

# Redes Neurais e Aprendizagem Profunda

## REDES NEURAIS CONVOLUCIONAIS

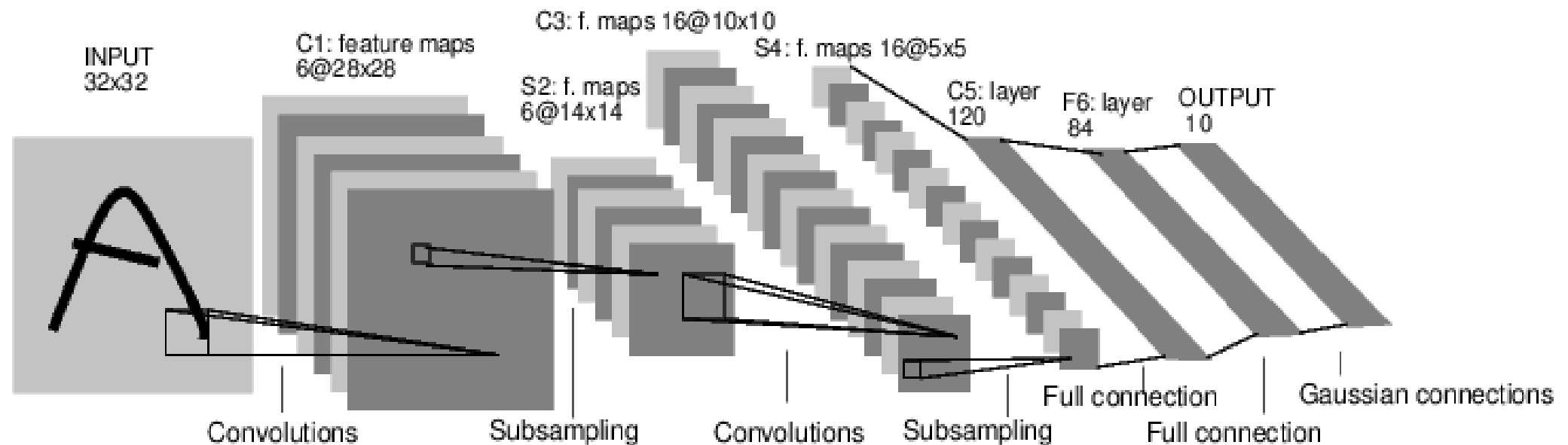
### LENET / ALEXNET / VGG

---

Zenilton K. G. Patrocínio Jr  
[zenilton@pucminas.br](mailto:zenilton@pucminas.br)

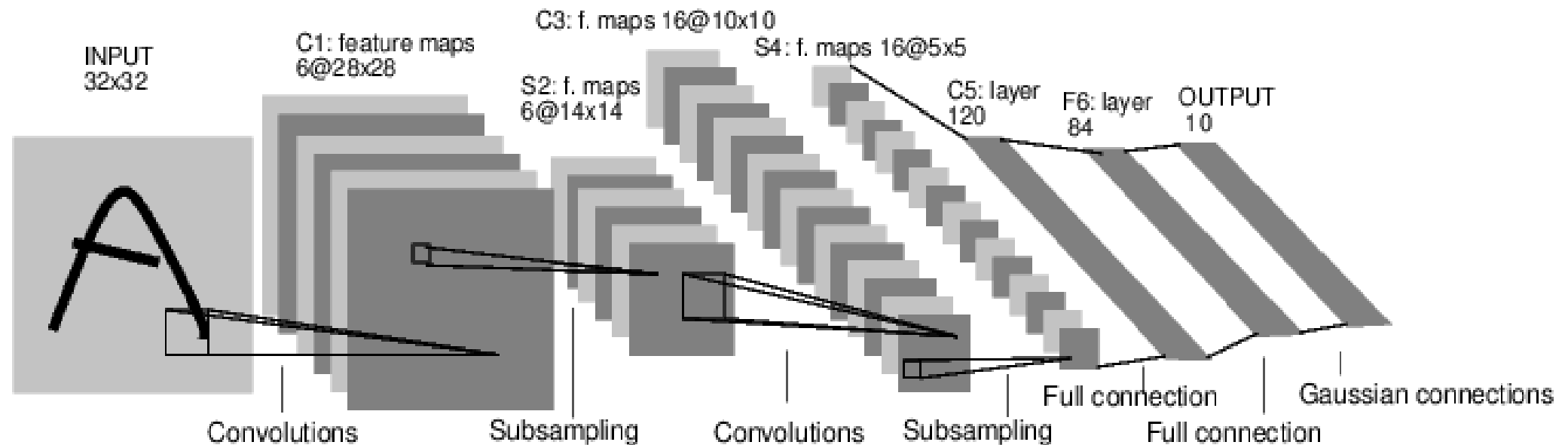
# ConvNets – LeNet

[LeCun et al., 1998]



# ConvNets – LeNet

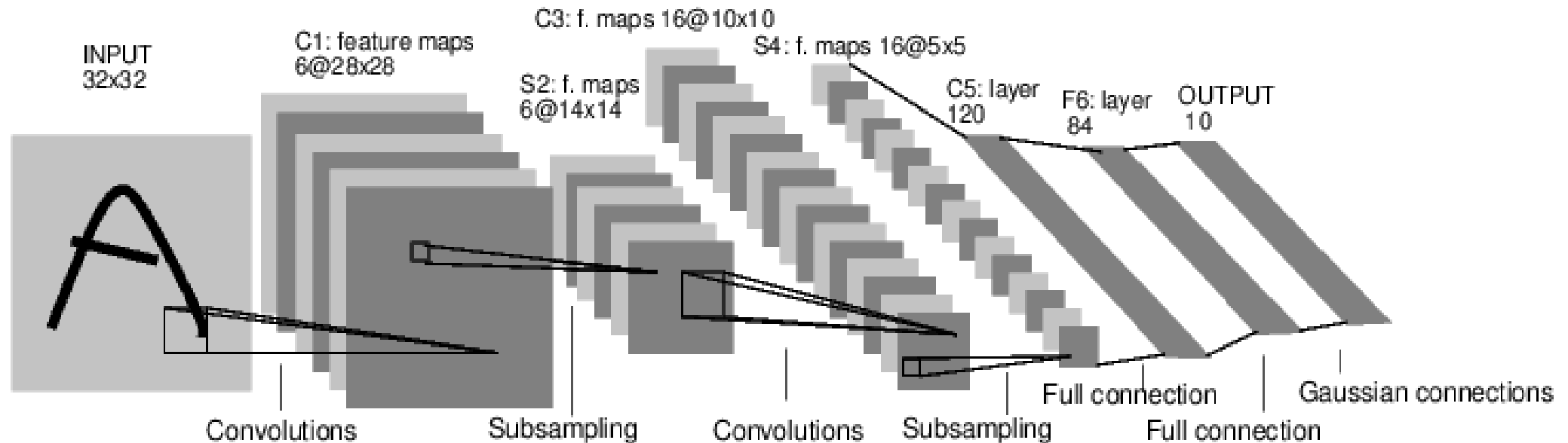
[LeCun et al., 1998]



- Filtros convolucionais 5×5 com passo 1

# ConvNets – LeNet

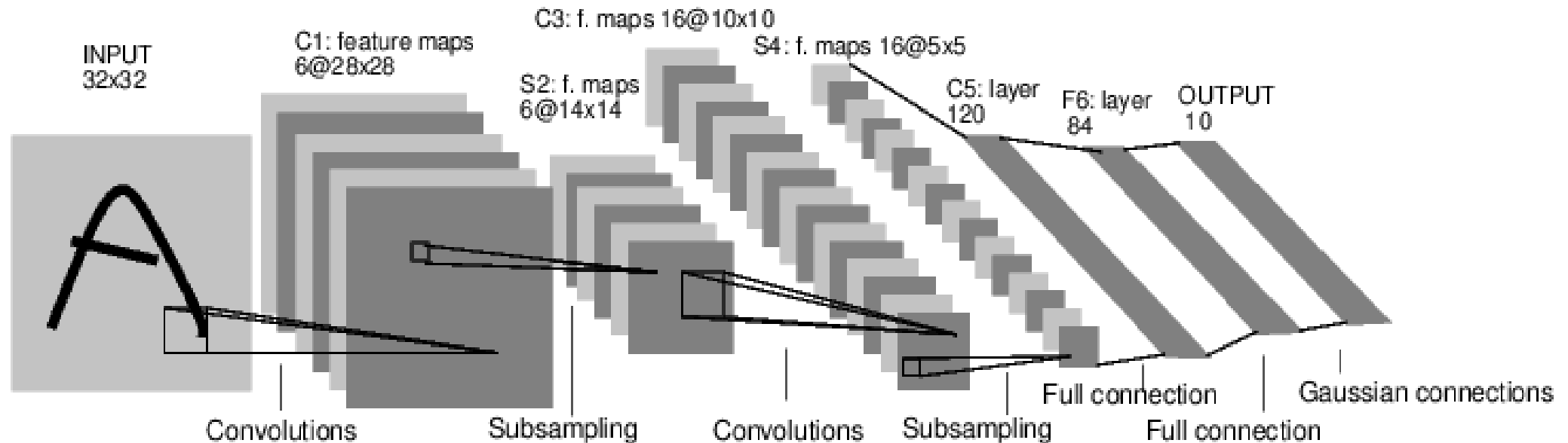
[LeCun et al., 1998]



- Filtros convolucionais 5×5 com passo 1
- Camadas de agrupamento (“subsampling”) com filtros 2×2 e passo 2

# ConvNets – LeNet

[LeCun et al., 1998]

