



**DECSAI**

**Departamento de Ciencias de la Computación e I.A.**

Universidad de Granada



# Redes convolutivas

Fernando Berzal, [berzal@acm.org](mailto:berzal@acm.org)

# Redes convolutivas



- Motivación: Invarianza
- Redes convolutivas
  - Convolución
  - Mapas de características
  - Campos receptivos locales
  - Pooling
- Resultados experimentales y aplicaciones
  - Clasificación de imágenes
  - Detección de objetos



# En la práctica...



## Algoritmo de aprendizaje de redes multicapa

Aspectos que debemos considerar en su diseño:

- **Parámetros:** ¿Qué topología de red utilizamos?...
- **Optimización:** ¿Cómo obtenemos los pesos?
- **Generalización:** ¿Cómo conseguimos que la red funcione bien con datos distintos a los del conjunto de entrenamiento?
- **Invarianza:** ¿Cómo conseguimos que la red sea robusta frente a transformaciones comunes en los datos?



**DECSAI**

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada

## Redes convolutivas

Motivación

# En la práctica: Invarianza



Técnicas [ad hoc] que nos permiten incorporar conocimiento previo en el diseño de una red neuronal:

- Restringir la arquitectura de la red (uso de conexiones locales, a.k.a. "campos receptivos").
- Restringir la selección de pesos de la red (uso de pesos compartidos por varias neuronas, i.e. "weight-sharing").



## En la práctica: Invarianza



### Motivación

El reconocimiento de objetos es difícil por varios motivos:

- Segmentación (varios objetos en la misma imagen).
- Ocultación (partes ocultas detrás de otros objetos).
- Iluminación (valores de los píxeles determinados tanto por la iluminación ambiental como por el objeto al que corresponden).
- Punto de vista (cambios en el punto de vista causan cambios en la imagen que se capta).
- Deformación (los objetos pueden aparecer de varias deformados de muchas maneras [no afines]), p.ej. textos manuscritos.

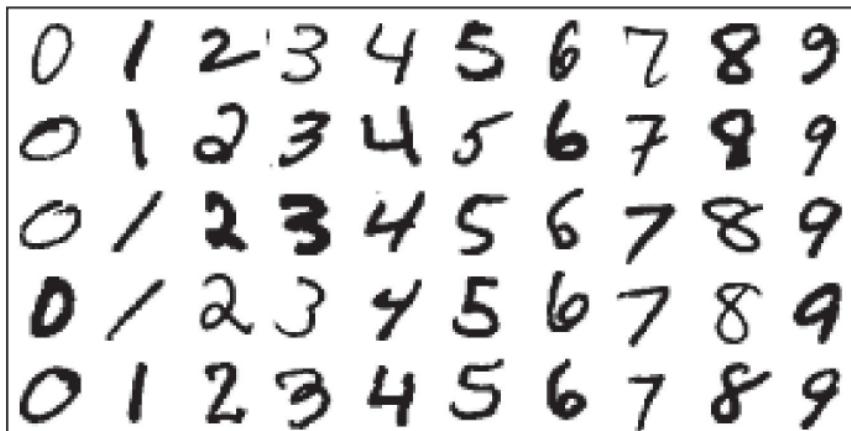


# En la práctica: Invarianza



## Justificación

Nuestros sistemas visual y auditivo parecen haberse especializado para tratar sin problemas determinadas transformaciones (sin que seamos conscientes de su complejidad real).

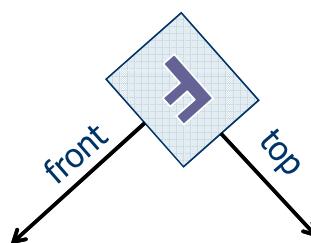


# En la práctica: Invarianza



## Segmentación: Preprocesamiento de los datos

Identificamos una región alrededor de un objeto y la utilizamos como marco de coordenadas de referencia para un conjunto de píxeles normalizado (p.ej. MNIST).



No siempre es fácil:

¡Hay que reconocer el objeto para delimitar su región!



# En la práctica: Invarianza



## Por fuerza bruta

- En la fase de entrenamiento, utilizamos imágenes bien segmentadas y ya normalizadas.
- En la fase de prueba, probamos con distintas regiones (variando escala y orientación).

The screenshot shows a Facebook interface with the title "Who's in These Photos?". It displays six small photographs with rectangular boxes highlighting faces. Below each box is a button labeled "Who is this?". The interface includes a search bar and a "Home" link at the top.



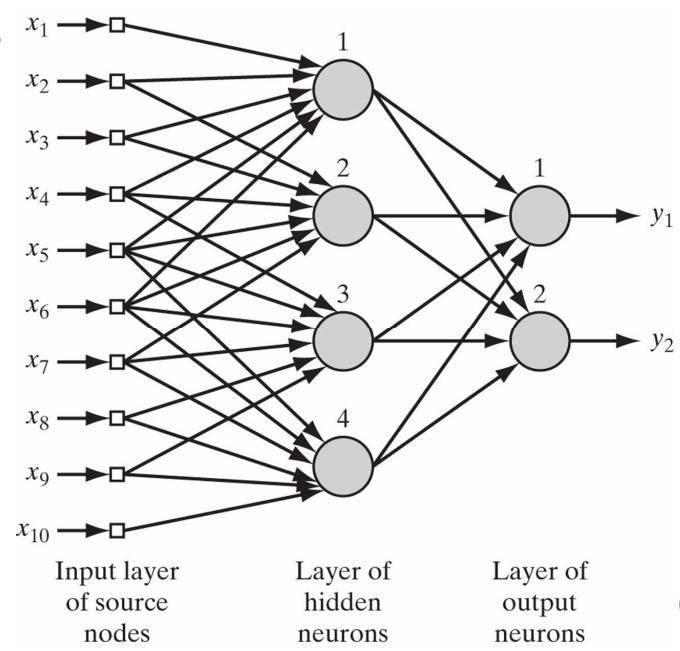
## EJEMPLO

Reconocimiento de caras

# Redes convolutivas



## Campos receptivos y pesos compartidos



[Haykin: "Neural Networks and Learning Machines", 3<sup>rd</sup> edition]



**DECSAI**

**Departamento de Ciencias de la Computación e I.A.**

Universidad de Granada



# Redes convolutivas

## Convolución

# Redes convolutivas



### Ideas clave

- Se usan múltiples copias de los mismos detectores de características en distintas posiciones.
- La replicación reduce el número de parámetros (pesos) que deben ajustarse.
- Se pueden utilizar distintos tipos de detectores (cada uno con su “mapa” de detectores replicados): cada fragmento de la imagen se representa de distintas formas.

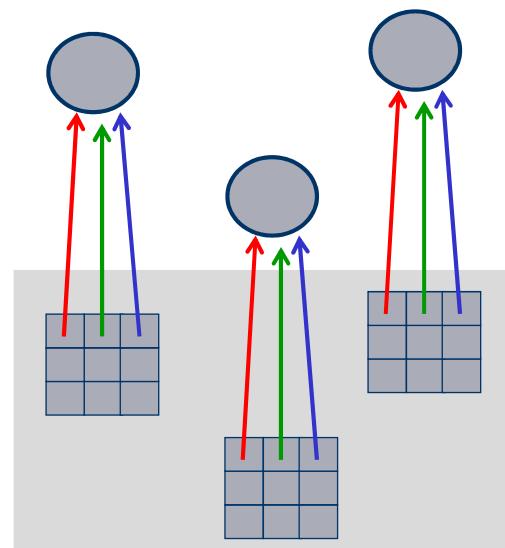


# Redes convolutivas



Cada unidad detecta una característica en una región diferente de la imagen.

Todas comparten los mismos pesos.



# Redes convolutivas



## Convolución

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Imagen de entrada

1	0	1
0	1	0
1	0	1

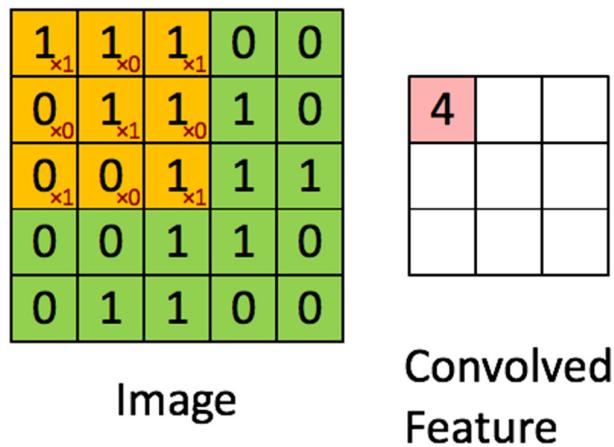
Máscara  
filtro / kernel /  
detector de características



# Redes convolutivas



## Convolución



<http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/>



# Redes convolutivas



Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	



# Redes convolutivas



<b>Sharpen</b>	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
<b>Box blur</b> (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
<b>Gaussian blur</b> (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	



