

MÓDULO 3

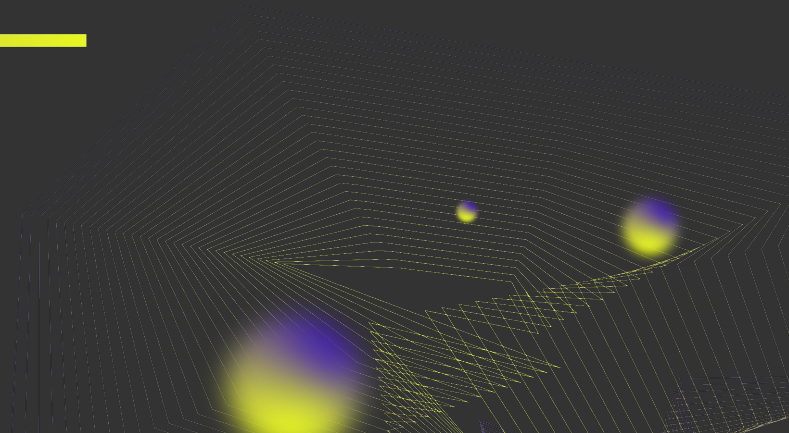
Principais Arquiteturas de CNNs

ESPECIALIZAÇÃO
INTELIGÊNCIA ARTIFICIAL
& CIÊNCIA DE DADOS

SEAD
UFES | Superintendência de
Educação a Distância

Bruno Légora Souza da Silva

Professor do Departamento de Informática/UFES



ÍNDICE



1. Principais Arquiteturas de CNNs
2. Laboratório 07 (Segmentação de Imagens com CNN - U-Net)



1. Principais Arquiteturas de CNNs



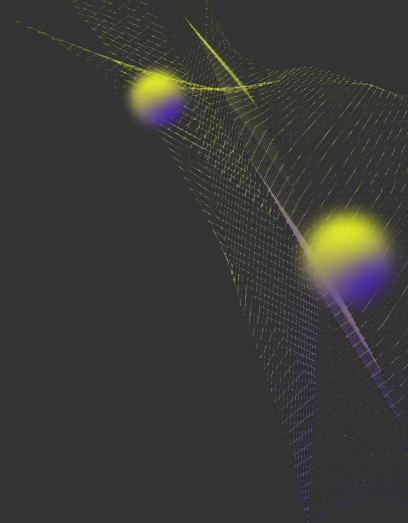
Principais Arquiteturas de CNNs

- Nesta aula, veremos algumas arquiteturas de CNNs:
 - AlexNet
 - VGG
 - GoogLeNet/Inception
 - ResNet
 - U-Net
 - GAN*
 - Vision Transformers*



AlexNet

- Há algum tempo atrás, existia uma competição chamada ImageNet Large Scale Visual Recognition Challenge (ILSVRC), com uma base (ImageNet) de 1,2 milhão de imagens pertencentes a 1000 classes;





AlexNet

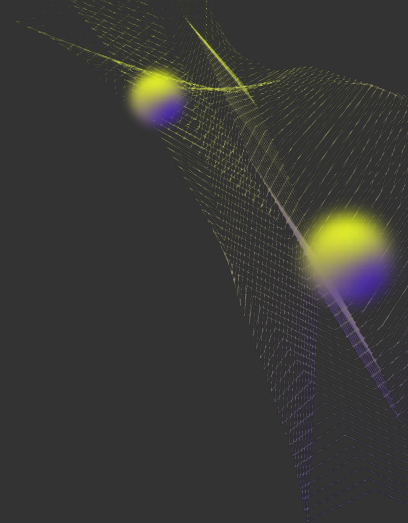
- Até a época, era um problema bastante difícil – os melhores trabalhos usavam alternativas “clássicas” e obtinham cerca de 25 a 30% de erro





AlexNet

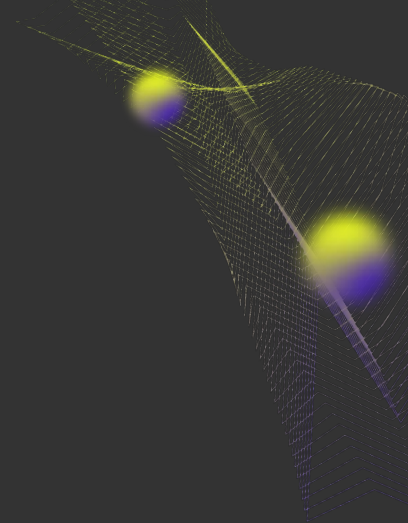
- Em 2012, 3 pesquisadores propuseram a arquitetura de rede convolucional AlexNet:
 - Alex Krizhevsky, Ilya Sutskever (co-fundou OpenAI), e Geoffrey Hinton (Nobel 2024)





AlexNet

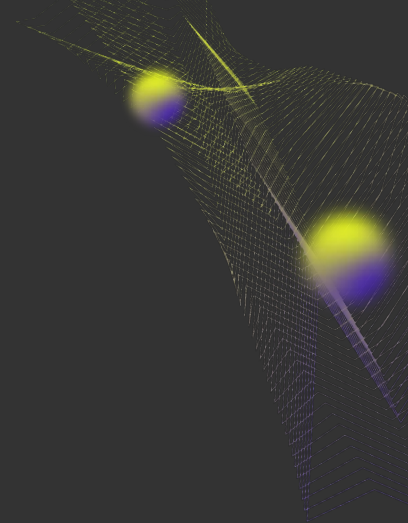
- Essa rede não possuía nada “novo”
 - Camadas totalmente conectadas
 - Camadas de pooling
 - Camadas Convolucionais
- Todas elas já existiam desde ao menos 1989 (na rede chamada LeNet, proposta pela equipe de Yann LeCunn)





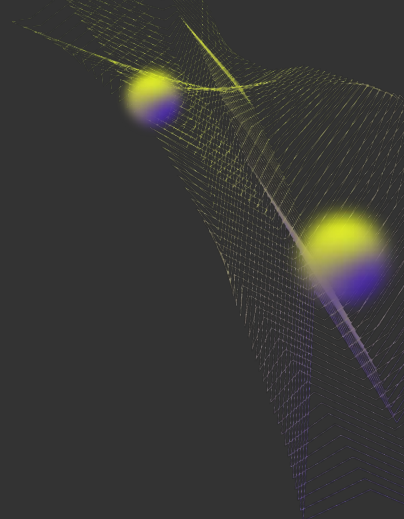
AlexNet

- O trabalho deles chegou a conclusão de que quanto mais profunda era a rede, melhores eram os resultados;
- O problema: treinar era muito difícil/demorado



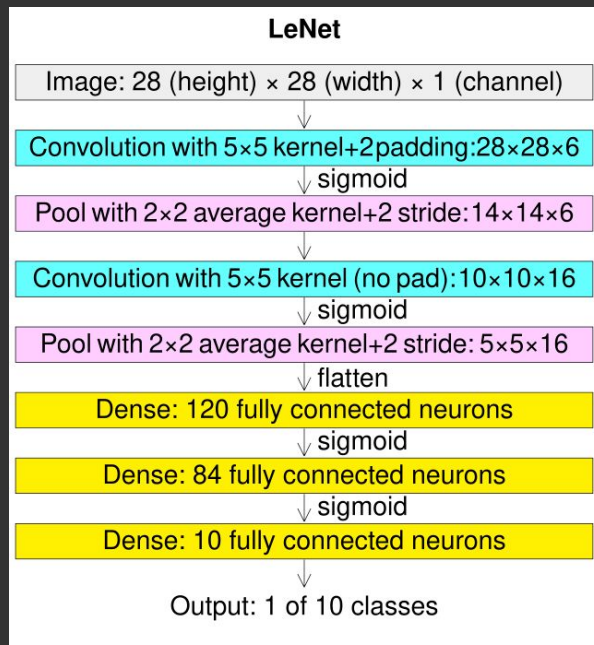


AlexNet

- A equipe usou duas GPUs GTX 580 (e CUDA) para tornar esse treinamento possível.
 - Nessa altura da disciplina, vocês já devem ter feito o EA3 e o Lab6, e devem ter percebido a diferença entre usar GPUs ou não;
- 