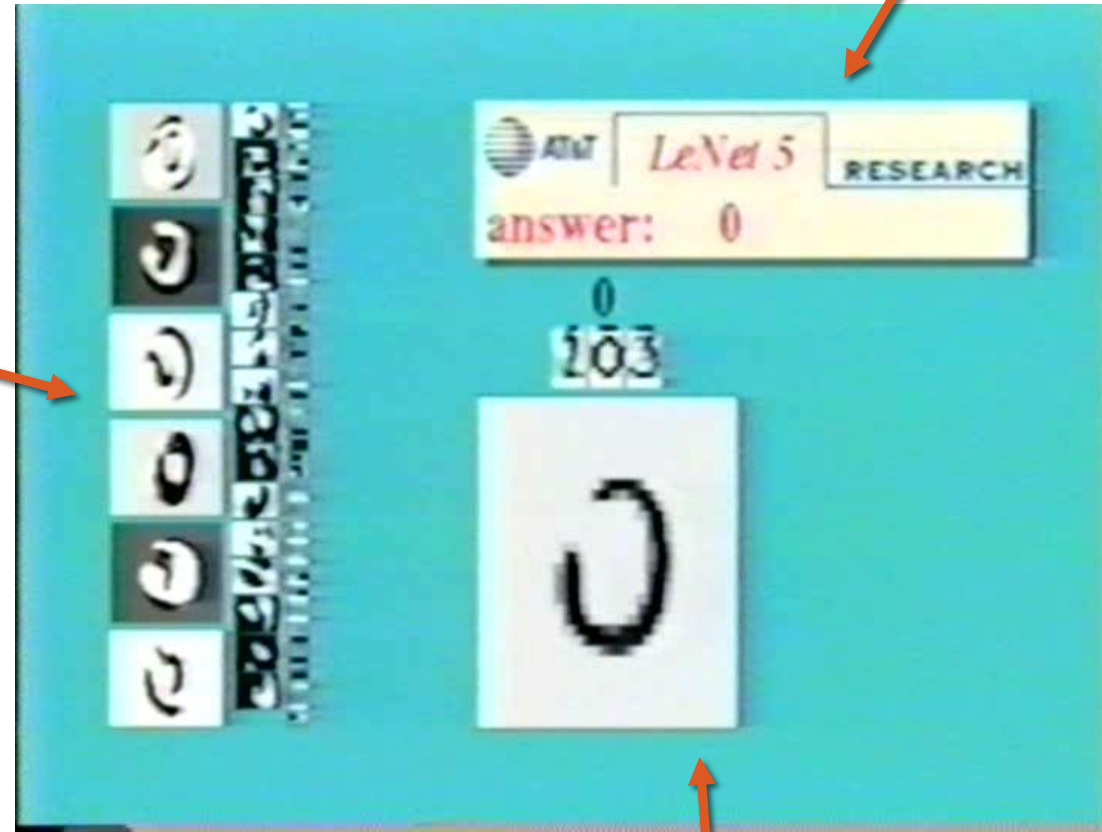


- Filtros convolucionais 5×5 com passo 1
- Camadas de agrupamento (“subsampling”) com filtros 2×2 e passo 2
- Arquitetura → [CONV-POOL-CONV-POOL-FC-FC]

# ConvNets – LeNet

## Classificação de Dígitos Manuscritos

Características  
extraídas pelo  
filtros

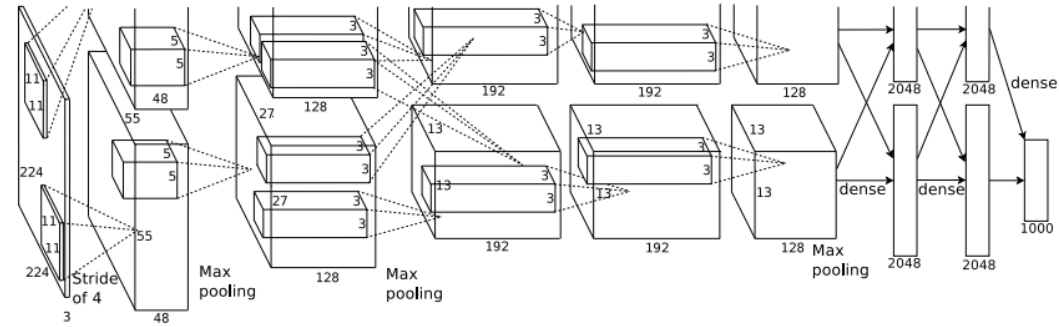


Saída

Entrada

# ConvNets – AlexNet

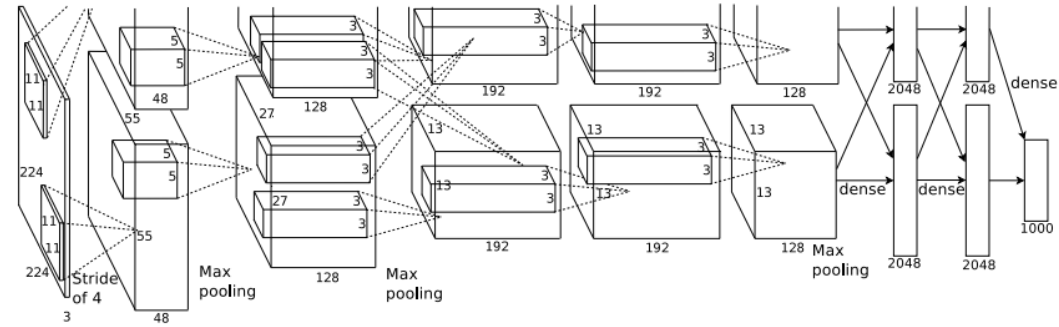
[Krizhevsky et al. 2012]



# ConvNets – AlexNet

[Krizhevsky et al. 2012]

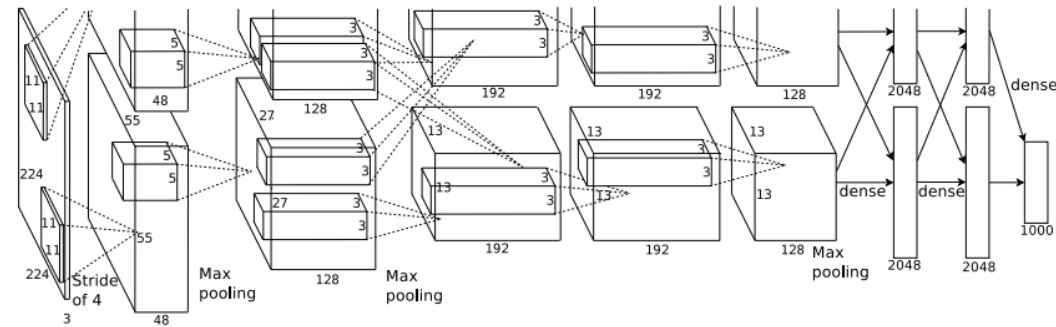
Entrada: imagem  $227 \times 227 \times 3$





# ConvNets – AlexNet

**[Krizhevsky et al. 2012]**

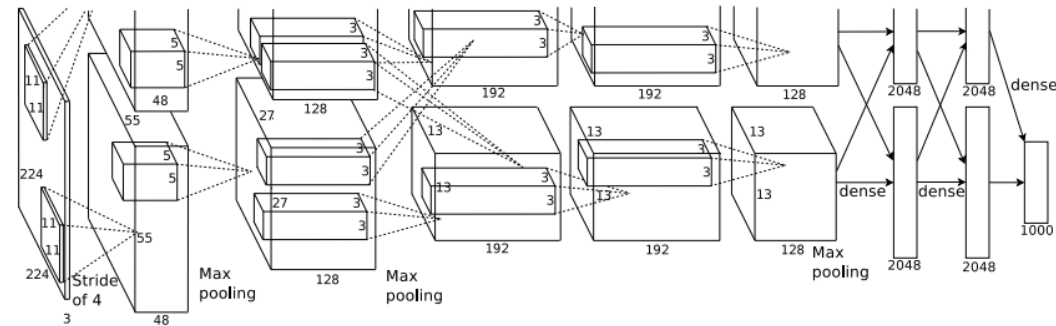


Entrada: imagem  $227 \times 227 \times 3$

**Primeira camada (CONV1):** 96 filtros  $11 \times 11$  com passo 4

# ConvNets – AlexNet

[Krizhevsky et al. 2012]



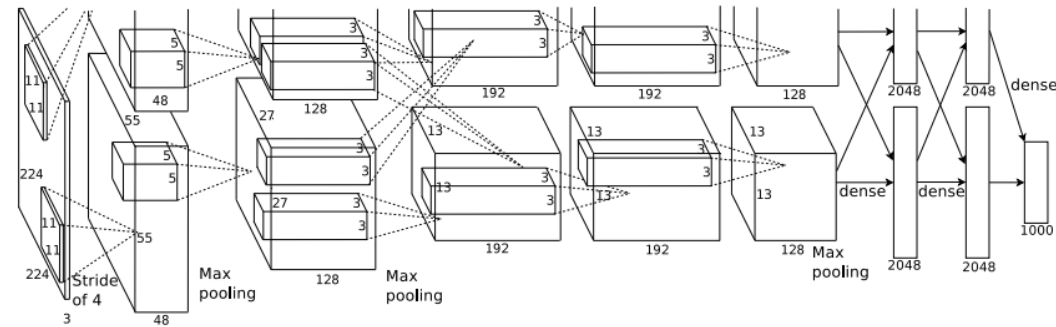
Entrada: imagem 227×227×3

**Primeira camada (CONV1):** 96 filtros 11×11 com passo 4

P: Qual é o tamanho do volume de saída? Dica:  $(227-11)/4+1 = 55$

# ConvNets – AlexNet

[Krizhevsky et al. 2012]



