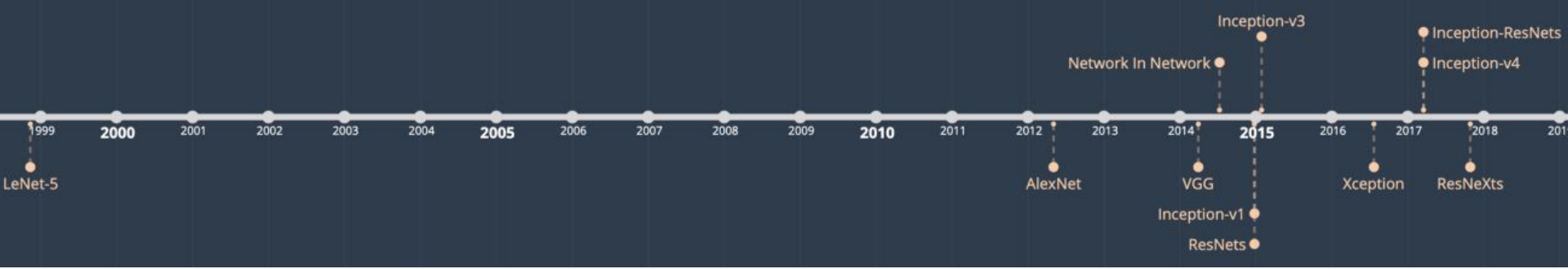
Arquitecturas CNNs

bibliografía

Andrew NG (Deep Learning)



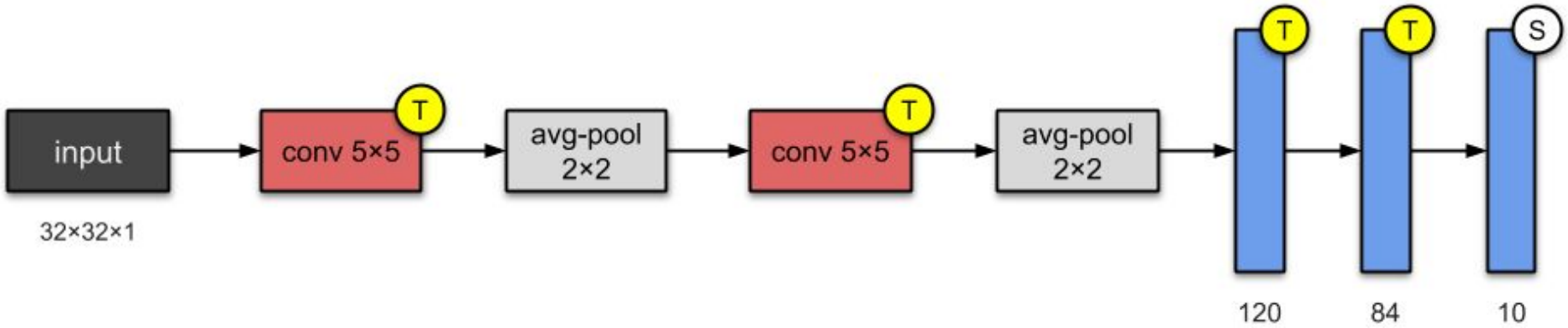
Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9

keras app

LeNet-5 (1998)

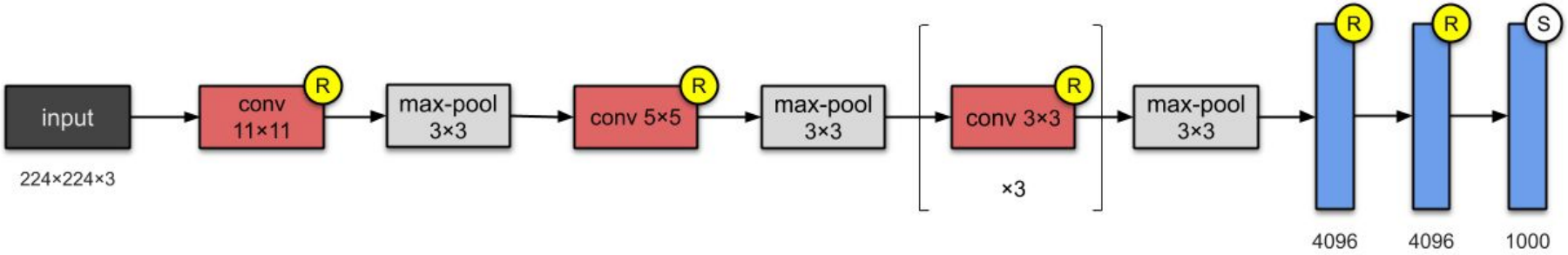
Stacking:

- Conv
- Activation
- Pooling
- Dense



AlexNet (2012) - Ver ejemplo en github 1-CNN-Teoria

- Relu
- Dropout



VGG-16 (2014)

Deeper network

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

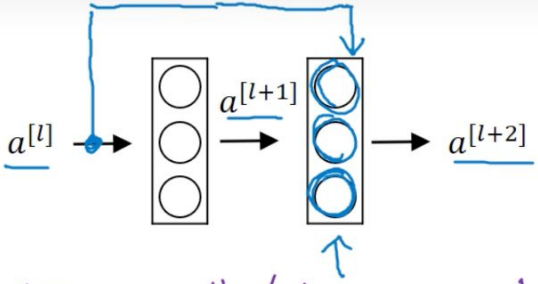
Resnet-50 (2015)

Popularised skip connections (they weren’t the first to use skip connections).

Deeper CNNs (up to 152 layers)

Batch normalization

Residual block



$a^{[l]}$ → Linear → ReLU → $a^{[l+1]}$ → Linear → (+) → ReLU → $a^{[l+2]}$

"main path"

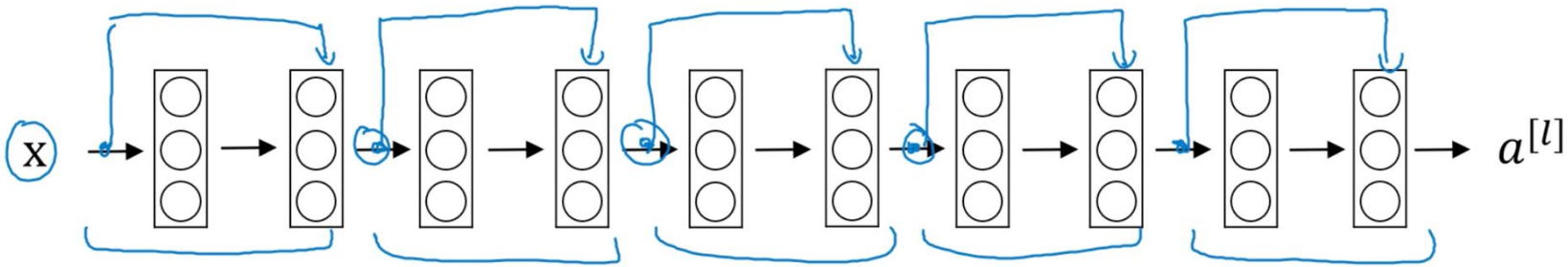
$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]} \qquad a^{[l+1]} = g(z^{[l+1]}) \qquad z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]} \qquad a^{[l+2]} = g(z^{[l+2]})$$

