PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA

MALTA

Machine Learning Theory and Applications Lab

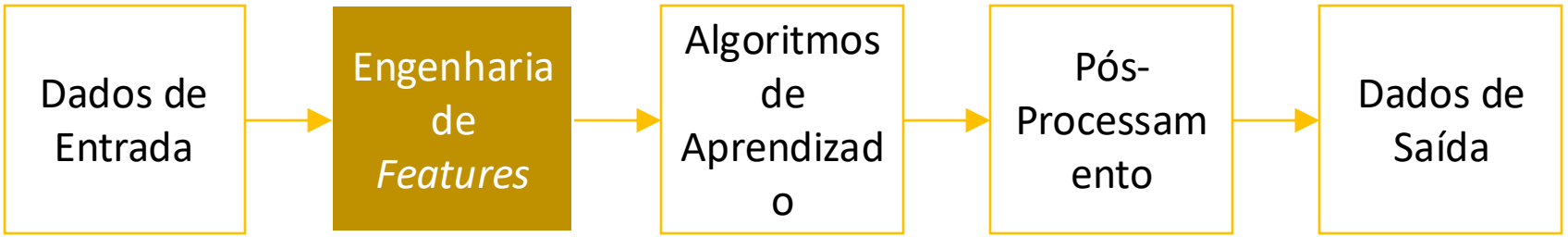
Aprendizado de Máquina

Paradigma baseado em Otimização
Redes Neurais IV

Prof. Me. Otávio Parraga

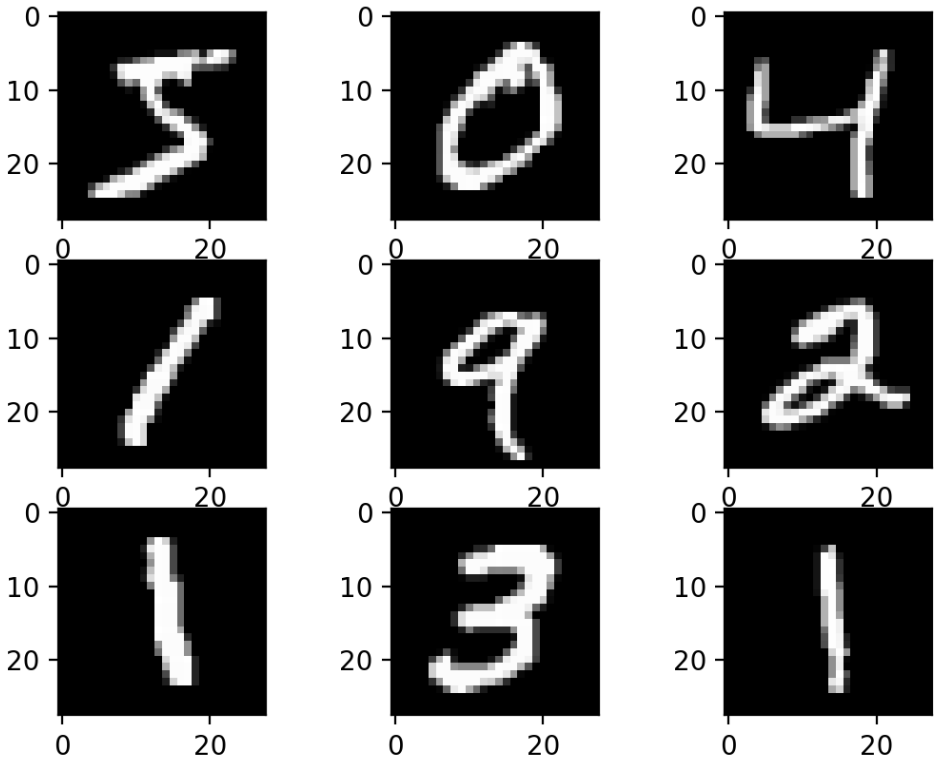
Engenharia de Features

- O processo de Aprendizado de Máquina
- Engenharia de Features como um dos processos **mais** custosos
  - E **menos** recompensadores!!!



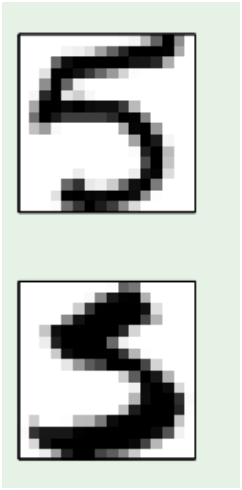
MNIST

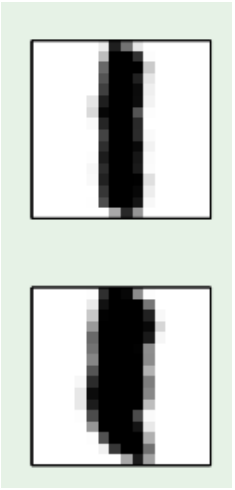
- Representar dados é uma tarefa difícil
- Como representar imagens de números de forma a diferenciar eles?



MNIST

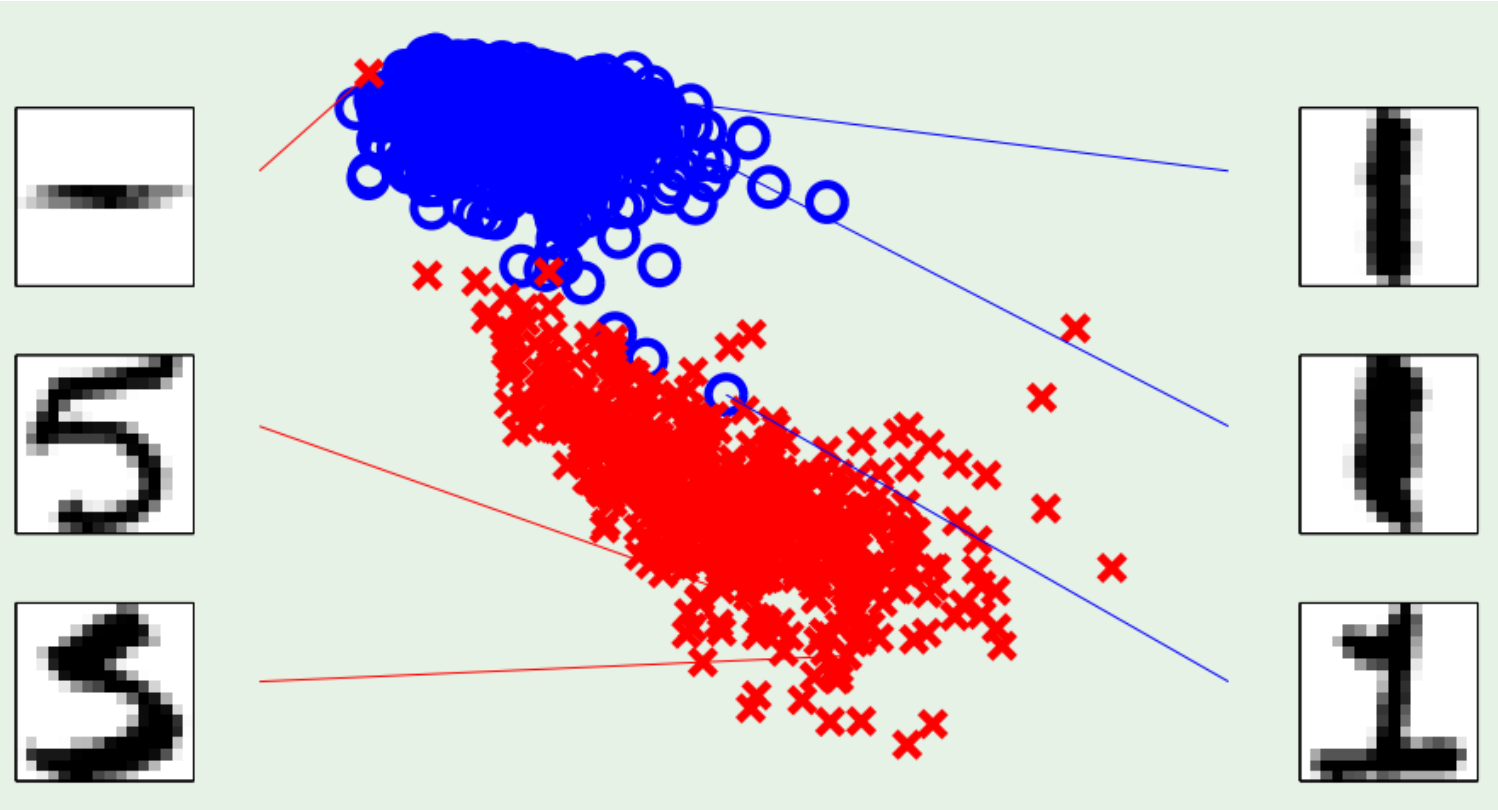
- Poderíamos diferenciar por duas características:
  - x1: Intensidade média dos pixels
  - x2: Simetria vertical





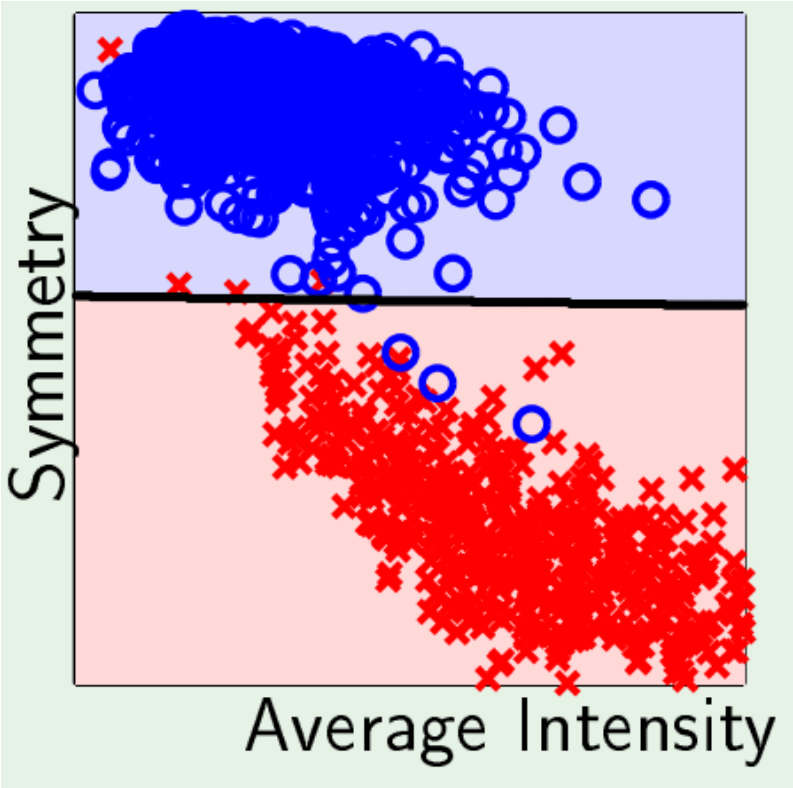
MNIST

- Poderíamos diferenciar por duas características:
  - x1: Intensidade média dos pixels
  - x2: Simetria vertical



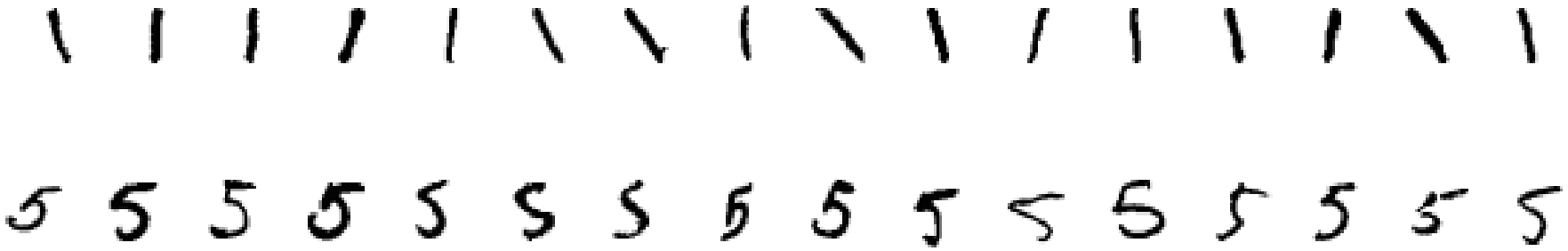
MNIST

- Conseguimos treinar um separador linear!



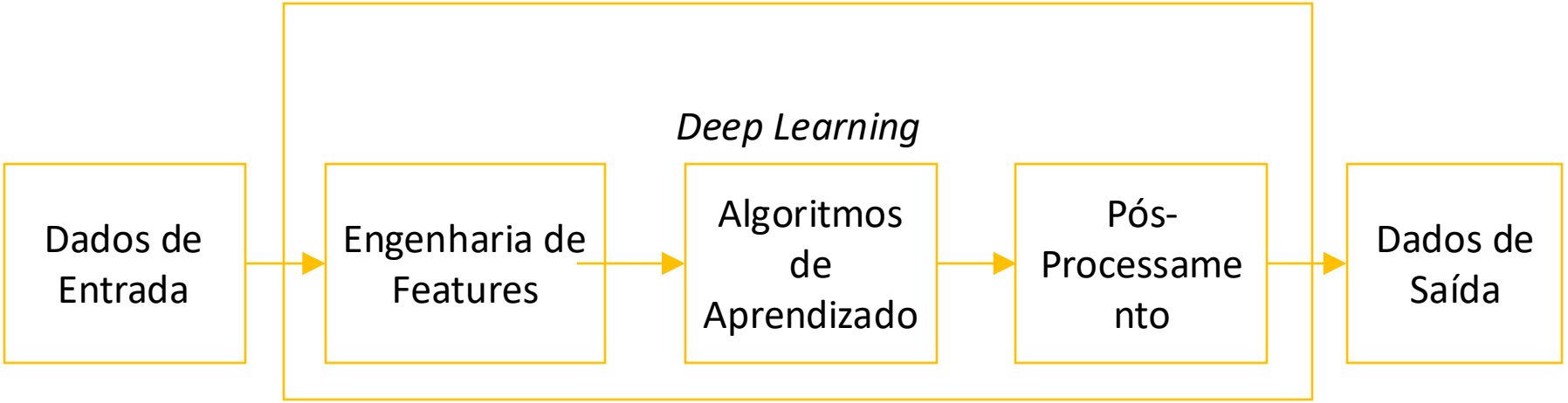
O Problema

- Como poderíamos lidar com a variabilidade?
- Engenharia de Features se torna uma tarefa **custosa e pouco recompensadora**
  - Especialmente com dados **não estruturados**

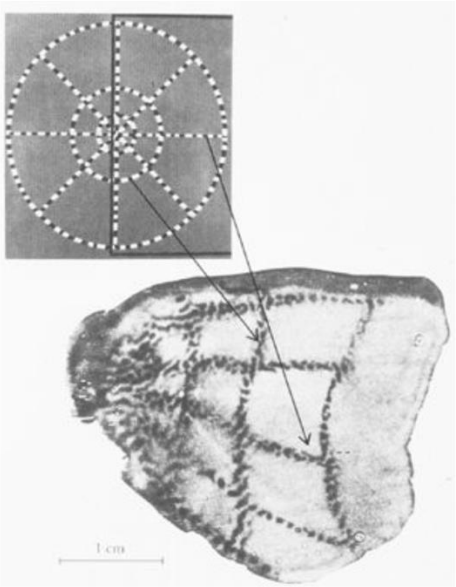


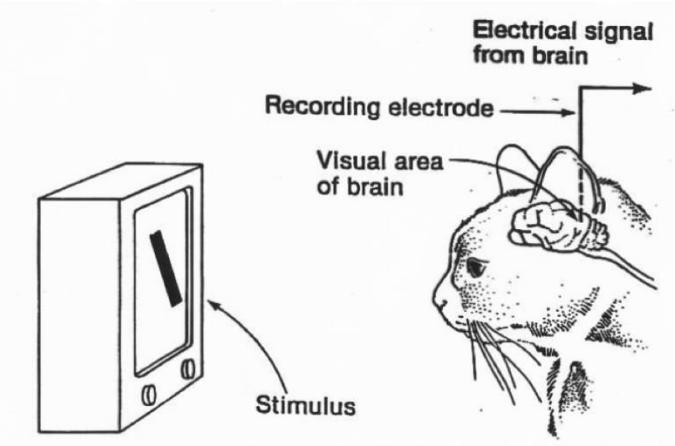
Deep Learning

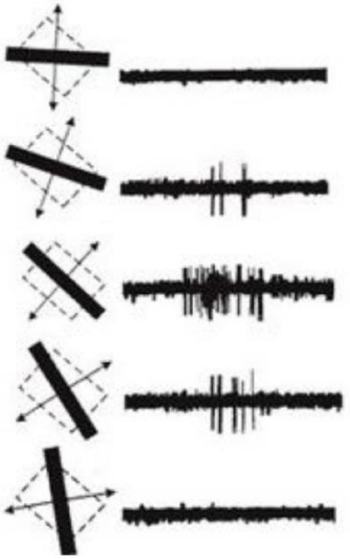
- E se delegássemos essa tarefa para o próprio algoritmo?

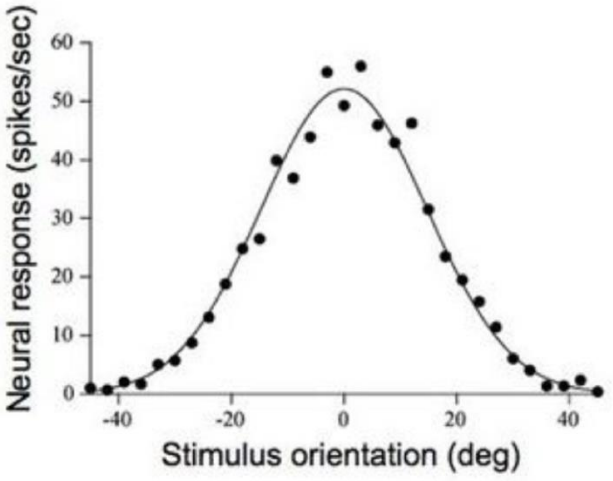


Um pouco de história...



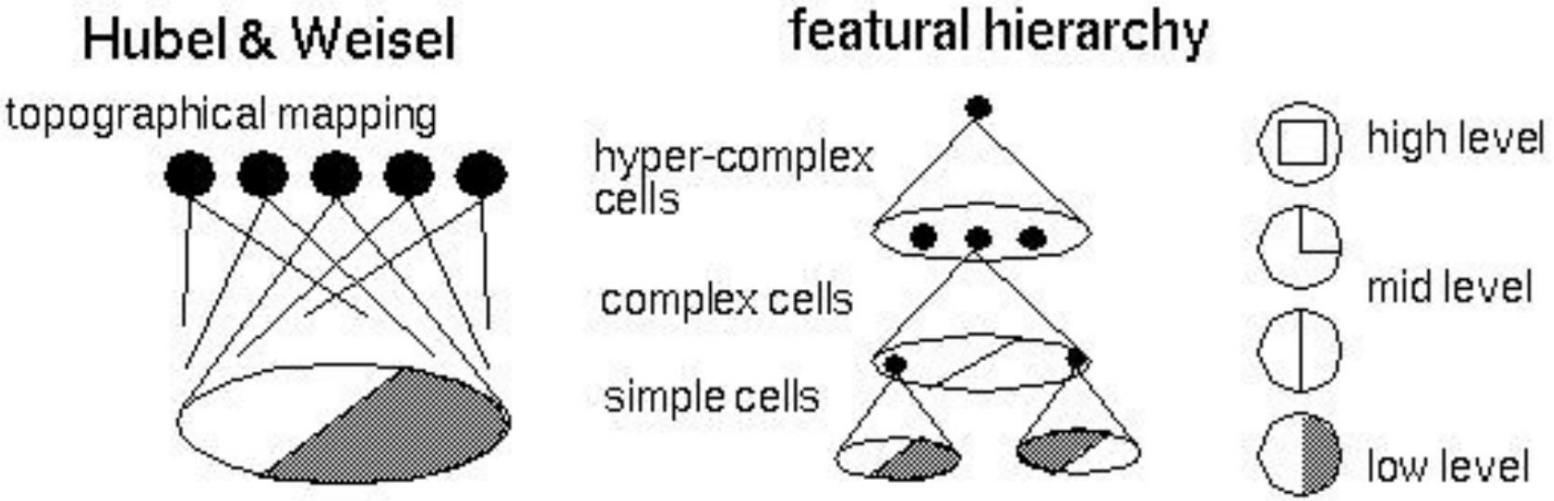






Hubel e Wiesel (1959)

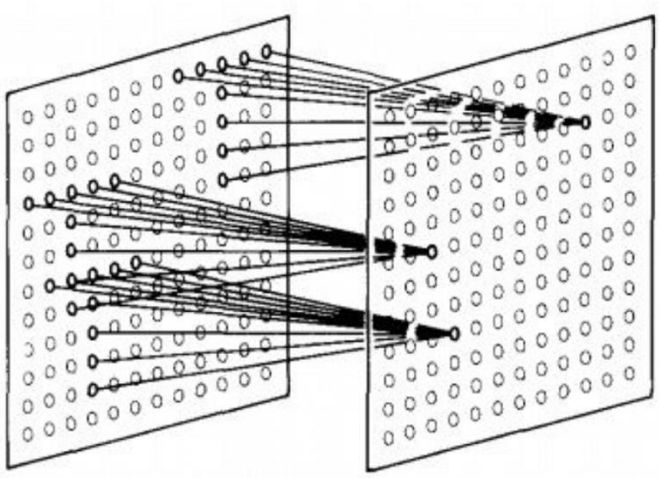
Um pouco de história…

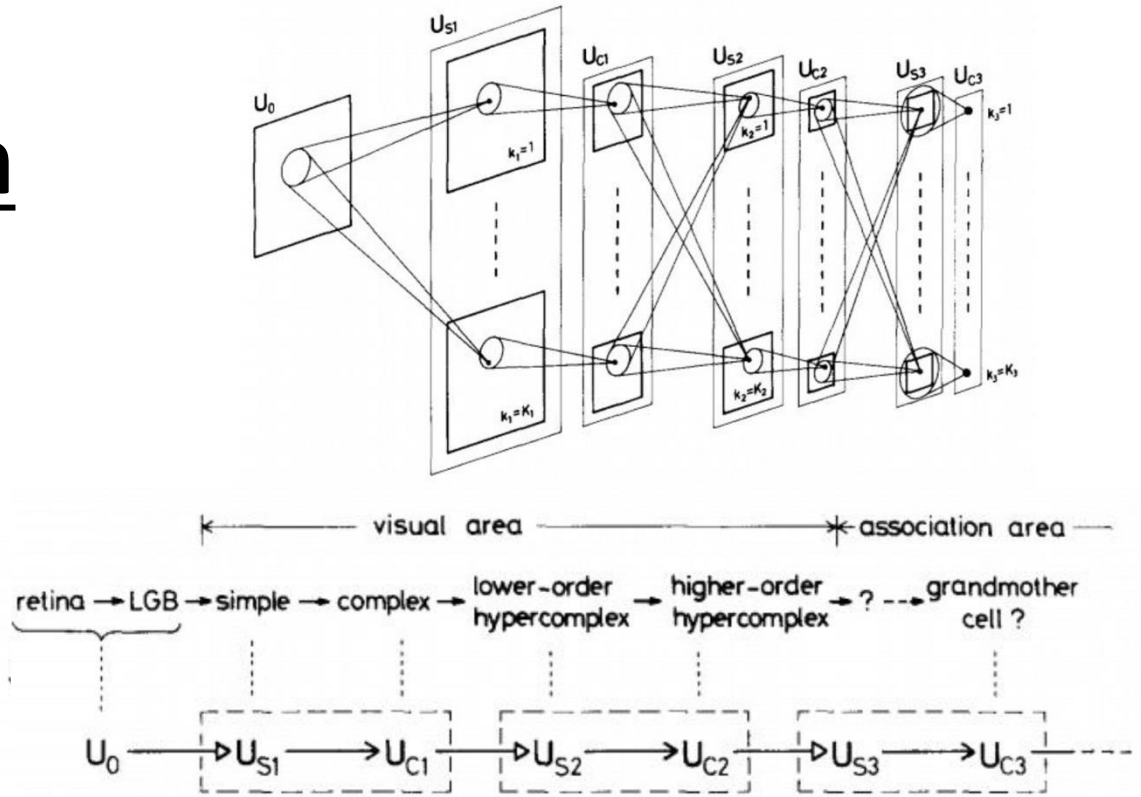


Um pouco de história...

