

Curso Inteligência Artificial: do Zero ao Infinito

Arquiteturas Baseadas em CNNs Parte 2

Universidade Federal de Mato Grosso

Agenda

- 1 Introdução
- 2 AlexNet
- 3 VGGNet
- 4 GoogLeNet
- 5 Vanishing Gradient Problem
- 6 ResNet
- 7 DenseNet

Agenda

1 Introdução

2 AlexNet

3 VGGNet

4 GoogLeNet

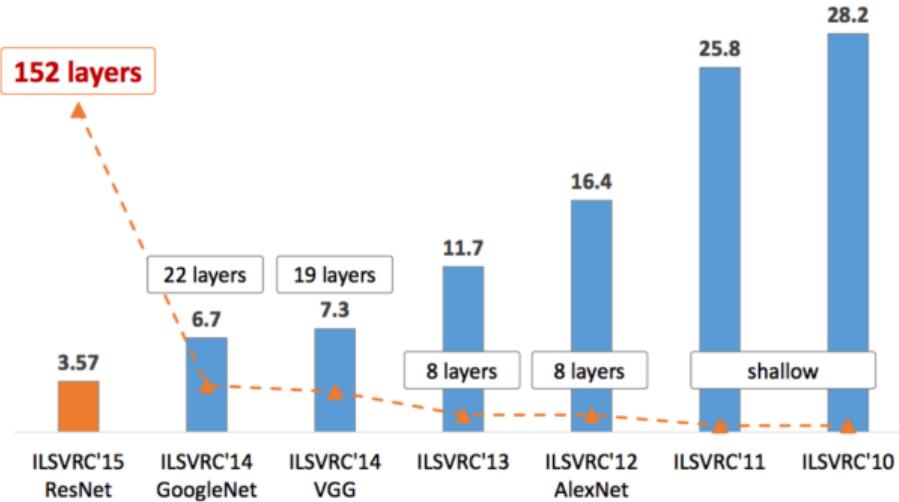
5 Vanishing Gradient Problem

6 ResNet

7 DenseNet

Introdução

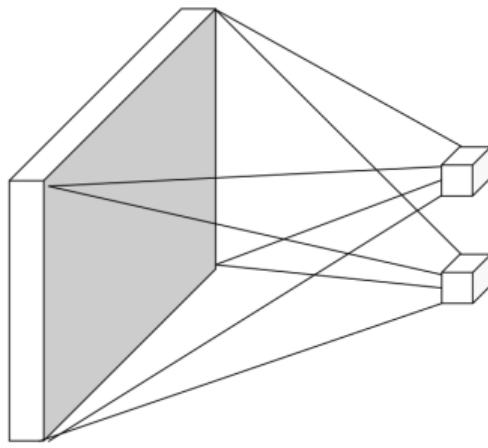
Relembrando...



Introdução

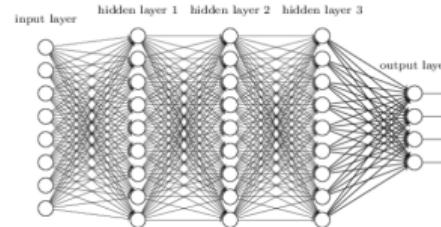
Convolução

Arquitetura antes do conceito de deep learning



image

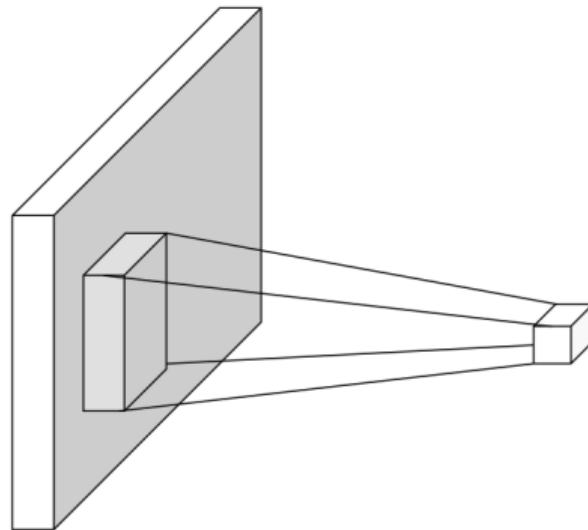
Fully connected layer



Introdução

Convolução

Arquitetura convolucional



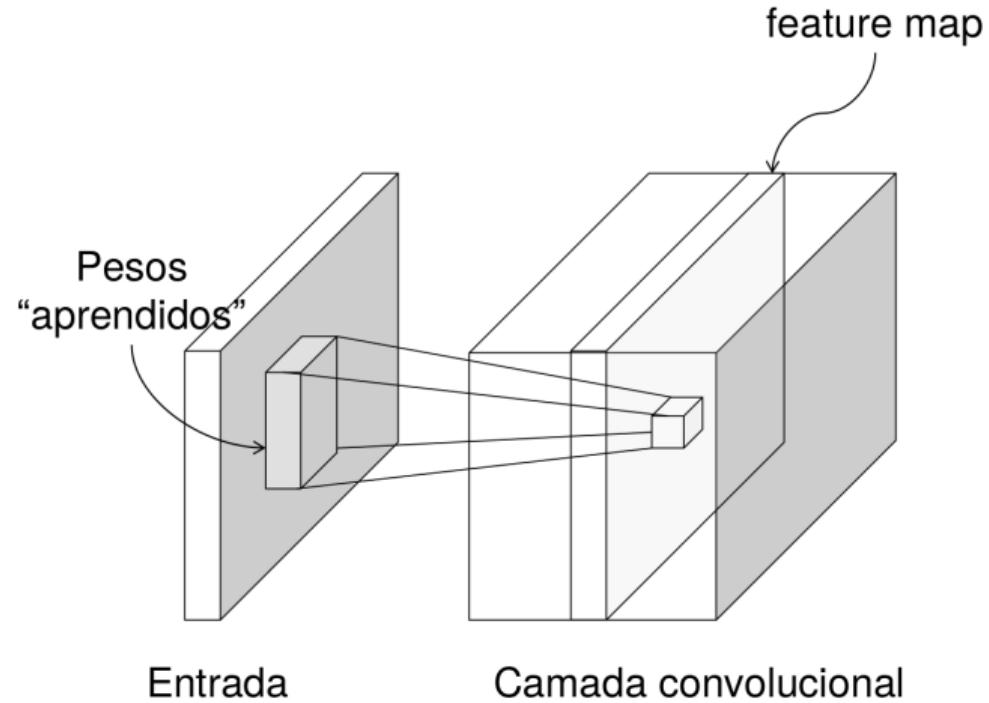
image

Convolutional layer

Introdução

Convolução

Arquitetura convolucional



Agenda

1 Introdução

2 AlexNet

3 VGGNet

4 GoogLeNet

5 Vanishing Gradient Problem

6 ResNet

7 DenseNet

Introdução

AlexNet

Relembrando as camadas:

- Entrada $224 \times 224 \times 3$
- Primeiro filtro $11 \times 11 \times 3$
- Filtros convolucionais menores a medida que aumenta a profundidade
- Ex. 5×5 , 3×3

O que acontece se usarmos apenas filtros pequenos? Por ex. 3×3 ?