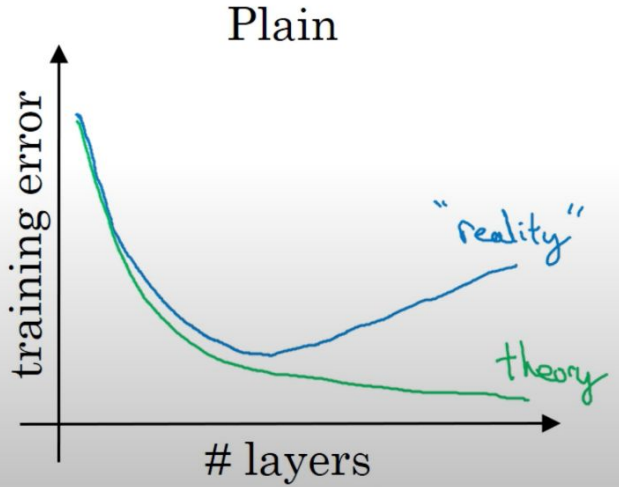
$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$$

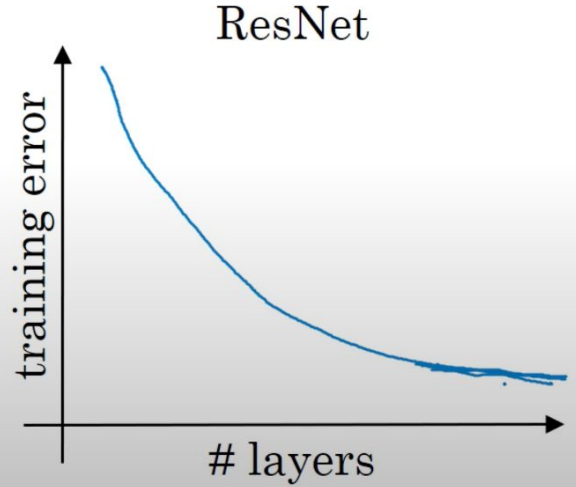
[He et al., 2015. Deep residual networks for image recognition]

Andrew Ng

Residual Network

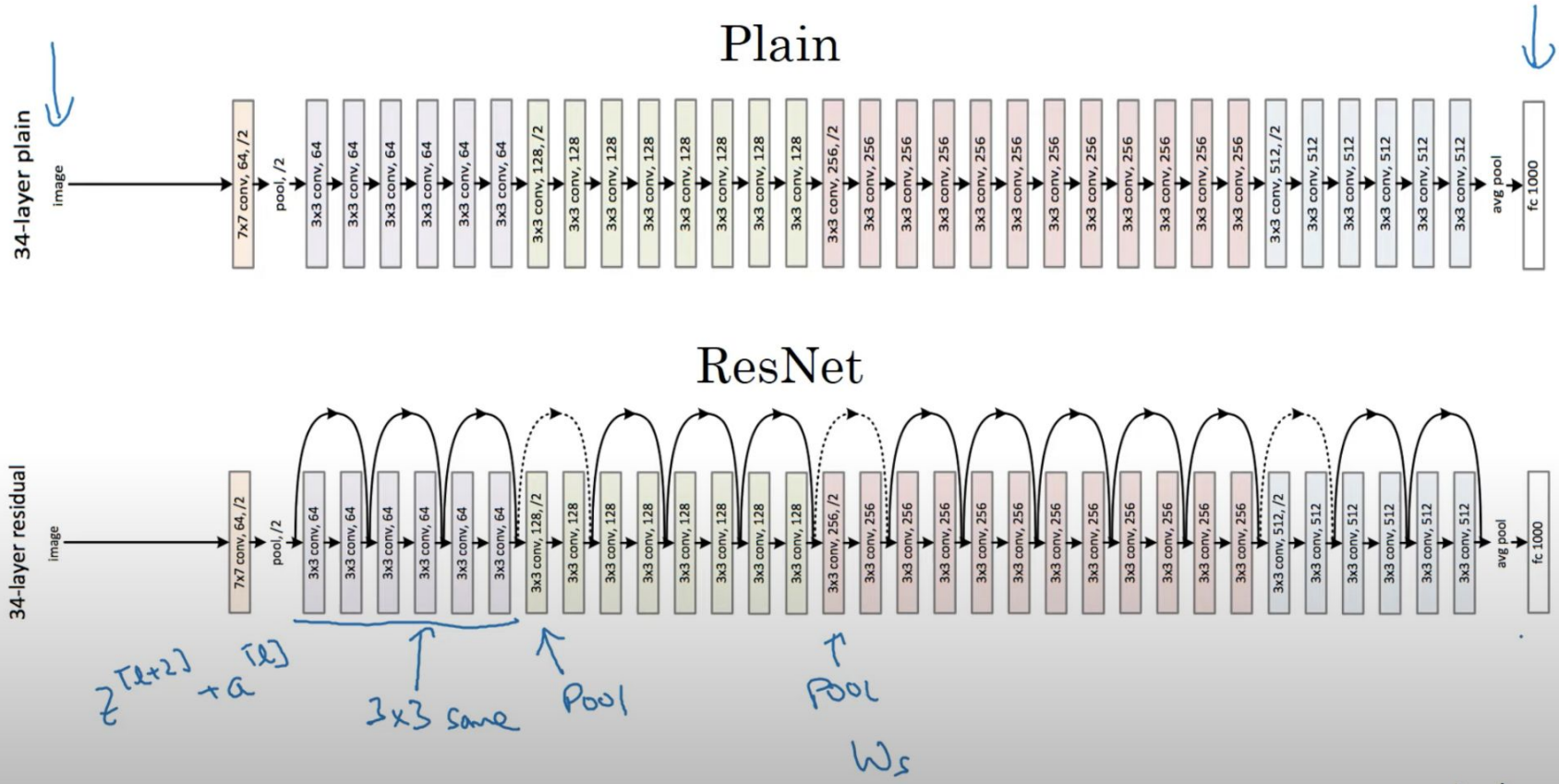






[He et al., 2015. Deep residual networks for image recognition]