(Fukushima, 1980)

Neurocognitron

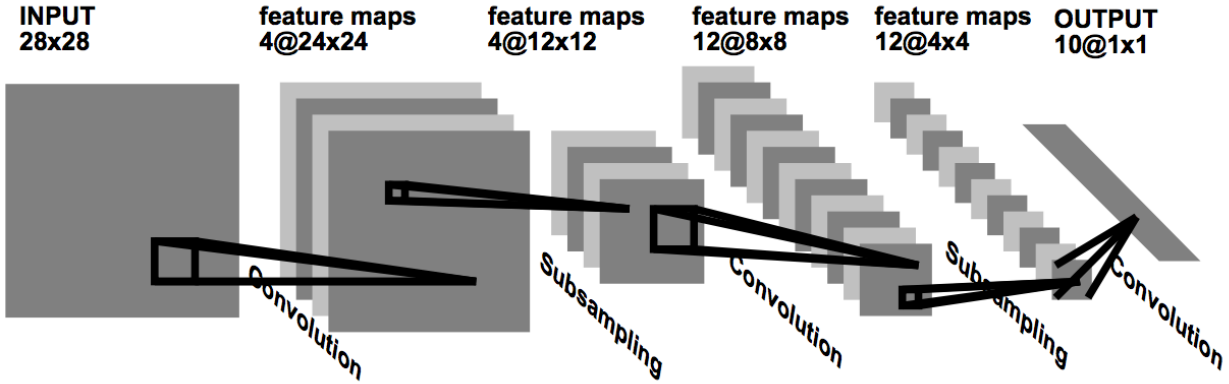


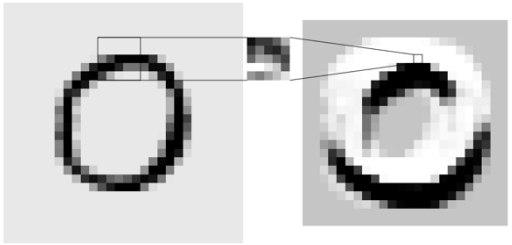


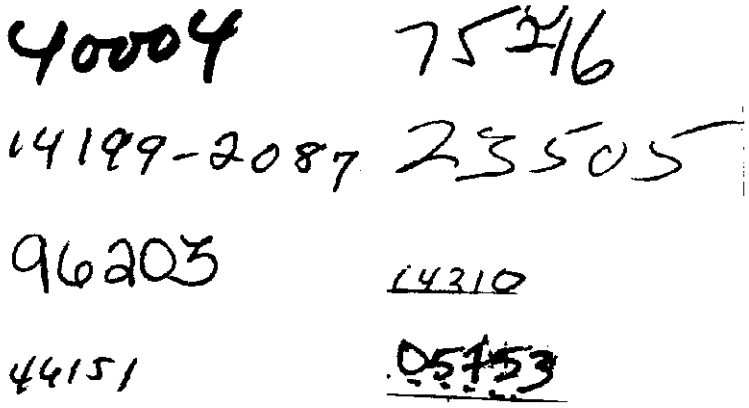
Um pouco de história...

1990 – A primeira convolucional! (LeNet)

98.3% de acurácia no MNIST







Y. LeCun, B. Boser, J. S. Denker, Richard E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. Handwritten digit recognition with a back-propagation network. In Advances in neural information processing systems, 1990.

Um pouco de história...

1995 – LeNet 5

99.1% de acurácia no MNIST

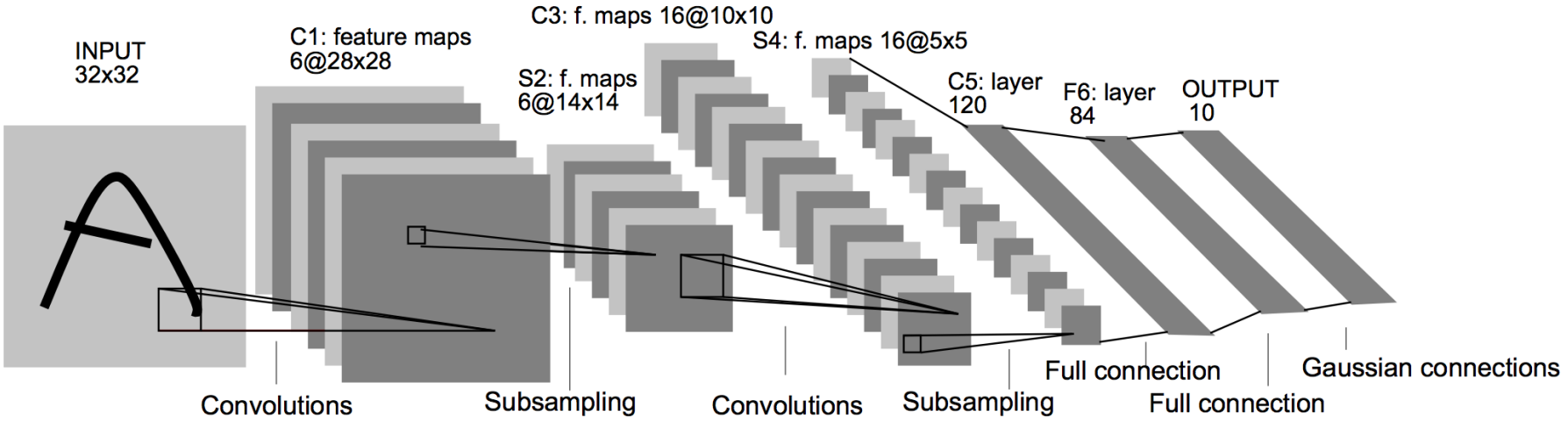


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In F. Fogelman and P. Gallinari, editors, *International Conference on Artificial Neural Networks*, pages 53-60, Paris, 1995.

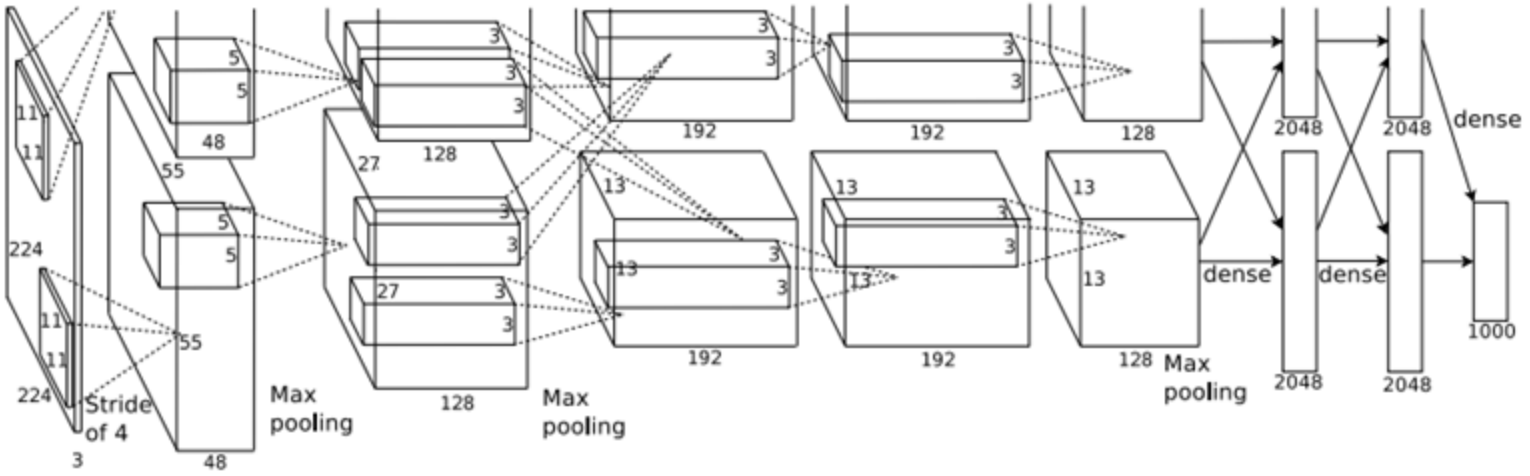
Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, november 1998.

Um pouco de história…

IMAGENET

84.7% de acurácia no ImageNet

2012 – AlexNet



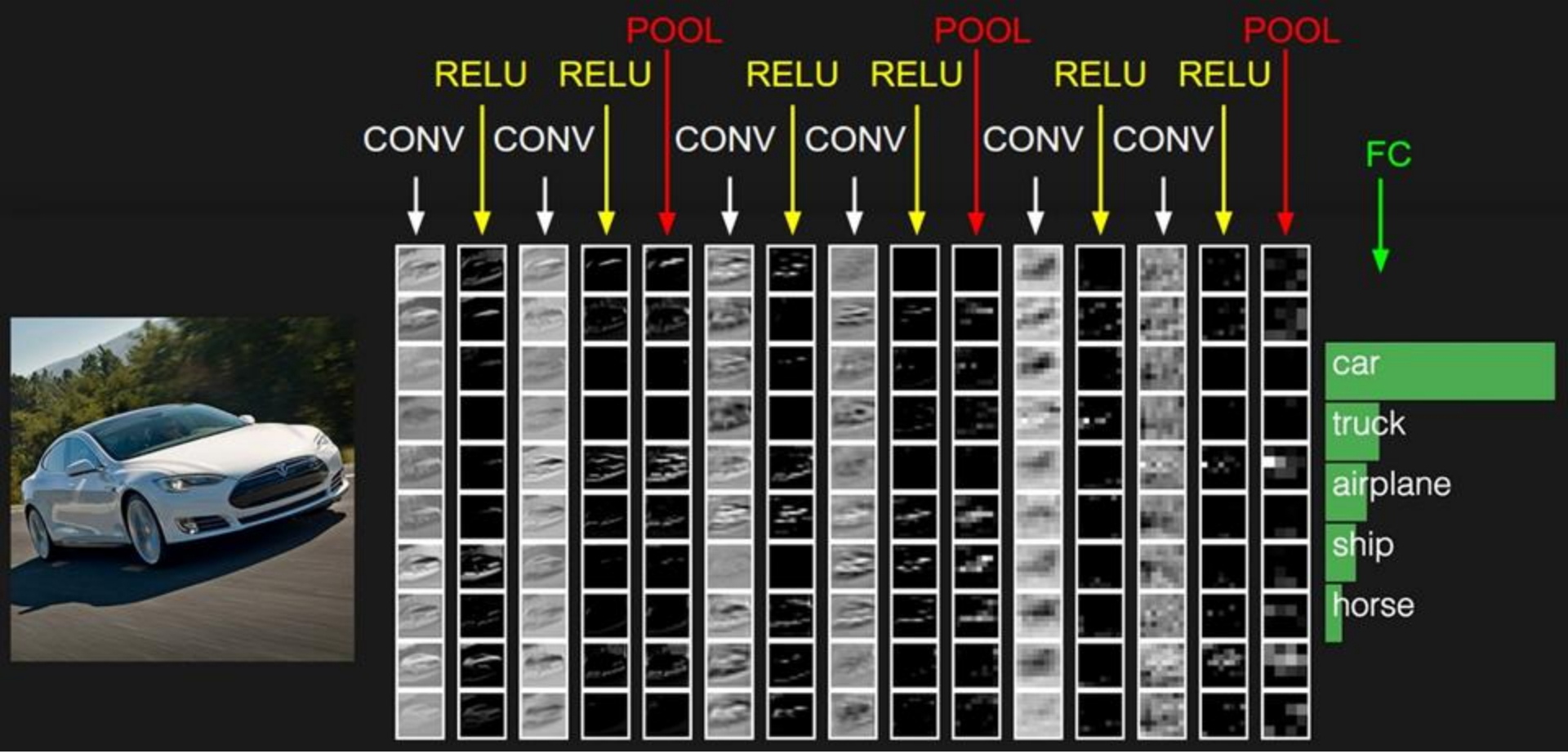
Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems , pages 1097–1105, 2012.

O que **É** uma Rede Neural Convolucional:

- ”Rede Neural *feed-forward* cujos neurônios estão dispostos de tal forma a responder por regiões sobrepostas (**campos receptores**) que preenchem o campo visual”

- ”Múltiplas camadas de **pequenas coleções de neurônios** que processam porções da imagem de entrada (campos receptores)”

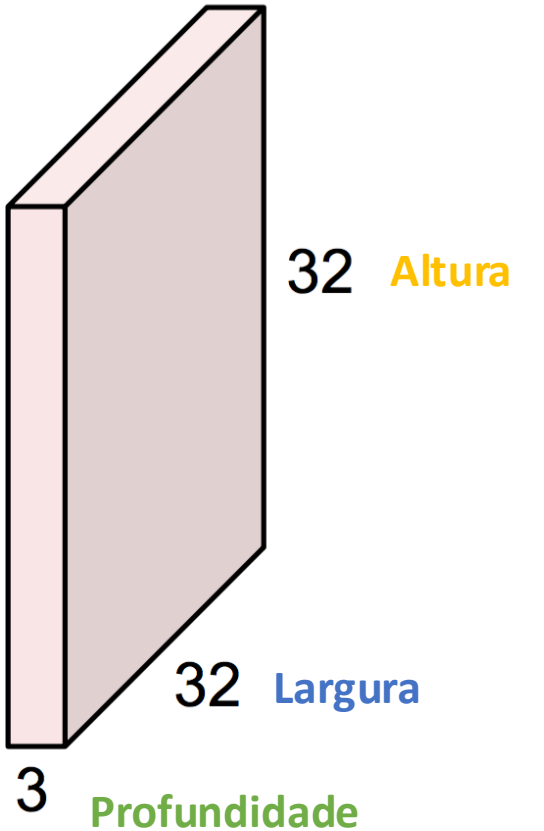
- ”Sequências de **camadas de convolução** intercaladas por funções de ativação”



Começaremos com a camada de convolução…

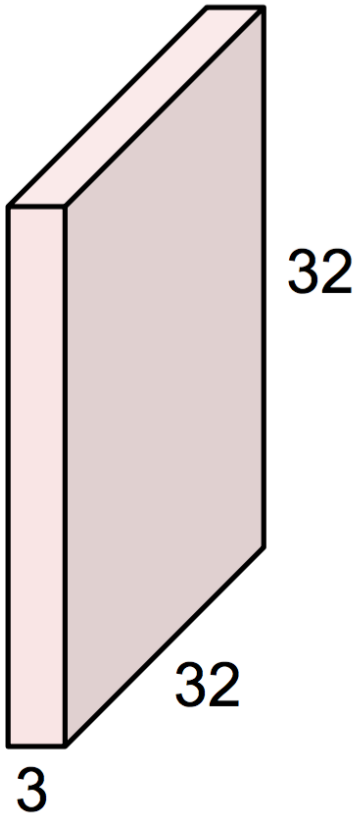
Camada de Convolução

Imagem $32 \times 32 \times 3$

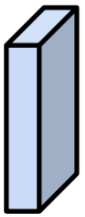


Camada de Convolução

Imagem $32 \times 32 \times 3$



Filtro $5 \times 5 \times 3$



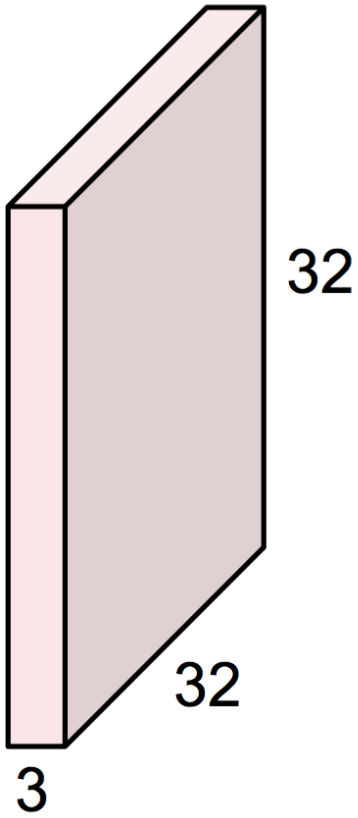
Convoluir o filtro com a imagem

Deslizar espacialmente o filtro pela imagem, computando produtos internos

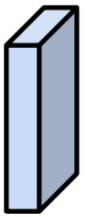
Camada de Convolução

Imagem $32 \times 32 \times 3$

Filtros sempre compreendem a profundidade inteira do volume de entrada



Filtro $5 \times 5 \times 3$



Convoluir o filtro com a imagem

Deslizar espacialmente o filtro pela imagem, computando produtos internos

Camada de Convolução

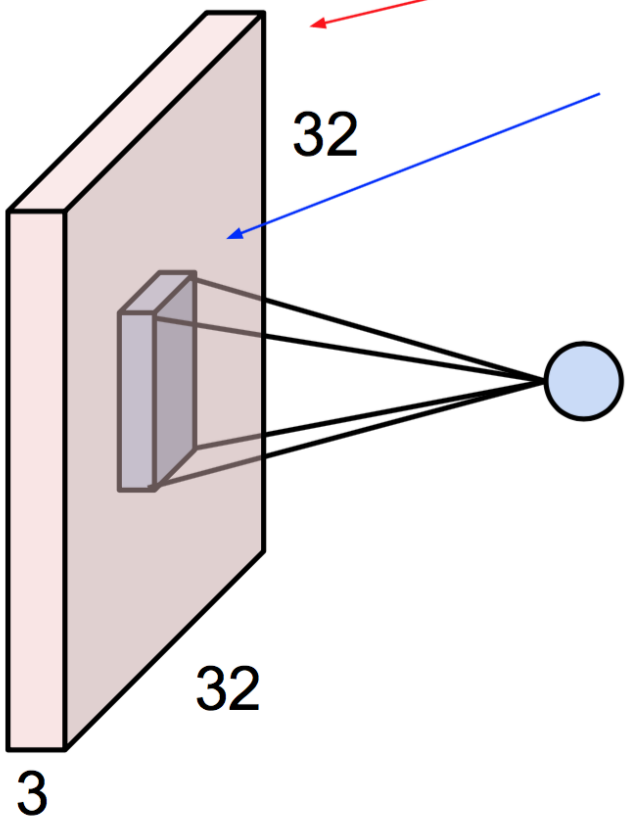
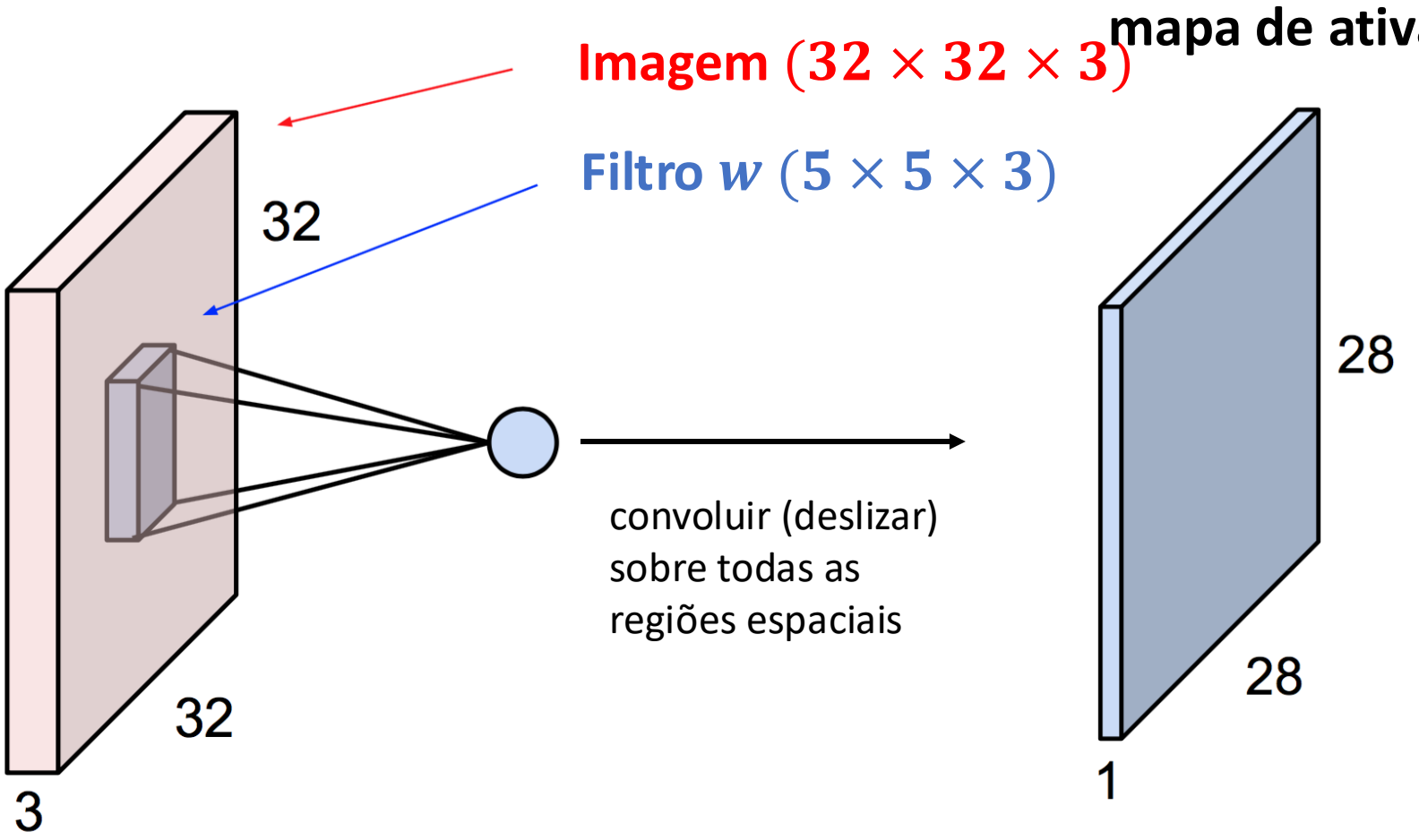


