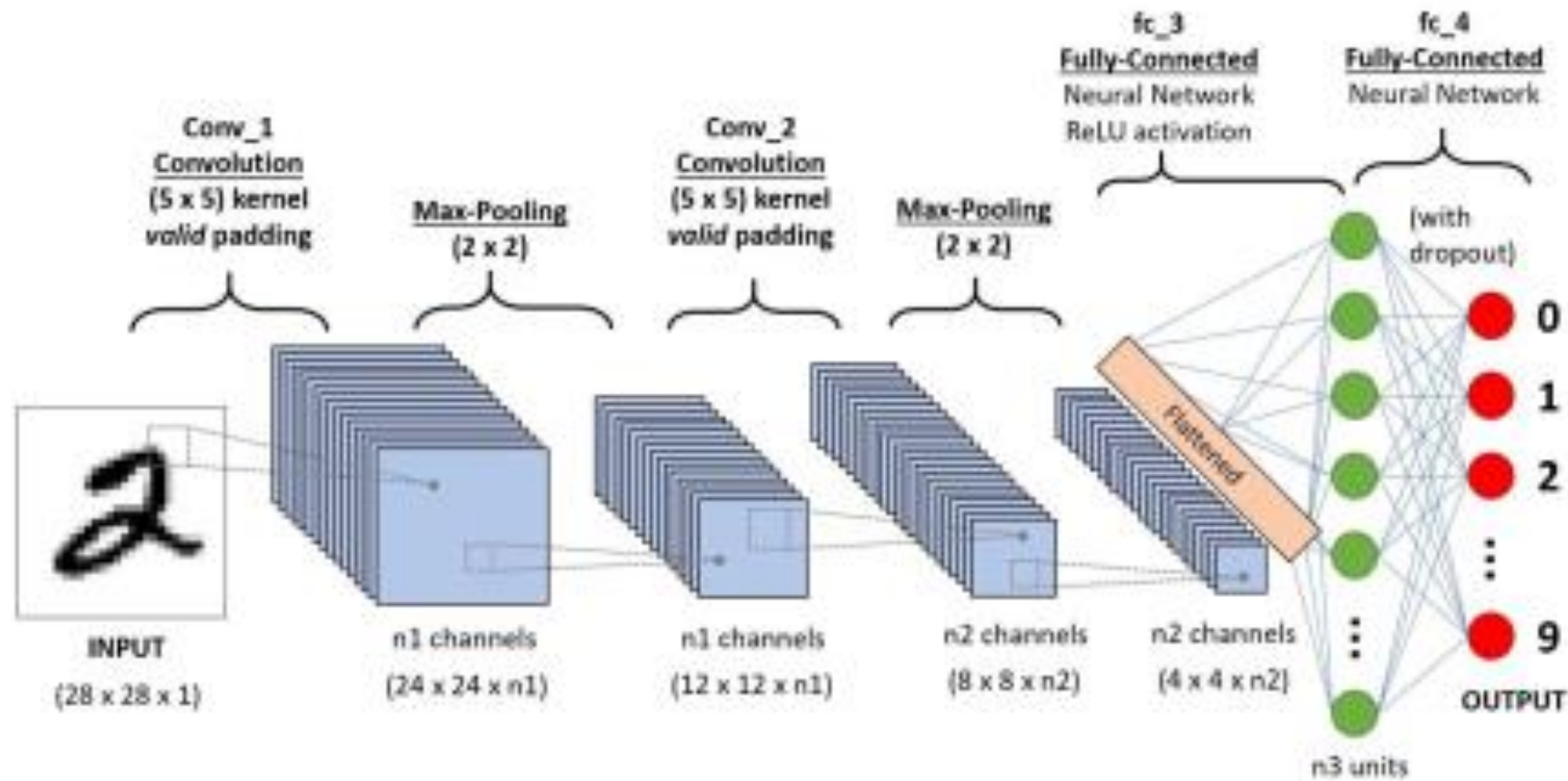
$F = 3, S = 1, P = 1$

$F = 5, S = 1, P = 2$

$F = 5, S = 2, P = ?$ (lo que se ajuste)

$F = 1, S = 1, P = 0$

Red Neuronal Convolutcional



<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Entrada a una red Densa (totalmente conectada)

