



Universidade Federal do Pará
Campus Universitário de Belém
Instituto de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica

Desbravando o Mundo da Visão Computacional: Introdução às Redes Neurais Convolucionais (CNN) e Algoritmos de Detecção de Objetos (YOLO)

AGENDA

1. Introdução à Visão Computacional
2. Fundamentos das Redes Neurais Convolucionais (CNN)
3. Algoritmos de Detecção de Objetos
4. Desafios e Boas Práticas em Visão Computacional
5. Aplicações Avançadas em Visão Computacional

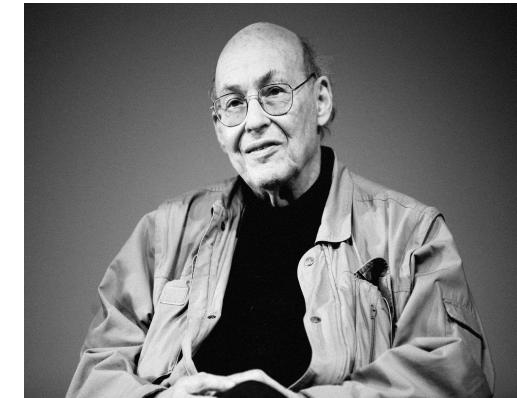
01

Introdução à Visão Computacional

Definição e Histórico da Visão Computacional

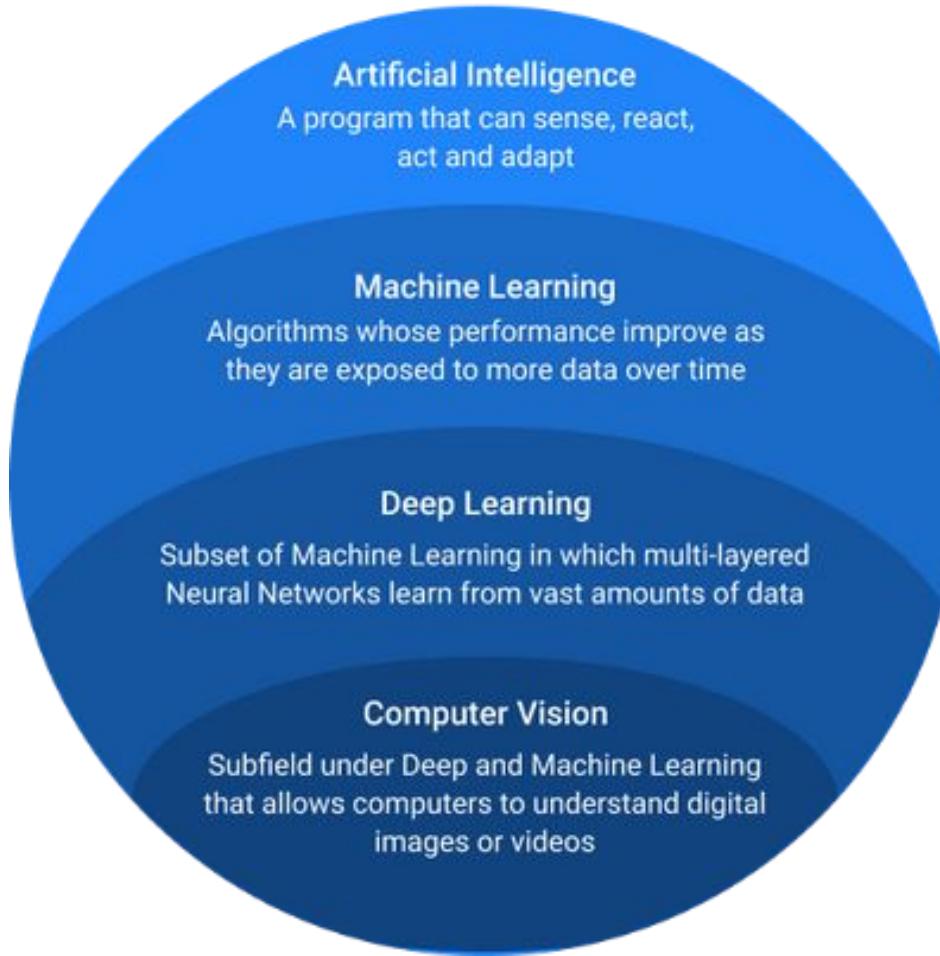
Área da inteligência artificial dedicada a permitir que máquinas "**vejam**" e **interpretem** o mundo visual ao seu redor

Desde a década de 1960, a visão computacional vem evoluindo com avanços em **hardware e algoritmos**, tornando possível a **análise automatizada de imagens e vídeos** em grande escala.



Marvin Minsky(1966):
“conectem uma câmera a
um computador e façam
ela descrever o que vê”

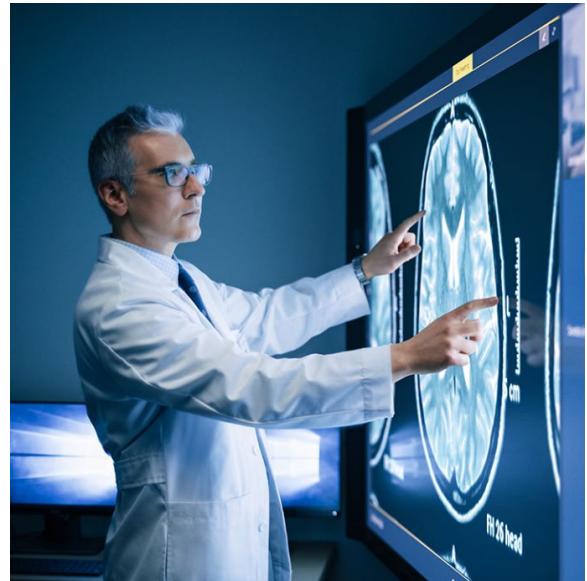
Deep Learning vs. Computer Vision



Áreas de Aplicação

Aplicações de visão computacional abrangem desde **reconhecimento facial** em smartphones até a segmentação de **imagens médicas para auxiliar diagnósticos**.

Além disso, é amplamente utilizada em **segurança, automação industrial e veículos autônomos**.



Principais Desafios e Tendências

Desafios: variações de iluminação, oclusão de objetos e a necessidade de **grande capacidade de processamento.**

Tendências: utilização de **redes neurais profundas** e **algoritmos de detecção de objetos.**

Discussão



Por que a visão
computacional é
importante?

...



Qual é a diferença
entre a visão
computacional e o
processamento de
imagem?



Como funciona a
visão
computacional?

...

Reconhecimento Humano x IA

01



02



03



04



05



06



Tarefas Fundamentais



Classificar

...



Detectar

...



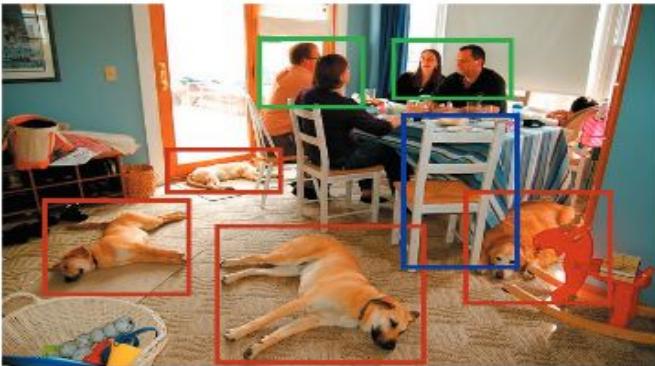
Segmentar

...