Imagem $(32 \times 32 \times 3)$

Filtro w $(5 \times 5 \times 3)$

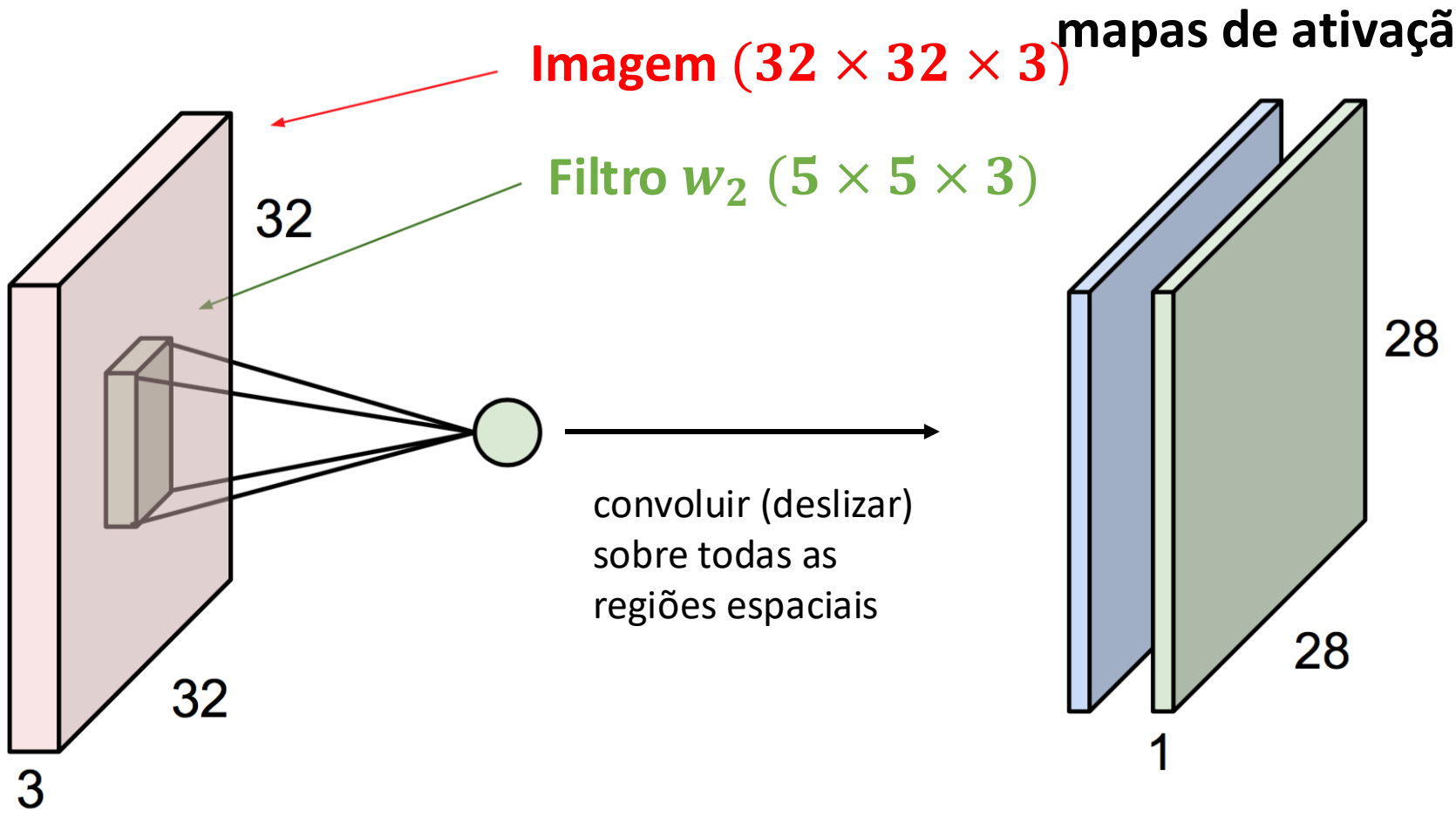
1 número:
resultado do produto interno entre o filtro e uma porção $5 \times 5 \times 3$ da imagem
$(5 \times 5 \times 3 = 75$ parâmetros + bias)

Camada de Convolução



Camada de Convolução

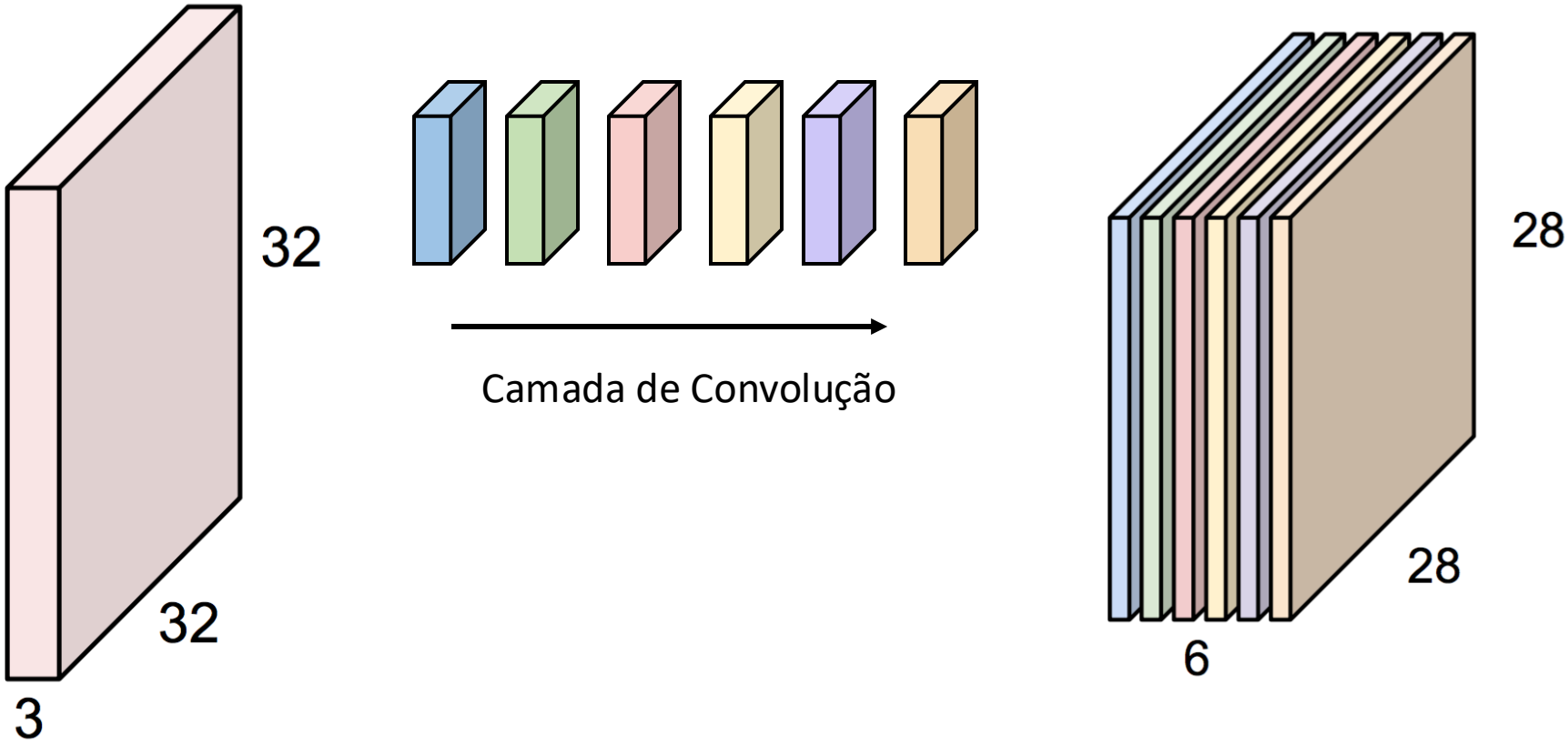
Considere a existência de um segundo filtro (verde)



Camada de Convolução

Se tivermos 6 filtros $5 \times 5 \times 3$, teremos 6 feature maps distintos:

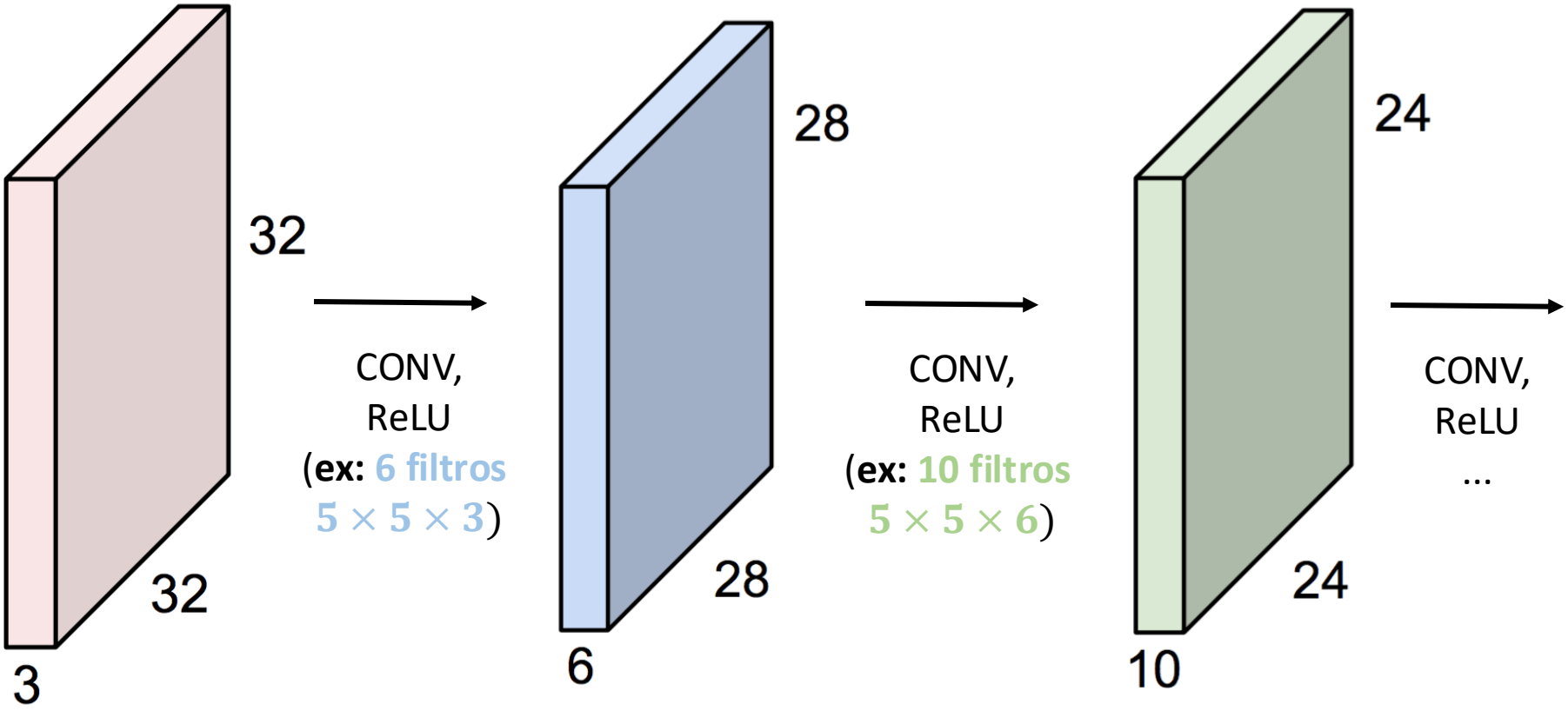
mapas de ativação



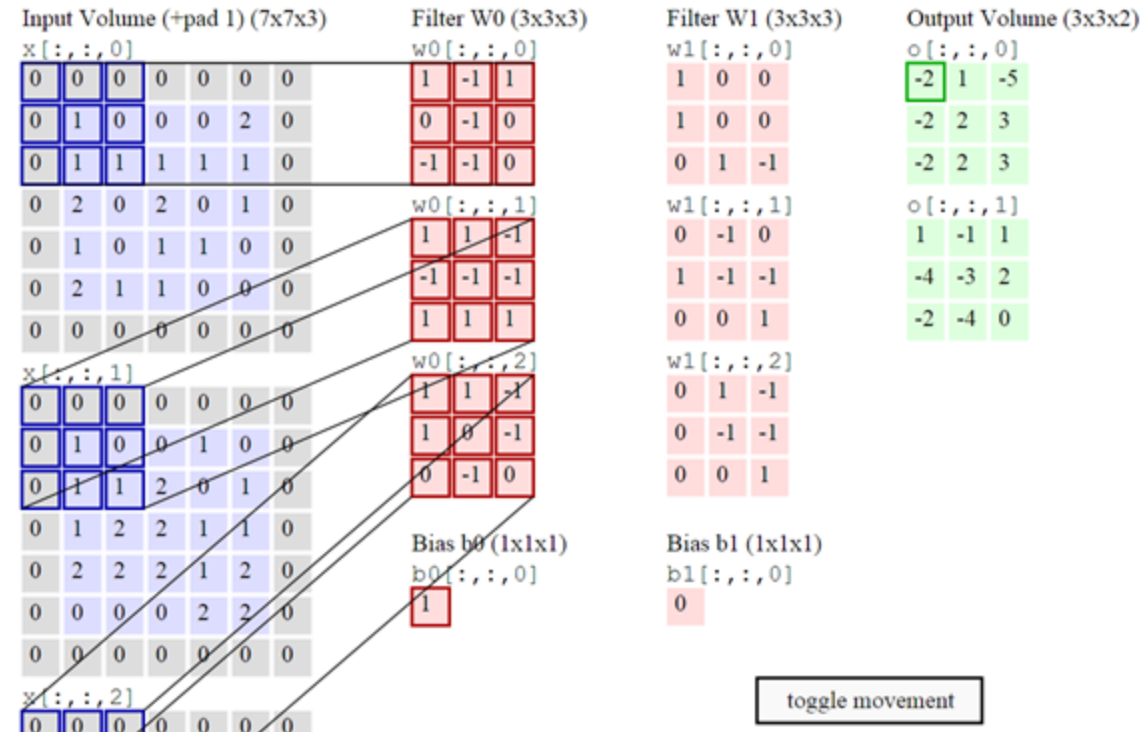
Empilhamos o *feature maps* para gerar uma ”nova imagem” de tamanho $28 \times 28 \times 6$

Rede Convolucional

”Sequência de camadas de convolução intercaladas por funções de ativação”

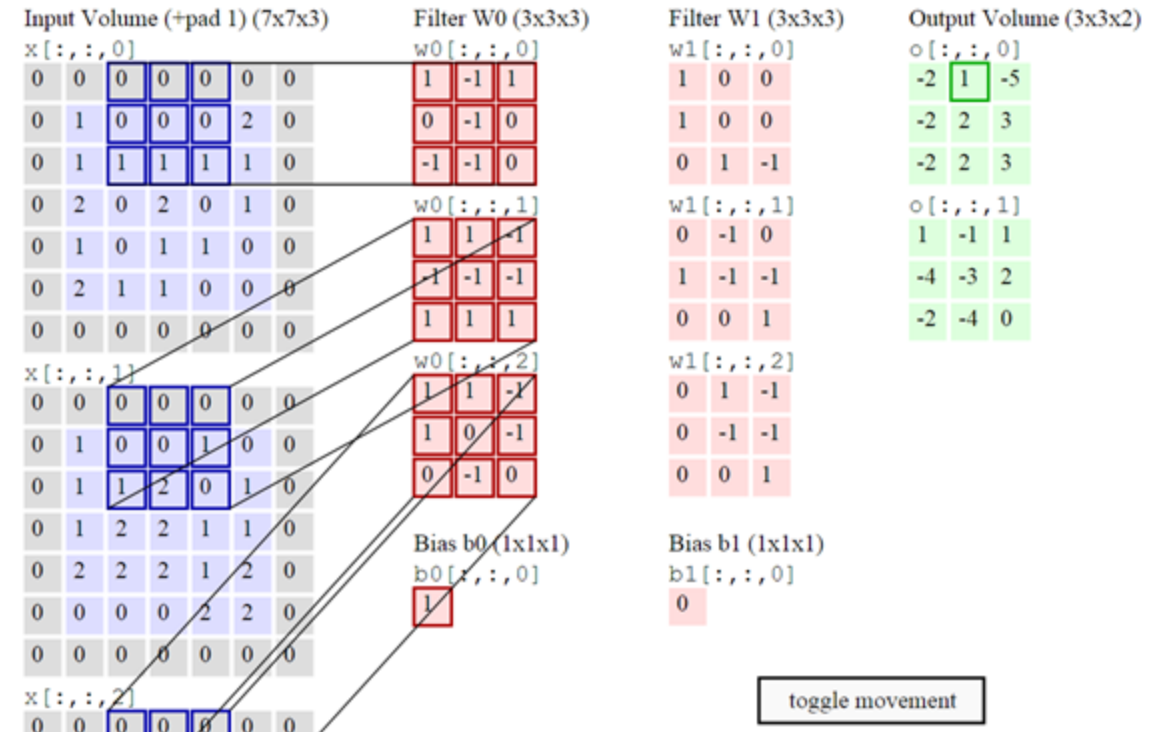


Redes Neurais Convolucionais



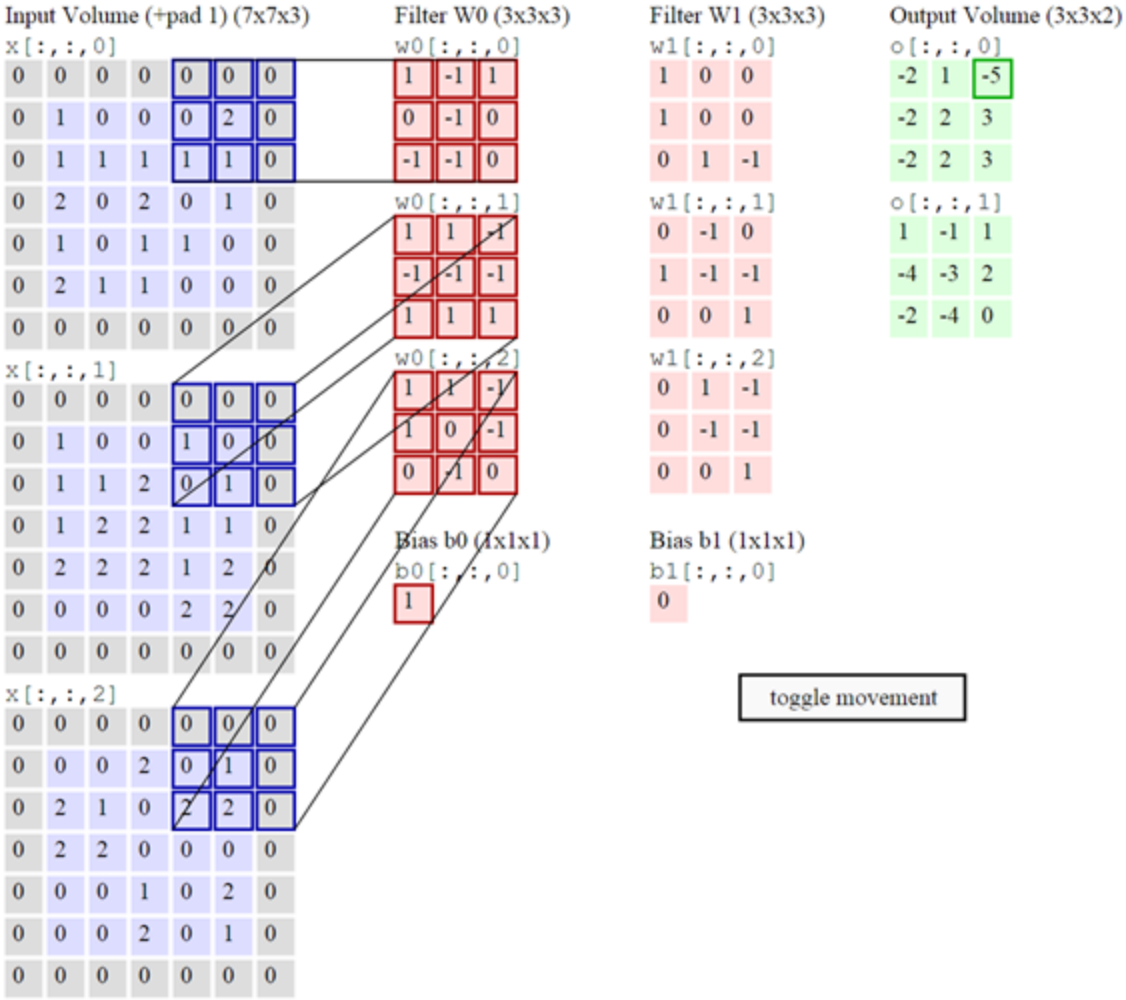
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