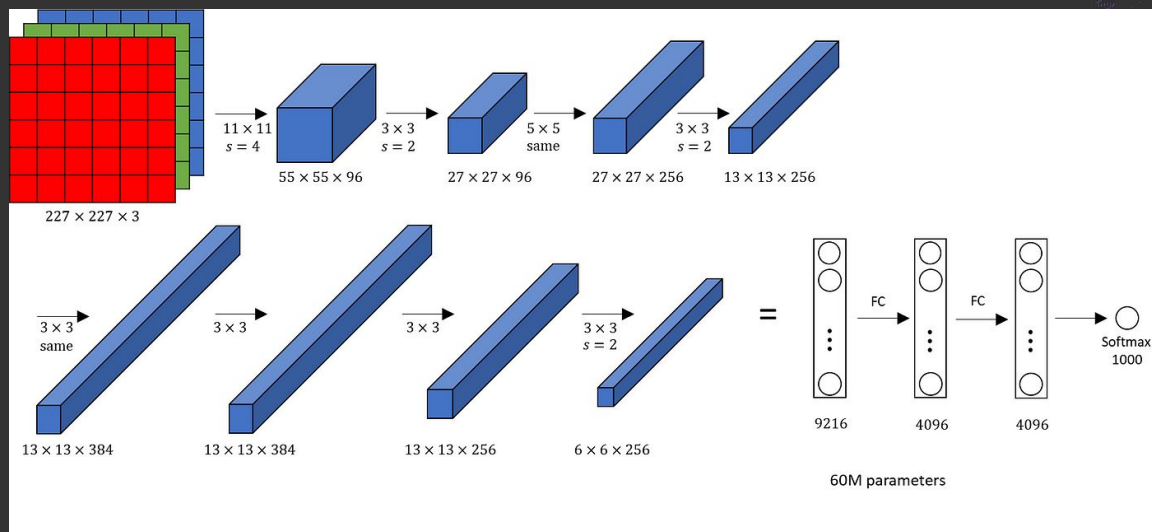
AlexNet

- A LeNet5, proposta em 1998:
 - ~60 mil parâmetros



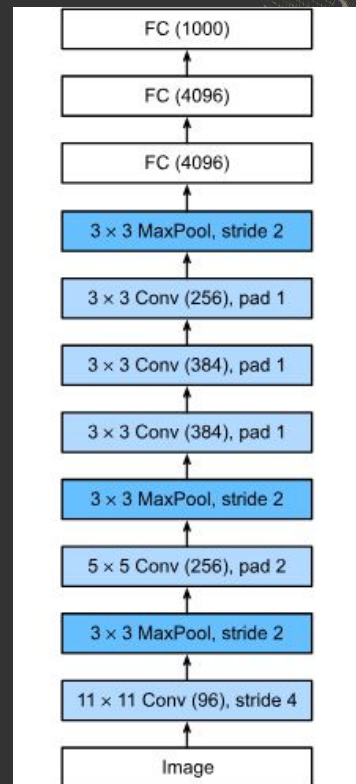
AlexNet

- A AlexNet era composta de 11 camadas:
 - ~60M de parâmetros



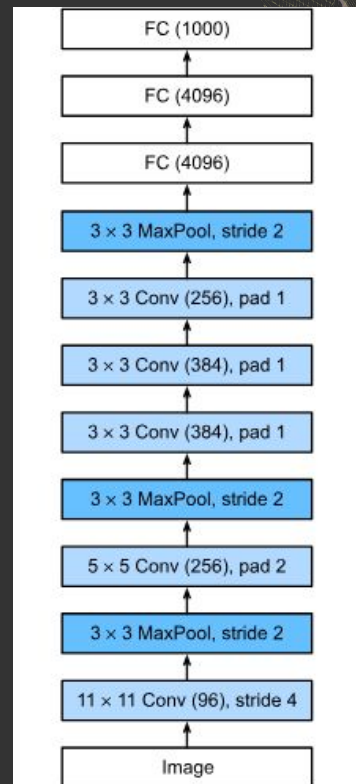
AlexNet

- Entrada - $227 \times 227 \times 3$
- C1: 96 filtros $3 \times 11 \times 11$ com stride 4
 - saída $55 \times 55 \times 96$
- C2: Max-Pooling 3×3 com stride 2:
 - saída: $27 \times 27 \times 96$



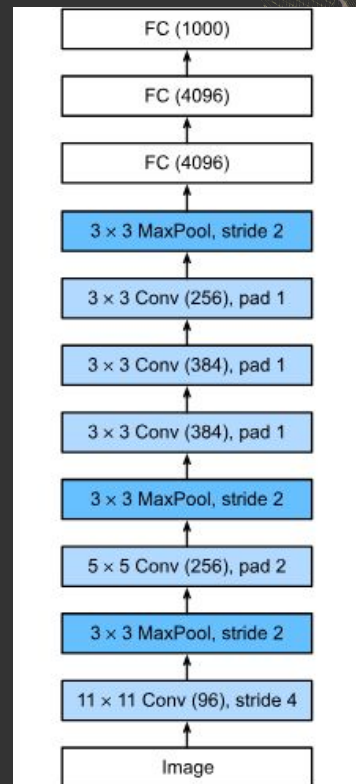
AlexNet

- C3: 256 filtros 96x5x5, stride 1, pad 2
 - Saída: 27x27x256
- C4: Max-Pooling 3x3, stride 2
 - Saída: 13x13x256



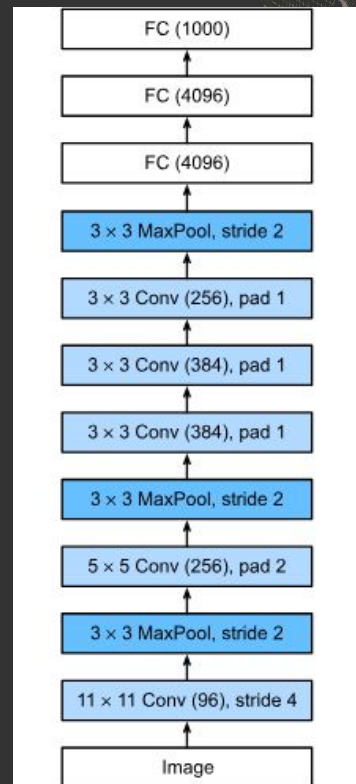
AlexNet

- C5: 384 filtros $256 \times 5 \times 5$, stride 1, pad 2
 - Saída: $13 \times 13 \times 284$
- C6: 384 filtros $384 \times 3 \times 3$, stride 1, pad 1
 - Saída: $13 \times 13 \times 384$
- C7: 256 filtros $384 \times 3 \times 3$, stride 1, pad 1
 - Saída: $13 \times 13 \times 256$



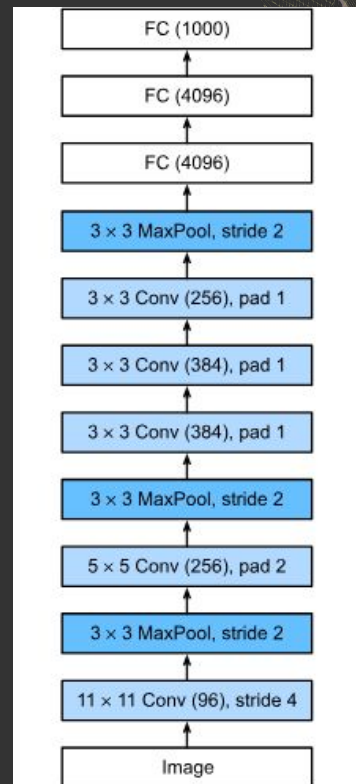
AlexNet

- C8: Max-Pooling 3x3, stride 2
 - saída: 6x6x256
- C9: Totalmente conectada com 4096 neurônios:
 - 9216 x 4096 parâmetros



AlexNet

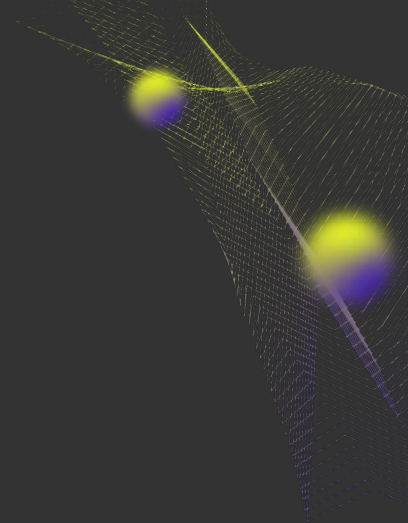
- C10: Totalmente conectada com 4096 neurônios:
 - 4096 x 4096 parâmetros
- C11: Totalmente conectada com 1000 neurônios:
 - 4096 x 1000 parâmetros