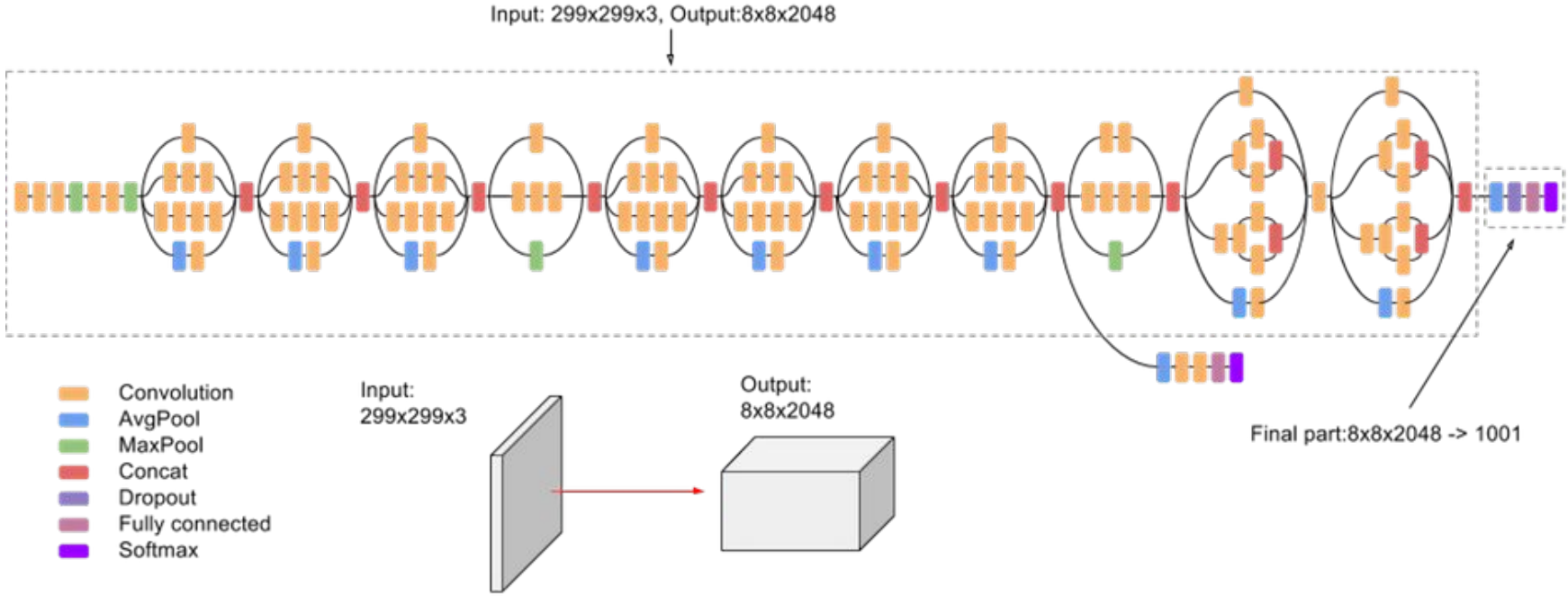
ResNet



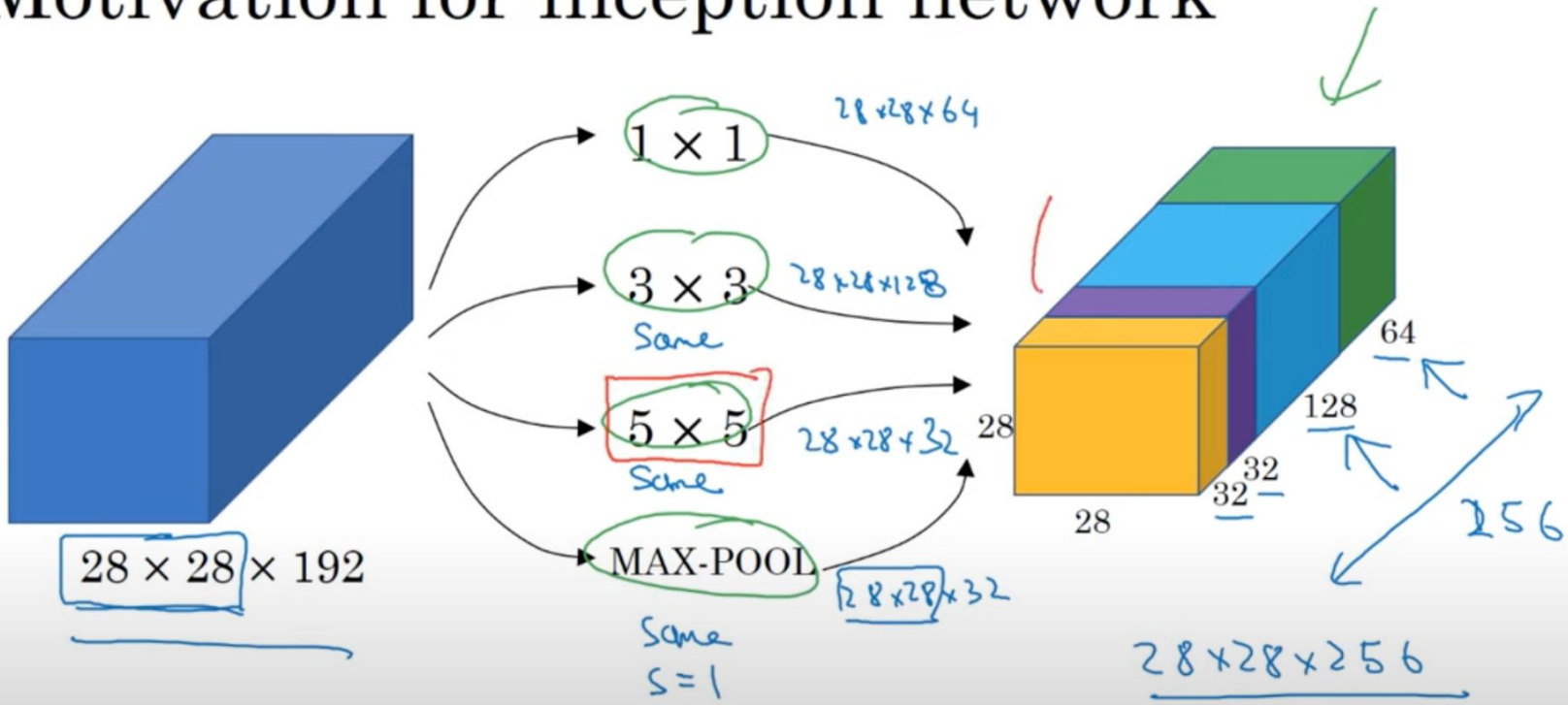
[He et al., 2015. Deep residual networks for image recognition]

Andrew Ng

Inception-V3 (2015)

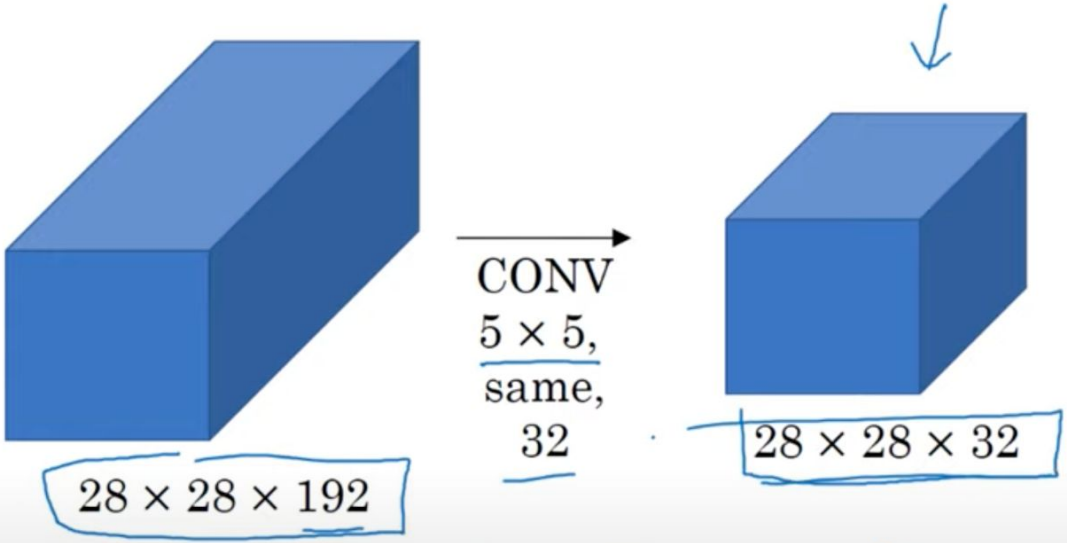


Motivation for inception network



[Szegedy et al. 2014. Going deeper with convolutions]