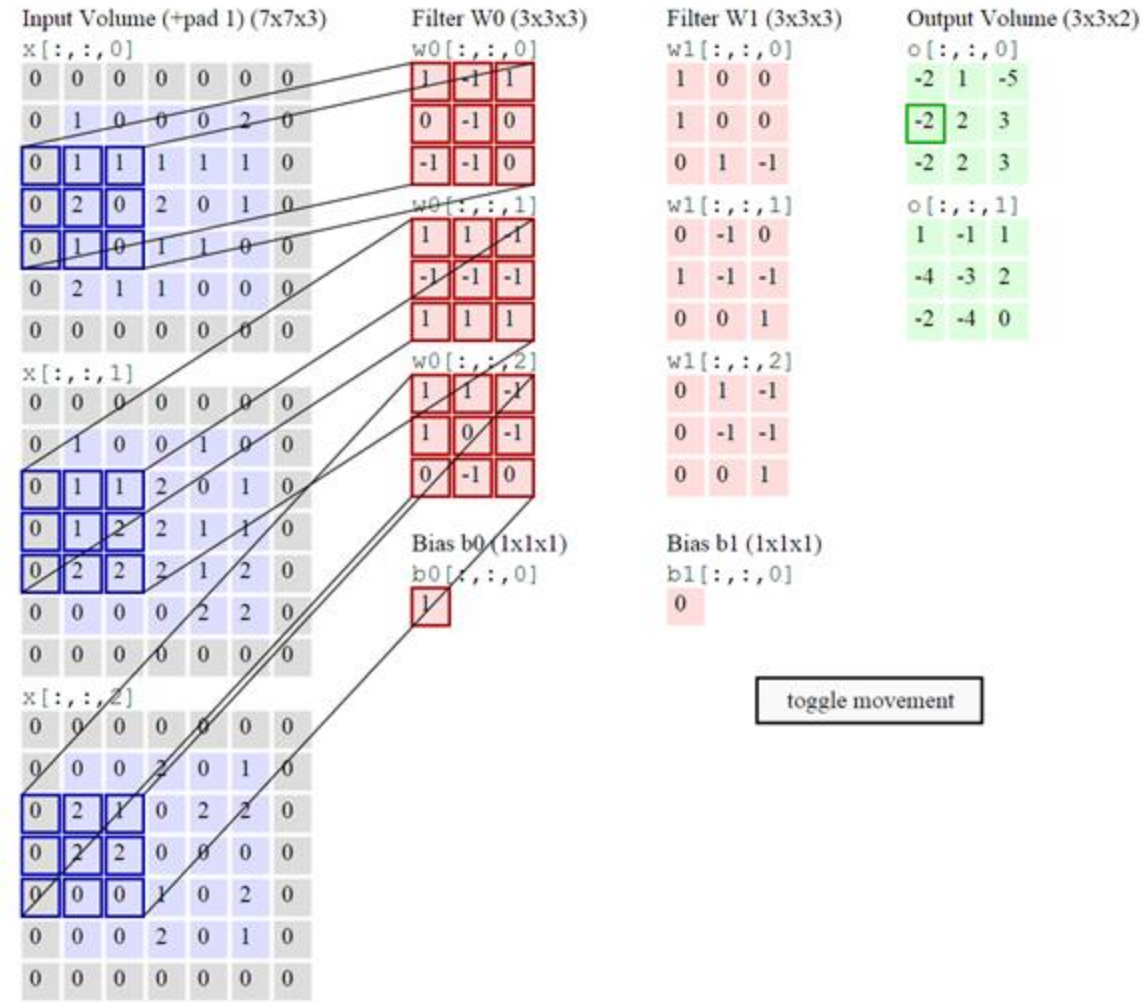
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



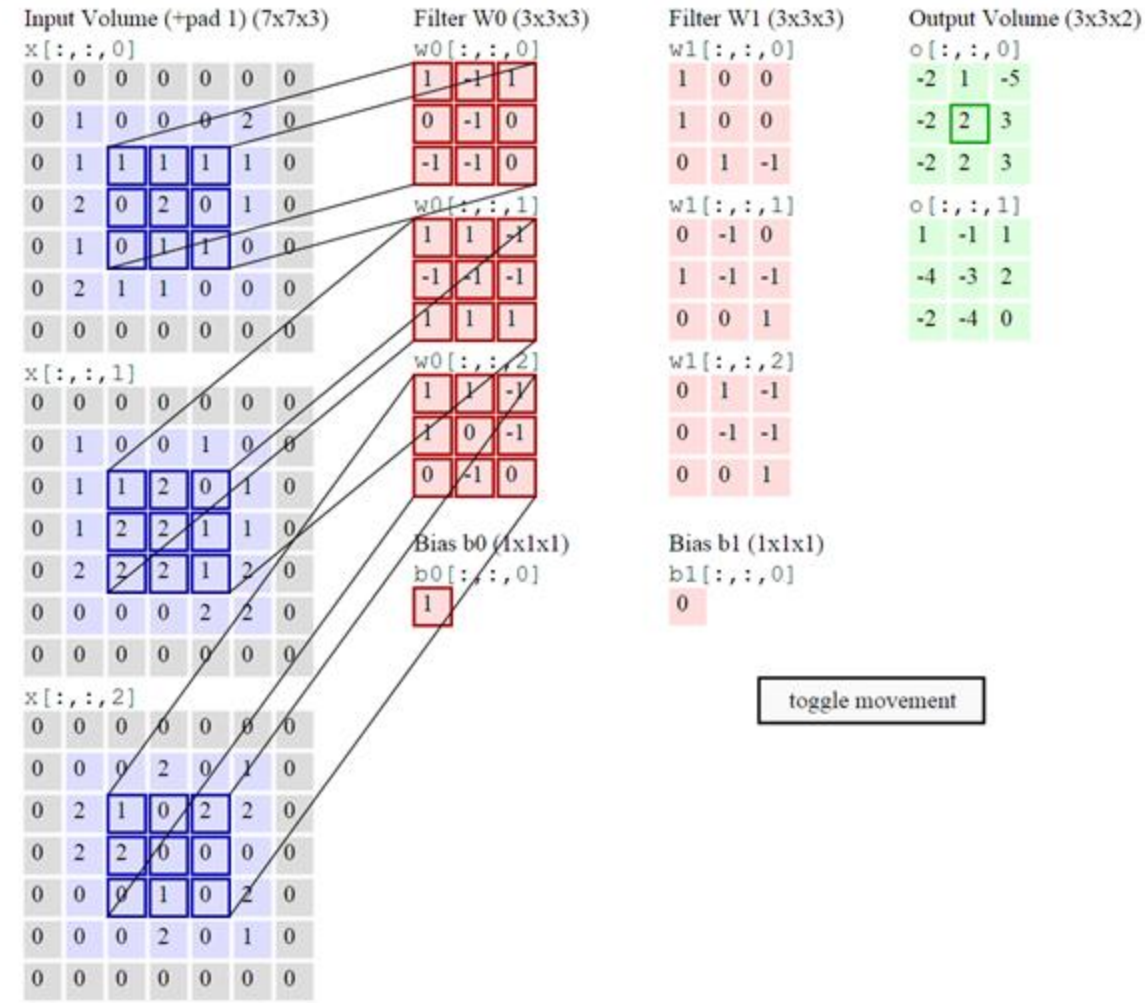
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



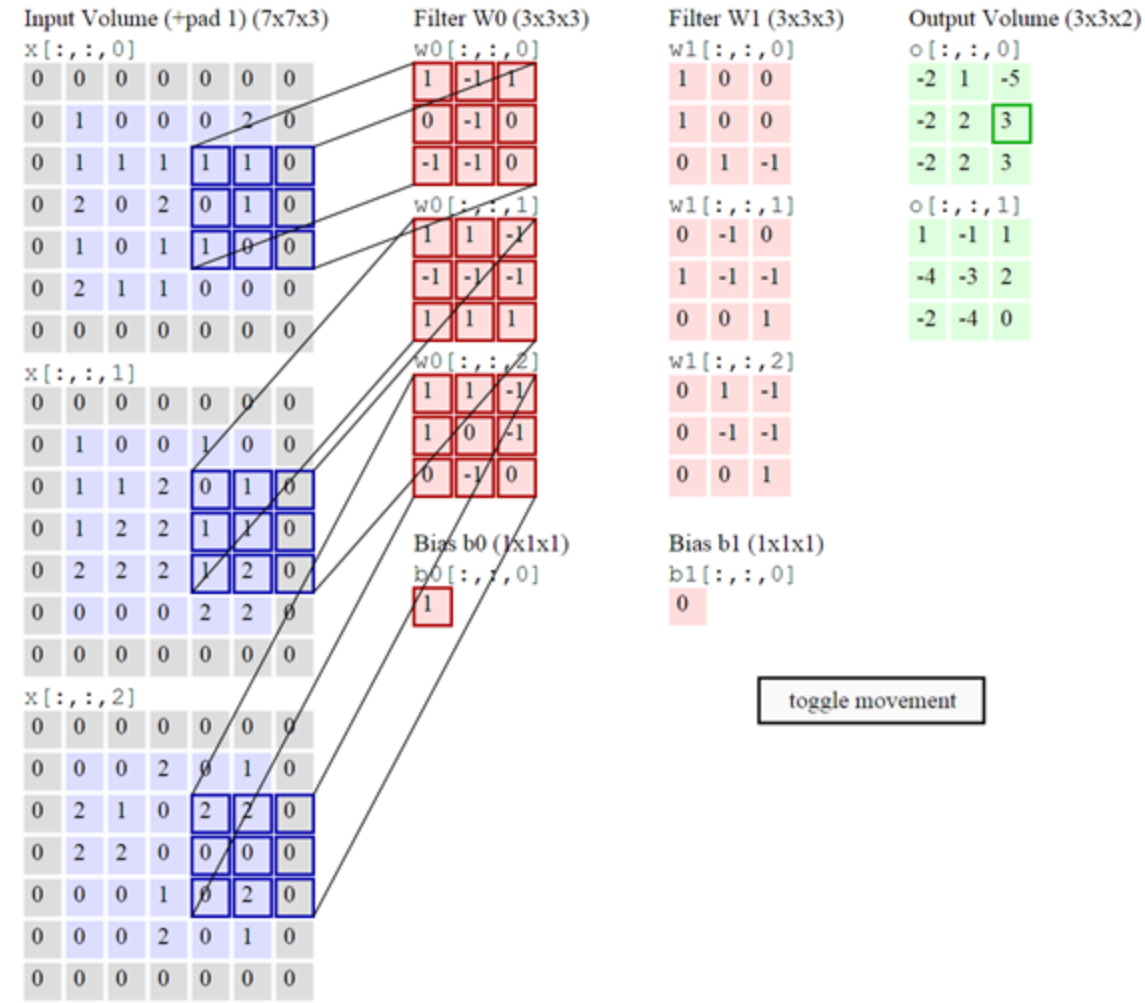
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



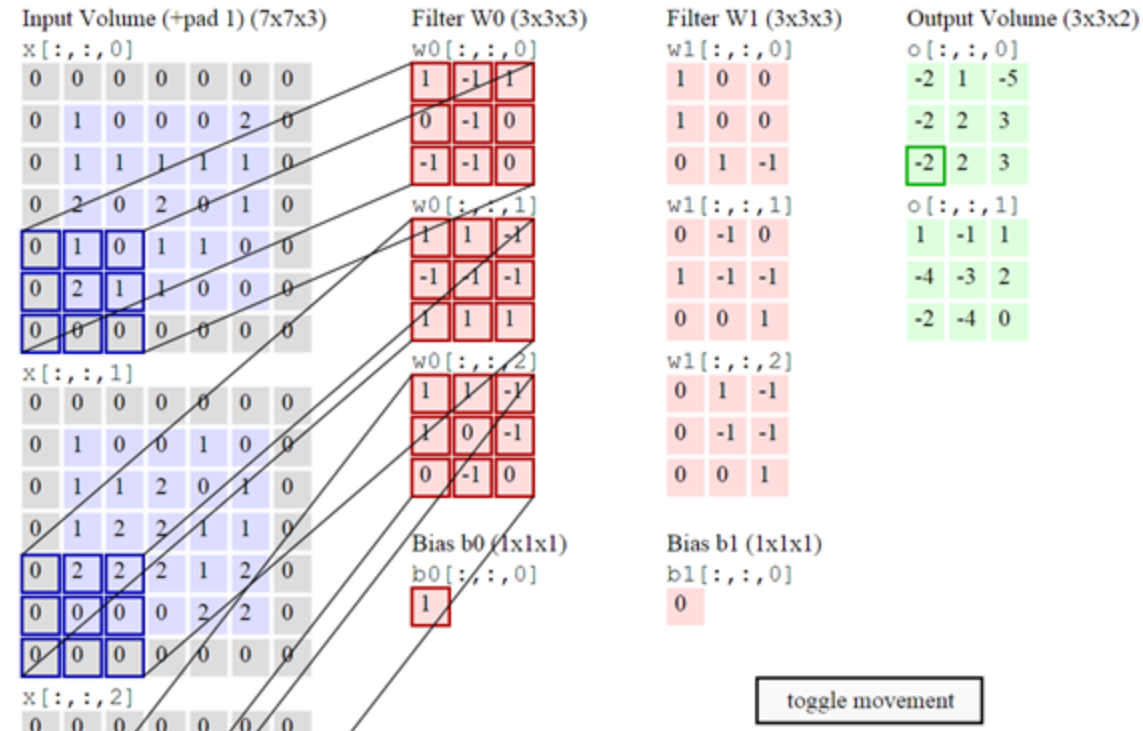
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



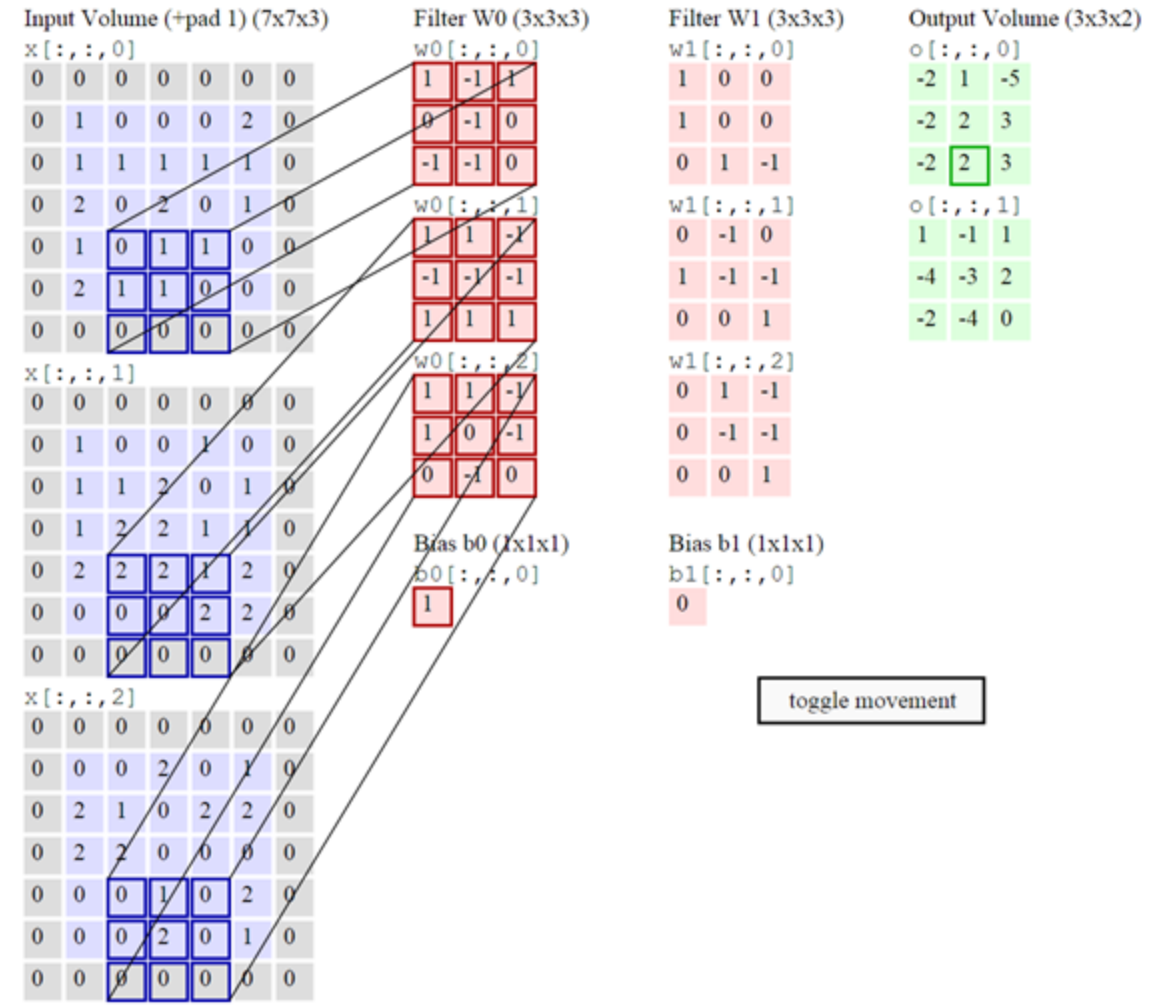
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



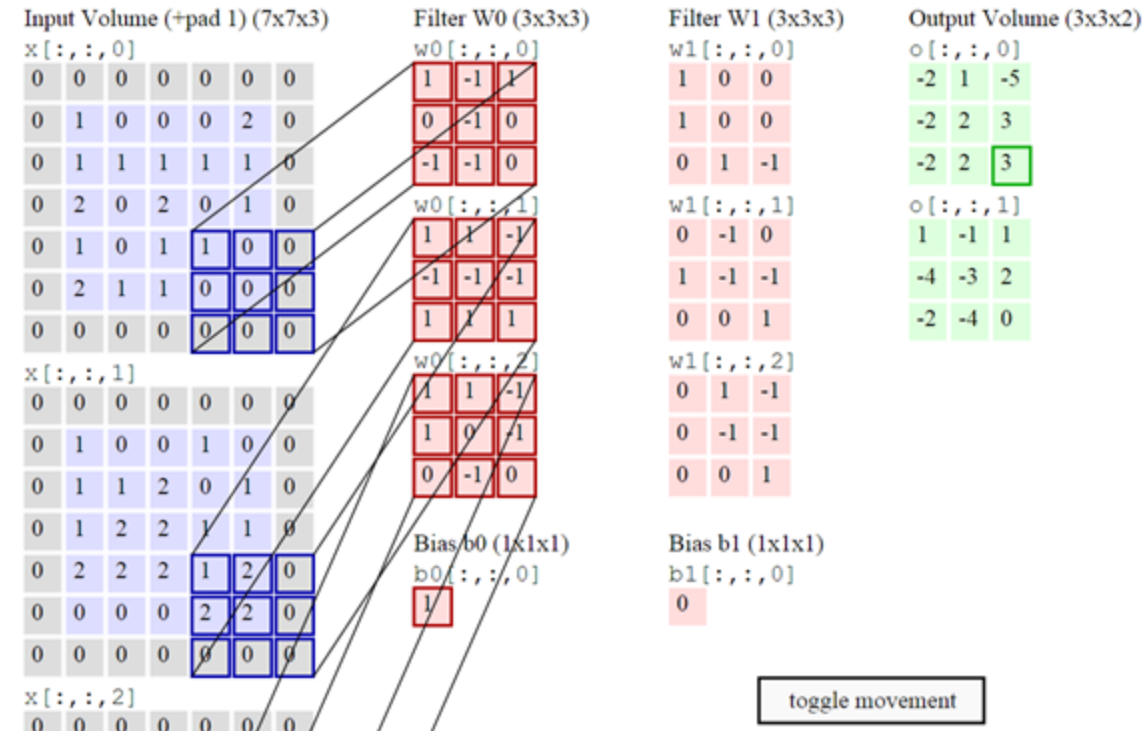
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



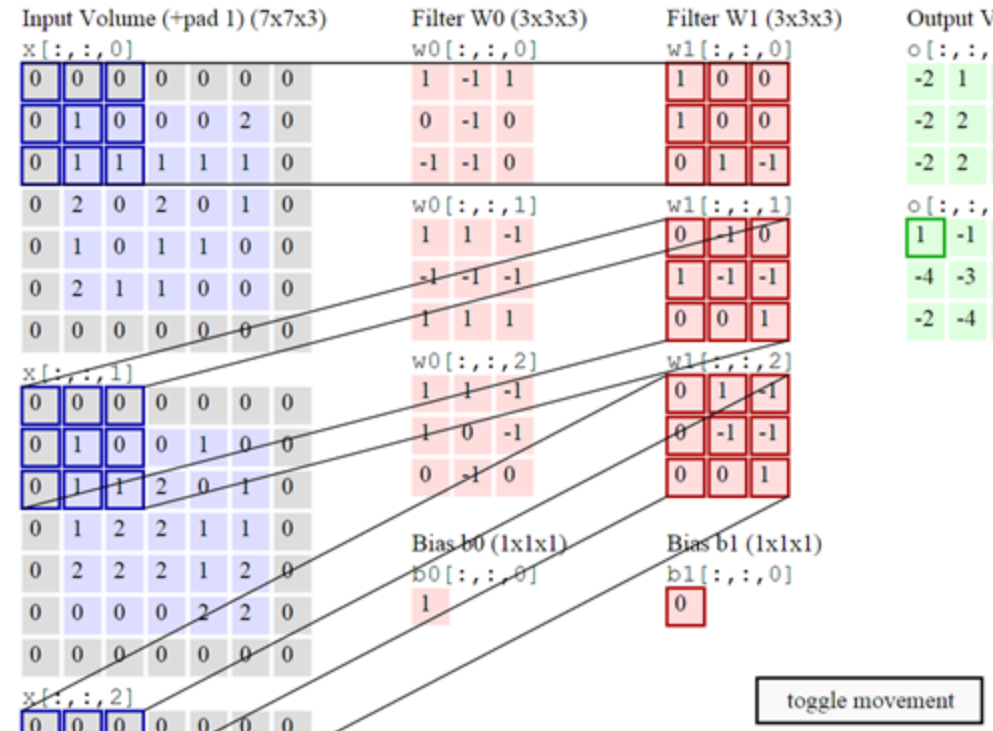
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



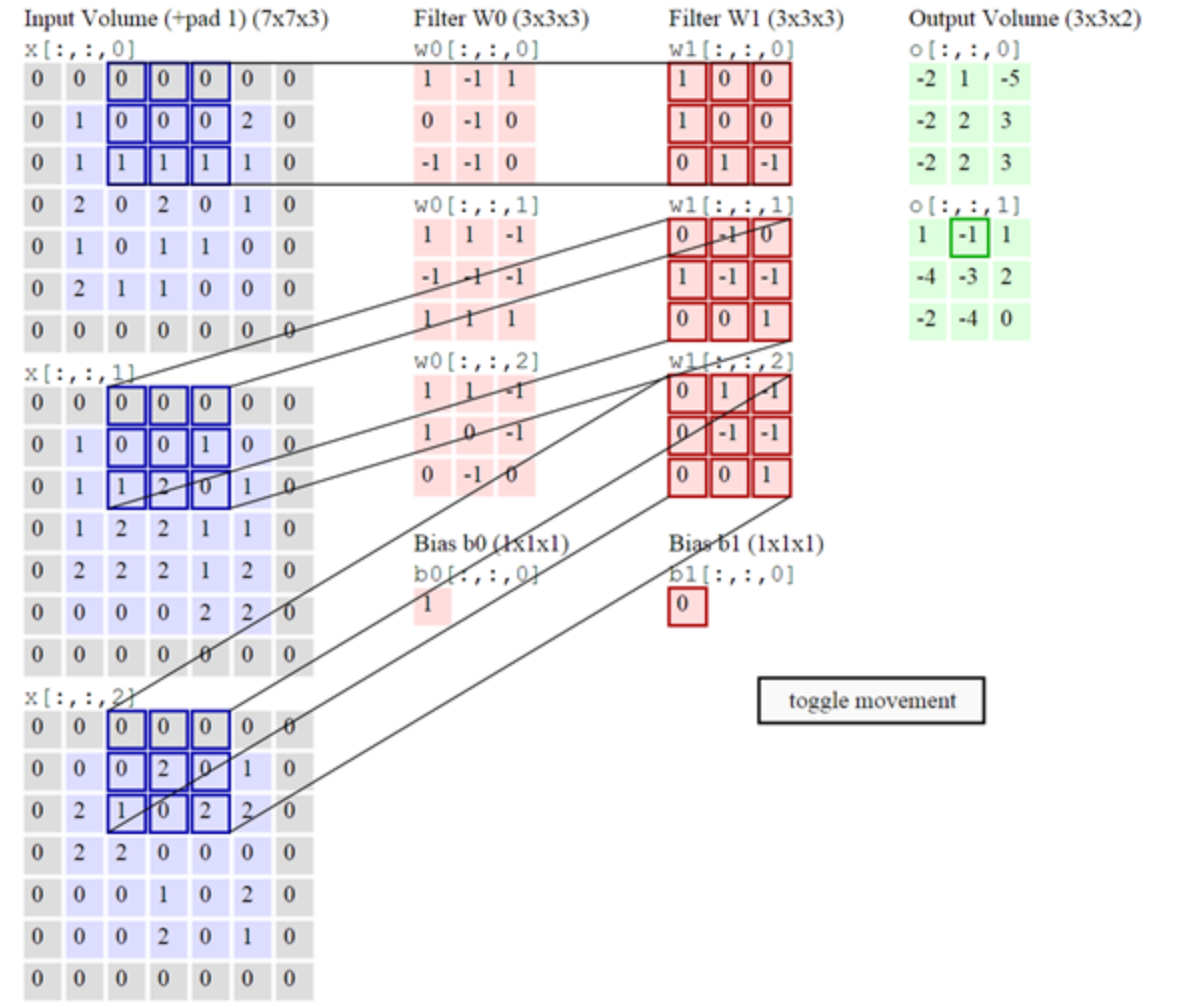
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