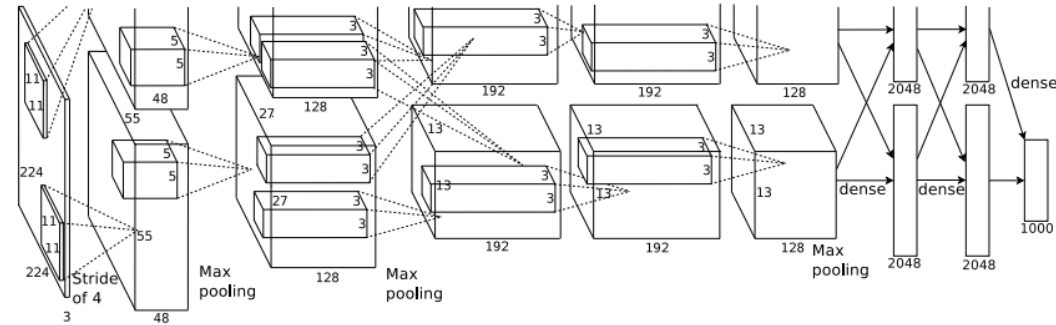
Entrada: imagem  $227 \times 227 \times 3$

**Primeira camada (CONV1):** 96 filtros  $11 \times 11$  com passo 4

Tamanho do volume de saída =  **$55 \times 55 \times 96$**

# ConvNets – AlexNet

[Krizhevsky et al. 2012]



Entrada: imagem  $227 \times 227 \times 3$

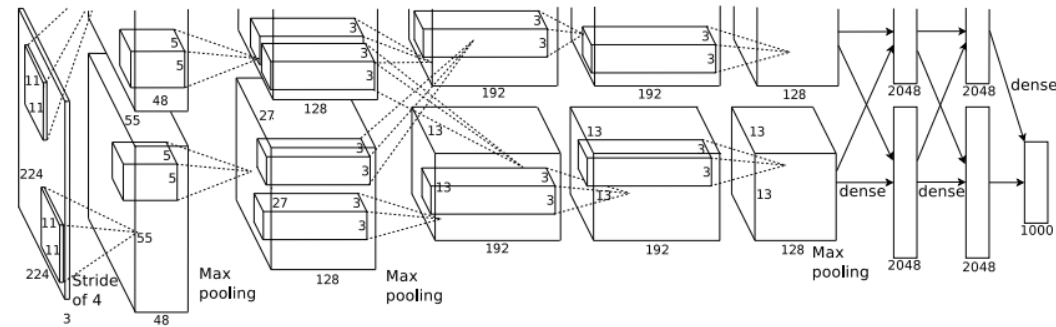
**Primeira camada (CONV1):** 96 filtros  $11 \times 11$  com passo 4

Tamanho do volume de saída =  **$55 \times 55 \times 96$**

P: Qual é o número total de parâmetros nessa camada?

# ConvNets – AlexNet

[Krizhevsky et al. 2012]



Entrada: imagem 227×227×3

**Primeira camada (CONV1):** 96 filtros 11×11 com passo 4

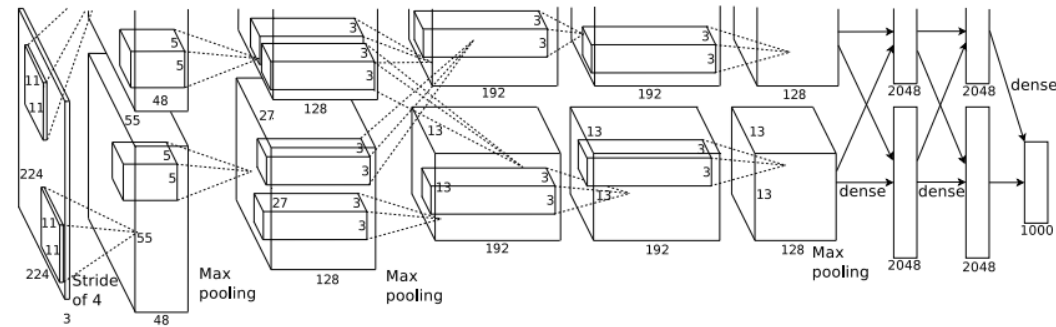
Tamanho do volume de saída = **[55×55×96]**

Número de parâmetros =  $(11 \times 11 \times 3 + 1) \times 96 \approx$  **35K**

# ConvNets – AlexNet

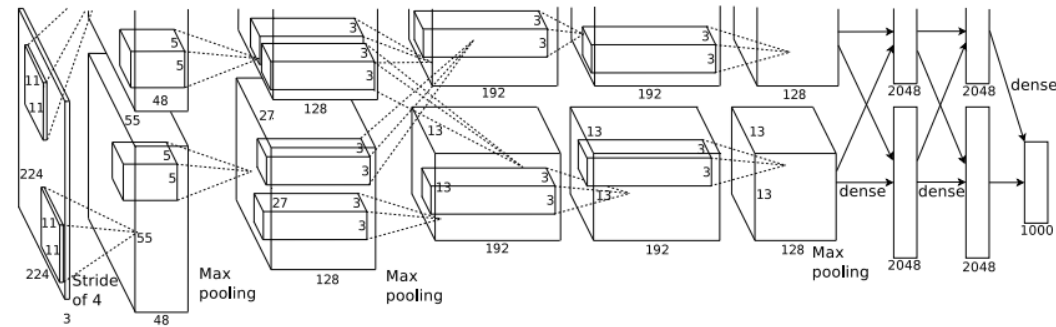
[Krizhevsky et al. 2012]

Entrada: imagem  $227 \times 227 \times 3$   
Depois de CONV1:  $55 \times 55 \times 96$



# ConvNets – AlexNet

[Krizhevsky et al. 2012]



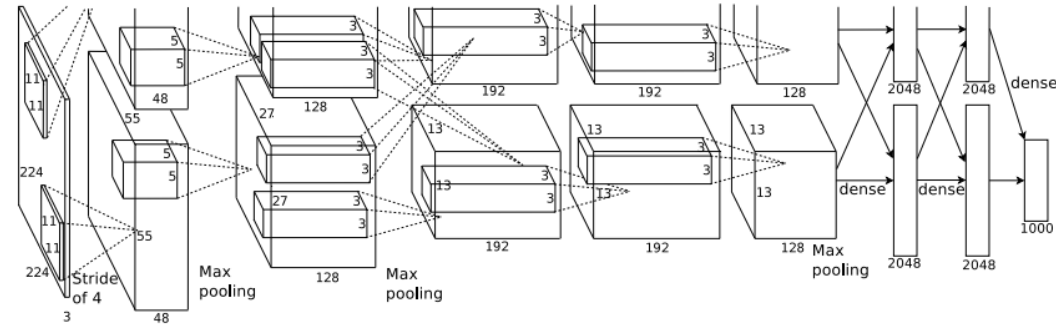
Entrada: imagem  $227 \times 227 \times 3$

Depois de CONV1:  $55 \times 55 \times 96$

**Segunda camada (POOL1):** 96 filtros  $3 \times 3$  com passo 2

# ConvNets – AlexNet

**[Krizhevsky et al. 2012]**



Entrada: imagem  $227 \times 227 \times 3$

Depois de CONV1:  $55 \times 55 \times 96$

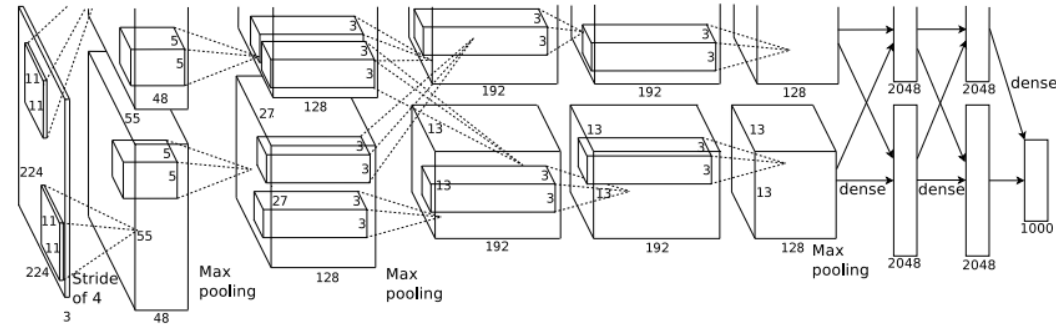
**Segunda camada (POOL1): 96 filtros 3×3 com passo 2**

P: Qual é o tamanho do volume de saída? Dica:  $(55-3)/2+1 = 27$



# ConvNets – AlexNet

[Krizhevsky et al. 2012]



Entrada: imagem  $227 \times 227 \times 3$

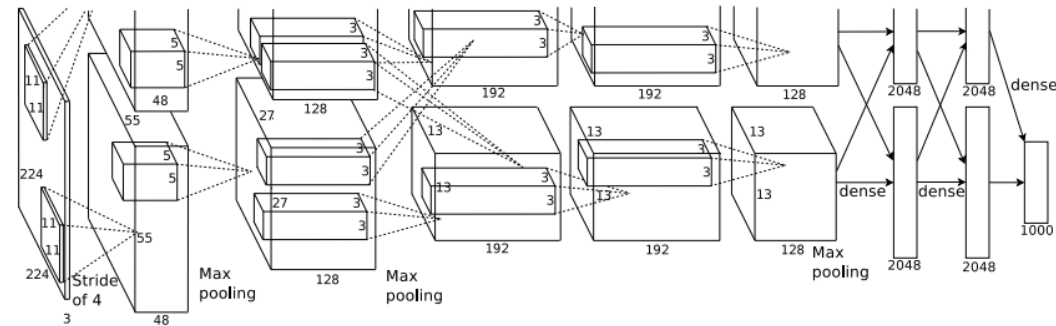
Depois de CONV1:  $55 \times 55 \times 96$