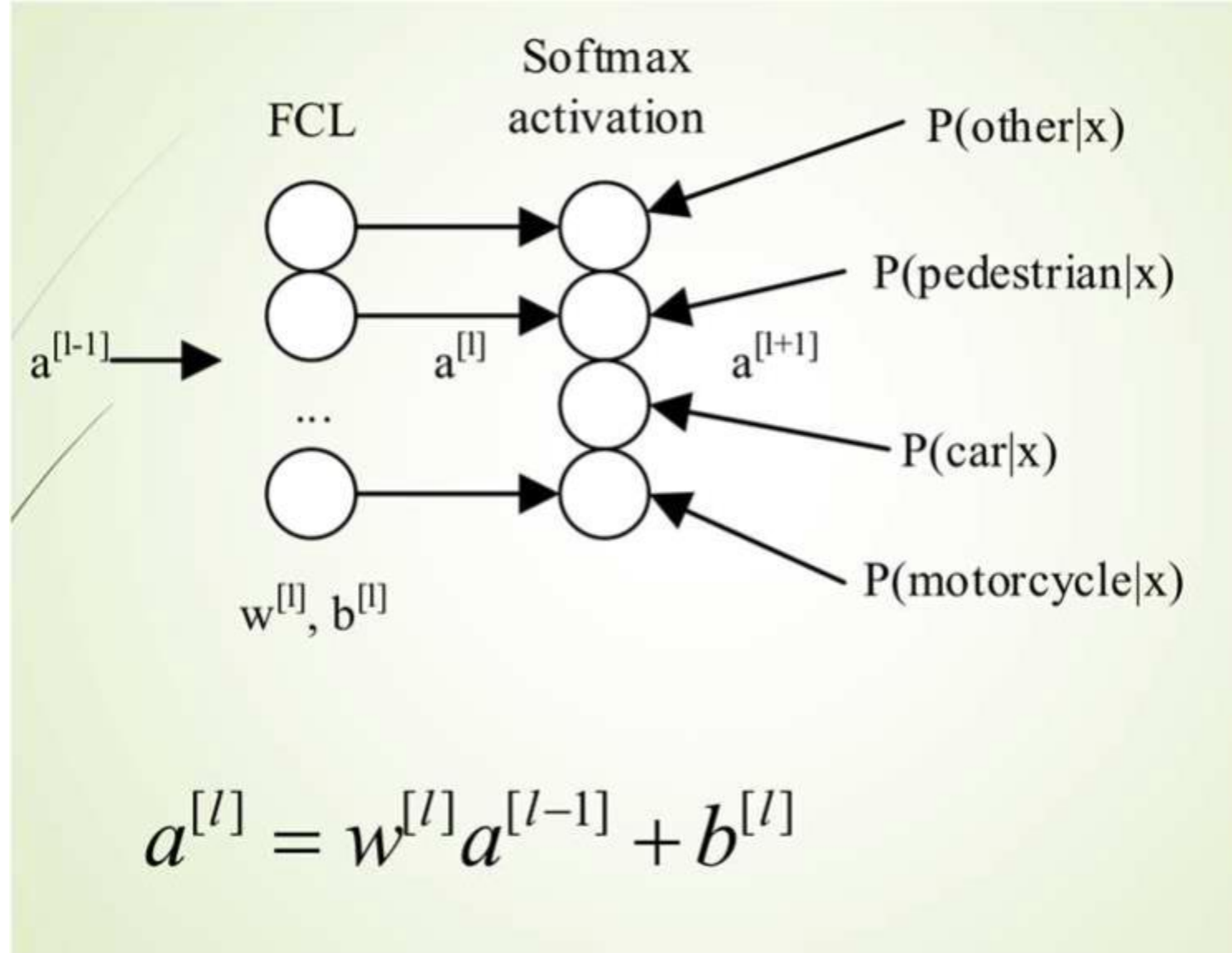
1	1	0
4	2	1
0	2	1

Mapa de Activación

Aplanado

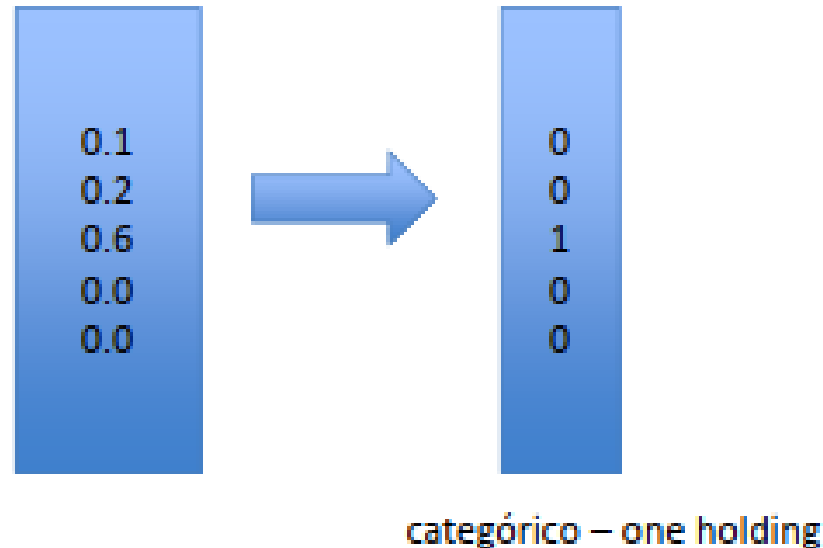
1
1
0
4
2
1
0
2
1

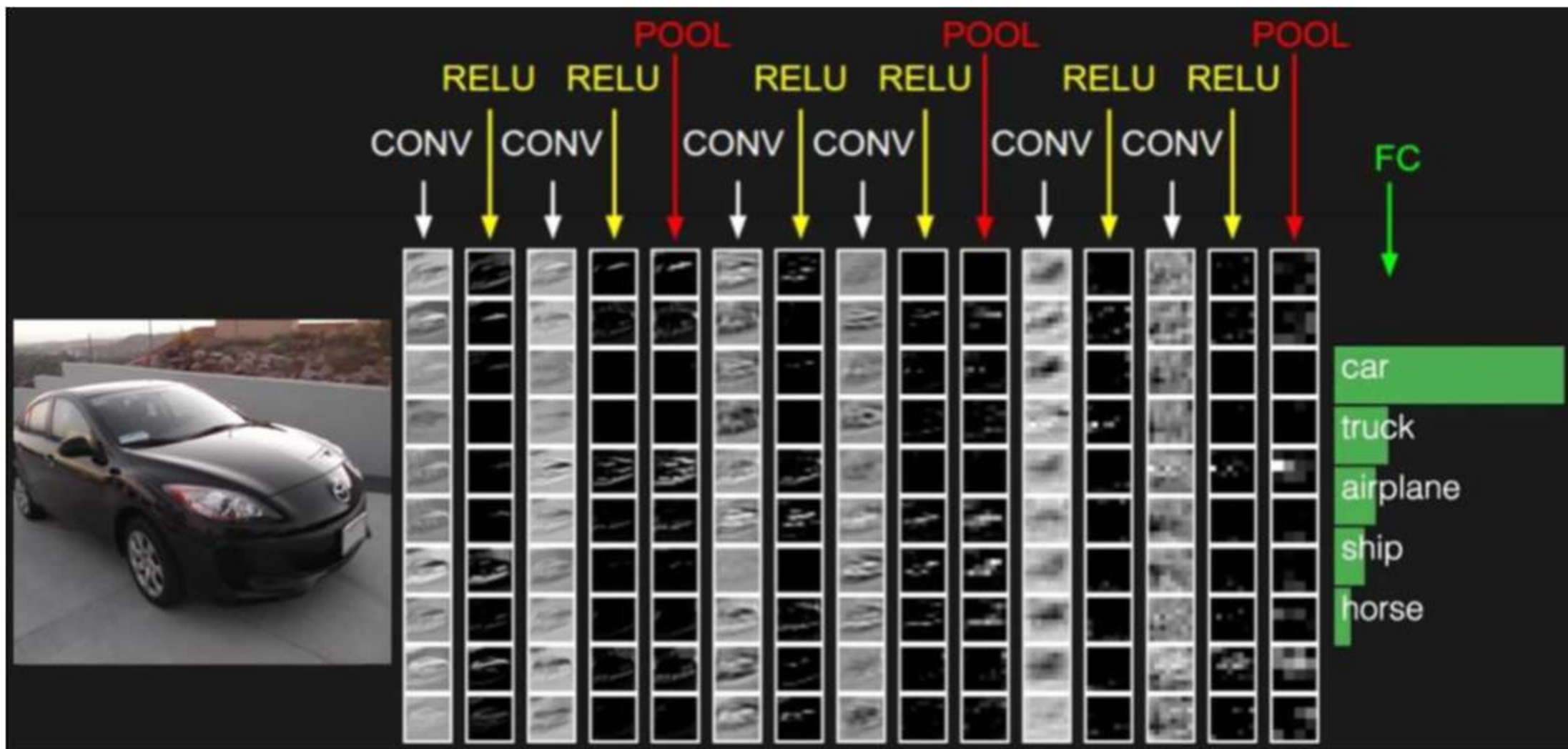
Por ejemplo: Clasificador multicategórico (Softmax)



Clasificador multicategórico (Softmax)

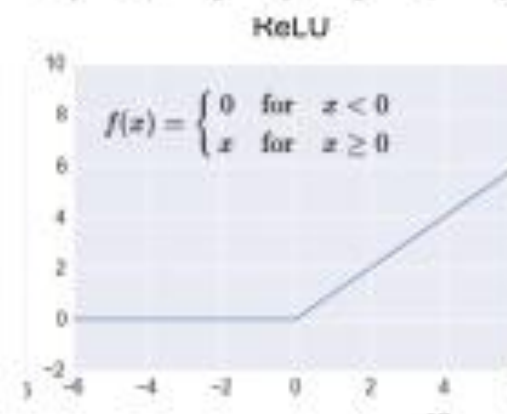
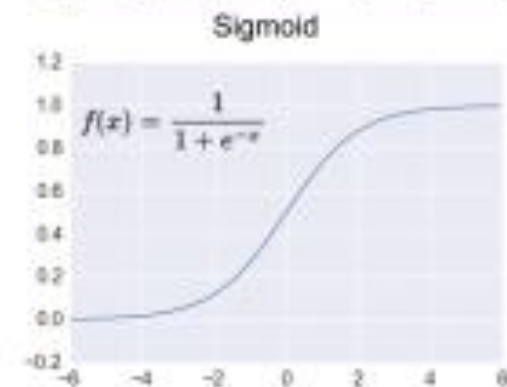
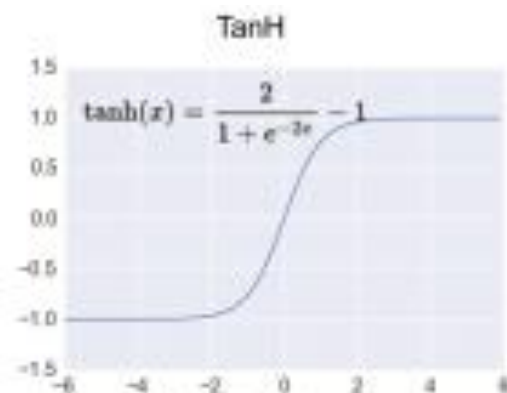
- Regresa un vector de probabilidades
- Tomando el valor máximo para la clasificación final.



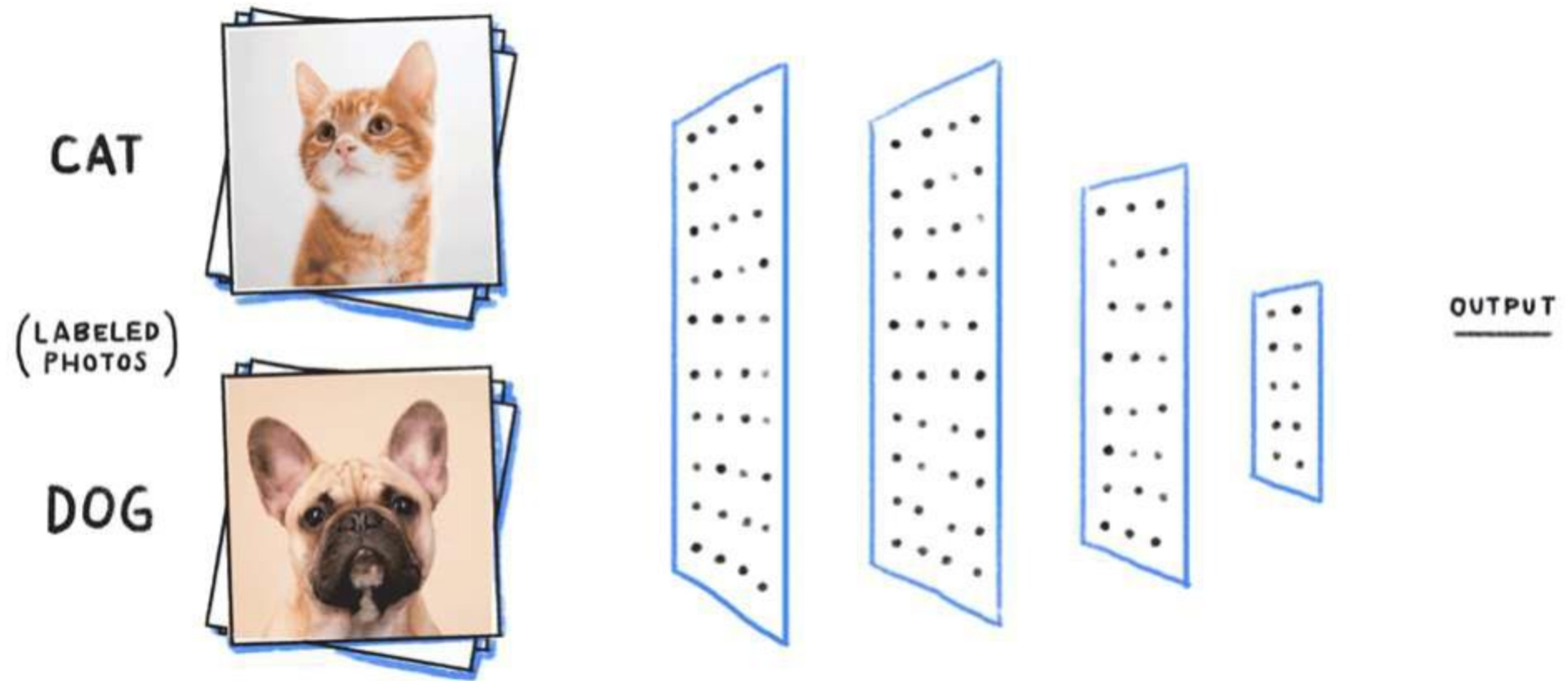


Funciones de activación

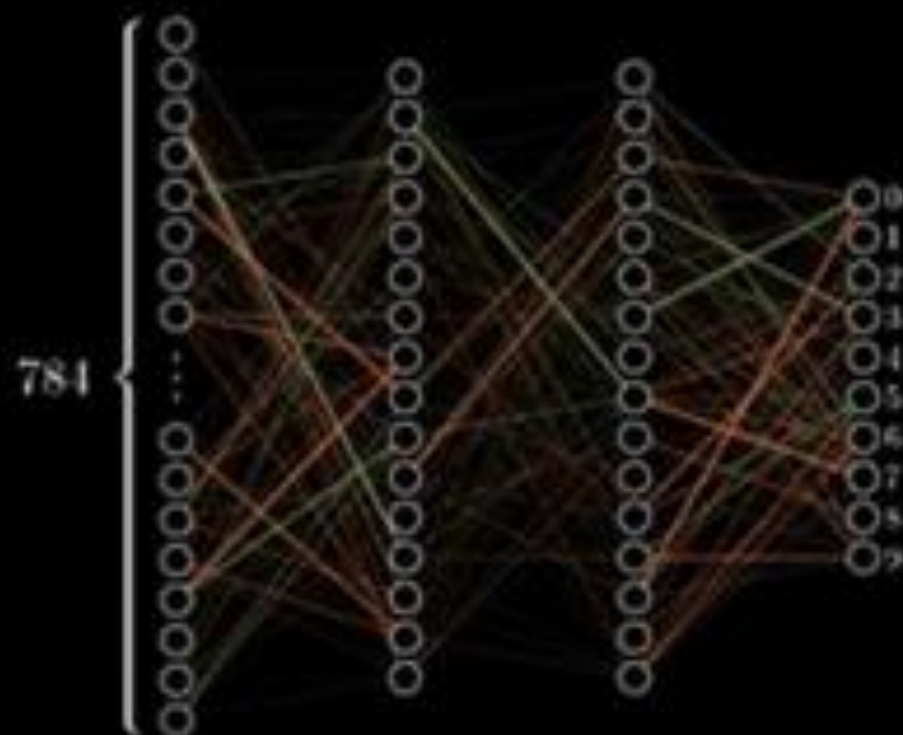
- Tangente hiperbólica (TanH)
- Sigmoide
- Rectified linear unit (ReLU) – más preferida
 - Simplifica la retropropagación (backpropagation)
 - Acelera el aprendizaje
 - Evita problemas de saturación
 - Una capa de ReLU aplicará una función de activación elemento a elemento, tal como $\text{Max}(0, x)$ (umbral en el cero)
- LeakyReLU



¿Entrenamiento hacia adelante?