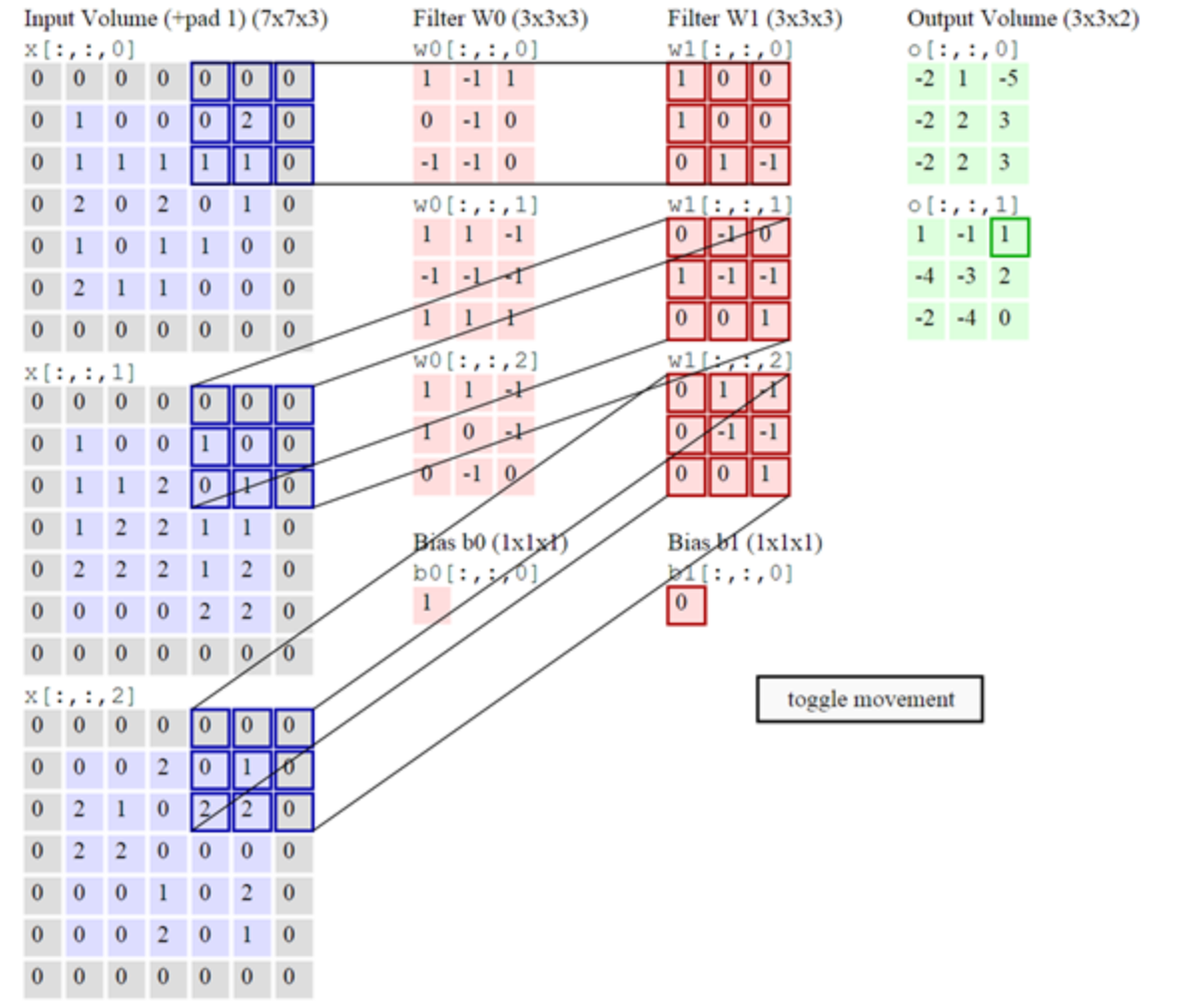
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



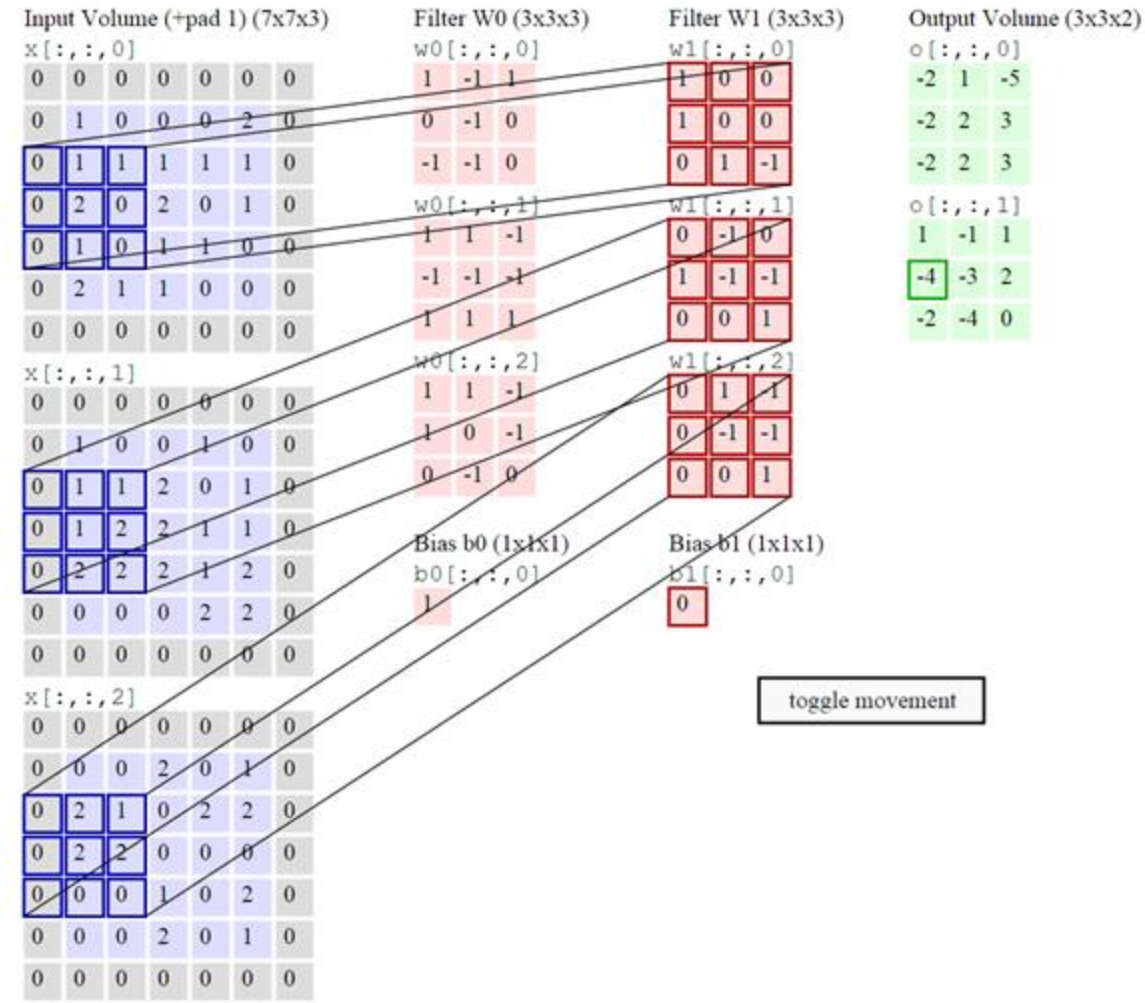
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



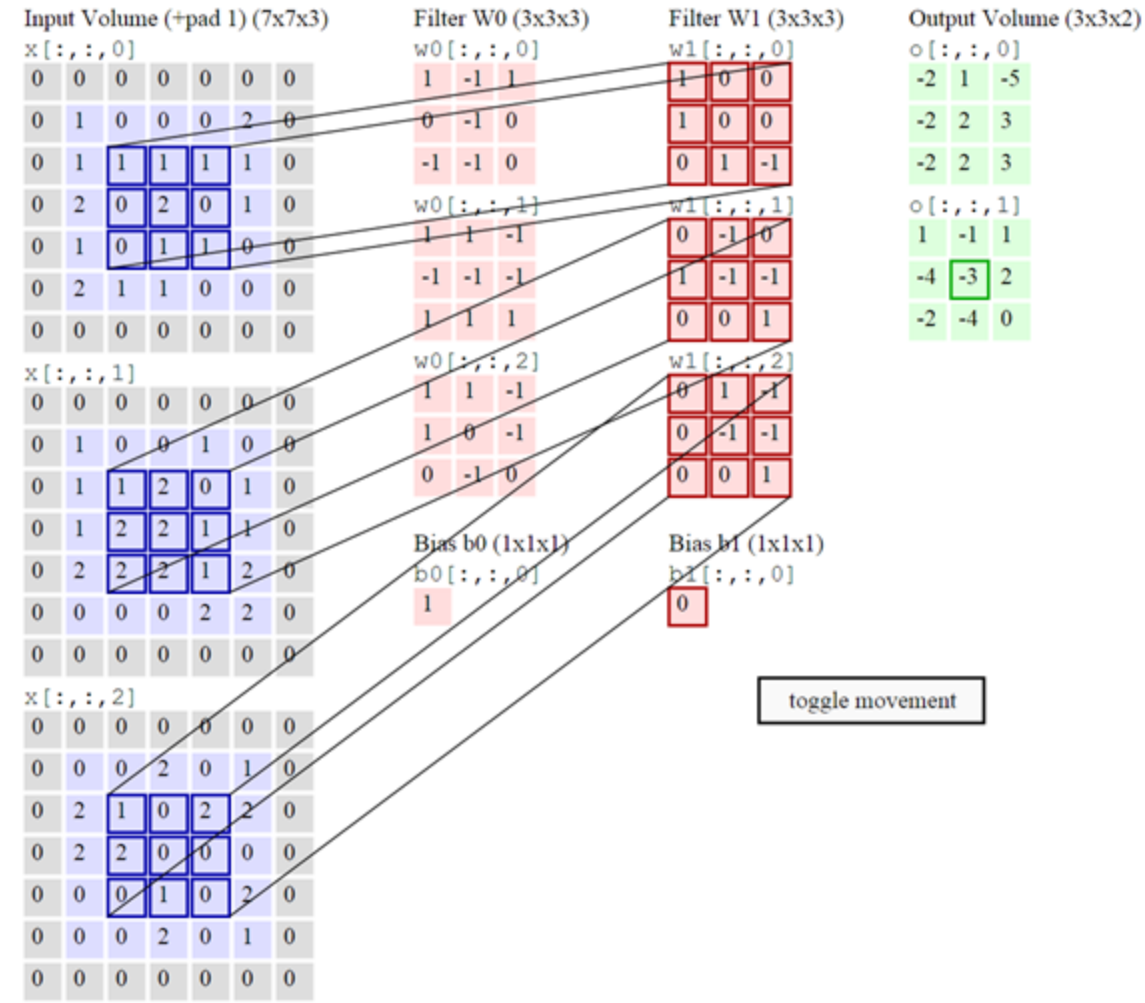
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



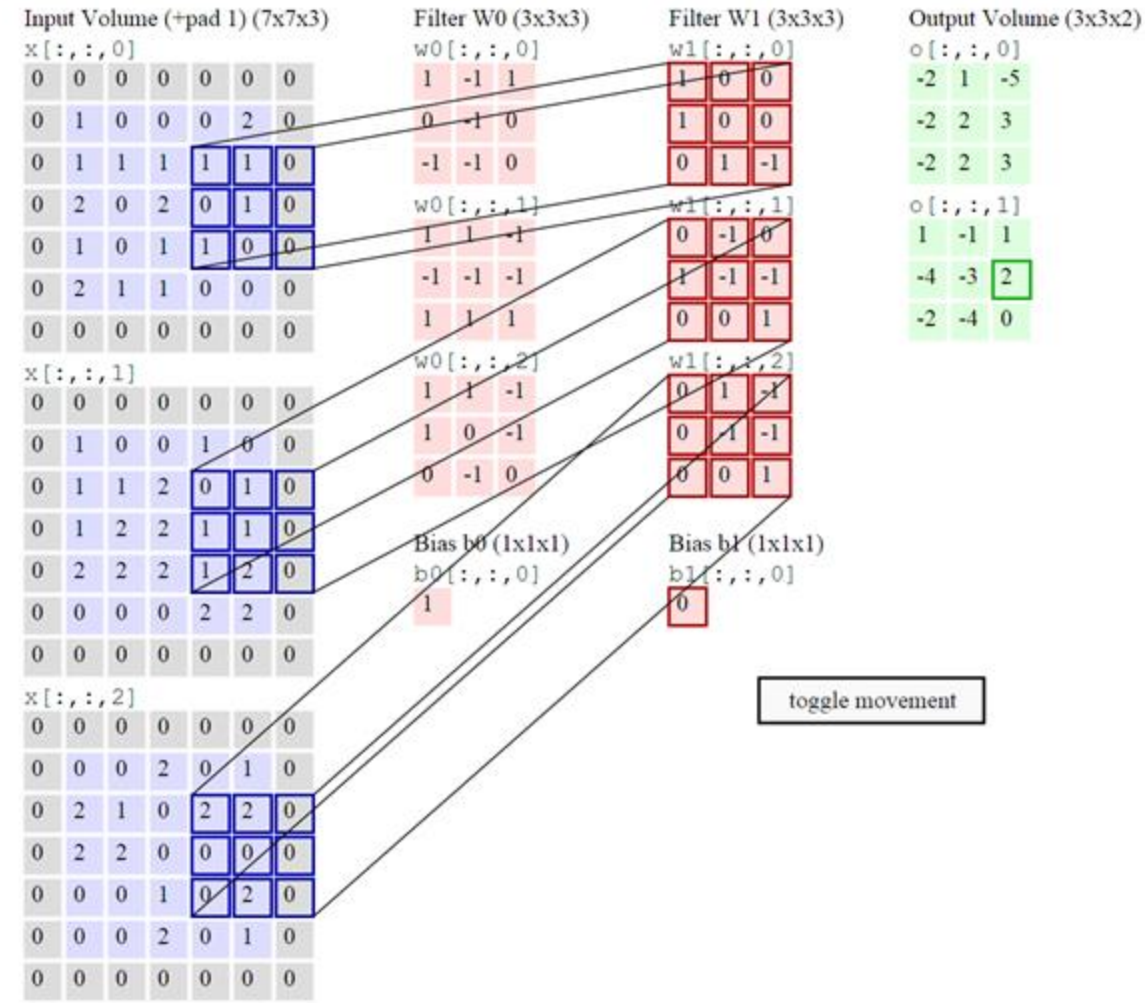
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



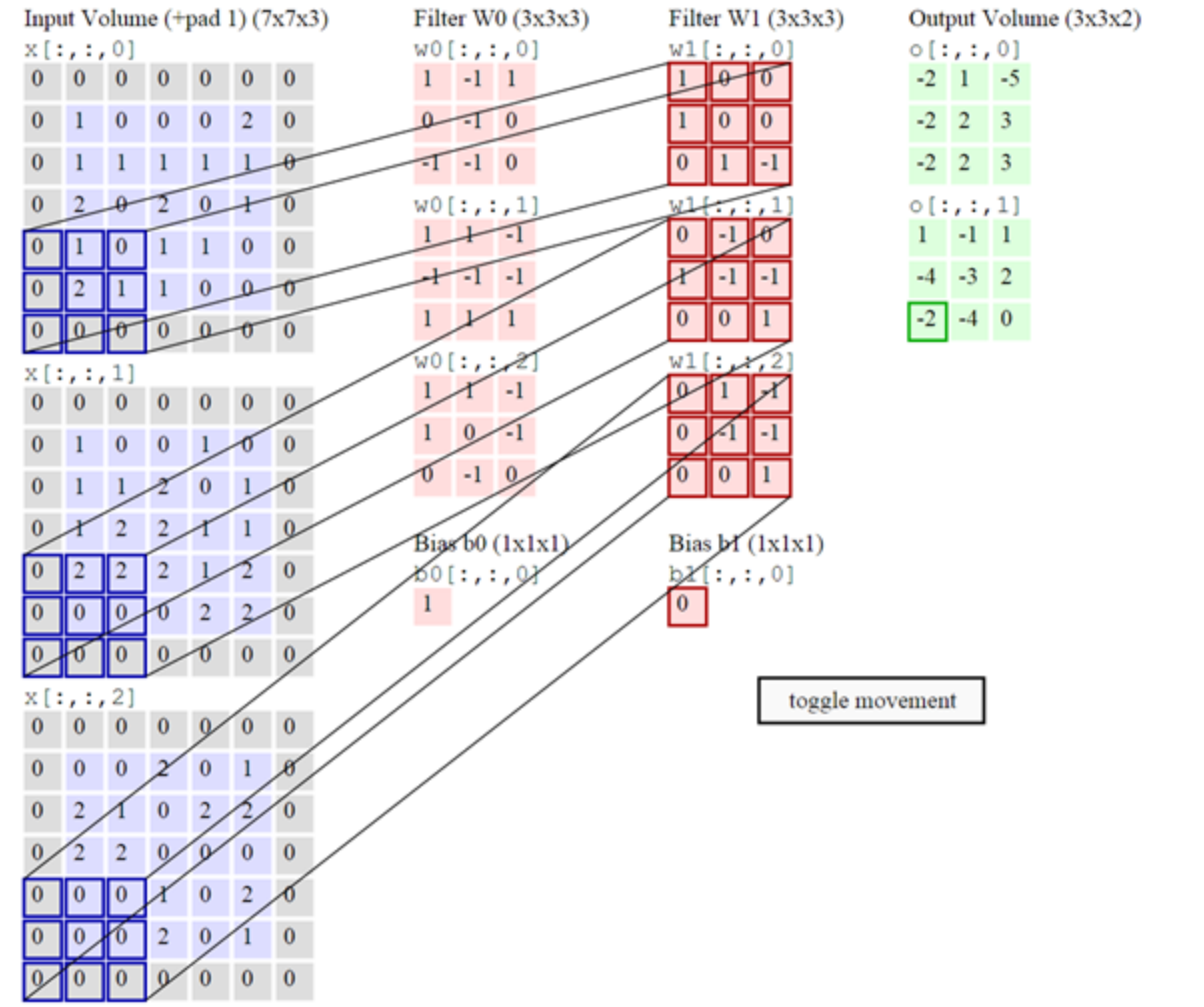
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



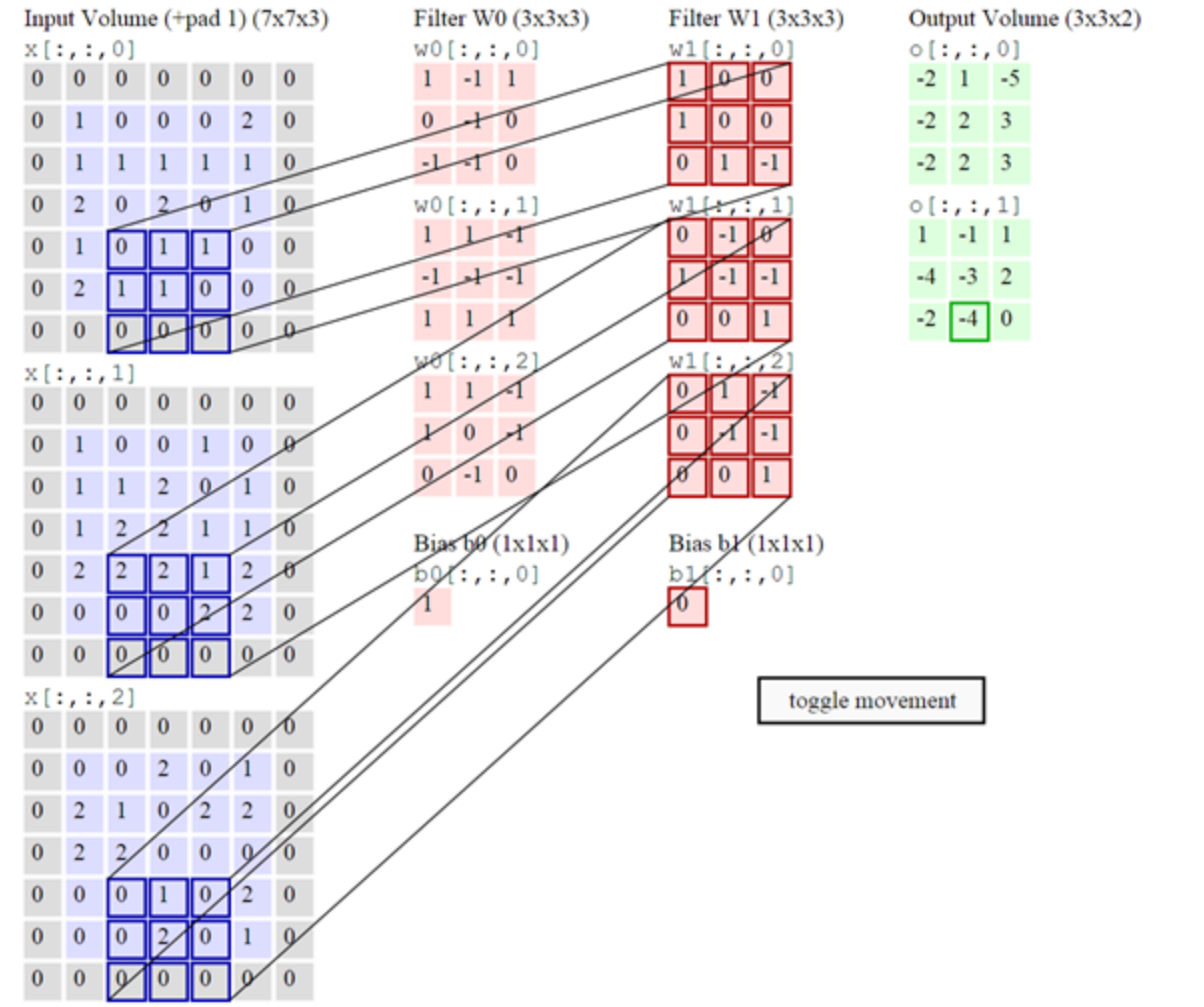
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