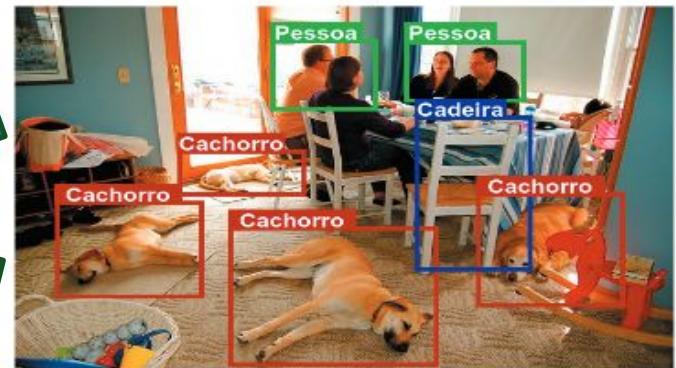
Classificação



Localização



Detecção



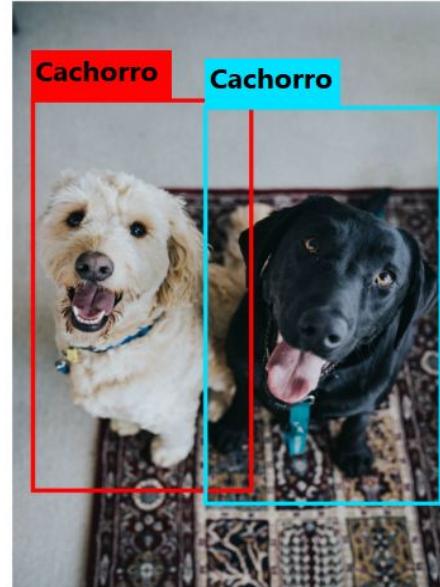
Diferenças entre classificação de imagens e detecção de objetos

Classificação de
imagens



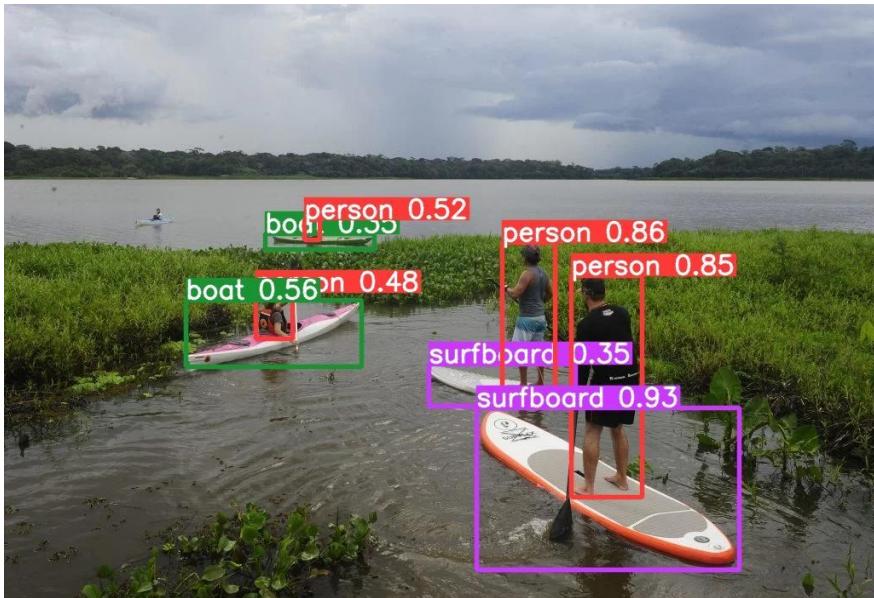
Cachorro

Detecção de objetos

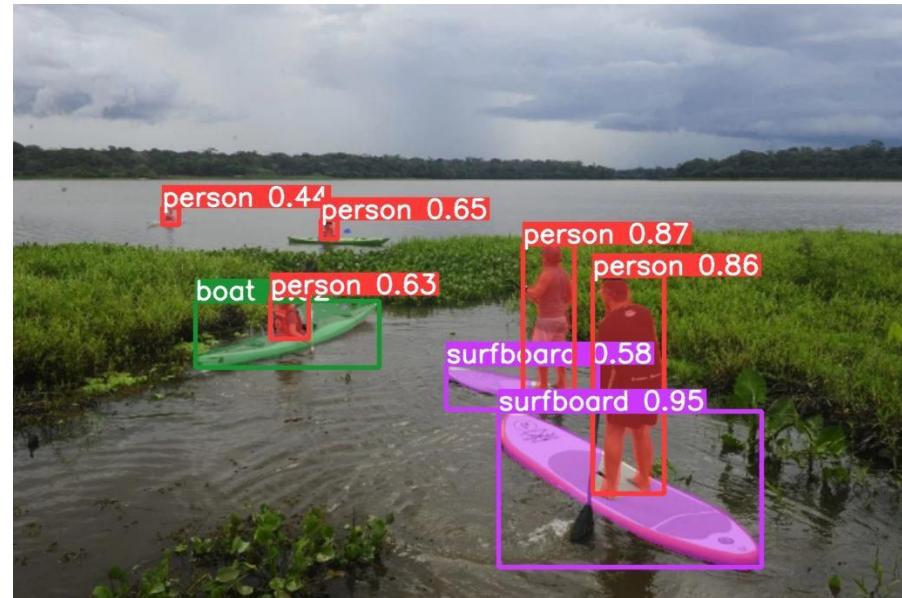


Exemplo de aplicação para as tarefas de detecção e segmentação

Detecção



Segmentação



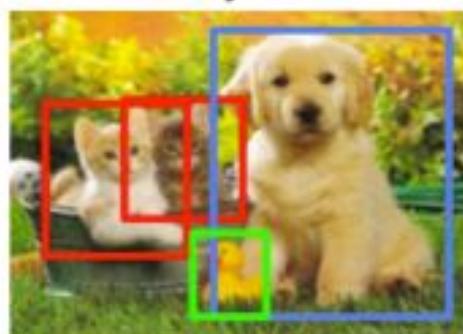
Classificação
de Imagem



Classificação +
Localização



Detecção de
Objetos



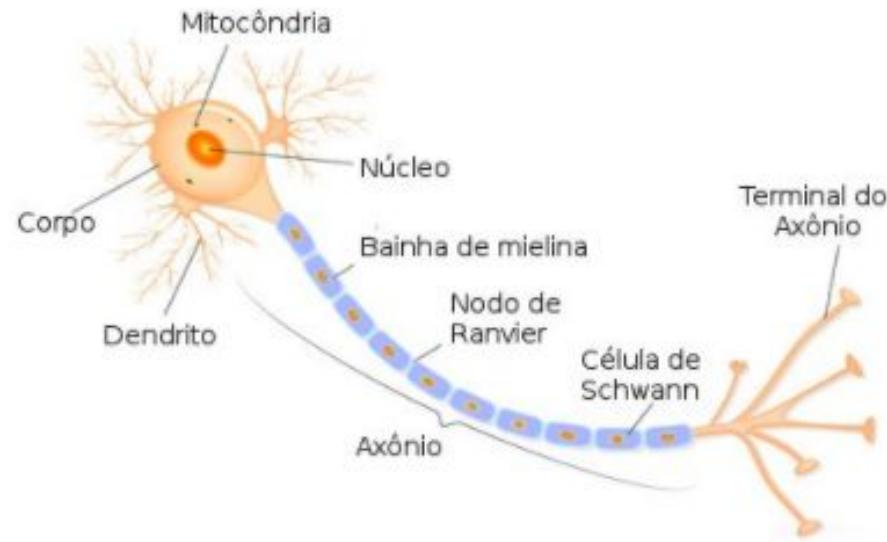
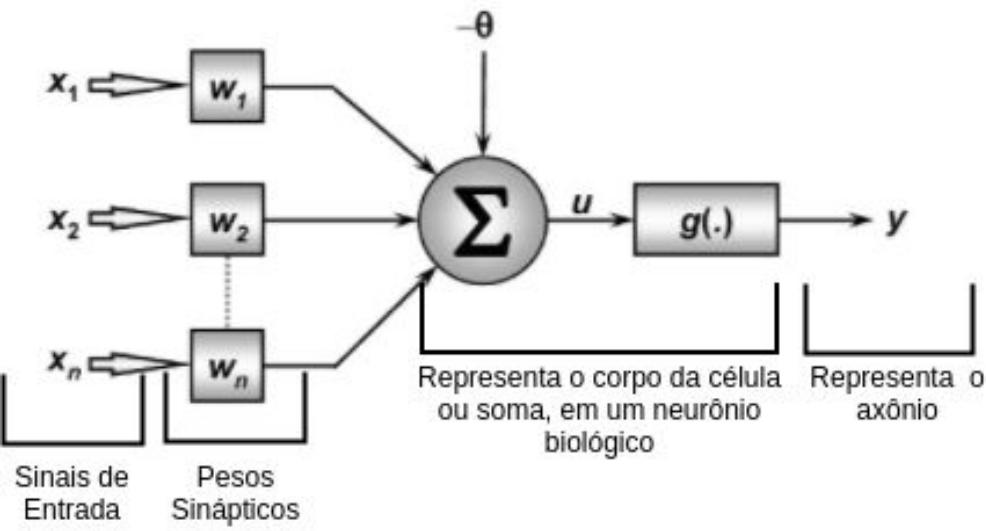
Segmentação
de Instâncias