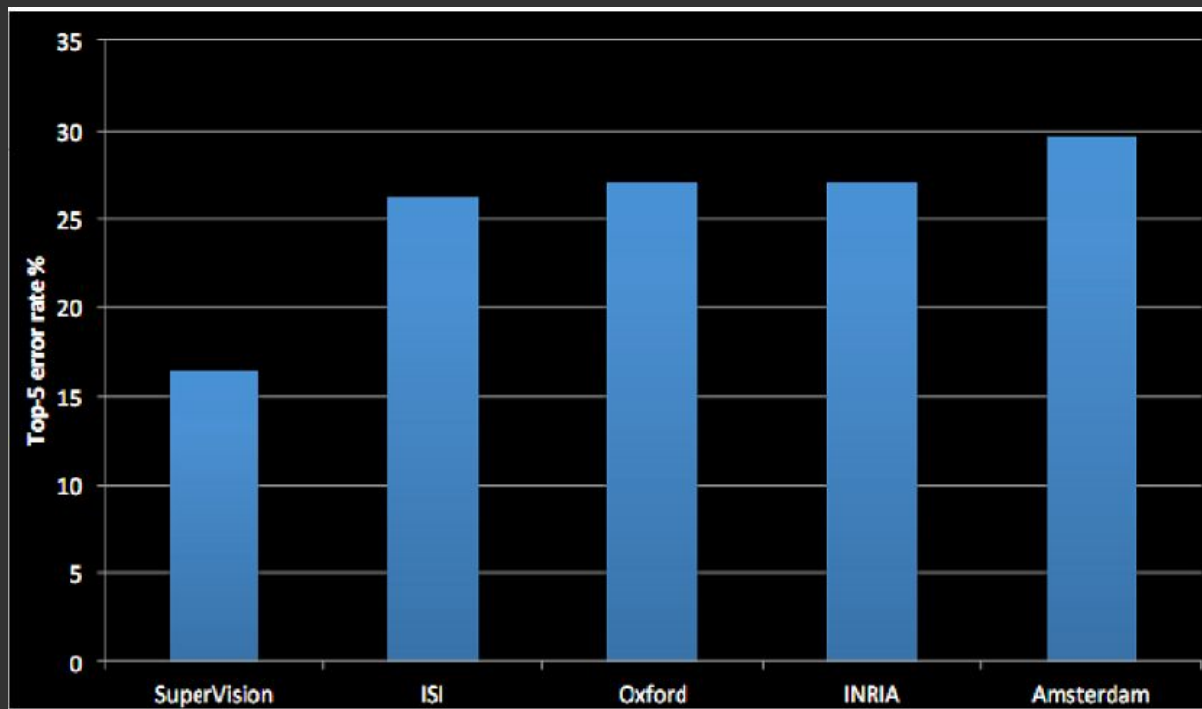


AlexNet

- Dessa forma, a arquitetura foi treinada na base de dados e ganhou a edição de 2012 da competição com uma taxa de erro aproximadamente 10% menor do que o segundo colocado!



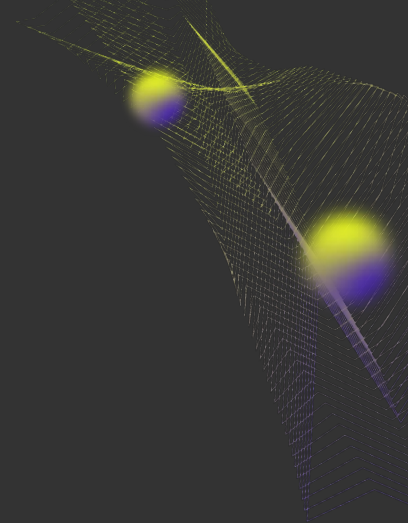
AlexNet





AlexNet

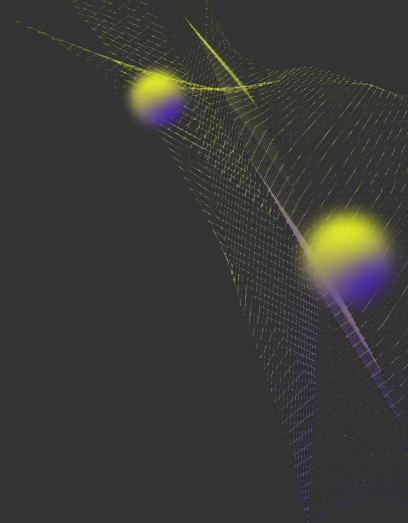
- A principal contribuição do trabalho da AlexNet foi iniciar a transição do treinamento de redes neurais de CPUs para GPUs, o que revolucionou o desenvolvimento da IA.





AlexNet

- Mesmo usando duas GPUs GTX580, o treinamento da rede na base ImageNet levava cerca de uma semana
 - Diversos conjuntos de hiperparâmetros foi testado!





VGG

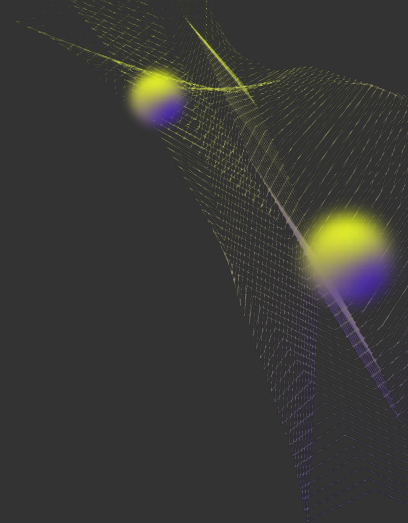
- Por volta de 2014, o grupo Visual Geometry Group (VGG) desenvolveu algumas CNNs
 - A que ficou mais conhecida foi a VGG16, com “16” camadas
 - ~138M parâmetros



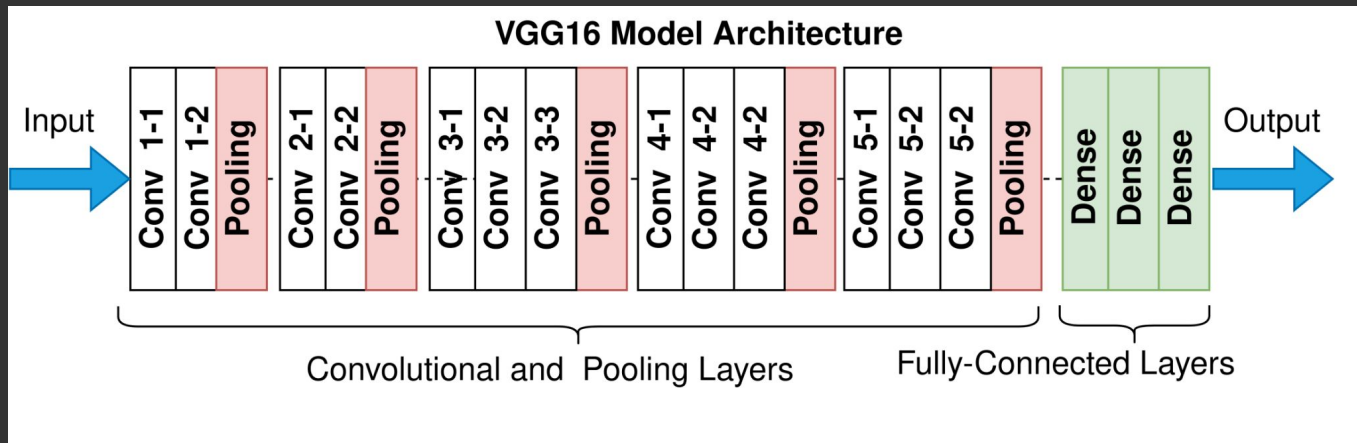


VGG

- Das “16” camadas, 13 são convolucionais, onde todas usam filtros 3x3
 - Divididas em “blocos” de convs+pooling
- As 3 últimas são totalmente conectadas



VGG



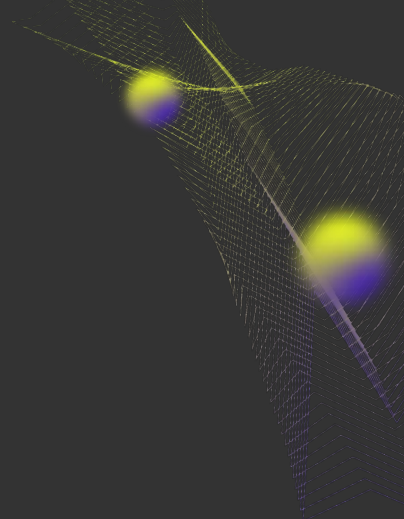


VGG

- O primeiro bloco (2 camadas) usa 64 filtros 3x3.
- O segundo bloco (2 camadas) usa 128 filtros 3x3.
- O terceiro bloco (3 camadas) usa 256 filtros 3x3.
- Os blocos 4 e 5 (3 camadas) usam 512 filtros 3x3.
- As camadas finais são iguais a da AlexNet
 - 4096, 4096, 1000 neurônios;