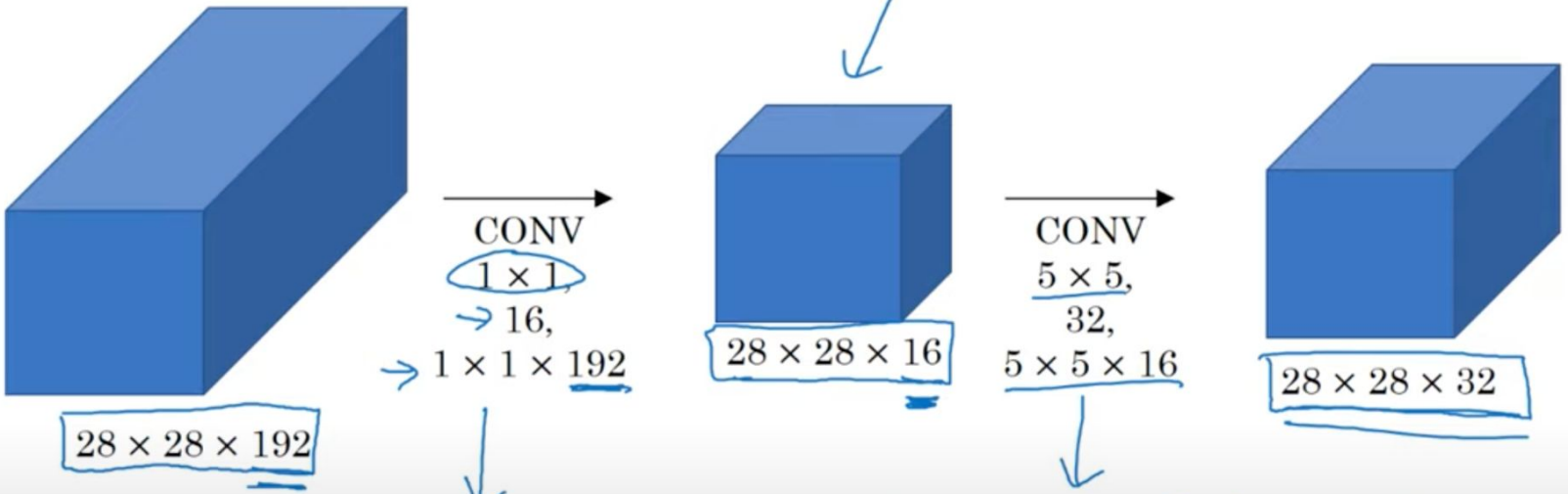
The problem of computational cost



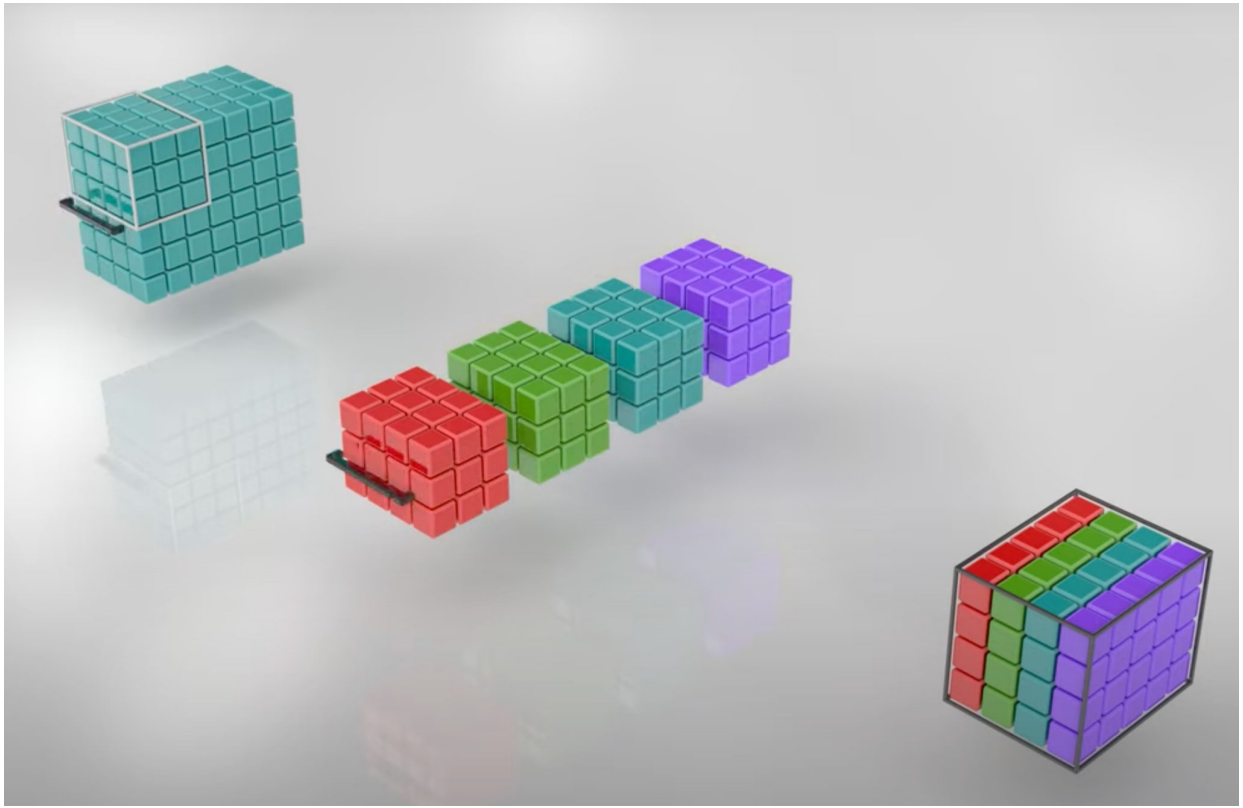
32 filters. filters are 5×5×192.

28×28×32 × 5×5×192 = 120M.