**Segunda camada (POOL1):** 96 filtros  $3 \times 3$  com passo 2

Tamanho do volume de saída =  **$27 \times 27 \times 96$**

# ConvNets – AlexNet

[Krizhevsky et al. 2012]



Entrada: imagem  $227 \times 227 \times 3$

Depois de CONV1:  $55 \times 55 \times 96$

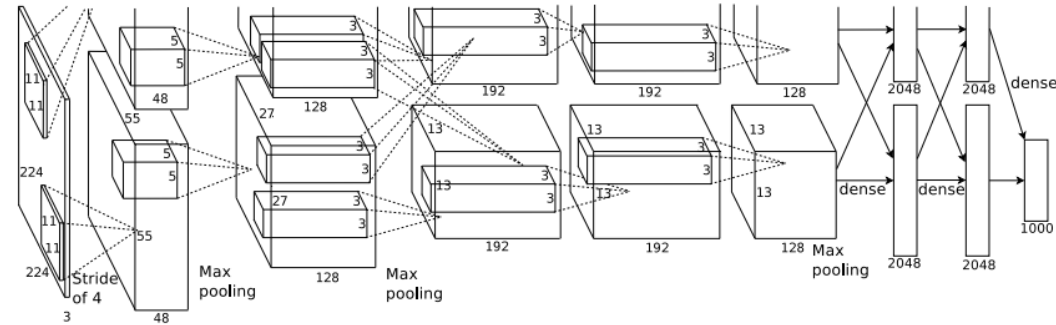
**Segunda camada (POOL1):** 96 filtros  $3 \times 3$  com passo 2

Tamanho do volume de saída =  **$27 \times 27 \times 96$**

P: Qual é o número total de parâmetros nessa camada?

# ConvNets – AlexNet

[Krizhevsky et al. 2012]



Entrada: imagem  $227 \times 227 \times 3$   
Depois de CONV1:  $55 \times 55 \times 96$

**Segunda camada (POOL1):** 96 filtros  $3 \times 3$  com passo 2

Tamanho do volume de saída =  **$[27 \times 27 \times 96]$**

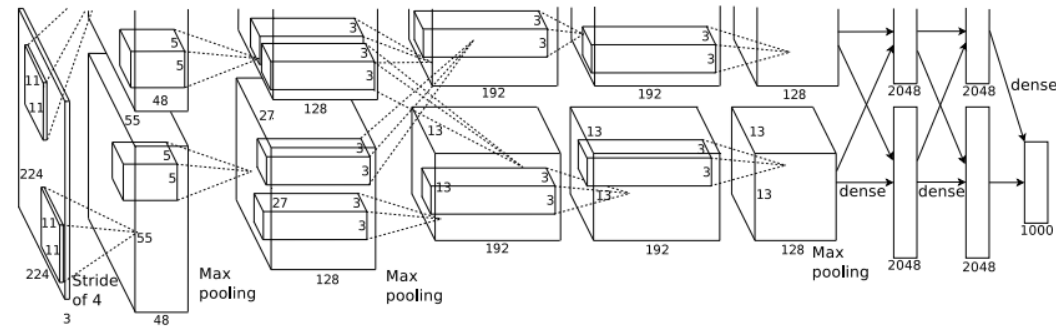
Número de parâmetros = **0** !

# ConvNets – AlexNet

[Krizhevsky et al. 2012]

Entrada: imagem  $227 \times 227 \times 3$   
Depois de CONV1:  $55 \times 55 \times 96$   
Depois de POOL1:  $27 \times 27 \times 96$

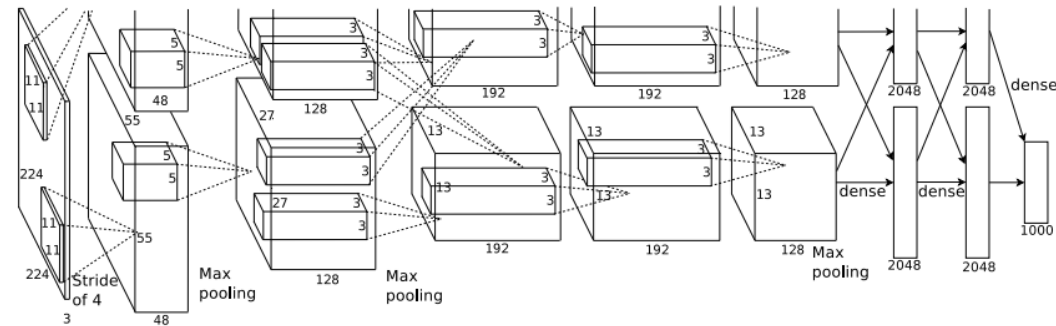
⋮



# ConvNets – AlexNet

[Krizhevsky et al. 2012]

## Arquitetura da AlexNet:

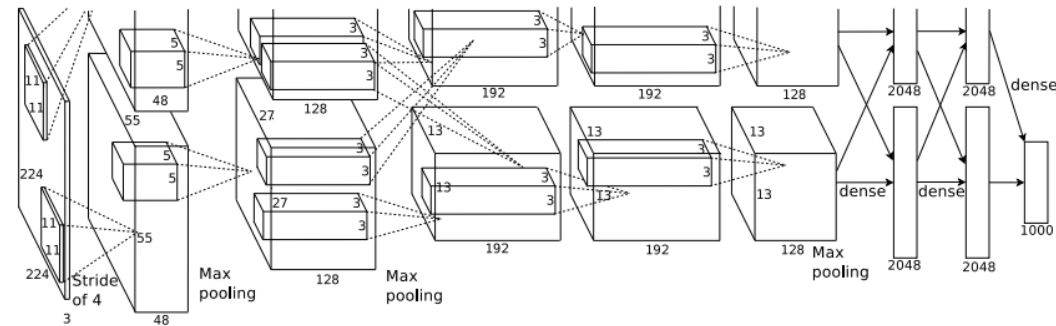


- [227×227×3] Entrada
- [55×55×96] **CONV1**: 96 filtros 11×11 passo 4, pad 0
- [27×27×96] **MAX POOL1**: 96 filtros 3×3 passo 2
- [27×27×96] **NORM1**: camada de normalização
- [27×27×256] **CONV2**: 256 filtros 5×5 passo 1, pad 2
- [13×13×256] **MAX POOL2**: 256 filtros 3×3 passo 2
- [13×13×256] **NORM2**: camada de normalização
- [13×13×384] **CONV3**: 384 filtros 3×3 passo 1, pad 1
- [13×13×384] **CONV4**: 384 filtros 3×3 passo 1, pad 1
- [13×13×256] **CONV5**: 256 filtros 3×3 passo 1, pad 1
- [6×6×256] **MAX POOL3**: filtros 3×3 passo 2
- [4096] **FC6**: 4096 neurônios
- [4096] **FC7**: 4096 neurônios
- [1000] **FC8**: 1000 neurônios (“scores” de classe)

# ConvNets – AlexNet

[Krizhevsky et al. 2012]

## Arquitetura da AlexNet:



- [227×227×3] Entrada
- [55×55×96] **CONV1**: 96 filtros 11×11 passo 4, pad 0
- [27×27×96] **MAX POOL1**: 96 filtros 3×3 passo 2
- [27×27×96] **NORM1**: camada de normalização
- [27×27×256] **CONV2**: 256 filtros 5×5 passo 1, pad 2
- [13×13×256] **MAX POOL2**: 256 filtros 3×3 passo 2
- [13×13×256] **NORM2**: camada de normalização
- [13×13×384] **CONV3**: 384 filtros 3×3 passo 1, pad 1
- [13×13×384] **CONV4**: 384 filtros 3×3 passo 1, pad 1
- [13×13×256] **CONV5**: 256 filtros 3×3 passo 1, pad 1
- [6×6×256] **MAX POOL3**: filtros 3×3 passo 2
- [4096] **FC6**: 4096 neurônios
- [4096] **FC7**: 4096 neurônios
- [1000] **FC8**: 1000 neurônios (“scores” de classe)

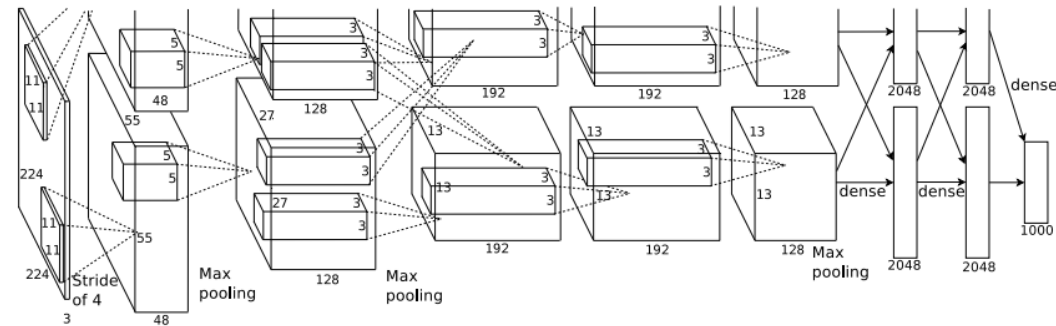
## Detalhes:

- Primeiro uso de ReLU

# ConvNets – AlexNet

[Krizhevsky et al. 2012]