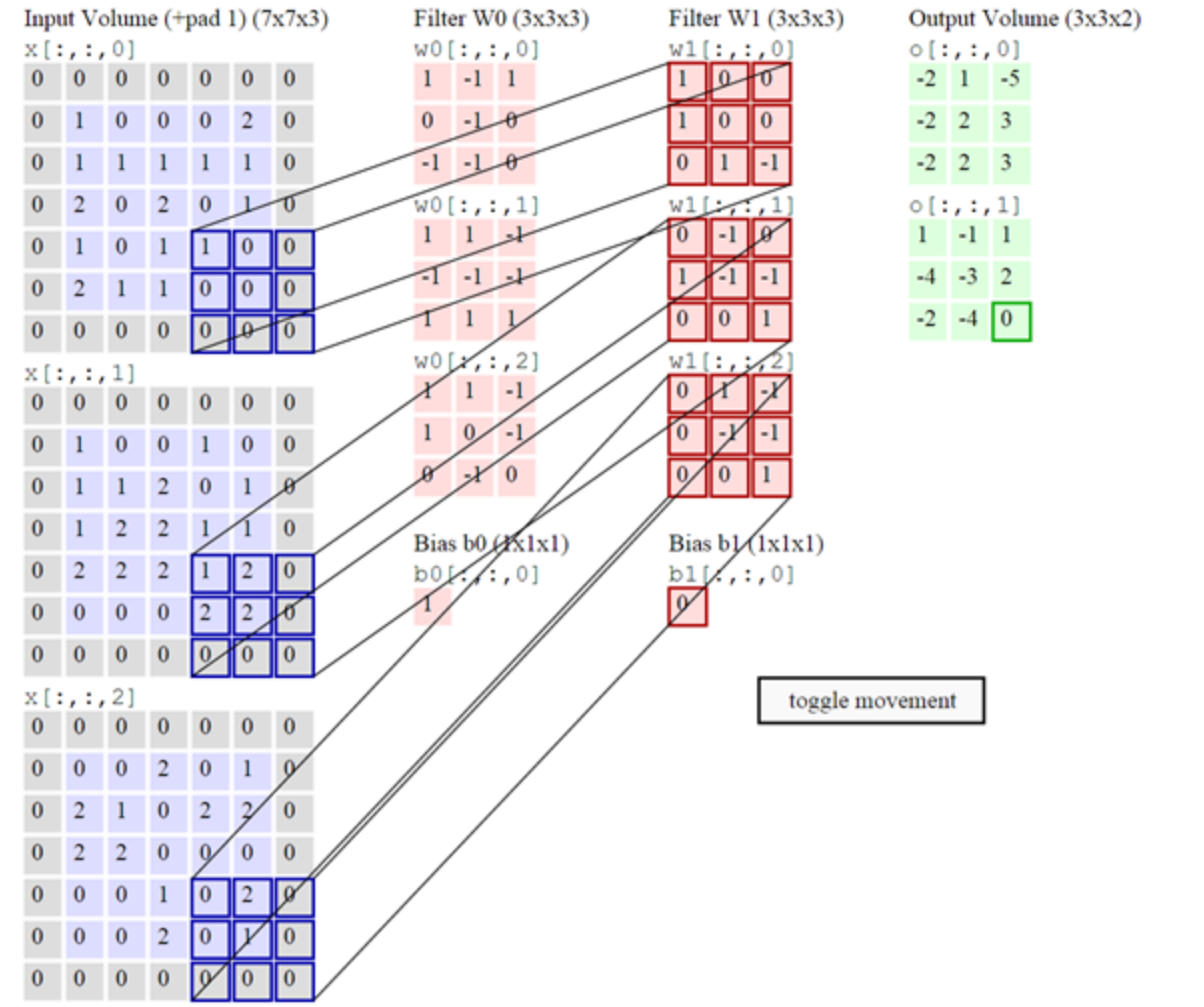
Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais

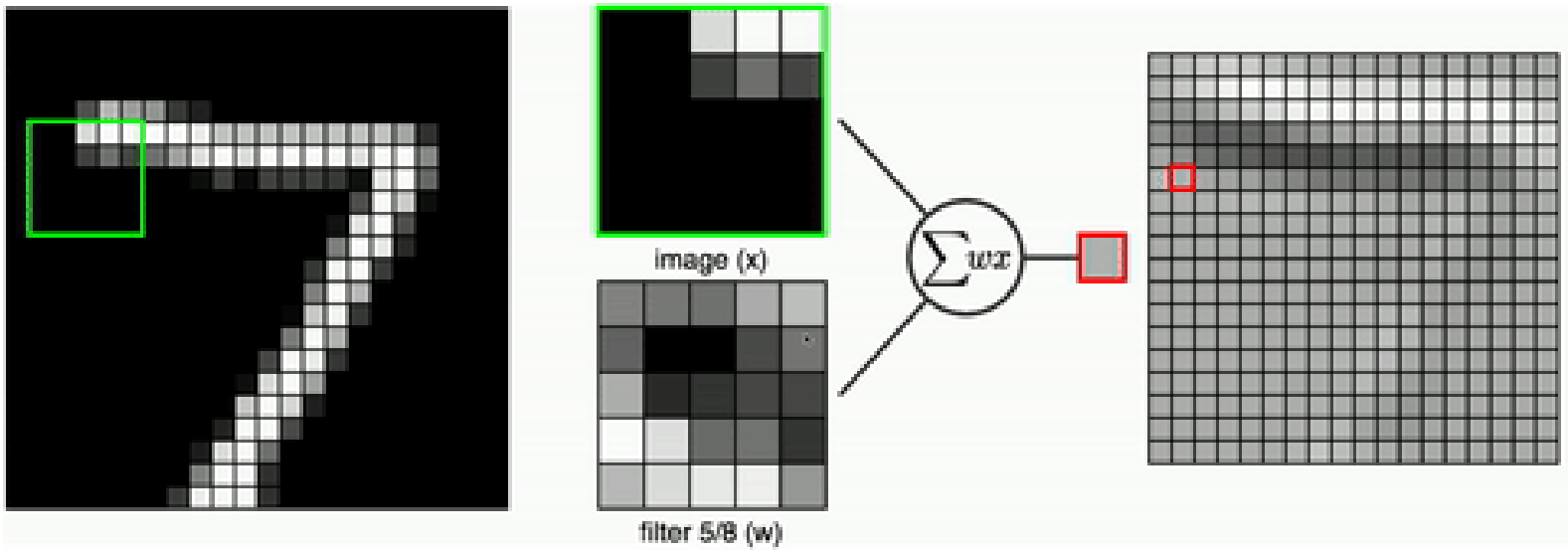


Fonte: http://cs231n.github.io/convolutional-networks/

Redes Neurais Convolucionais



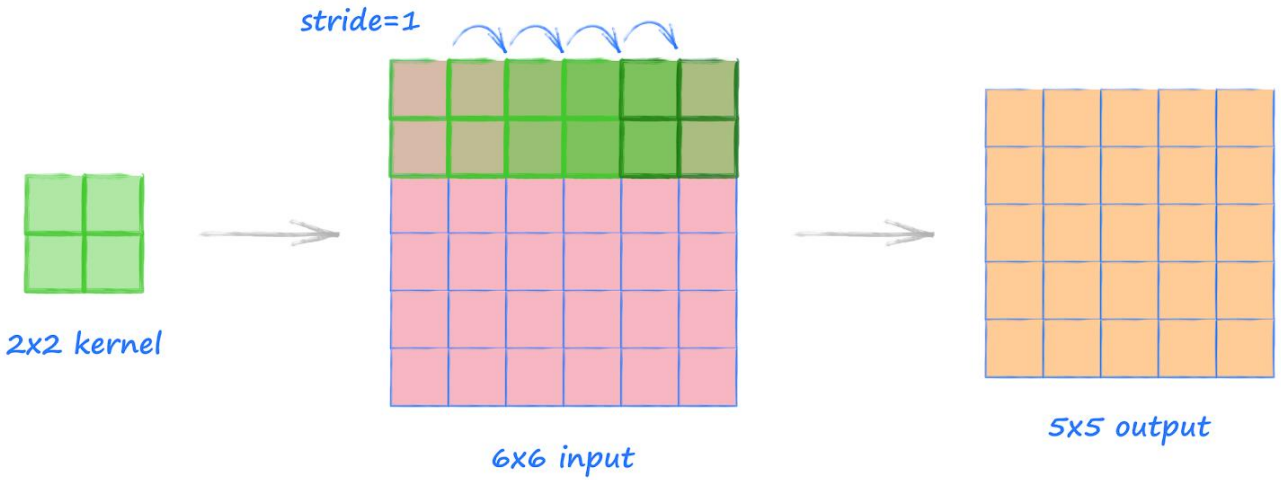
Fonte: http://cs231n.github.io/convolutional-networks/



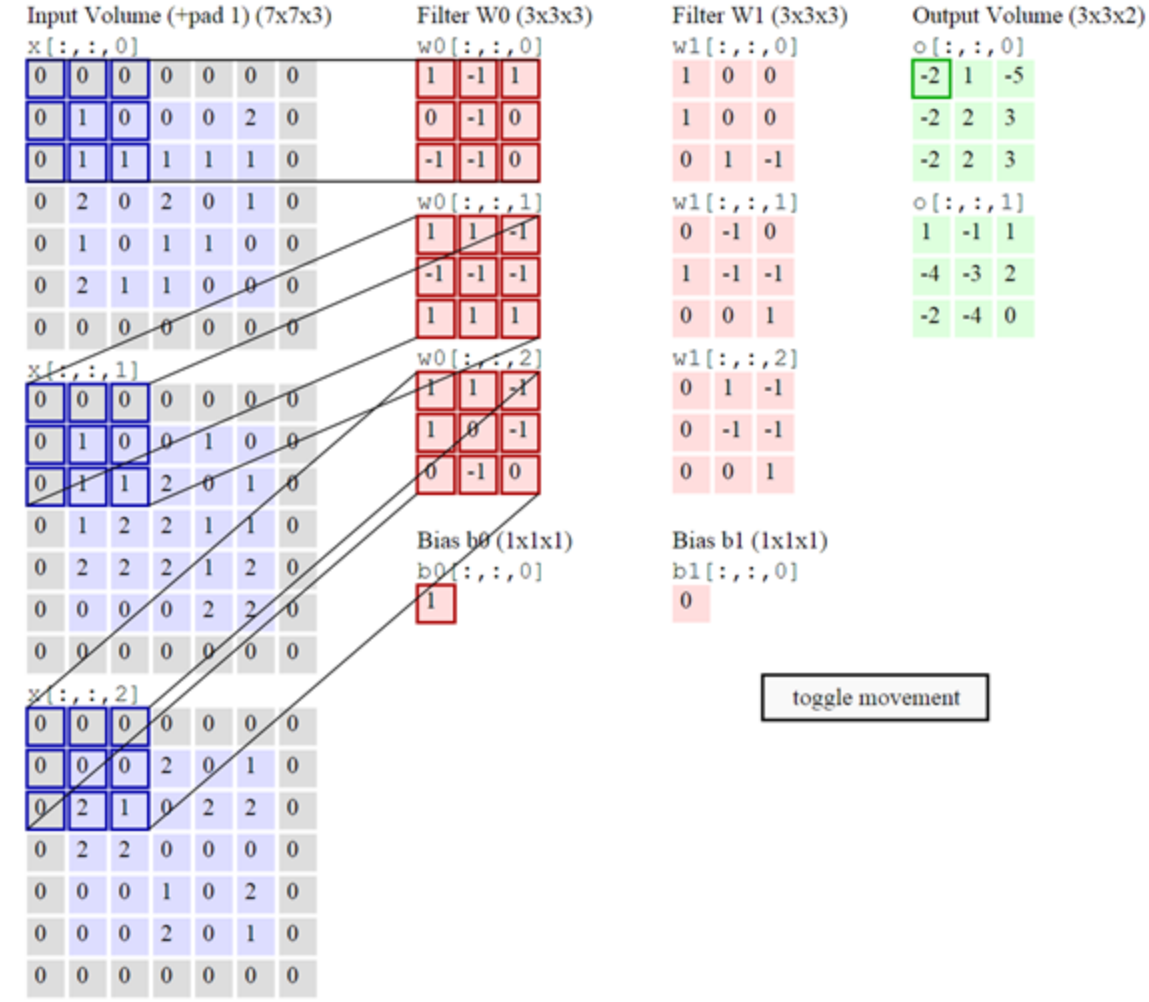
http://ml4a.github.io/dev/demos/demo_convolution.html

Conceitos

- Padding = Preencher as bordas com 0
- Stride = Passo que é dado pelo filtro aplicado

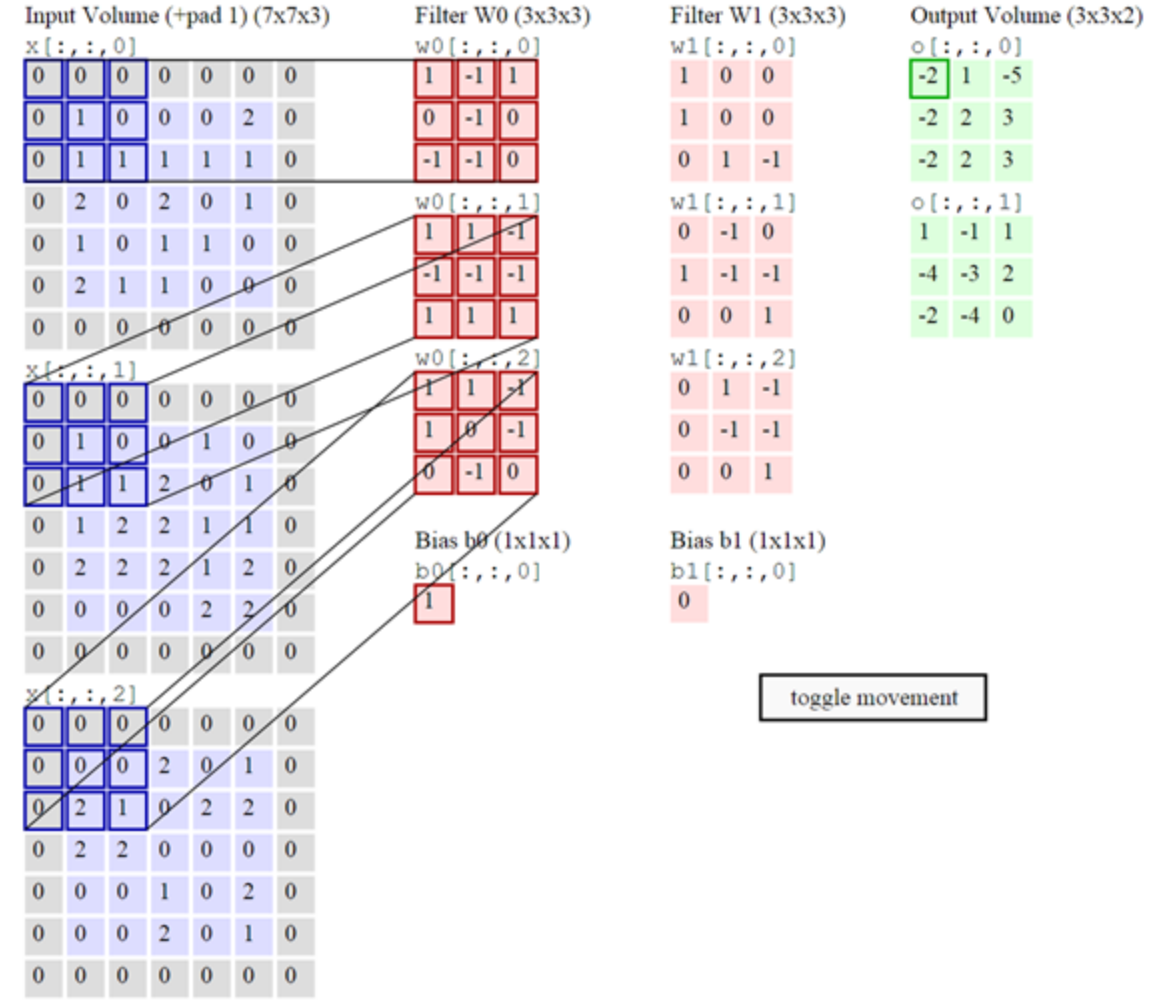


Vamos praticar?



Número de parâmetros desta camada?

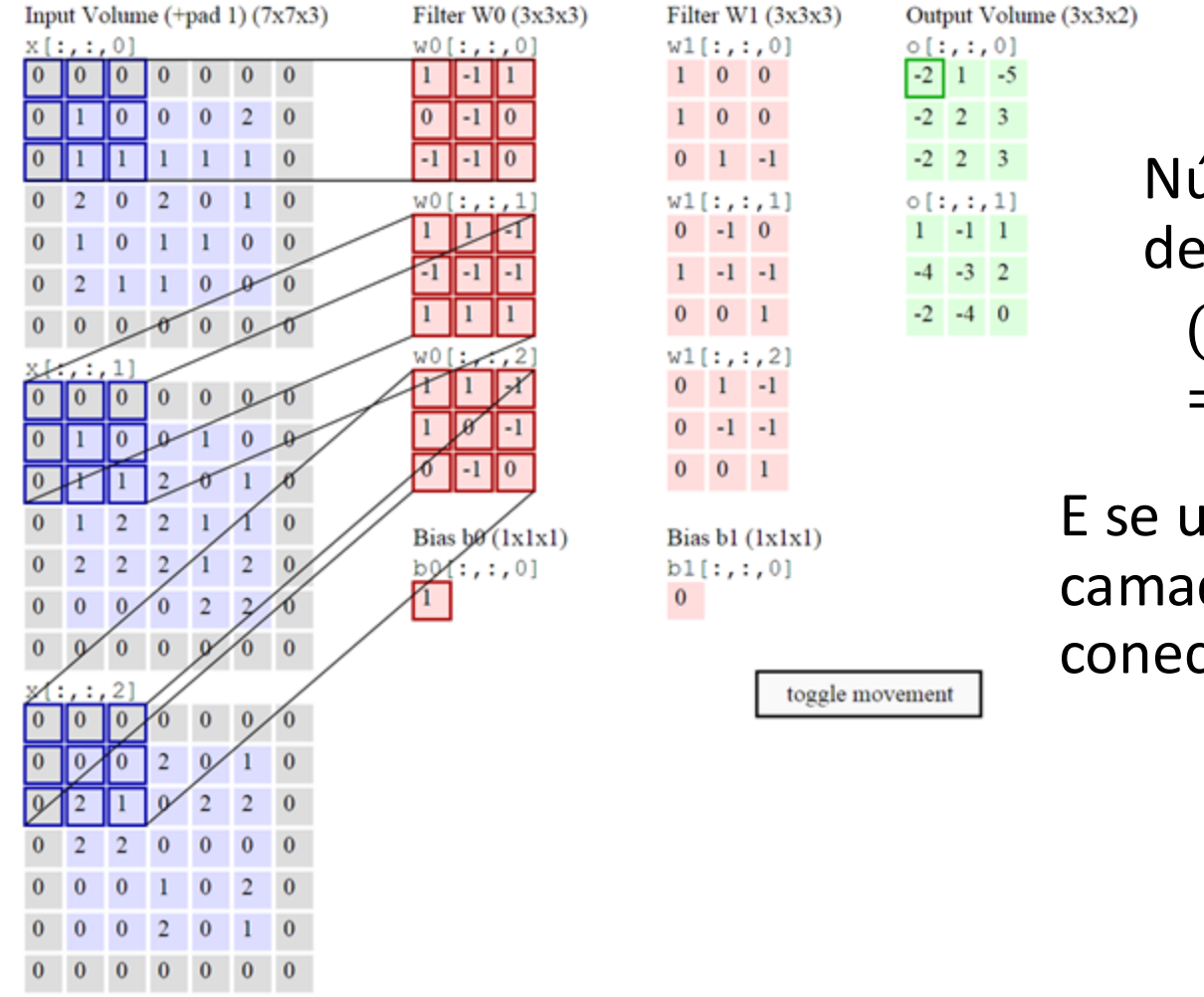
Vamos praticar?



Número de parâmetros desta camada?

$$(3 \times 3 \times 3 + 1) \times 2 = 56$$

Vamos praticar?