Agenda

- 1 Introdução
- 2 AlexNet
- 3 VGGNet
- 4 GoogLeNet
- 5 Vanishing Gradient Problem
- 6 ResNet
- 7 DenseNet

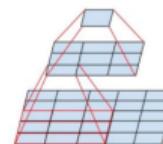
VGGNet

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
maxpool					
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 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 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 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					

Table 2: Number of parameters (in millions).

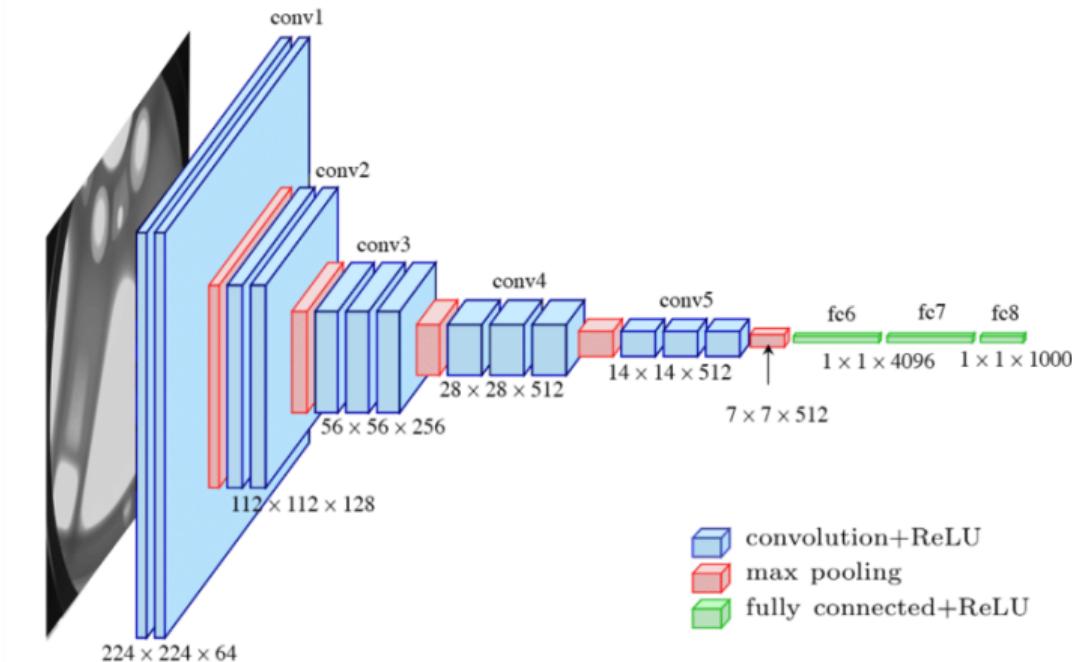
Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

- Sequence of deeper networks trained progressively
- Campos receptivos substituídos por filtros 3x3 (com ReLU entre eles)

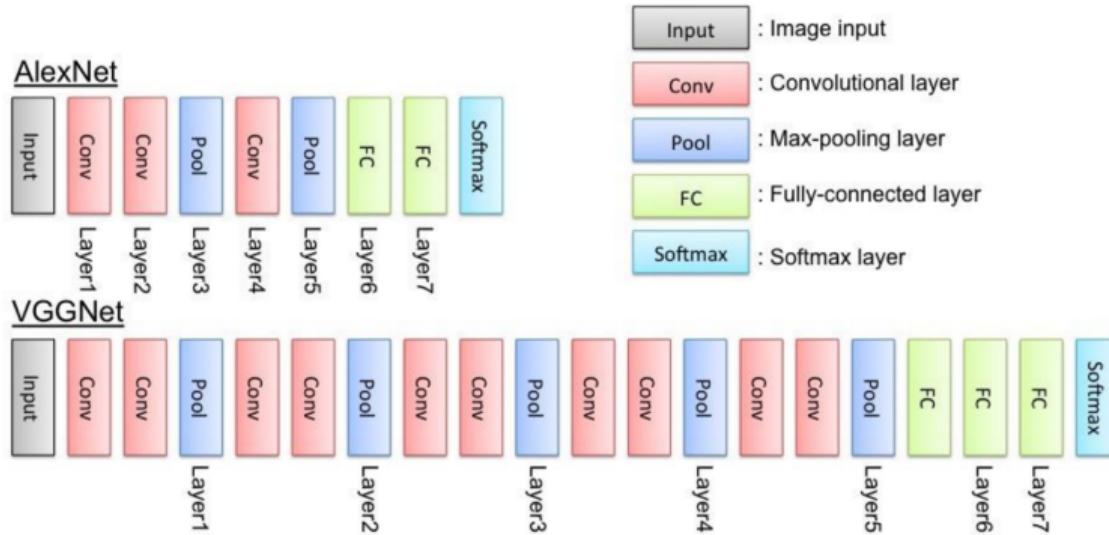


- Uma camada 7x7 com C mapas precisa de $49C^2$ pesos, camadas 3x3 precisam de $27C^2$ pesos