## Arquitetura da AlexNet:



[227×227×3] Entrada  
[55×55×96] **CONV1**: 96 filtros 11×11 passo 4, pad 0  
[27×27×96] **MAX POOL1**: 96 filtros 3×3 passo 2  
[27×27×96] **NORM1**: camada de normalização  
[27×27×256] **CONV2**: 256 filtros 5×5 passo 1, pad 2  
[13×13×256] **MAX POOL2**: 256 filtros 3×3 passo 2  
[13×13×256] **NORM2**: camada de normalização  
[13×13×384] **CONV3**: 384 filtros 3×3 passo 1, pad 1  
[13×13×384] **CONV4**: 384 filtros 3×3 passo 1, pad 1  
[13×13×256] **CONV5**: 256 filtros 3×3 passo 1, pad 1  
[6×6×256] **MAX POOL3**: filtros 3×3 passo 2  
[4096] **FC6**: 4096 neurônios  
[4096] **FC7**: 4096 neurônios  
[1000] **FC8**: 1000 neurônios (“scores” de classe)

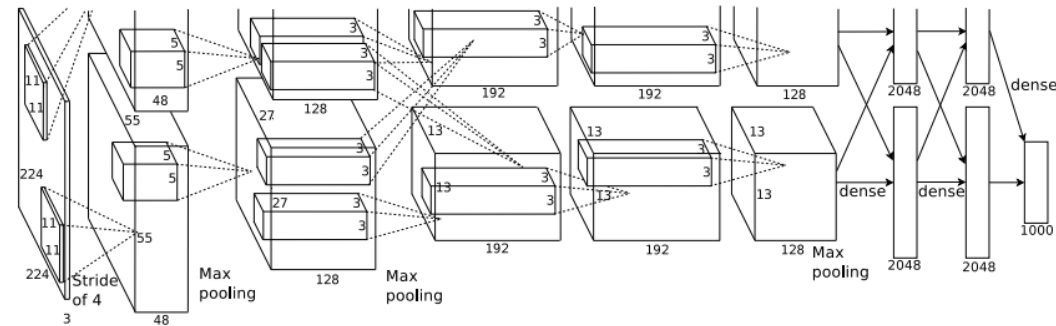
## Detalhes:

- Primeiro uso de ReLU
- Uso de camada de normalização (desuso)

# ConvNets – AlexNet

[Krizhevsky et al. 2012]

## Arquitetura da AlexNet:



[ $227 \times 227 \times 3$ ] Entrada  
[ $55 \times 55 \times 96$ ] **CONV1**: 96 filtros  $11 \times 11$  passo 4, pad 0  
[ $27 \times 27 \times 96$ ] **MAX POOL1**: 96 filtros  $3 \times 3$  passo 2  
[ $27 \times 27 \times 96$ ] **NORM1**: camada de normalização  
[ $27 \times 27 \times 256$ ] **CONV2**: 256 filtros  $5 \times 5$  passo 1, pad 2  
[ $13 \times 13 \times 256$ ] **MAX POOL2**: 256 filtros  $3 \times 3$  passo 2  
[ $13 \times 13 \times 256$ ] **NORM2**: camada de normalização  
[ $13 \times 13 \times 384$ ] **CONV3**: 384 filtros  $3 \times 3$  passo 1, pad 1  
[ $13 \times 13 \times 384$ ] **CONV4**: 384 filtros  $3 \times 3$  passo 1, pad 1  
[ $13 \times 13 \times 256$ ] **CONV5**: 256 filtros  $3 \times 3$  passo 1, pad 1  
[ $6 \times 6 \times 256$ ] **MAX POOL3**: filtros  $3 \times 3$  passo 2  
[4096] **FC6**: 4096 neurônios  
[4096] **FC7**: 4096 neurônios  
[1000] **FC8**: 1000 neurônios (“scores” de classe)

## Detalhes:

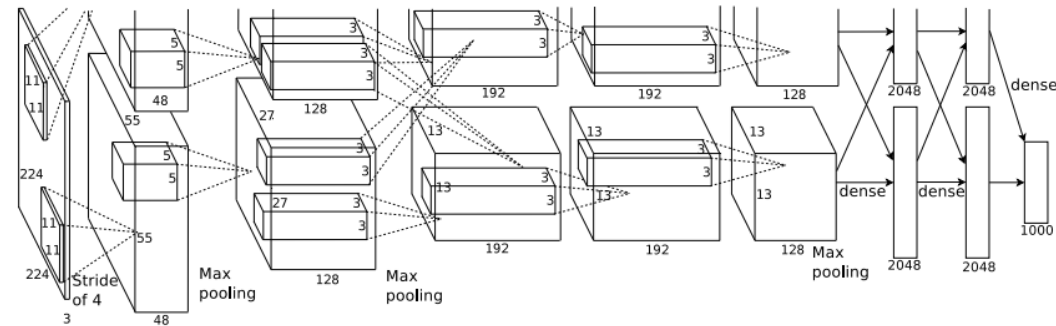
- Primeiro uso de ReLU
- Uso de camada de normalização (desuso)
- Grande uso de “data augmentation”



# ConvNets – AlexNet

[Krizhevsky et al. 2012]

## Arquitetura da AlexNet:



[227×227×3] Entrada  
[55×55×96] **CONV1**: 96 filtros 11×11 passo 4, pad 0  
[27×27×96] **MAX POOL1**: 96 filtros 3×3 passo 2  
[27×27×96] **NORM1**: camada de normalização  
[27×27×256] **CONV2**: 256 filtros 5×5 passo 1, pad 2  
[13×13×256] **MAX POOL2**: 256 filtros 3×3 passo 2  
[13×13×256] **NORM2**: camada de normalização  
[13×13×384] **CONV3**: 384 filtros 3×3 passo 1, pad 1  
[13×13×384] **CONV4**: 384 filtros 3×3 passo 1, pad 1  
[13×13×256] **CONV5**: 256 filtros 3×3 passo 1, pad 1  
[6×6×256] **MAX POOL3**: filtros 3×3 passo 2  
[4096] **FC6**: 4096 neurônios  
[4096] **FC7**: 4096 neurônios  
[1000] **FC8**: 1000 neurônios (“scores” de classe)

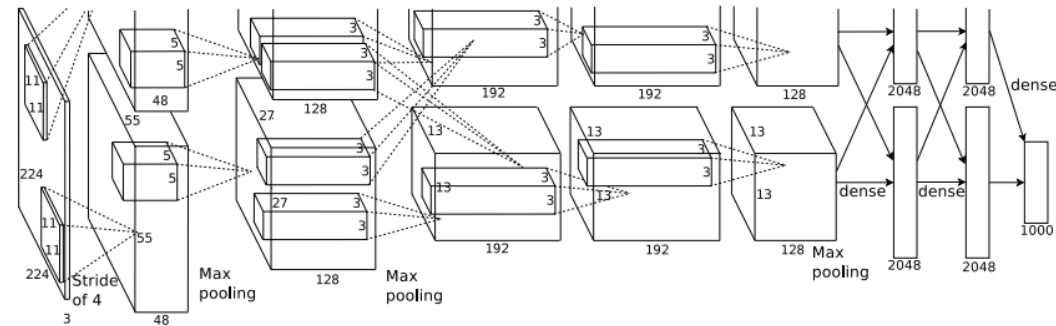
## Detalhes:

- Primeiro uso de ReLU
- Uso de camada de normalização (desuso)
- Grande uso de “data augmentation”
- Tamanho de *minibatch* = 128

# ConvNets – AlexNet

[Krizhevsky et al. 2012]

## Arquitetura da AlexNet:



[227×227×3] Entrada  
[55×55×96] **CONV1**: 96 filtros 11×11 passo 4, pad 0  
[27×27×96] **MAX POOL1**: 96 filtros 3×3 passo 2  
[27×27×96] **NORM1**: camada de normalização  
[27×27×256] **CONV2**: 256 filtros 5×5 passo 1, pad 2  
[13×13×256] **MAX POOL2**: 256 filtros 3×3 passo 2  
[13×13×256] **NORM2**: camada de normalização  
[13×13×384] **CONV3**: 384 filtros 3×3 passo 1, pad 1  
[13×13×384] **CONV4**: 384 filtros 3×3 passo 1, pad 1  
[13×13×256] **CONV5**: 256 filtros 3×3 passo 1, pad 1  
[6×6×256] **MAX POOL3**: filtros 3×3 passo 2  
[4096] **FC6**: 4096 neurônios  
[4096] **FC7**: 4096 neurônios  
[1000] **FC8**: 1000 neurônios (“scores” de classe)

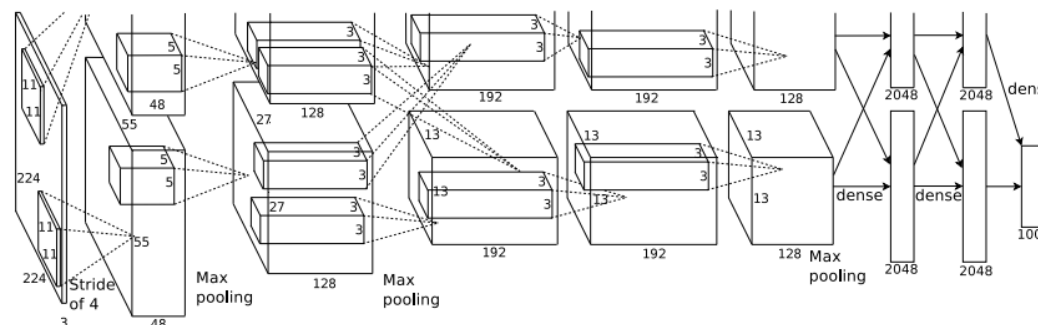
## Detalhes:

- Primeiro uso de ReLU
- Uso de camada de normalização (desuso)
- Grande uso de “data augmentation”
- Tamanho de *minibatch* = 128
- *SGD+Momentum* com constante de 0,9

# ConvNets – AlexNet

[Krizhevsky et al. 2012]

