

REDES CONVOLUCIONAIS

Alcione de Paiva Oliveira - DPI/UFV

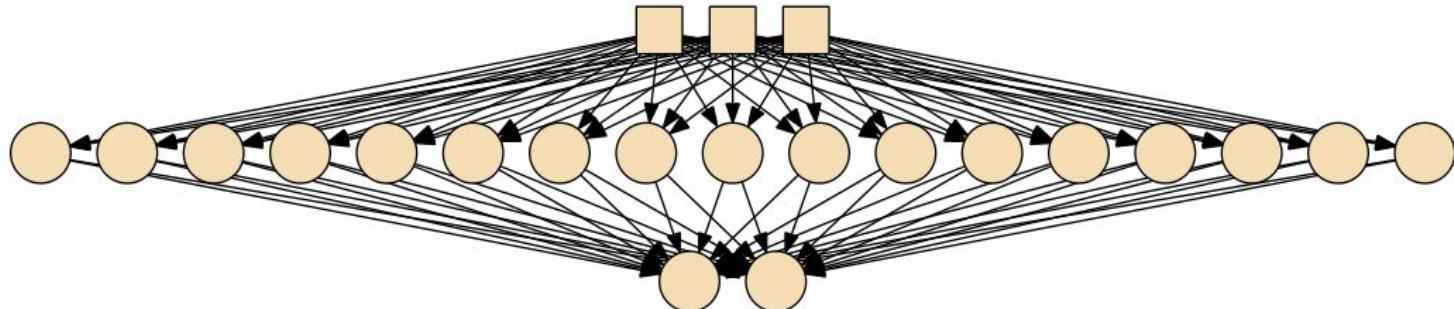


REDES CONVOLUCIONAIS

Introdução

Por que Aprendizagem profunda?

- Teorema da Aproximação Universal (1991): "Com uma única camada oculta, é possível aproximar qualquer função contínua limitada, contanto que se possa definir a quantidade suficiente de neurônios"



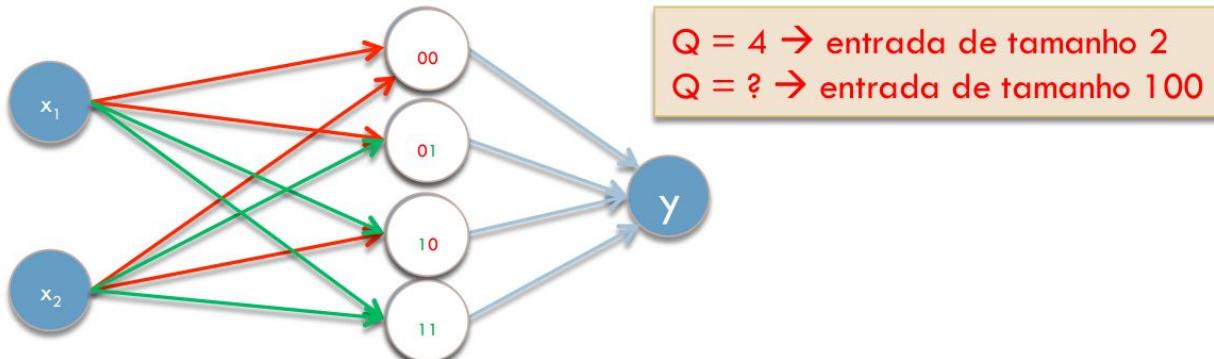


REDES CONVOLUCIONAIS

Introdução

Por que Aprendizagem profunda?

- Teoria **não** dá garantias sobre o limite superior da quantidade Q de neurônios na camada oculta.
- Dependendo do problema, Q pode crescer de forma **exponencial** com o tamanho da entrada.
- O problema da maldição da dimensionalidade.
- Se uma é rede completamente conectada com uma camada oculta para uma imagem de 28×28 (MNIST) que tenha o dobro de unidades da camada de entrada, então teríamos da ordem de 10^6 parâmetros para ajustar durante o treinamento!

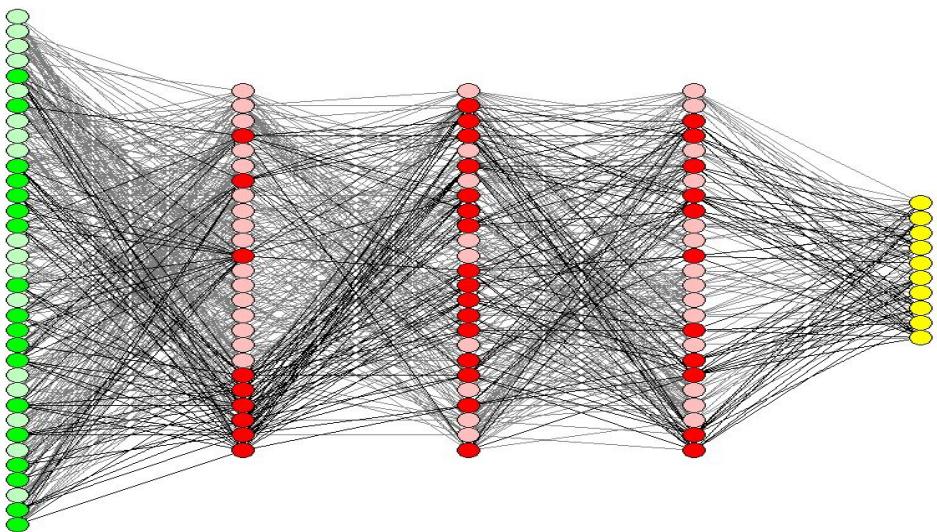




REDES CONVOLUCIONAIS

Introdução

Por que Aprendizagem profunda?



- resulta em uma quantidade grande de parâmetros à risco de sobreajuste (overfitting).
- inadequadas a imagens de alta resolução.
- tempo para computar as pré-ativações.



REDES CONVOLUCIONAIS

Introdução

Por que Aprendizagem profunda?

- **Expressividade:** é possível modelar uma função altamente não linear formada de uma hierarquia de funções não lineares mais simples.



REDES CONVOLUCIONAIS

Introdução

Por que Aprendizagem profunda?

Quanto mais dados e unidades, melhor!

- [...] what we're seeing consistently is that the bigger you can run these models, the better they perform. If you train one of these algorithms on one computer, you know, it will do pretty well. If you train them on 10, it will do even better. If you train on 100, even better. And we found that when we trained it on 16,000 CPU cores, [...], that was the best model we were able to train."
- "Now, why do we need so many processors? [...] The point was to have a software, maybe a little simulated baby brain."

Andrew Yan-Tak Ng





REDES CONVOLUCIONAIS

Introdução

Por que Aprendizagem profunda?

Quanto mais dados e unidades, melhor!

- “What was wrong in the 80’s is that we didn’t have enough data and we didn’t have enough computer power”