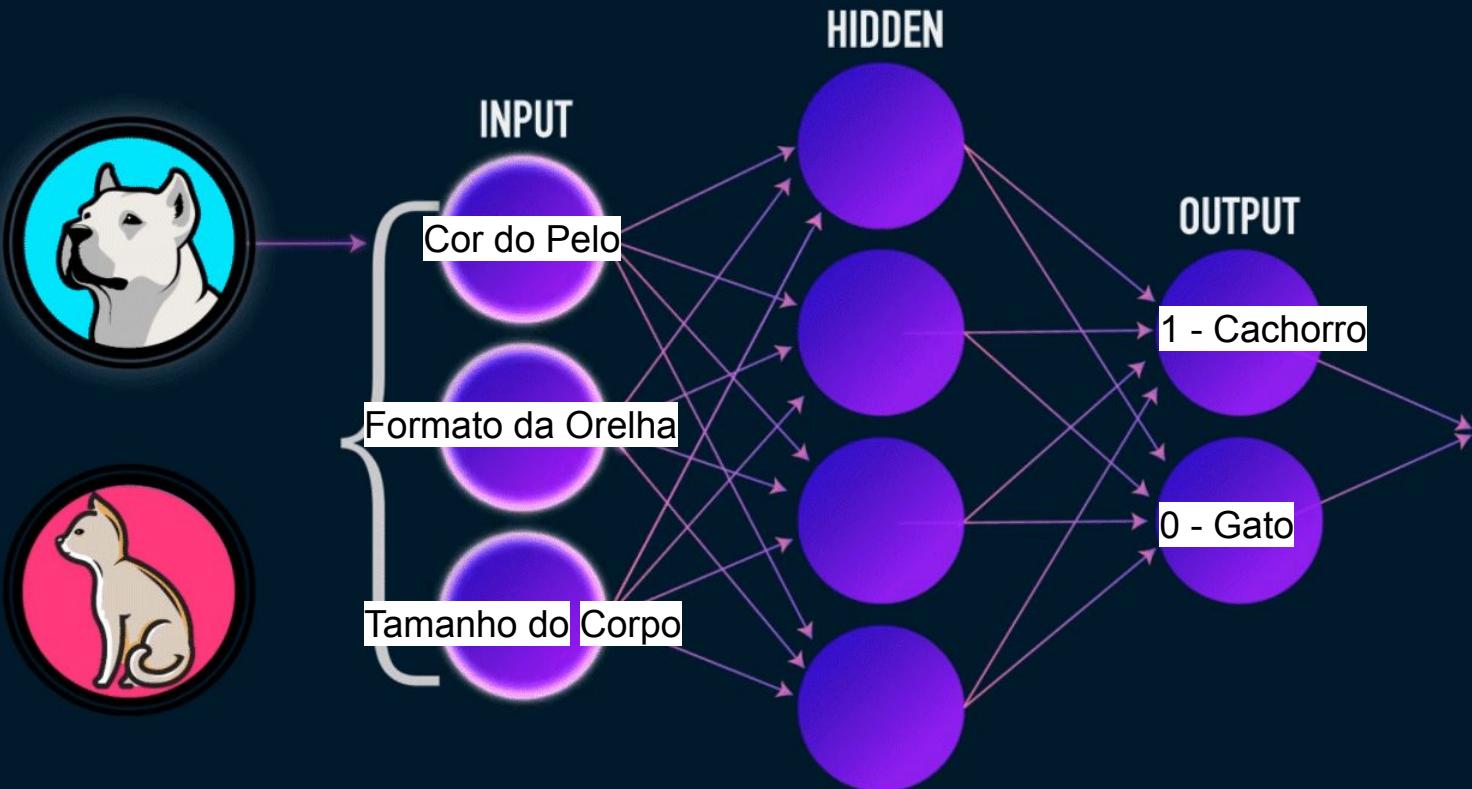
02

Fundamentos das Redes Neurais Convolucionais (CNN)

Neurônio Artificial



Em 1943, W. McCulloch e W. Pitts desenvolveram um modelo que propõe elementos computacionais retirados das **propriedades fisiológicas de um neurônio biológico** e de suas conexões, introduzindo assim a referência número um para a **teoria de Redes Neurais Artificiais**



Exemplos de área de aplicação de MLPs:

- **Auxílio ao Diagnóstico Médico**
- **Finanças e Previsão de Risco**
- **Processamento de Linguagem Natural (PLN)**
- **Sistemas de Controle e Automação**
- **Predição e Modelagem de Séries Temporais Simples**

<https://github.com/akhildevvr/Wine-Classification-using-MLP-on-MS-Excel>

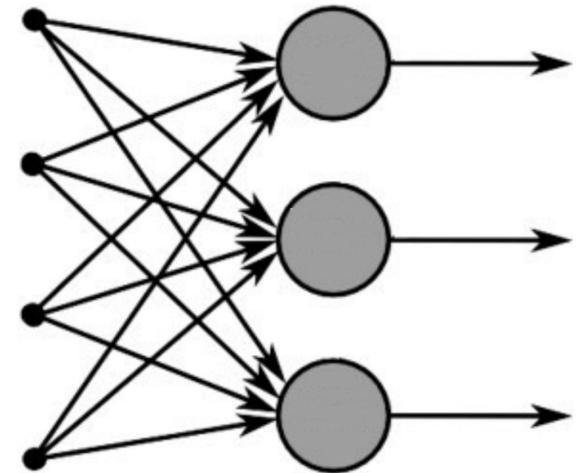
Redes Neurais

- **Redes Neurais**
 - Redes Neurais Feedforward - FNN
 - Redes Neurais Recorrentes - RNN
 - Long Short-Term Memory Networks - LSTM
 - Redes Generativas Adversariais - GAN
- **Redes Neurais Convolucionais (Convolutional Neural Networks - CNN)**
 - Classe especial de redes neurais profundas
 - Processamento de dados em formato de grade

Redes Neurais

Redes Neurais Feedforward – FNN

- Estrutura simples onde os dados fluem em uma única direção: **da entrada para a saída.**
- **Não têm memória interna**, ou seja, não retêm informações de estados anteriores.
- Utilizadas em tarefas como classificação de imagens e reconhecimento de padrões.