## Arquitetura da AlexNet:



[227×227×3] Entrada  
[55×55×96] **CONV1**: 96 filtros 11×11 passo 4, pad 0  
[27×27×96] **MAX POOL1**: 96 filtros 3×3 passo 2  
[27×27×96] **NORM1**: camada de normalização  
[27×27×256] **CONV2**: 256 filtros 5×5 passo 1, pad 2  
[13×13×256] **MAX POOL2**: 256 filtros 3×3 passo 2  
[13×13×256] **NORM2**: camada de normalização  
[13×13×384] **CONV3**: 384 filtros 3×3 passo 1, pad 1  
[13×13×384] **CONV4**: 384 filtros 3×3 passo 1, pad 1  
[13×13×256] **CONV5**: 256 filtros 3×3 passo 1, pad 1  
[6×6×256] **MAX POOL3**: filtros 3×3 passo 2  
[4096] **FC6**: 4096 neurônios  
[4096] **FC7**: 4096 neurônios  
[1000] **FC8**: 1000 neurônios (“scores” de classe)

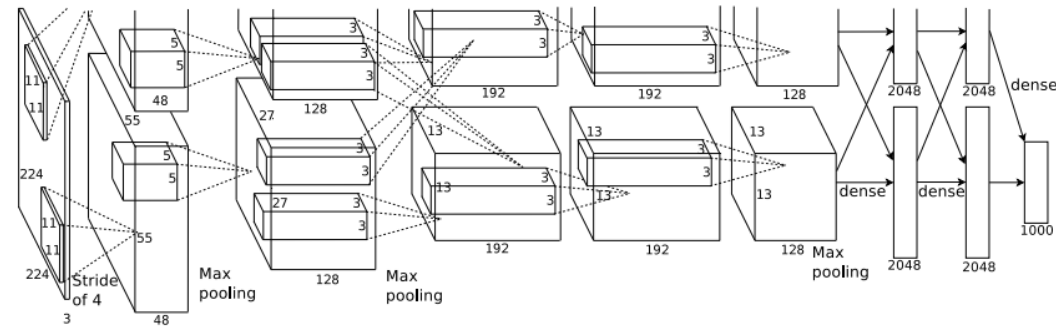
## Detalhes:

- Primeiro uso de ReLU
- Uso de camada de normalização (desuso)
- Grande uso de “data augmentation”
- Tamanho de *minibatch* = 128
- SGD+*Momentum* com constante de 0,9
- Tx. Aprendizado =  $10^{-2}$  e reduzida manualmente em “plateaus”

# ConvNets – AlexNet

[Krizhevsky et al. 2012]

## Arquitetura da AlexNet:



[227×227×3] Entrada  
[55×55×96] **CONV1**: 96 filtros 11×11 passo 4, pad 0  
[27×27×96] **MAX POOL1**: 96 filtros 3×3 passo 2  
[27×27×96] **NORM1**: camada de normalização  
[27×27×256] **CONV2**: 256 filtros 5×5 passo 1, pad 2  
[13×13×256] **MAX POOL2**: 256 filtros 3×3 passo 2  
[13×13×256] **NORM2**: camada de normalização  
[13×13×384] **CONV3**: 384 filtros 3×3 passo 1, pad 1  
[13×13×384] **CONV4**: 384 filtros 3×3 passo 1, pad 1  
[13×13×256] **CONV5**: 256 filtros 3×3 passo 1, pad 1  
[6×6×256] **MAX POOL3**: filtros 3×3 passo 2  
[4096] **FC6**: 4096 neurônios  
[4096] **FC7**: 4096 neurônios  
[1000] **FC8**: 1000 neurônios (“scores” de classe)

## Detalhes:

- Primeiro uso de ReLU
- Uso de camada de normalização (desuso)
- Grande uso de “data augmentation”
- Tamanho de *minibatch* = 128
- SGD+*Momentum* com constante de 0,9
- Tx. Aprendizado =  $10^{-2}$  e reduzida manualmente em “plateaus”
- “Ensemble” de 7 CNNs: 18,2% → 15,4%

# ConvNets – VGG

[Simonyan and Zisserman, 2014]

Uso de filtros menores em redes mais profundas



# ConvNets – VGG

[Simonyan and Zisserman, 2014]