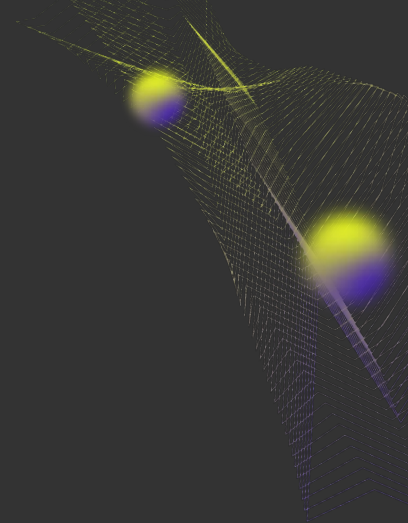
VGG

- Em termos de resultado, ela obteve cerca de 7,3% de taxa de erro na ImageNet (segundo lugar da edição de 2014)
 - Uma contribuição relevante da VGG foi a utilização de “blocos” iguais ao longo da rede
- 



GoogLeNet/Inception

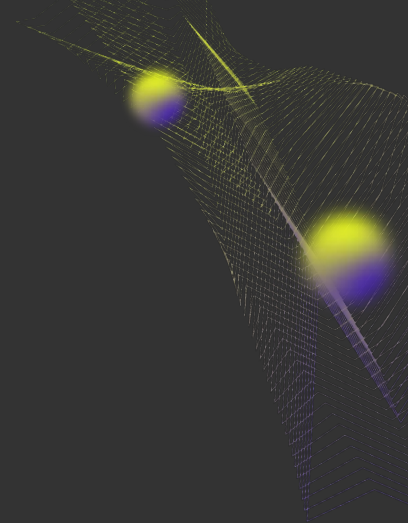
- Em 2014, uma equipe da Google desenvolveu a GoogLeNet, que era uma arquitetura com 22 camadas
 - o nome “Inception” vem do filme!
 - ~7M parâmetros





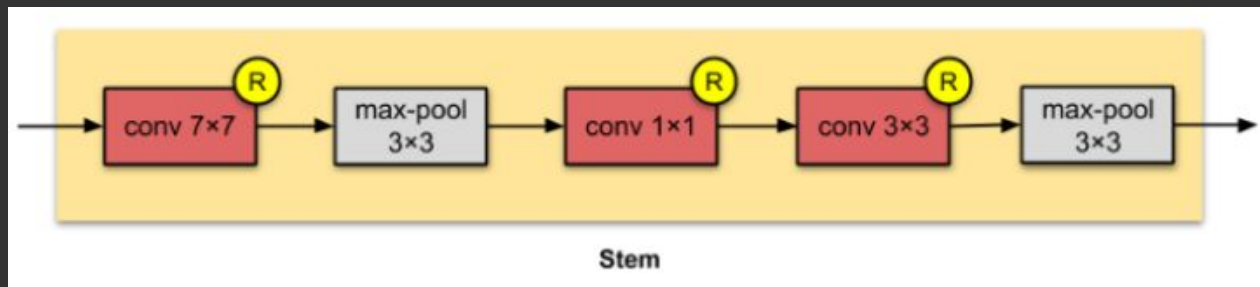
GoogLeNet/Inception

- A GoogLeNet trouxe 3 “partes”:
 - A parte inicial – chamada de “stem”
 - A parte de processamento – chamada de “body”
 - A parte de predição – chamada de “head”



GoogLeNet/Inception

- A “stem” faz o “pré-processamento”





GoogLeNet/Inception

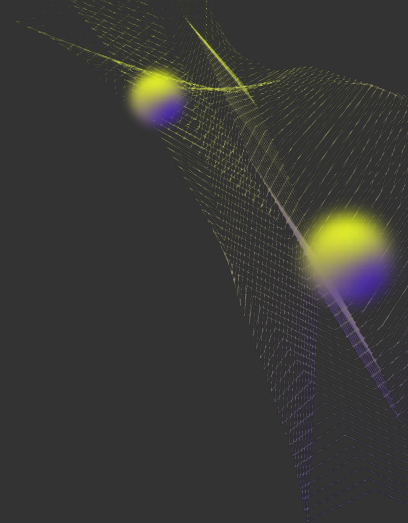
- O processamento principal é feito por módulos “inception”
 - O segredo do número de parâmetros está aqui!





GoogLeNet/Inception

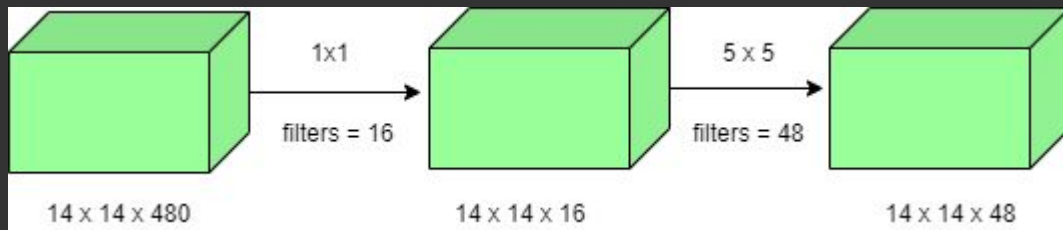
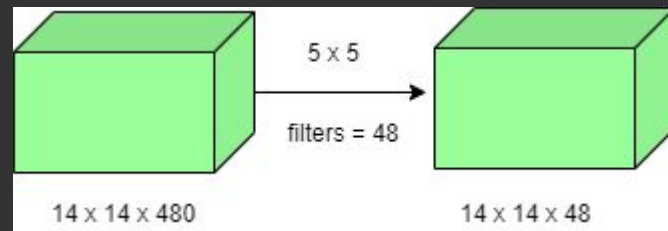
- Ao invés de processar convoluções com muitos canais, elas são divididas em duas – uma 1×1 com o número de canais original, que leva para uma dimensão menor, e em seguida a convolução “real”



GoogLeNet/Inception

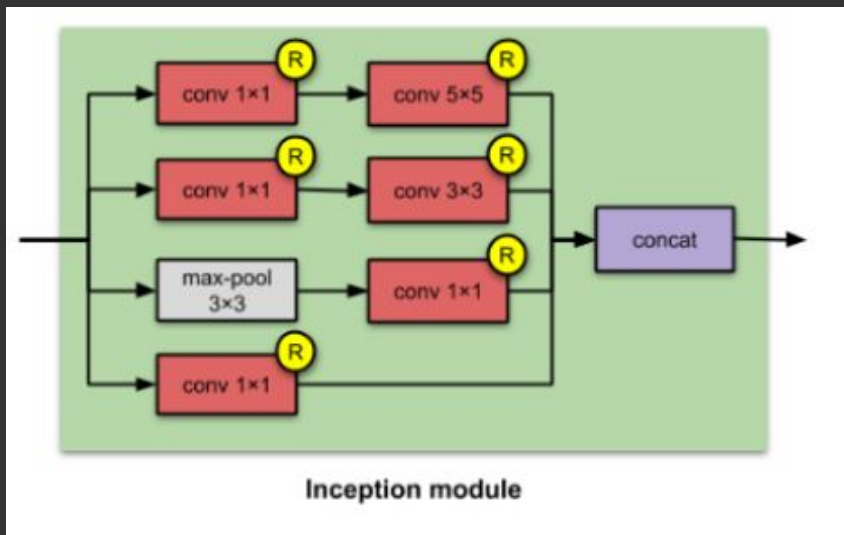
Exemplo!

- Ao invés termos ~576k parâmetros
 - 48 filtros 5x5x480
- Teremos $7.7k + 19.2k \cong 27k$:
 - 16 filtros 1x1x480
 - 48 filtros 5x5x16



GoogLeNet/Inception

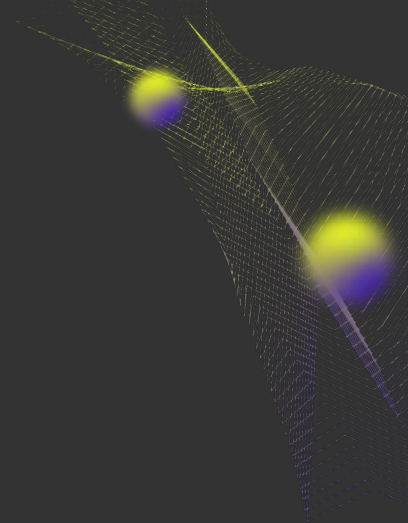
- O módulo inception:





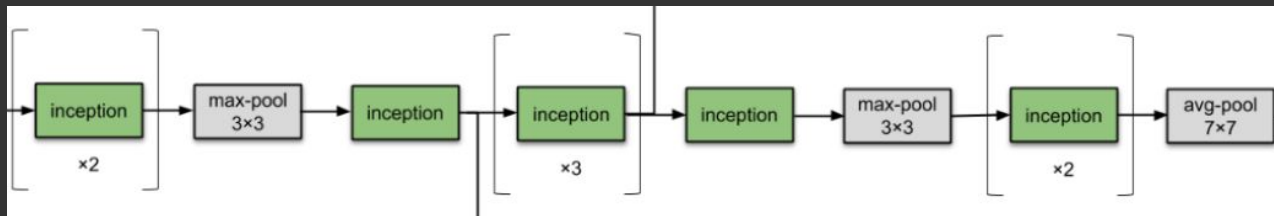
GoogLeNet/Inception

- O “body” da rede é composto de:
 - 2x inception
 - max pool 3x3
 - 5x inception
 - maxpool 3x3
 - 2x inception
 - avg. pool