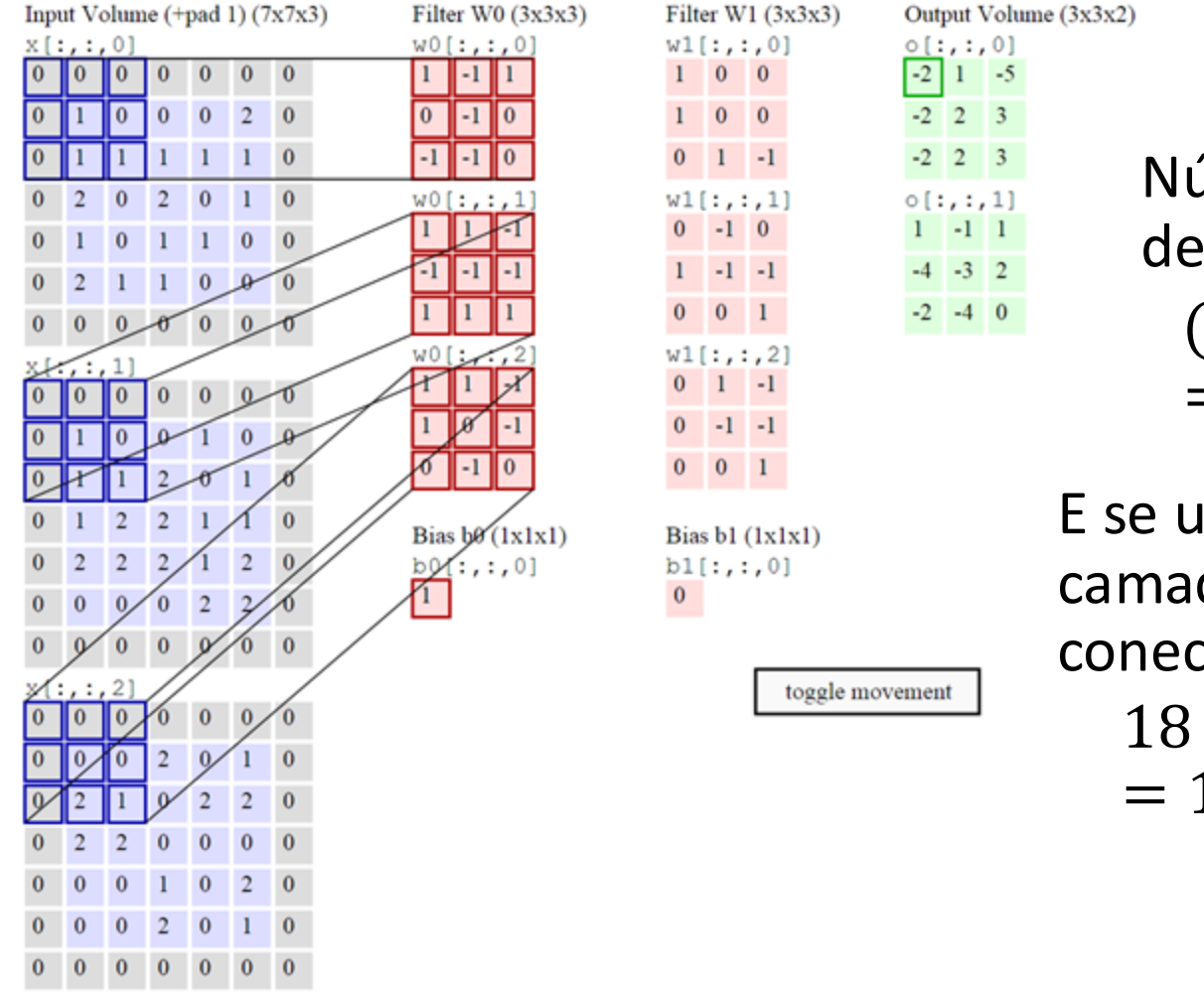


Número de parâmetros desta camada?

$$(3 \times 3 \times 3 + 1) \times 2 = 56$$

E se utilizássemos uma camada totalmente conectada com 18 neurônios?

Vamos praticar?

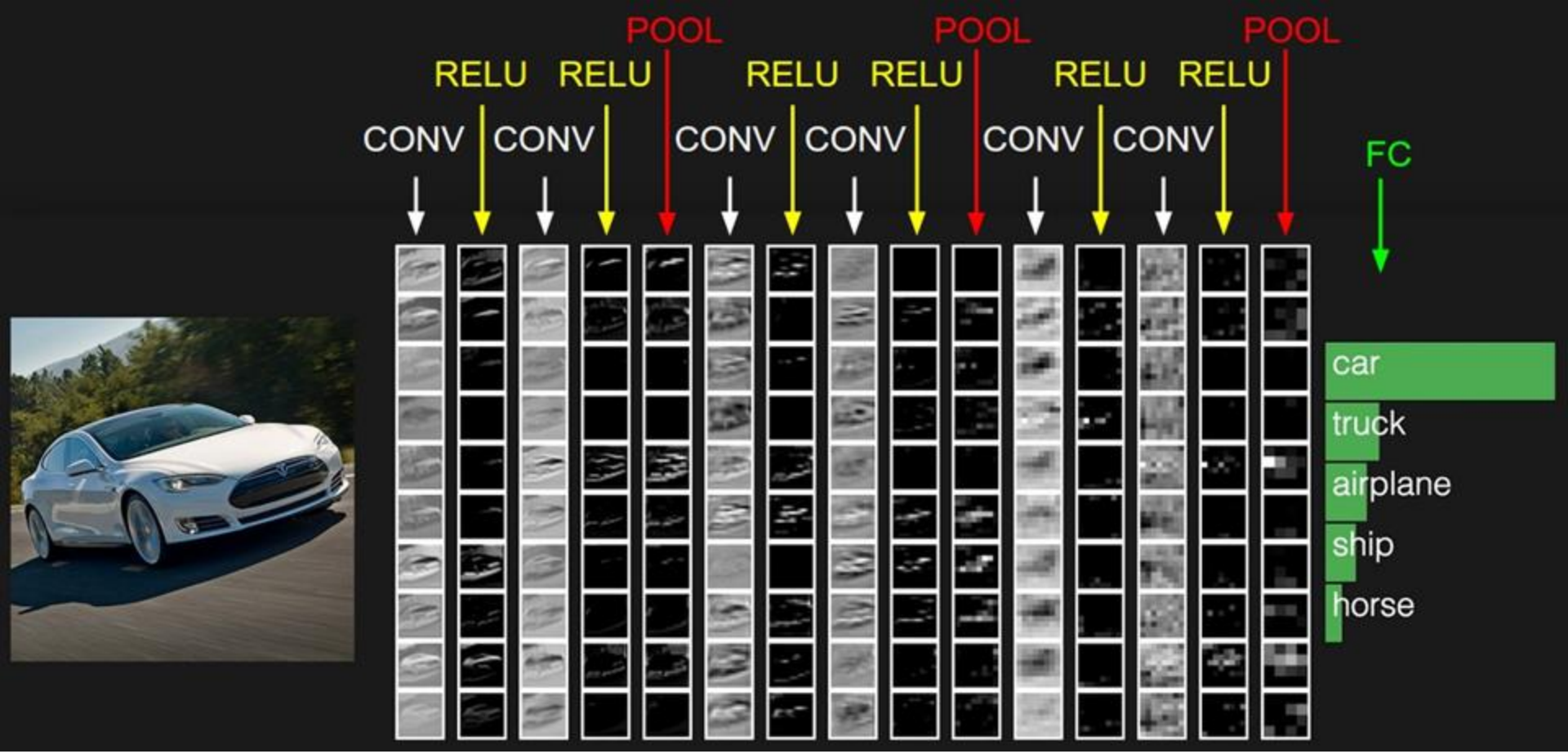


Número de parâmetros desta camada?

$$(3 \times 3 \times 3 + 1) \times 2 = 56$$

E se utilizássemos uma camada totalmente conectada com 18 neurônios?

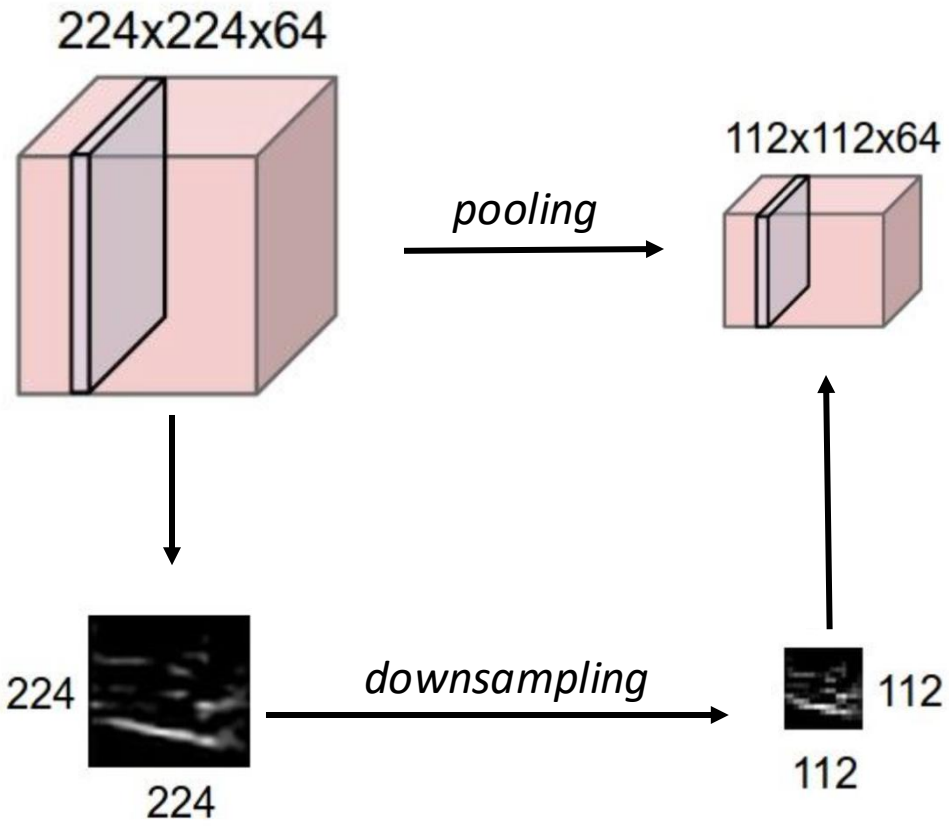
$$18 \times (5 \times 5 \times 3) + 18 = 1368$$



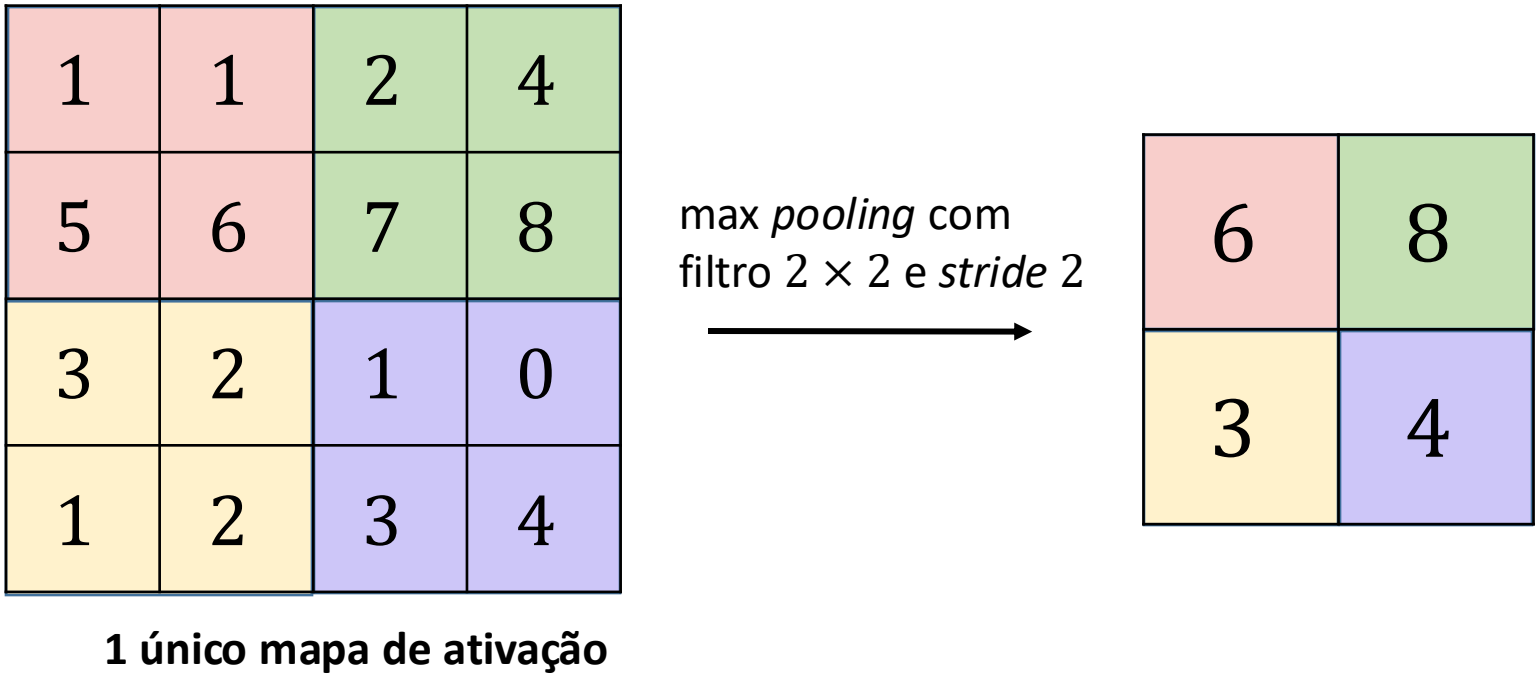
Próxima camada: Pooling!

Camada de *Pooling*

- **Reduz os volumes** para que se tornem gerenciáveis
- Atua sobre cada mapa de ativação de **maneira independente**

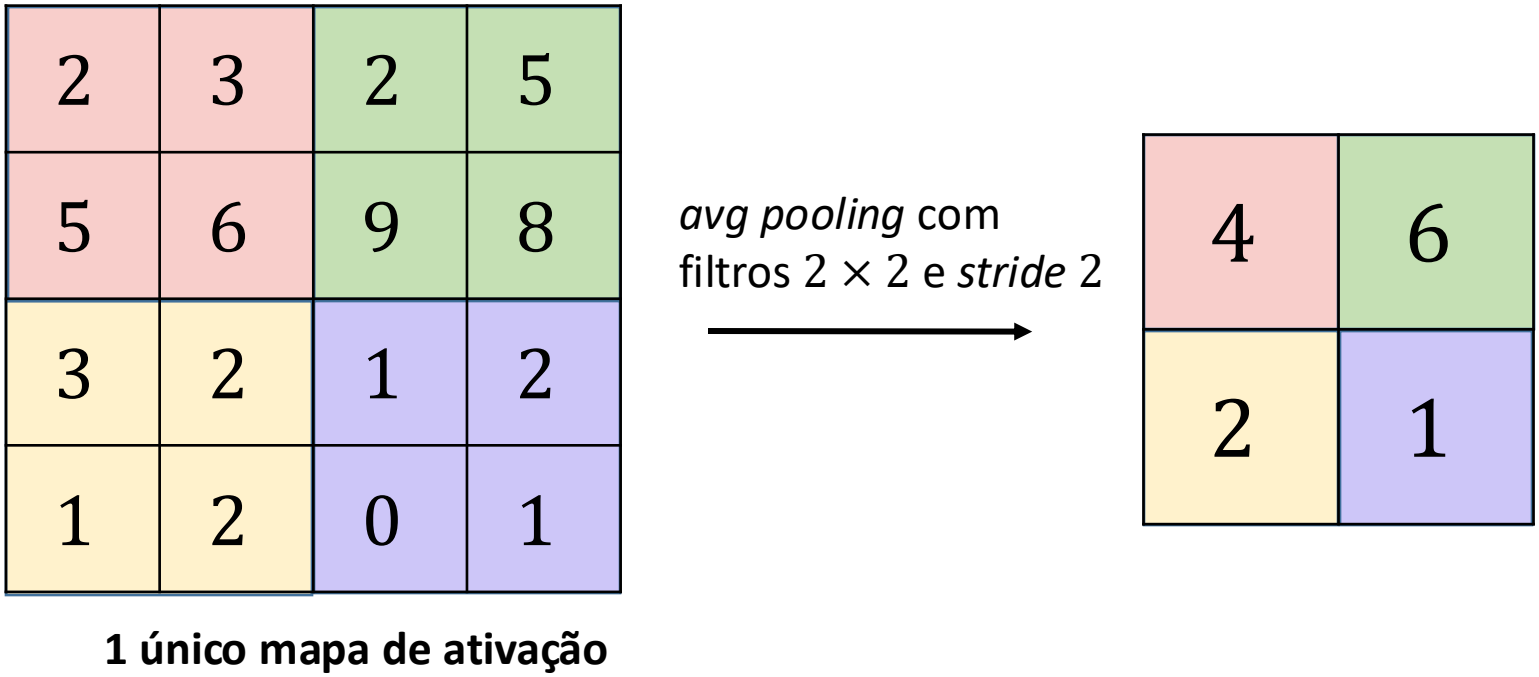


MAX *Pooling*

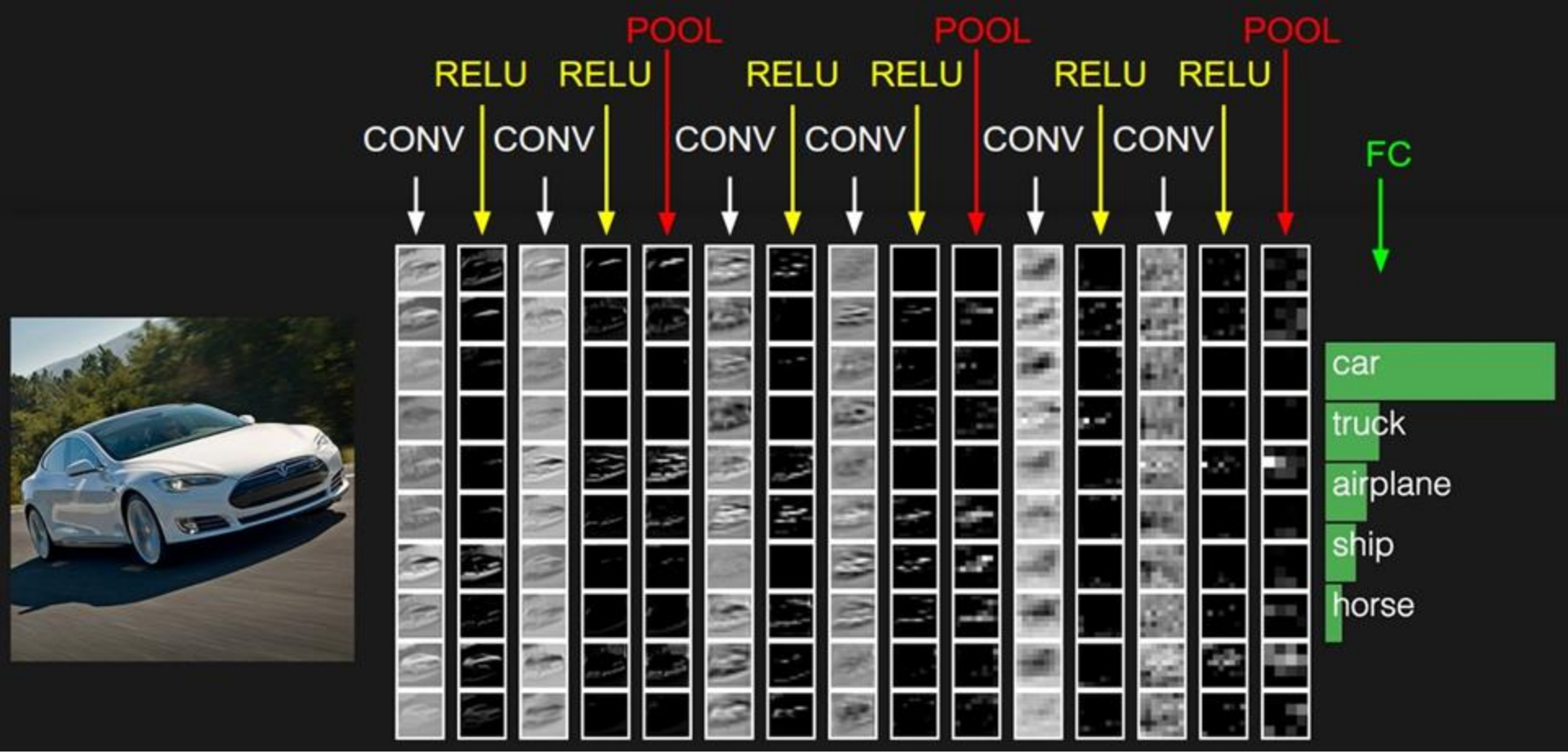


1 único mapa de ativação

AVERAGE (AVG) *Pooling*



1 único mapa de ativação

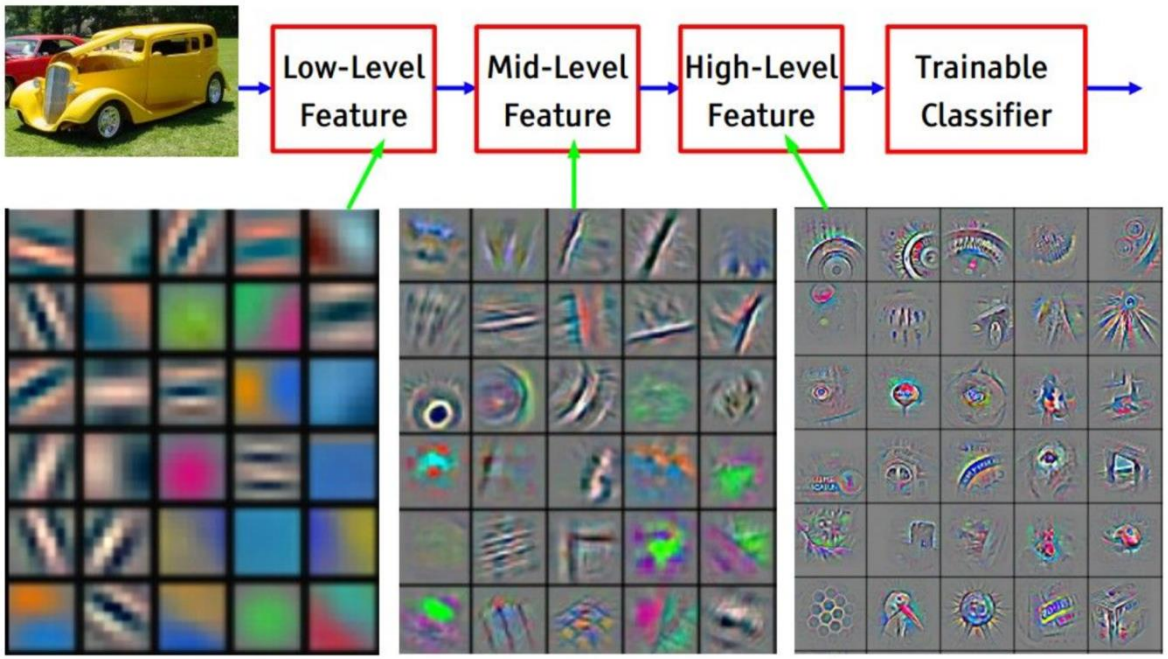


Finalmente: FC (Fully-Connected Layer)

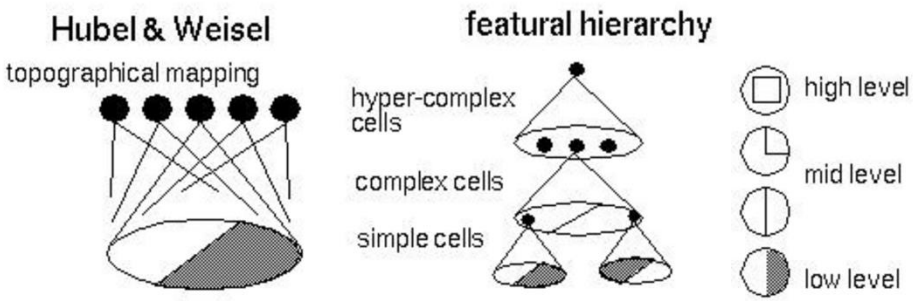
Uma rede convolucional pode conter uma (ou várias) camadas totalmente conectadas, onde os neurônios estão conectados com toda a entrada (redes neurais tradicionais)

Hierarquia de Features

Slides Yann LeCun



Visualização de Features (Rede treinada no ImageNet) [Zeiler & Fergus, 2013]



Exemplos de Arquiteturas: LeNet-5

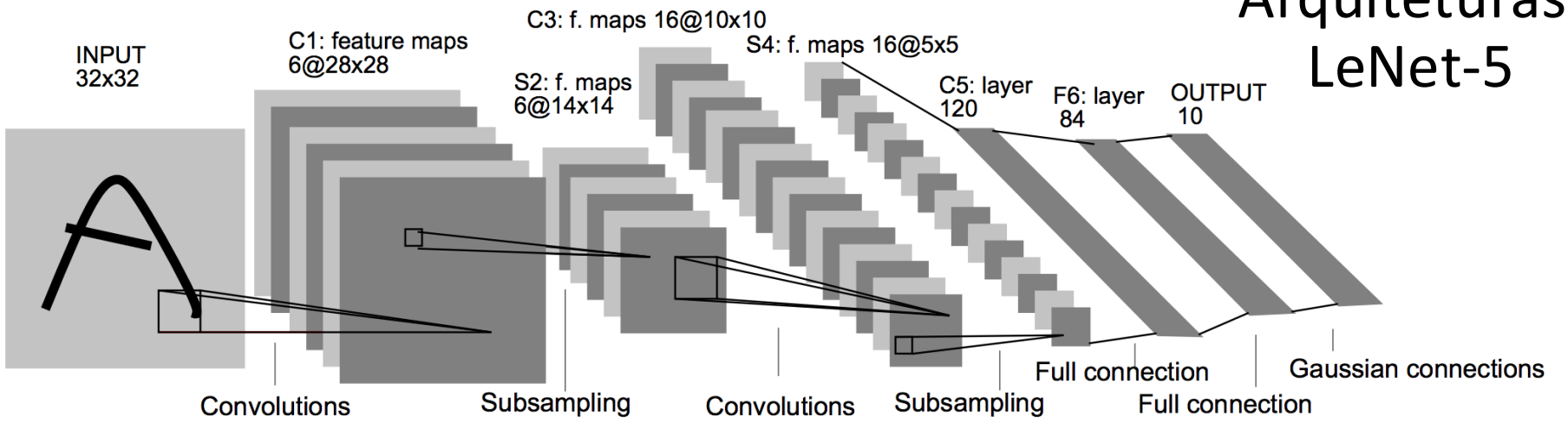


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Filtros Convolucionais: 5×5
Stride: 1

Filtros de Pooling: 2×2
Stride: 2

Arquitetura: [CONV-POOL-CONV-POOL-FC1-FC2-FC3]

Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In F. Fogelman and P. Gallinari, editors, *International Conference on Artificial Neural Networks*, pages 53-60, Paris, 1995.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, november 1998.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In ICLR, 2015.