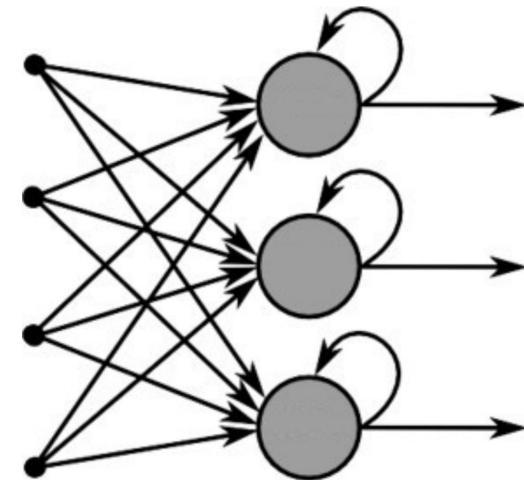


Feed-Forward Neural Network

Redes Neurais

Redes Neurais Recorrentes (RNN):

- Têm **ciclos** na rede, permitindo que a informação passada **influencie os cálculos futuros**.
- **Possuem memória interna**, o que permite que mantenham o contexto ao longo do tempo.

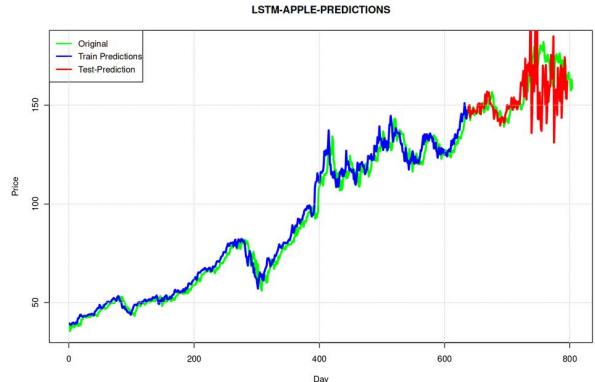
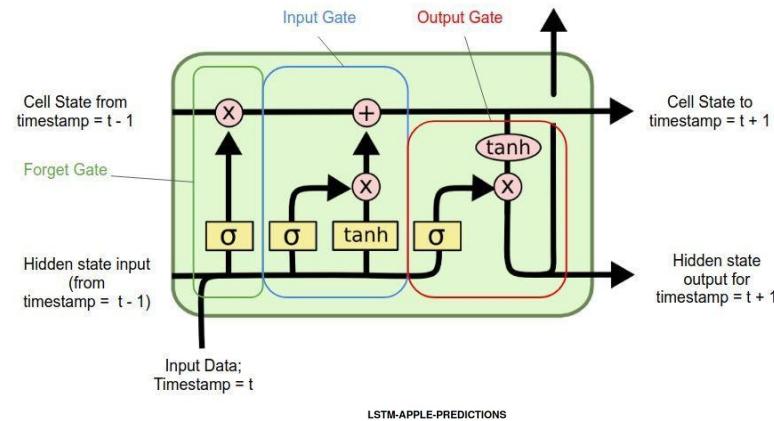


Recurrent Neural Network

Redes Neurais

Long Short-Term Memory (LSTM):

- Um tipo específico de RNN projetada para resolver problemas de **longo prazo**.
- Contém células de memória e mecanismos de controle para preservar informações relevantes por longos períodos.
- **Análise de séries temporais.**



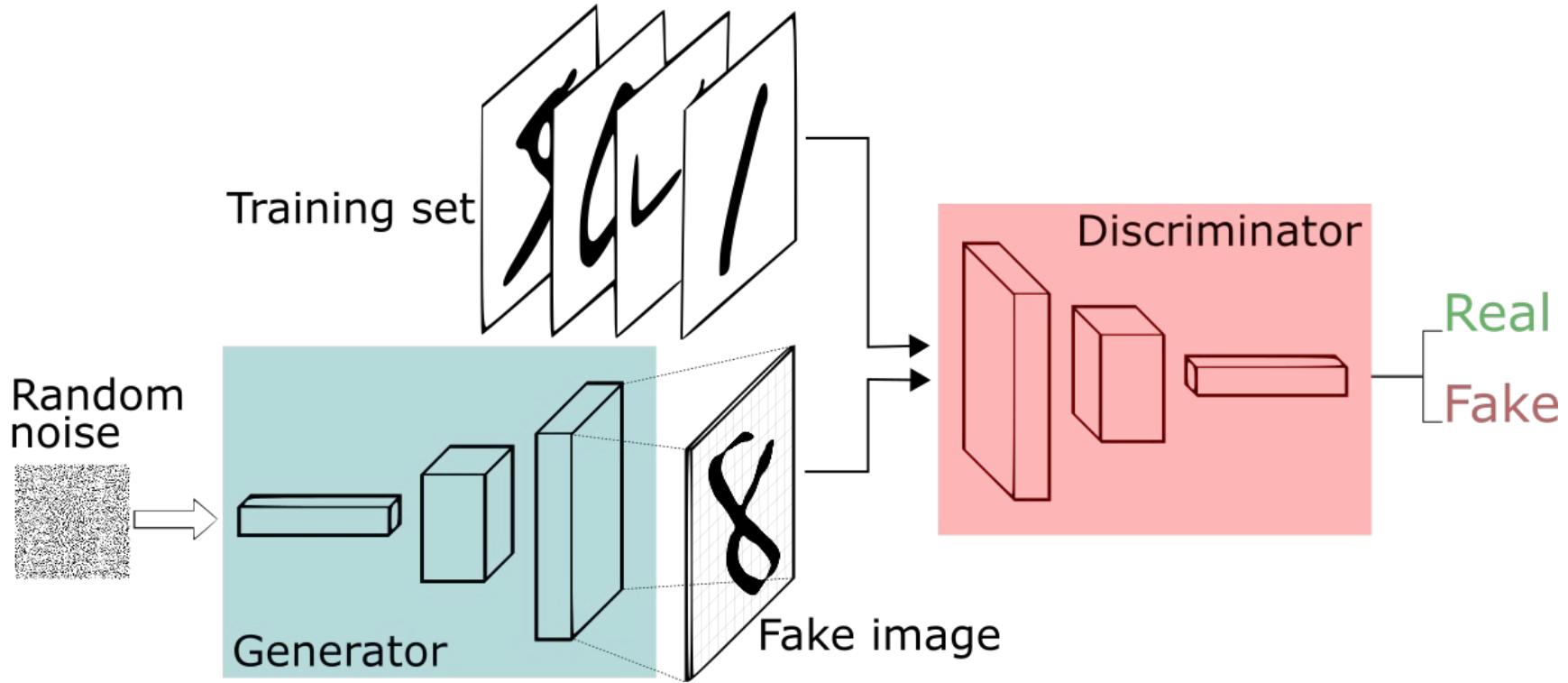
Redes Neurais

Redes Generativas Adversariais - GAN

- Compostas por duas redes: um **gerador e um discriminador**, que **competem** entre si.
- O **gerador** tenta criar **dados realistas**, enquanto o **discriminador** tenta **distinguir** dados gerados de dados reais.