GoogLeNet/Inception

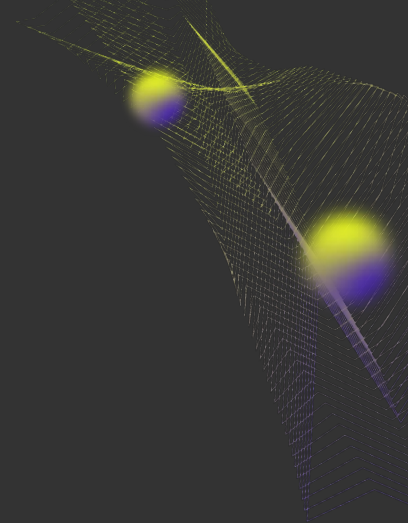
- O “body” da rede é composto de:



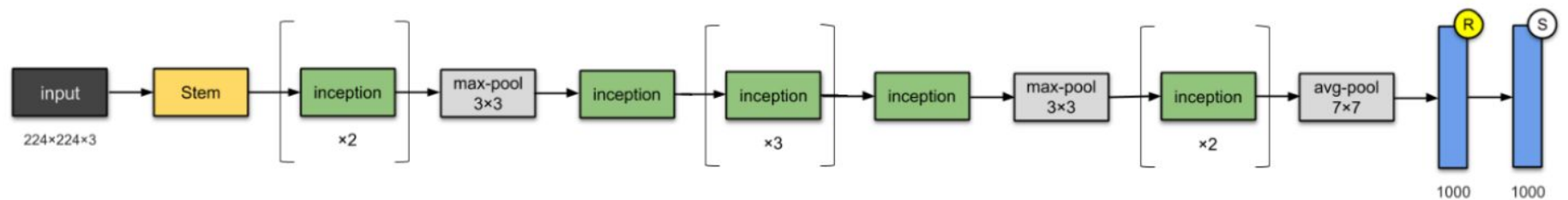


GoogLeNet/Inception

- E a rede é finalizada com a “head”, composta de 2 camadas totalmente conectadas com 1000 neurônios.
 - Com exceção da última camada, que usa ativação softmax, as demais são ReLU.



GoogLeNet/Inception





GoogLeNet/Inception

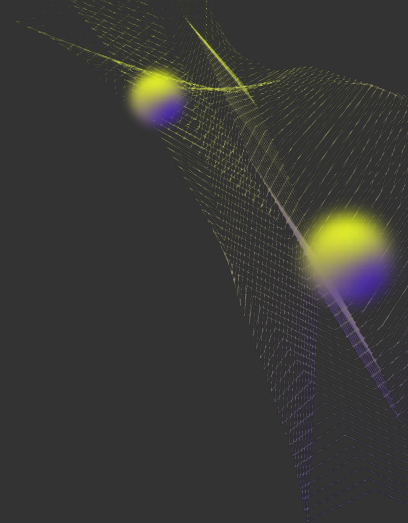
- A rede GoogLeNet trouxe a ideia de uma “rede dentro da rede” – cada módulo era uma subrede com várias camadas
 - Daí a relação com o filme Inception!





GoogLeNet/Inception

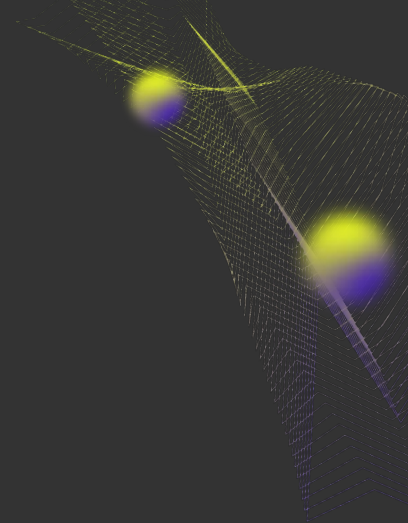
- Mesmo com um número de parâmetros muito menor que AlexNet e a VGG16, a GoogLeNet obteve 6,67% de erro na ImageNet.
- Ganhou da VGG em 2014!





GoogLeNet/Inception

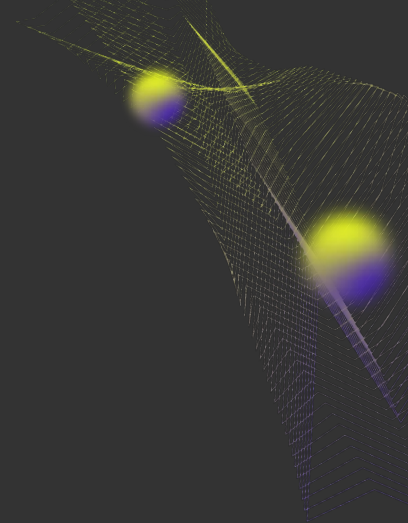
- Possui um problema de gradiente “dissipado” (*vanishing gradient*), o que dificulta seu treinamento.
 - Causado pela grande quantidade de camadas ocultas!





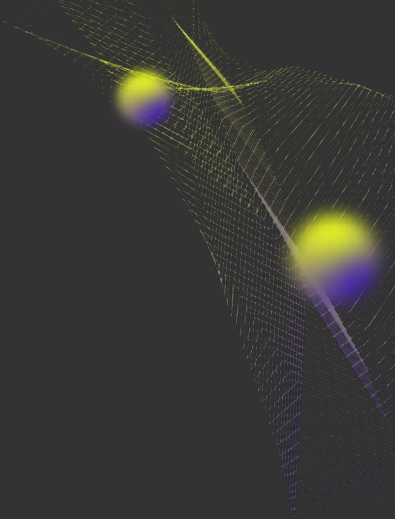
Resnet

- Nos trabalhos da VGG e da GoogLeNet, foi detectado que tornar a rede mais profunda causava um **aumento do erro**;
 - Problema do gradiente “dissipado”



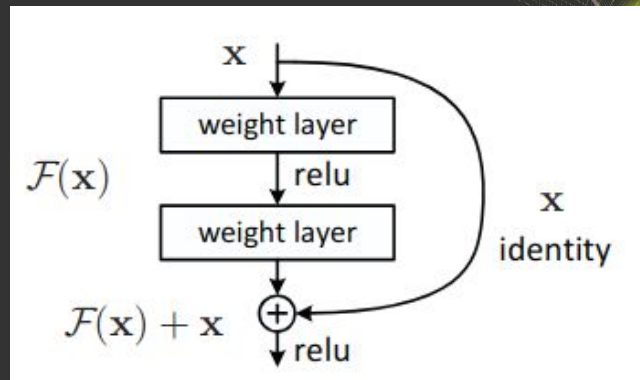


Resnet

- Em 2015, uma nova estratégia de rede neural foi proposta.
 - Assim como a inception, usa a ideia de “rede dentro da rede”;
 - Porém, a estratégia proposta resolvia o problema do gradiente “dissipado”, tornando possível treinar redes muito profundas!
- 

Resnet

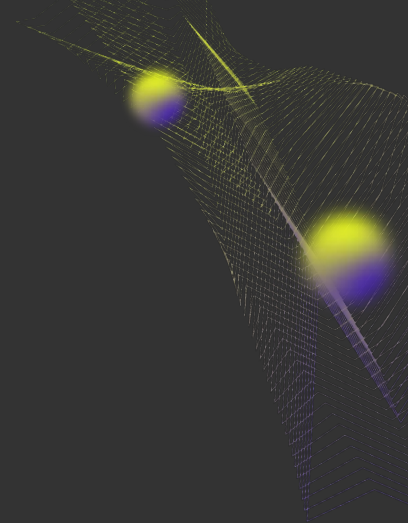
- A proposta foi construir módulos com a seguinte estrutura:
 - Colocar conexões “residuais” entre entrada e saída do módulo
 - E isso facilita muito o processo de otimização dos pesos da rede!