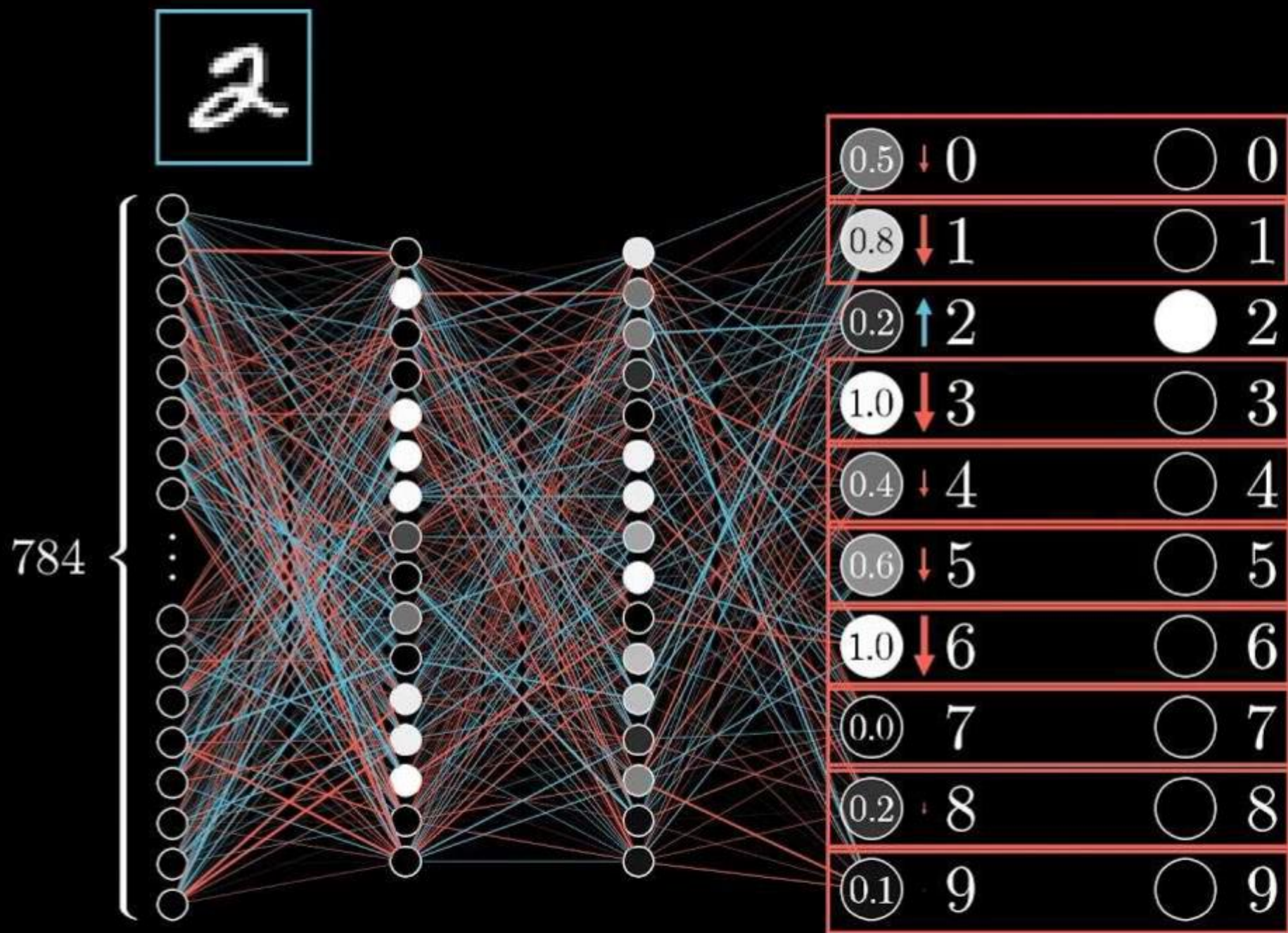


Entrenamiento en progreso

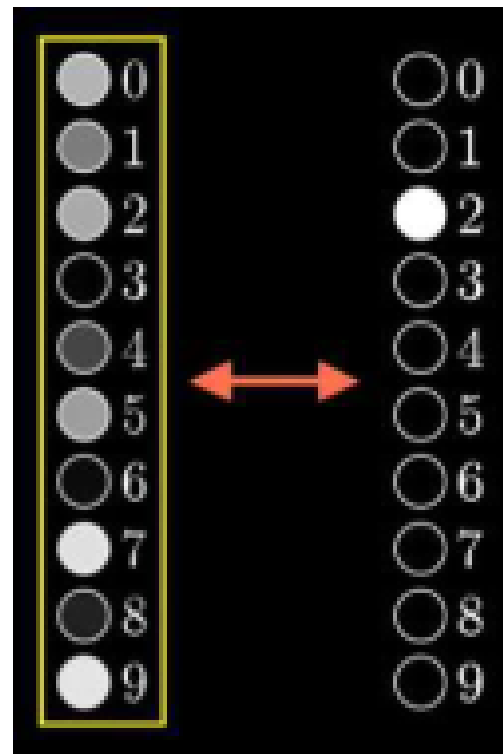


3BLUE1BROWN SERIES- But what "is" a Neural Network?

https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi



Entrenamiento



Salida

Salida
deseada

Average cost of
all training data...

Cost of 2

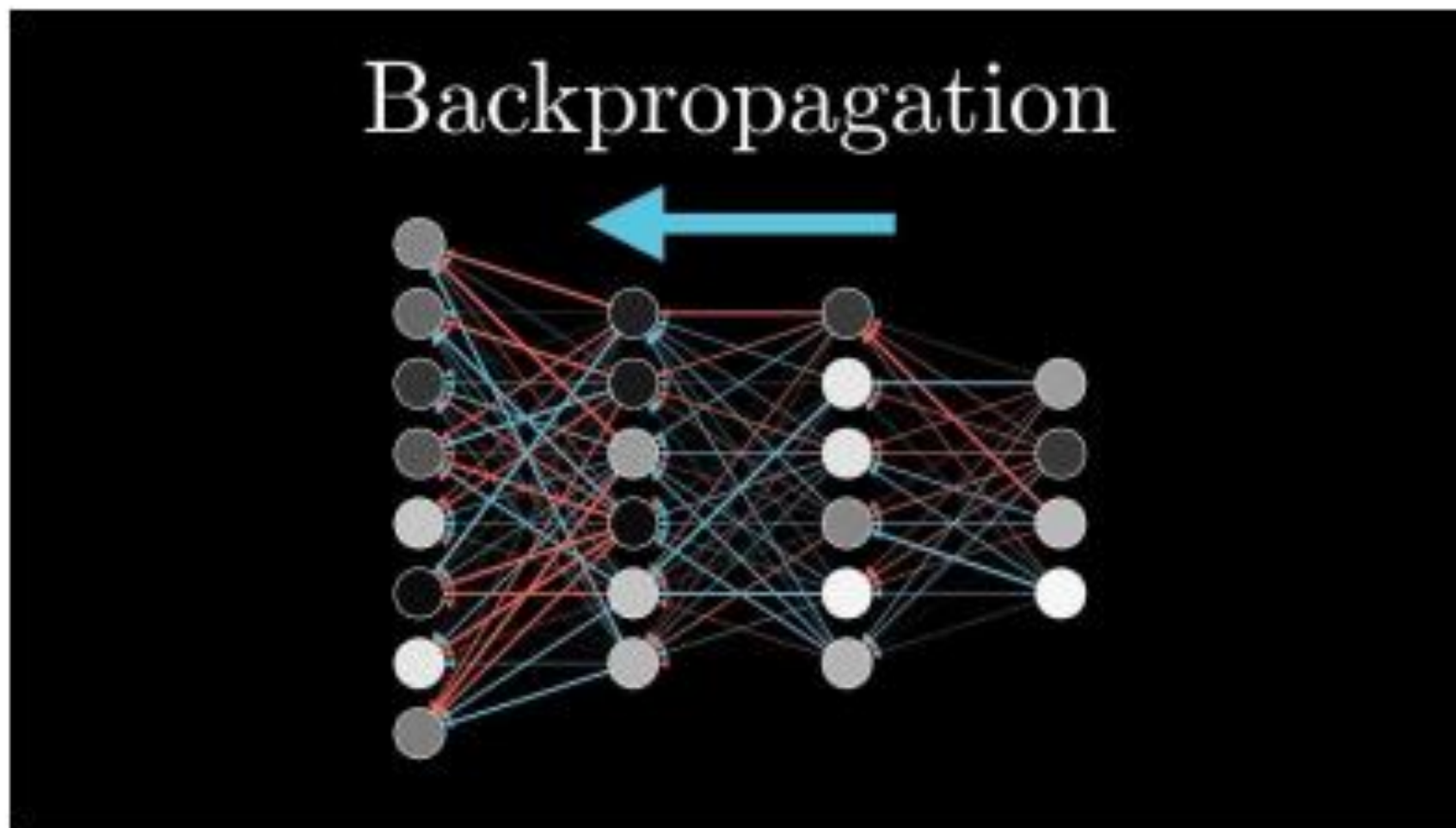
$$\left\{ \begin{array}{l} (0.71 - 0.00)^2 + \\ (0.49 - 0.00)^2 + \\ (0.67 - 1.00)^2 + \\ (0.02 - 0.00)^2 + \\ (0.27 - 0.00)^2 + \\ (0.62 - 0.00)^2 + \\ (0.05 - 0.00)^2 + \\ (0.88 - 0.00)^2 + \\ (0.14 - 0.00)^2 + \\ (0.90 - 0.00)^2 \end{array} \right.$$