Redes Neurais

Redes Generativas Adversariais - GAN



Introdução - CNN

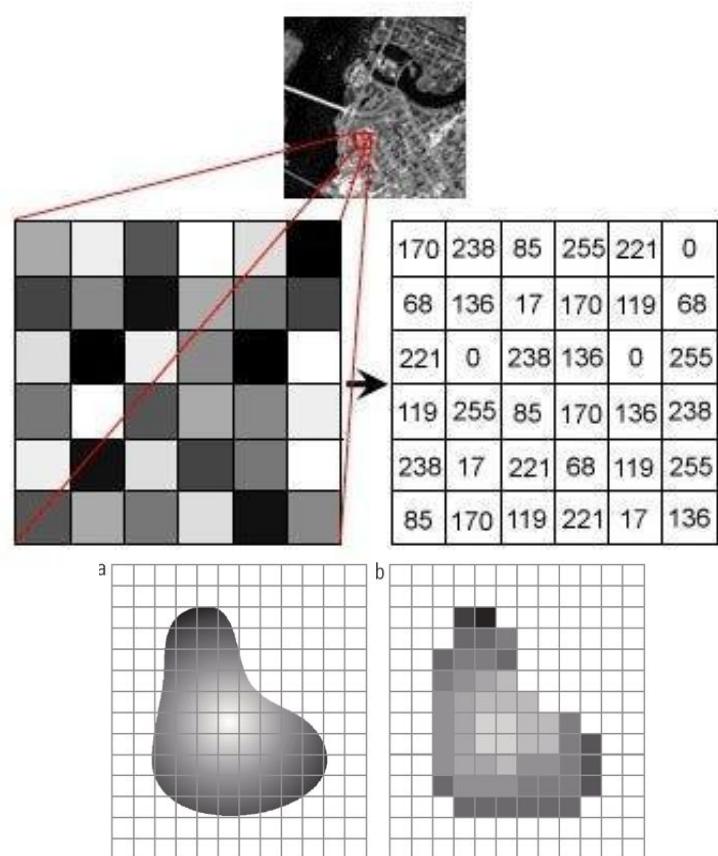
- **Função do córtex visual humano:** combinação de biologia, matemática e computação
- A primeira CNN, proposta em 1998 por **LeCun**, denominada LeNet
- Conjunto de Aprimorações e diversos modelos propostos
- **Empresas e CNN:** Google na pesquisa de fotos, Amazon na recomendação de produtos, Instagram na estrutura de pesquisa, Facebook no sistema de marcação automática

Introdução - CNN

- Utilizadas para **classificação de imagens**.
- Existem **diversas arquiteturas** de CNN, cada uma com suas **especificidades**;
- Como decidir qual a **melhor**?
 - Literatura;
 - Problemas similares;
 - Testes
- É necessário compreender como uma **imagem é representada computacionalmente**.

Representação Computacional de Imagens

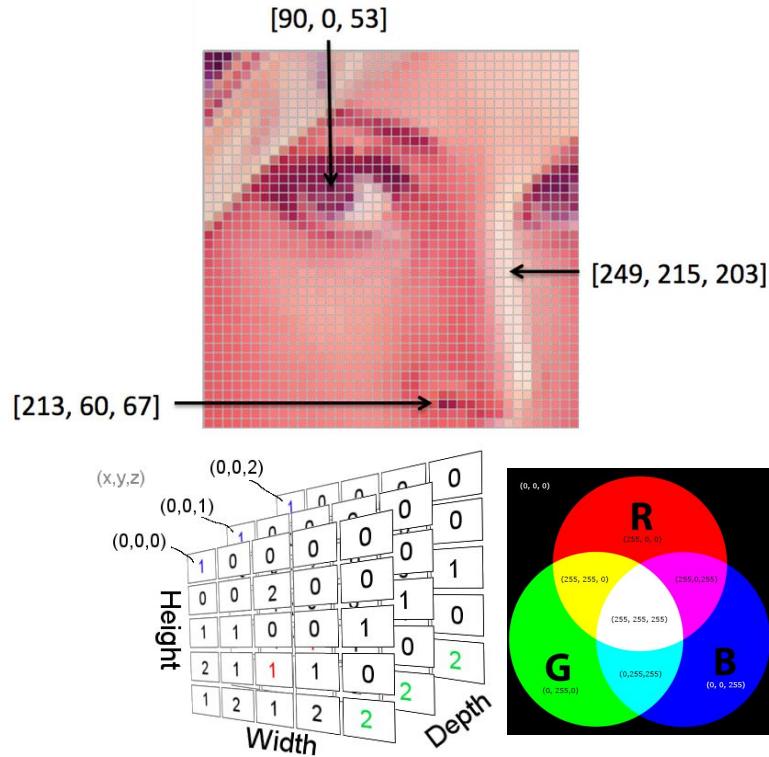
- **Imagen em preto e branco (grayscale):**
 - Representada como uma matriz 2D
 - Cada posição representa um pixel da imagem
 - Variação de cada pixel representa a intensidade de cor
 - 0 (preto)
 - 255 (branco)



Representação Computacional de Imagens

- **Imagen colorida:**

- Representada como uma matriz 3D
- Cada pixel é uma combinação das cores vermelho, verde e azul (RGB)



Curiosidade

A imagem "Lena" é bem conhecida por muitos pesquisadores de visão computacional. Originalmente, era uma ilustração de revista de 1972 com a modelo sueca *Lena Forsén*. A imagem foi escolhida por Alexander Sawchuk e sua equipe na Universidade da Califórnia do Sul em 1973, quando eles precisavam urgentemente de **uma imagem de alta qualidade** para um artigo de conferência.

Tecnicamente, áreas da imagem com detalhes ricos correspondem a sinais de alta frequência, que são mais difíceis de processar, enquanto sinais de baixa frequência são mais simples. **A imagem "Lena" possui uma riqueza de detalhes, contraste claro e escuro e áreas de transição suave, tudo em proporções adequadas, tornando-a um ótimo teste para algoritmos de compressão de imagem.**



<https://swe.org/magazine/letting-go-of-leena/>

https://www.reddit.com/r/computervision/comments/1byusil/ieee_computer_society_bans_leena_image_in_papers/?tl=pt-br&rdt=52958

Problemática com as Redes MLP Comuns

Dificuldade em Capturar Estruturas Espaciais e Temporais

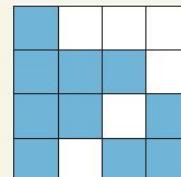
As redes MLP não capturam bem dependências espaciais (imagens, por exemplo) e temporais (dados sequenciais, como áudio e séries temporais), uma vez que cada neurônio na camada oculta recebe uma entrada como um vetor achataido e não mantém informações sobre a ordem ou posição dos dados.



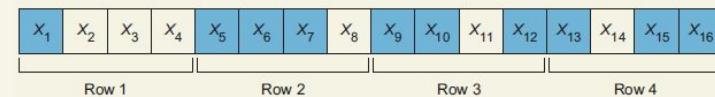
$$= 784 \text{ pixels}$$

Visualizing input vectors

To help visualize the flattened input vector, let's look at a much smaller matrix (4, 4):



The input (x) is a flattened vector with the dimensions (1, 16):



So, if we have pixel values of 0 for black and 255 for white, the input vector will be as follows:

Input = [0, 255, 255, 255, 0, 0, 0, 255, 0, 0, 255, 0, 0, 255, 0, 0]

Problemática com as Redes MLP Comuns

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

Figure 3.5 If the pixel value 1 is white and 0 is black, this is what our matrix looks like for identifying a square.