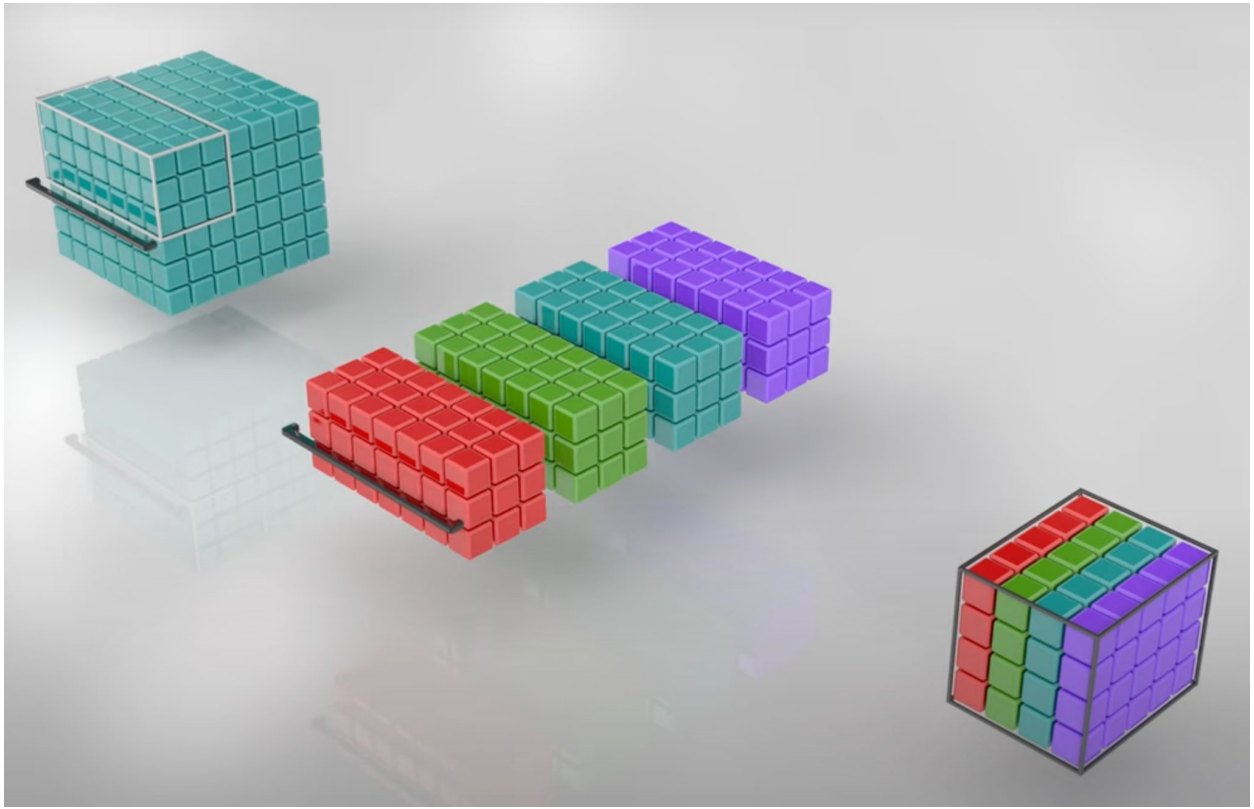
Using 1×1 convolution



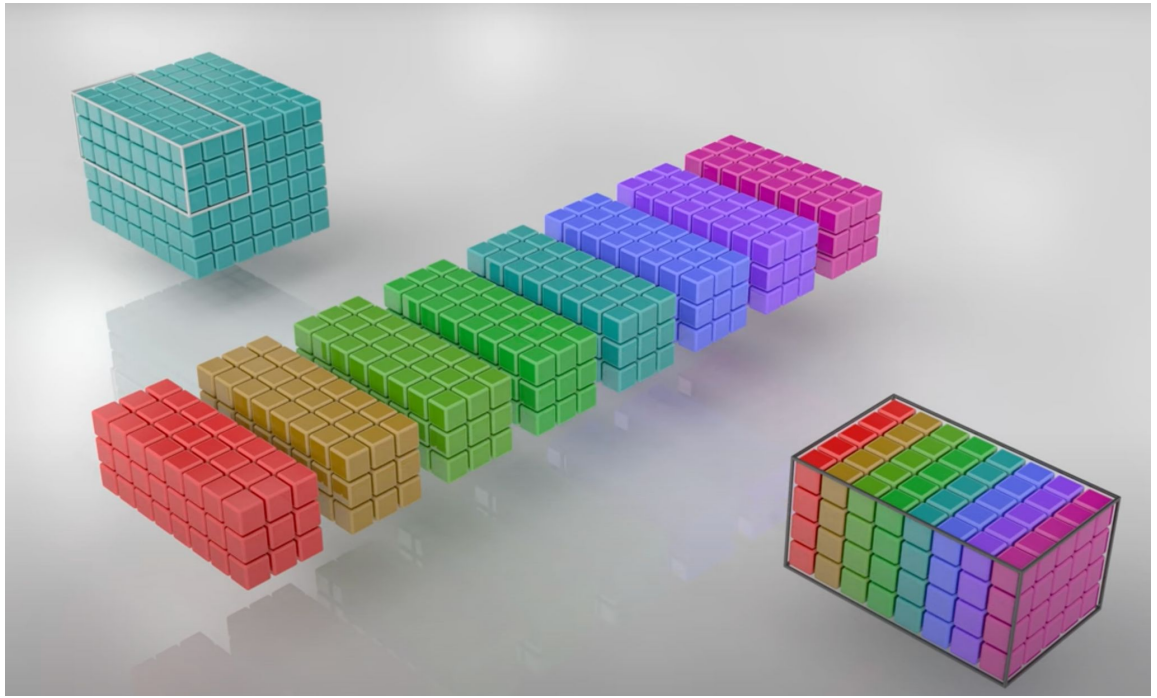
MobileNet / EfficientNet (Groups) Ver video



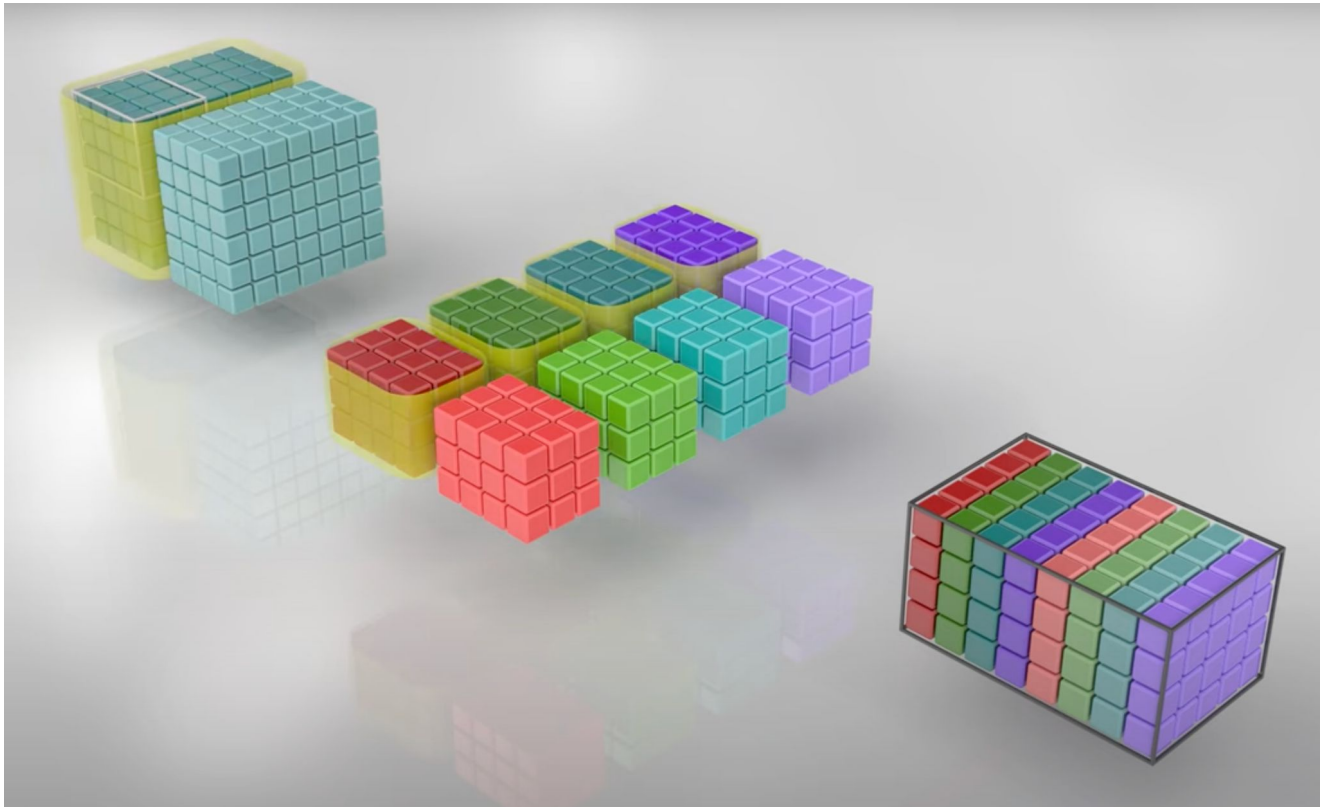
Aumento la cantidad de canales de entrada