Uso de filtros menores em redes mais profundas



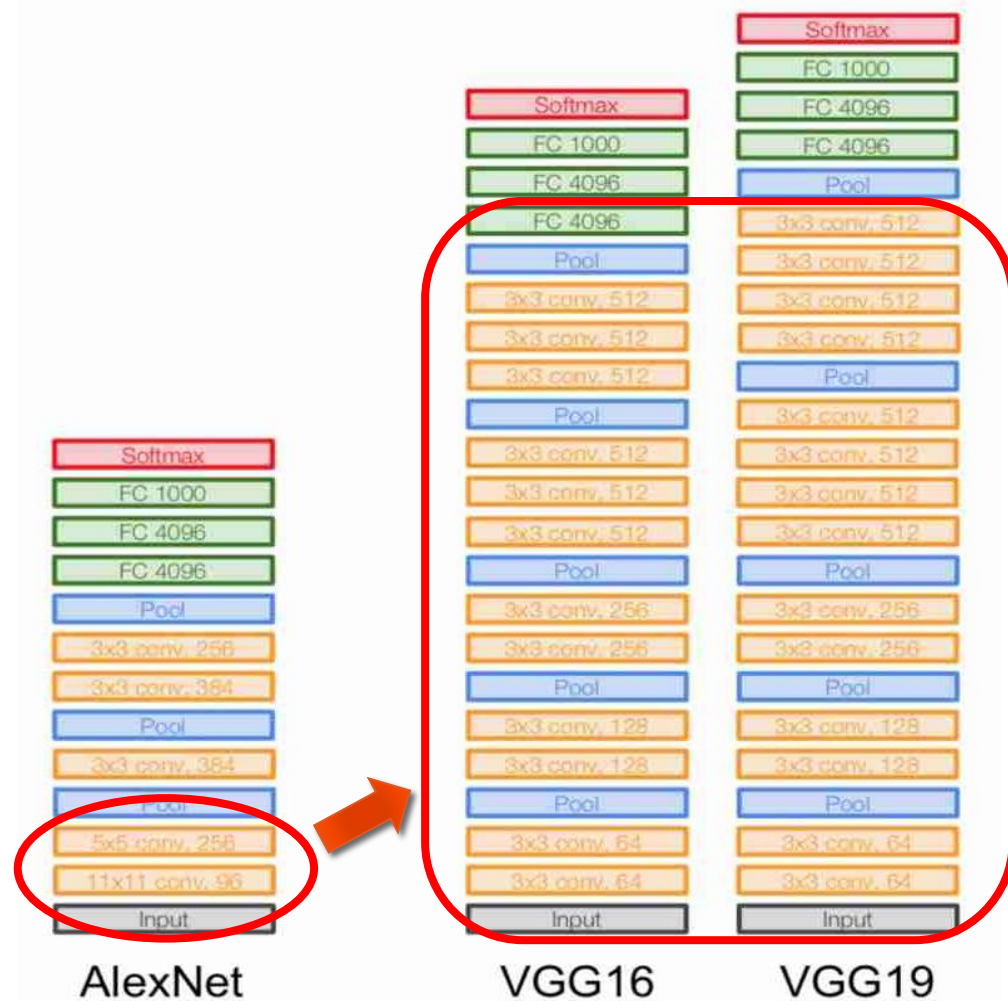


# ConvNets – VGG

**[Simonyan and Zisserman, 2014]**

## Uso de filtros menores em redes mais profundas

Apenas CONV 3×3 passo 1, pad 1  
e  
MAX POOL 2×2 passo 2



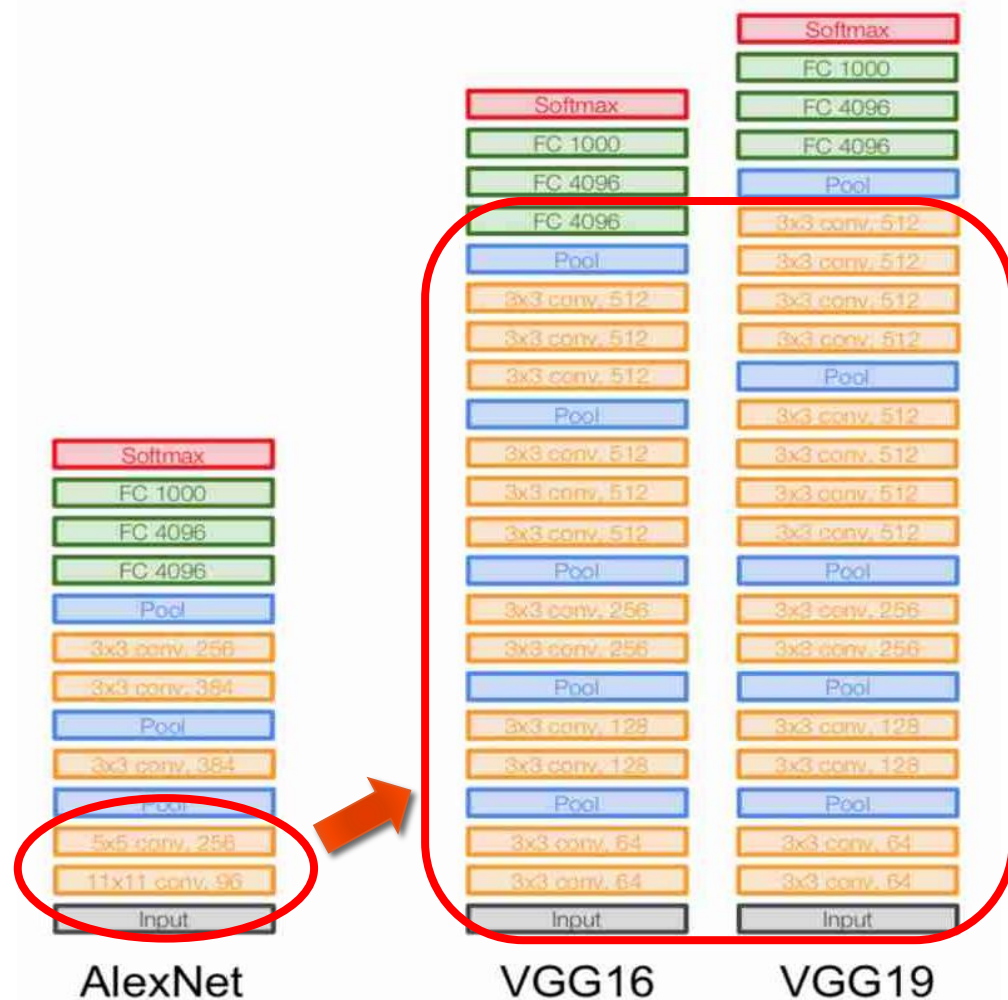
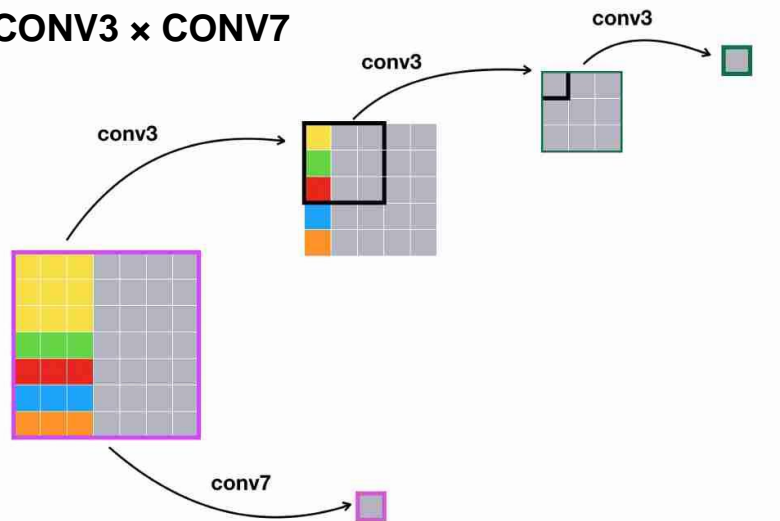
# ConvNets – VGG

[Simonyan and Zisserman, 2014]

Uso de filtros menores em redes mais profundas

Apenas CONV 3×3 passo 1, pad 1  
e  
MAX POOL 2×2 passo 2

CONV3 × CONV7





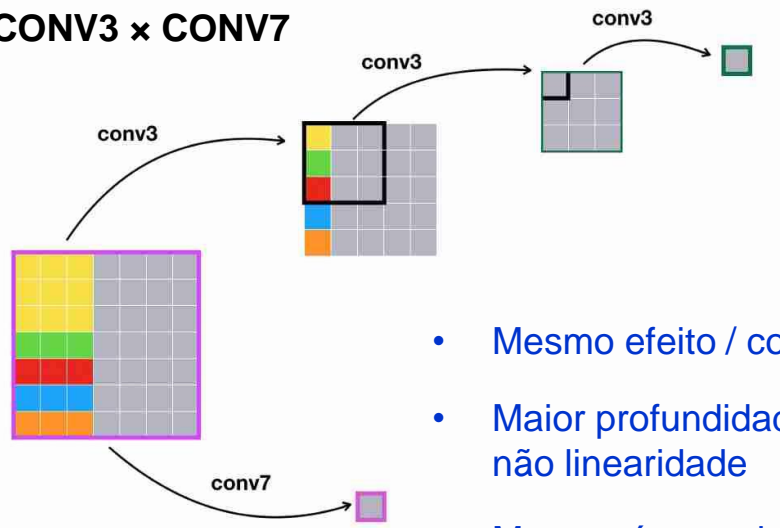
# ConvNets – VGG

[Simonyan and Zisserman, 2014]

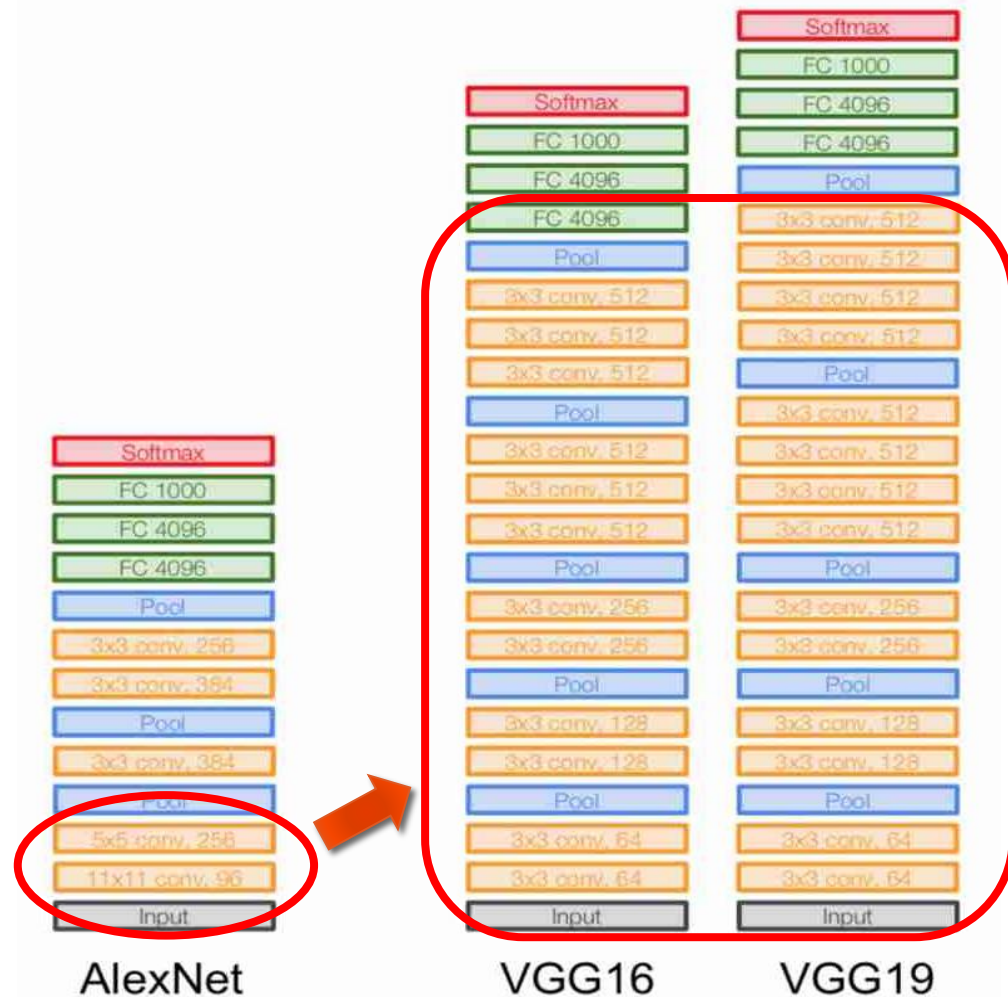
Uso de filtros menores em redes mais profundas

Apenas CONV 3×3 passo 1, pad 1  
e  
MAX POOL 2×2 passo 2

CONV3 × CONV7



- Mesmo efeito / cobertura
- Maior profundidade  $\Rightarrow$  mais não linearidade
- Menor número de parâmetros



# ConvNets – VGG

[Simonyan and Zisserman, 2014]

Uso de filtros menores em redes mais profundas

Apenas CONV 3×3 passo 1, pad 1  
e  
MAX POOL 2×2 passo 2

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 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	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 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

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

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

# ConvNets – VGG

[Simonyan and Zisserman, 2014]

Uso de filtros menores em redes mais profundas

Apenas CONV 3×3 passo 1, pad 1  
e  
MAX POOL 2×2 passo 2

Melhor modelo

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 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	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 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: Number of parameters (in millions).

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

# ConvNets – VGG

[Simonyan and Zisserman, 2014]

Uso de filtros menores em redes mais profundas

Apenas CONV 3×3 passo 1, pad 1  
e  
MAX POOL 2×2 passo 2

Melhor modelo

Erro (top 5) = 11,2% - ILSVRC 2013



Erro (top 5) = 7,3%

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 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	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 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: Number of parameters (in millions).