Resnet

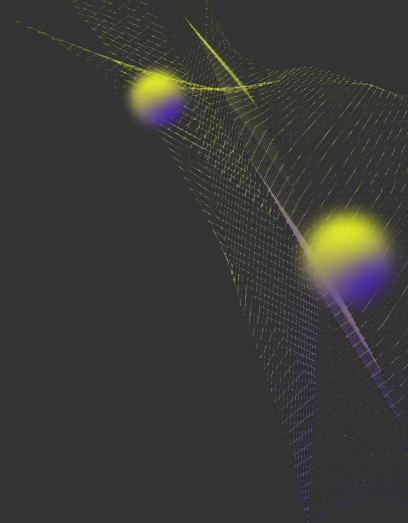
- Algumas redes foram propostas pelos autores da ResNet:
 - 18, 34, 50, 101 e 152 camadas
 - 11.7M, 21.7M, 25.6M, 44.7M, 60.4M parâmetros, respectivamente





Resnet

- As camadas de 50, 101 e 152 combinam a conexão residual com a estratégia de convolução 1D, usada na Inception
 - Desta forma, possuem proporcionalmente menos parâmetros



Resnet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				



Resnet

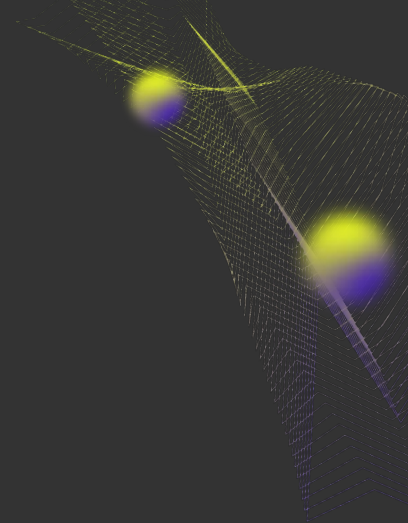
- Em termos de erro na ImageNet:

method	top-5 err. (test)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	3.57



Resnet

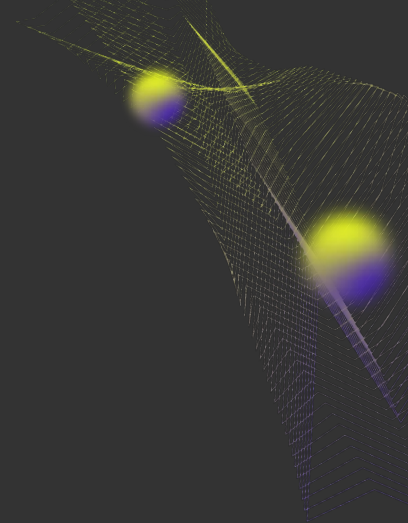
- A ideia de conexões residuais não era nova;
 - Conceito proposto desde os primórdios das redes neurais
- O problema do gradiente “dissipado” também já era conhecido;





Resnet

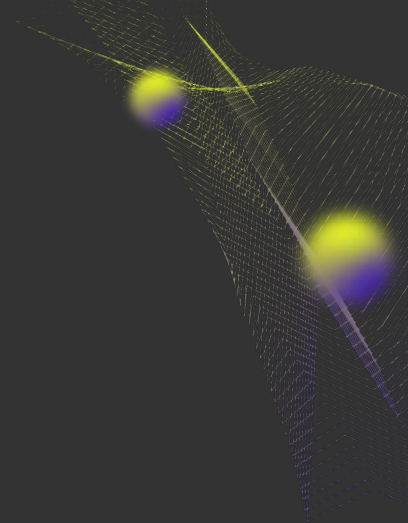
- Os autores da ResNet recuperaram esse conceito e aplicaram ao aprendizado profundo, conseguindo treinar redes muito mais profundas que as “concorrentes”
 - até 152 camadas, contra as 22 da GoogLeNet, ou 16/19 da VGG





AlexNet, VGG, Inception e ResNet

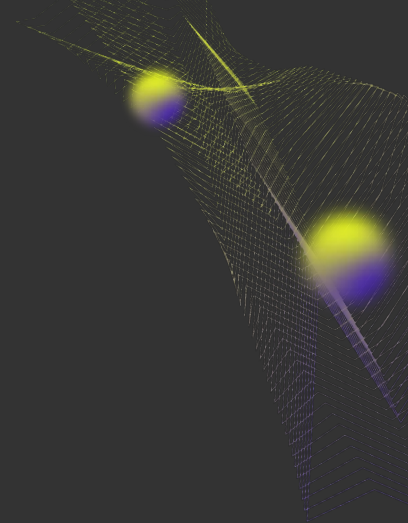
- As 4 redes citadas são consideradas por muitos as mais importantes dos “primórdios” do aprendizado profundo
- Todas elas propostas para **classificar** imagens da base ImageNet;





U-Net

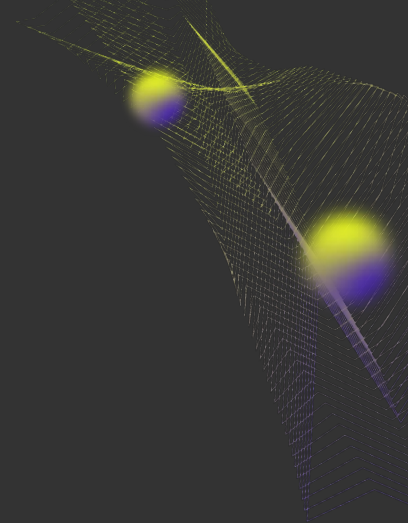
- Todas as redes apresentadas foram usadas para problemas de classificação das imagens;
- Elas processam imagens e resultado num número (uma classe)



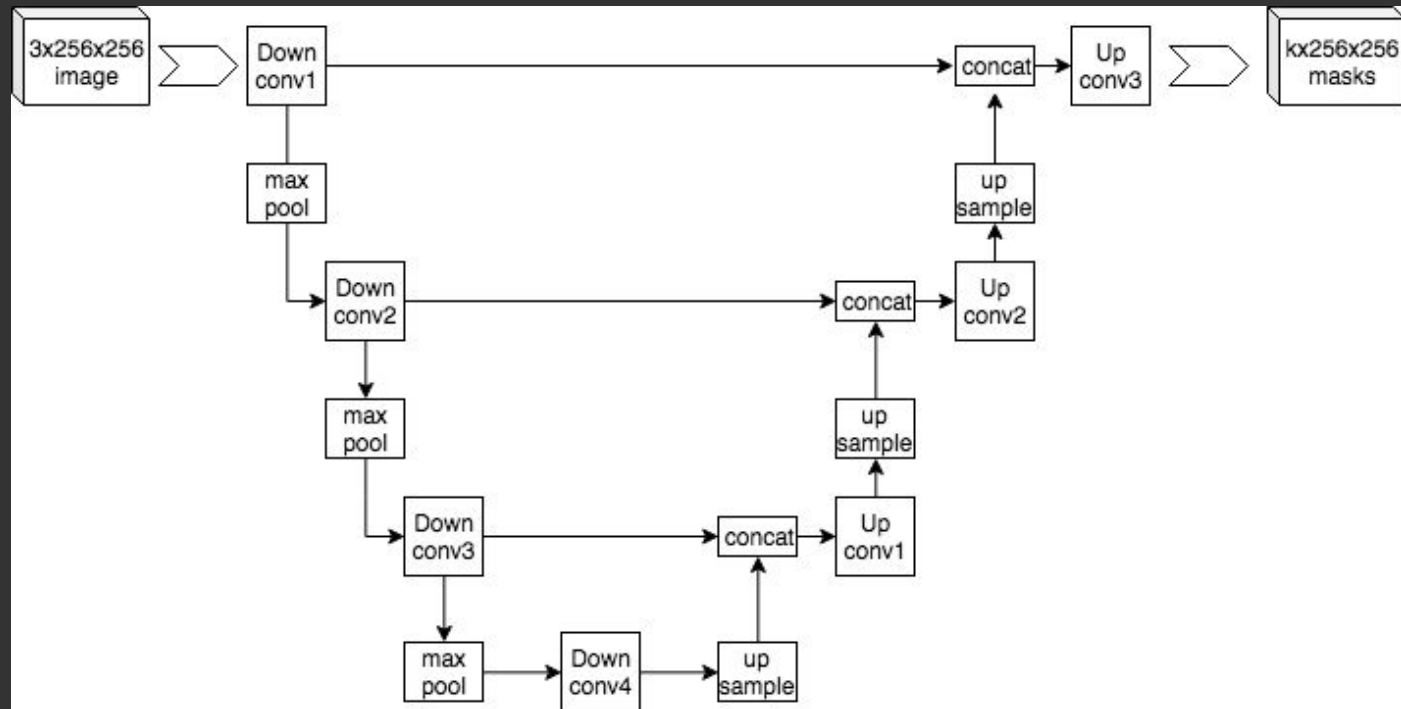


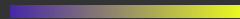
U-Net

- Outras redes foram propostas com outros objetivos:
 - Por exemplo, retornar um “mapa de calor” (uma outra imagem) a partir de uma imagem de entrada
- Um exemplo é a U-net!