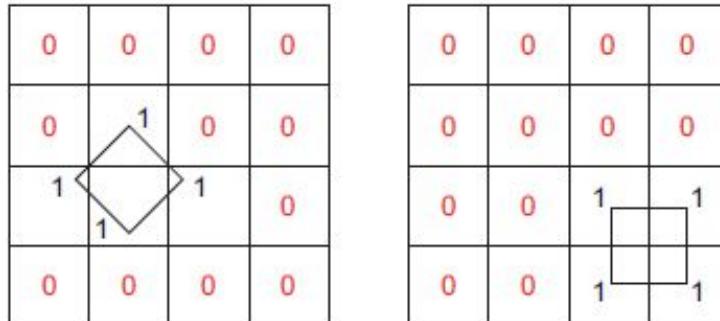
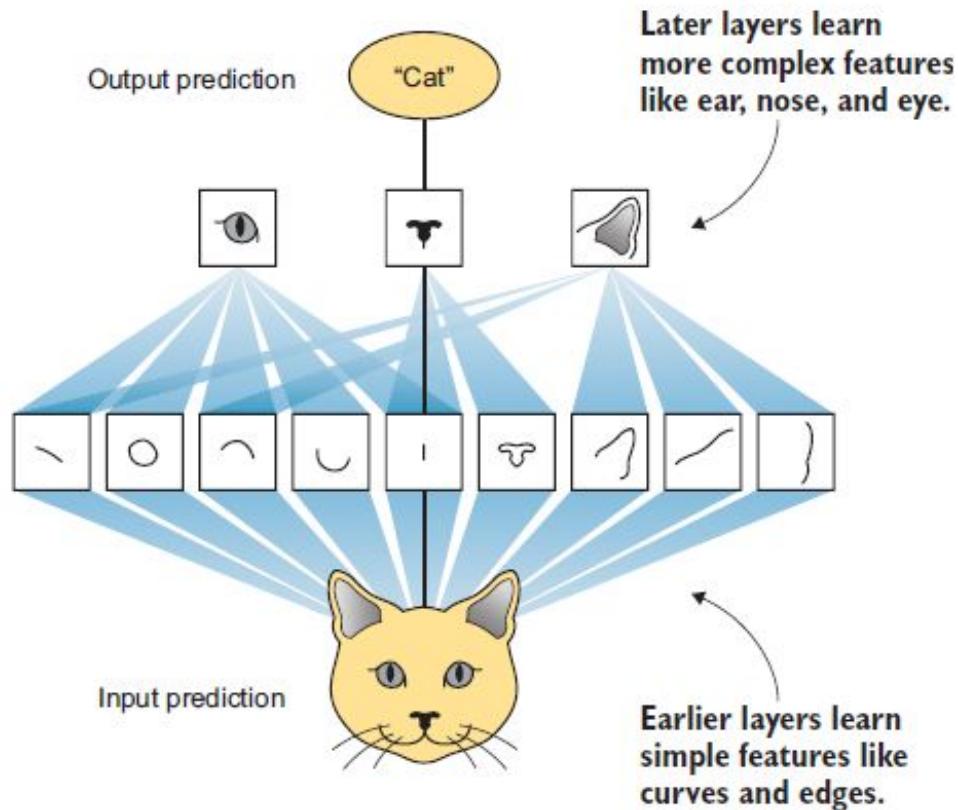


Figure 3.6 Square shapes in different areas of the image

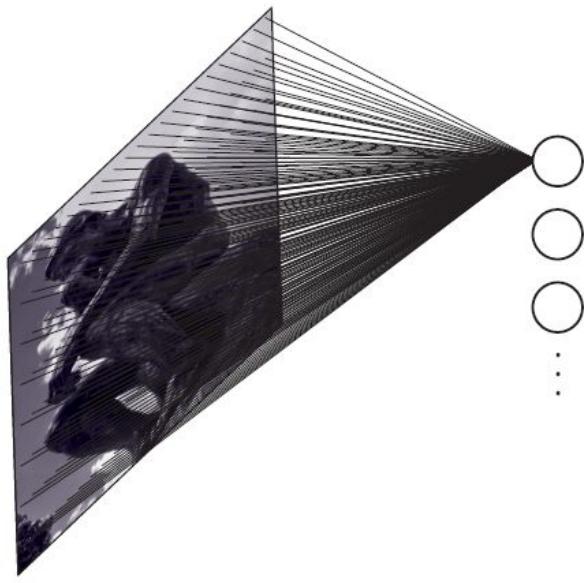
Problemática com as Redes MLP Comuns



Redes MLP geralmente precisam de **muitas camadas e neurônios para resolver problemas complexos**, o que aumenta drasticamente a quantidade de **parâmetros (pesos e biases)** que precisam ser treinados.

Problemática com as Redes MLP Comuns

Fully connected neural net



Locally connected neural net

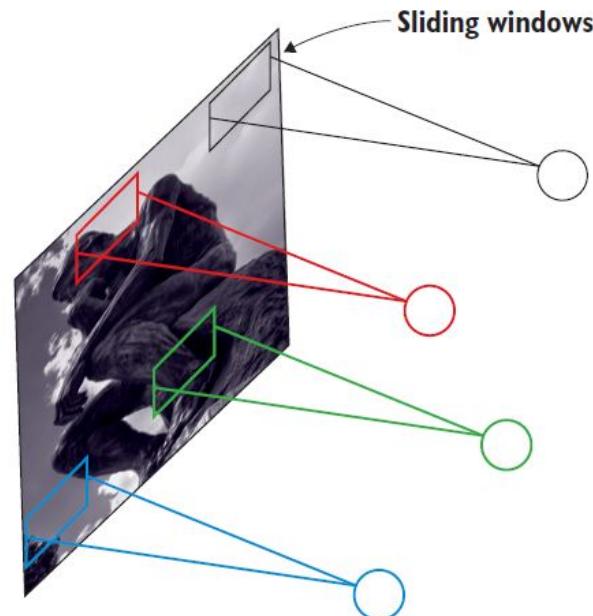
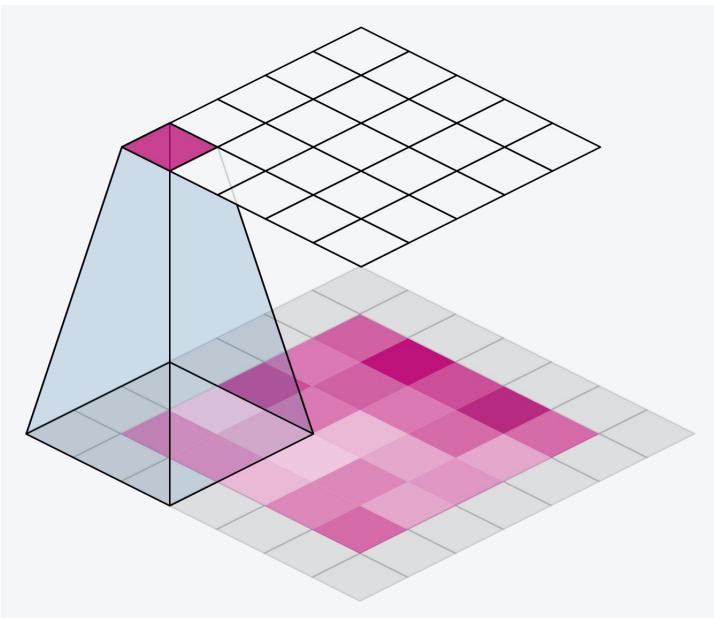