Error o "función de coste"

Como hacemos para que el valor sea mas pequeño

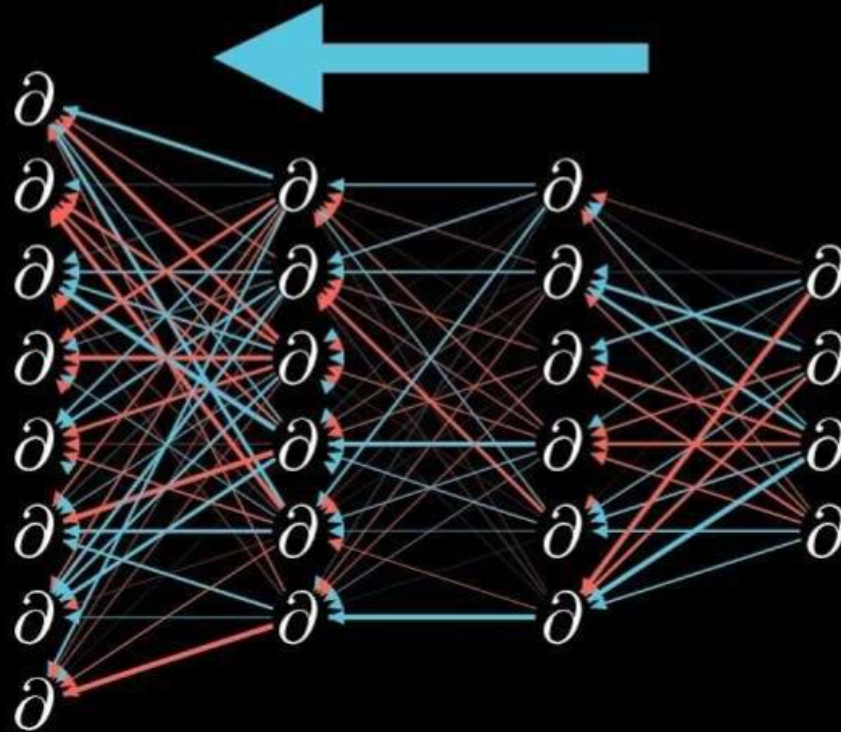
Retropropagación

Gradiente descendiente $W^{t+1} = W^t - \eta * \nabla(\text{función error})$



Entrenamiento Retropropagación

Backpropagation
calculus



Parámetros en la retropropagación

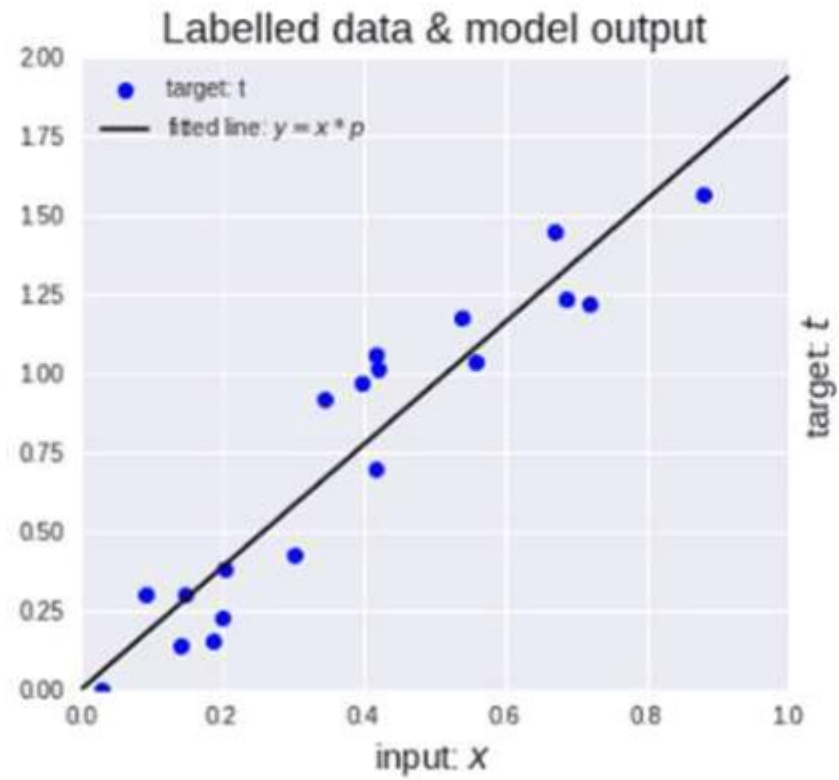
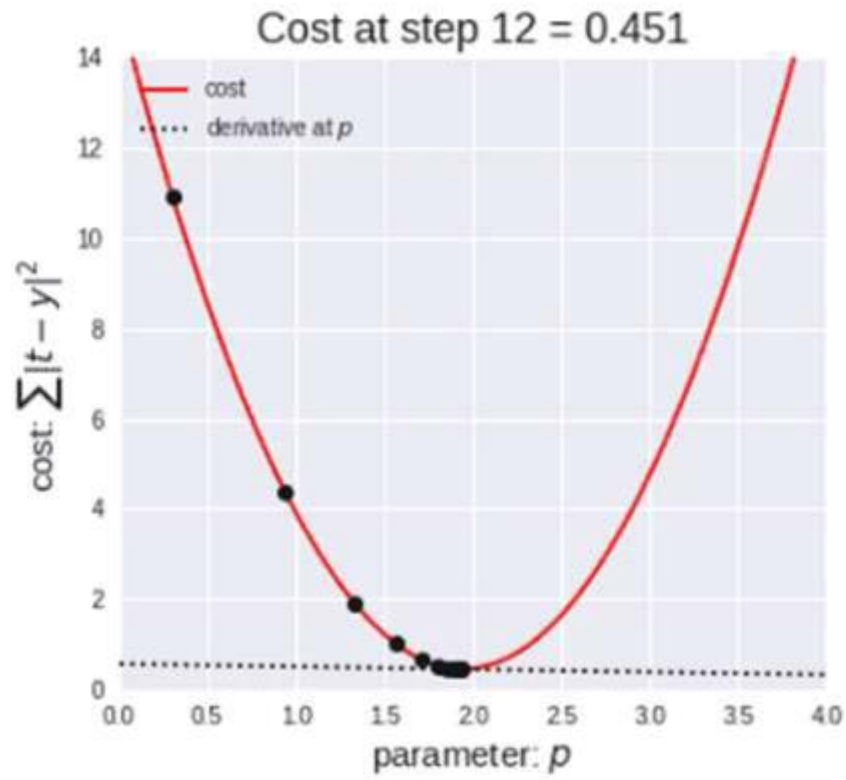
Ecuación de gradiente con la tasa de aprendizaje

$$W^{t+1} = W^t - \eta * \nabla \alpha$$

$$\nabla \alpha = \left[\frac{\partial \alpha}{\partial W_1}, \frac{\partial \alpha}{\partial W_2}, \frac{\partial \alpha}{\partial W_3}, \dots \right]$$

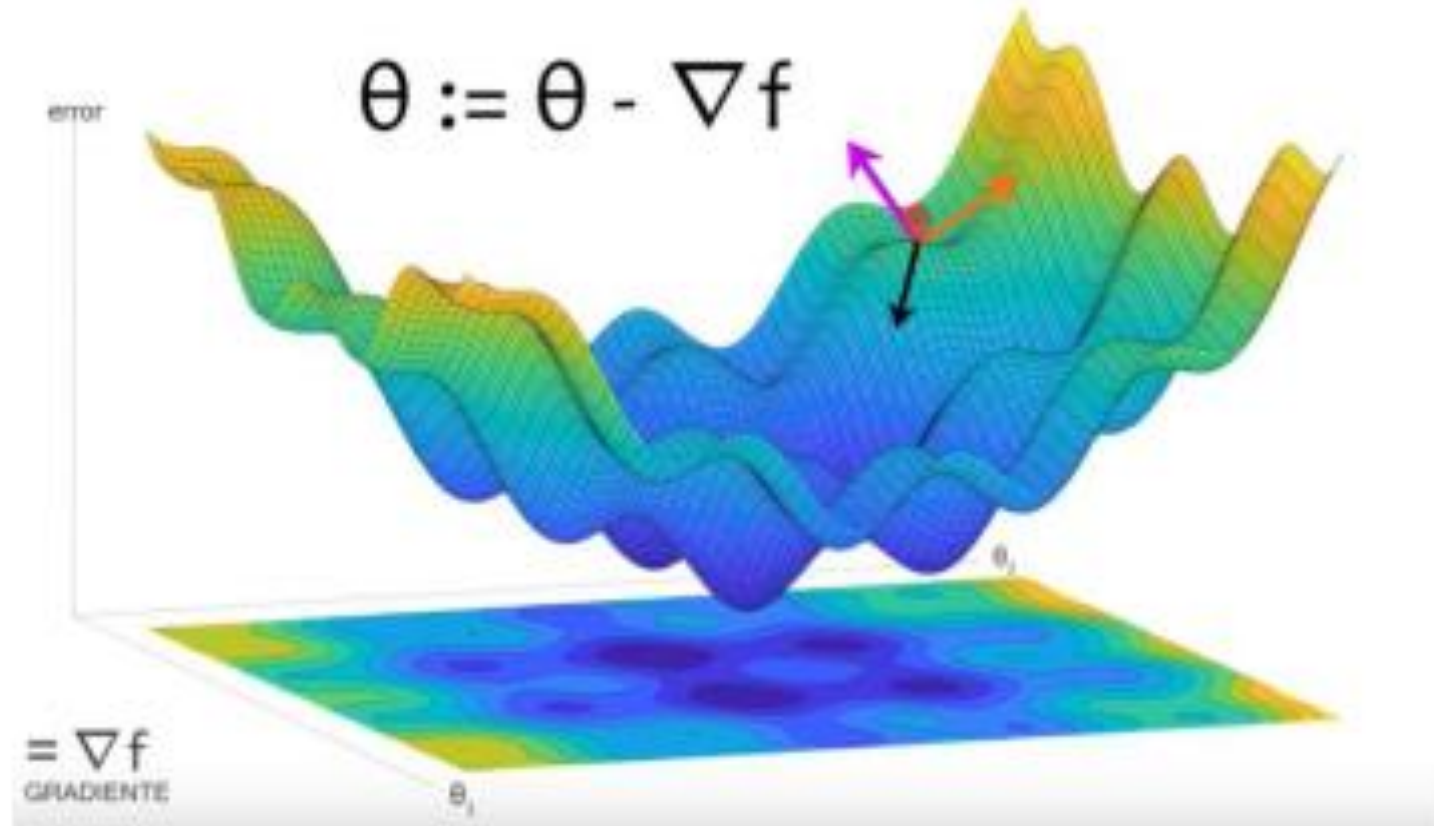
- Tasa de aprendizaje $\eta = 0.001$ a 0.1