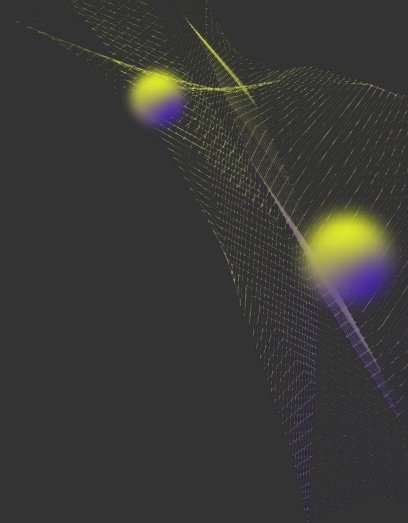
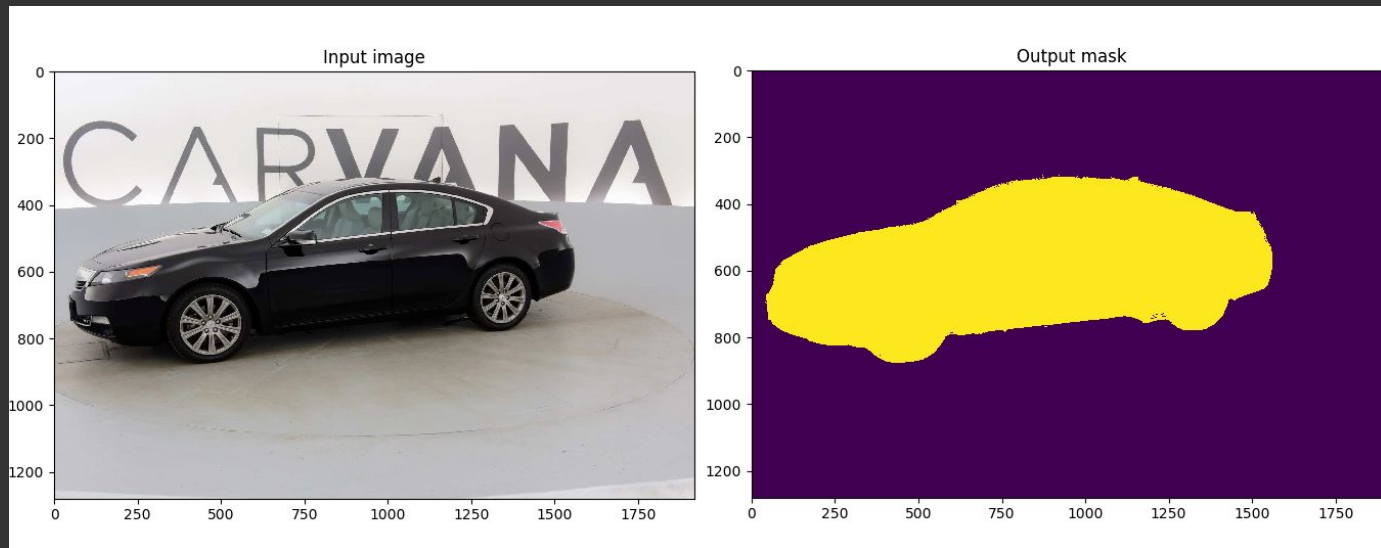


U-Net





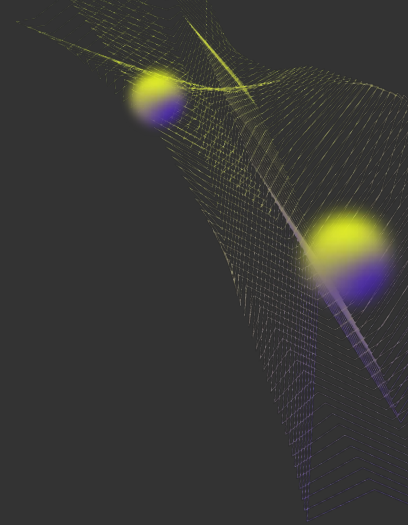
U-Net





U-Net

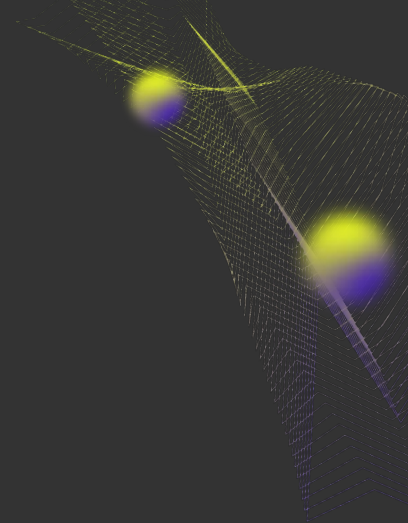
- Veremos ela em mais detalhes no Laboratório 7!





Outras Redes

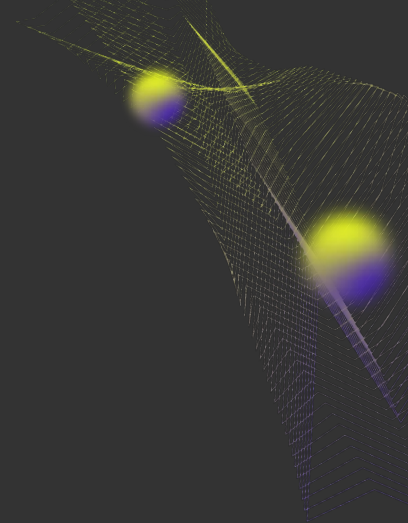
- Além das redes de classificação e redes como a U-net, outras abordagens também foram propostas.
- Não entraremos em detalhes delas nesta aula, mas há um material para estudo no AVA!





Outras Redes

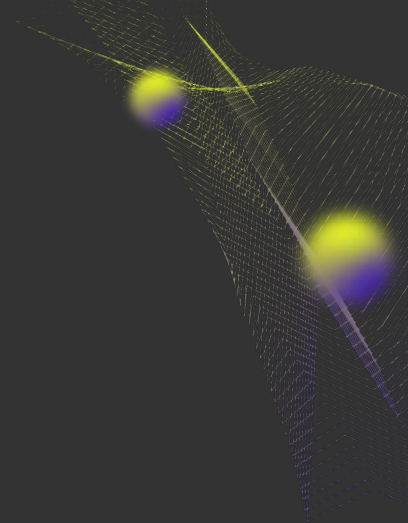
- Exemplos:
 - GAN
 - Vision Transformers





GAN

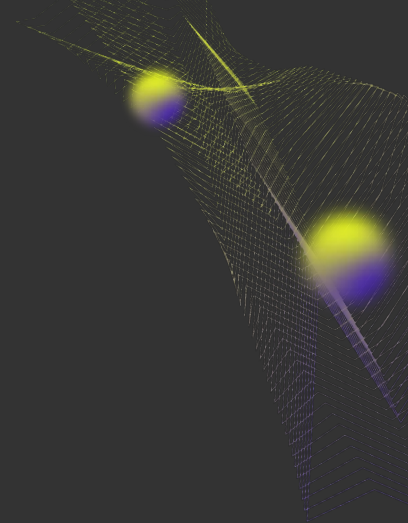
- Sem entrar em muitos detalhes, as GAN são um framework de rede neurais treinados com o objetivo de gerar imagens a partir de ruído.



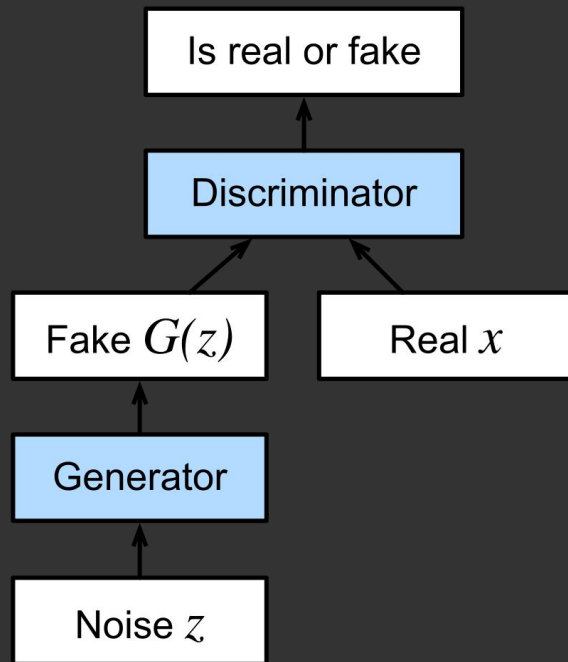


GAN

- Desenvolvidos em 2014 com duas redes com objetivos bem distintos:
 - Um gerador de imagens
 - Um discriminador de imagens