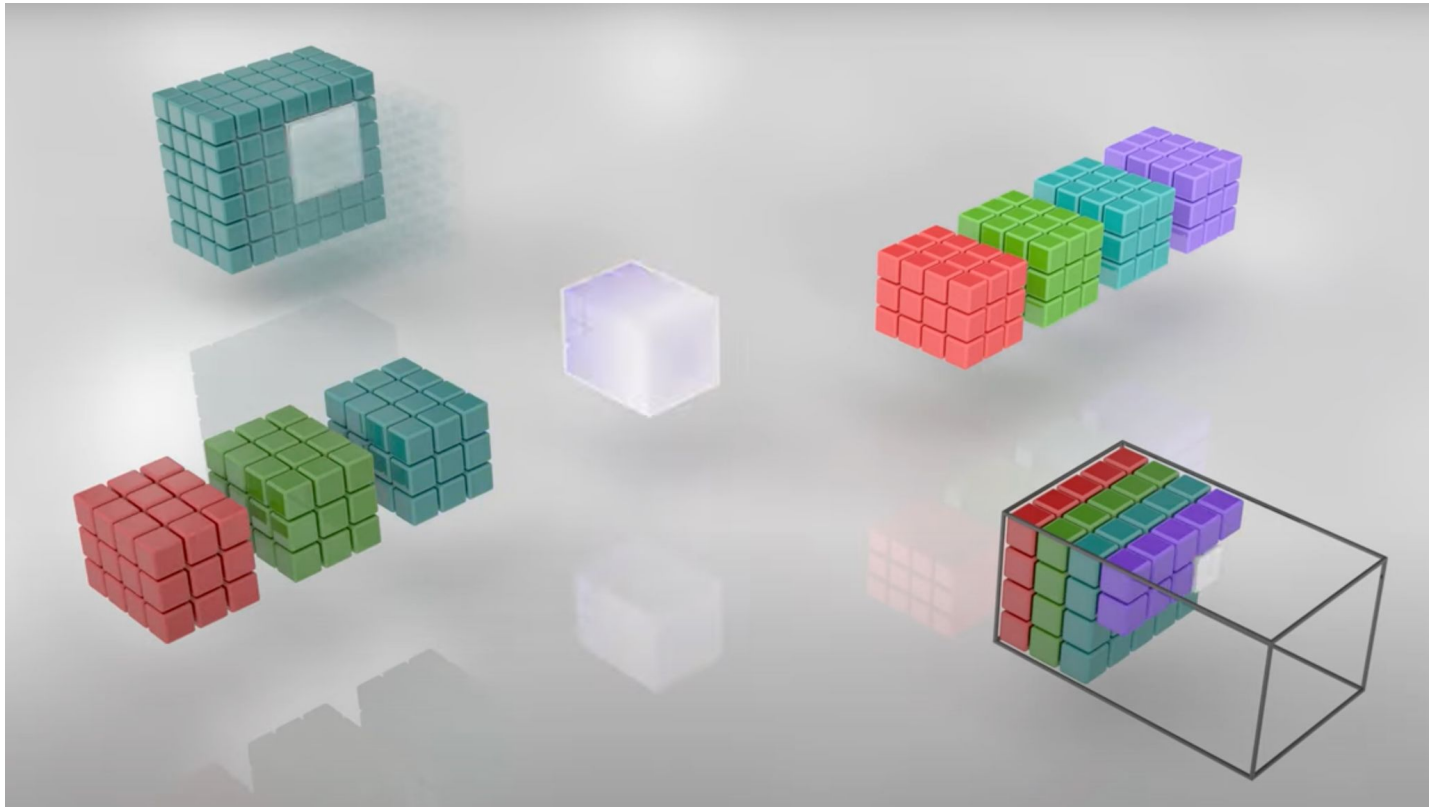
Aumento la cantidad de features (cantidad de filtros)



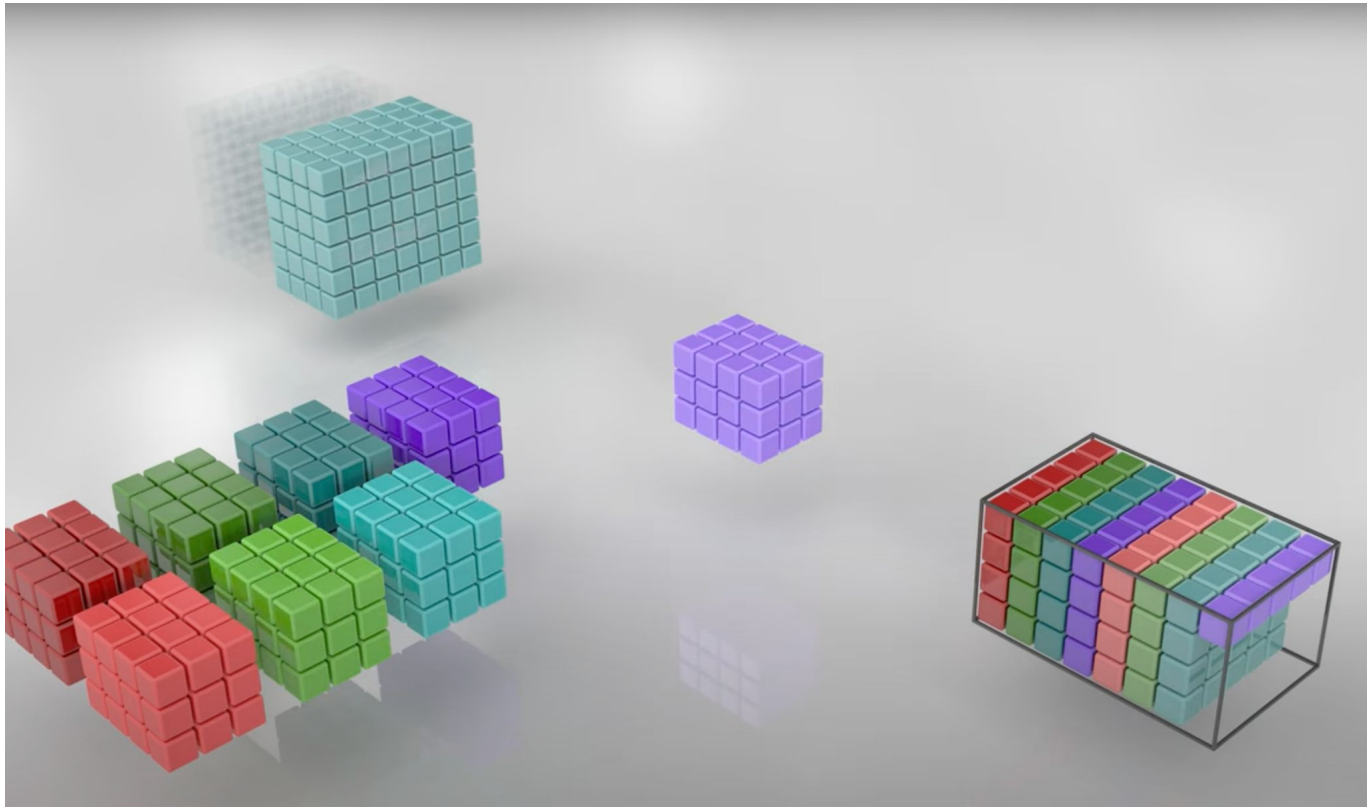
Dividimos en 2 la cantidad de canales y la cantidad de filtros