VGGNet



VGGNet





Agenda

1 Introdução

2 AlexNet

3 VGGNet

4 GoogLeNet

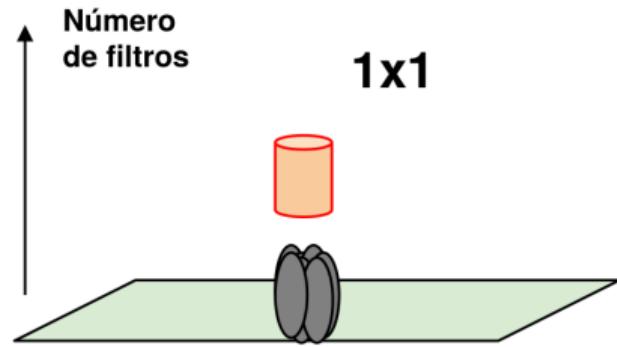
5 Vanishing Gradient Problem

6 ResNet

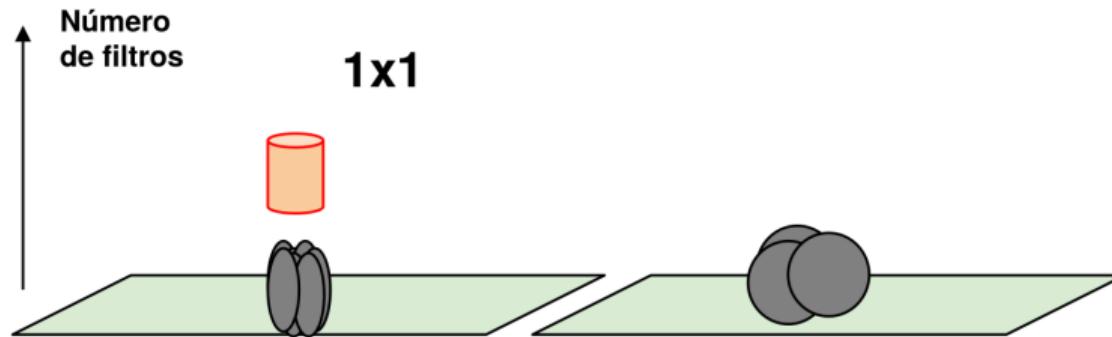
7 DenseNet

Conv 1×1

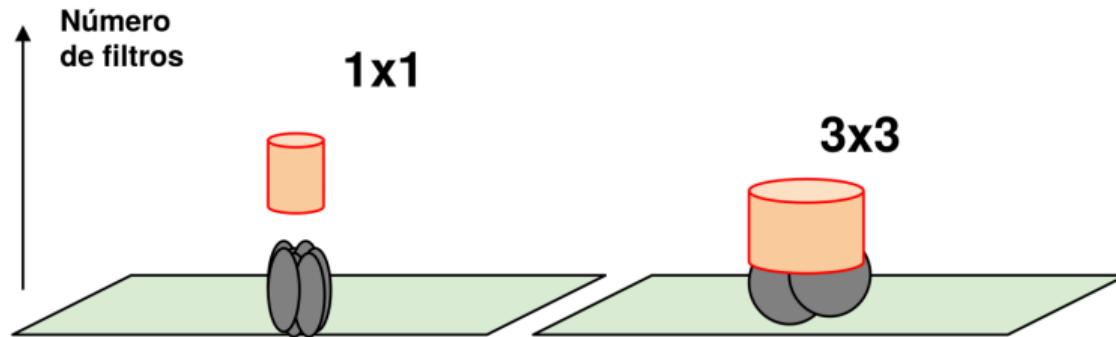
Ideia: cobrir clusters locais com convoluções $1x1$



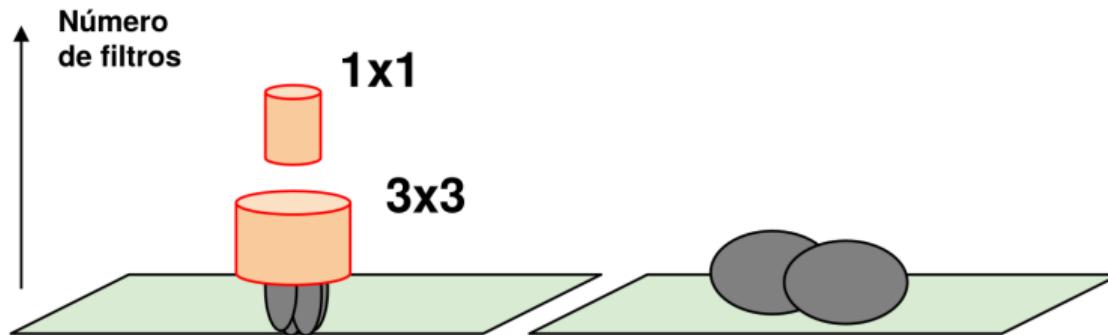
Correlações menos espalhadas



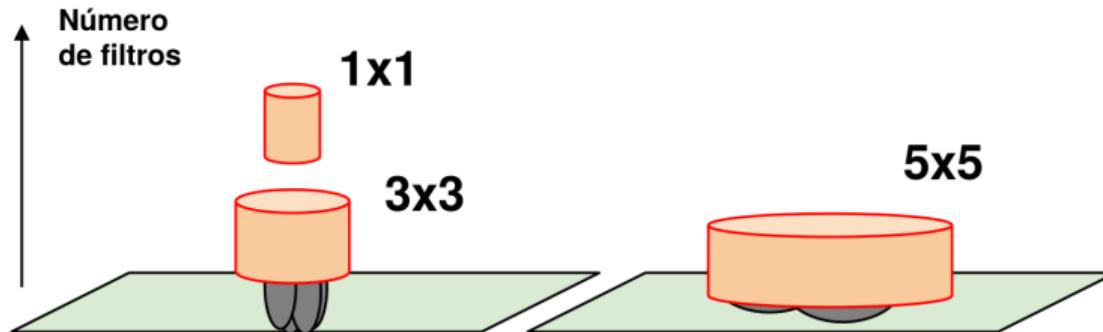
Correlações menos espalhadas



Correlações menos espalhadas

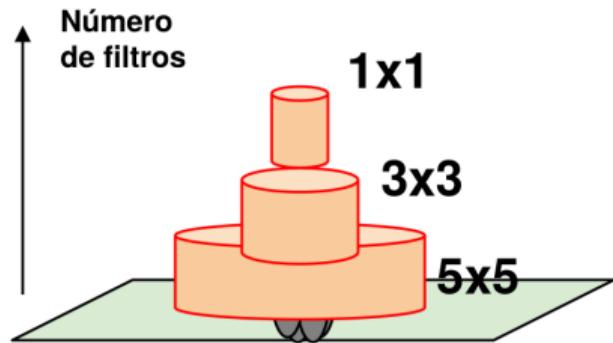


Correlações menos espalhadas



SLIDE CREDIT: GOOGLE INC

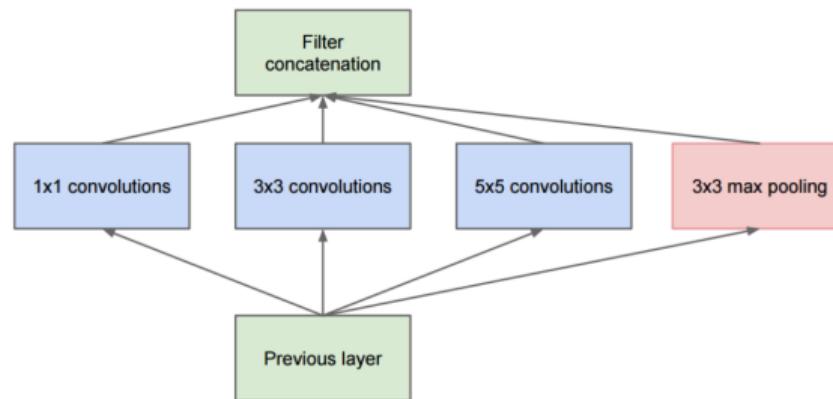
Conv 1×1



GoogLeNet

Inception Module v1

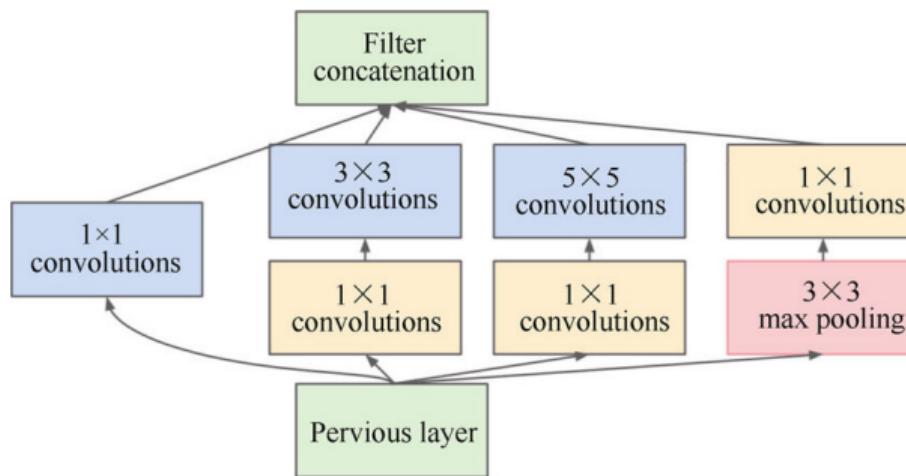
- E se inseríssemos todos os filtros em uma única camada?
- Filtros convolucionais em paralelo com campos receptivos diferentes.