Exemplos de Arquiteturas: VGGNet

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					

Apenas convoluções 3 × 3 (stride 1, pad 1) e *max pooling* 2 × 2 (stride 2)

melhor modelo

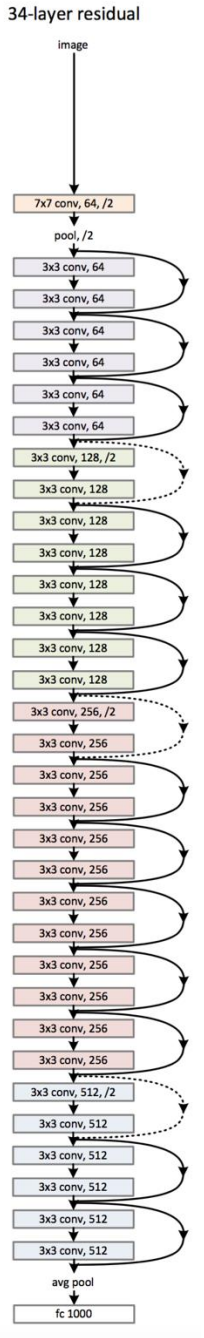
Erro top-5 ImageNet:
7.0% (uma única rede)
6.8% (ensemble com 2 redes)

Table 2: **Number of parameters** (in millions).

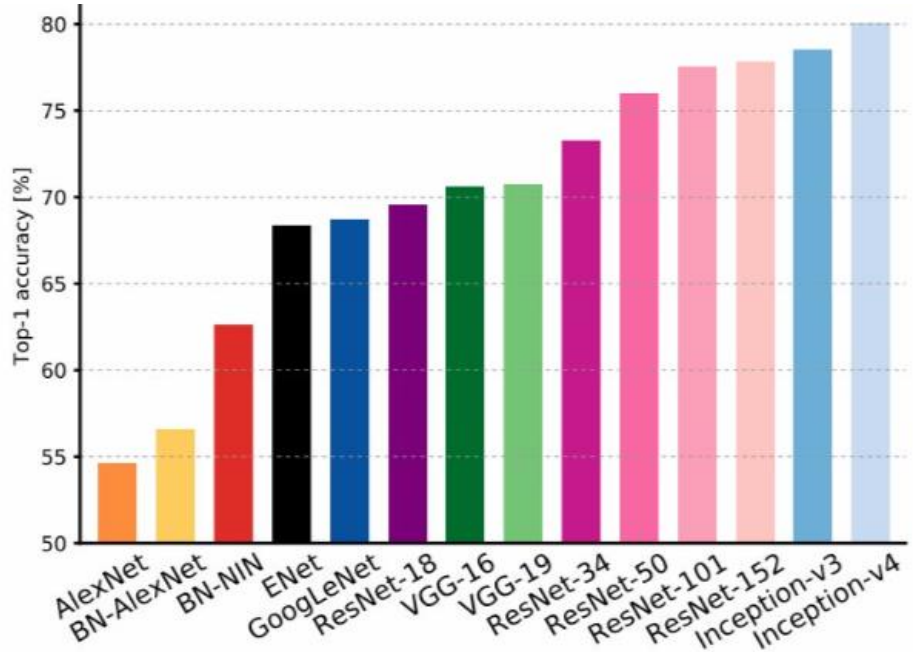
Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

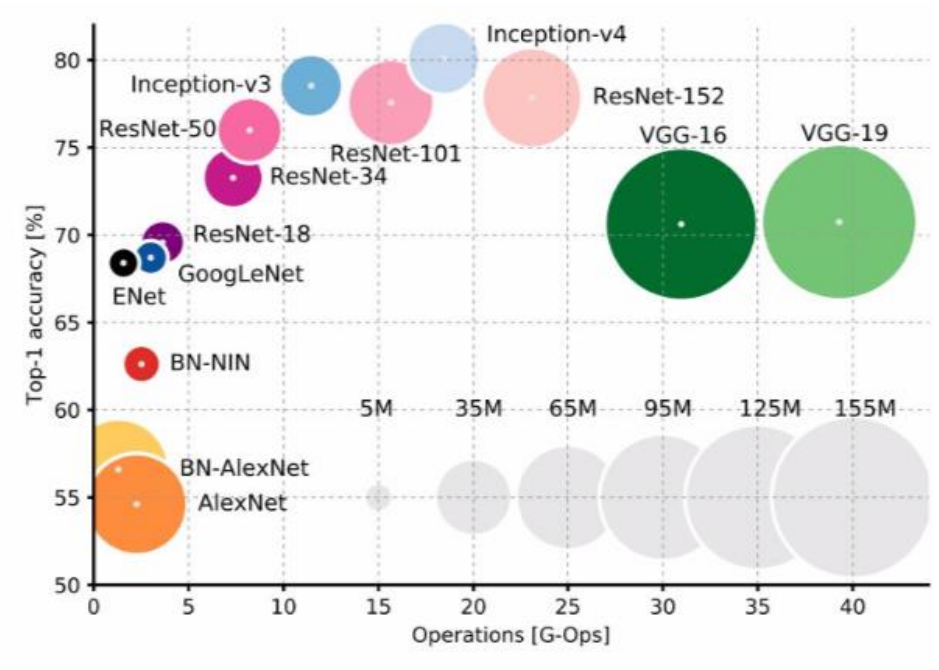
Exemplos de Arquiteturas: ResNet

He, Kaiming, et al. **Deep Residual Learning for Image Recognition**. arXiv preprint arXiv:1512.03385, 2015.



Comparação de Complexidade





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

Transfer Learning

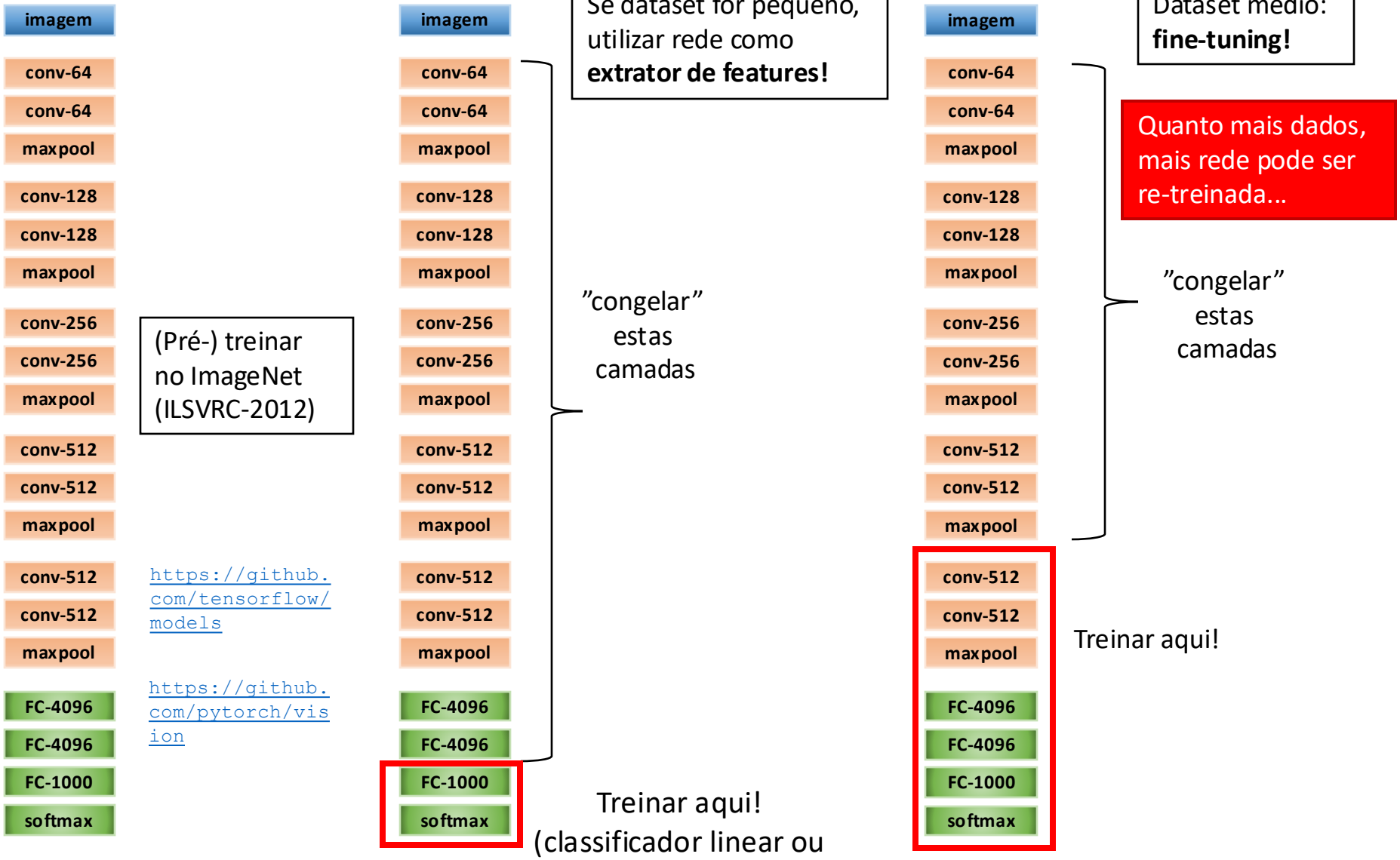
”Precisamos utilizar uma base de dados gigante” 

”Precisamos treinar uma ConvNet do zero” 

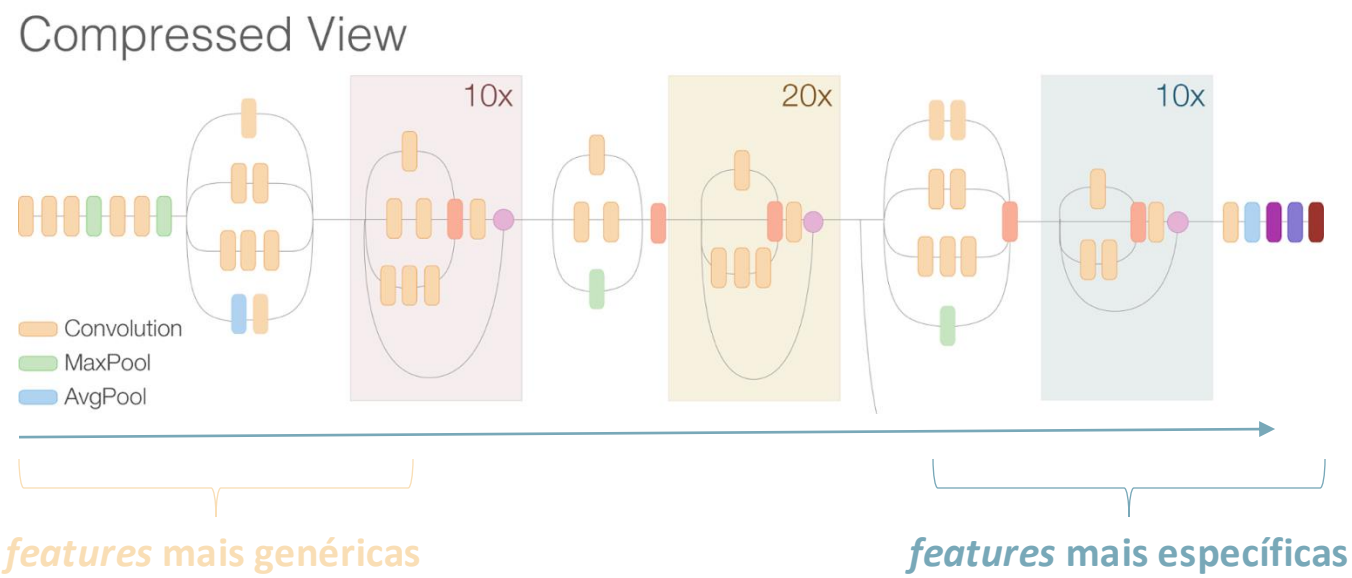
”*Transfer learning* (*inductive transfer*) é uma área do aprendizado de máquina que foca em **armazenar conhecimento adquirido** ao se resolver determinada tarefa para **posterior aplicação** em **problema diferente mas relacionado**.”

West, Jeremy, Dan Ventura, and Sean Warnick. Spring Research Presentation: **A Theoretical Foundation for Inductive Transfer**. Brigham Young University, College of Physical and Mathematical Sciences. 2007.

Transfer Learning em ConvNets



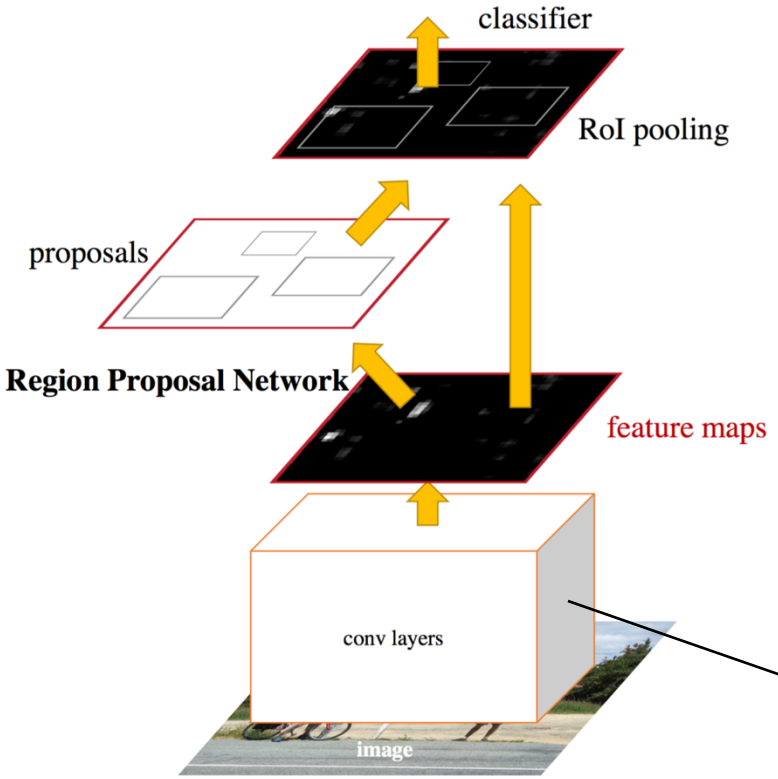
Transfer Learning em ConvNets



	Dados similares ao contexto em questão	**Dados muito diferentes**
Poucos dados	Usar um classificador linear na saída	😔 😢
Muitos dados	Fine-tune algumas camadas...	Fine-tune algumas camadas a mais...

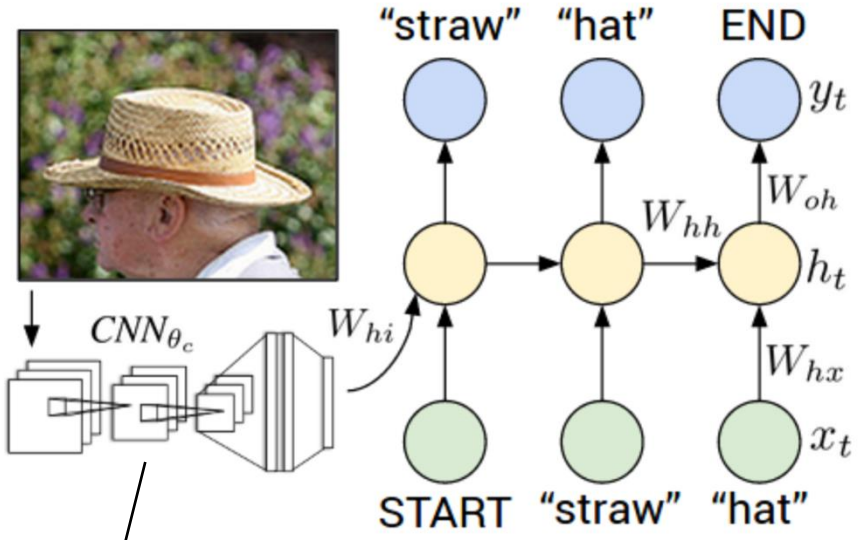
Transfer Learning em ConvNets

Transfer Learning em ConvNets é a regra (e não exceção!!)



Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun. **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.** NIPS 2015, 91-99

Andrej Karpathy, Fei-Fei Li. **Deep visual-semantic alignments for generating image descriptions.** CVPR 2015, 3128-3137



redes pré-treinadas no ImageNet!!

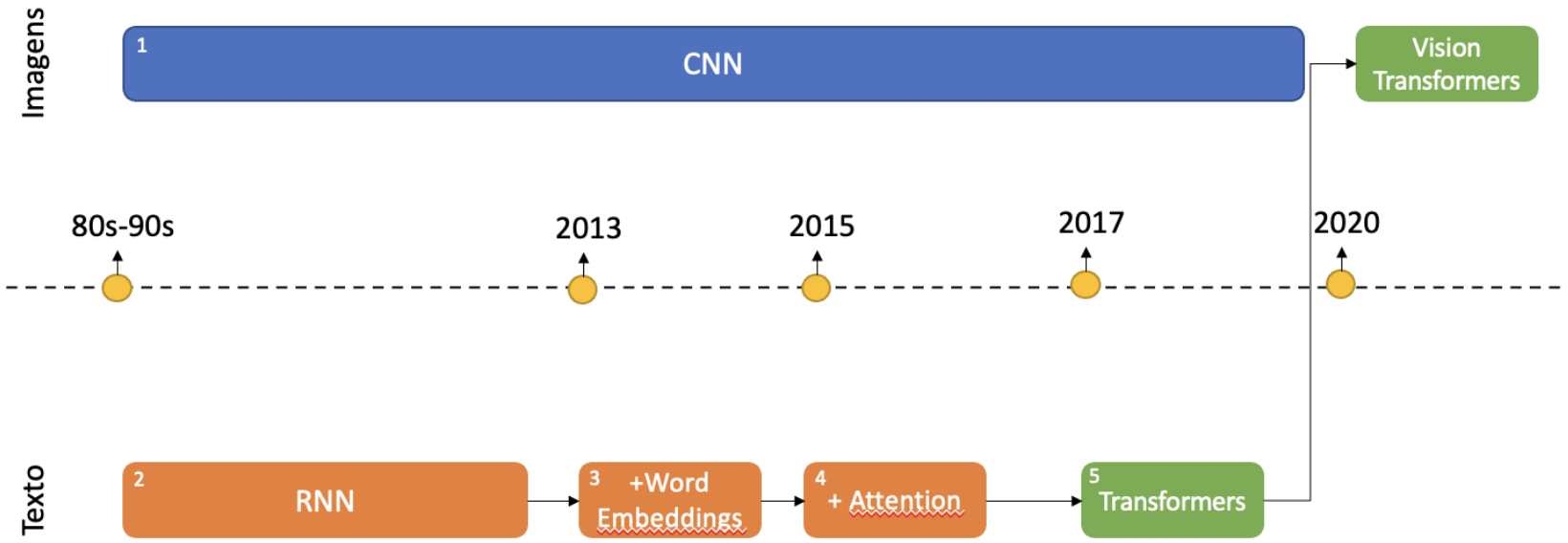
Transfer Learning em ConvNets

Conclusão:

1. Encontre dataset muito grande (>1M) com dados similares aos do seu interesse e (pré-) treine uma ConvNet lá! (ou pegue o modelo treinado de algum repositório de modelos, e.g. model zoo)
2. Faça transfer learning para o seu dataset (atenção para escolha da estratégia mais adequada de acordo com o que foi apresentado)

Caffe → https://github.com/BVLC/caffe/wiki/Model-Zoo
TensorFlow → https://github.com/tensorflow/models
PyTorch → https://github.com/pytorch/vision

Deep Learning



Leituras Sugeridas

- Capítulo 9 do livro: