UNICAMP

INF-0618
Tópicos em Aprendizado de Máquina II

Aula 2 – Redes Neurais Convolucionais

Profa. Fernanda Andaló

2018

Instituto de Computação - Unicamp

Roteiro

Deep Learning e CNNs

Camada convolucional

Camada fully-connected

Camada de pooling

Exemplo CNN

Prática

Deep Learning e CNNs

Deep Learning

Aprendizado por meio de uma rede neural profunda (*deep*), que pode ser MLP, convolucional (CNN), recorrente (RNN), ...

Deep Learning

Aprendizado por meio de uma rede neural profunda (*deep*), que pode ser MLP, convolucional (CNN), recorrente (RNN), ...

Por que deep learning?

Deep Learning

Aprendizado por meio de uma rede neural profunda (*deep*), que pode ser MLP, convolucional (CNN), recorrente (RNN), ...

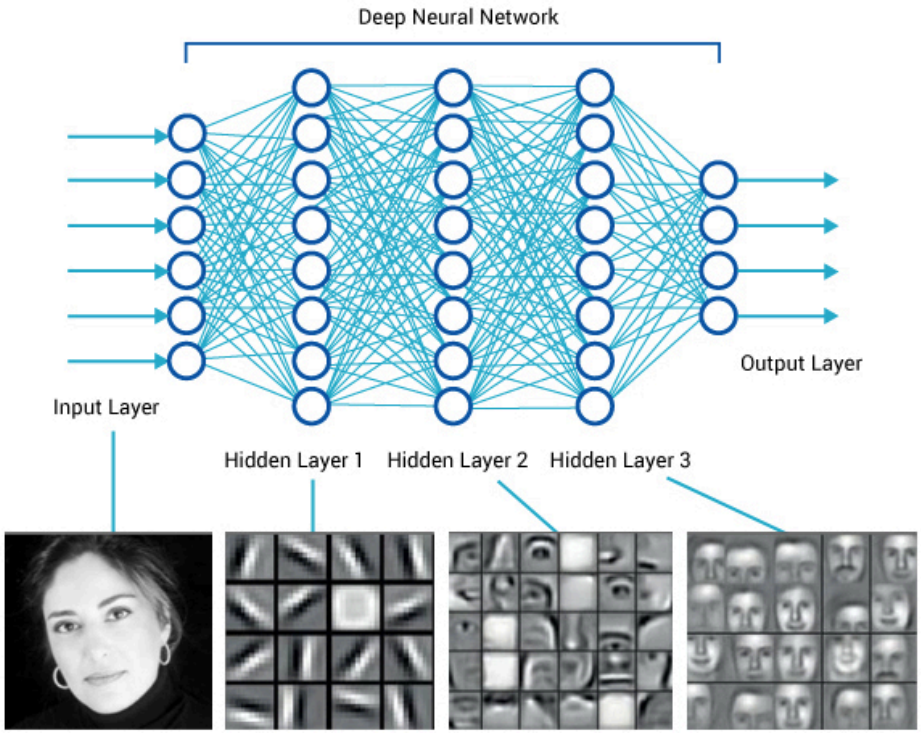
Por que deep learning?

Quanto mais camadas intermediárias (*hidden*) e quando mais dados para treinar estas camadas, mais complexas são as funções que podem ser aprendidas pelos modelos.

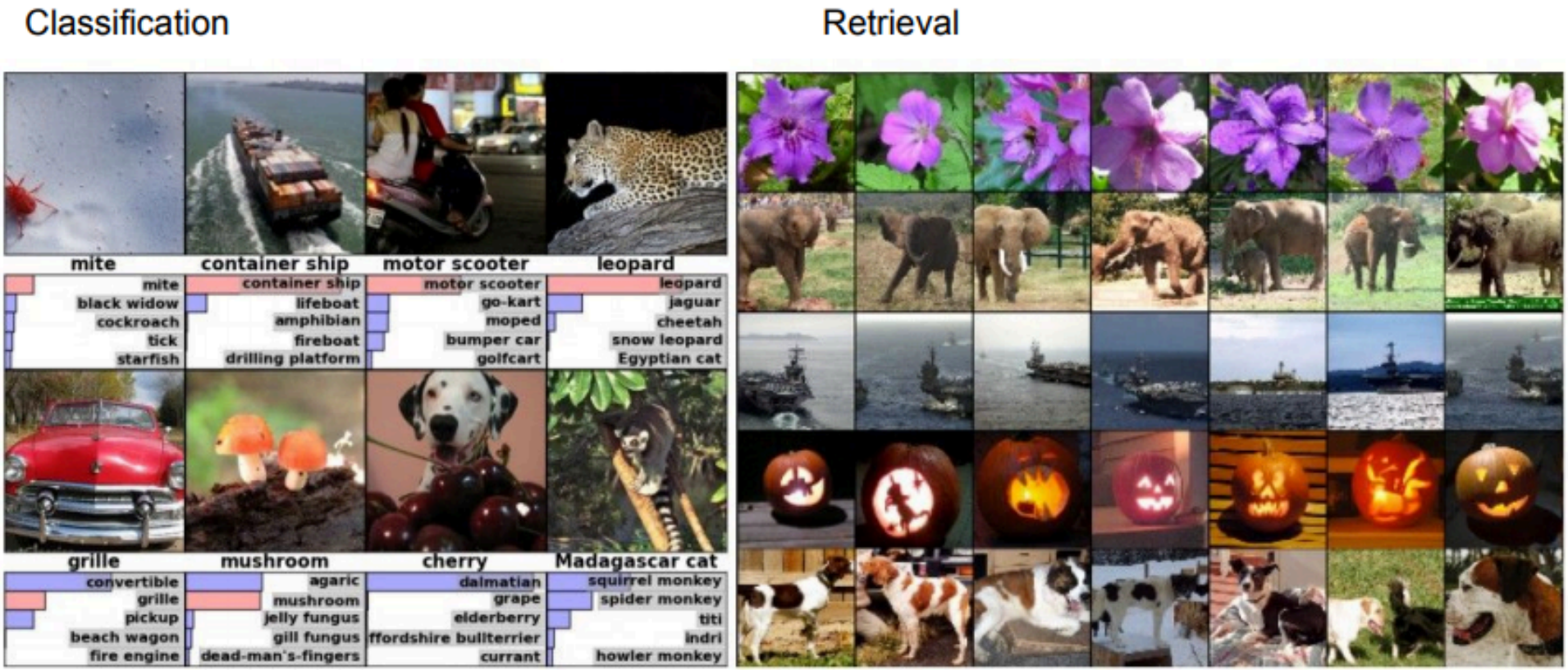
Deep Learning

Aprendizado por meio de uma rede neural profunda (*deep*), que pode ser MLP, convolucional (CNN), recorrente (RNN), ...

Por que deep learning?



Aplicações



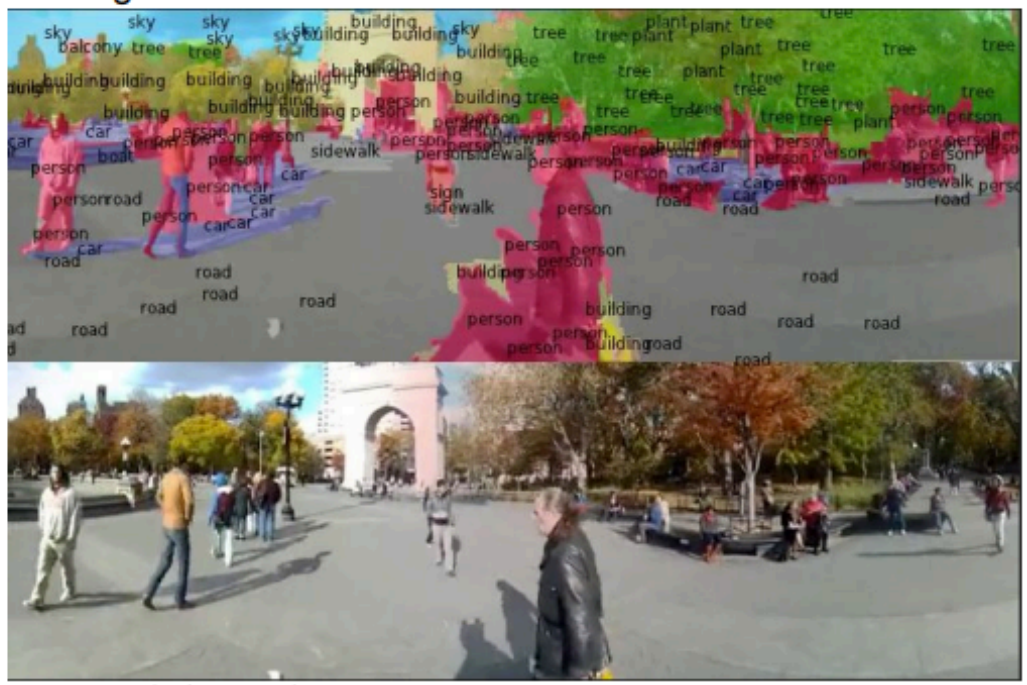
Aplicações

Detection

Figures copyright Shaoqing Ren, Kaiming He, Ross Girschick, Jian Sun, 2015. Reproduced with permission.

[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation



Figures copyright Clement Farabet, 2012. Reproduced with permission.

[Farabet et al., 2012]

Aplicações

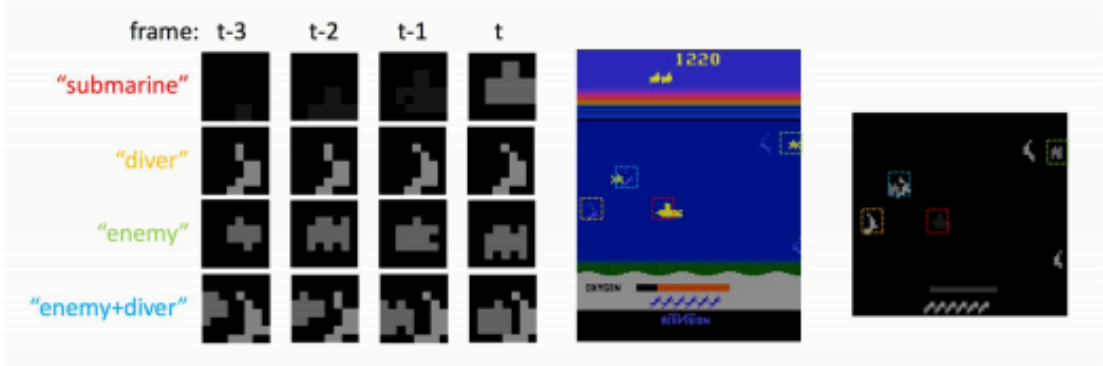
Photo by Lane McIntosh. Copyright CS231n 2017.

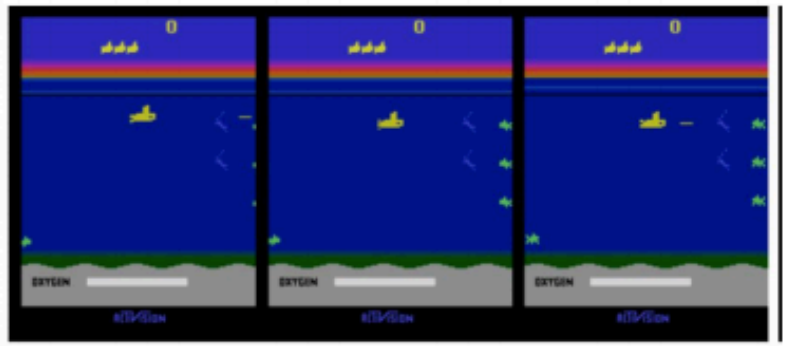
self-driving cars

Aplicações

Images are examples of pose estimation, not actually from Toshev & Szegedy 2014. Copyright Lane McIntosh.

[Toshev, Szegedy 2014]

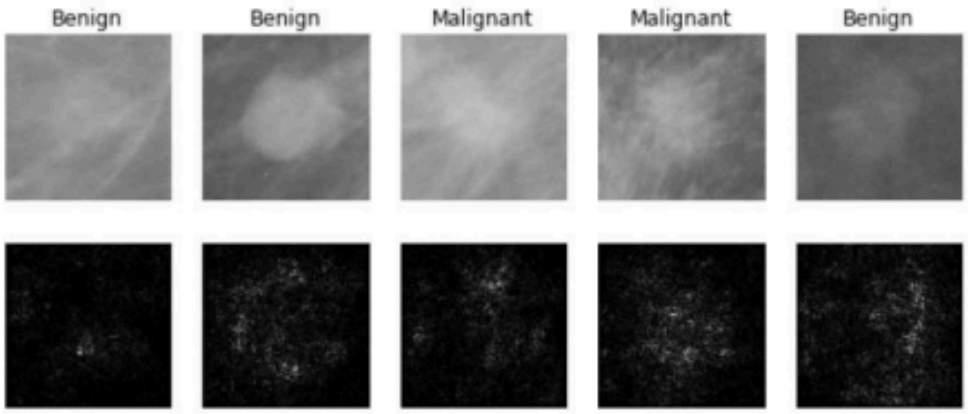




[Guo et al. 2014]

Figures copyright Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard Lewis, and Xiaoshi Wang, 2014. Reproduced with permission.

Aplicações



[Levy et al. 2016]

[Dieleman et al. 2014]

[Sermanet et al. 2011]
[Ciresan et al.]

Aplicações

This image by Christin Khan is in the public domain and originally came from the U.S. NOAA.

Whale recognition, Kaggle Challenge

Photo and figure by Lane McIntosh; not actual example from Mnih and Hinton, 2010 paper.

Mnih and Hinton, 2010

Aplicações

No errors

A white teddy bear sitting in the grass

A man riding a wave on top of a surfboard

Minor errors

A man in a baseball uniform throwing a ball

A cat sitting on a suitcase on the floor

Somewhat related

A woman is holding a cat in her hand

A woman standing on a beach holding a surfboard

Image Captioning

[Vinyals et al., 2015]
[Karpathy and Fei-Fei, 2015]

All images are CC0 Public domain:
https://pixabay.com/en/luggage-antique-cat-1643010/
https://pixabay.com/en/teddy-plush-bears-cute-teddy-bear-1623436/
https://pixabay.com/en/surf-wave-summer-sport-litoral-1668716/
https://pixabay.com/en/woman-female-model-portrait-adult-983967/
https://pixabay.com/en/handstand-lake-meditation-496008/
https://pixabay.com/en/baseball-player-shortstop-infield-1045263/

Captions generated by Justin Johnson using NeuralTalk2

Aplicações

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

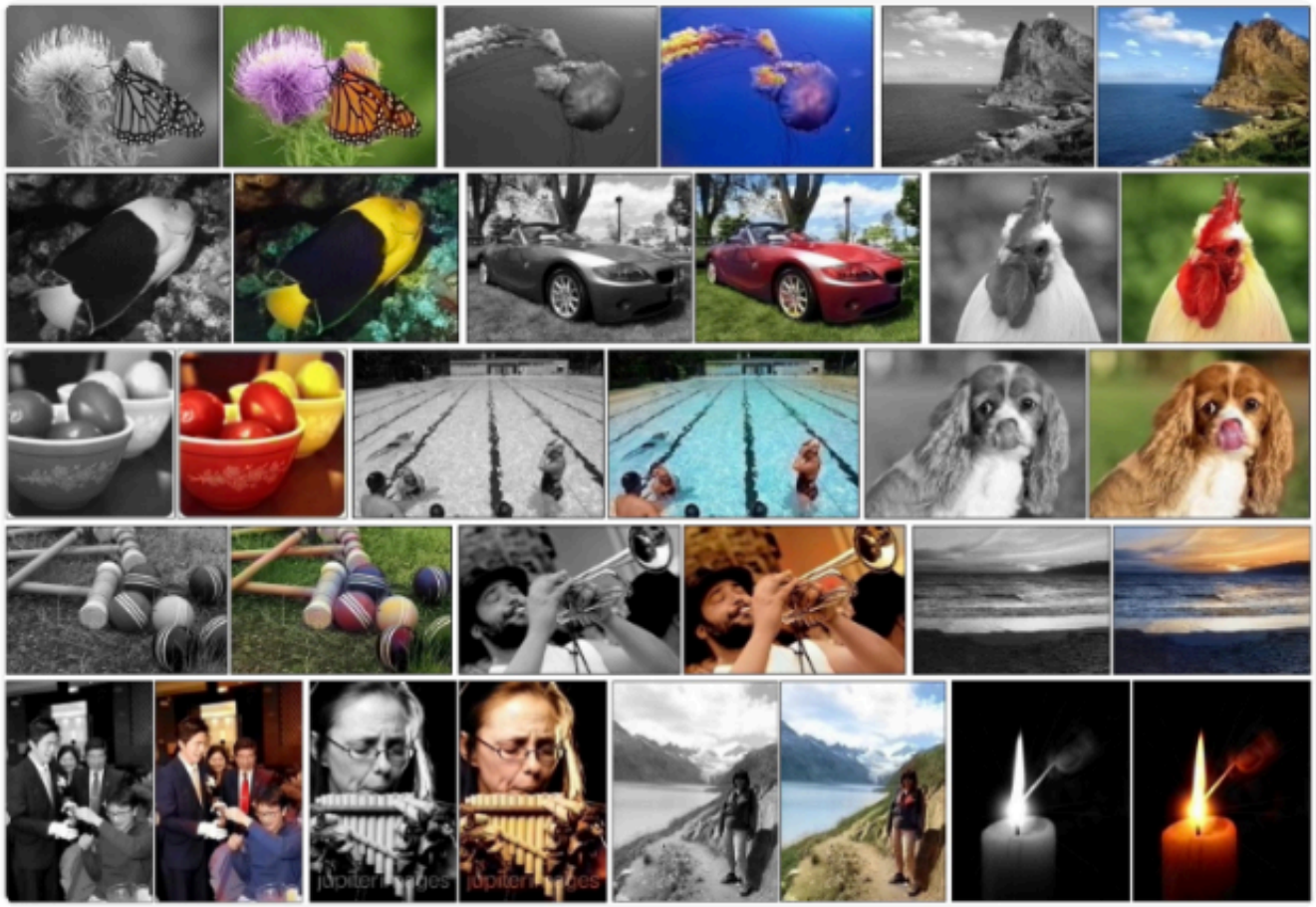
Automatic Image Caption Generation
Sample taken from Andrej Karpathy, Li Fei-Fei

Aplicações

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;
reproduced with permission

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

Aplicações



Colorization of Black and White Photographs

Image taken from Richard Zhang, Phillip Isola and Alexei A. Efros.

Aplicações

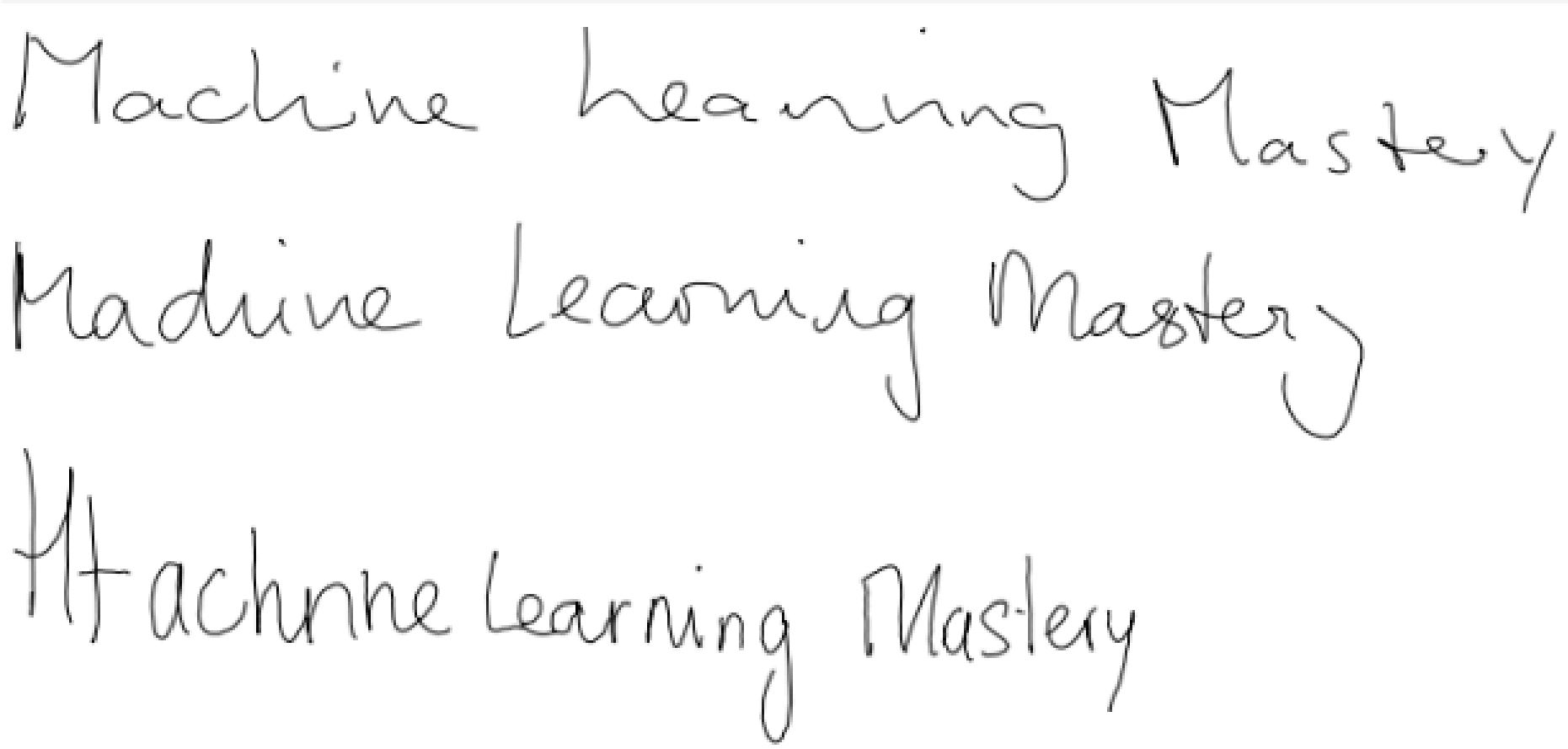
Visually Indicated Sounds [Owens, 2015]

Aplicações

Instant Visual Translation

Example of instant visual translation, taken from the Google Blog.

Aplicações



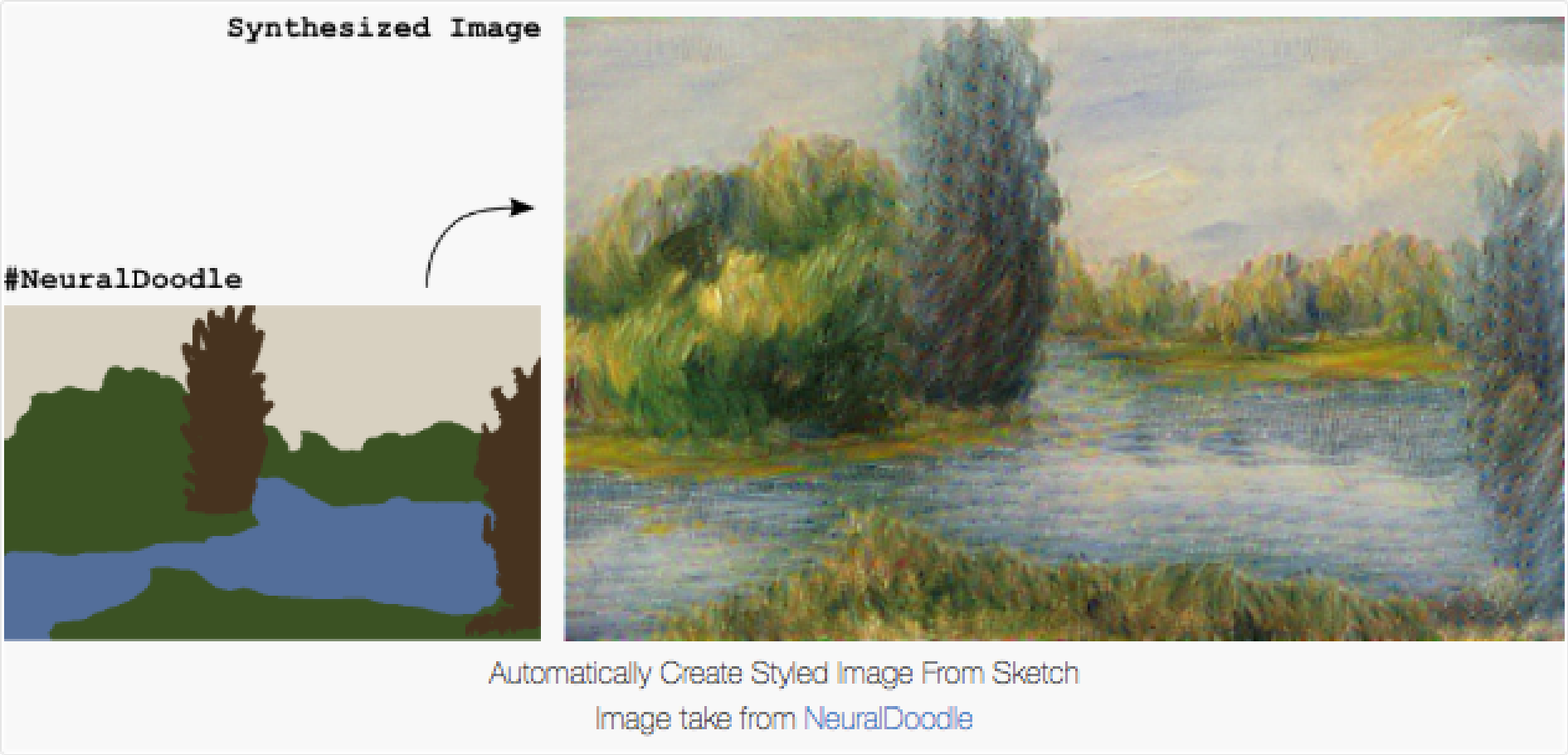
Sample of Automatic Handwriting Generation

Aplicações

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Automatic Text Generation Example of Shakespeare
Example taken from Andrej Karpathy blog post

Aplicações

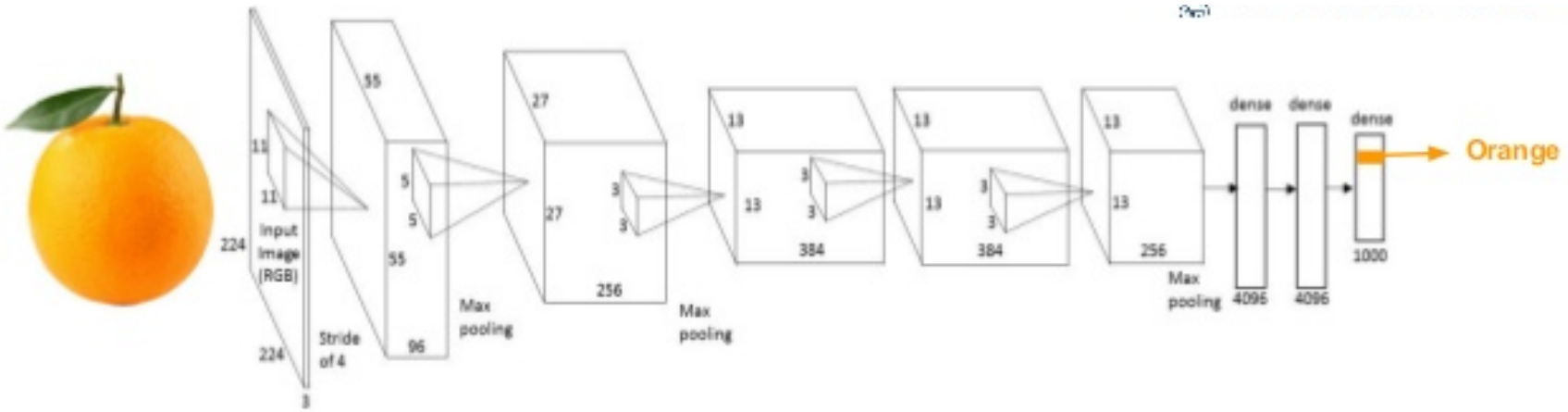


Automatically Create Styled Image From Sketch

Image take from NeuralDoodle

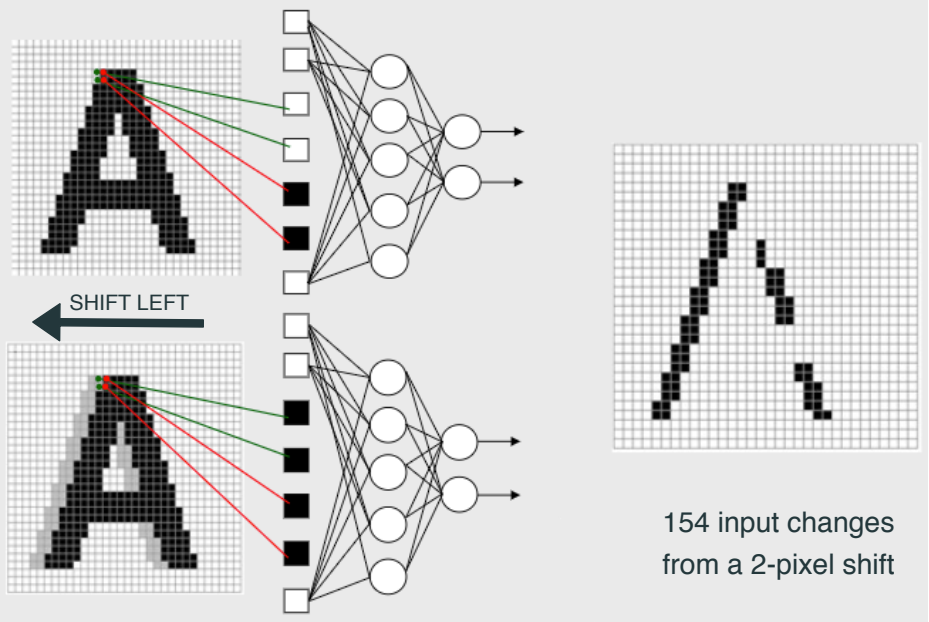
CNN

Rede neural cujas principais operações são convoluções, geralmente profunda, aplicada a dados quem possuem correlação local, como imagens.



Desvantagens de MLP neste domínio:

- MLPs, por serem totalmente conectados, não escalam bem para imagens de alta resolucão.
- MLPs possuem pouca invariância a distorções:

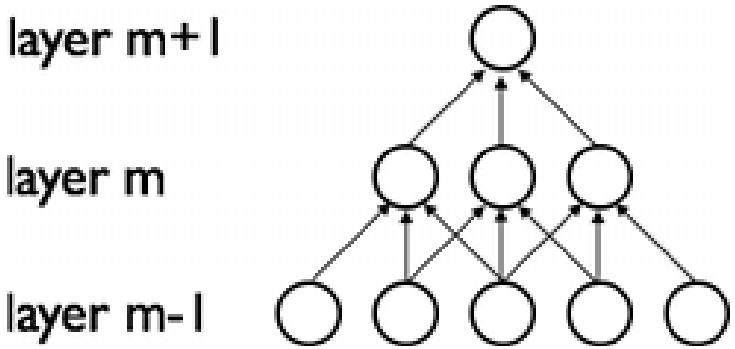


Desvantagens de MLP neste domínio:

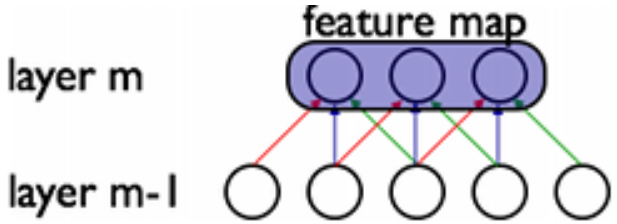
- Não leva em conta a estrutura dos dados: trata pixels que estão longe da mesma foram que pixels que estão perto na imagem.

Vantagens da CNN:

- Conectividade esparsa e local

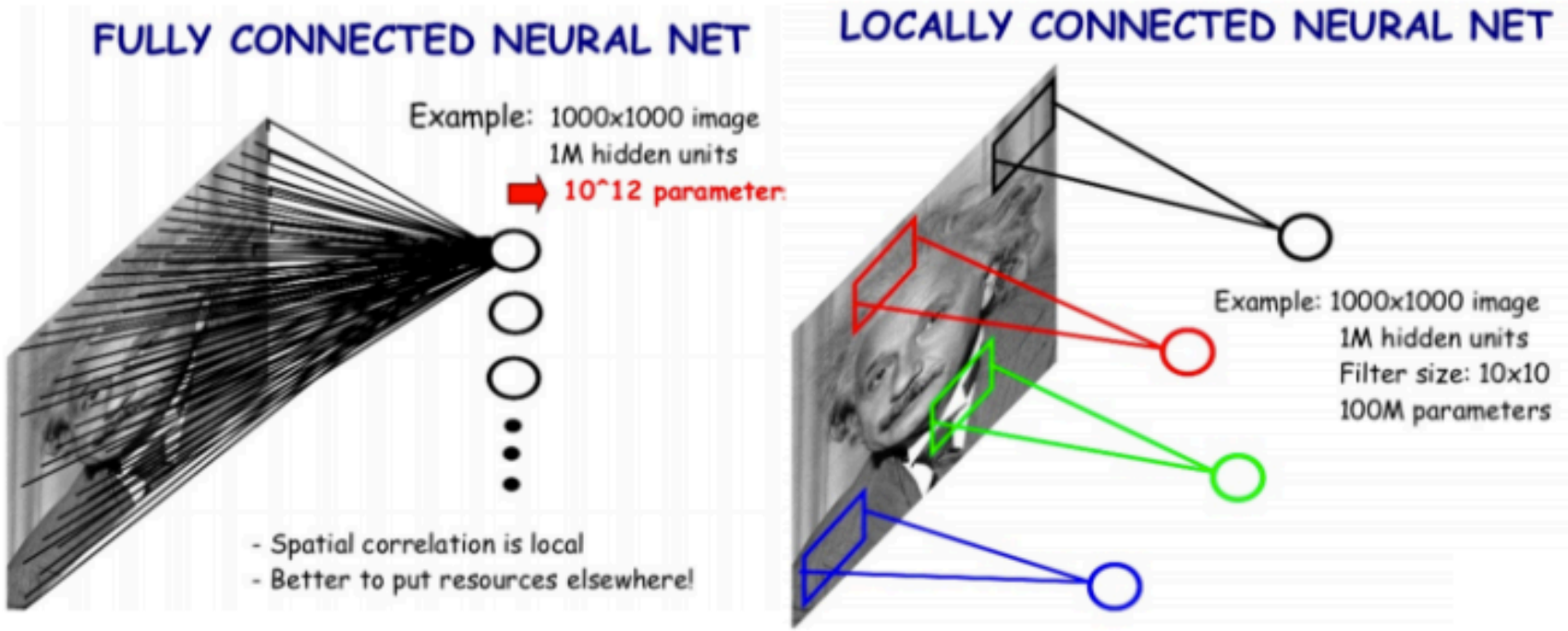


- Pesos compartilhados

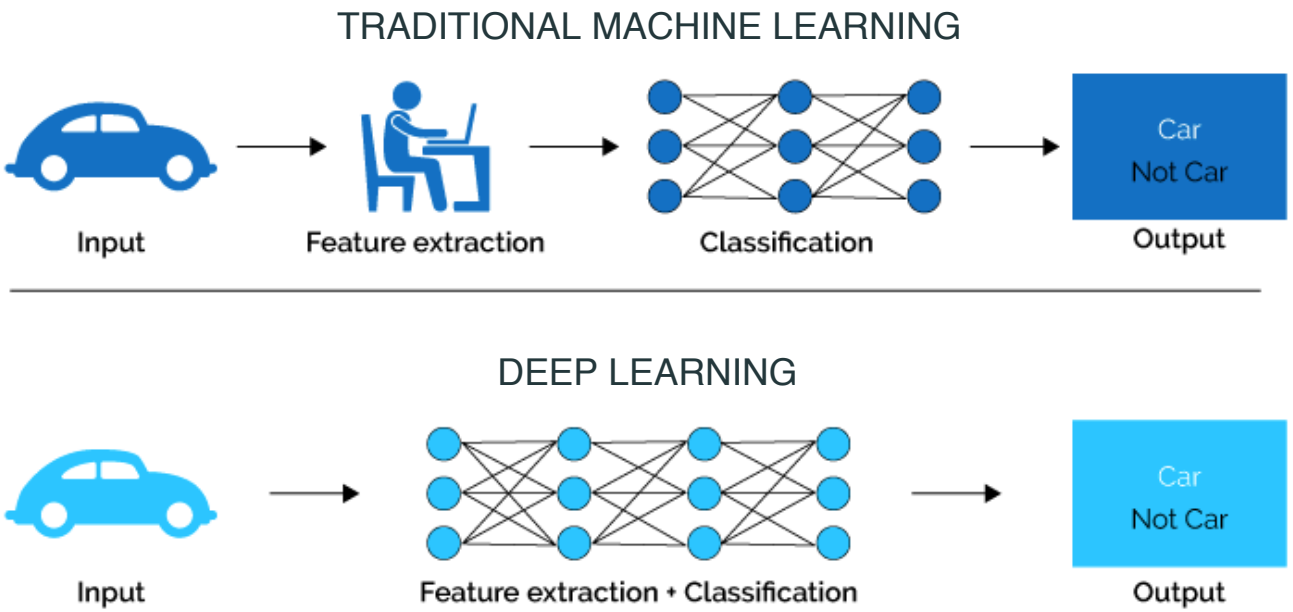


Vantagens da CNN:

- Conectividade local e pesos compartilhados



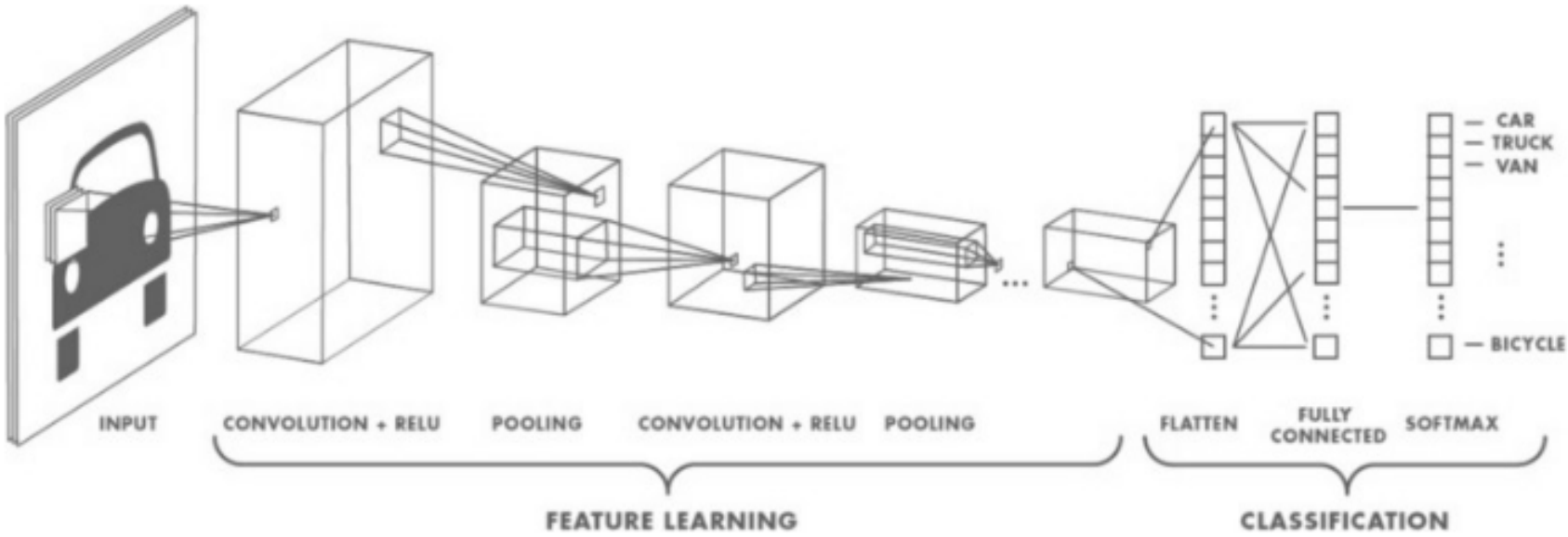
- Aprendizado ponta-a-ponta:



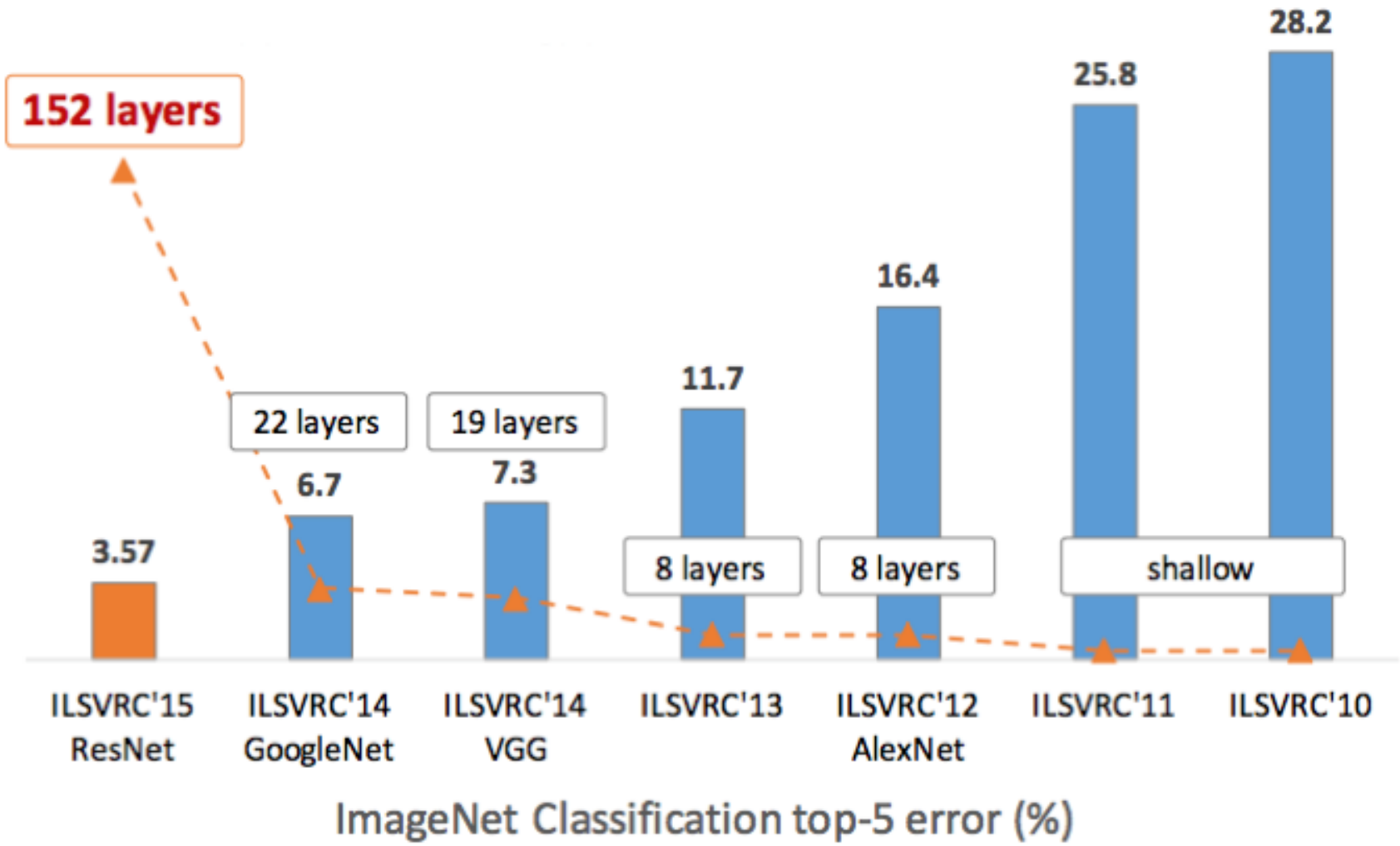
CNN

CNN típica possui três tipos de camadas principais:

- Convolucional seguida de uma função de ativação
- Pooling
- Fully-connected



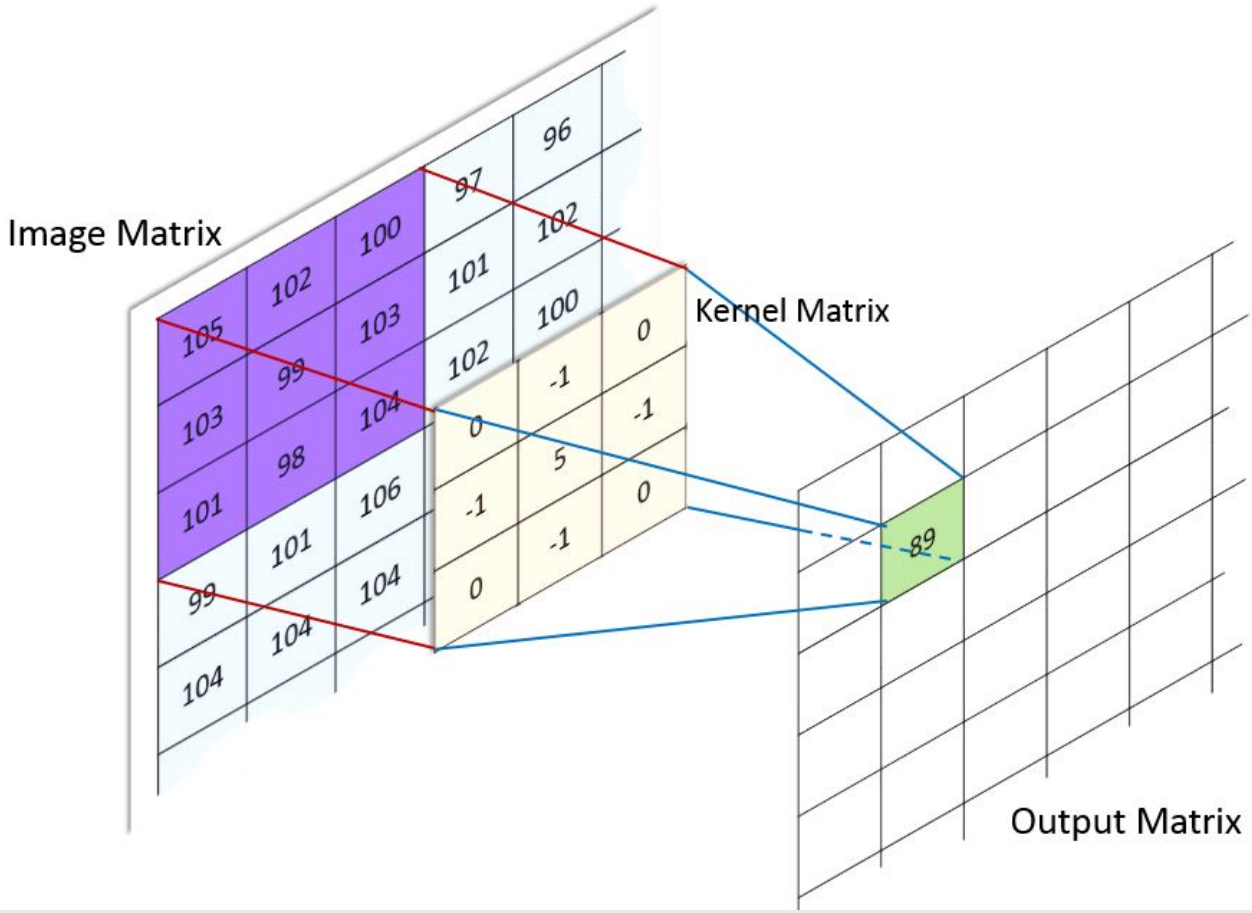
Revolução da profundidade



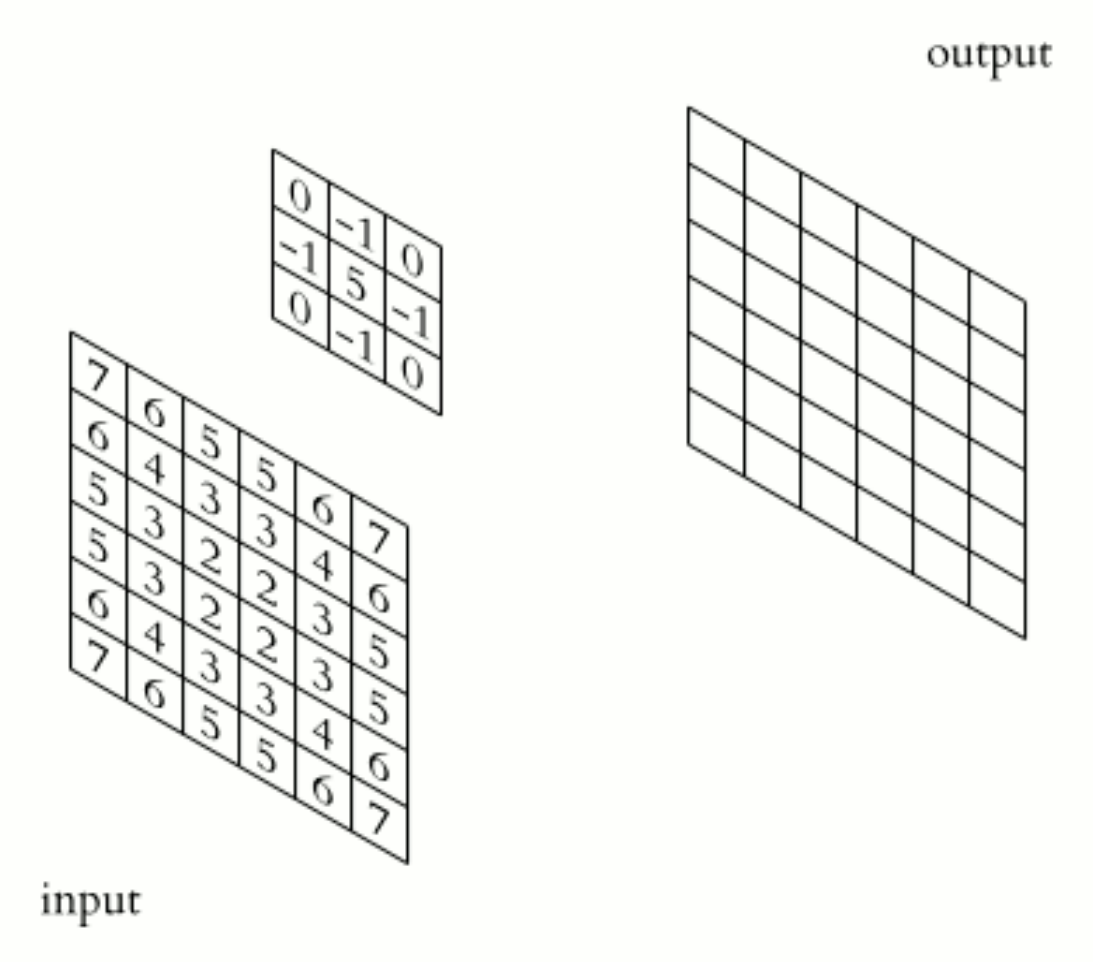
Camada convolucional

Convolução

Operador linear que aplica um filtro, definido por um *kernel*, a uma imagem, gerando uma outra imagem filtrada.



Convolução


output
0 -1 0
-1 5 -1
0 -1 0
7 6 5 5 6 7
6 4 3 3 4 6
5 3 2 2 3 5
5 3 2 2 3 5
6 4 3 3 4 6
7 6 5 5 6 7
input

Convolução – Exemplos

$*$

-1	-1	-1
-1	8	-1
-1	-1	-1

$=$

Convolução – Exemplos

$*$

-1	-1	-1
-1	8	-1
-1	-1	-1

$=$

Convolução – Exemplos

$*$

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

$=$

Convolução – Exemplos

$*$

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

$=$

Convolução – Exemplos

$*$

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

$=$

Convolução – Exemplos

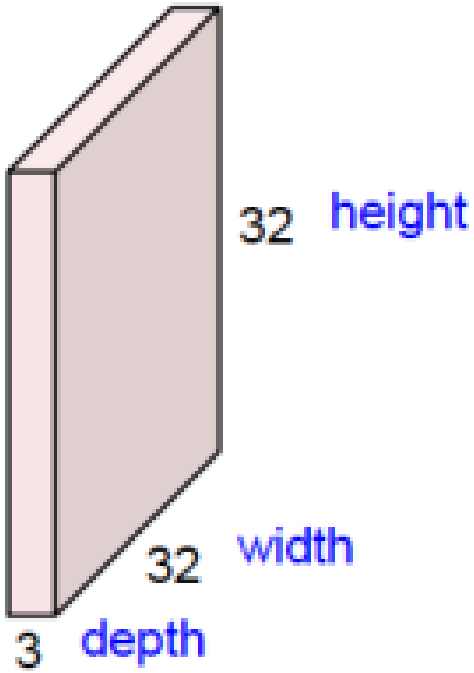
$*$

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

$=$

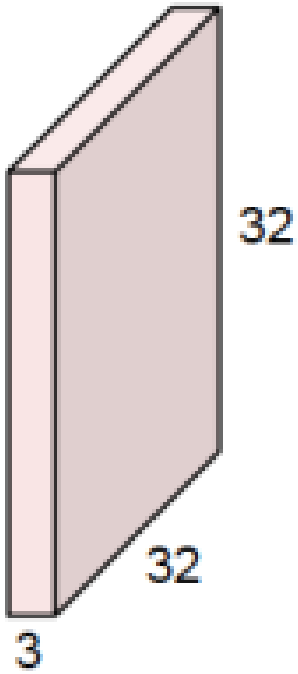
Camada convolucional

imagem 32x32x3

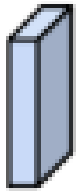


Camada convolucional

imagem 32x32x3



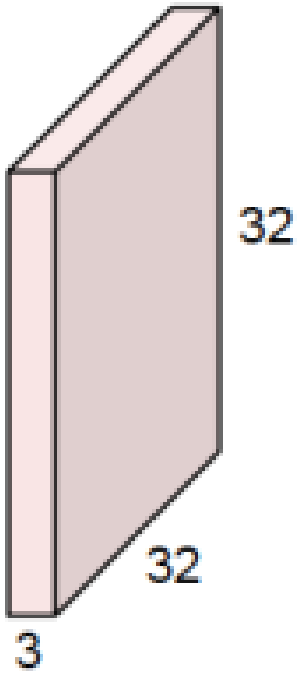
filtro 5x5x3



Convolucione o filtro com a imagem, i.e., desloque o filtro sobre a imagem, computando produtos internos.

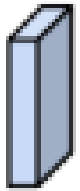
Camada convolucional

imagem 32x32x**3**



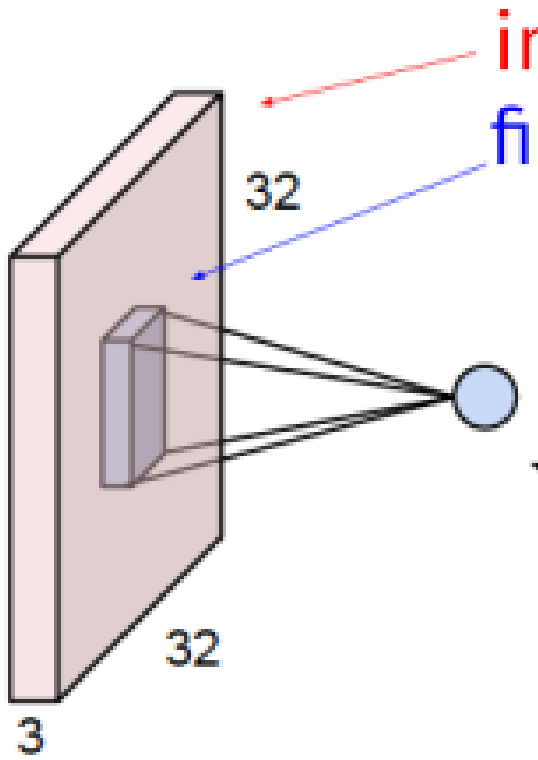
A profundidade do filtro sempre é a mesma da imagem.

filtro 5x5x**3**



Convolucione o filtro com a imagem, i.e., desloque o filtro sobre a imagem, computando produtos internos.

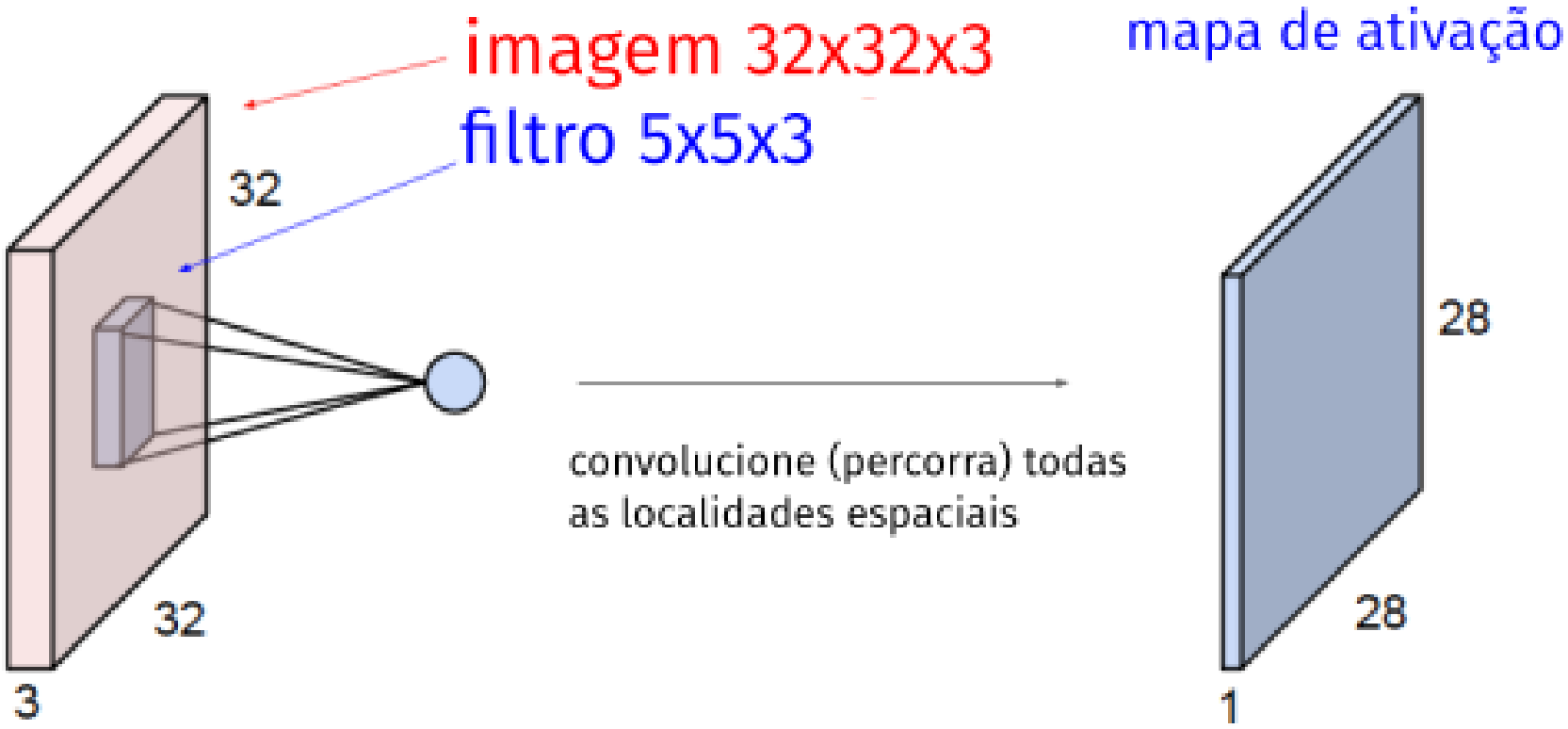
Camada convolucional



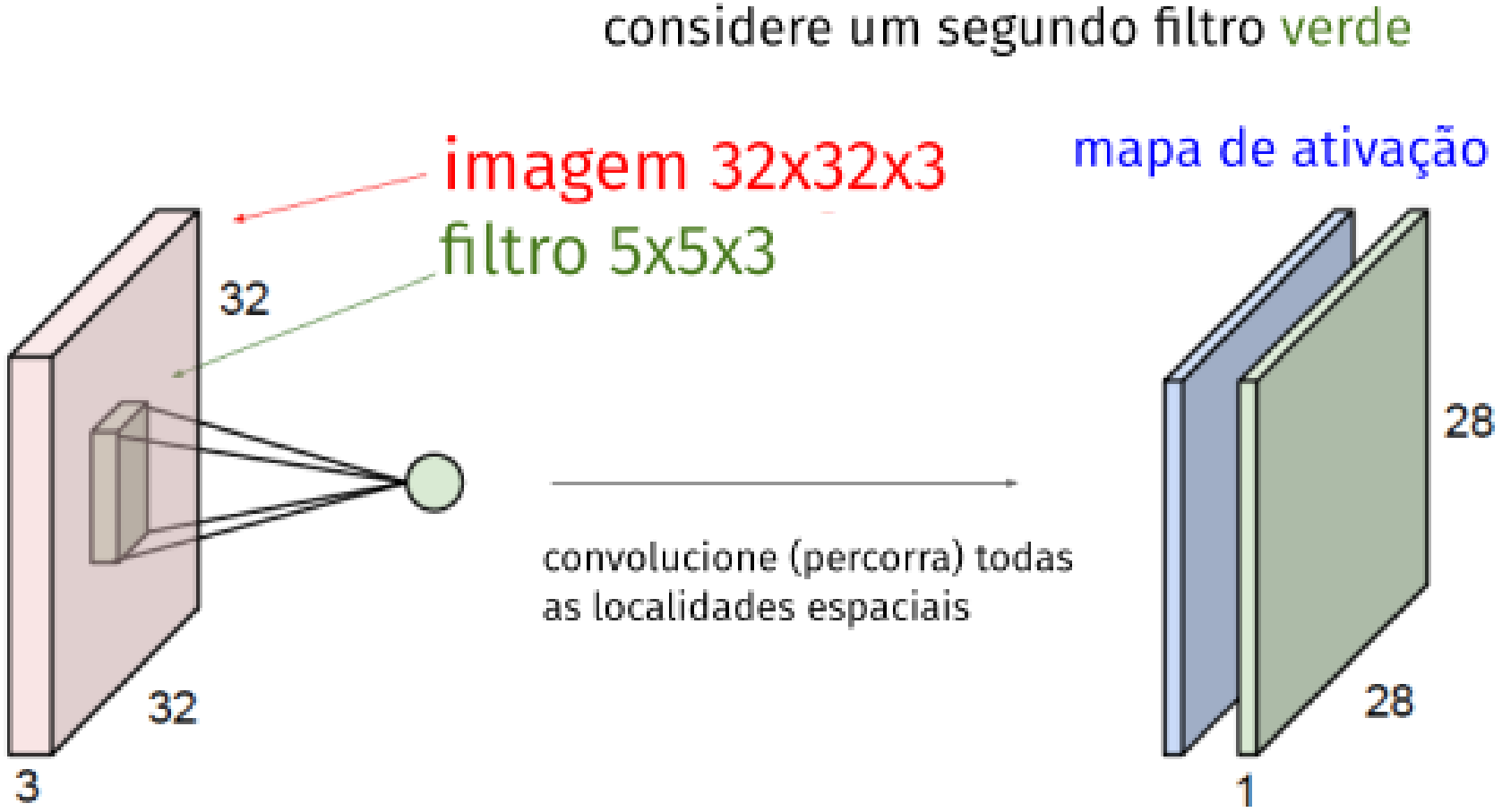
Um número:
Resultado do produto interno entre
o filtro e um pedaço 5x5x3 da imagem.
I.e., 5*5*3 = produto interno 75-d + bias.

$$w^T x + b$$

Camada convolucional

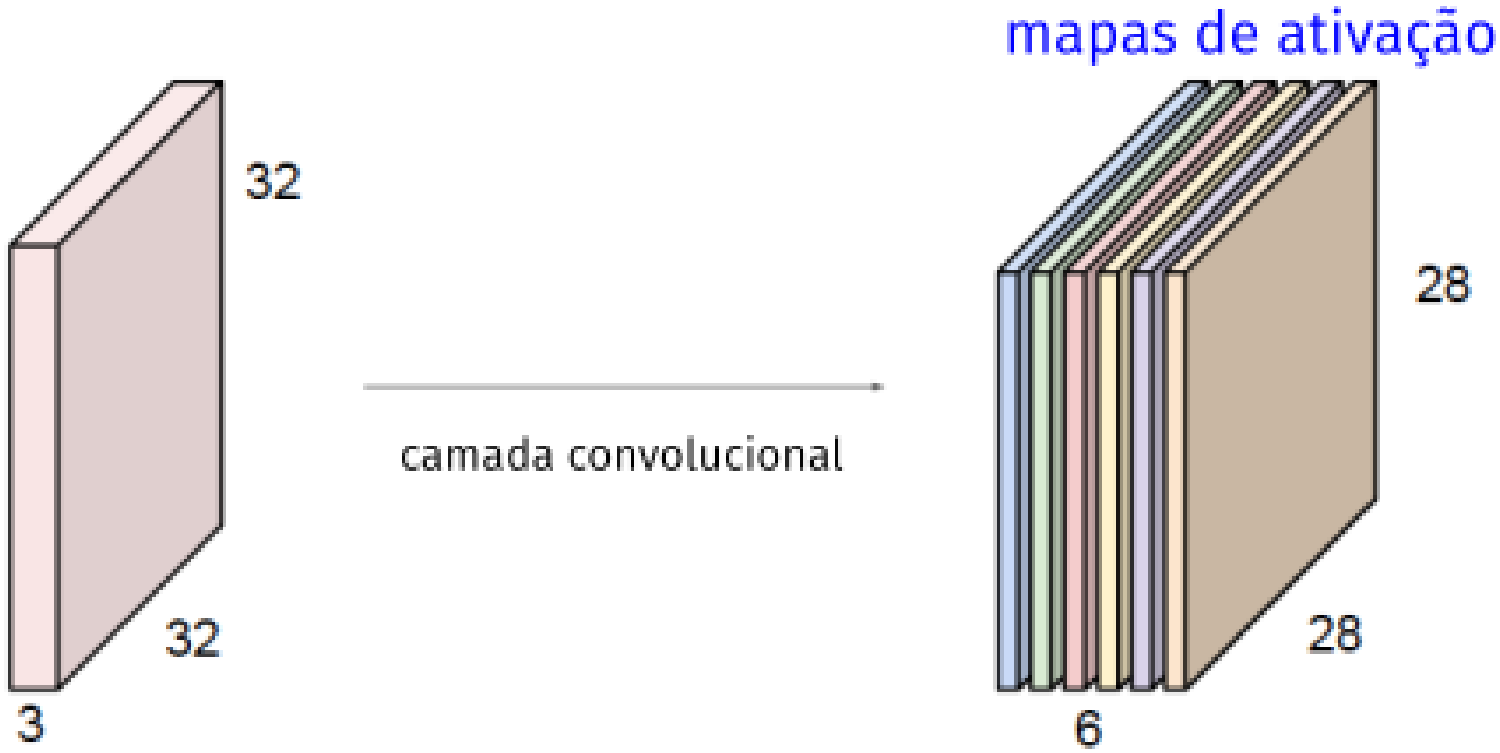


Camada convolucional



Camada convolucional

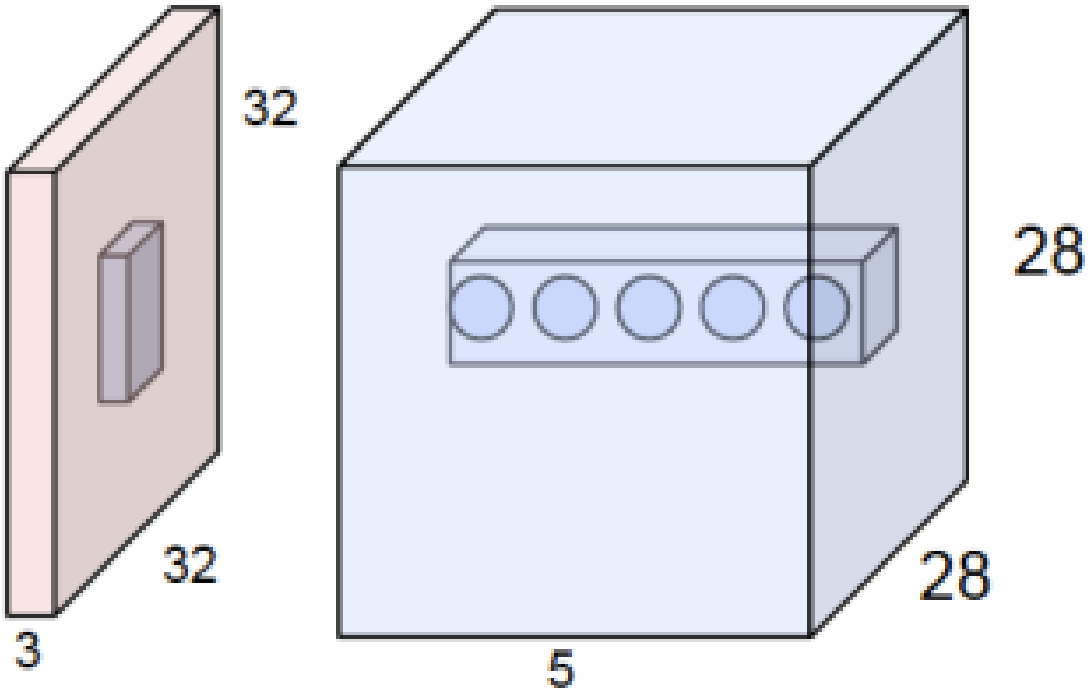
6 filtros 5x5 geram 6 mapas de ativação:



Os mapas de ativação formam um novo volume de tamanho 28x28x6

Camada convolucional

Exemplo de cinco neurônios (filtros) na formação do volume de saída

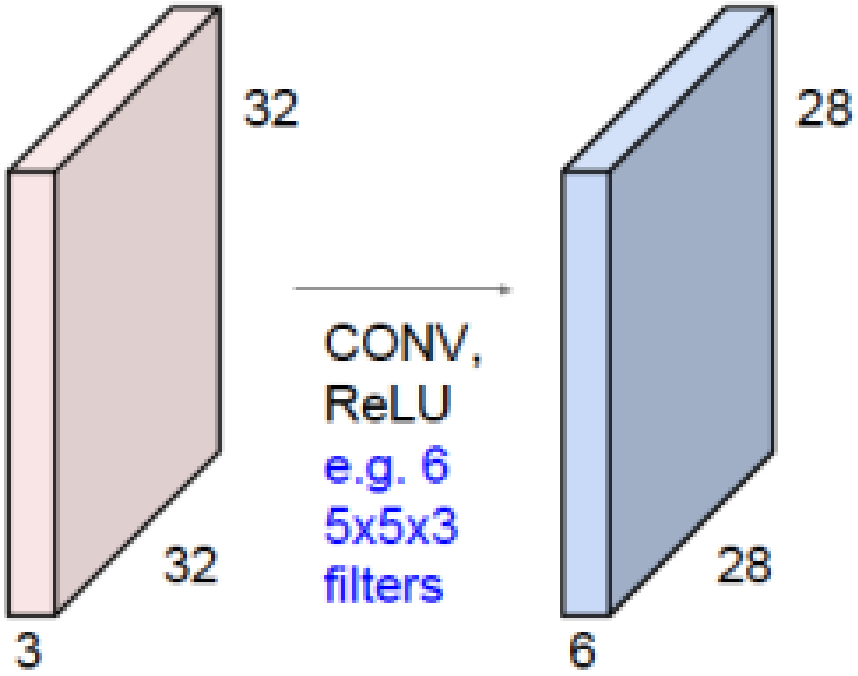


Ex.: a camada convolucional consiste de neurônios (filtros) organizados em um grid 3D de tamanho 28x28x5.

5 neurônios diferentes estão "olhando" a mesma região do volume de input.

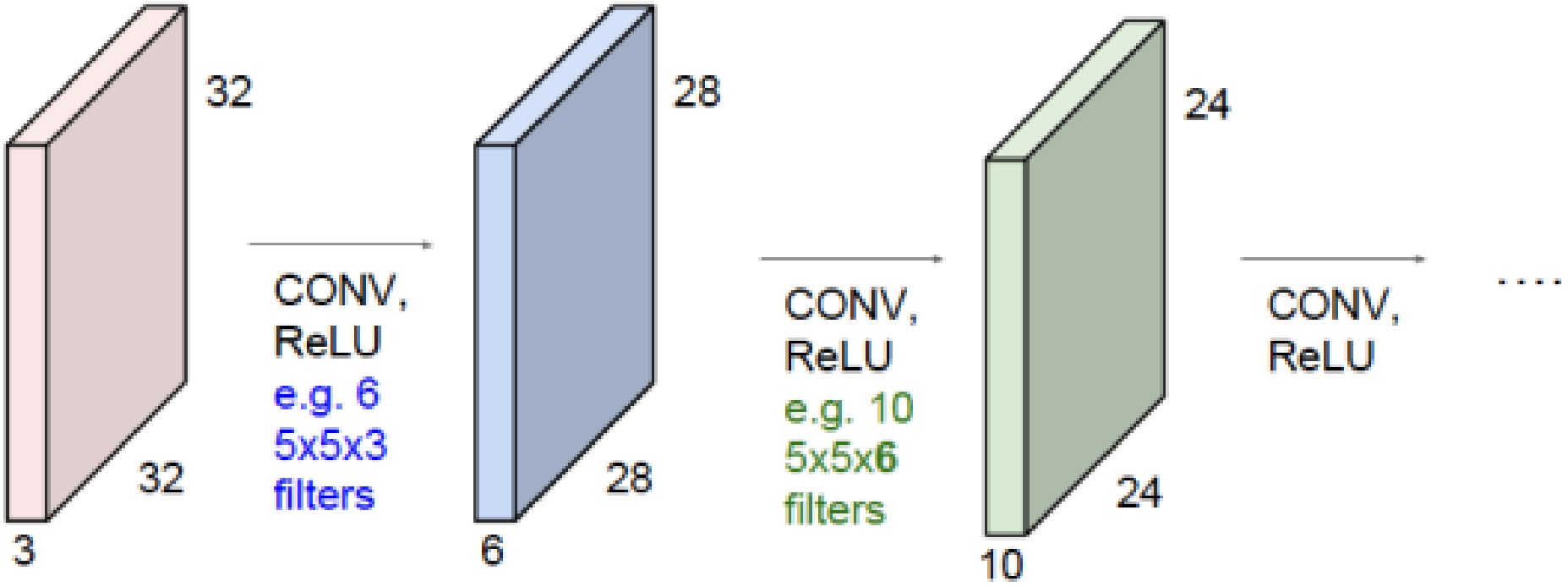
Camada convolucional

CNN é uma sequência de camadas convolucionais, intercaladas por funções de ativação.

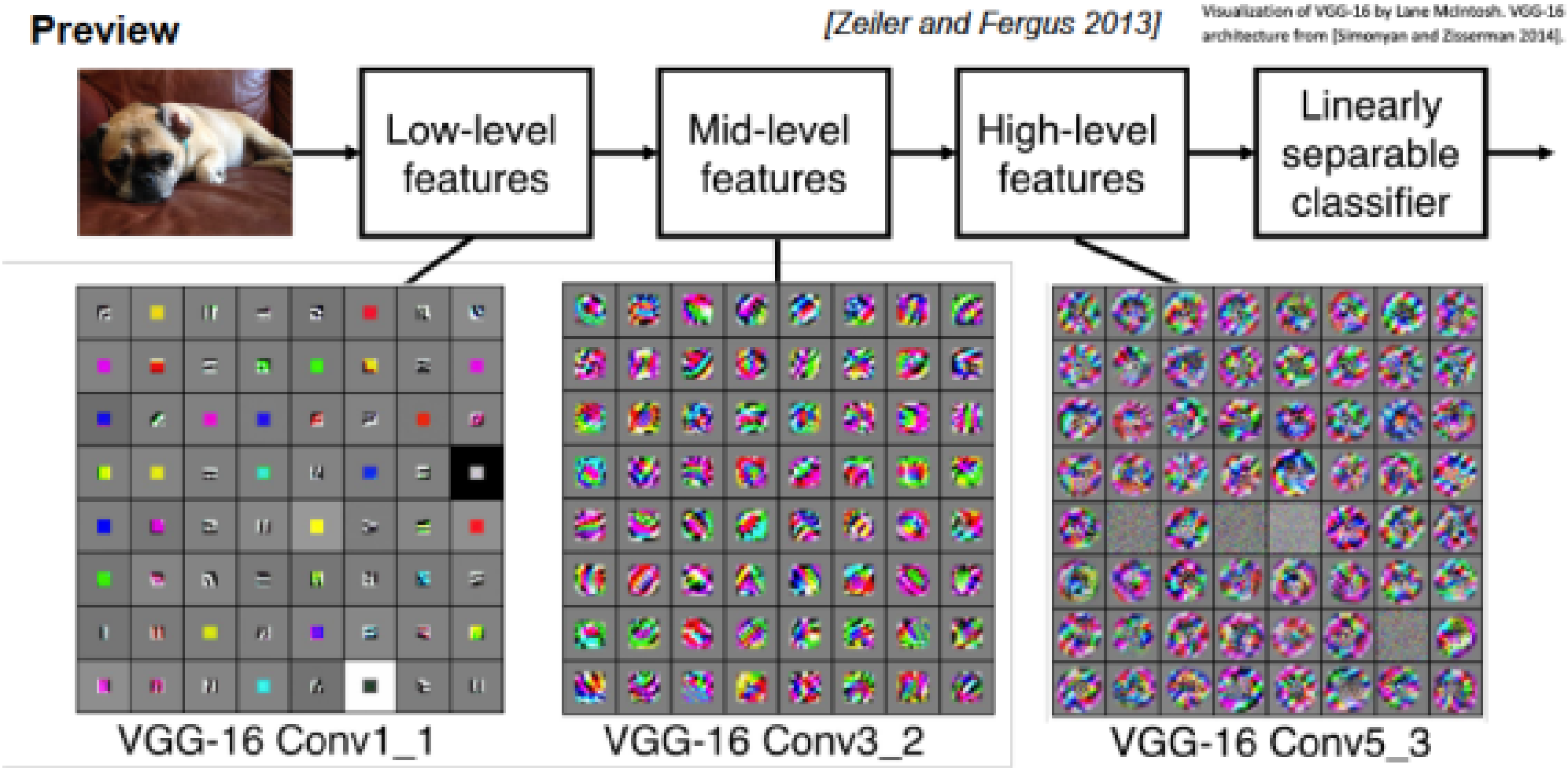


Camada convolucional

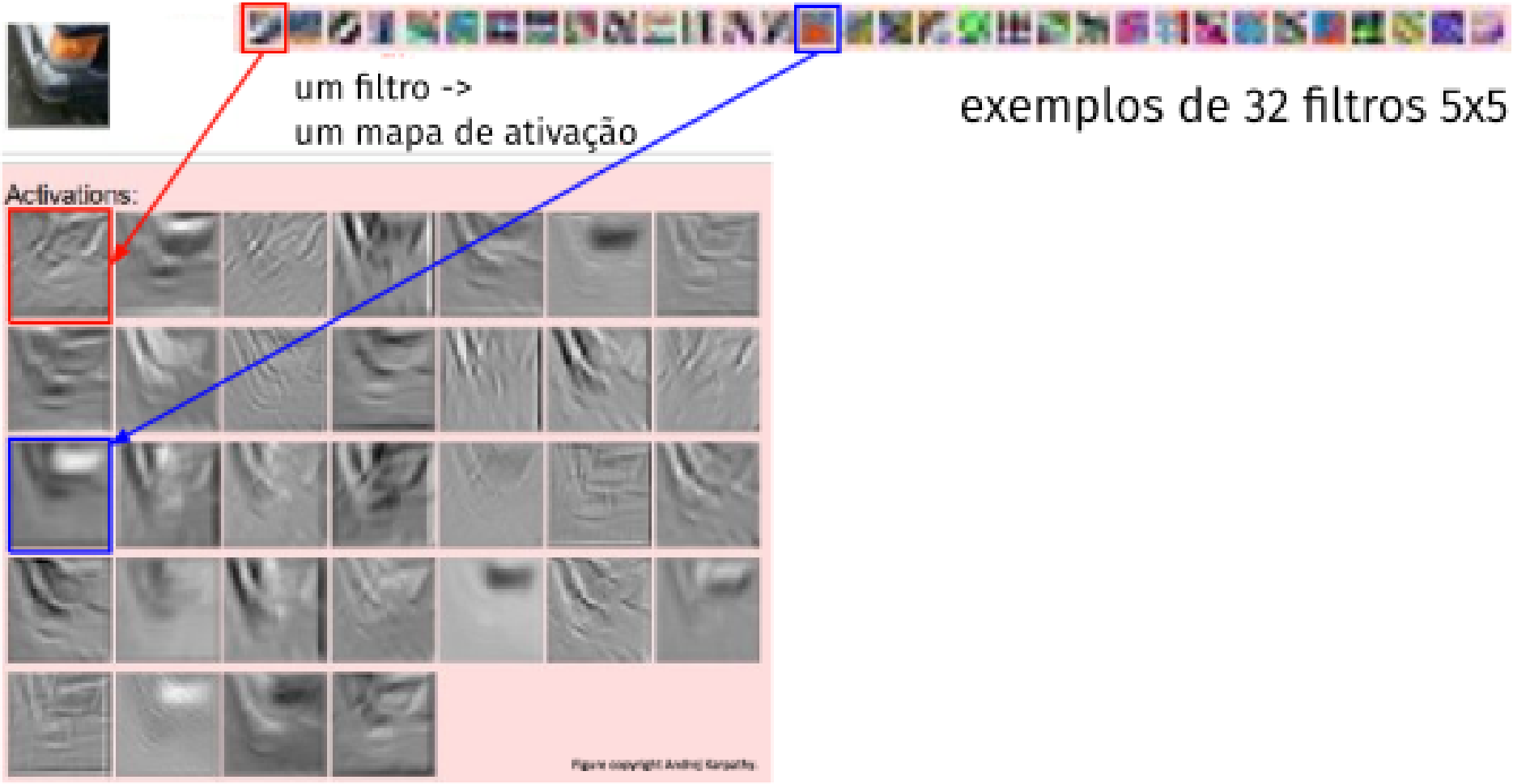
CNN é uma sequência de camadas convolucionais, intercaladas por funções de ativação.



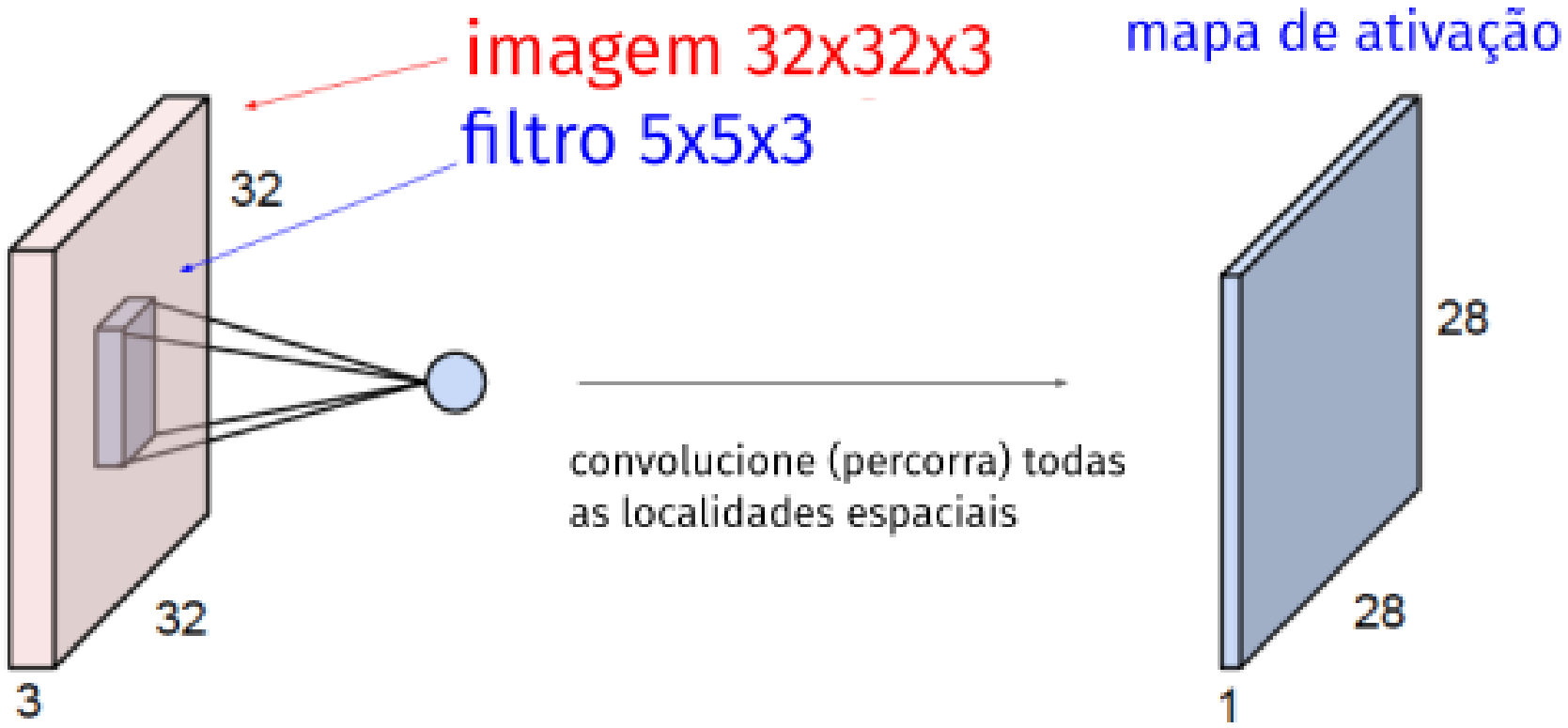
Camada convolucional



Camada convolucional

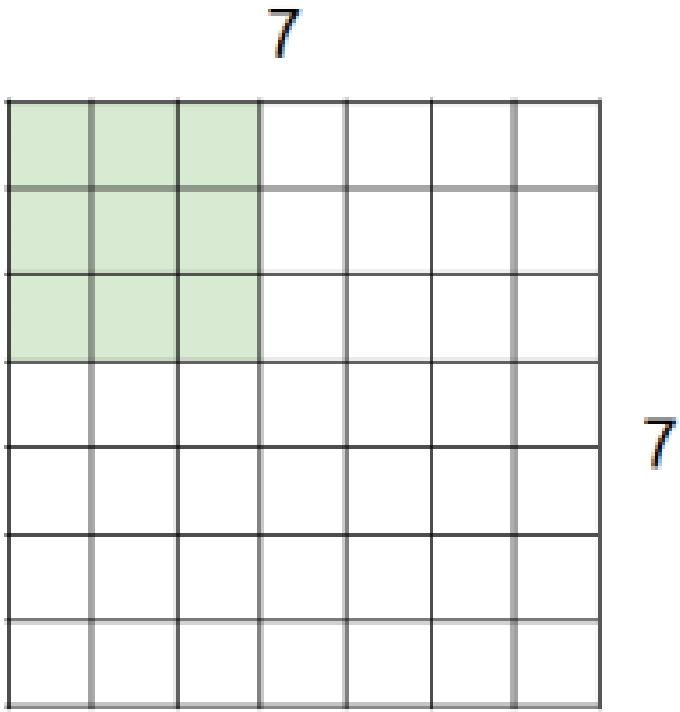


Analisando as dimensões espaciais:



Camada convolucional

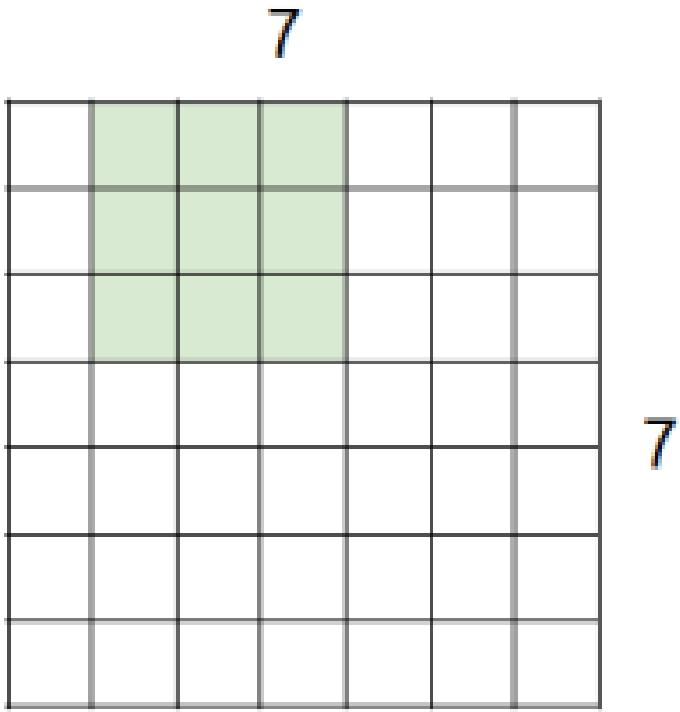
Analisando as dimensões espaciais:



input 7x7
filtro 3x3

Camada convolucional

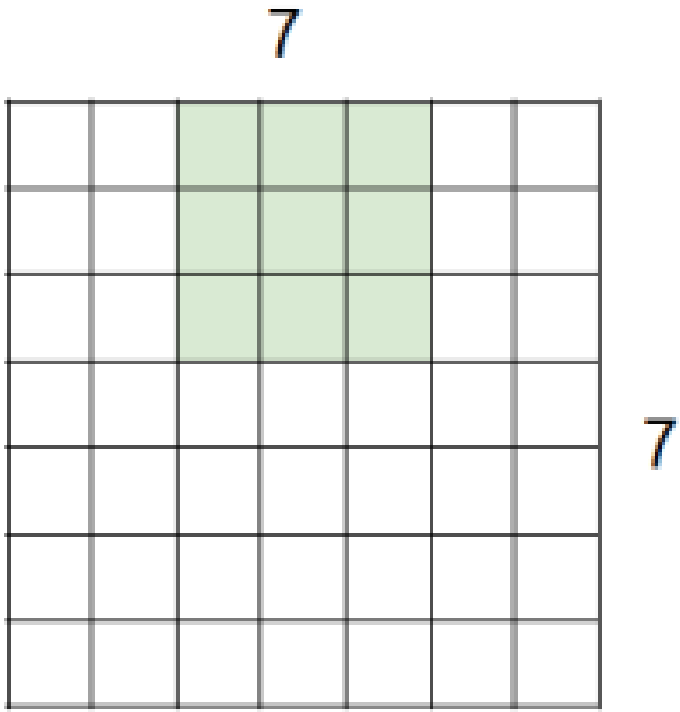
Analisando as dimensões espaciais:



input 7x7
filtro 3x3

Camada convolucional

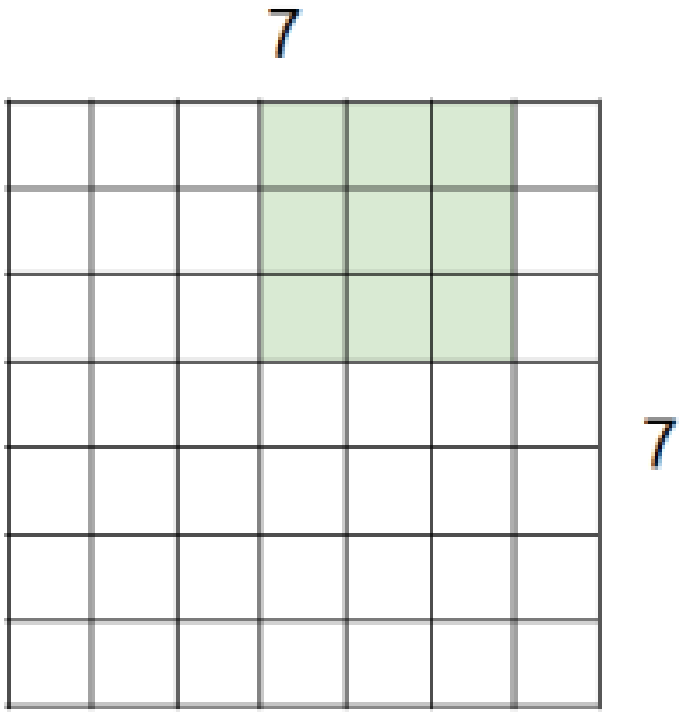
Analisando as dimensões espaciais:



input 7x7
filtro 3x3

Camada convolucional

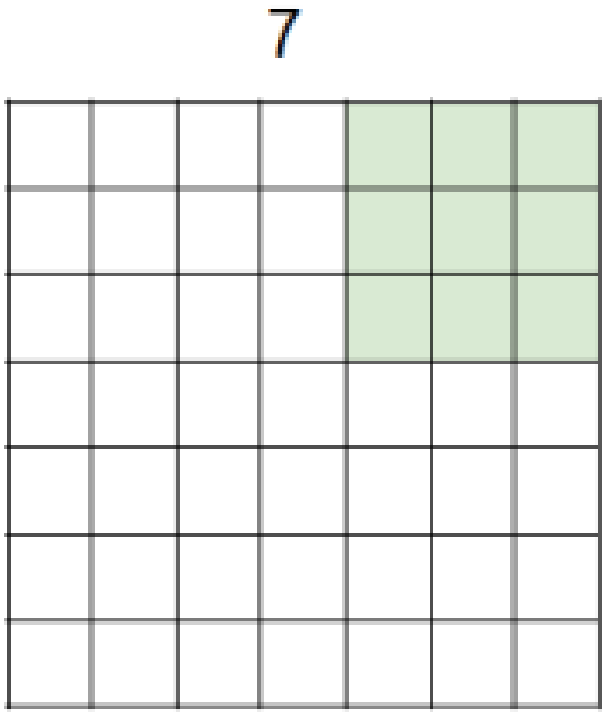
Analisando as dimensões espaciais:



input 7x7
filtro 3x3

Camada convolucional

Analisando as dimensões espaciais:

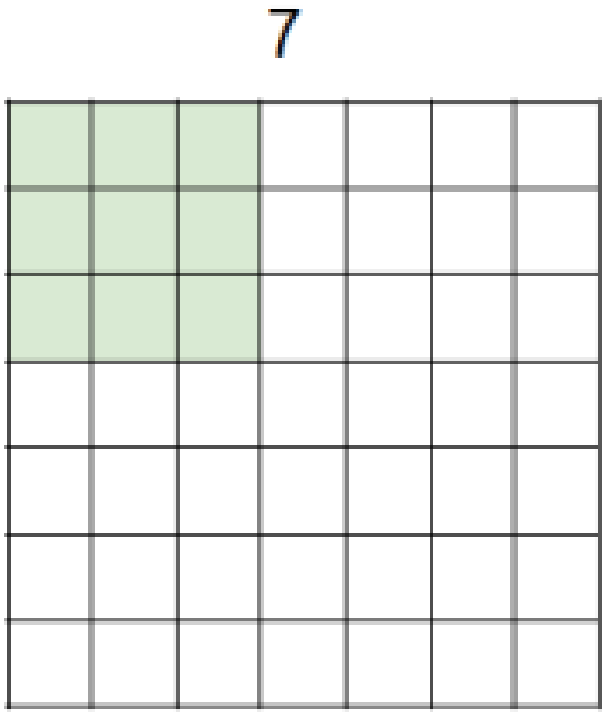


input 7x7
filtro 3x3

=> output 5x5

Camada convolucional

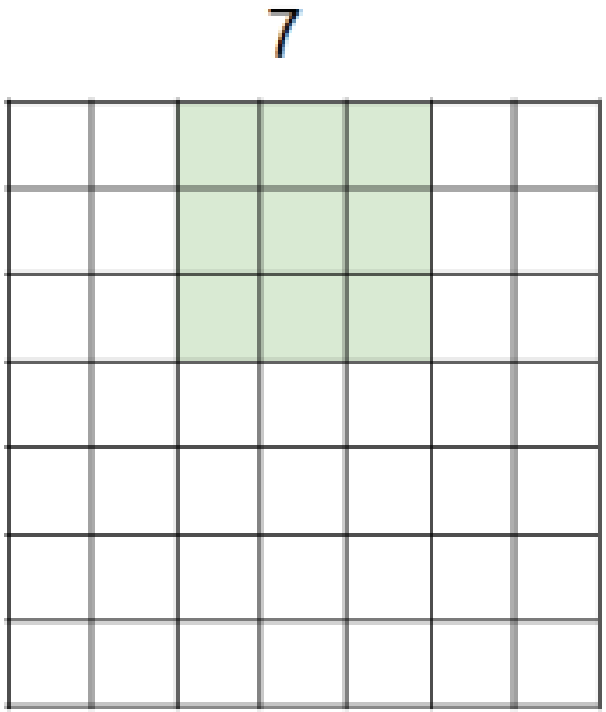
Analisando as dimensões espaciais:



input 7x7
filtro 3x3
com **stride 2**

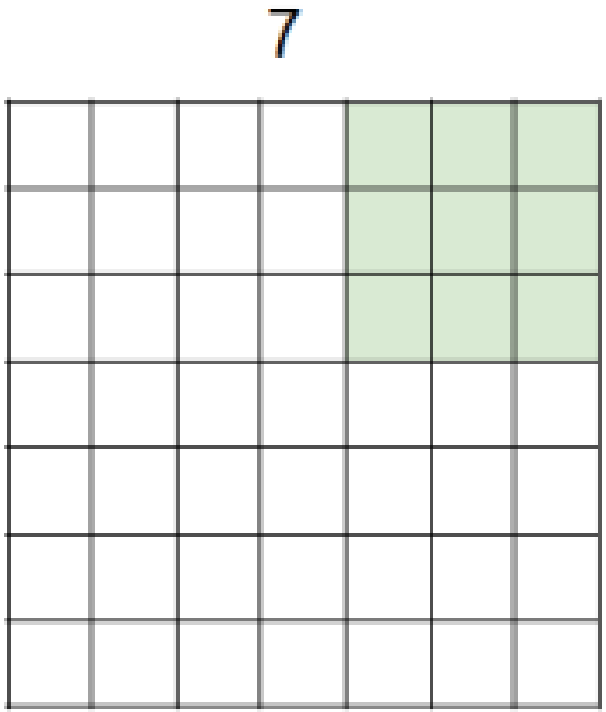
Camada convolucional

Analisando as dimensões espaciais:



input 7x7
filtro 3x3
com **stride 2**

Analisando as dimensões espaciais:

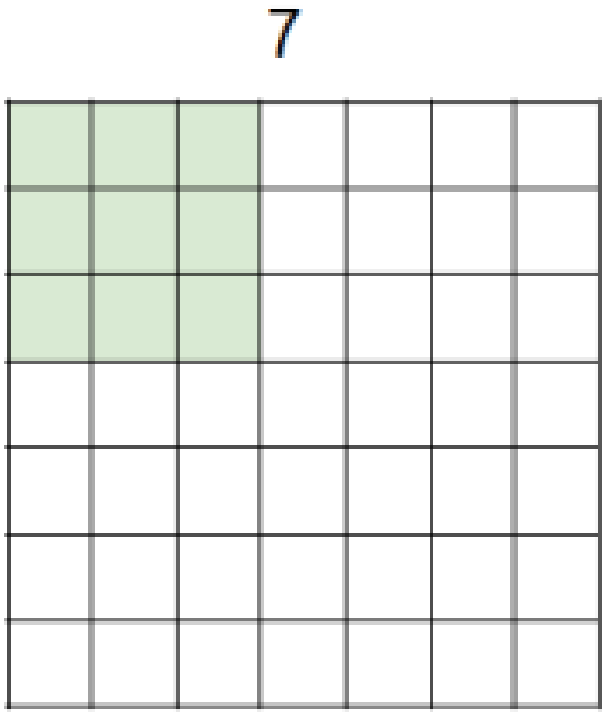


input 7x7
filtro 3x3
com **stride 2**

=> 3x3 output

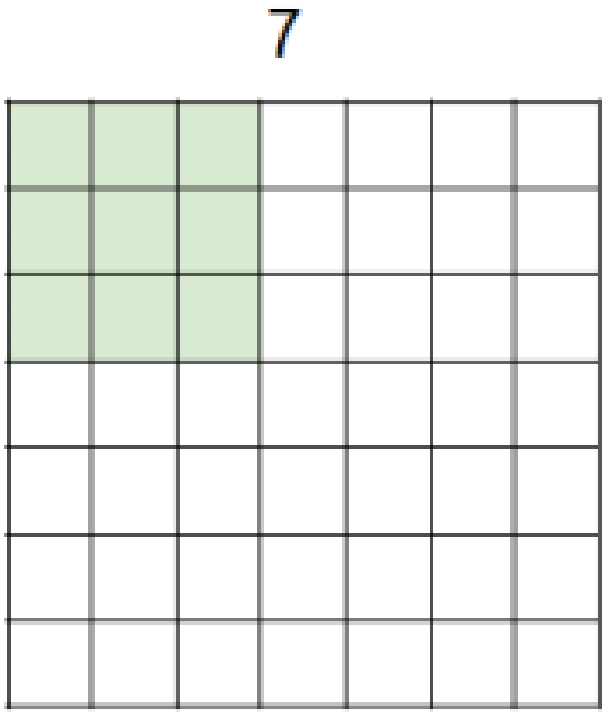
Camada convolucional

Analisando as dimensões espaciais:



input 7x7
filtro 3x3
com **stride 3**

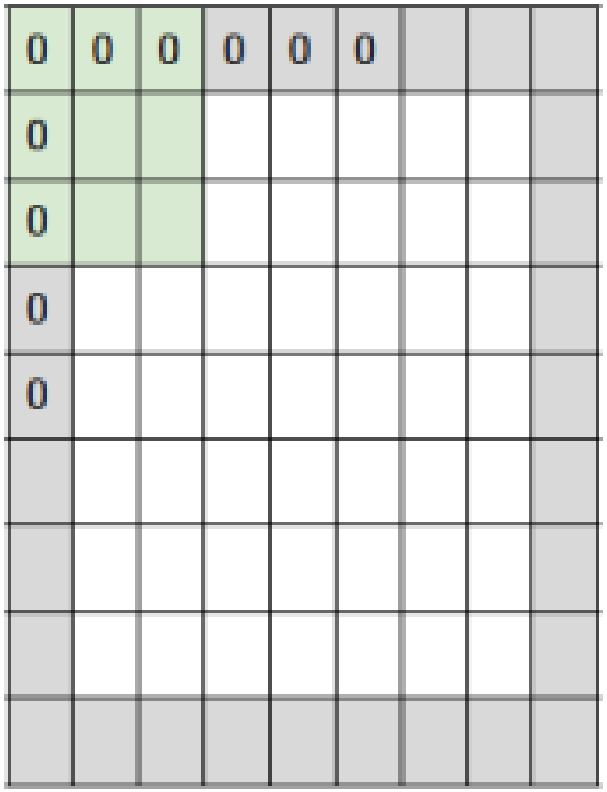
Analisando as dimensões espaciais:



input 7x7
filtro 3x3
com **stride 3**

não cabe!
não é possível aplicar um filtro 3x3
em um input 7x7 com stride 3.

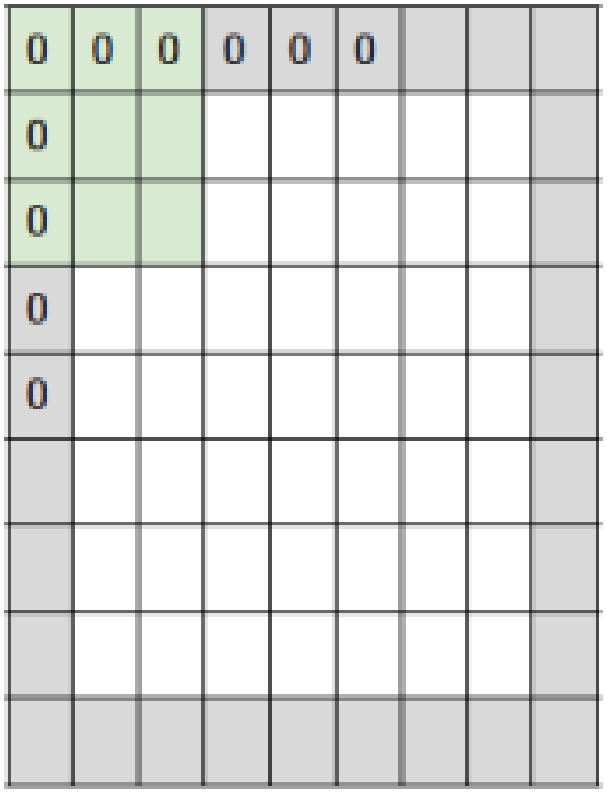
Na prática: é comum usar **zero pad**



input 7x7
filtro 3x3
com **stride 1** e **pad de 1 pixel**

=> qual é o tamanho do output?

Na prática: é comum usar **zero pad**

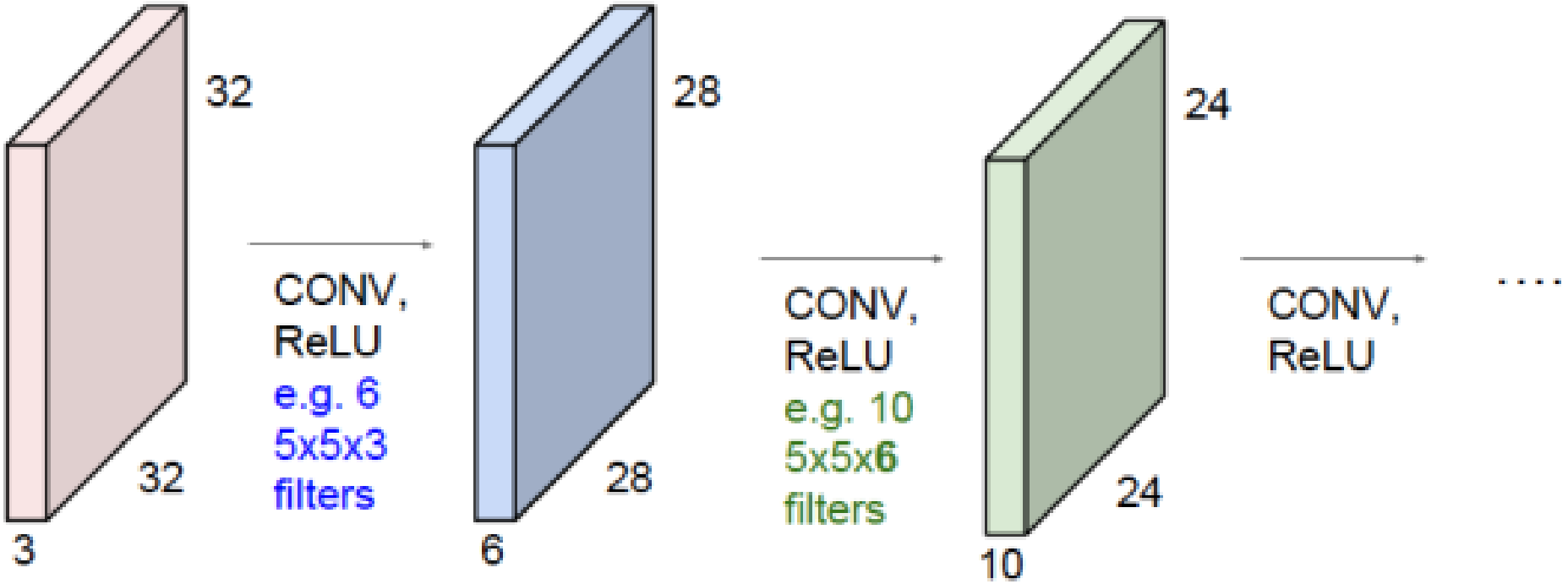


input 7x7
filtro 3x3
com **stride 1** e **pad de 1 pixel**

=> output 7x7

Camada convolucional

Exemplo: input 32x32 com repetidas convoluções com filtro 5x5.
Volume encolhe espacialmente (32->28->24 ...)

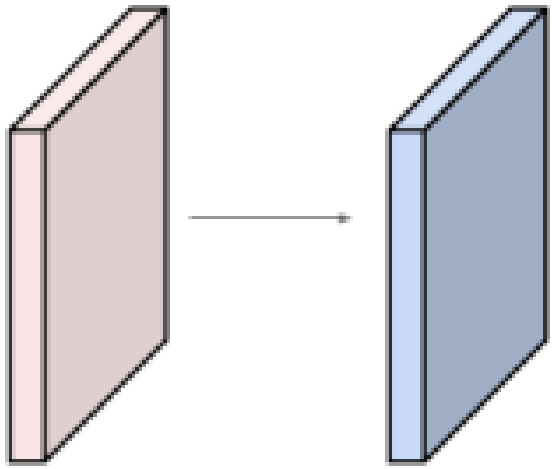


Camada convolucional

Parâmetros

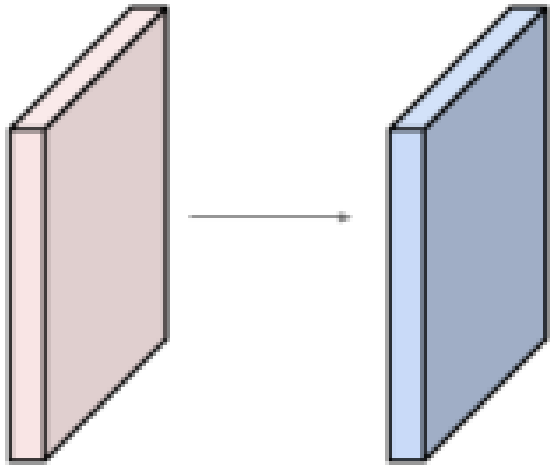
input 32x32x3
10 filtros 5x5

Qual a quantidade de parâmetros nesta camada?