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



# ConvNets – VGG

[Simonyan and Zisserman, 2014]

Uso de filtros menores em redes mais profundas

Apenas CONV 3×3 passo 1, pad 1  
e  
MAX POOL 2×2 passo 2

Melhor modelo

Erro (top 5) = 11,2% - ILSVRC 2013



Erro (top 5) = 7,3%

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 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	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 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: Number of parameters (in millions)

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

138M

# ConvNets – VGG

ENTRADA: [224x224x3] **memória:**  $224*224*3=150K$  **params:** 0 (sem contar vieses)

CONV3-64: [224x224x64] **memória:**  $224*224*64=3.2M$  **params:**  $(3*3*3)*64 = 1.728$

CONV3-64: [224x224x64] **memória:**  $224*224*64=3.2M$  **params:**  $(3*3*64)*64 = 36.864$

POOL2: [112x112x64] **memória:**  $112*112*64=800K$  **params:** 0

CONV3-128: [112x112x128] **memória:**  $112*112*128=1.6M$  **params:**  $(3*3*64)*128 = 73.728$

CONV3-128: [112x112x128] **memória:**  $112*112*128=1.6M$  **params:**  $(3*3*128)*128 = 147.456$

POOL2: [56x56x128] **memória:**  $56*56*128=400K$  **params:** 0

CONV3-256: [56x56x256] **memória:**  $56*56*256=800K$  **params:**  $(3*3*128)*256 = 294.912$

CONV3-256: [56x56x256] **memória:**  $56*56*256=800K$  **params:**  $(3*3*256)*256 = 589.824$

CONV3-256: [56x56x256] **memória:**  $56*56*256=800K$  **params:**  $(3*3*256)*256 = 589.824$

POOL2: [28x28x256] **memória:**  $28*28*256=200K$  **params:** 0

CONV3-512: [28x28x512] **memória:**  $28*28*512=400K$  **params:**  $(3*3*256)*512 = 1.179.648$

CONV3-512: [28x28x512] **memória:**  $28*28*512=400K$  **params:**  $(3*3*512)*512 = 2.359.296$

CONV3-512: [28x28x512] **memória:**  $28*28*512=400K$  **params:**  $(3*3*512)*512 = 2.359.296$

POOL2: [14x14x512] **memória:**  $14*14*512=100K$  **params:** 0

CONV3-512: [14x14x512] **memória:**  $14*14*512=100K$  **params:**  $(3*3*512)*512 = 2.359.296$

CONV3-512: [14x14x512] **memória:**  $14*14*512=100K$  **params:**  $(3*3*512)*512 = 2.359.296$

CONV3-512: [14x14x512] **memória:**  $14*14*512=100K$  **params:**  $(3*3*512)*512 = 2.359.296$

POOL2: [7x7x512] **memória:**  $7*7*512=25K$  **params:** 0

FC: [1x1x4096] **memória:** 4096 **params:**  $7*7*512*4096 = 102.760.448$

FC: [1x1x4096] **memória:** 4096 **params:**  $4096*4096 = 16.777.216$

FC: [1x1x1000] **memória:** 1000 **params:**  $4096*1000 = 4.096.000$

[Simonyan and Zisserman, 2014]

ConvNet Configuration			
B	C	D	E
13 weight layers	16 weight layers	16 weight layers	19 weight layers
Input (224 × 224 RGB image)			
conv3-64	conv3-64	conv3-64	conv3-64
<b>conv3-64</b>	conv3-64	conv3-64	<b>conv3-64</b>
maxpool			
conv3-128	conv3-128	conv3-128	conv3-128
<b>conv3-128</b>	conv3-128	conv3-128	<b>conv3-128</b>
maxpool			
conv3-256	conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256	conv3-256
	<b>conv1-256</b>	<b>conv3-256</b>	<b>conv3-256</b>
maxpool			
conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512
	<b>conv1-512</b>	<b>conv3-512</b>	<b>conv3-512</b>
maxpool			
conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512
	<b>conv1-512</b>	<b>conv3-512</b>	<b>conv3-512</b>
maxpool			
FC-4096			
FC-4096			
FC-1000			
soft-max			

# ConvNets – VGG

ENTRADA: [224x224x3] **memória:**  $224*224*3=150K$  **params:** 0 (sem contar vieses)

CONV3-64: [224x224x64] **memória:**  $224*224*64=3.2M$  **params:**  $(3*3*3)*64 = 1.728$

CONV3-64: [224x224x64] **memória:**  $224*224*64=3.2M$  **params:**  $(3*3*64)*64 = 36.864$

POOL2: [112x112x64] **memória:**  $112*112*64=800K$  **params:** 0

CONV3-128: [112x112x128] **memória:**  $112*112*128=1.6M$  **params:**  $(3*3*64)*128 = 73.728$

CONV3-128: [112x112x128] **memória:**  $112*112*128=1.6M$  **params:**  $(3*3*128)*128 = 147.456$

POOL2: [56x56x128] **memória:**  $56*56*128=400K$  **params:** 0

CONV3-256: [56x56x256] **memória:**  $56*56*256=800K$  **params:**  $(3*3*128)*256 = 294.912$

CONV3-256: [56x56x256] **memória:**  $56*56*256=800K$  **params:**  $(3*3*256)*256 = 589.824$

CONV3-256: [56x56x256] **memória:**  $56*56*256=800K$  **params:**  $(3*3*256)*256 = 589.824$

POOL2: [28x28x256] **memória:**  $28*28*256=200K$  **params:** 0

CONV3-512: [28x28x512] **memória:**  $28*28*512=400K$  **params:**  $(3*3*256)*512 = 1.179.648$

CONV3-512: [28x28x512] **memória:**  $28*28*512=400K$  **params:**  $(3*3*512)*512 = 2.359.296$

CONV3-512: [28x28x512] **memória:**  $28*28*512=400K$  **params:**  $(3*3*512)*512 = 2.359.296$

POOL2: [14x14x512] **memória:**  $14*14*512=100K$  **params:** 0

CONV3-512: [14x14x512] **memória:**  $14*14*512=100K$  **params:**  $(3*3*512)*512 = 2.359.296$

CONV3-512: [14x14x512] **memória:**  $14*14*512=100K$  **params:**  $(3*3*512)*512 = 2.359.296$

CONV3-512: [14x14x512] **memória:**  $14*14*512=100K$  **params:**  $(3*3*512)*512 = 2.359.296$

POOL2: [7x7x512] **memória:**  $7*7*512=25K$  **params:** 0

FC: [1x1x4096] **memória:** 4096 **params:**  $7*7*512*4096 = 102.760.448$

FC: [1x1x4096] **memória:** 4096 **params:**  $4096*4096 = 16.777.216$

FC: [1x1x1000] **memória:** 1000 **params:**  $4096*1000 = 4.096.000$

**TOTAL memória:** 24M \* 4 bytes  $\sim$  93MB / imagem (apenas *forward*!  $\sim$  x2 para *backward*)

**TOTAL params:** 138M parâmetros

[Simonyan and Zisserman, 2014]

ConvNet Configuration			
B	C	D	E
13 weight layers	16 weight layers	16 weight layers	19 weight layers
Input (224 × 224 RGB image)			
conv3-64	conv3-64	conv3-64	conv3-64
<b>conv3-64</b>	conv3-64	conv3-64	<b>conv3-64</b>
maxpool			
conv3-128	conv3-128	conv3-128	conv3-128
<b>conv3-128</b>	conv3-128	conv3-128	<b>conv3-128</b>
maxpool			
conv3-256	conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256	conv3-256
	<b>conv1-256</b>	<b>conv3-256</b>	<b>conv3-256</b>
maxpool			
conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512
	<b>conv1-512</b>	<b>conv3-512</b>	<b>conv3-512</b>
maxpool			
conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512
	<b>conv1-512</b>	<b>conv3-512</b>	<b>conv3-512</b>
maxpool			
FC-4096			
FC-4096			
FC-1000			
soft-max			

# ConvNets – VGG

[Simonyan and Zisserman, 2014]

ENTRADA: [224x224x3] memória:  $224*224*3=150K$  params: 0 (sem contar vieses)  
CONV3-64: [224x224x64] memória:  $224*224*64=3.2M$  params:  $(3*3*3)*64 = 1.728$   
CONV3-64: [224x224x64] memória:  $224*224*64=3.2M$  params:  $(3*3*64)*64 = 36.864$   
POOL2: [112x112x64] memória:  $112*112*64=800K$  params: 0  
CONV3-128: [112x112x128] memória:  $112*112*128=1.6M$  params:  $(3*3*64)*128 = 73.728$   
CONV3-128: [112x112x128] memória:  $112*112*128=1.6M$  params:  $(3*3*128)*128 = 147.456$   
POOL2: [56x56x128] memória:  $56*56*128=400K$  params: 0  
CONV3-256: [56x56x256] memória:  $56*56*256=800K$  params:  $(3*3*128)*256 = 294.912$   
CONV3-256: [56x56x256] memória:  $56*56*256=800K$  params:  $(3*3*256)*256 = 589.824$   
CONV3-256: [56x56x256] memória:  $56*56*256=800K$  params:  $(3*3*256)*256 = 589.824$   
POOL2: [28x28x256] memória:  $28*28*256=200K$  params: 0  
CONV3-512: [28x28x512] memória:  $28*28*512=400K$  params:  $(3*3*256)*512 = 1.179.648$   
CONV3-512: [28x28x512] memória:  $28*28*512=400K$  params:  $(3*3*512)*512 = 2.359.296$   
CONV3-512: [28x28x512] memória:  $28*28*512=400K$  params:  $(3*3*512)*512 = 2.359.296$   
POOL2: [14x14x512] memória:  $14*14*512=100K$  params: 0  
CONV3-512: [14x14x512] memória:  $14*14*512=100K$  params:  $(3*3*512)*512 = 2.359.296$   
CONV3-512: [14x14x512] memória:  $14*14*512=100K$  params:  $(3*3*512)*512 = 2.359.296$   
CONV3-512: [14x14x512] memória:  $14*14*512=100K$  params:  $(3*3*512)*512 = 2.359.296$   
POOL2: [7x7x512] memória:  $7*7*512=25K$  params: 0  
FC: [1x1x4096] memória: 4096 params:  $7*7*512*4096 = 102.760.448$   
FC: [1x1x4096] memória: 4096 params:  $4096*4096 = 16.777.216$   
FC: [1x1x1000] memória: 1000 params:  $4096*1000 = 4.096.000$

OBS:

Maior parte da memória está nas camadas CONV iniciais

Maioria dos params estão nas camadas FC finais

TOTAL memória:  $24M * 4 \text{ bytes} \approx 93MB$  / imagem (apenas *forward*! ~  $\times 2$  para *backward*)  
TOTAL params: 138M parâmetros