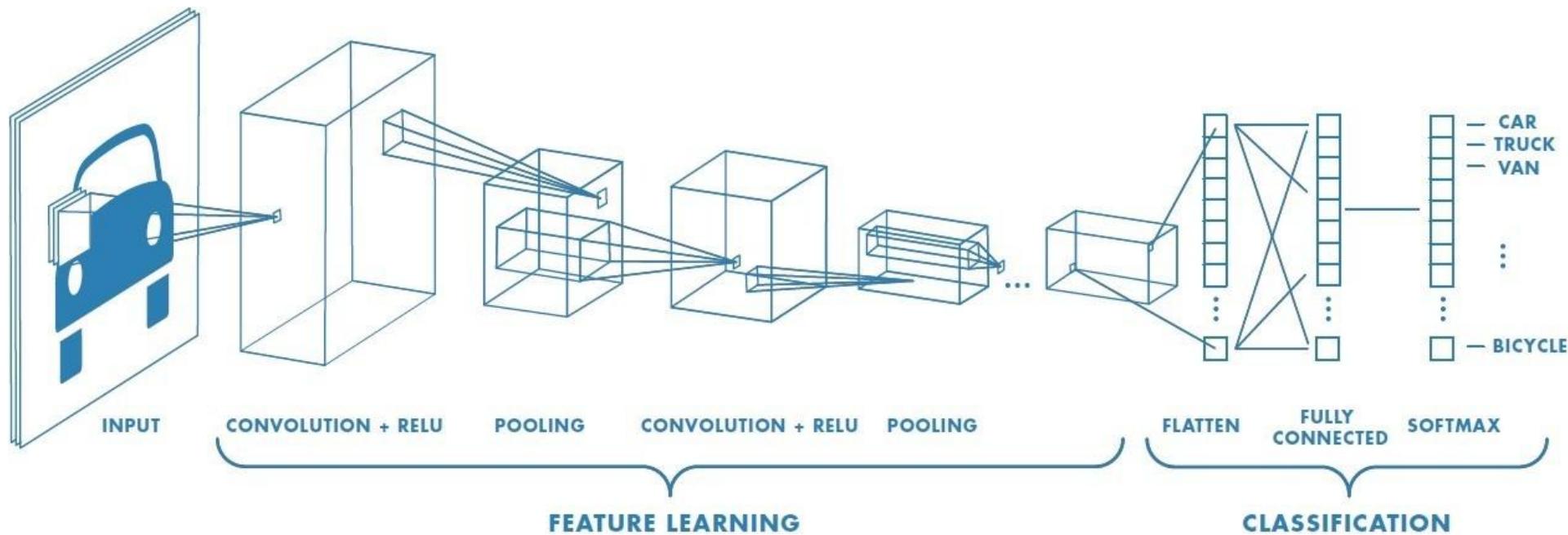
Figure 3.8 (Left) Fully connected neural network where all neurons are connected to all pixels of the Image. (Right) Locally connected network where only a subset of pixels is connected to each neuron. These subsets are called *sliding windows*.

Como funciona uma CNN?

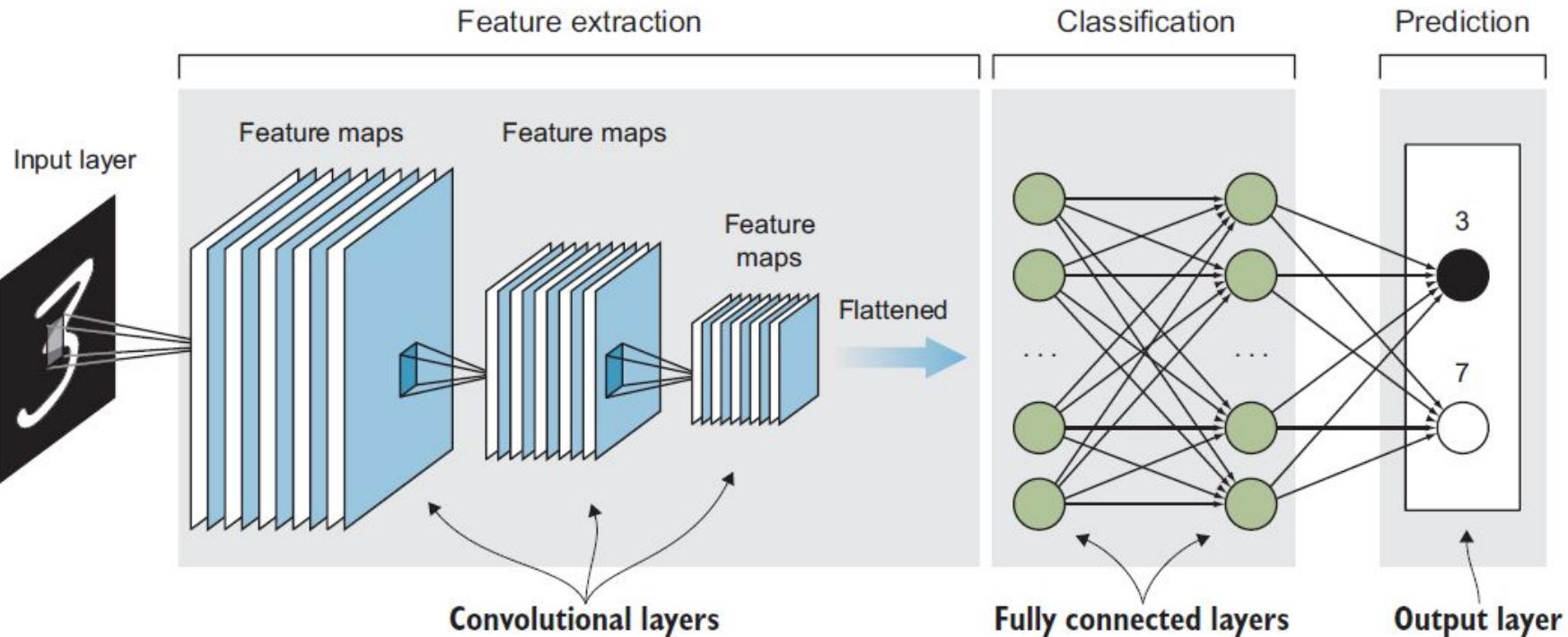


Ovelha

Estrutura da CNN



Estrutura da CNN



Camadas Convolucionais

Filtros são usados para **extrair características locais das entradas**. Camadas responsáveis por **aprender e detectar padrões** hierárquicos nas imagens, como bordas, texturas ou outros padrões.

Filtros ou Kernel:

É uma matriz pequena usada para **convoluir com uma imagem**. Na convolução, o kernel desliza sobre a imagem e **realiza operações locais**, destacando características específicas, **como bordas, texturas ou padrões**.

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

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

Camadas Convolucionais

Um kernel é uma matriz utilizada para uma operação de **multiplicação de matrizes**. Esta operação é aplicada diversas vezes em **diferentes regiões da imagem**.

A cada aplicação, a região é alterada por um parâmetro conhecido como **stride**.

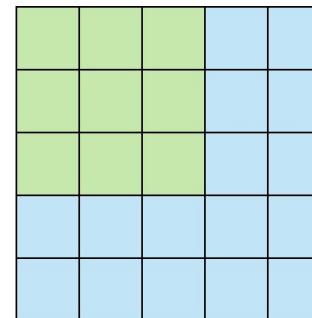
Normalmente o stride possui o valor 1, o que significa que a transformação será aplicada em todos os pixels da imagem.

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

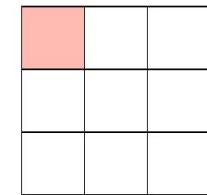
4		

Camadas Convolucionais

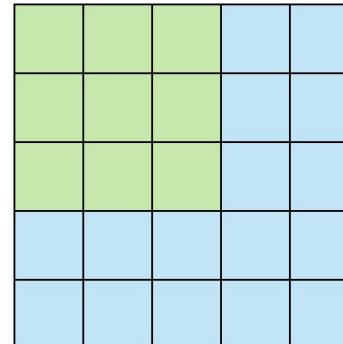
Stride: determina como o filtro de convolução é movido em cada etapa.