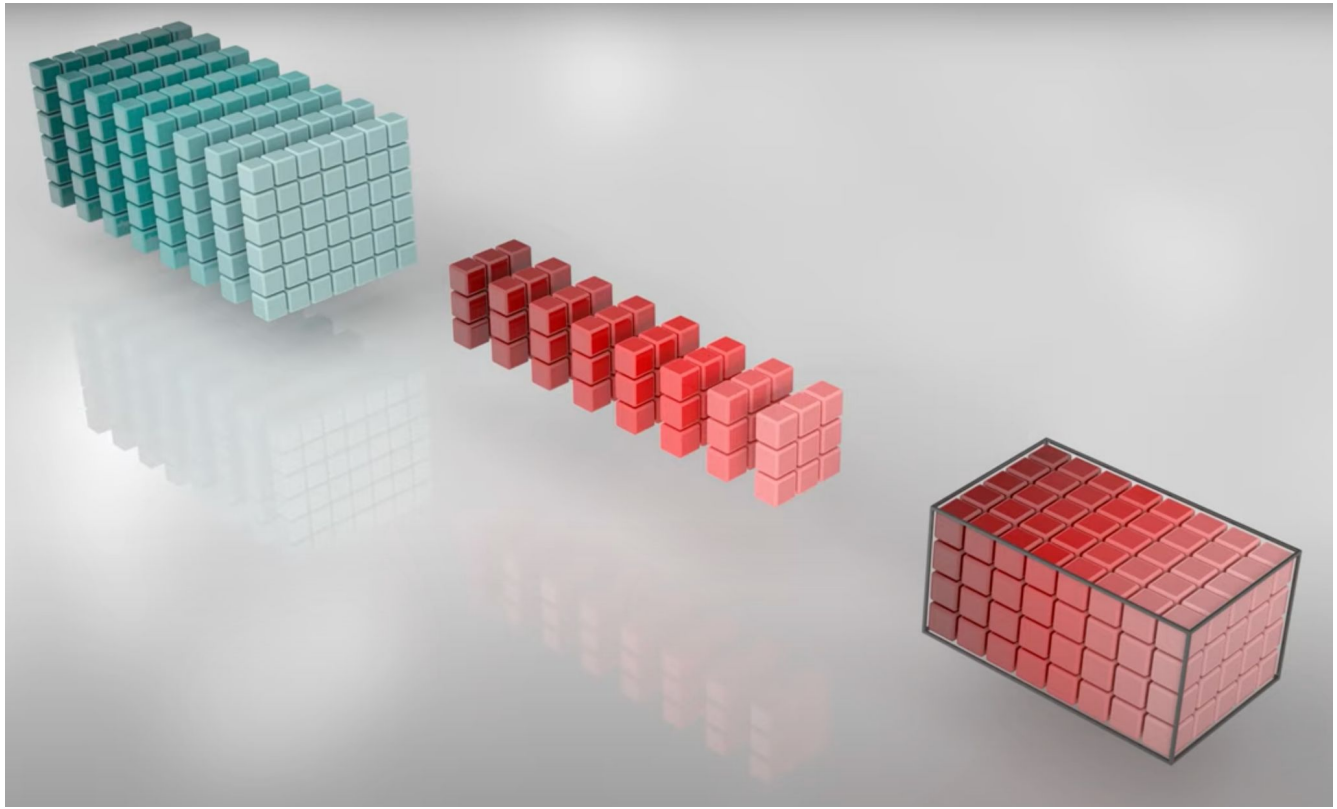
Cada filtro se calcula de manera independiente