Geoffrey Everest Hinton



REDES CONVOLUCIONAIS

Introdução

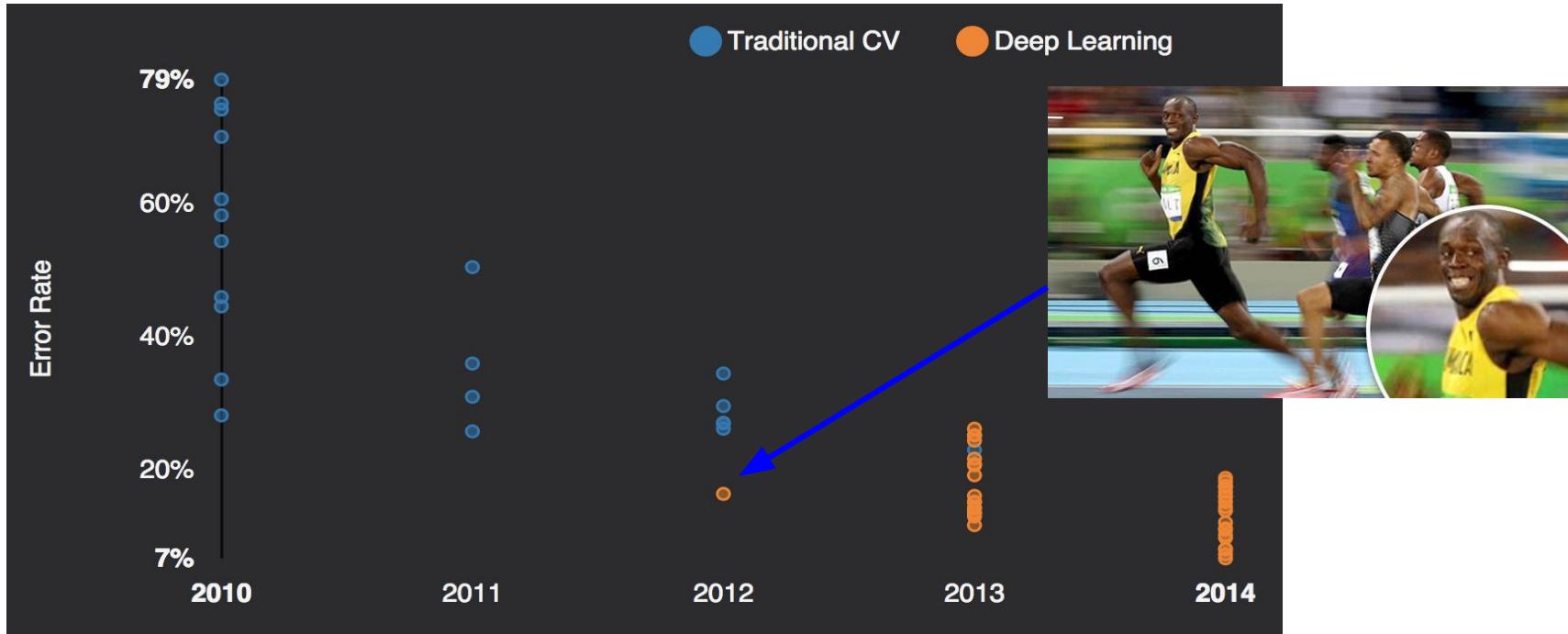


Gráfico reproduzido do material de Mathew Zeiler (Clarifai)



REDES CONVOLUCIONAIS



REDES CONVOLUCIONAIS

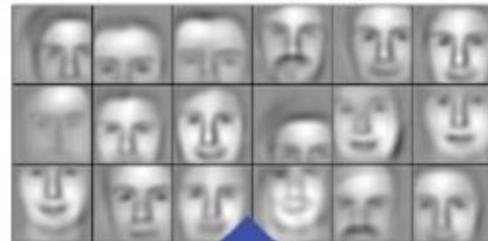
Hierarquia de características.

Quanto mais camadas melhor.

Características nos níveis mais altos da hierarquia são formadas pela combinação de características de mais baixo nível.

Composição de funções é um dos pilares da aprendizagem profunda.

Successive model layers learn deeper intermediate representations

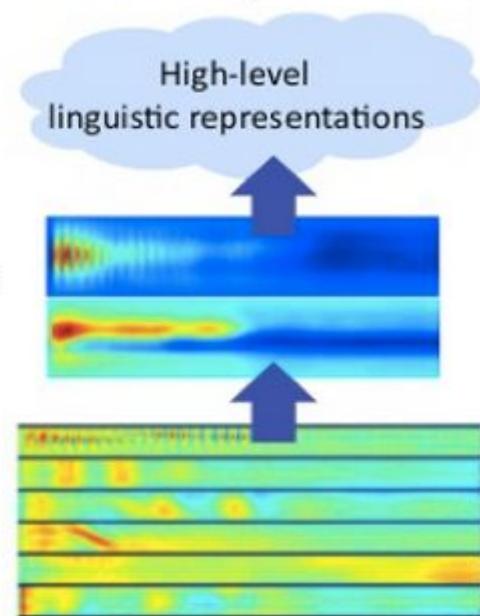


Layer 3

Parts combine
to form objects

Layer 2

Layer 1



Prior: underlying factors & concepts compactly expressed w/ multiple levels of abstraction



REDES CONVOLUCIONAIS

Hierarquia de características.

- Isso é conhecido como hierarquia de features, e é uma hierarquia de complexidade e abstração crescentes.
- Ela torna as redes de aprendizagem profunda capazes de lidar com conjuntos de dados muito grandes e de alta dimensão com bilhões de parâmetros que passam por funções não-lineares.
- Elas são capazes de descobrir estruturas latentes em dados não etiquetados e não estruturados, que são a grande maioria dos dados no mundo : Imagens, textos, vídeo e gravações de áudio.



REDES CONVOLUCIONAIS

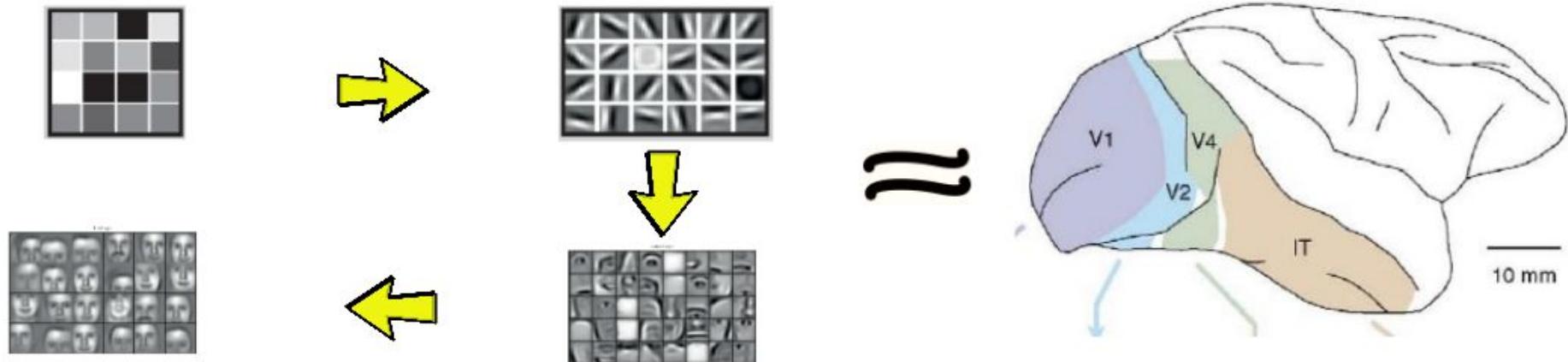
Hierarquia de características.

- Portanto, um dos problemas que a aprendizagem profunda resolve é o processamento e agrupamento de dados brutos, discernindo semelhanças e anomalias em dados que nenhum humano organizou em um banco de dados relacional.
- As redes de aprendizagem profunda realizam a extração automática de características sem intervenção humana, ao contrário da maioria dos algoritmos tradicionais de aprendizado de máquina.