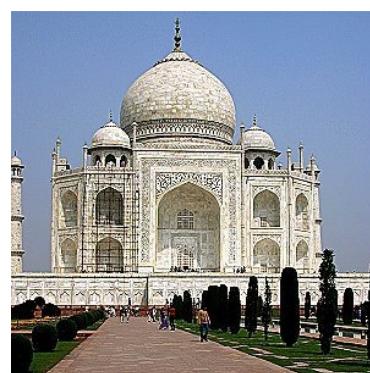
# Redes convolutivas



## Ejemplos: GIMP

### ■ Sharpen

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



### ■ Blur

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



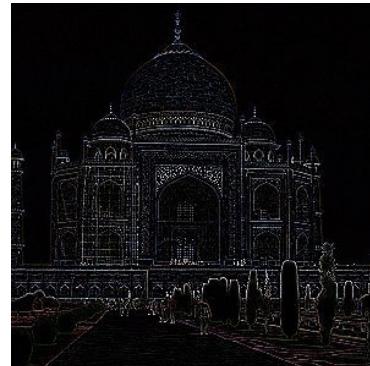
# Redes convolutivas



Ejemplos: GIMP

- Edges

0	1	0
1	-4	1
0	1	0



- Emboss

-2	-1	0
-1	1	1
0	1	2



# Redes convolutivas



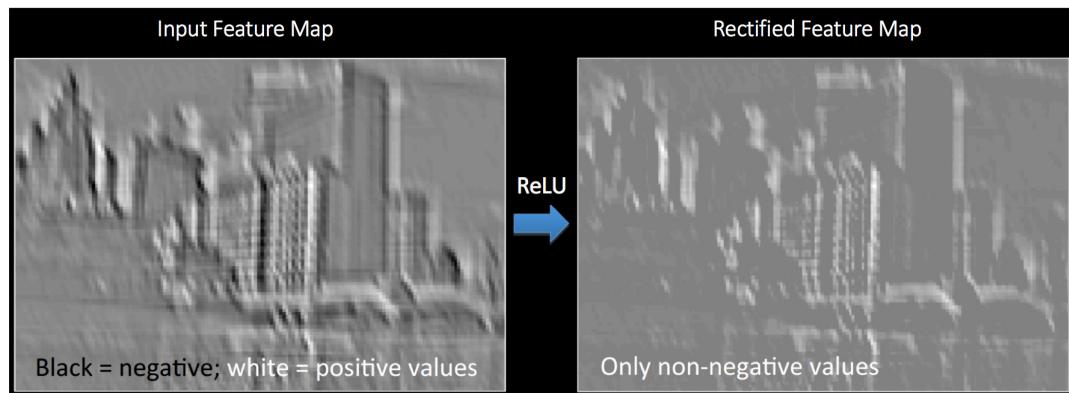
## Mapas de características ["Feature maps"]



# Redes convolutivas



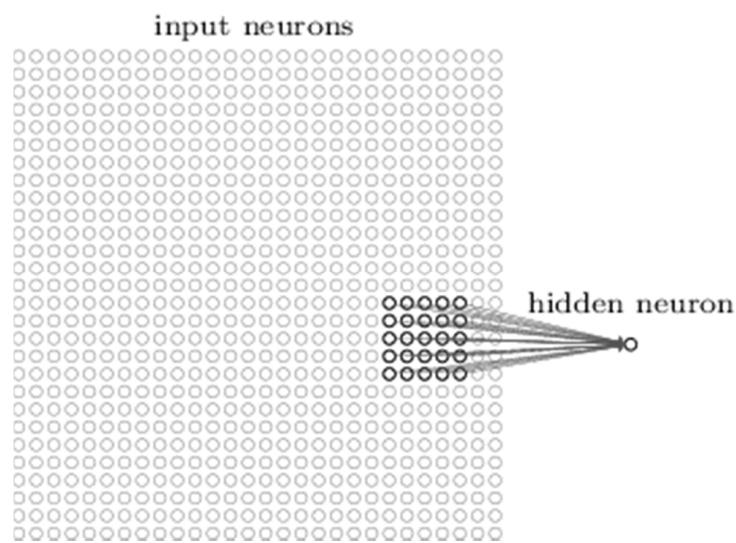
"Rectified feature maps"  
ReLU @ convolutional layer



# Redes convolutivas



**Campos receptivos locales**  
["local receptive fields"]



# Redes convolutivas



Backpropagation con restricciones sobre los pesos:

*Para obligar  $w_1 = w_2$   
tenemos que garantizar  $\Delta w_1 = \Delta w_2$*

*Calculamos  $\frac{\partial E}{\partial w_1}$  y  $\frac{\partial E}{\partial w_2}$*

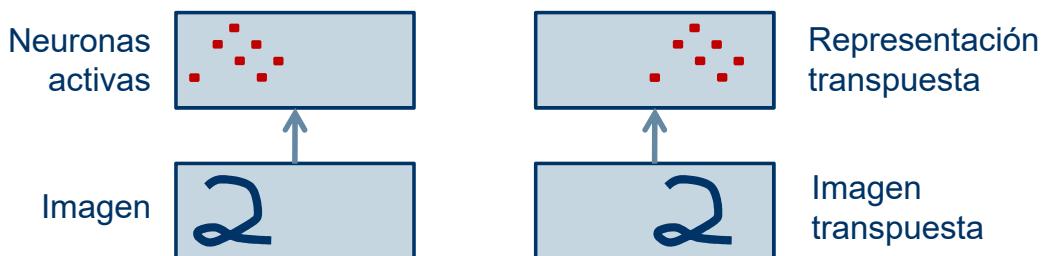
*Usamos  $\frac{\partial E}{\partial w_1} + \frac{\partial E}{\partial w_2}$  para  $w_1$  y  $w_2$*



# Redes convolutivas



¿Qué consiguen los detectores replicados?



Los detectores replicados no hacen que la actividad neuronal sea invariante frente a traslaciones:  
su actividad es “equivariante”.

Si la característica resulta útil en algunas posiciones, la red incorpora detectores para esa característica en todas las posiciones