GoogLeNet

Inception Module v1

- Usa filtros 1×1 para redução da dimensionalidade antes de operações de “alto custo” antes das convoluções maiores.

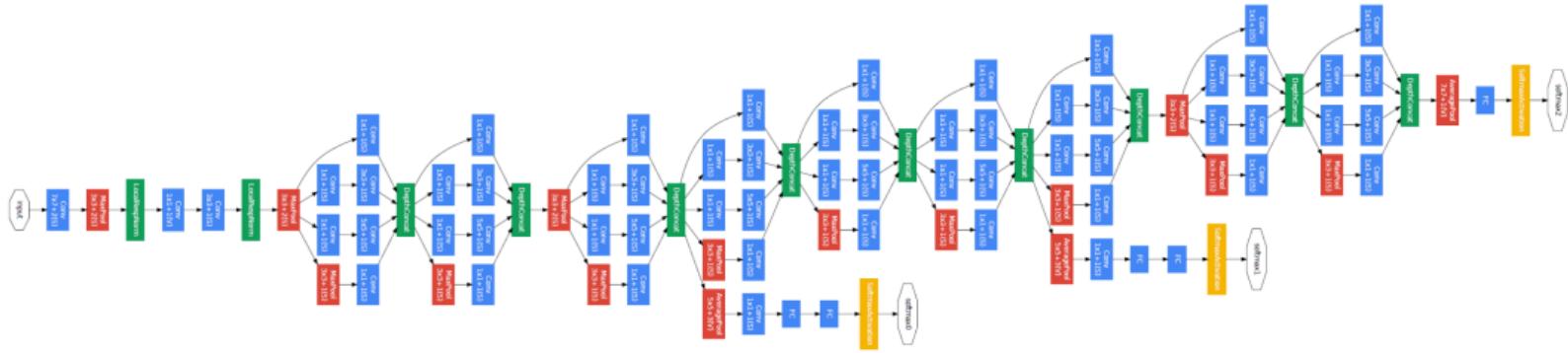


Convolução 1×1

- Reduz o esforço computacional antes de filtros 3×3 e 5×5 .
- Inclui o uso de ReLU.
- Exemplo:
 - ▶ Entrada de 256 mapas de filtros 56×56 .
 - ▶ Inserção de camada $1 \times 1 \times 64$.
 - ▶ $56 \times 56 \times 256 \rightarrow 56 \times 56 \times 64$ (redução de $4 \times$).

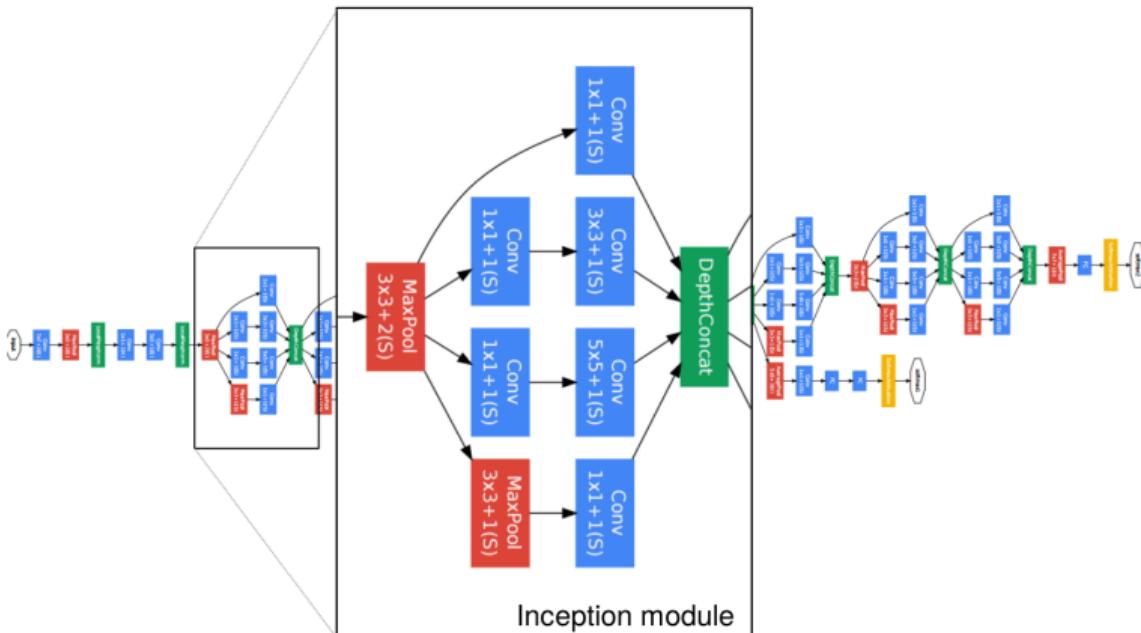
Fonte: One by One [1×1] Convolution - counter-intuitively useful

GoogLeNet

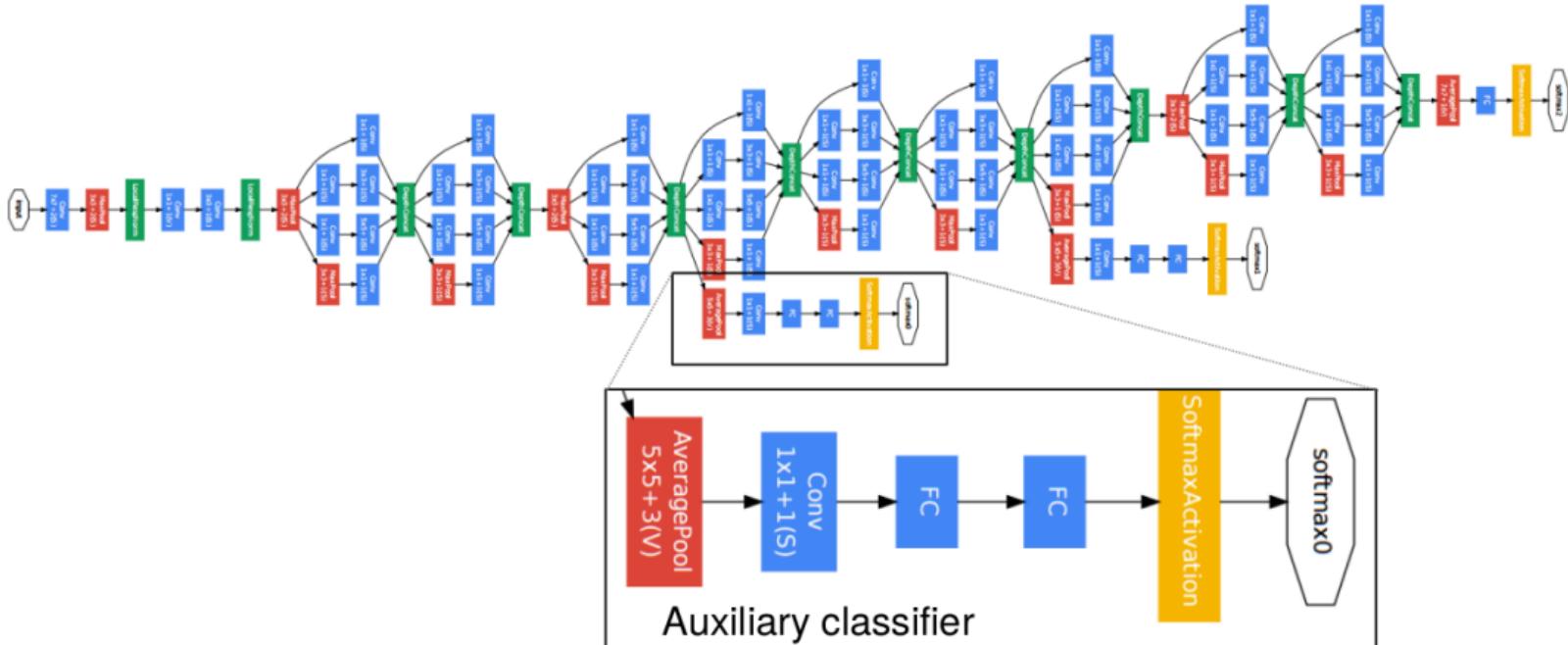


Paper: Going Deeper with Convolutions

GoogLeNet



GoogLeNet



GoogLeNet

Inception v2

- **Inception v2** introduziu a fatoração (fatorar convoluções em convoluções menores) e algumas pequenas alterações no *Inception v1* .
- As convoluções 7x7 tradicionais foram fatoradas em três convoluções 3x3.

Paper: Rethinking the Inception Architecture for Computer Vision

GoogLeNet

Inception v3

- **Inception v3** é uma variante do *Inception v2* que adiciona BN-auxiliar.
- BN auxiliar refere-se à versão em que a camada totalmente conectada do classificador auxiliar também é normalizada, não apenas convoluções.
- Portanto, o modelo *Inception-v3* trata-se do [Inception-v2 + BN auxiliar].

Paper: Rethinking the Inception Architecture for Computer Vision

Winning



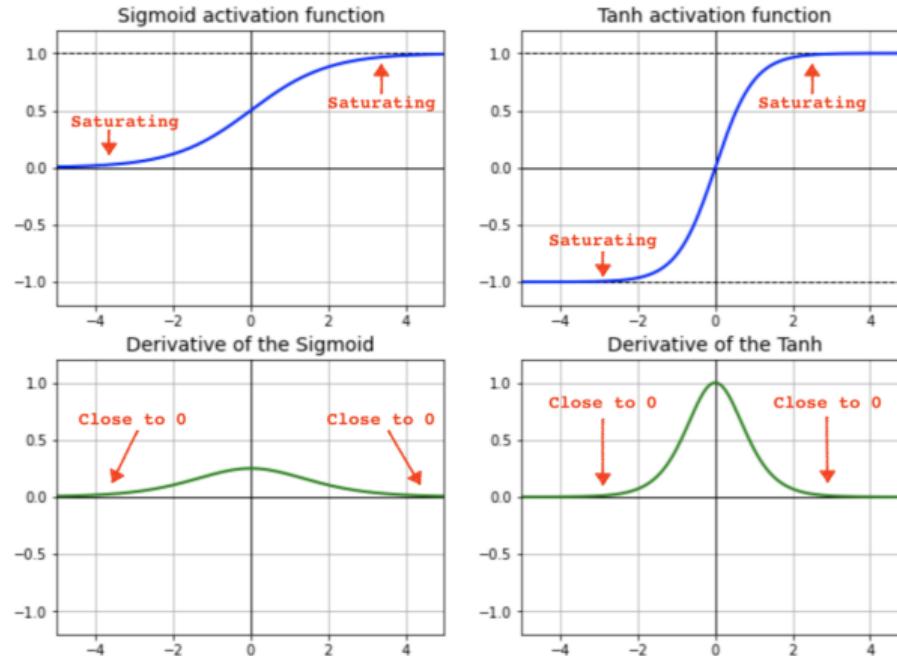
Agenda

- 1 Introdução
- 2 AlexNet
- 3 VGGNet
- 4 GoogLeNet
- 5 Vanishing Gradient Problem
- 6 ResNet
- 7 DenseNet

Vanishing Gradient Problem

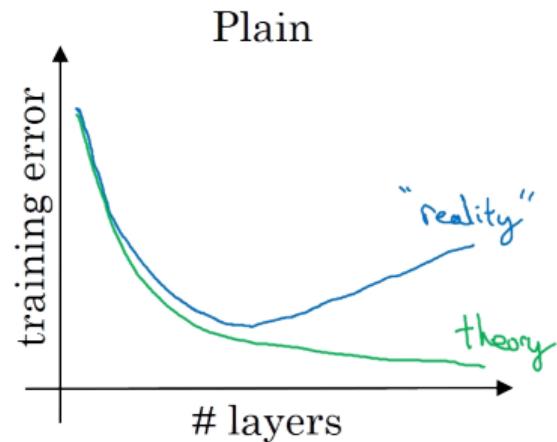
- Quanto mais camadas se adiciona em uma rede, menor ficam os gradientes utilizados para atualização dos pesos durante a etapa de *back propagation*.
- Isso dificulta o treinamento de redes neurais muito profundas, contrário ao que se imaginava.
- Esse problema é chamado de **Vanishing Gradient Problem**

Vanishing Gradient Problem



Fonte: Why Rectified Linear Unit (ReLU) in Deep Learning and the best practice to use it with TensorFlow