REDES CONVOLUCIONAIS

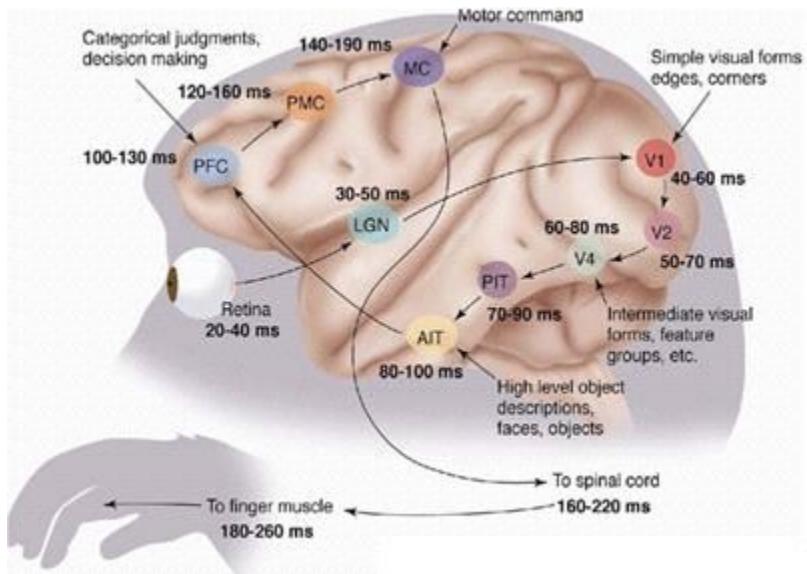
Base biológica





REDES CONVOLUCIONAIS

Base biológica

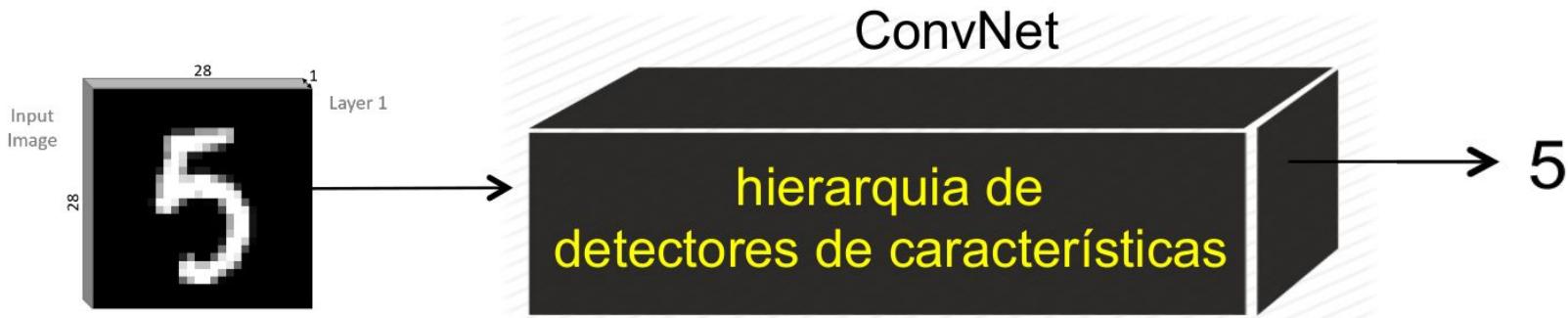




REDES CONVOLUCIONAIS

Base biológica

- Se inspiram no funcionamento do córtex visual.
- Sua arquitetura é adaptada para explorar a correlação espacial existente em imagens naturais.
- Evitam o problema da maldição da dimensionalidade por não ser totalmente conectada.
- São redes feed forward.





REDES CONVOLUCIONAIS

Conceitos Fundamentais

- Baseiam-se em algumas ideias básicas:
 - campos receptivos locais (local receptive fields)
 - Compartilhamento de pesos (shared weights)
 - Convolução (convolution)
 - Subamostragem (subsampling ou pooling).



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução

- A Convolução é uma operação matemática entre duas funções, e pode ser representada, na notação discreta, na forma:

$$(f * g) = \sum_{\tau} f(\tau)g(t - \tau)$$



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução

- A camada convolutiva consiste em dois parâmetros: peso e bias.
- Peso é especialmente chamado de kernel (filtro). Cada kernel é convoluído através de uma imagem de entrada.
- Suponha uma imagem I em 2D e o kernel K , então os kernels são convoluídos da seguinte forma.

$$(I * K)(i, t) = \sum_m \sum_n I(m, n)K(i - m, j - n)$$

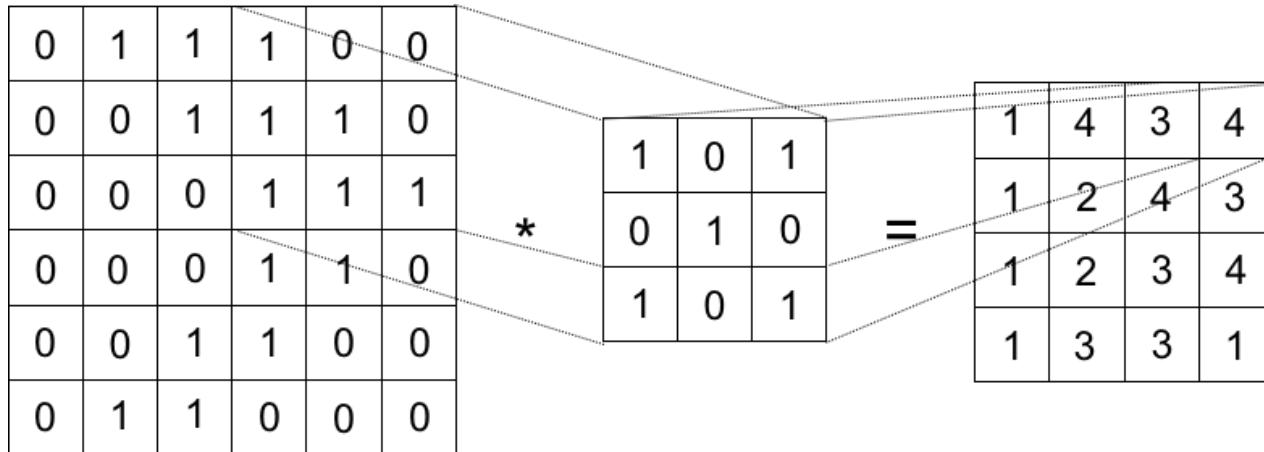


REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução

- Ao longo deste processo, a camada convolucional encontra o kernel ideal que é ativado quando a camada detecta características específicas na imagem.
- Assim, a camada poderá compactar a imagem e extrair as features.

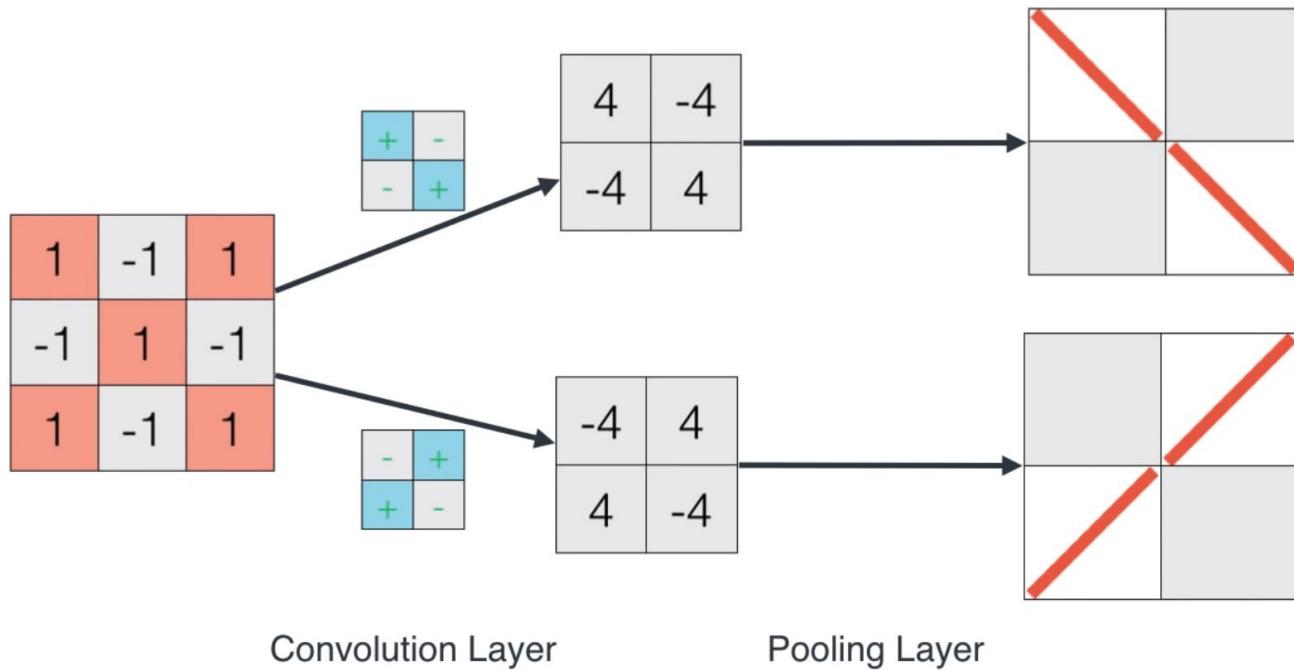




REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução





REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução

$$\begin{matrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{matrix}$$



$$\begin{matrix} \text{X} \end{matrix}$$

$$\begin{matrix} -1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & -1 \end{matrix}$$



$$\begin{matrix} \text{O} \end{matrix}$$

$$\begin{matrix} -1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{matrix}$$



$$\begin{matrix} / \end{matrix}$$

$$\begin{matrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{matrix}$$



$$\begin{matrix} \text{X} \end{matrix}$$



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução

1 <small>×1</small>	1 <small>×0</small>	1 <small>×1</small>	0	0
0 <small>×0</small>	1 <small>×1</small>	1 <small>×0</small>	1	0
0 <small>×1</small>	0 <small>×0</small>	1 <small>×1</small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução

- As imagens naturais têm a propriedade de ser "estacionárias", o que significa que as estatísticas de uma parte da imagem são as mesmas de qualquer outra parte.
- Isso sugere que as features que aprendemos em uma parte da imagem também podem ser aplicadas a outras partes da imagem, e podemos usar as mesmas features em todos os locais.



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução

- Mais precisamente, tendo aprendido features em pequenos (digamos 8×8) patches amostrados aleatoriamente a partir da imagem maior, podemos então aplicar este detector de feature 8×8 em qualquer lugar da imagem.
- Especificamente, podemos aproveitar as features 8×8 aprendidas e “convolucioná-las” com a imagem maior, obtendo assim um valor de ativação de feature diferente em cada local na imagem.

