

Aprendizaje profundo

REDES CONVOLUCIONALES

Gibran Fuentes-Pineda

Septiembre 2023

Red neuronal convolucional

- Compuesta de dos bloques principales
 1. *Extracción de características*: múltiples capas convolucionales y de submuestreo
 2. *Clasificación*: Una o más capas completamente conectadas (incluyendo la capa de salida)

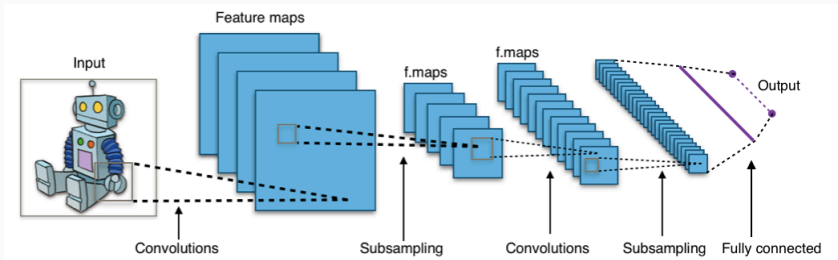


Imagen del usuario Aphex34 de Wikipedia. CC BY-SA 4.0

Arquitecturas de redes convolucionales: LeNet

- Red poco profunda inspirada en la propuesta originalmente por LeCun et al. 1998

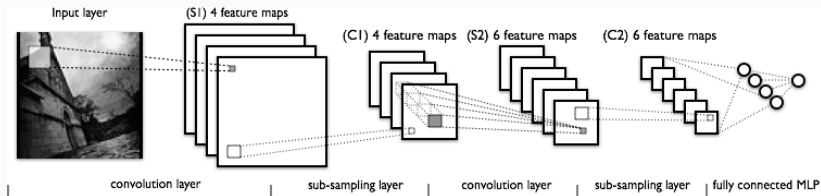


Imagen tomada de <http://deeplearning.net/tutorial/lenet.html>

Arquitecturas de redes convolucionales: AlexNet

- Usa función de activación ReLU
- Entrenamiento con version optimizada para 2 GPUs
- Normaliza respuestas
- Submuestreo con traslape

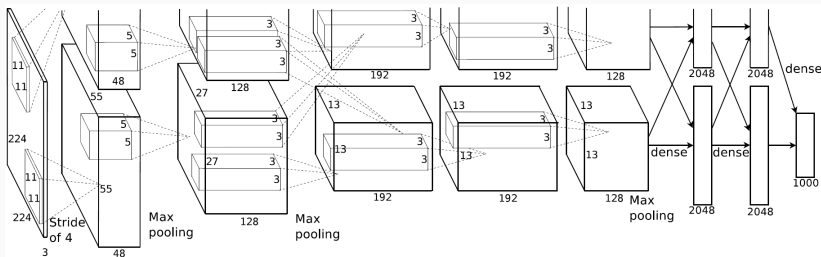


Imagen tomada de Krizhevsky et al. *ImageNet Classification with Deep Convolutional Neural Networks*, 2012

Arquitecturas de redes convolucionales: ZFNet

- Muy similar a AlexNet pero usa filtros más pequeños con menor desplazamiento

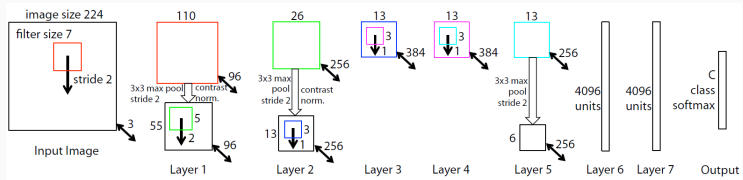


Imagen tomada de Zeiler and Fergus. *Visualizing and Understanding Convolutional Networks*, 2014

Arquitecturas de redes convolucionales: VGGNet

- 19 capas
- Filtros de 3×3 con desplazamientos de 1
- Max-pooling de 2×2 con desplazamientos de 2



Imagen tomada de diapositivas de Simonyan (ILSVRC Workshop 2014)

Arquitecturas de redes convolucionales: GoogleNet (1)

- 22 capas
- Utiliza bloque de capas convolucionales paralelas cuyas salidas se concatenan

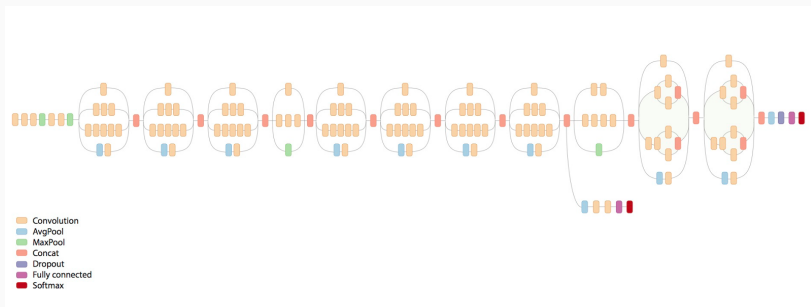


Imagen tomada de <https://research.googleblog.com/2016/03/train-your-own-image-classifier-with.html>