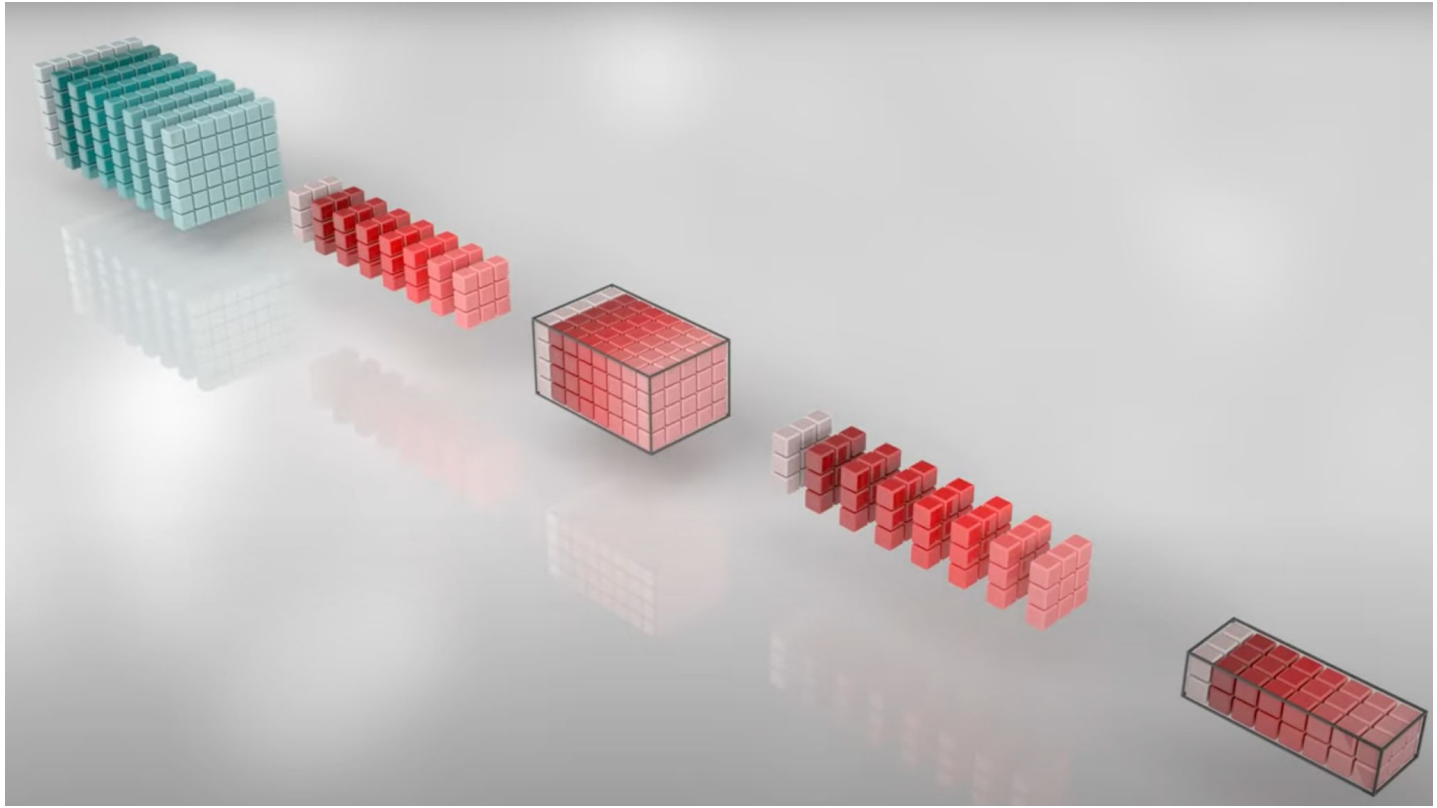
La otra mitad



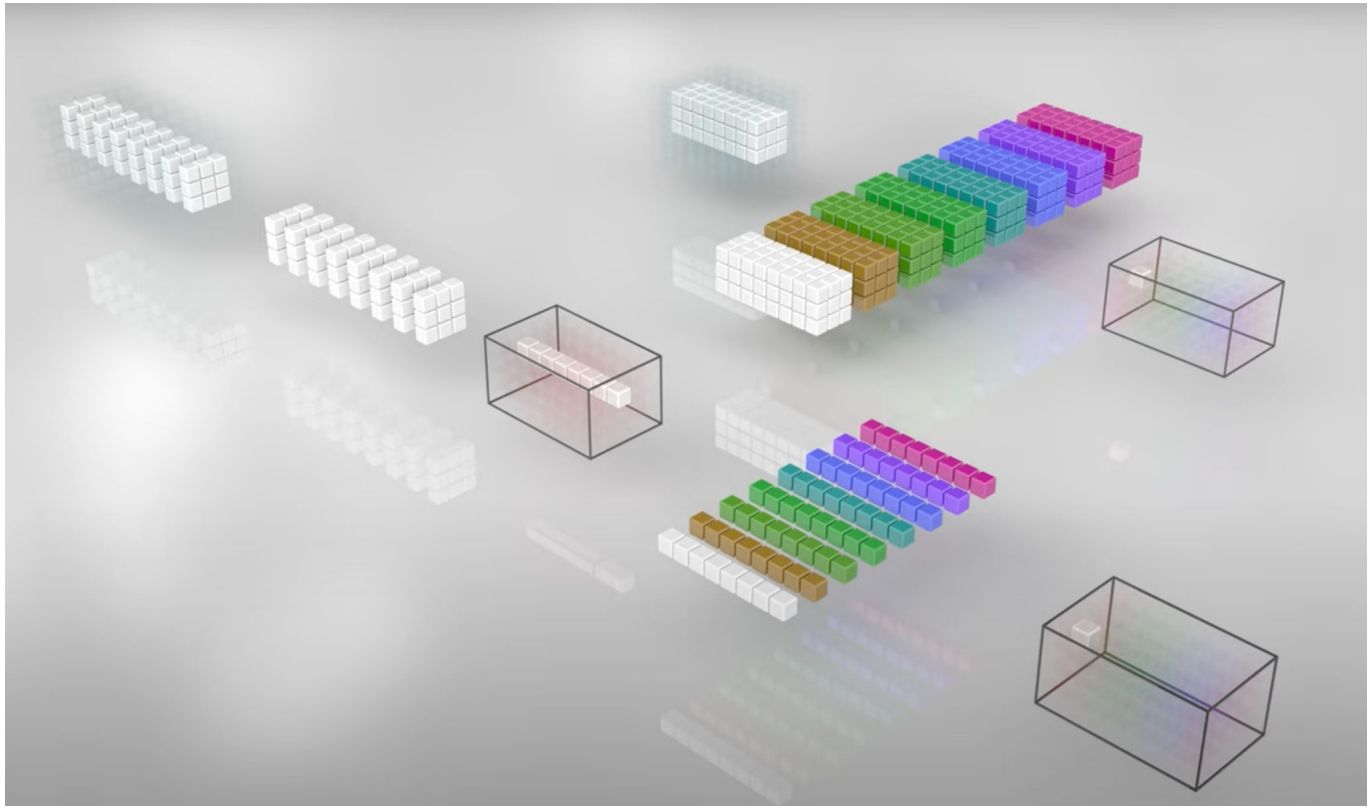
Depthwise convolutions

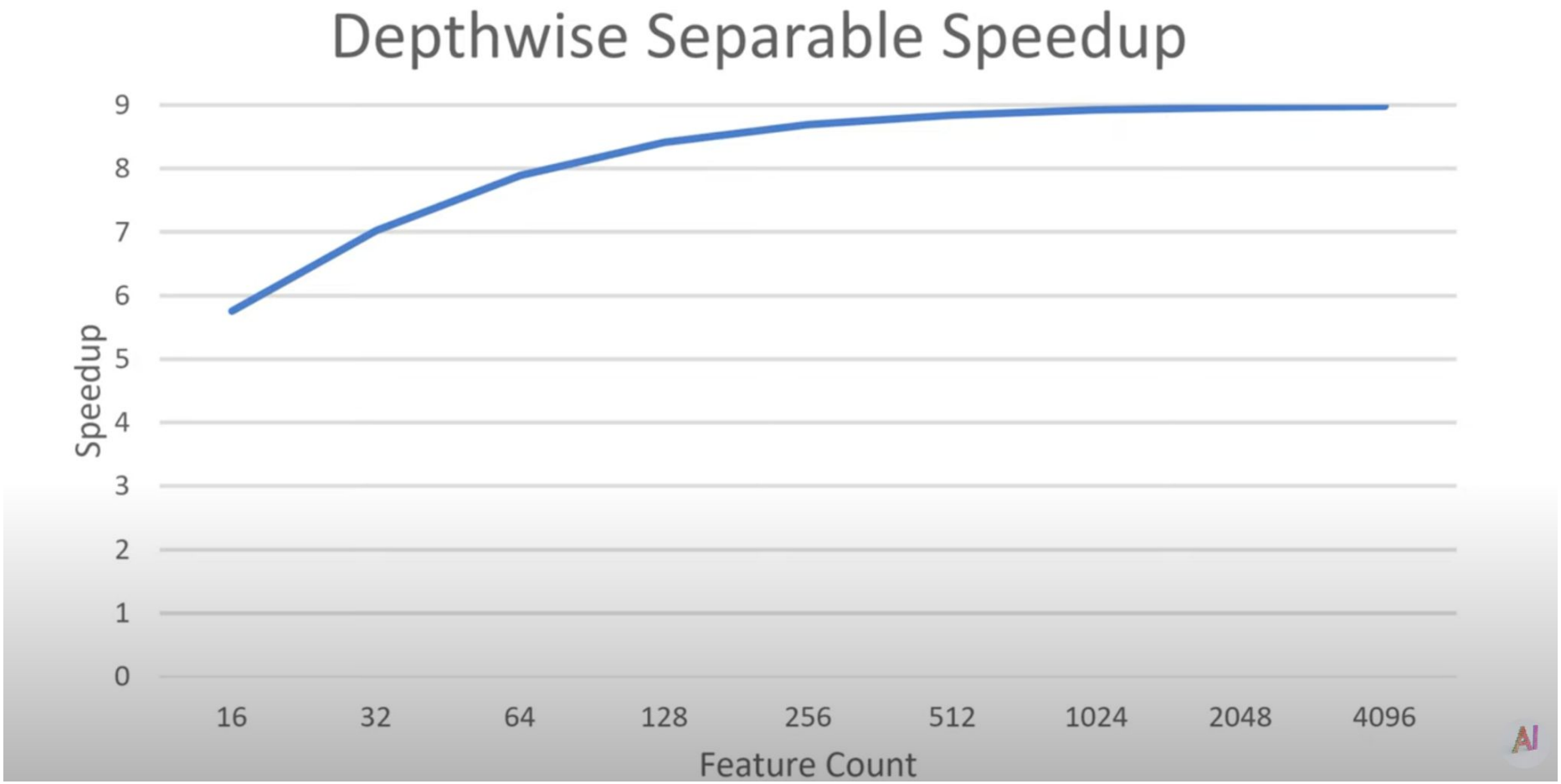


Problema: no hay flujo entre canales



Depthwise + pointwise (1x1 conv) = Depthwise separable




Depthwise Separable Speedup
Speedup
0
1
2
3
4
5
6
7
8
9
16
32
64
128
256
512
1024
2048
4096
Feature Count