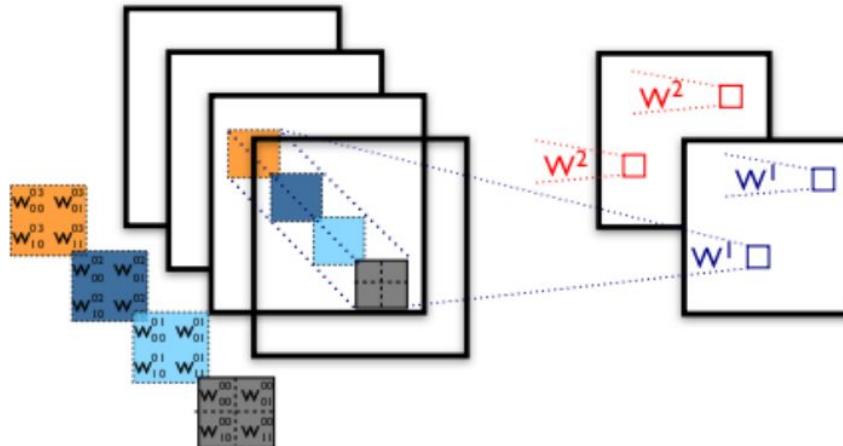


REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução

- Cada unidade em um filtro realiza uma convolução sobre seu respectivo campo receptivo.





REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução - stride

- Um hiperparâmetro da convolução é chamado de Stride.
- Embora o kernel geralmente move 1 posição vertical e horizontalmente pela imagem, definindo valores maiores que 1 no parâmetro stride, o filtro do kernel moverá este tamanho pela imagem.
- Por exemplo, se definirmos 2 como parâmetro stride, o kernel moverá 2 posições vertical e horizontalmente.



REDES CONVOLUCIONAIS

Conceitos Fundamentais Convolução - stride

0	0	0	0	0	0	0
0	105	102	100	97	96	
0	103	99	103	101	102	
0	101	98	104	102	100	
0	99	101	106	104	99	
0	104	104	104	100	98	

Image Matrix

$$\begin{aligned}
 & 0 * 0 + 0 * -1 + 0 * 0 \\
 & + 0 * -1 + 105 * 5 + 102 * -1 \\
 & + 0 * 0 + 103 * -1 + 99 * 0 = 320
 \end{aligned}$$

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

320

Output Matrix

0	0	0	0	0	0	0
0	105	102	100	97	96	
0	103	99	103	101	102	
0	101	98	104	102	100	
0	99	101	106	104	99	
0	104	104	104	100	98	

Image Matrix

$$\begin{aligned}
 & 0 * 0 + 0 * -1 + 0 * 0 \\
 & + 0 * -1 + 105 * 5 + 102 * -1 \\
 & + 0 * 0 + 103 * -1 + 99 * 0 = 320
 \end{aligned}$$

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

Output Matrix

Convolution with horizontal and vertical strides = 1

Convolution with horizontal and vertical strides = 2



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Convolução - Padding

Padding é uma técnica para preencher valores ao redor da imagem. Normalmente, o zero é preenchido em torno da imagem e é especificamente chamado de padding com zero.

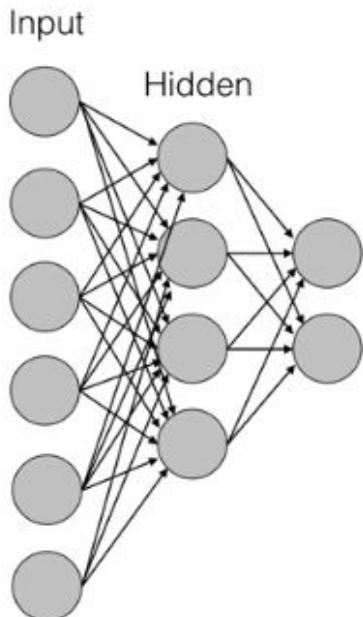
0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0
0	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Por que Convolução?



MNIST dataset: 28 x28 pixels (784 pixels)
First layer weights: ~78k parameters

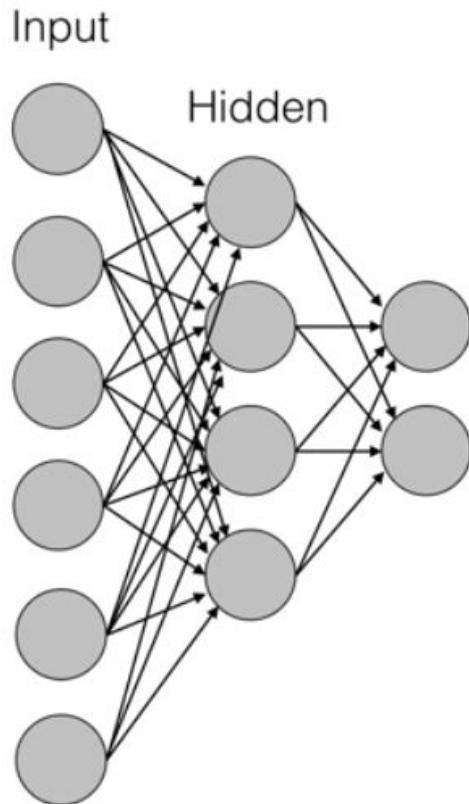
Typical Image: 256 x 256 (56,000 pixels)
First layer weights: 560k parameters !



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Por que Convolução?

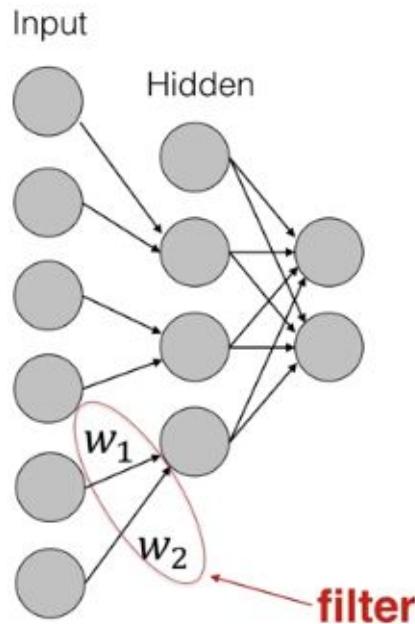




REDES CONVOLUCIONAIS

Conceitos Fundamentais

Por que Convolução?



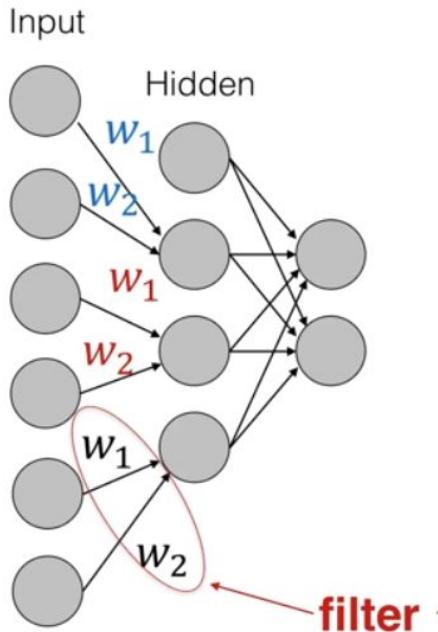
$$y = w_1x_1 + w_2x_2$$



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Por que Convolução?



$$y = w_1x_1 + w_2x_2$$

If $(w_1, w_2) = (1, -1)$: $y = x_1 - x_2$

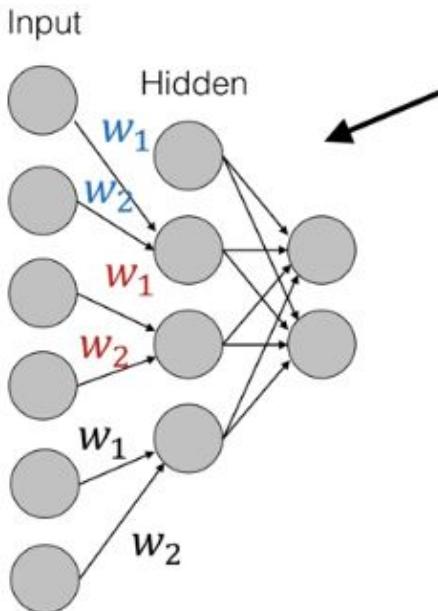
y maximal when $(x_1, x_2) = (1, 0)$



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Por que Convolução?