Vanishing Gradient Problem



Fonte: DeepLearningAI - C4W2L03 Resnets - Andrew Ng

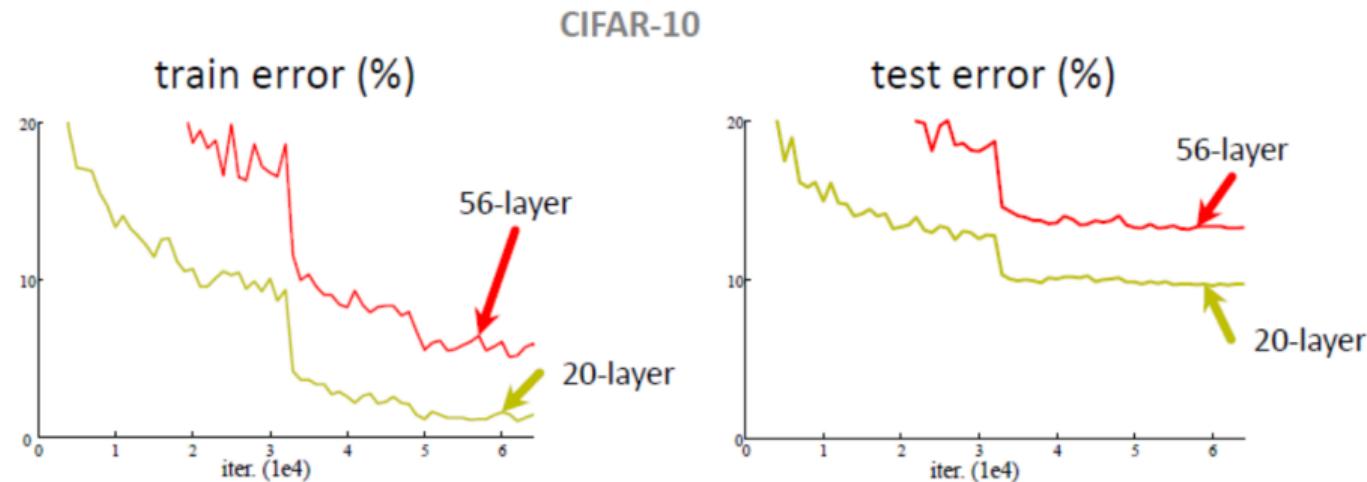
Agenda

- 1 Introdução
- 2 AlexNet
- 3 VGGNet
- 4 GoogLeNet
- 5 Vanishing Gradient Problem
- 6 ResNet
- 7 DenseNet

ResNet

Problema

Empilhar convoluções 3×3 funcionava bem até 20/30 camadas.



Paper: Deep Residual Learning for Image Recognition

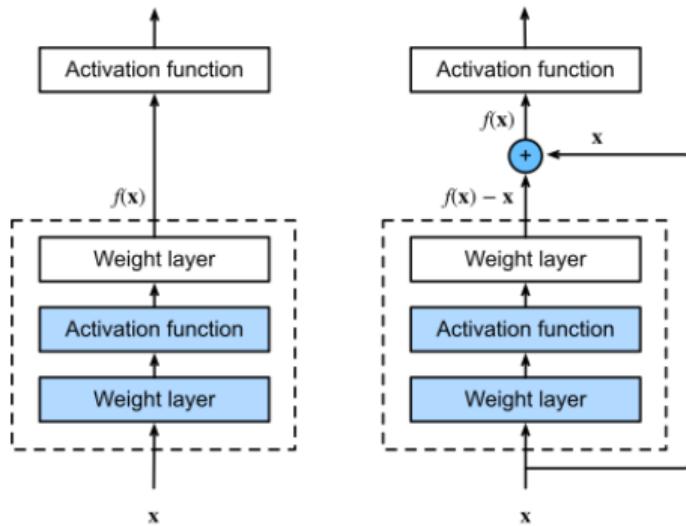
ResNet

- O modelo **ResNet** propõe uma solução muito simples: introduzir um “atalho” entre as camadas.
- Dessa forma, os erros de treinamento não aumentarão, pois o que foi aprendido nas camadas anteriores não é perdido.
- E como a informação ‘flui’ diretamente pelo atalho, evita o *Vanishing Gradient Problem*.

Paper: Deep Residual Learning for Image Recognition

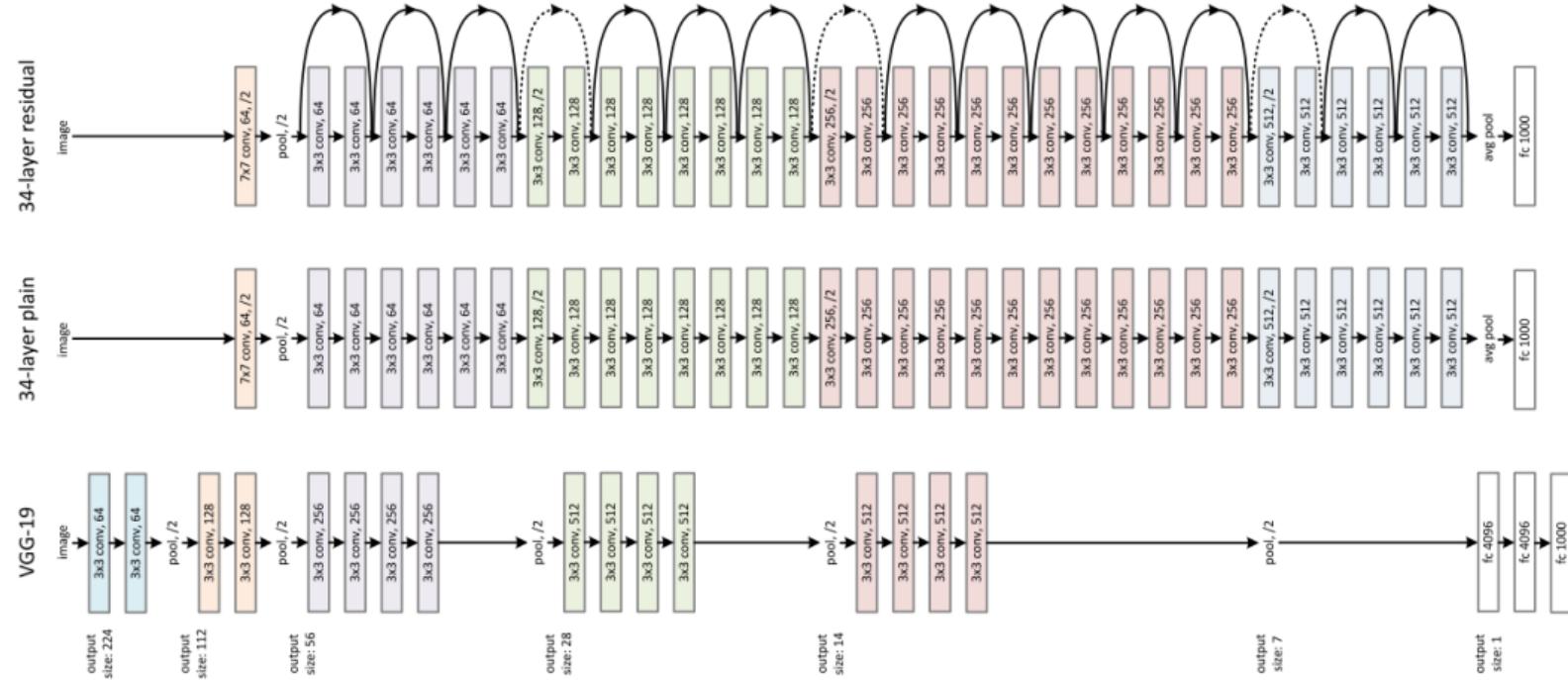
ResNet

Os “atalhos” são chamados de **módulos residuais** ou *skip connections*.