Función de Error (Costo)



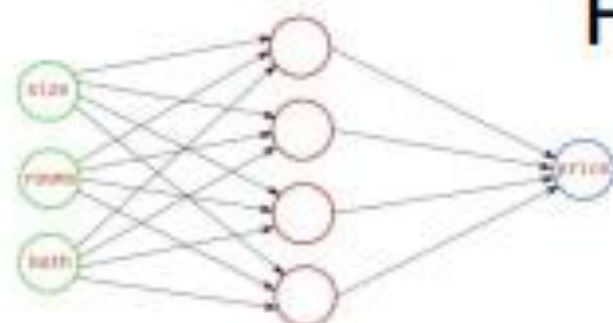
Descenso del gradiente

- Encontrar el valor de los pesos \mathcal{W} 's que hacen mínimo el error



Tasa de aprendizaje: <http://www.benfrederickson.com/numerical-optimization/>

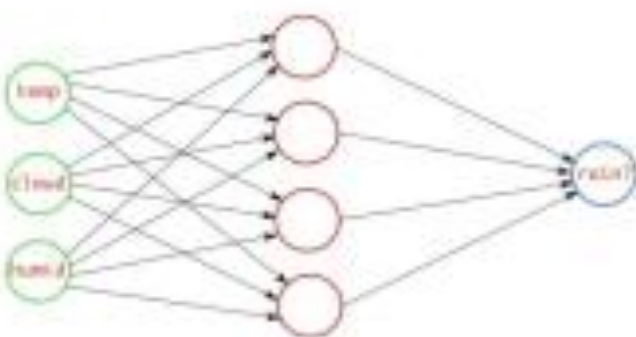
Funciones de pérdida



Error cuadrático medio
Tareas de regresión

$$E = \frac{1}{2} \sum_j (t_j - y_j)^2$$

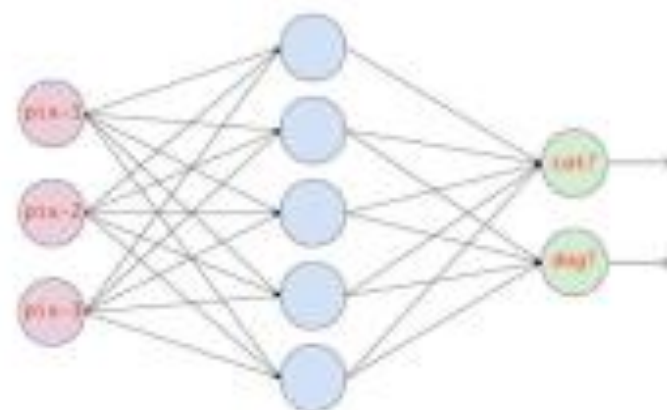
Entropía cruzada binaria



Tareas de clasificación binaria.
Salida pasa por una sigmoide entre
0 y 1

$$E = - \sum_j t_j \log y_j + (1 - t_j) \log(1 - y_j)$$

Entropía cruzada categórica



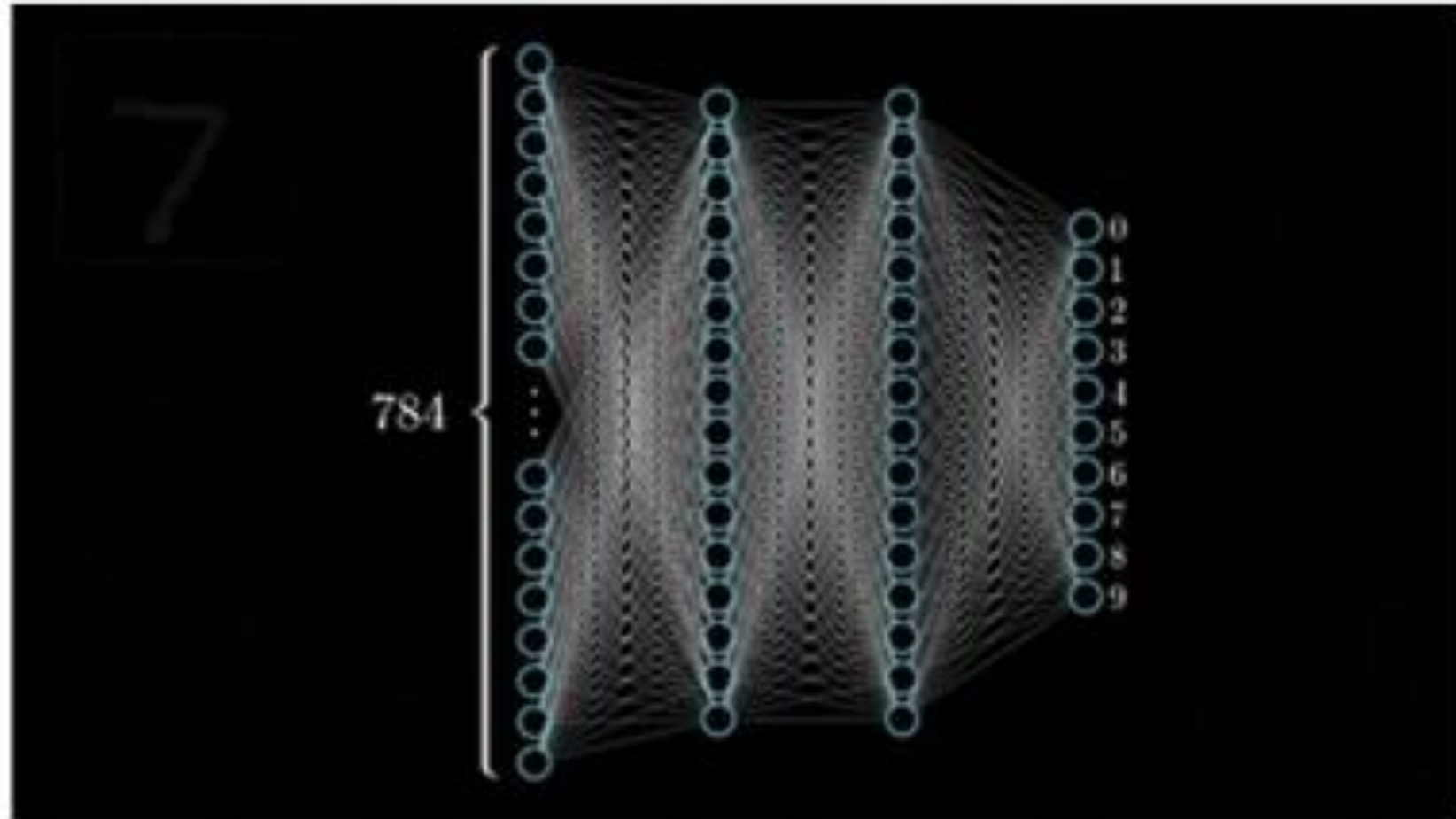
Tareas de clasificación multiclase.
Salida pasa a través de una softmax
para que cada nodo de salida
proporcione una probabilidad entre
0 y 1. Se usa con one hot encoders.

Optimizadores

Su principal objetivo es ir variando los parámetros para dar “el paso adecuado” y así llegar a un mínimo global de manera rápida.

- Descenso del gradiente estocástico - sgd
- Descenso del gradiente con momento - Acumula el gradiente de pasos previos para el siguiente avance (acelera).
- Propagación de la raíz promedio cuadrático – RMSprop, al elevar al cuadrado evita que se cancelen los gradientes
- Optimización adaptiva con momento - ADAM
- Otros: AdaDelta, SGD Nesterov, AdaGrad

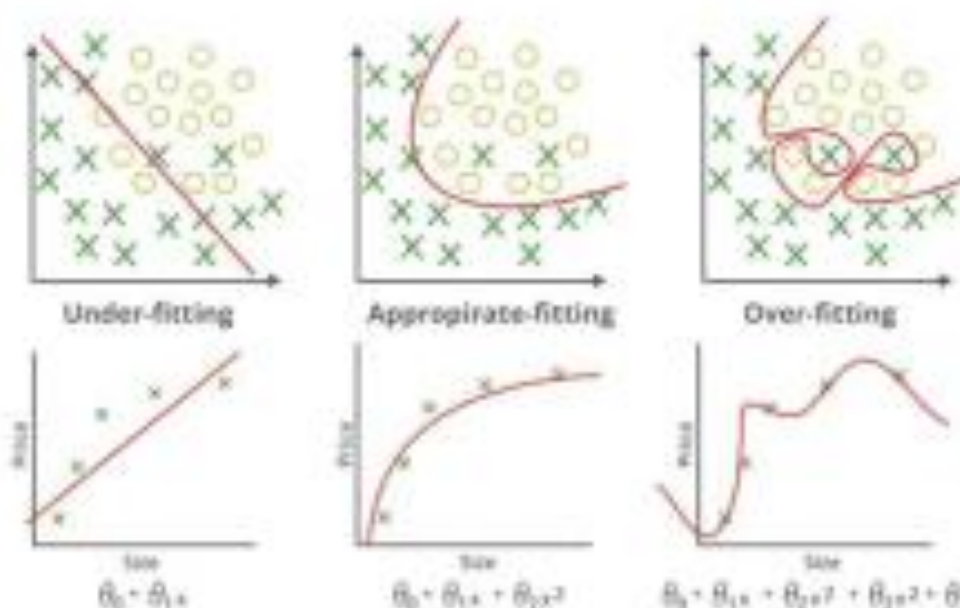
Probar la Red convolucional



¡Un problema muy común!

- Desvanecimiento del gradiente
 - Valores muy grandes en los filtros y otros demasiado pequeños, se tiene una divergencia de datos
- Sobre entrenamiento (overfitting), qué tan bien responde tu red a nuevos datos

Cómo evitar el sobreajuste (overfitting): regularización



Solución: agregar términos de regularización a la función de pérdida para reducir el número de parámetros o forzar que algunos de ellos sean cercanos a cero, o bien, eliminar nodos aleatoriamente (dropout)

$$E = \frac{1}{2} \sum_j (t_j - y_j)^2 + \frac{\lambda}{2} \sum_{i,j} w_{ij}^2 \quad E = \frac{1}{2} \sum_j (t_j - y_j)^2 + \frac{\lambda}{2} \sum_{i,j} |w_{ij}| \quad E = \frac{1}{2} \sum_j (t_j - y_j)^2 + \frac{\lambda}{2} \sum_{i,j} \frac{w_{ij}^2}{1 + w_{ij}^2 / w_0^2}$$

Entrenamiento por lotes (batches)

“Divide y vencerás”

- Se entrena a por secciones.
- Menos datos
- Más rápido



Ejemplo

- 2000 imágenes
- Batches =100 imágenes
20 pasos de cálculo de gradiente por época
- Número de épocas por ejemplo *epochs=100, 500, 1000...*

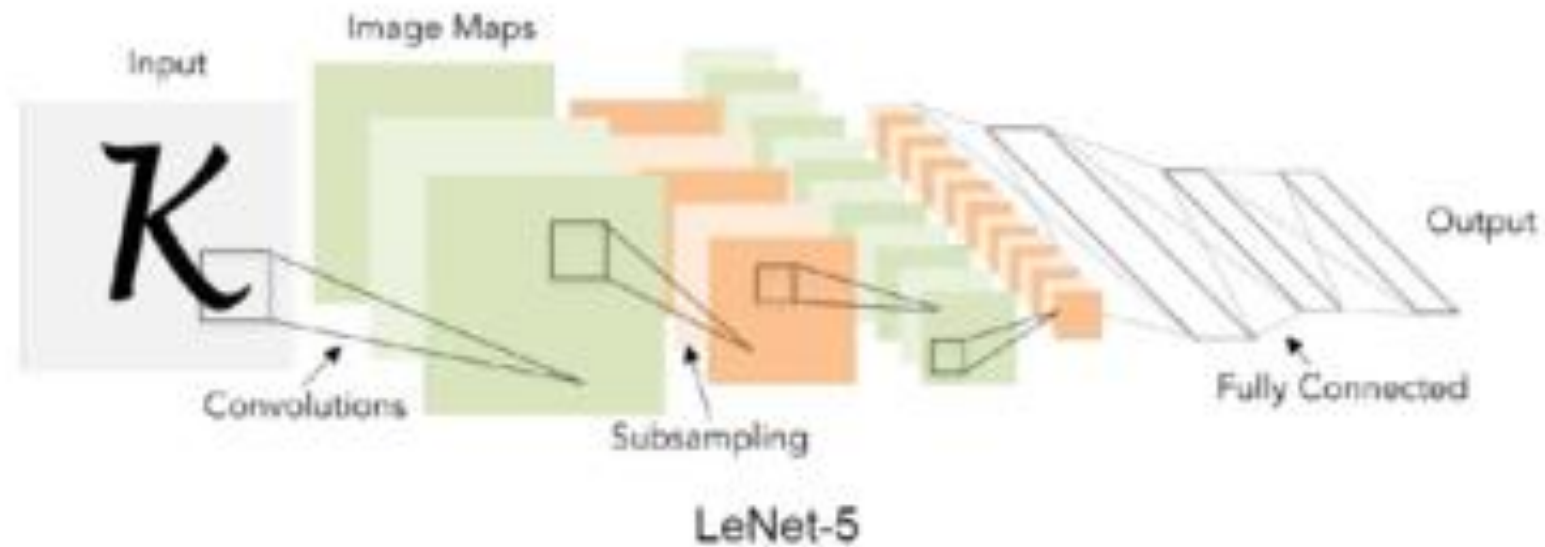
Arquitecturas Típicas

- AlexNet
 - Team Name: SuperVision
 - Team Member: Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton, University of Toronto
- VGG
 - Karen Simonyan, Andrew Zisserman, University of Oxford
 - Visual Geometry Group (Engineering Science)
- GoogLeNet
 - Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Drago Anguelov, Dumitru Erhan, Andrew Rabinovich
- ResNet

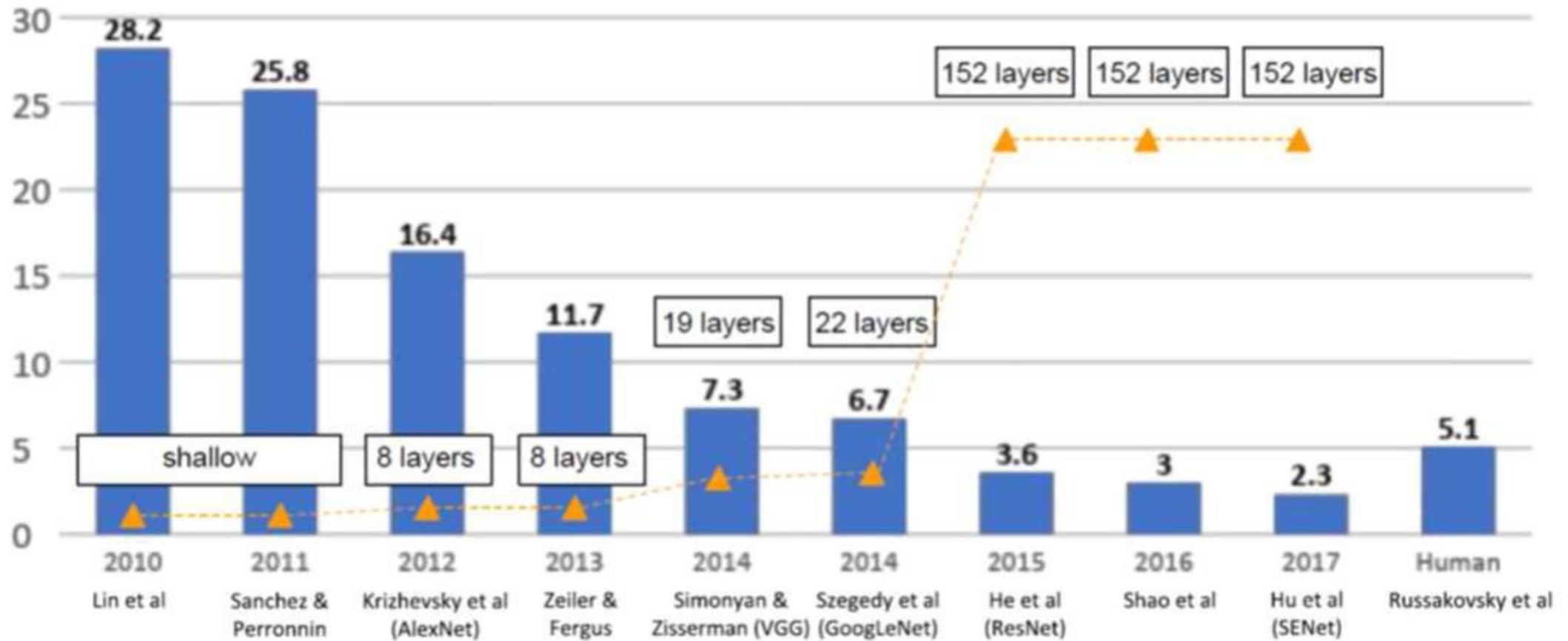
Un poco de historia..

Gradient-based learning applied to document recognition

[LeCun, Bottou, Bengio, Haffner 1998]



ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Primeros que utilizan 2 GPUs en paralelo que
comparten informacion entre ellos

ImageNet Classification with Deep Convolutional Neural Networks

[Krizhevsky, Sutskever, Hinton, 2012]

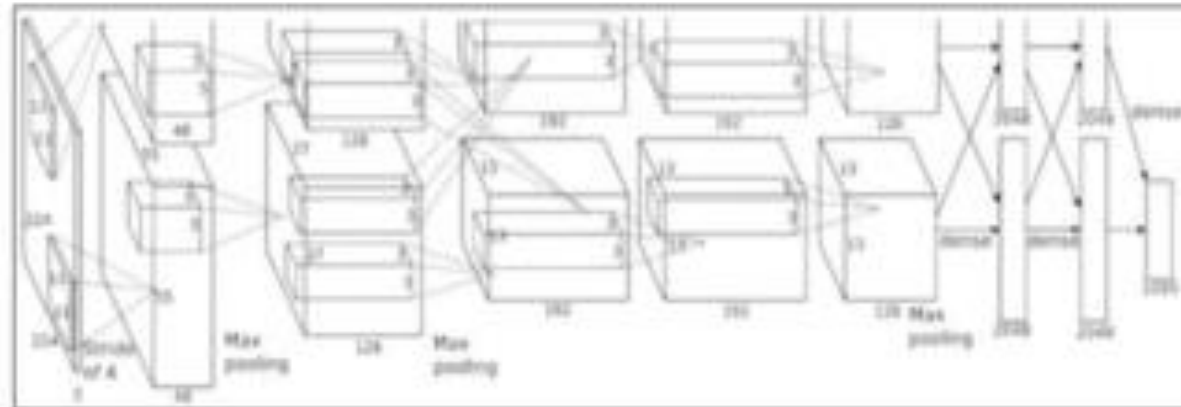


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

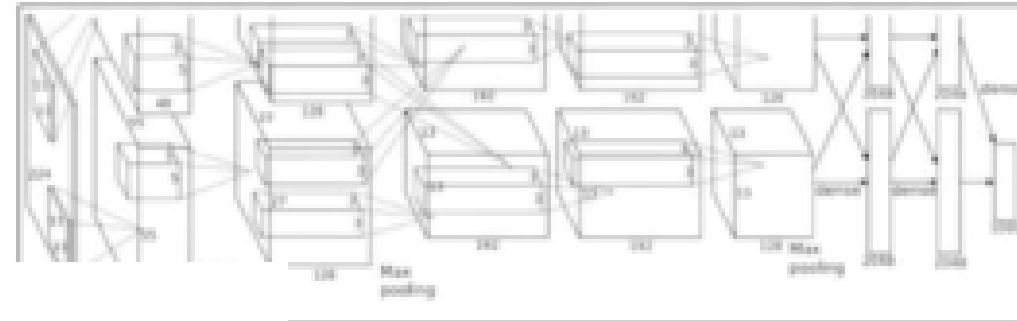
“AlexNet”

ImageNet Classification with Deep Convolutional Neural Networks

[Krizhevsky, Sutskever, Hinton, 2012]

Arquitectura:

[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)



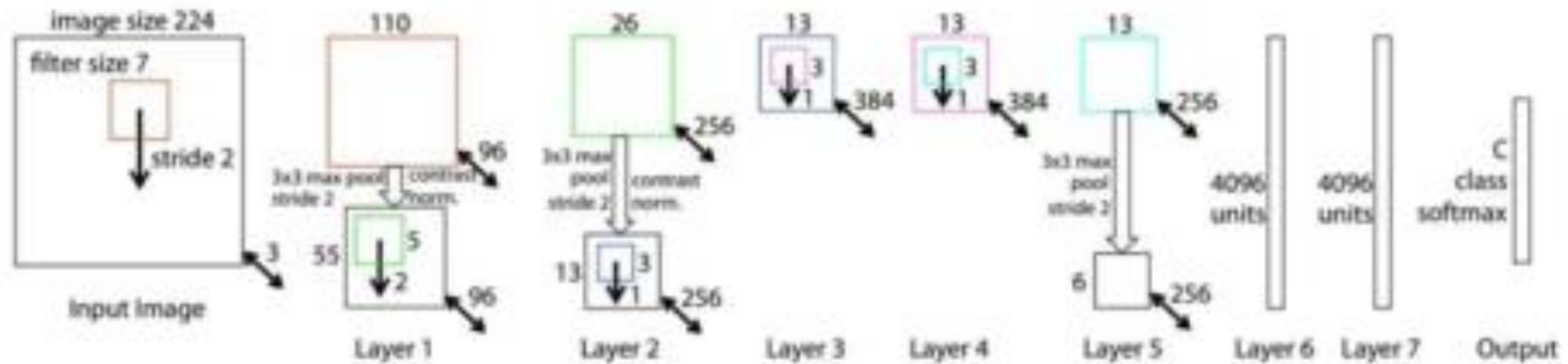
Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

“AlexNet”

Mejora de hiperparámetros vs. AlexNet

ZFNet

[Zeiler and Fergus, 2013]



AlexNet but:

CONV1: change from (11x11 stride 4) to (7x7 stride 2)

CONV3,4,5: instead of 384, 384, 256 filters use 512, 1024, 512

ImageNet top 5 error: 16.4% -> 11.7%

;; Deeper networks!!

VGG 16 y VGG19

[Simonyan and Zisserman, 2014]

Small filters, Deeper networks

8 layers (AlexNet)

-> 16 - 19 layers (VGG16Net)

Only 3x3 CONV stride 1, pad 1
and 2x2 MAX POOL stride 2

11.7% top 5 error in ILSVRC'13
(ZFNet)

-> 7.3% top 5 error in ILSVRC'14

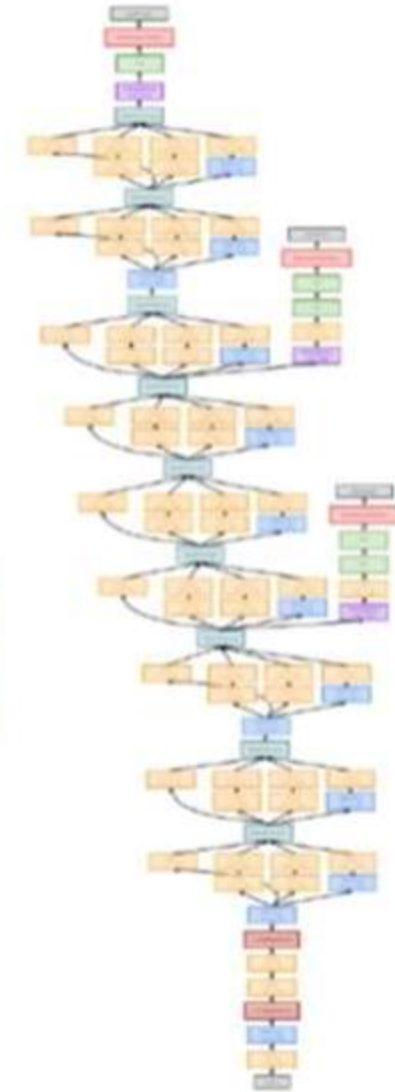
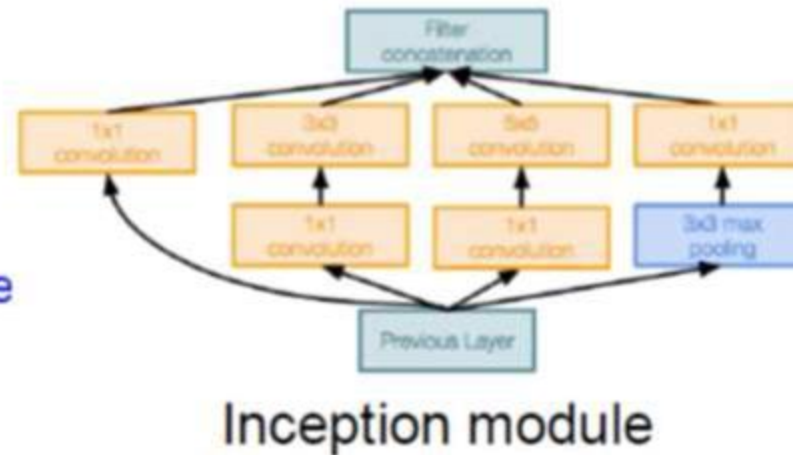


GoogLeNet

[Szegedy et al., 2014]

Deeper networks, with computational efficiency

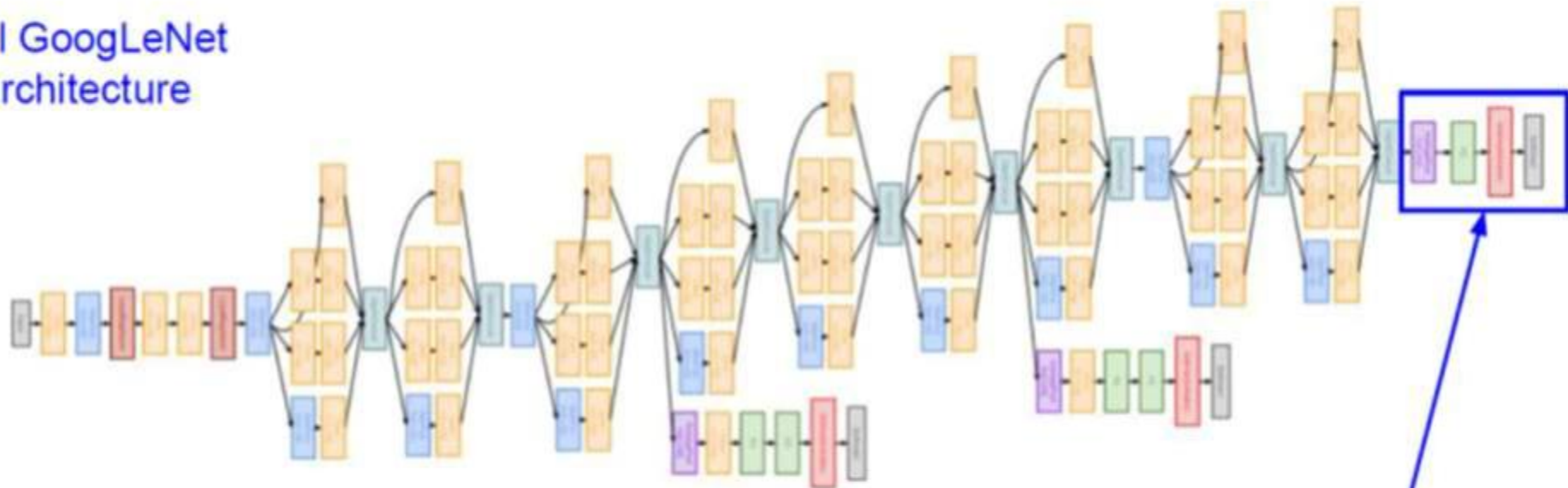
- 22 layers
- Efficient “Inception” module
- No FC layers
- Only 5 million parameters!
12x less than AlexNet
- ILSVRC'14 classification winner
(6.7% top 5 error)



GoogLeNet

[Szegedy et al., 2014]

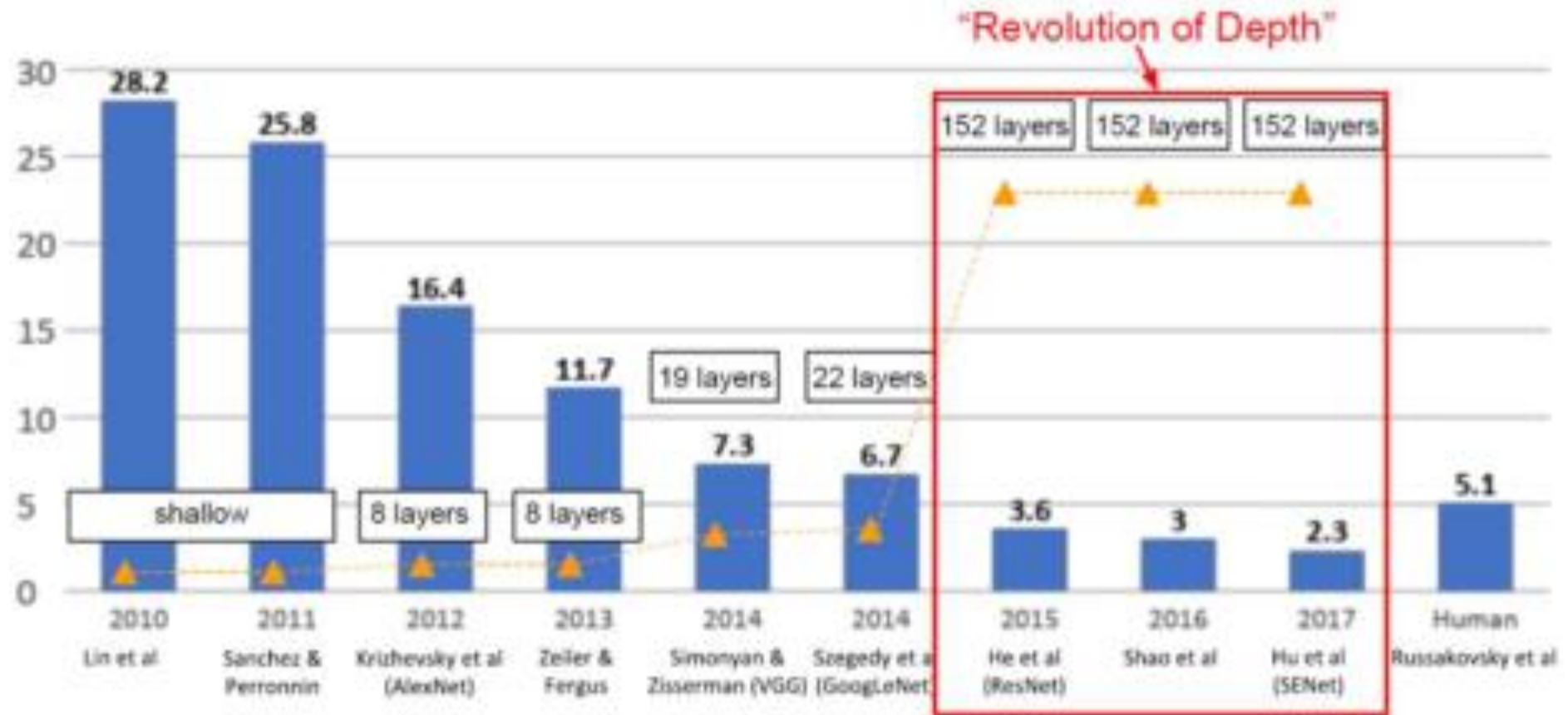
Full GoogLeNet
architecture



Classifier output
(removed expensive FC layers!)

Mas profundidad...

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

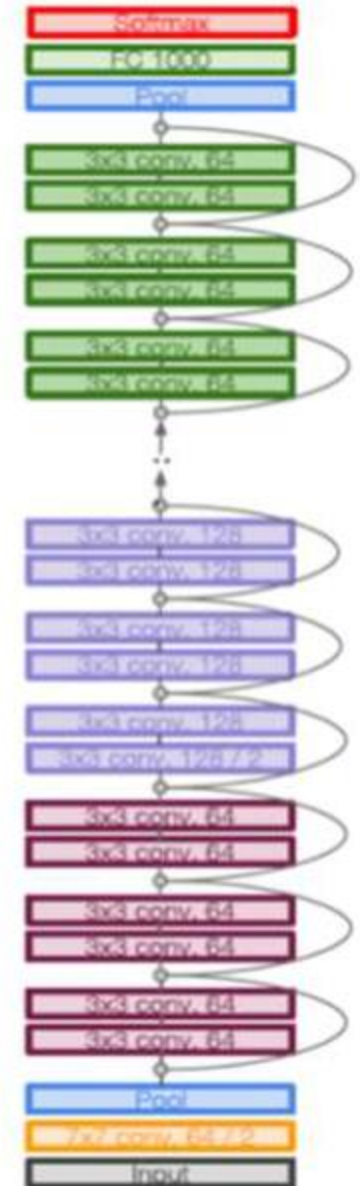
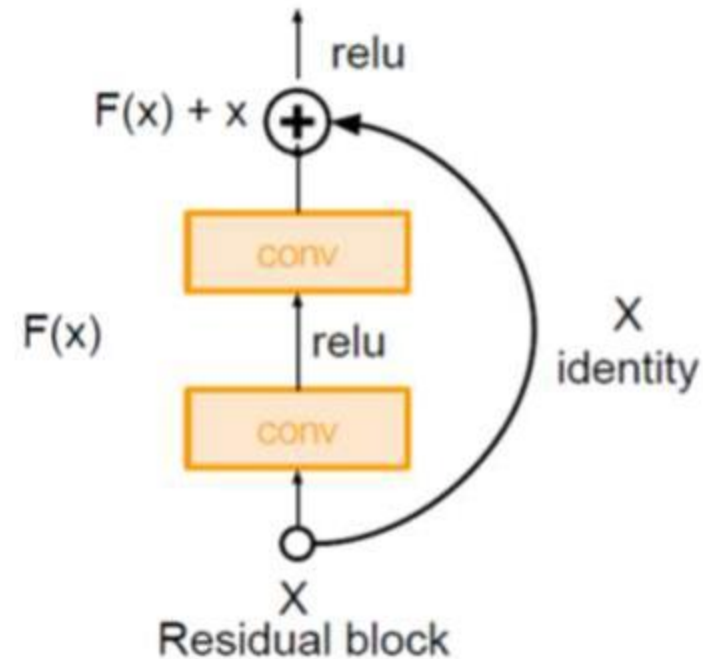


ResNet

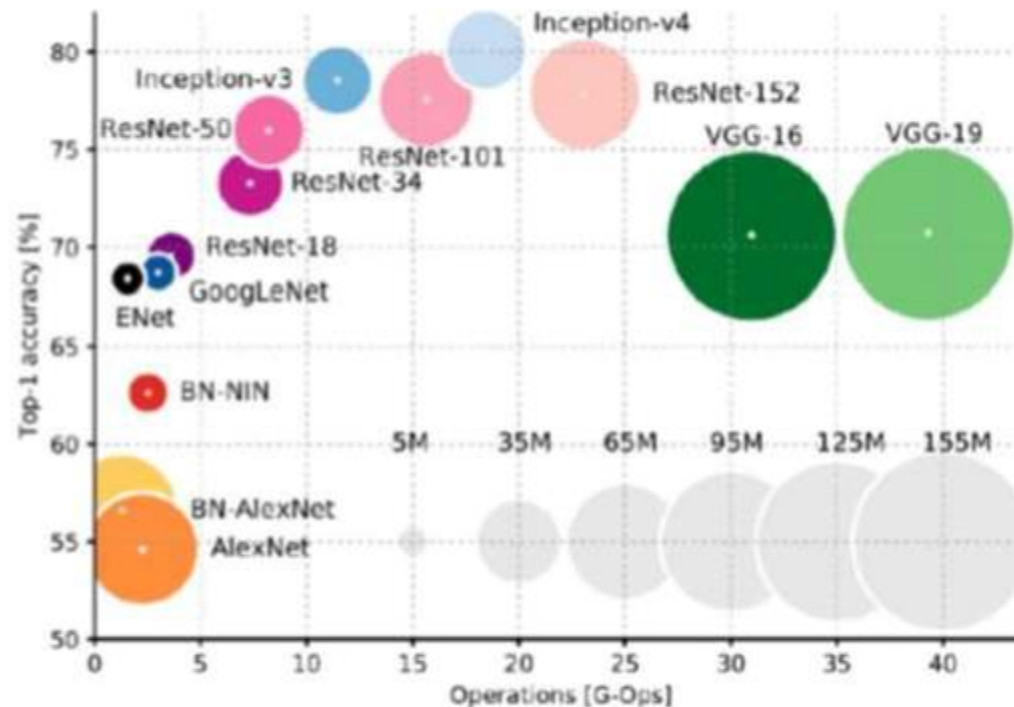
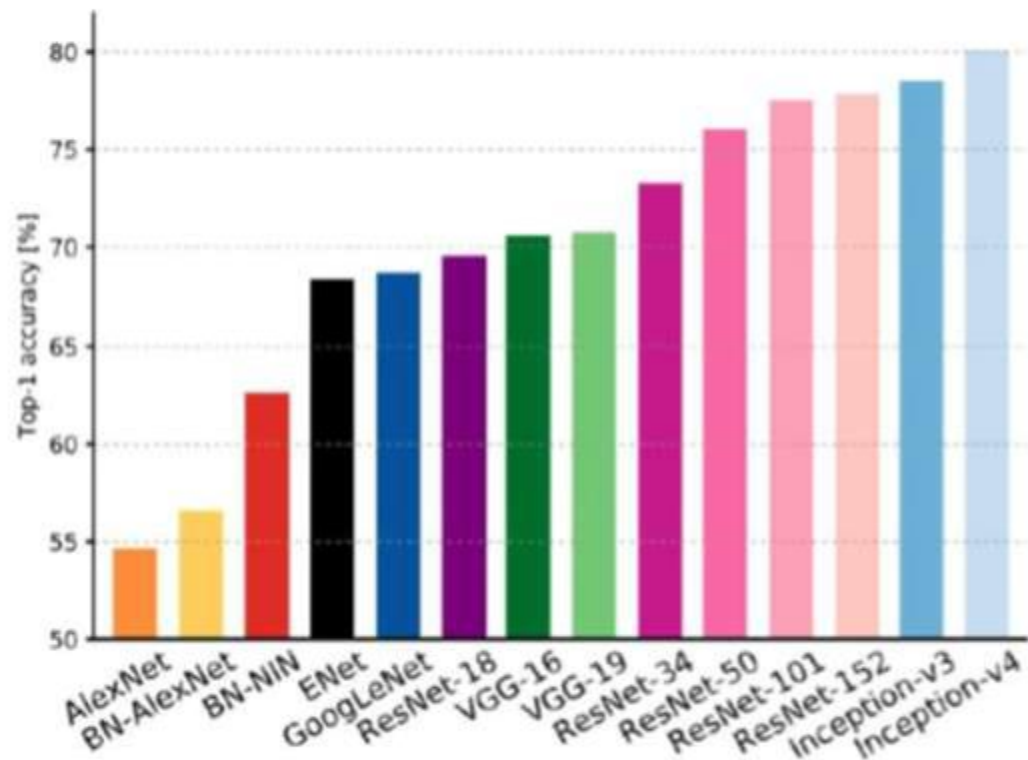
[He et al., 2015]

Very deep networks using residual connections

- 152-layer model for ImageNet
- ILSVRC'15 classification winner (3.57% top 5 error)
- Swept all classification and detection competitions in ILSVRC'15 and COCO'15!



Comparing complexity...



An Analysis of Deep Neural Network Models for Practical Applications, 2017.