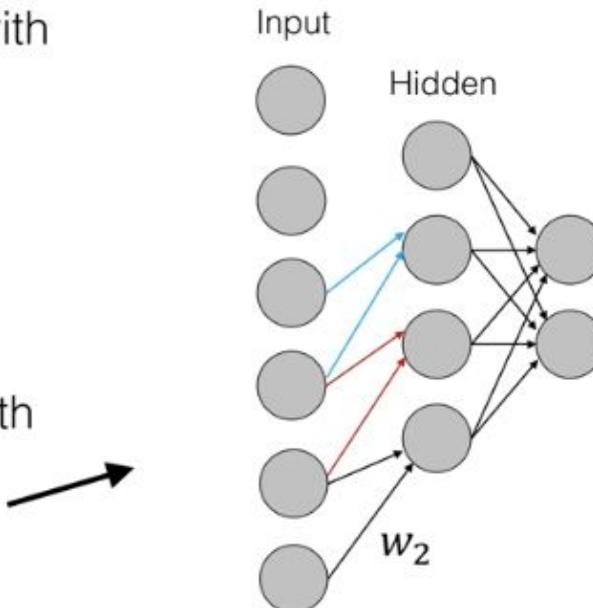


1-d convolution with

- filters: 1
- filter size: 2
- stride: 2

1-d convolution with

- filters: 1
- filter size: 2
- stride: **1**

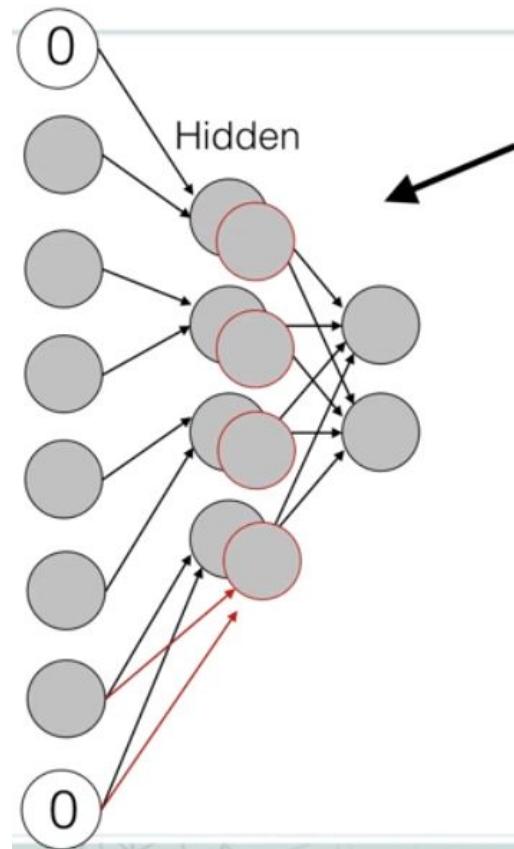




REDES CONVOLUCIONAIS

Conceitos Fundamentais

Por que Convolução?



1-d convolution with

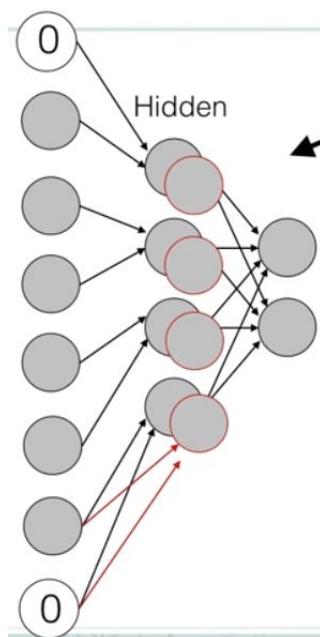
- filters: **2**
- filter size: 2
- stride: 2
- padding: 1



REDES CONVOLUCIONAIS

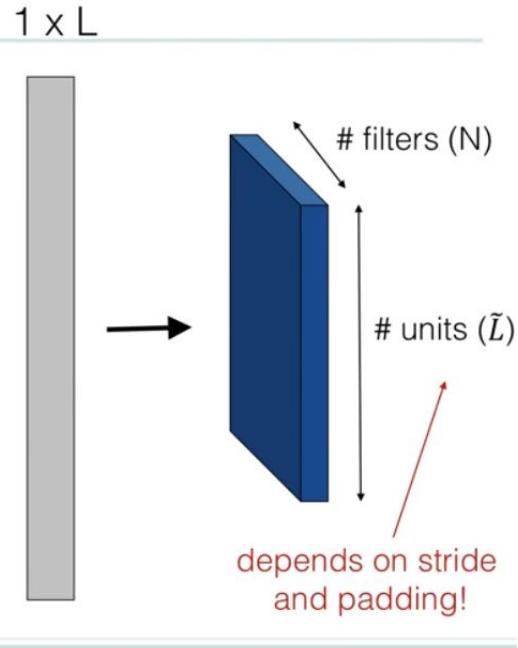
Conceitos Fundamentais

Por que Convolução?



1-d convolution with

- filters: **2**
- filter size: 2
- stride: 2
- padding: 1





REDES CONVOLUCIONAIS

Conceitos Fundamentais

Pooling

- Semelhante à camada convolucional, a camada de pooling também possui uma pequena janela (kernel).
- Aplicando a janela através de imagens, a camada de pooling realiza um processo estatístico.
- Existem duas camadas de pooling: average pooling e max pooling.



REDES CONVOLUCIONAIS

Conceitos Fundamentais

Pooling

21	8	8	12
12	19	9	7
8	10	4	3
18	12	9	10

avg pooling

15	9
12	7

max pooling

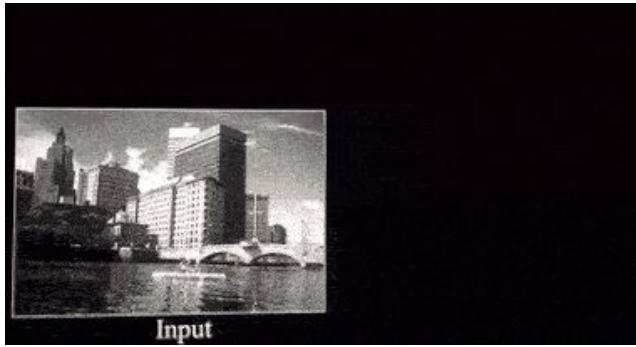
21	12
18	10



REDES CONVOLUCIONAIS

Execução

A operação mais importante na rede neural convolucional é a convolução, seja uma imagem $32 \times 32 \times 3$ se aplicarmos uma convolução $5 \times 5 \times 3$ nessa imagem (a profundidade do filtro deve ter a mesma profundidade da entrada), o resultado será um mapa de ativação $28 \times 28 \times 1$.

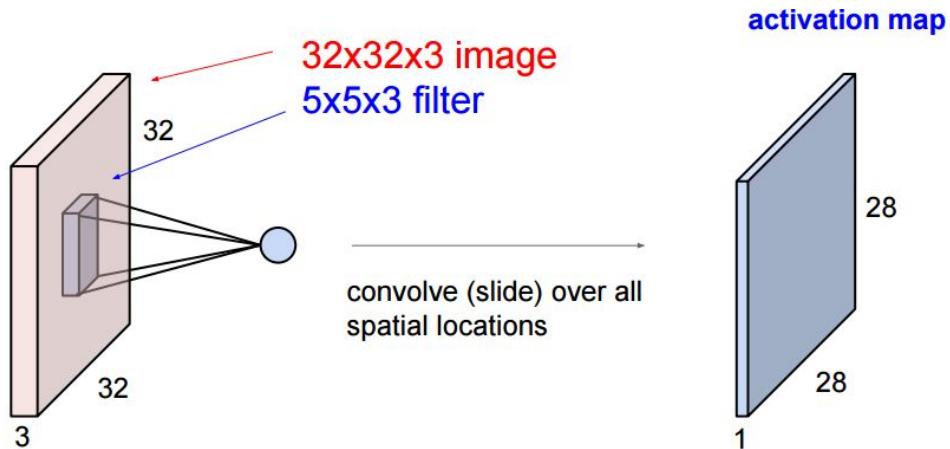




REDES CONVOLUCIONAIS

Execução

A operação mais importante na rede neural convolucional é a convolução, seja uma imagem $32 \times 32 \times 3$ se aplicarmos uma convolução $5 \times 5 \times 3$ nessa imagem (a profundidade do filtro deve ter a mesma profundidade da entrada), o resultado será um mapa de ativação $28 \times 28 \times 1$.