Fonte: Dive into Deep Learning Book

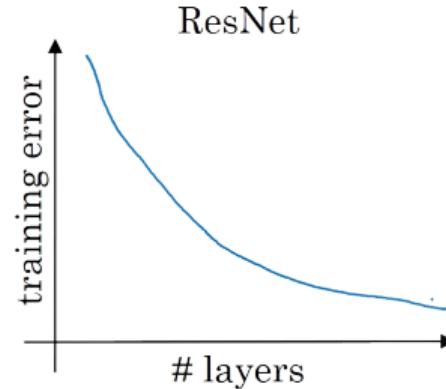
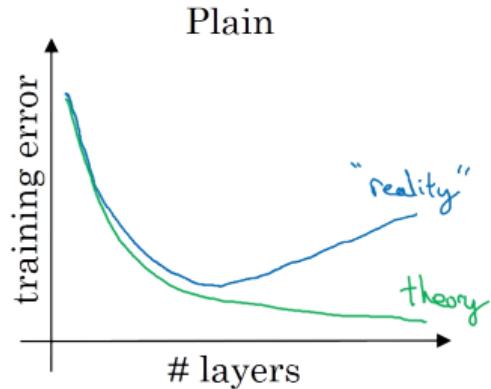
ResNet



Fonte: UFSC: Deep Learning::Reconhecimento de Imagens

ResNet

Vanishing Gradient Problem



Fonte: DeepLearningAI - C4W2L03 Resnets - Andrew Ng

Agenda

- 1 Introdução
- 2 AlexNet
- 3 VGGNet
- 4 GoogLeNet
- 5 Vanishing Gradient Problem
- 6 ResNet
- 7 DenseNet

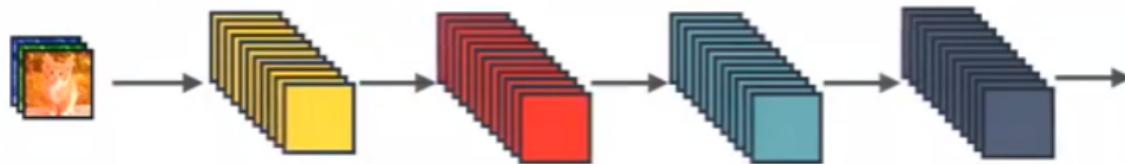
DenseNet

- O modelo **DenseNet** é muito similar ao ResNet, com algumas diferenças fundamentais.
- Utiliza uma estratégia aditiva, que combina a saída da camada anterior com a saída da próxima camada.
- As saídas das camadas são concatenadas, aumentando a profundidade da informação extraída.

Paper: Densely Connected Convolutional Networks

DenseNet

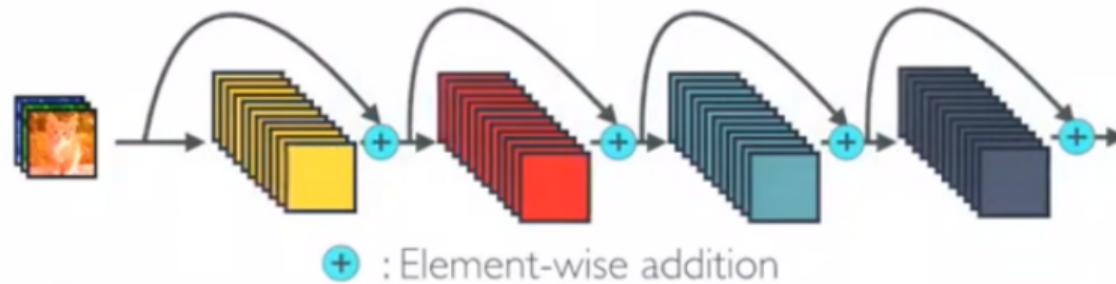
Nas CNNs tradicionais, a imagem de entrada passa por múltiplas convoluções e obtém *features* de alto nível.



Fonte: Review: DenseNet - Dense Convolutional Network

DenseNet

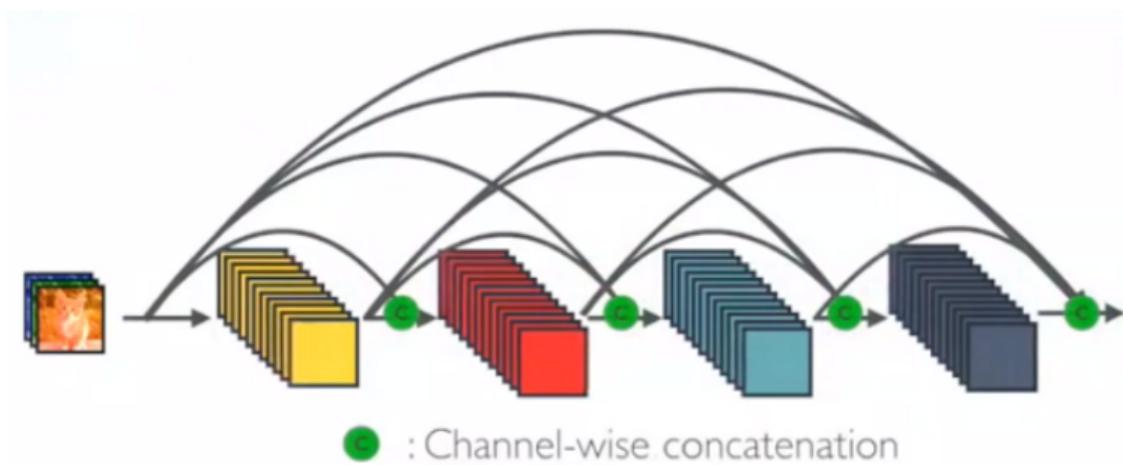
Nas **ResNet** utiliza-se soma elemento-a-elemento (*element-wise*), o que faz com que a informação flua de um módulo para outro.



Fonte: Review: DenseNet - Dense Convolutional Network

DenseNet

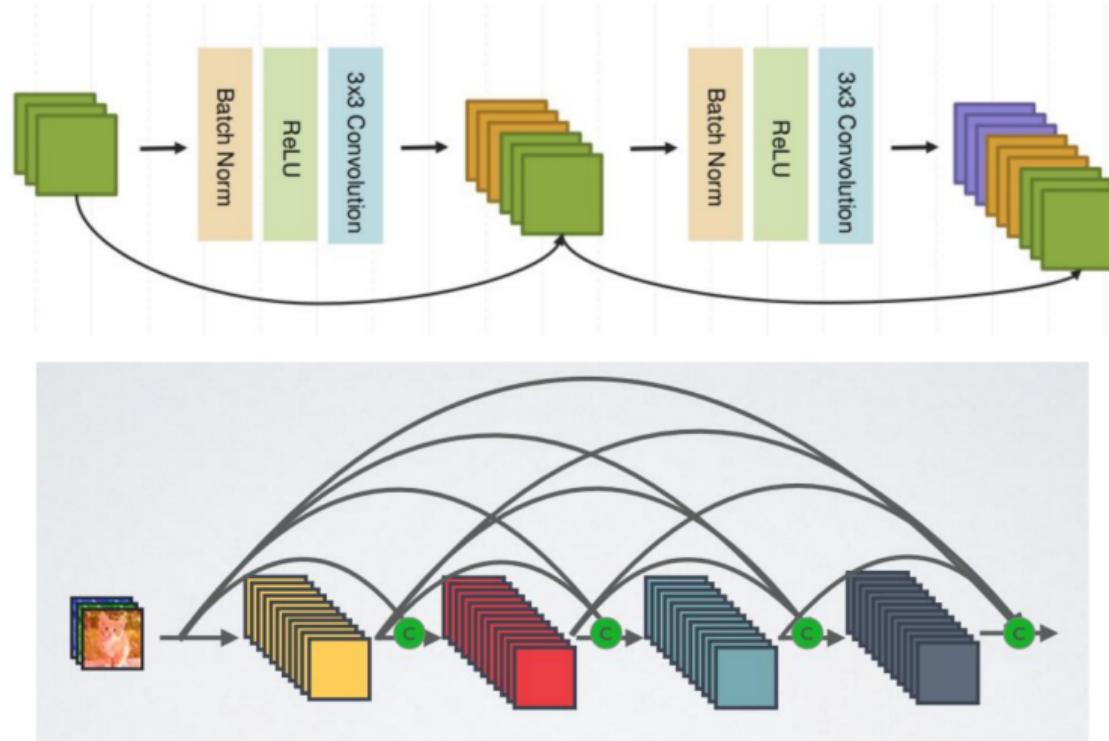
Na **DenseNet**, a saída de cada camada é conectada com todas as camadas posteriores, utilizando concatenação.