Arquitecturas de redes convolucionales: GoogleNet (2)

type	patch size/ stride	output size	depth	# 1 × 1	# 3 × 3 reduce	# 3 × 3	# 5 × 5 reduce	# 5 × 5	pool proj	params	ops
convolution	7 × 7 / 2	112 × 112 × 64	1							2.7K	34M
max pool	3 × 3 / 2	56 × 56 × 64	0								
convolution	3 × 3 / 1	56 × 56 × 192	2		64	192				112K	360M
max pool	3 × 3 / 2	28 × 28 × 192	0								
inception (3a)		28 × 28 × 256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28 × 28 × 480	2	128	128	192	32	96	64	380K	304M
max pool	3 × 3 / 2	14 × 14 × 480	0								
inception (4a)		14 × 14 × 512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14 × 14 × 512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14 × 14 × 512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14 × 14 × 528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14 × 14 × 832	2	256	160	320	32	128	128	840K	170M
max pool	3 × 3 / 2	7 × 7 × 832	0								
inception (5a)		7 × 7 × 832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7 × 7 × 1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7 × 7 / 1	1 × 1 × 1024	0								
dropout (40%)		1 × 1 × 1024	0								
linear		1 × 1 × 1000	1							1000K	1M
softmax		1 × 1 × 1000	0								

Tabla tomada de Szegedy et al. Going deeper with convolutions, 2014

Bloques Inception

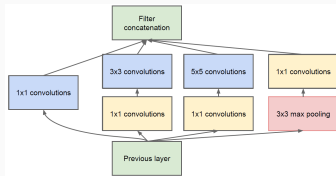
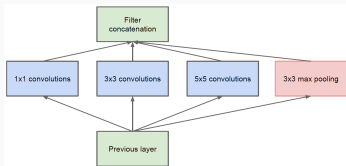


Imagen tomada de Szegedy et al. Going deeper with convolutions, 2014

Entrenando redes más profundas: ResNet (1)

- Reformula los mapeos que se desean aprender a residuales
- Más fácil de optimizar los residuales que el mapeo original¹

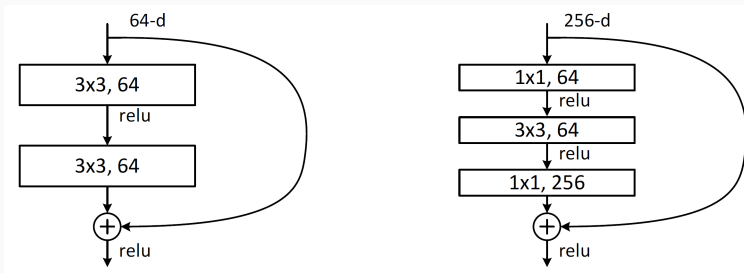


Imagen tomada de He et al. Deep Residual Learning for Image Recognition, 2015

¹Se ha mostrado que son aproximadores universales: Lin y Jegelka. ResNet with one-neuron hidden layers is a Universal Approximator, 2018.

Entrenando redes más profundas: ResNet (2)

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

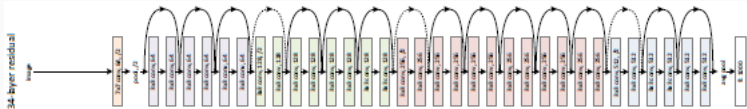


Imagen y tabla tomadas de He et al. *Deep Residual Learning for Image Recognition*, 2015

Entrenando redes más profundas: ResNet (3)

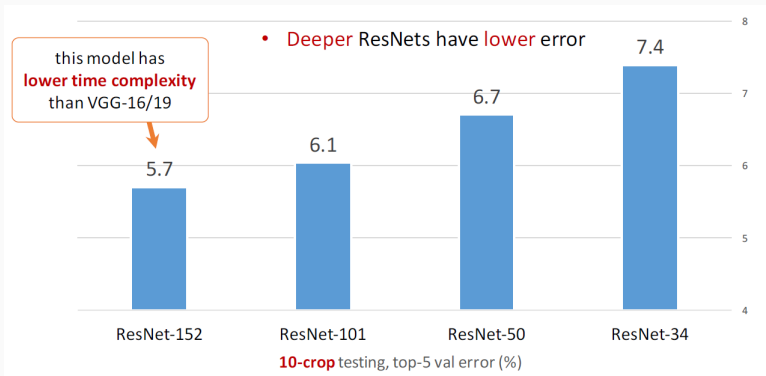


Imagen tomada de diapositivas de Kaiming He (ICML 2016)

Entrenando redes más profundas: DenseNet (1)

- Utiliza bloques que conectan la salida de cada capa con las entradas de capas subsecuentes.

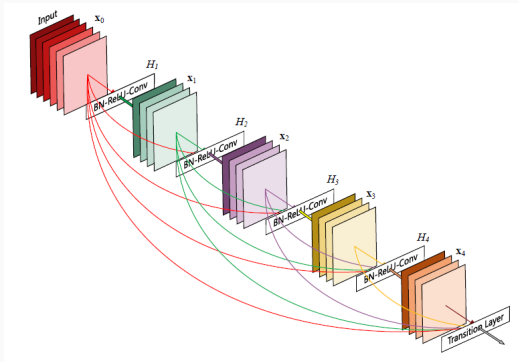


Imagen tomada de Huang et al. Densely Connected Convolutional Networks, 2018

Entrenando redes más profundas: DenseNet (2)

Layers	Output Size	DenseNet-121 ($k = 32$)	DenseNet-169 ($k = 32$)	DenseNet-201 ($k = 32$)	DenseNet-161 ($k = 48$)
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56 28×28	1×1 conv			
		2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28 14×14	1×1 conv			
		2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition Layer (3)	14×14 7×7	1×1 conv			
		2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification Layer	1×1	7×7 global average pool			
Layer		1000D fully-connected, softmax			

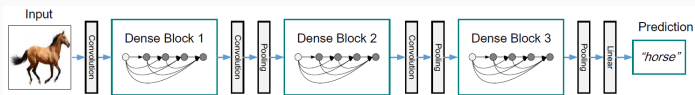


Imagen y tabla tomadas de Huang et al. Densely Connected Convolutional Networks, 2018