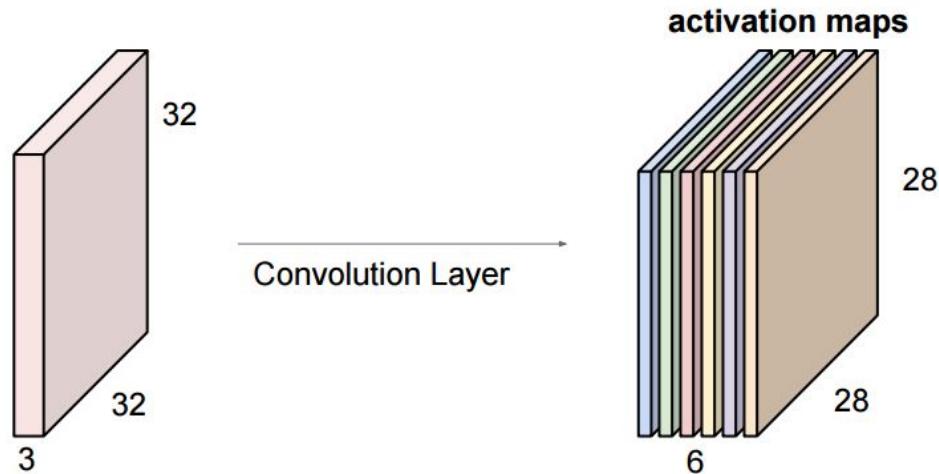




REDES CONVOLUCIONAIS

Execução

Agora, considere que queremos que nossa camada de convolução procure por 6 coisas diferentes. Neste caso, nossa camada de convolução terá 6 filtros $5 \times 5 \times 3$. Cada um procurando por um padrão particular na imagem.

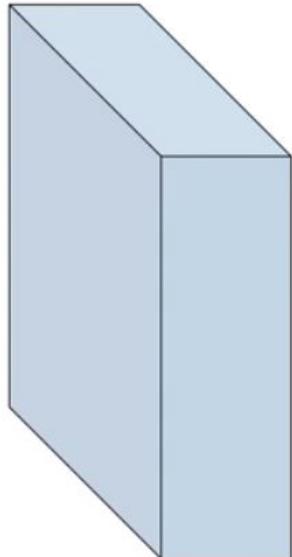




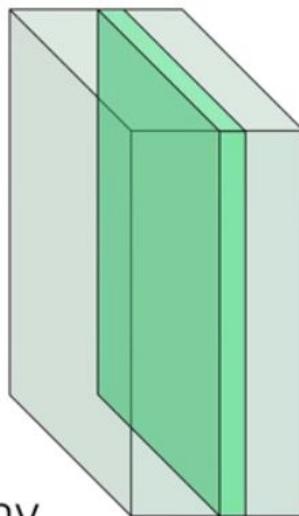
REDES CONVOLUCIONAIS

Execução

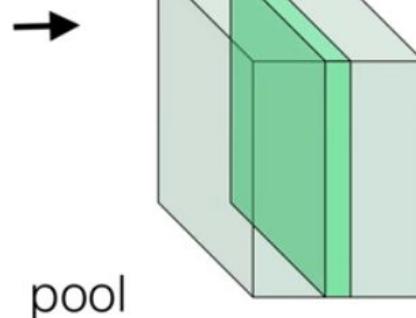
$224 \times 224 \times 3$



$224 \times 224 \times 64$



$112 \times 112 \times 64$



conv

pool

0	1	4	9
3	2	5	8
1	2	3	1
3	1	7	4

Max pool:
2x2 filters
Stride 2

3	9
3	7

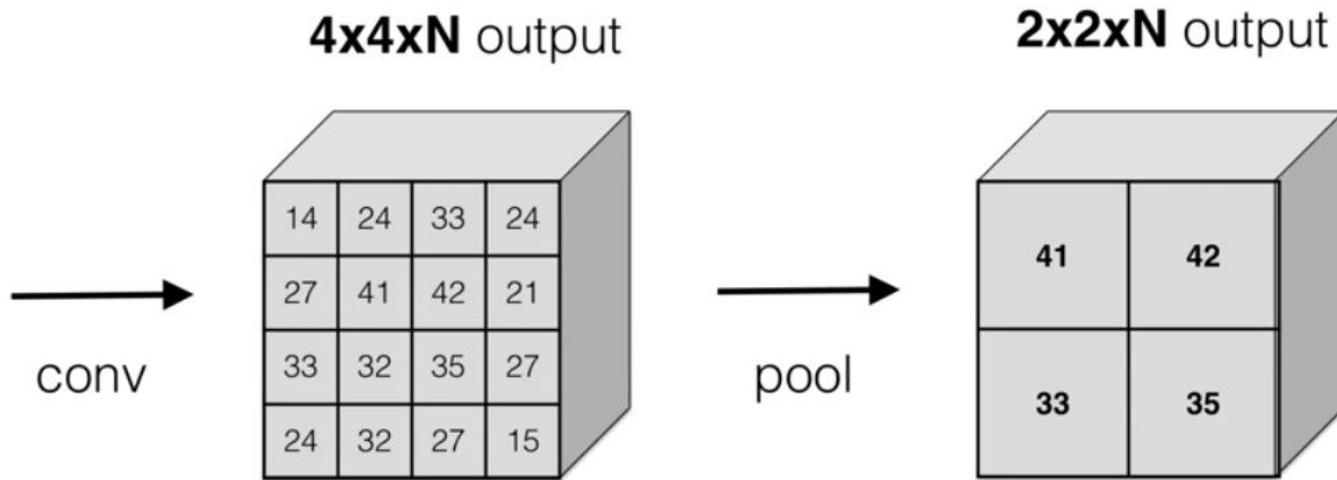


REDES CONVOLUCIONAIS

Execução

- Conv layers (3x3 filter with stride 1 and pad 1)
- Pool layers (2x2 max pooling with stride 2 and pad 0)

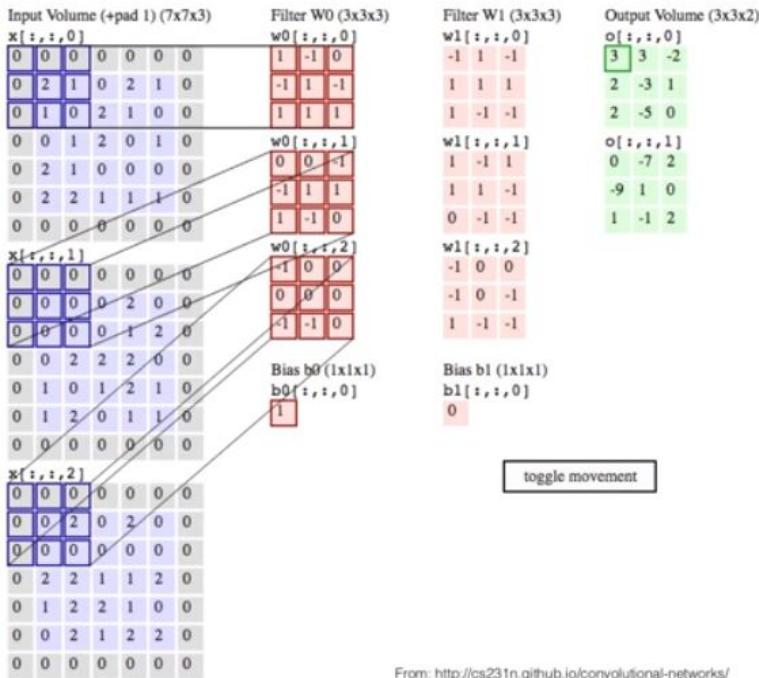
0	0	0	0	0	0
0	1	5	3	9	0
0	4	4	7	5	0
0	6	7	5	6	0
0	6	5	3	1	0
0	0	0	0	0	0



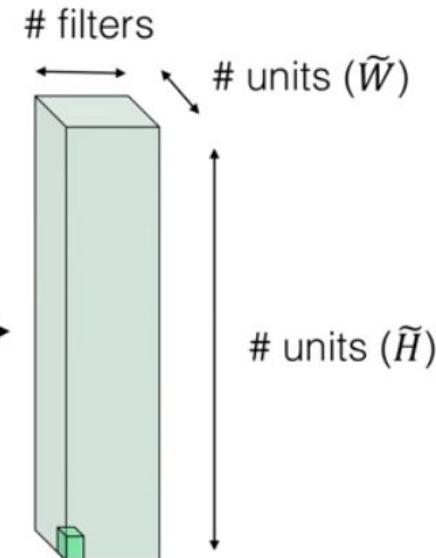
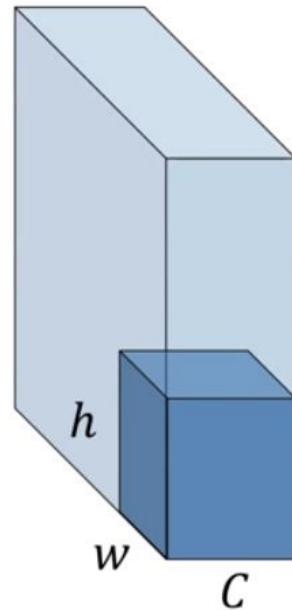


REDES CONVOLUCIONAIS

Execução



$H \times W \times C$



$$f_1 = (w, h, C)$$



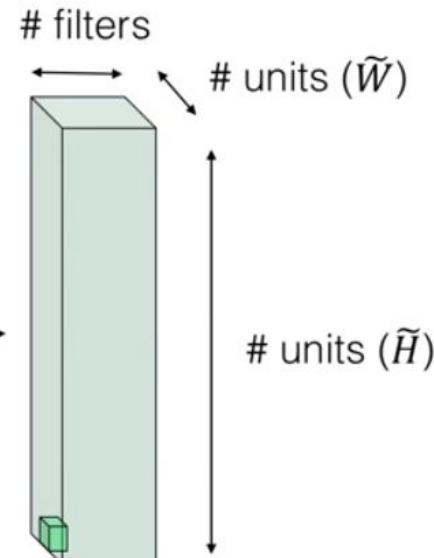
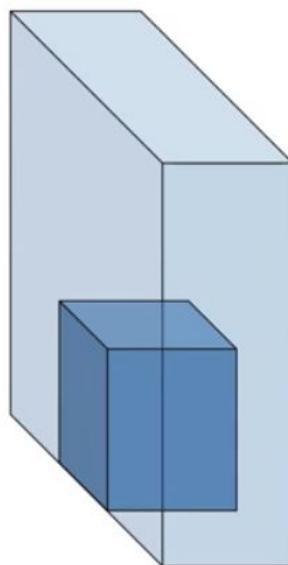
REDES CONVOLUCIONAIS

Execução

Input Volume (+pad 1) (7x7x3)	Filter W0 (3x3x3)	Output Volume (3x3x2)
$x[:, :, 0]$ 0 0 0 0 0 0 0 0 2 1 0 2 1 0 0 1 0 2 1 0 0 0 0 1 2 0 1 0 0 2 1 0 0 0 0 0 2 2 1 1 1 0 0 0 0 0 0 0 0	$w0[:, :, 0]$ 1 -1 0 -1 1 -1 1 1 1 $w0[:, :, 1]$ 0 0 -1 -1 1 1 1 -1 0 $w0[:, :, 2]$ -1 0 0 0 0 0 -1 -1 0	$o[:, :, 0]$ 3 3 -2 1 1 1 1 -1 -1 $o[:, :, 1]$ 1 -1 1 1 1 -1 0 -1 -1 $o[:, :, 2]$ -1 0 0 -1 0 -1 1 -1 -1
$x[:, :, 1]$ 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 1 2 0 0 0 2 2 2 0 0 0 1 0 1 2 1 0 0 1 2 0 1 1 0 0 0 0 0 0 0 0		
$x[:, :, 2]$ 0 0 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 2 2 1 1 2 0 0 1 2 2 1 0 0 0 0 2 1 2 2 0 0 0 0 0 0 0 0		

toggle movement

$H \times W \times C$

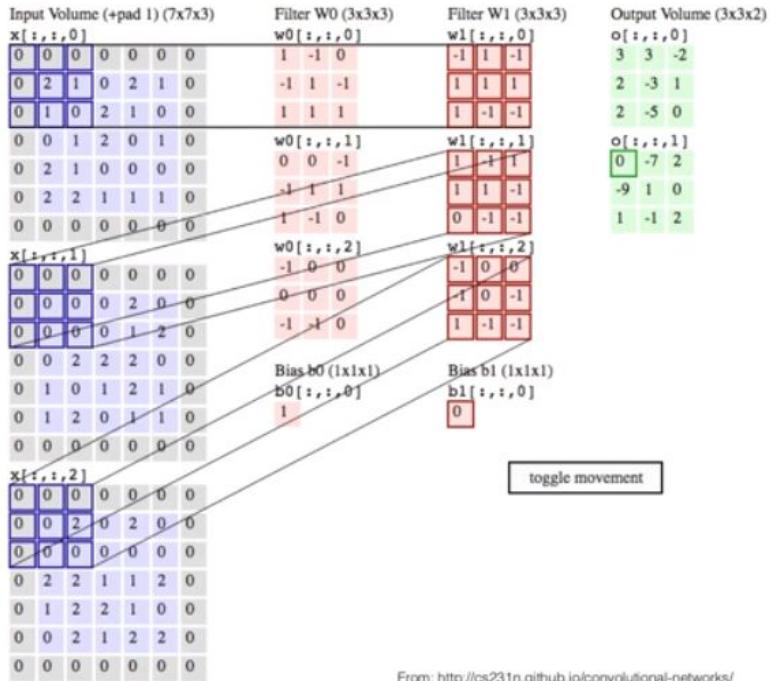


$$f_1 = (w, h, C)$$

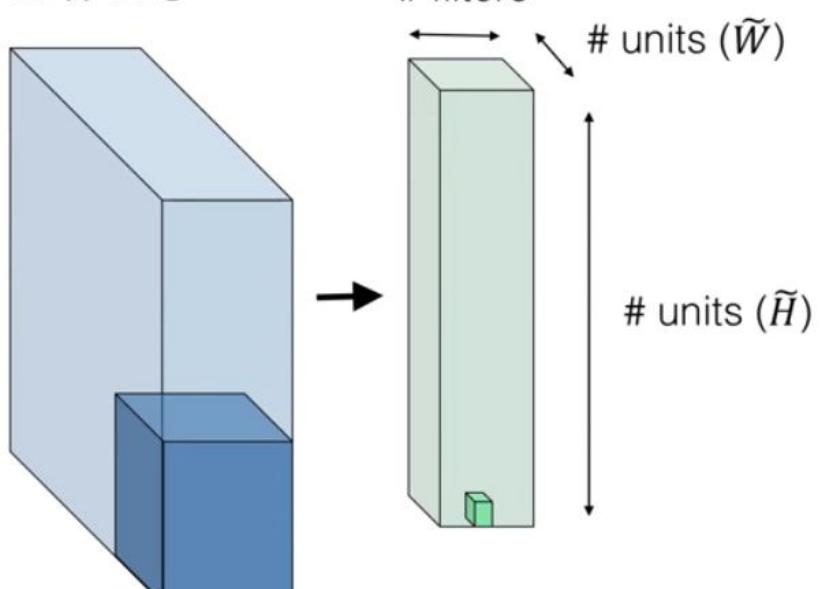


REDES CONVOLUCIONAIS

Execução



$H \times W \times C$

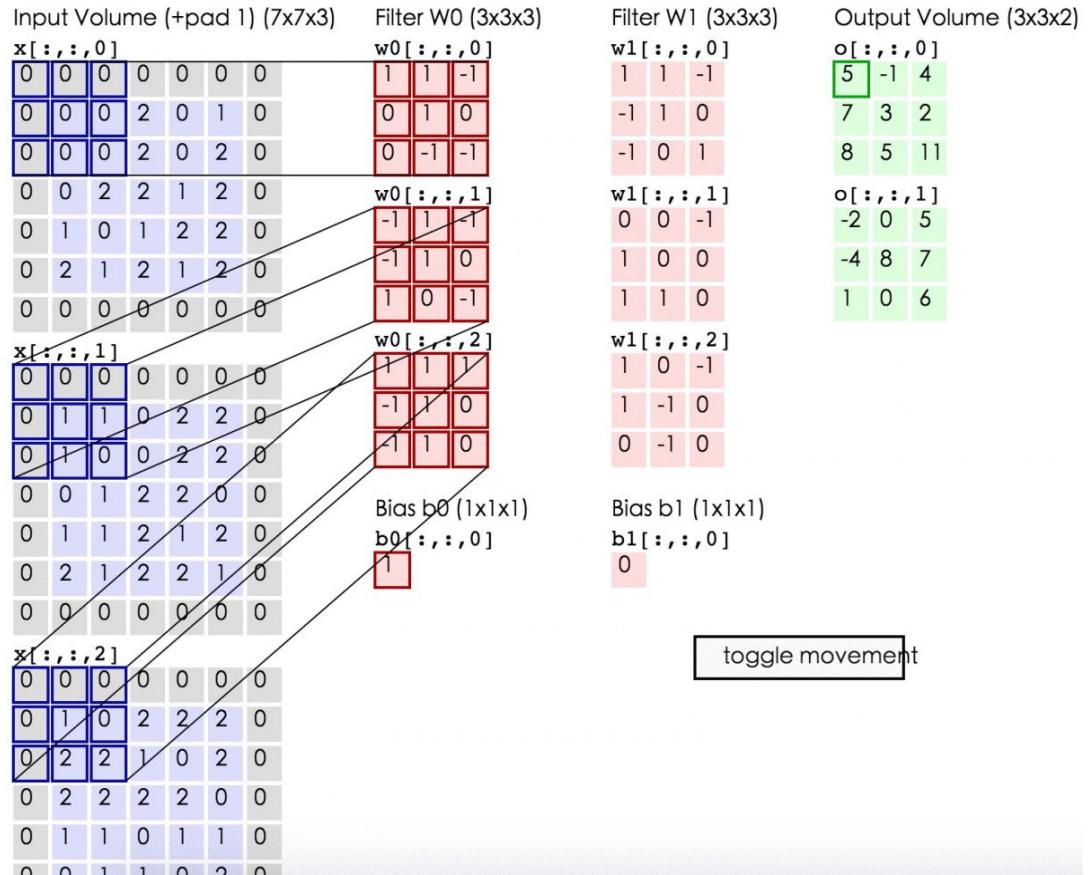


$$f_2 = (w, h, C)$$



REDES CONVOLUCIONAIS

Execução

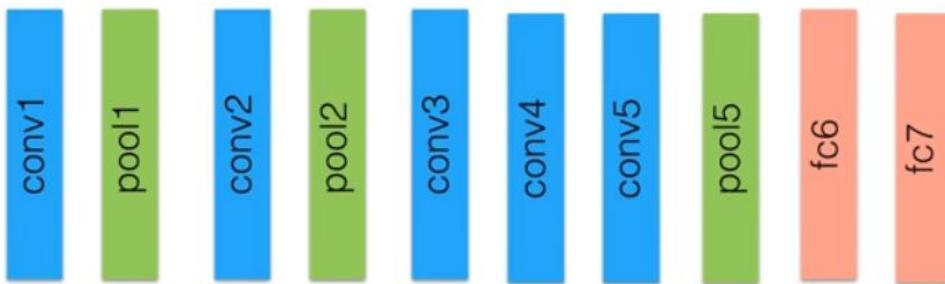
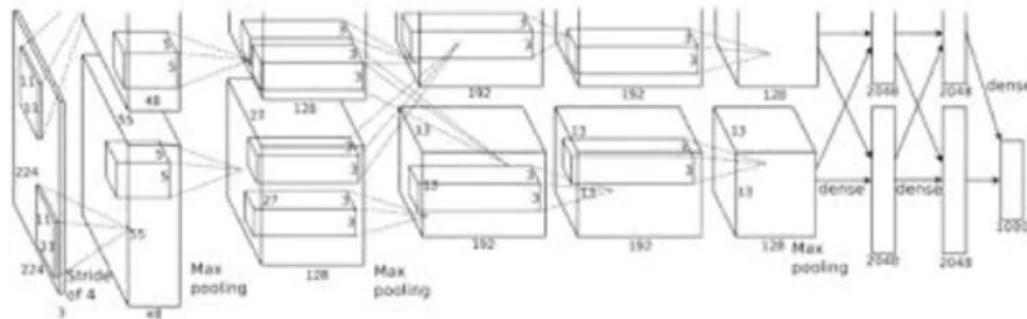




REDES CONVOLUCIONAIS

Alguns exemplos

- **AlexNet (ILSVRC 2012 winner)**
- ZF Net (2013 winner)
- GoogLeNet (2014 winner)
- VGG (2014 runner-up)
- ResNet (2015 winner)





REDES CONVOLUCIONAIS

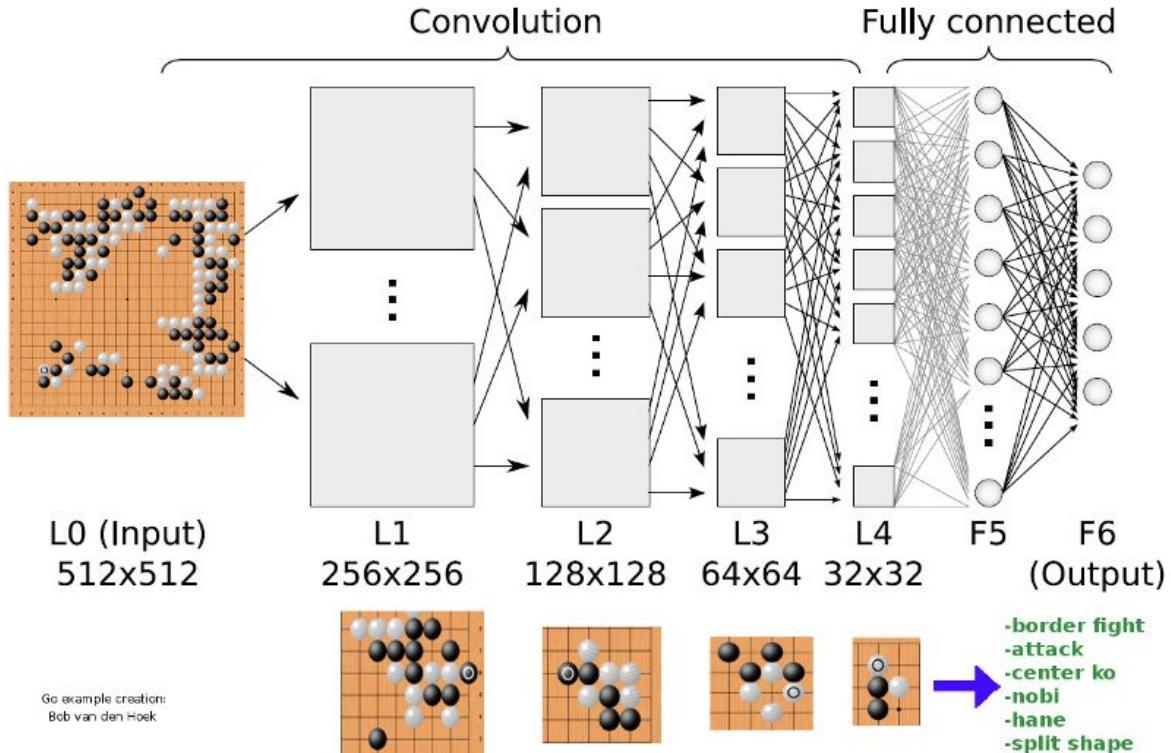
Alguns exemplos

Layer	Output shape
Input	(224, 224, 3)
CONV (3x3x64)	(224, 224, 64)
CONV (3x3x64)	(224, 224, 64)
POOL (2x2)	(112, 112, 64)
CONV (3x3x128)	(224, 224, 128)
CONV (3x3x128)	(224, 224, 128)
POOL (2x2)	(56, 56, 128)
CONV (3x3x256)	(56, 56, 256)
CONV (3x3x256)	(56, 56, 256)
CONV (3x3x256)	(56, 56, 256)
POOL (2x2)	(28, 28, 256)
CONV (3x3x256)	(28, 28, 512)
CONV (3x3x256)	(28, 28, 512)
CONV (3x3x256)	(28, 28, 512)
POOL (2x2)	(14, 14, 512)
CONV (3x3x512)	(14, 14, 512)
CONV (3x3x512)	(14, 14, 512)
CONV (3x3x512)	(14, 14, 512)
POOL (2x2)	(7, 7, 512)
AFFINE (4096 units)	(4096, 1)
AFFINE (4096 units)	(4096, 1)
AFFINE (100 units)	(100, 1)



REDES CONVOLUCIONAIS

Alguns exemplos





REDES CONVOLUCIONAIS

Links interessantes

[https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/convolutional neural networks.html](https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/convolutional_neural_networks.html) (livro online)

<http://ufldl.stanford.edu/tutorial/> (curso de Stanford)

http://machinelearninguru.com/computer_vision/basics/convolution/convolution_layer.html

<http://cs231n.github.io/> (Outro curso de Stanford)