Dessa forma, cada camada recebe um “conhecimento coletivo” de todas as camadas precedentes.

Fonte: Review: DenseNet - Dense Convolutional Network

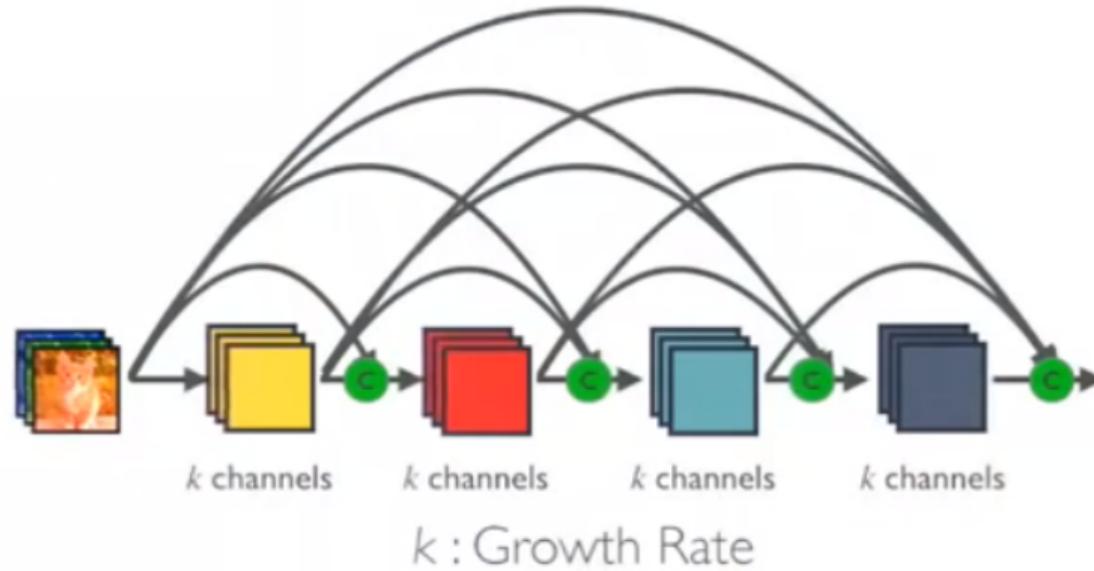
DenseNet



Fonte: Review: DenseNet - Dense Convolutional Network

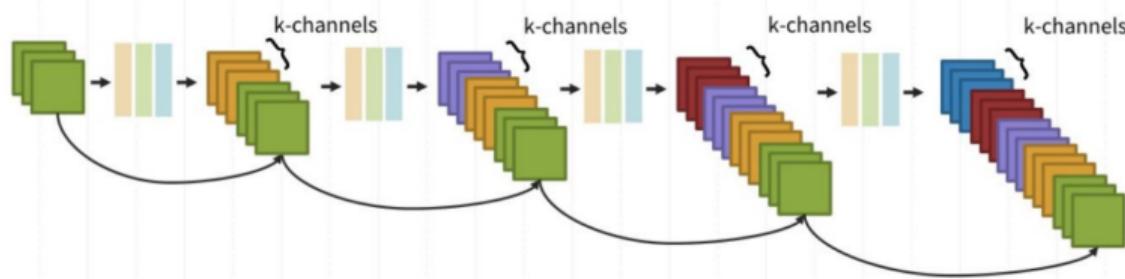
DenseNet

Uma taxa de crescimento k é o número de camadas que são adicionadas em cada camada.



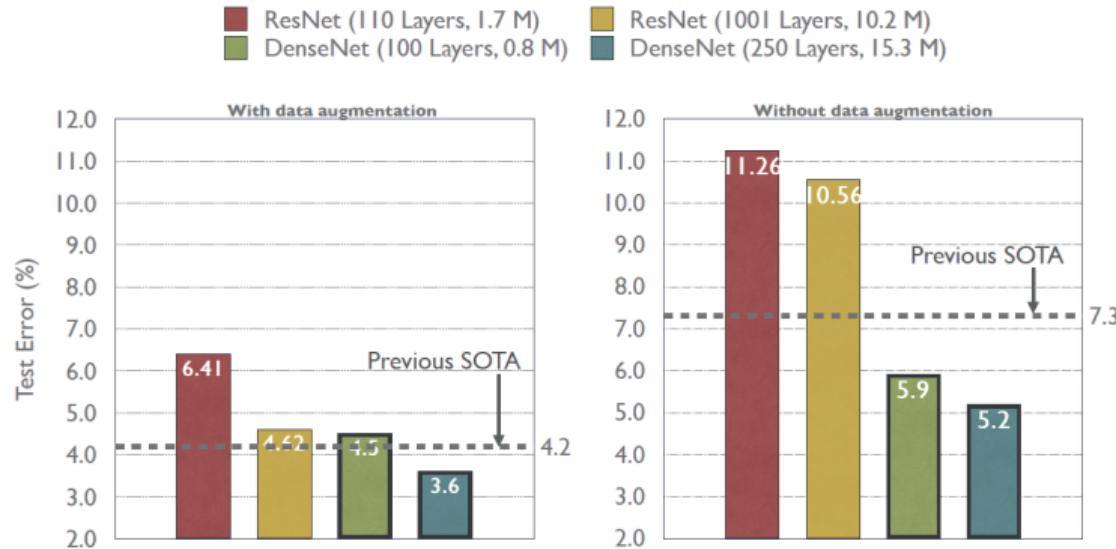
Como cada camada recebe *features* de todas as camadas precedentes, a rede pode apresentar um menor número de canais. Fonte: Review: DenseNet - Dense Convolutional Network

DenseNet



Fonte: Review: DenseNet - Dense Convolutional Network

DenseNet



Referências

- Going Deeper with Convolutions
- Rethinking the Inception Architecture for Computer Vision
- Deep Residual Learning for Image Recognition
- Densely Connected Convolutional Networks
- A Neural Algorithm of Artistic Style

Curso Inteligência Artificial: do Zero ao Infinito

Arquiteturas Baseadas em CNNs Parte 2

Universidade Federal de Mato Grosso