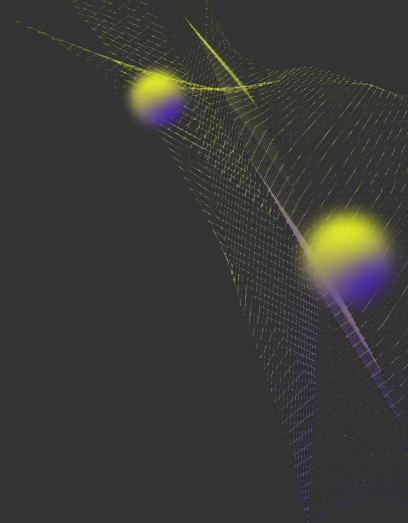
GAN





GAN

- Ambas as redes são treinadas em conjunto na forma de um jogo de soma zero (se um ganha, outro perde)
 - O gerador é treinado para gerar o discriminador, que é treinado para identificar imagens falsas;



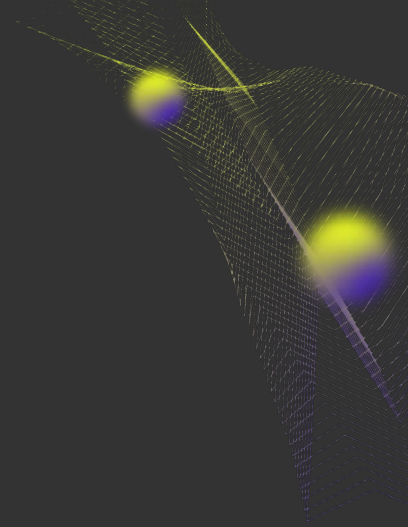


GAN

- A matemática por trás do treinamento é complexa;
- Após o treinamento, idealmente temos um gerador e um discriminador teoricamente muito bons para uma determinada tarefa;



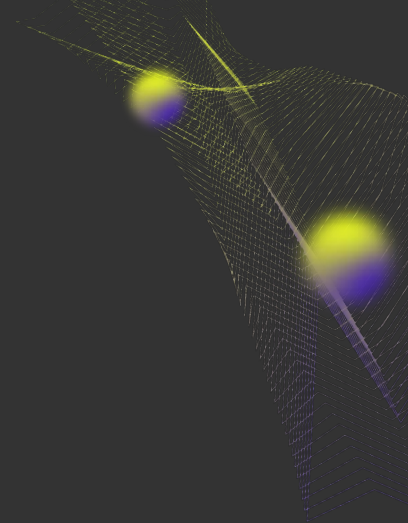
GAN





Vision Transformers*

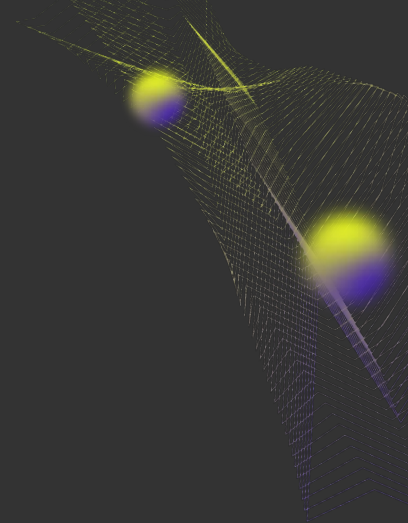
- Em 2017, a arquitetura Transformer foi proposta.
- Originalmente para processamento de linguagem natural, ela adiciona um “mecanismo de atenção” que aprende a “dar atenção às partes mais importantes”.





Vision Transformers*

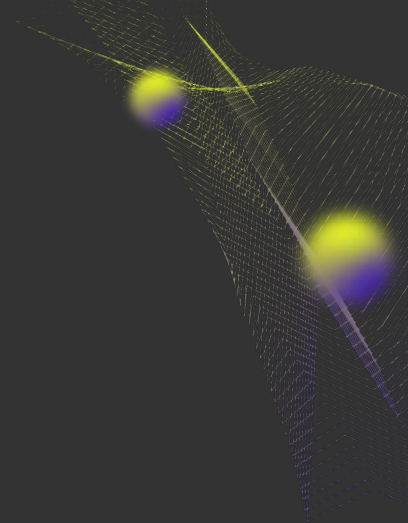
- Transformers são a base do modelo GPT.
- Vocês estudarão eles na próxima disciplina!
- Por volta de 2020, o mecanismo de atenção foi aplicado a imagens, dando origem aos chamados Vision Transformers!



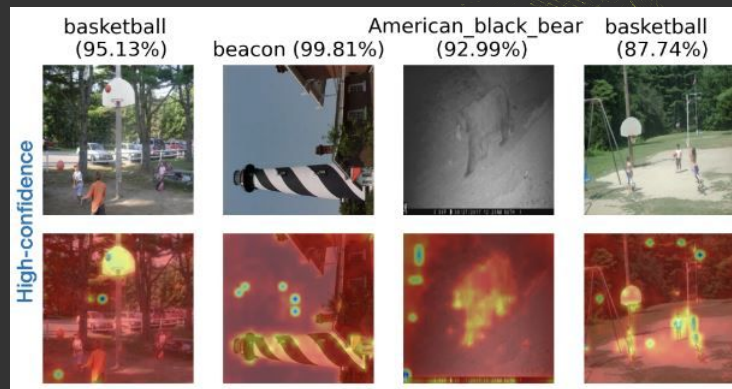


Vision Transformers*

- O mecanismo, na teoria, aprende a “dar mais peso” (ou mais atenção) as partes que ele julga mais importantes em uma imagem, criando um mapa de atenção.

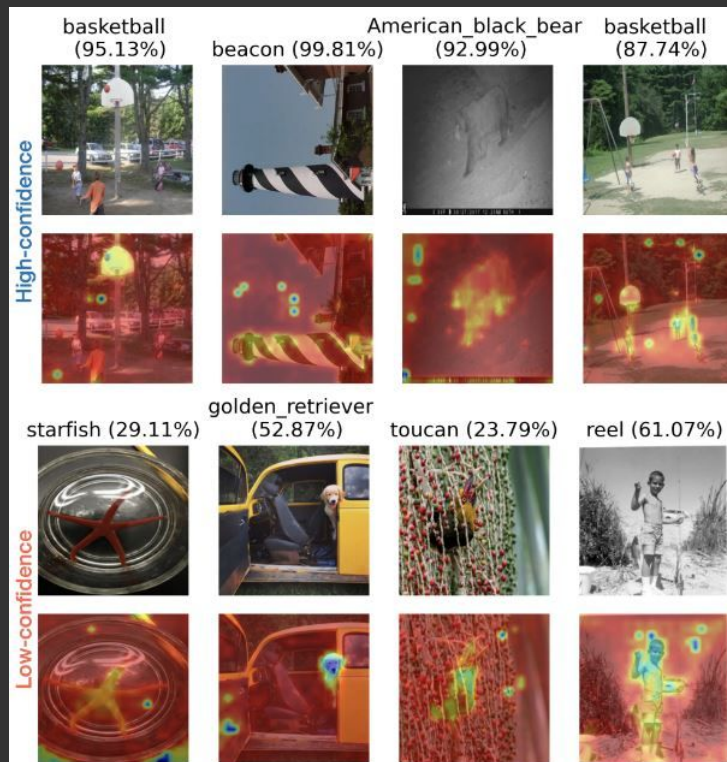


Vision Transformers*



- Por exemplo, para cada imagem, um mapa de atenção é gerado:
 - Regiões mais “quentes” são consideradas mais importantes para a classificação daquela imagem.

Vision Transformers*





2. Laboratório 07



Laboratório 7

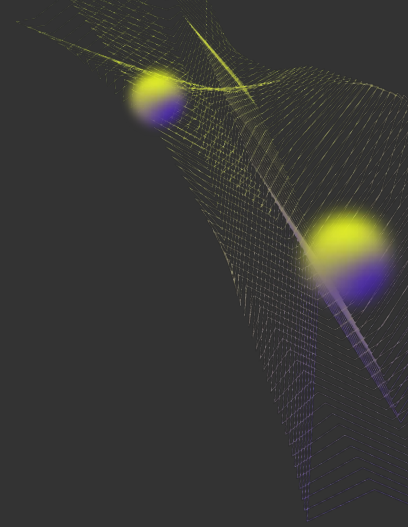
- No 7º laboratório da disciplina, vocês irão ter contato com a tarefa de segmentação de imagens usando U-Net
- No Moodle!





Avaliação Presencial

- Além do laboratório 7, temos o EA4 e uma avaliação presencial nesta semana!
- Fiquem de olho nos prazos mostrados no Moodle!



INTELIGÊNCIA
ARTIFICIAL &
CIÊNCIA DE DADOS

Bruno Légora Souza da Silva

Professor do Departamento de
Informática/UFES

bruno.l.silva@ufes.br