Camada convolucional

Parâmetros



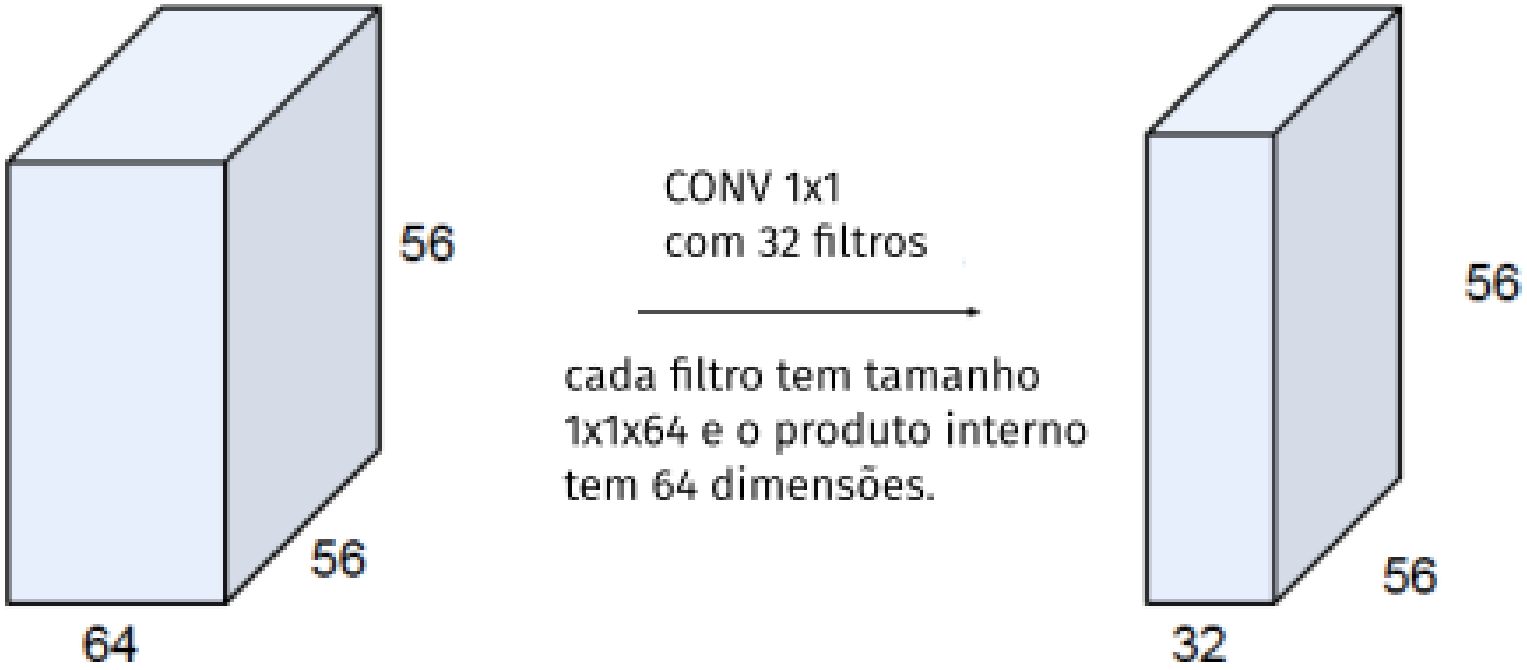
input 32x32x3
10 filtros 5x5

Qual a quantidade de parâmetros nesta camada?

Cada filtro tem 5*5*3 + 1 = **76 parâmetros** (+1 bias)

=> 76 * 10 = **760 parâmetros no total**

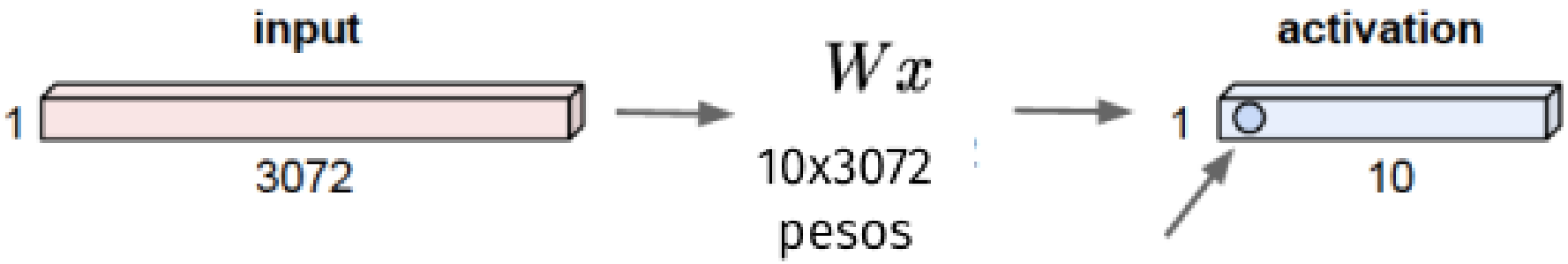
Camadas convolucionais com filtros 1x1 existem!



Camada fully-connected

Camada fully-connected

input 32x32x3 -> 3072x1

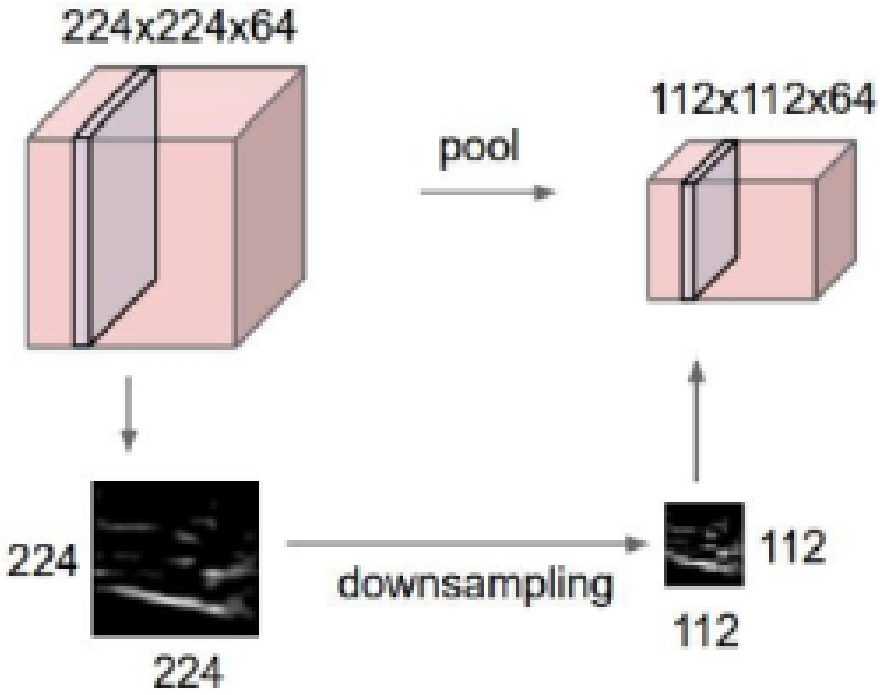


Um número:
Resultado do produto interno entre uma linha de W e o input.

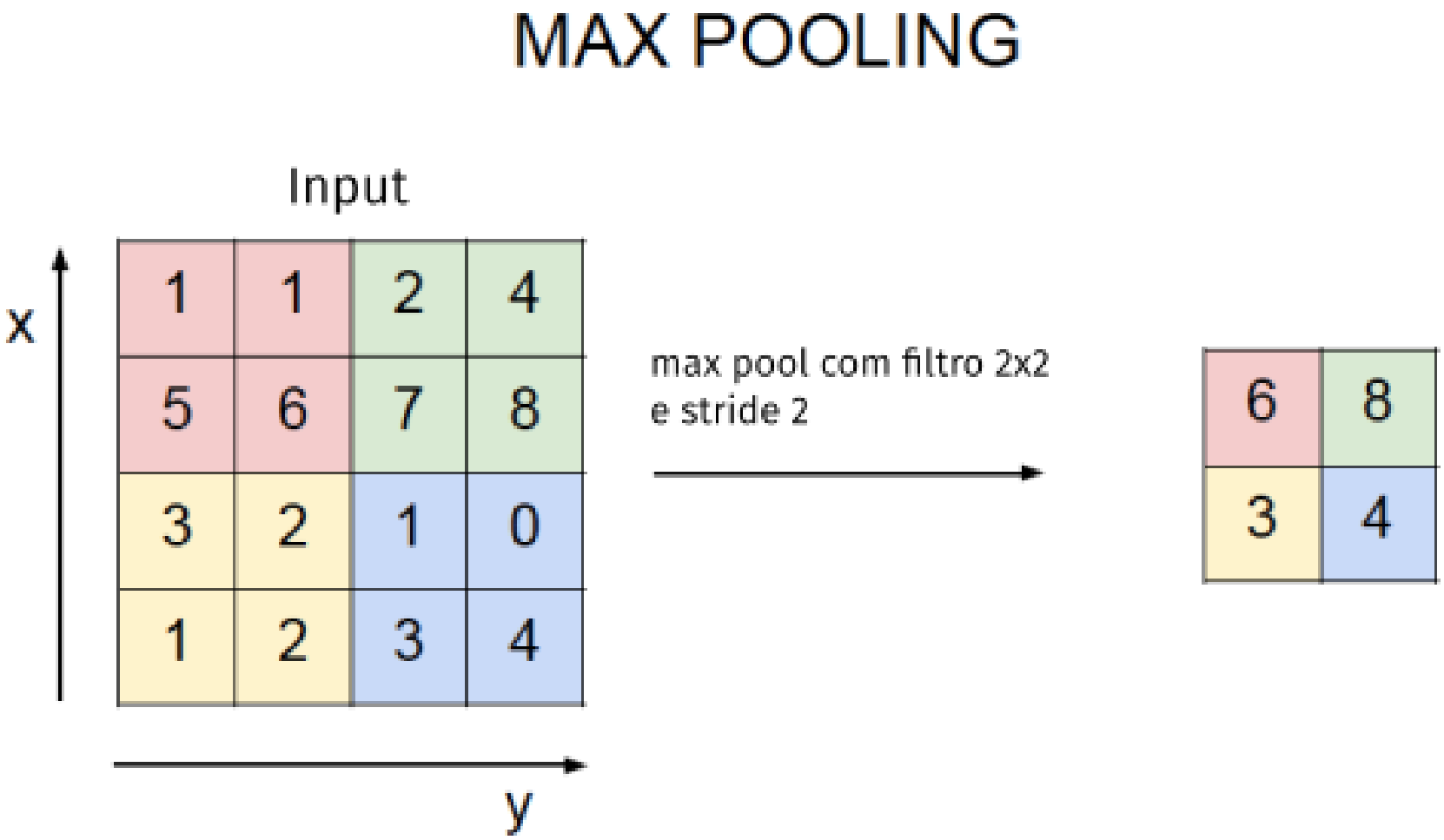
Camada de pooling

Camada de pooling

Opera sobre cada mapa de ativação independentemente, gerando mapas menores

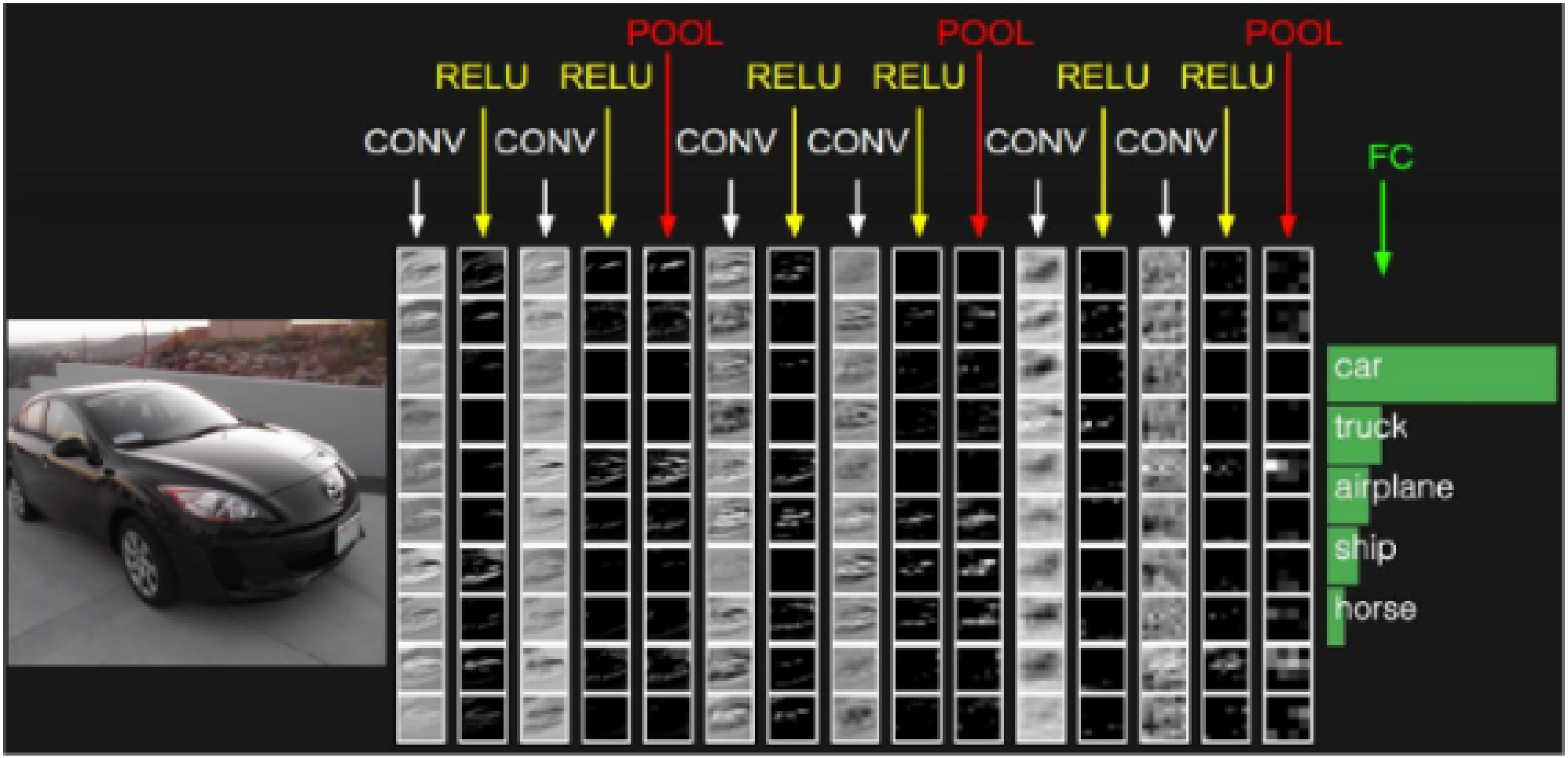


Camada de pooling



Exemplo CNN

Exemplo CNN

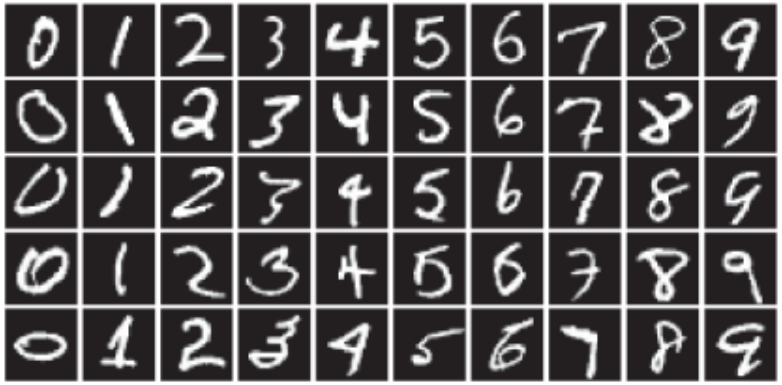

POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

Prática

Prática

Exemplos – dataset MNIST

Jupyter notebook:
INF0618_Aula02_Exemplos_CNN_MNIST.ipynb



- 70.000 dígitos escritos a mão
- *Labels* de 0 a 9
- Cada imagem tem 28×28 pixels (matriz de 28 por 28 com valores de 0 a 255).
- Problema: treinar uma ANN para classificar uma imagem em um dígito de 0 a 9.