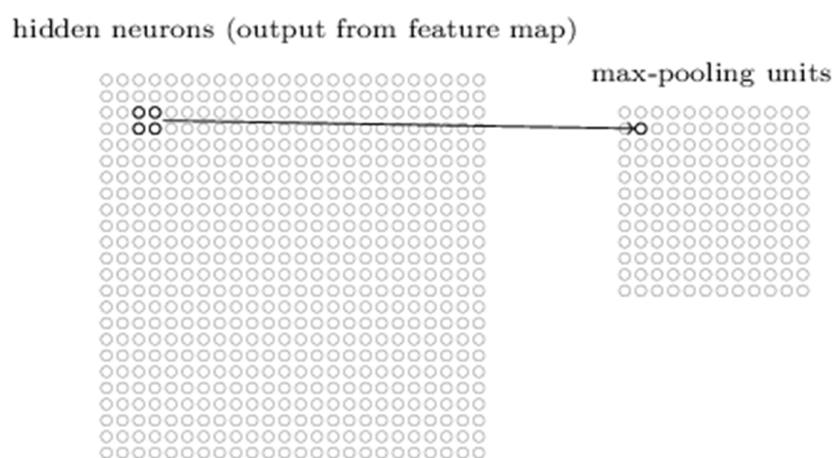
# Redes convolutivas

## Pooling

# Redes convolutivas



## **Pooling** (reducción de la dimensionalidad)

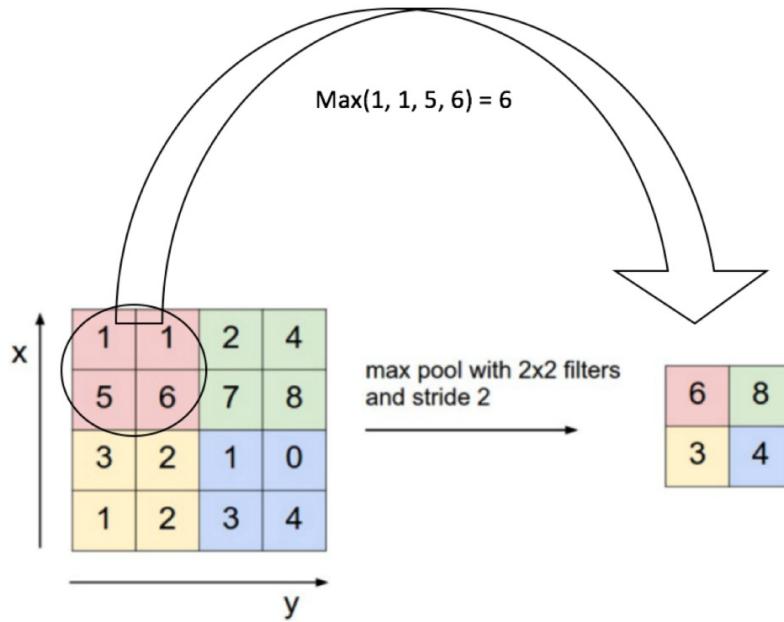


# Redes convolutivas



## Pooling

a.k.a. spatial pooling / subsampling / downsampling

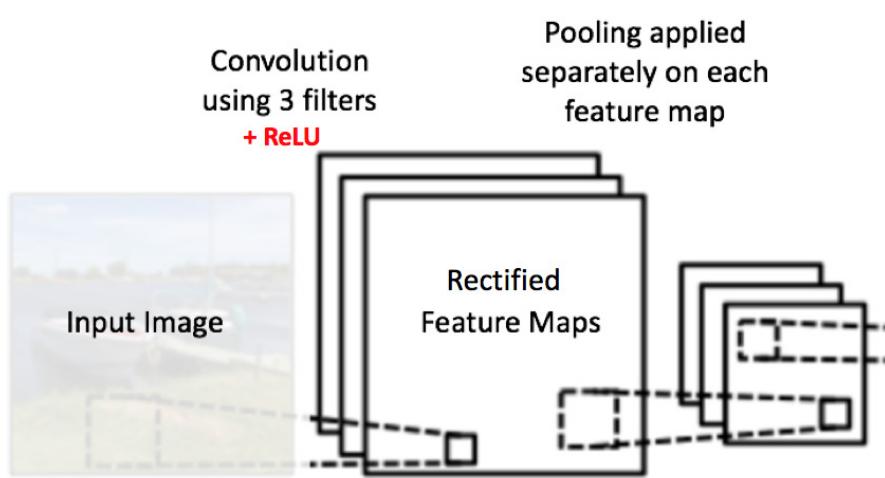


# Redes convolutivas



## Pooling

a.k.a. spatial pooling / subsampling / downsampling

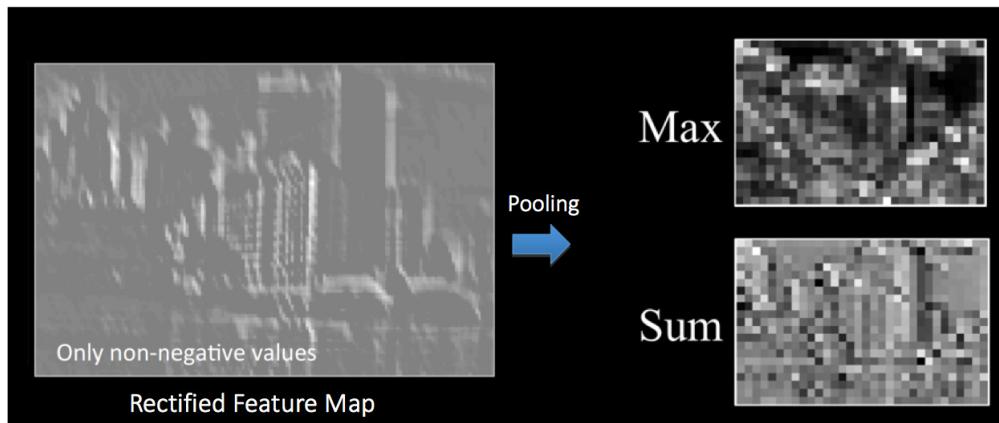


# Redes convolutivas



## Pooling

a.k.a. spatial pooling / subsampling / downsampling



# Redes convolutivas



## Pooling

¿Qué se consigue agregando las salidas de receptores de características replicados?

- Una reducción del número de entradas de las siguientes capas de la red neuronal, lo que nos permite calcular más características distintas en paralelo.
- Una pequeña cantidad de invarianza frente a traslaciones en cada nivel (p.ej. 2x2 max-pooling).
- Problema: Tras varios niveles de “pooling”, se pierde información acerca de la posición exacta de las cosas.





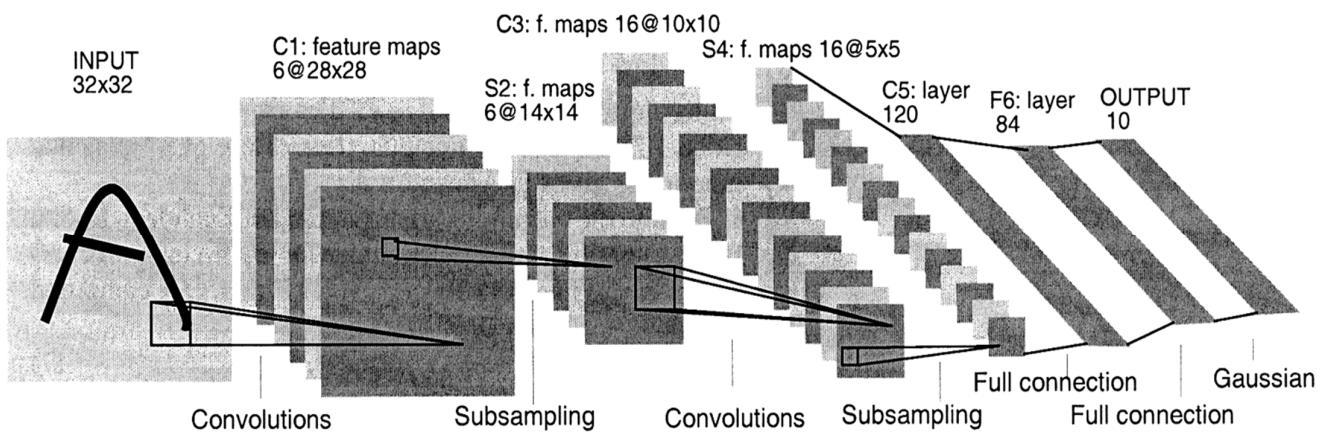
# Redes convolutivas

Aplicaciones: Clasificación de imágenes

## Aplicaciones



### Clasificación de dígitos manuscritos (MNIST)



EJEMPLO: **LeNet5**

<http://yann.lecun.com/exdb/lenet/>



# Aplicaciones



EJEMPLO: Los 82 errores de **LeNet5**

<http://yann.lecun.com/exdb/lenet/>



# Aplicaciones



## Clasificación de dígitos manuscritos (MNIST)



DEMO: MNIST

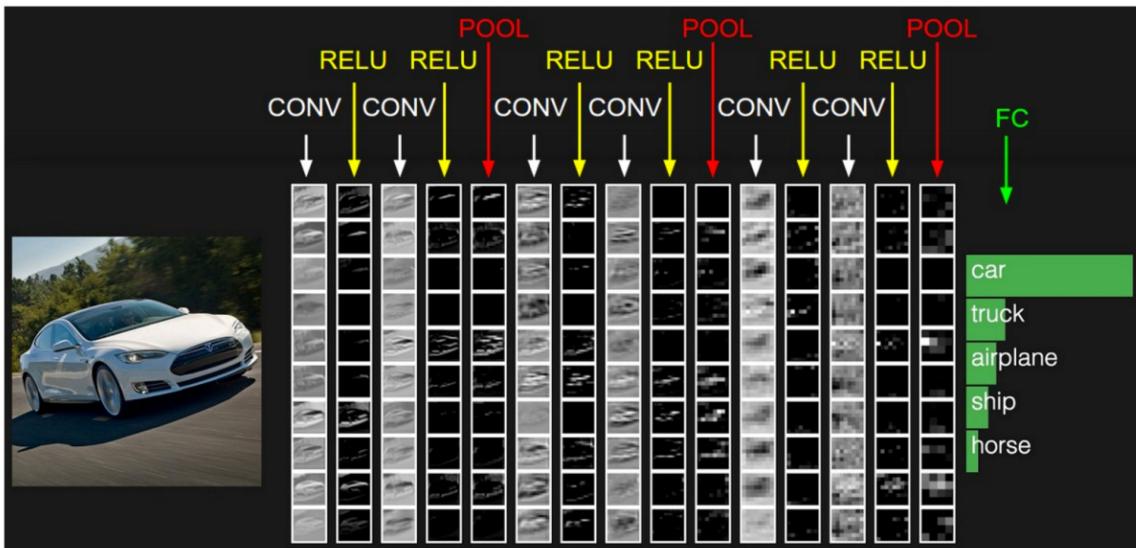
<http://scs.ryerson.ca/~aharley/vis/conv/flat.html>



# Aplicaciones



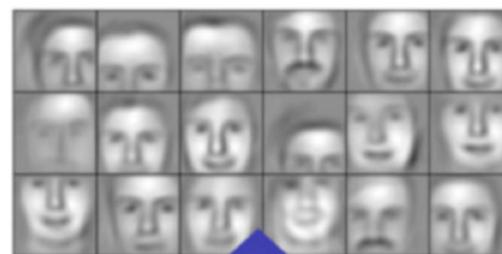
## Clasificación de imágenes



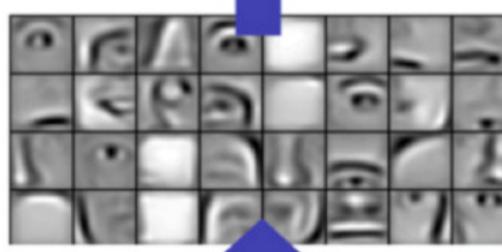
# Aplicaciones



## Reconocimiento facial



Layer 3



Layer 2



Layer 1

DEMOS:

Buscar en YouTube



# Aplicaciones



## Clasificación de imágenes: Evolución

- AlexNet @ ILSVRC'2012: Alex Krizhevsky et al.  
University of Toronto: 60M parameters <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- ZF Net @ ILSVSC'2013: Matthew Zeiler & Rob Fergus  
NYU: <https://arxiv.org/abs/1311.2901>
- GoogLeNet @ ILSVC'2014: Szegedy et al.  
Google "Inception module": 4M parameters <http://arxiv.org/abs/1409.4842>
- VGGNet @ ILSVC'2014 (2nd place):  
Oxford Visual Geometry Group  
[http://www.robots.ox.ac.uk/~vgg/research/very\\_deep/](http://www.robots.ox.ac.uk/~vgg/research/very_deep/)
- ResNets [residual nets] @ ILSVC'2015:  
Microsoft Research: <https://arxiv.org/abs/1512.03385>
- DenseNet (2016):  
Densely-connected convolutional networks. <http://arxiv.org/abs/1608.06993>



# Aplicaciones

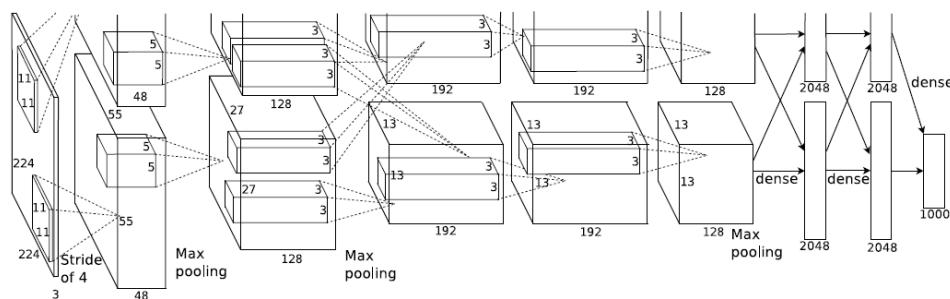


IMAGENET

## Large Scale Visual Recognition Challenge

Red neuronal diseñada por Alex Krizhevsky (NIPS 2012)

- "Deep network" con topología compleja: 7 capas ocultas (las primeras convolutivas, las últimas completamente conectadas).

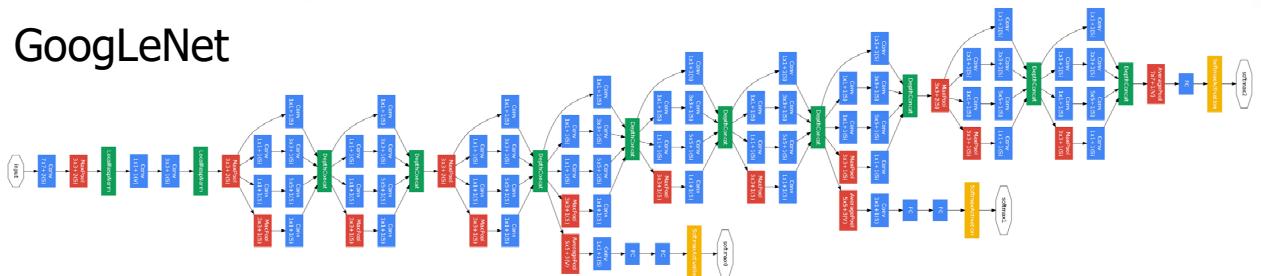


- Múltiples trucos para mejorar su capacidad de generalización ("image patches" & "dropout").

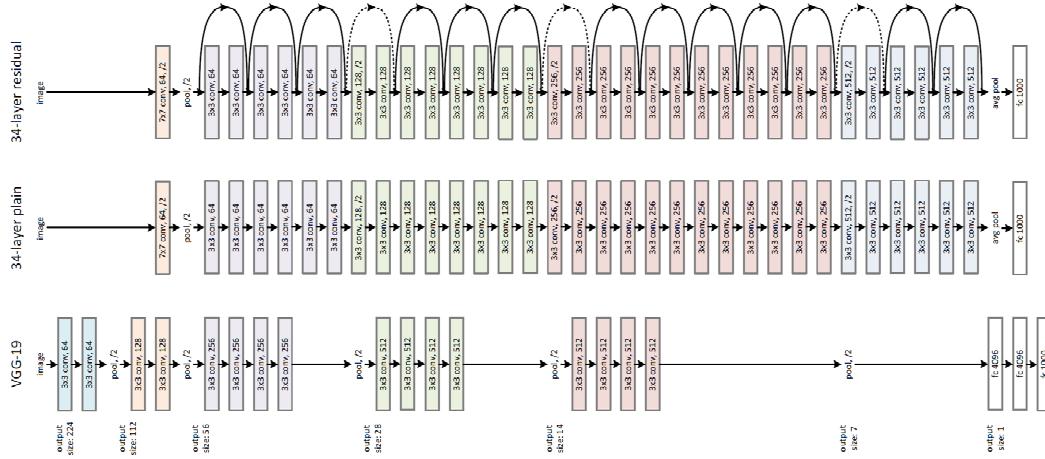


# Aplicaciones

## GoogLeNet

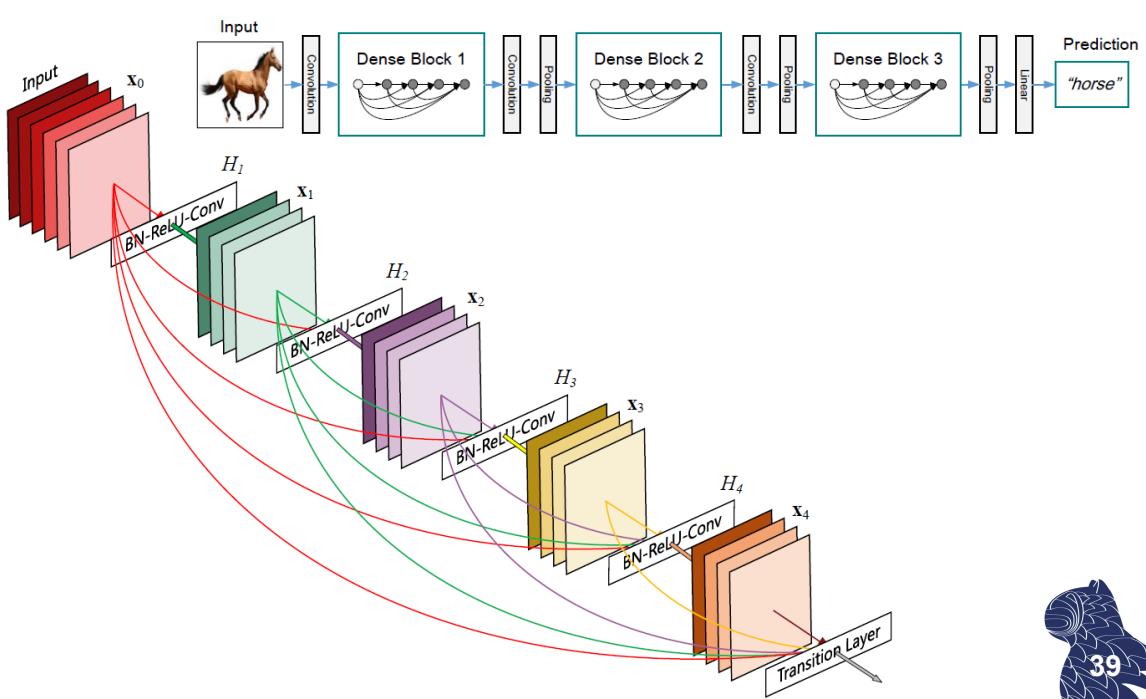


## ResNets



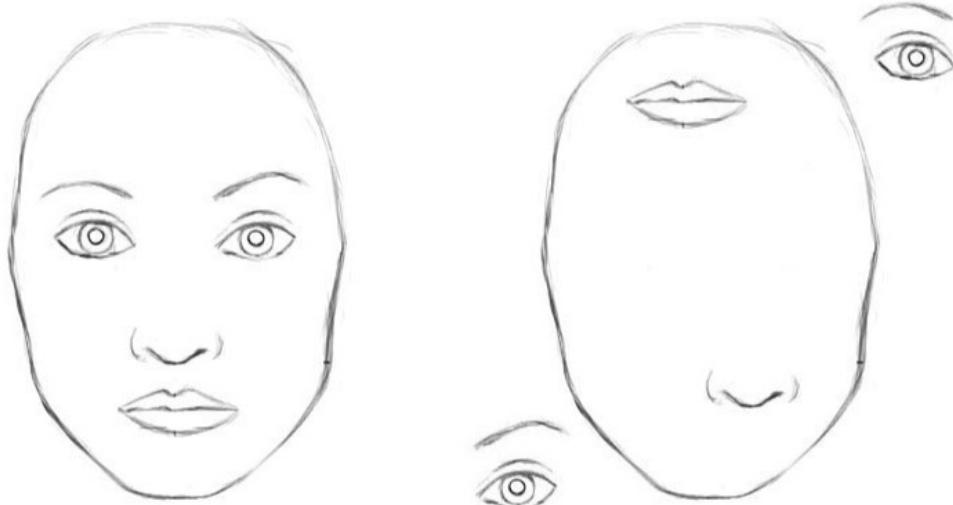
# Aplicaciones

## DenseNets



# Limitaciones del Deep Learning

Las redes convolutivas [CNNs] funcionan muy bien en la práctica, pero...



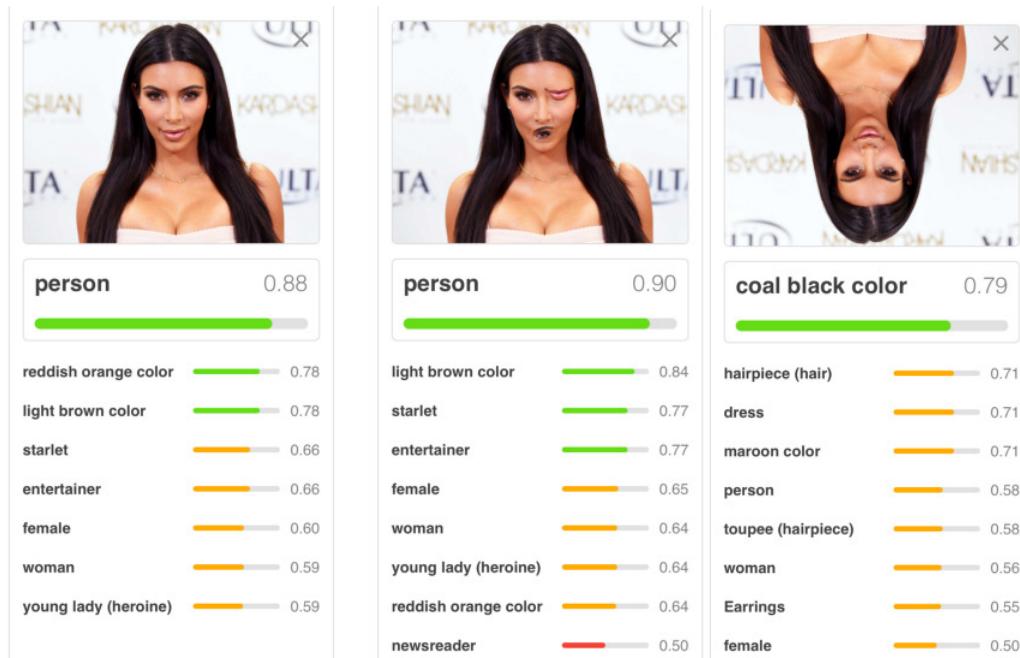
... para una CNN, ambas imágenes son similares ☹



# Limitaciones del Deep Learning

“Convolutional neural networks are doomed”

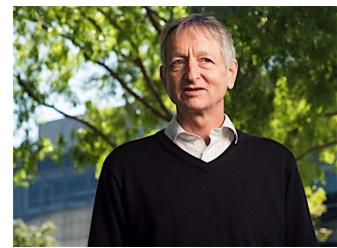
—Geoffrey Hinton



# Limitaciones del Deep Learning

- Las redes convolutivas detectan características, pero no su colocación relativa (traslación & rotación).
- Las redes convolutivas ignoran las posiciones relativas utilizando “pooling”, un apaño que funciona sorprendentemente bien en la práctica:

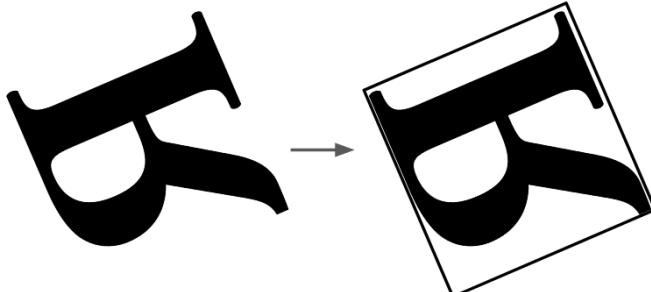
*"The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster." – Geoffrey Hinton*



# Limitaciones del Deep Learning

## Problema clave de las redes convolutivas

La representación interna de una red convolutiva no tiene en cuenta las relaciones espaciales entre objetos, ni la jerarquía existente entre objetos simples y los objetos compuestos de los que forman parte.



# Limitaciones del Deep Learning



## Limitaciones

Falta de compresión (en sentido humano) ...



The boy is holding a baseball bat.



**DECSAI**

**Departamento de Ciencias de la Computación e I.A.**

Universidad de Granada



## Redes convolutivas

En la práctica: Invarianza & robustez

# En la práctica: Invarianza



Técnicas que nos permiten diseñar redes robustas, invariantes frente a determinadas transformaciones (rotaciones o cambios de escala en imágenes; cambios de volumen, velocidad o tono en reconocimiento de voz):

- Invarianza por **extracción de características**.
- Invarianza por **estructura**.
- Invarianza por **entrenamiento**.

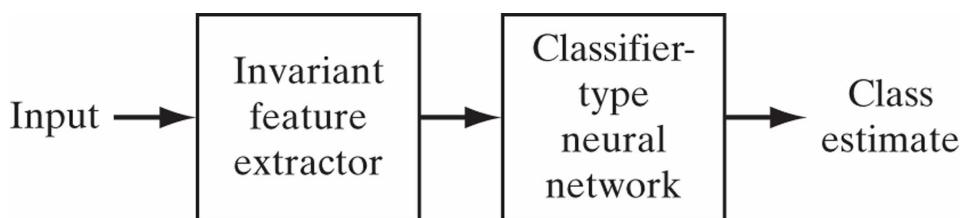


## En la práctica: Invarianza



### **Invarianza por extracción de características**

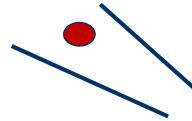
Preprocesamiento del conjunto de entrenamiento, del que se extraen características “esenciales” que sean invariantes con respecto a las transformaciones deseadas).



# En la práctica: Invarianza



## Invarianza por extracción de características



### EJEMPLO

Par de líneas, más o menos paralelas, con algo en medio.

### IDEA

Con un número suficiente de características de este tipo, podríamos esperar ser capaces de identificar un objeto...



# En la práctica: Invarianza



## Invarianza por estructura, p.ej. redes convolutivas

Red diseñada de forma que sus conexiones sinápticas hagan que versiones transformadas de la entrada produzcan salidas similares.

- Conectividad de la red ("local receptive fields")
- Restricciones sobre los pesos ("weight sharing")
- ...

Menos tedioso que diseñar/extraer a mano características, aunque introduce prejuicios acerca de la forma particular de resolver el problema que tengamos en mente.



# En la práctica: Invarianza



## Invarianza por entrenamiento

Red entrenada a partir de un conjunto de entrenamiento ampliado, en el que se incluyen versiones transformadas de los ejemplos originales del conjunto de entrenamiento.

- El entrenamiento puede ser mucho más costoso.
- La optimización de los pesos de la red puede descubrir formas novedosas de utilizar la red multicapa
  - ... que a nosotros no se nos hayan ocurrido :-)
  - ... que nosotros nunca sepamos interpretar :-(



# En la práctica: Invarianza



## Invarianza por entrenamiento

LeNet5 utiliza conocimiento del problema para diseñar la estructura de la red (invarianza por estructura):

82 errores (0.82%)

Ciresan et al. (2010) crearon **cuidadosamente** nuevos ejemplos de entrenamiento aplicando distintos tipos de transformaciones, tras lo que entrenaron una “deep, dumb net” usando una GPU:

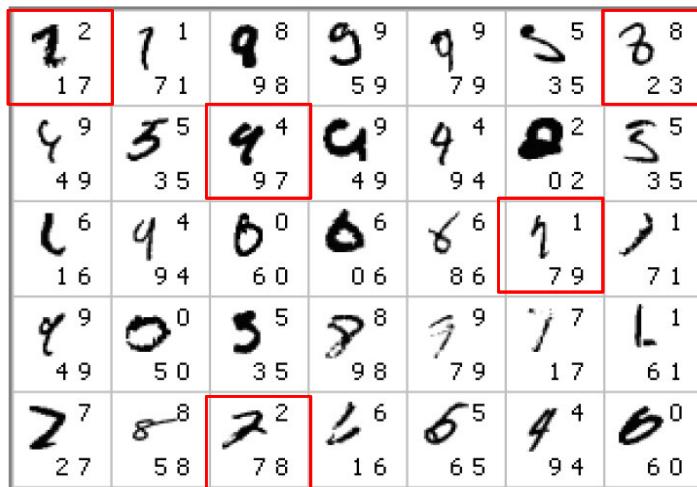
35 errores (0.35%)



# En la práctica: Invarianza



## Invarianza por entrenamiento



EJEMPLO: Los 35 errores de Ciresan et al. (2010)

<http://arxiv.org/abs/1003.0358>



# En la práctica: Invarianza



## Invarianza por entrenamiento

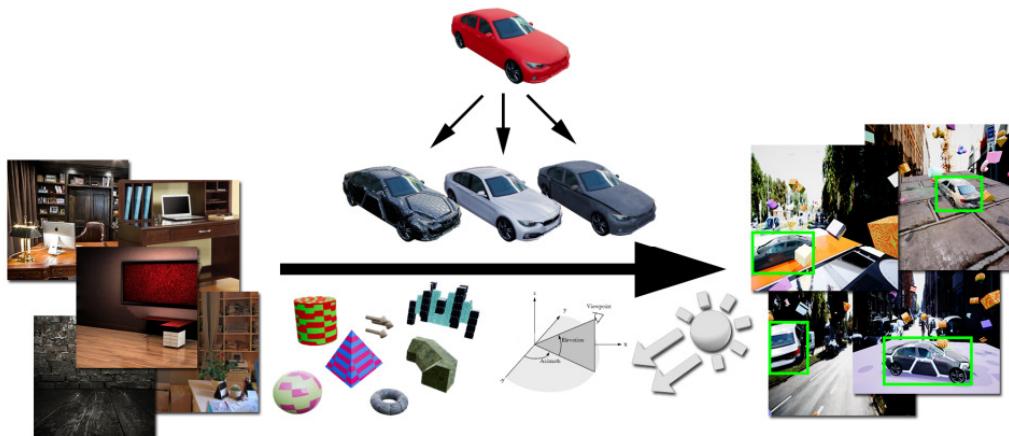


Figure 1. Domain randomization for object detection. Synthetic objects (in this case cars, top-center) are rendered on top of a random background (left) along with random flying distractors (geometric shapes next to the background images) in a scene with random lighting from random viewpoints. Before rendering, random texture is applied to the objects of interest as well as to the flying distractors. The resulting images, along with automatically-generated ground truth (right), are used for training a deep neural network.

**Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization.** CVPR 2018 Workshop on Autonomous Driving <https://arxiv.org/abs/1804.06516>





**DECSAI**

**Departamento de Ciencias de la Computación e I.A.**

Universidad de Granada



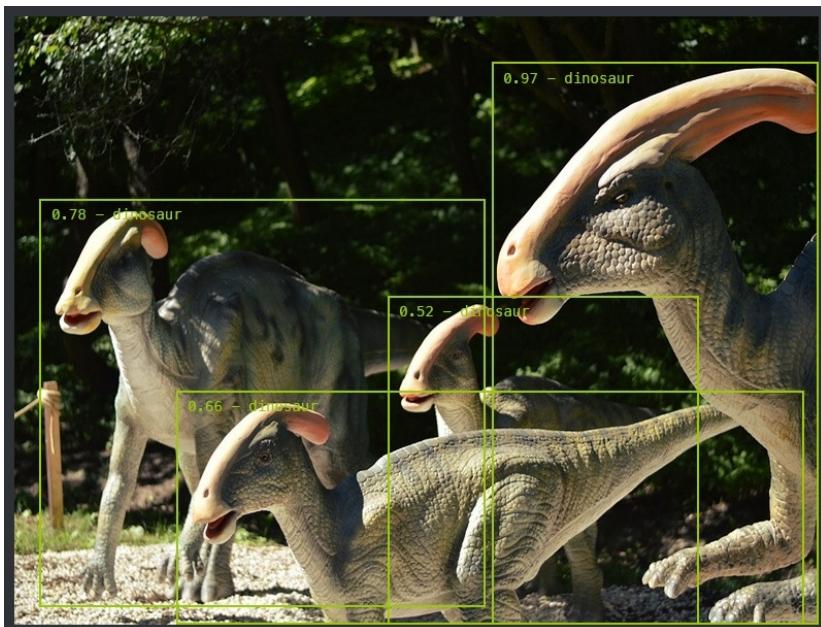
# Redes convolutivas

Aplicaciones: Detección de objetos

## Apéndice Detección de objetos



### El problema

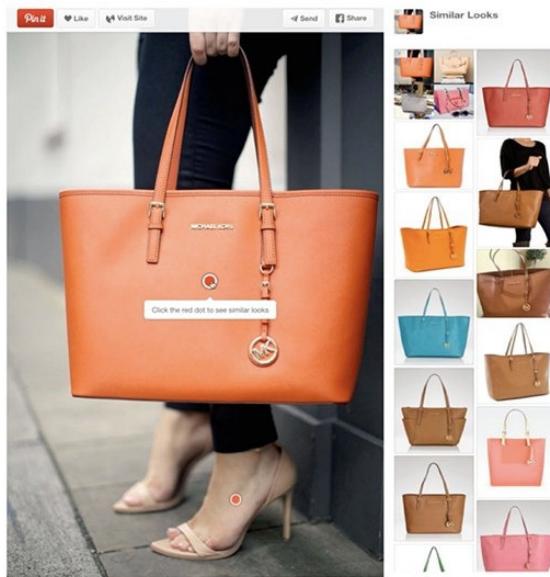


<https://tryolabs.com/blog/2017/08/30/object-detection-an-overview-in-the-age-of-deep-learning>

# Apéndice Detección de objetos



## Aplicaciones: Visual Search Engine (Pinterest)



[https://labs.pinterest.com/assets/paper/visual\\_search\\_at\\_pinterest.pdf](https://labs.pinterest.com/assets/paper/visual_search_at_pinterest.pdf)

<https://tryolabs.com/blog/2017/08/30/object-detection-an-overview-in-the-age-of-deep-learning>



# Apéndice Detección de objetos



## Aplicaciones: Análisis de imágenes aéreas



<https://tensorflight.com/>

<https://tryolabs.com/blog/2017/08/30/object-detection-an-overview-in-the-age-of-deep-learning>

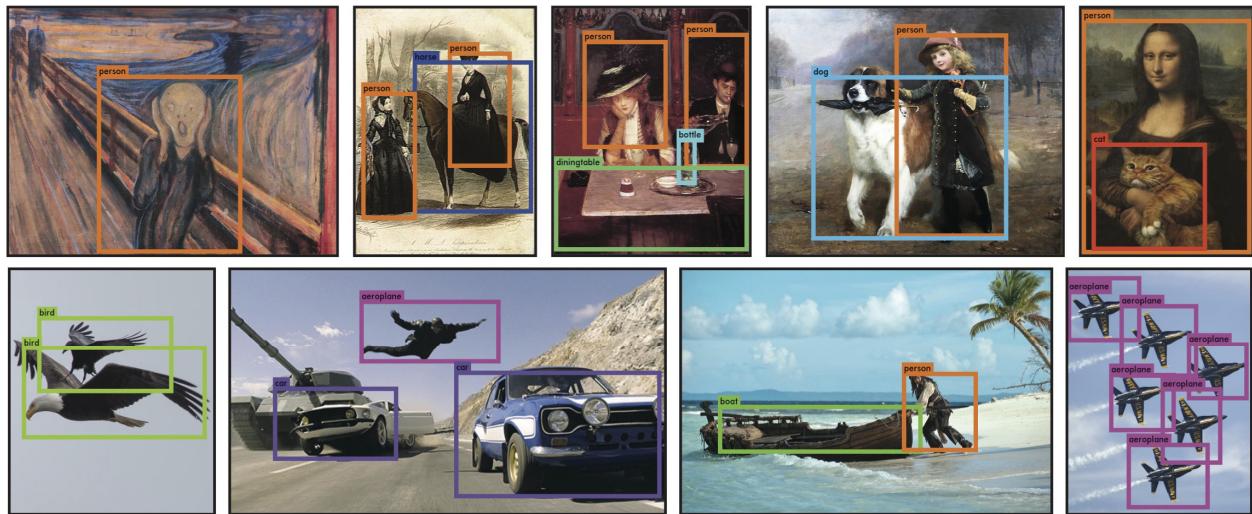


# Apéndice

# Detección de objetos



## Problema técnico:



¿Número de objetos, tamaño y posición?



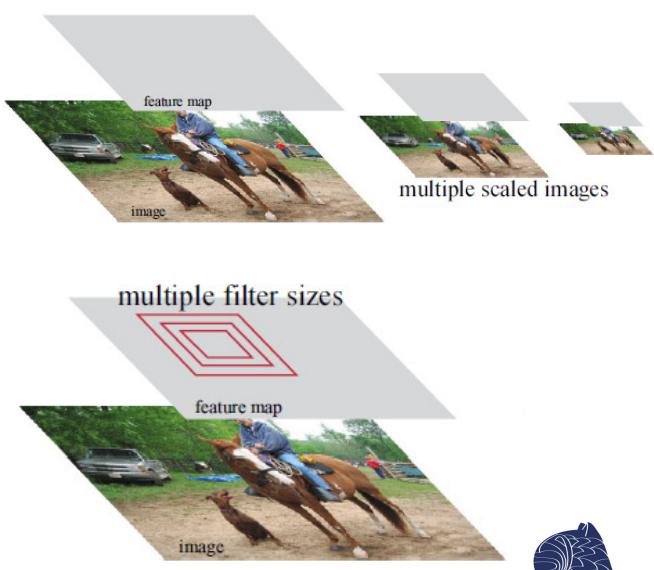
# Apéndice

# Detección de objetos



## Una primera solución

### Ventana deslizante



# Apéndice

# Detección de objetos

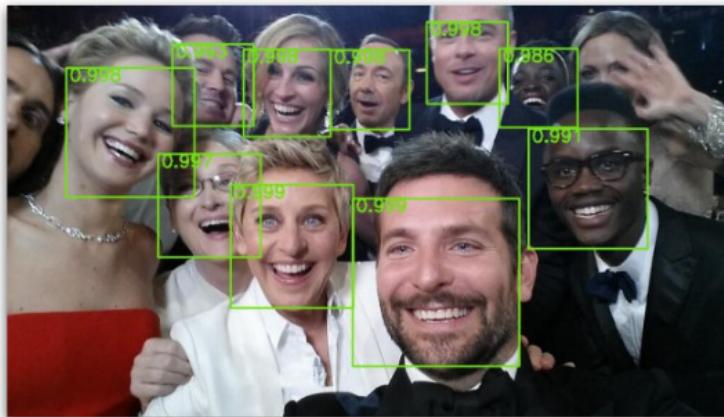


## Soluciones clásicas

Paul Viola & Michael Jones (Compaq):

"Robust Real-time Object Detection", 2001

- Miles de clasificadores utilizando características Haar.
- Detección de caras en tiempo real (p.ej. cámaras)



# Apéndice

# Detección de objetos



## Soluciones clásicas

HOG [Histogram of Oriented Gradients]

U.S. Patent 1986 – CVPR'2005

e.g. Detección de peatones



Input image

Histogram of Oriented Gradients



Input example

Average gradients



# Apéndice

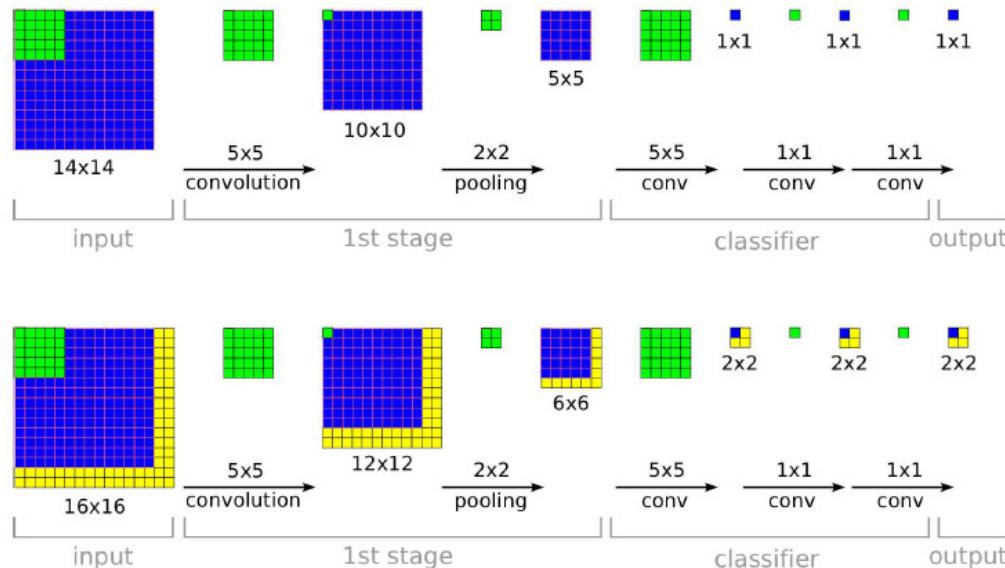
# Detección de objetos



## Soluciones basadas en “deep learning”

### OverFeat (NYU’2013)

Multi-scale sliding window using CNNs



# Apéndice

# Detección de objetos

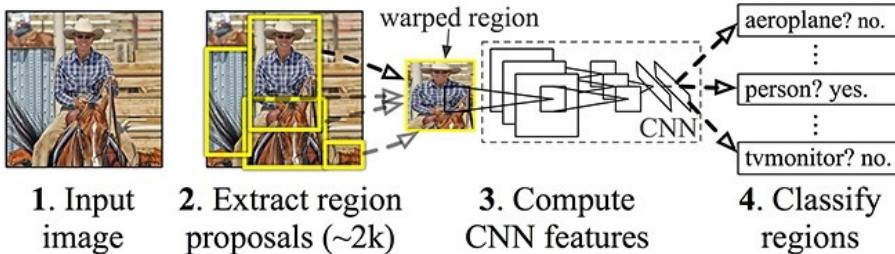


## Soluciones basadas en “deep learning”

### R-CNN (UC Berkeley ’2014)

Regions with CNN features

R-CNN: *Regions with CNN features*



# Apéndice

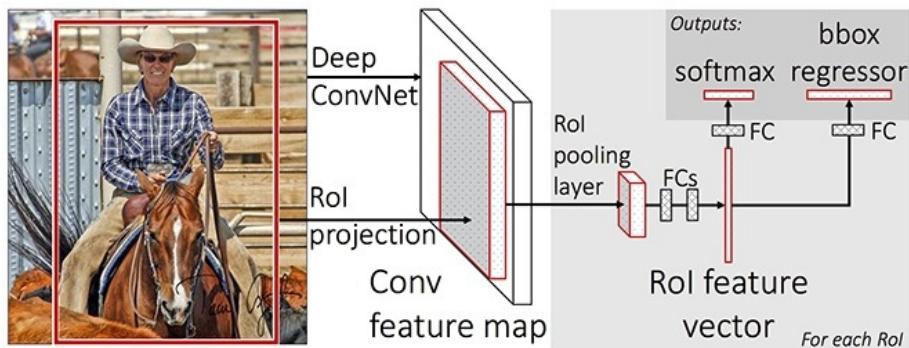
# Detección de objetos



**Soluciones basadas en “deep learning”**

**Fast R-CNN (Microsoft Research ‘2014)**

Region of Interest (RoI) Pooling



# Apéndice

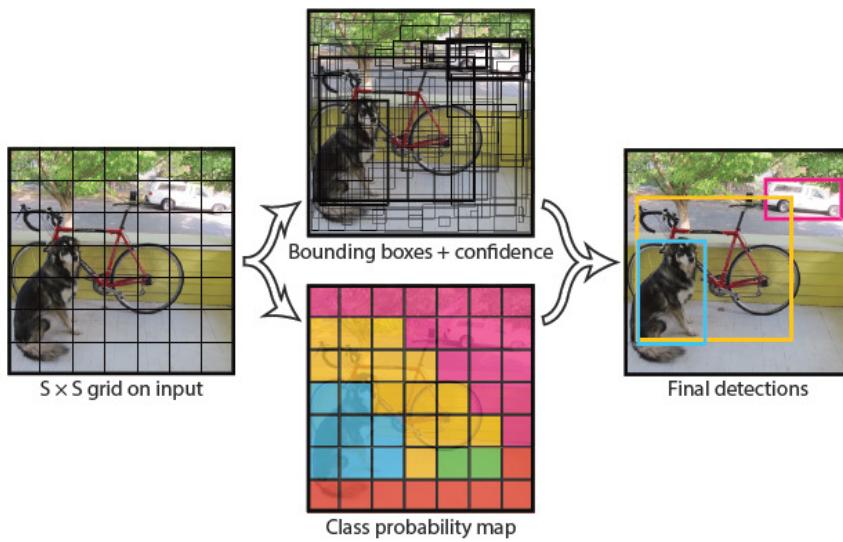
# Detección de objetos



**Soluciones basadas en “deep learning”**

**YOLO: You Only Look Once ‘2015**

El problema de detección como un problema de regresión



# Apéndice

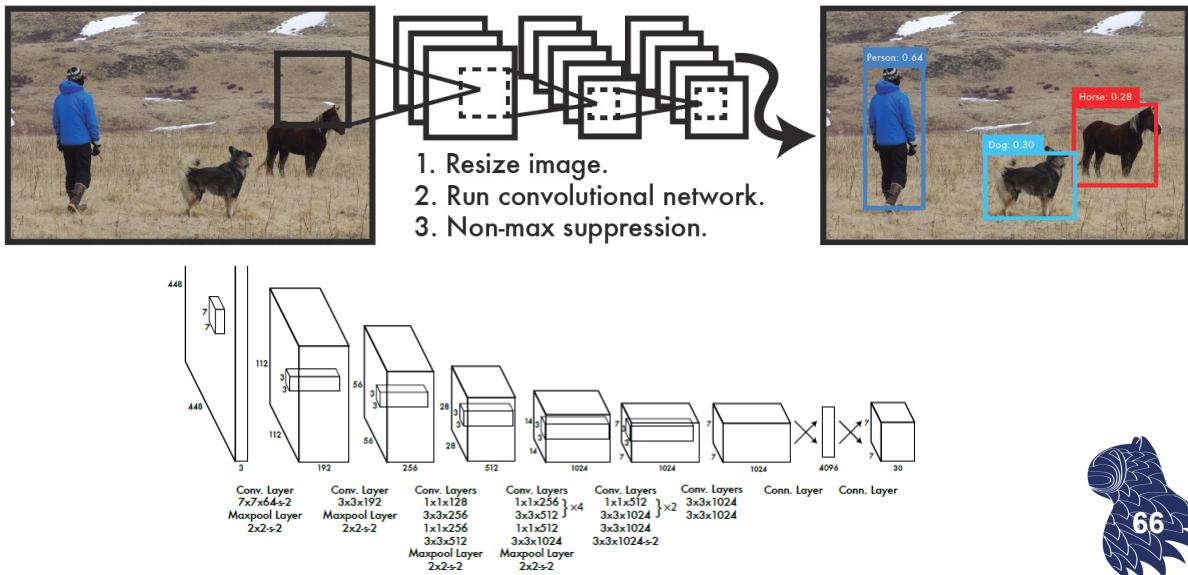
# Detección de objetos



## Soluciones basadas en “deep learning”

### YOLO: You Only Look Once ‘2015

Red convolutiva única



# Apéndice

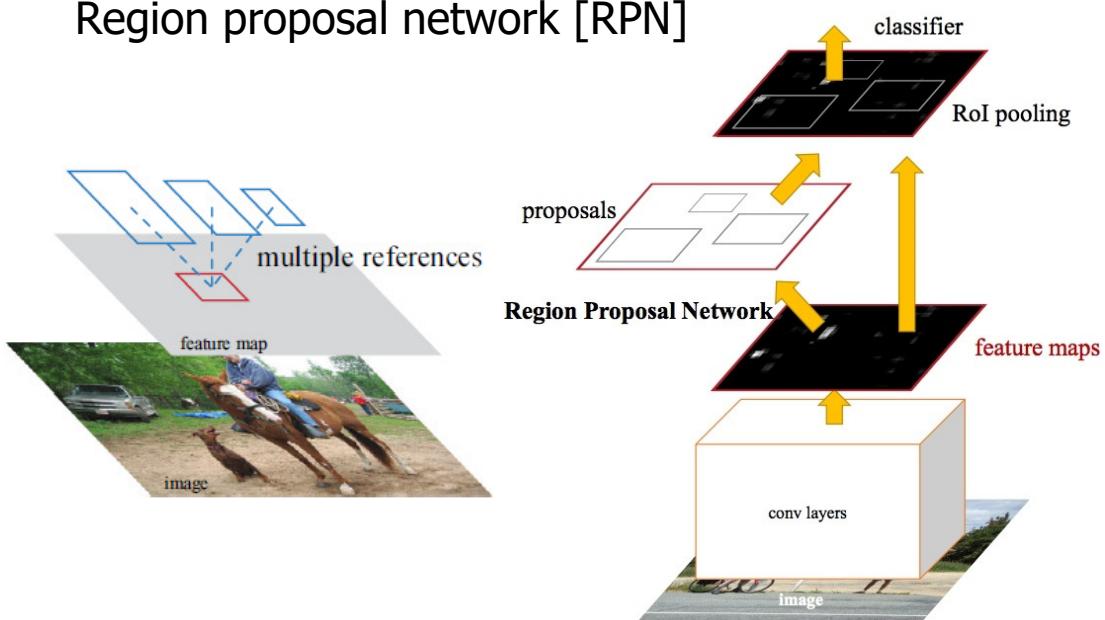
# Detección de objetos



## Soluciones basadas en “deep learning”

### Faster R-CNN (Microsoft Research ‘2016)

Region proposal network [RPN]



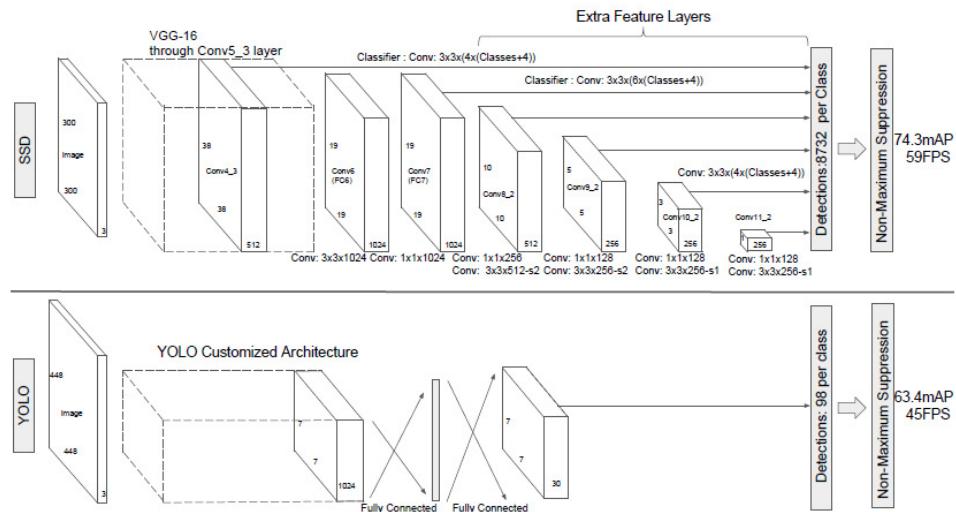
# Apéndice

# Detección de objetos



## Soluciones basadas en “deep learning”

### Más variantes... de YOLO



SSD [Single Shot Detector]



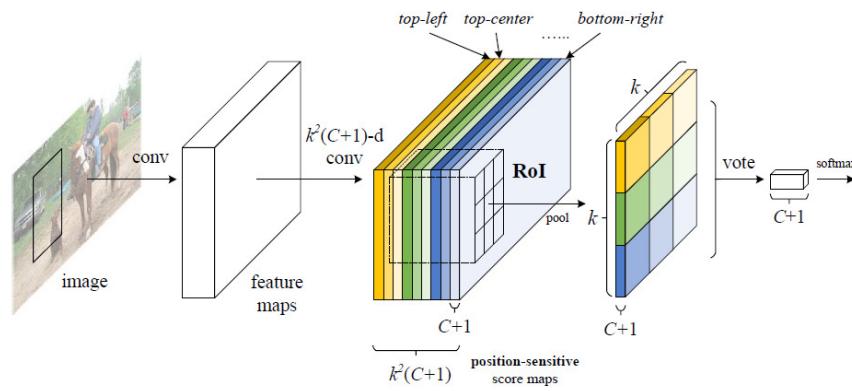
# Apéndice

# Detección de objetos



## Soluciones basadas en “deep learning”

### Más variantes... de Faster R-CNN



R-FCN [Region-based Fully Convolutional Network]



# Apéndice

# Detección de objetos



## Tutoriales y demos

### Técnicas tradicionales

- HOG [Histogram of Oriented Gradients]

<https://www.learnopencv.com/histogram-of-oriented-gradients/>

### Técnicas basadas en deep learning

- YOLO Object Detection

<https://pjreddie.com/darknet/yolo/>

[https://www.youtube.com/watch?v=4eIBisqx9\\_q](https://www.youtube.com/watch?v=4eIBisqx9_q)

[https://github.com/lISourcell/YOLO\\_Object\\_Detection](https://github.com/lISourcell/YOLO_Object_Detection)



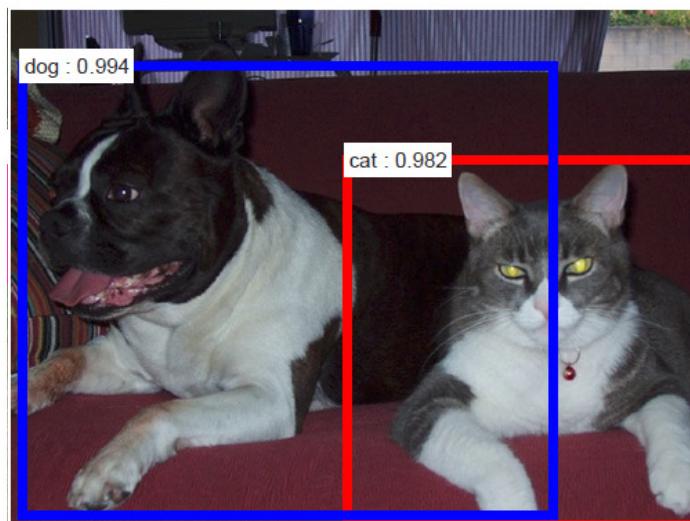
# Apéndice

# Detección de objetos



## Soluciones basadas en “deep learning”

### Resultados



# Apéndice

# Detección de objetos



## Soluciones basadas en “deep learning”

### Resultados



## Cursos

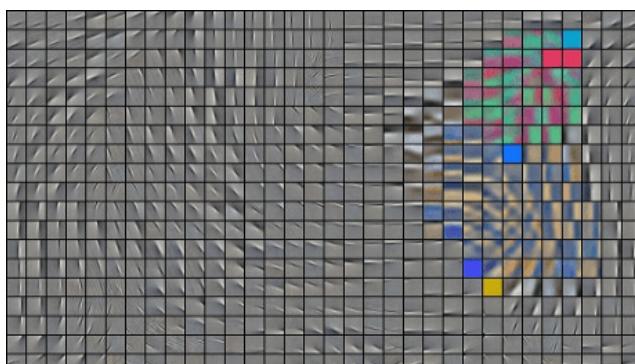


### Neural Networks for Machine Learning

by Geoffrey Hinton

(University of Toronto & Google)

<https://www.coursera.org/course/neuralnets>



# Cursos



## Deep Learning Specialization

by Andrew Ng, 2017

- Neural Networks and Deep Learning
- Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization
- Structuring Machine Learning Projects
- Convolutional Neural Networks
- Sequence Models



deeplearning.ai



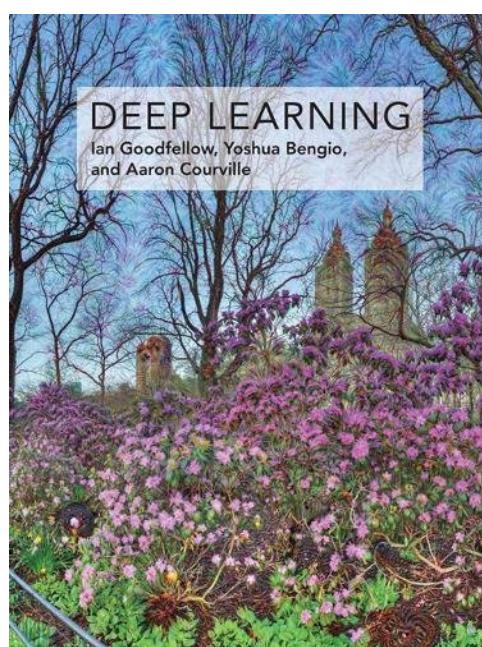
<https://www.coursera.org/specializations/deep-learning>



# Bibliografía

## Lecturas recomendadas

Ian Goodfellow,  
Yoshua Bengio  
& Aaron Courville:  
**Deep Learning**  
MIT Press, 2016  
ISBN 0262035618



<http://www.deeplearningbook.org>



75

# Bibliografía



## Lecturas recomendadas

Fernando Berzal:  
**Redes Neuronales  
& Deep Learning**

CAPÍTULO 13  
**Redes convolutivas**