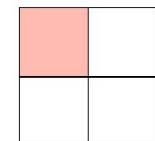
Stride 1



Feature Map



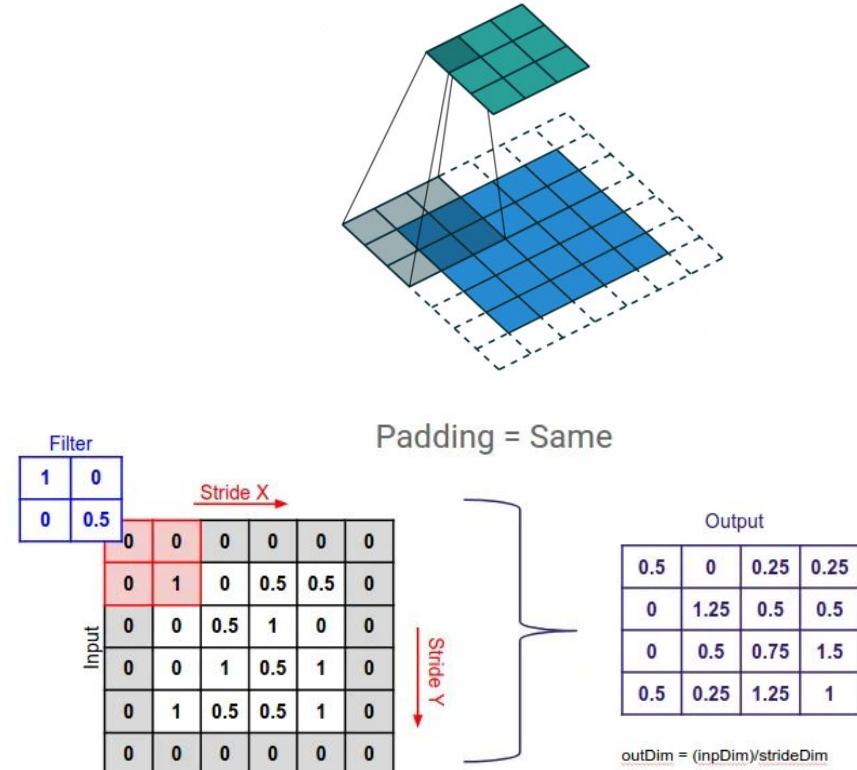
Stride 2



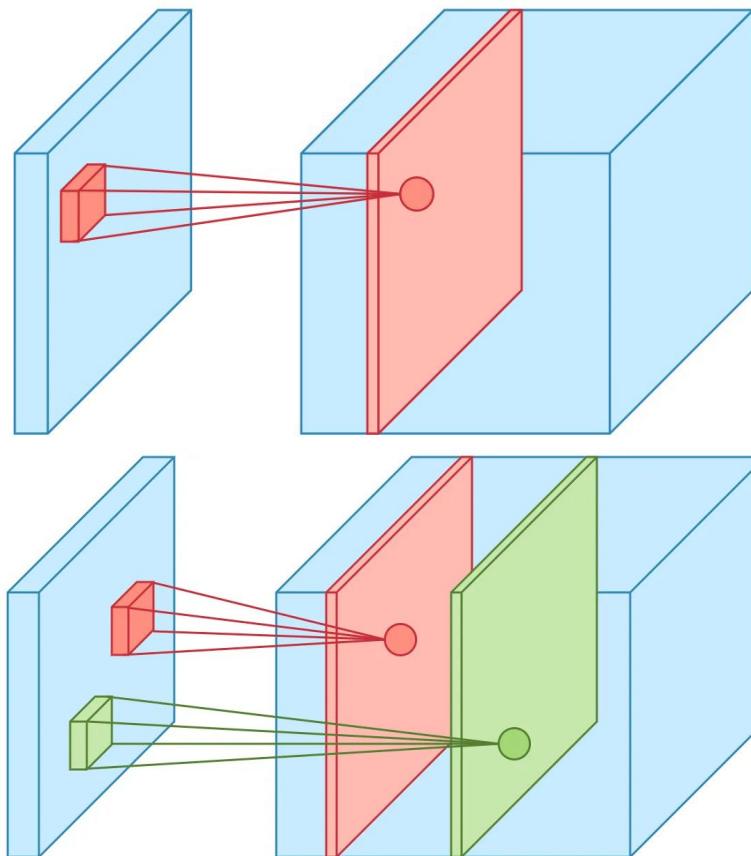
Feature Map

Camadas Convolucionais

Padding: é um processo em que **alguns pixels são adicionados ao redor da imagem** antes da operação de convolução, de forma a manter a dimensionalidade na imagem resultante durante a operação.



Camadas Convolucionais



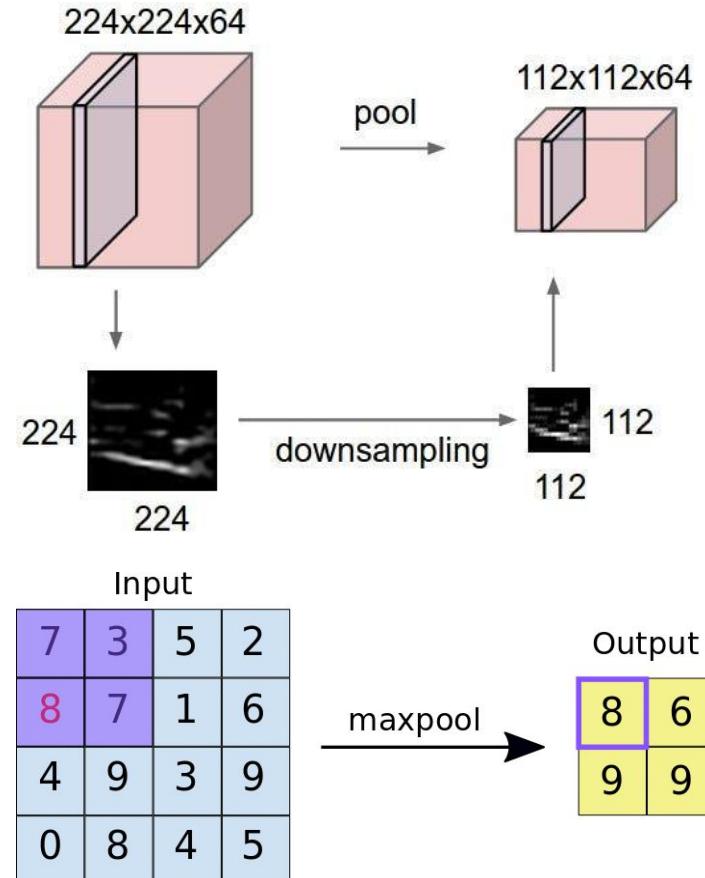
Operation	Kernel ω	Image result $g(x,y)$
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	

Camadas Convolucionais

Camada Pooling: É um processo simples de redução da dimensionalidade/features maps.

Visa **diminuir a variância** de pequenas alterações e a redução dos parâmetros treinados pela rede.

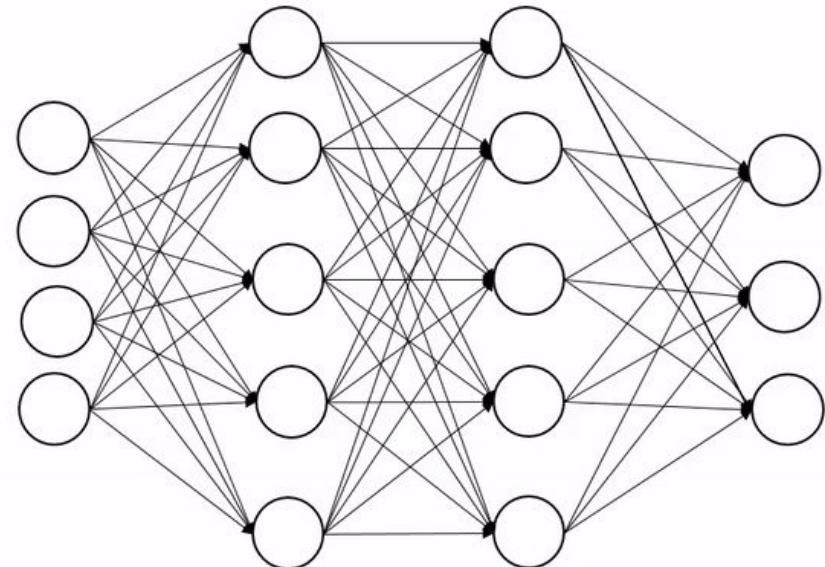
Destaca os valores máximos em regiões locais, preservando as características mais importantes e tornando a representação mais robusta.



Camadas Convolucionais

Camada Dropout: é utilizada para evitar que determinadas partes da rede neural tenham muita responsabilidade e possam ficar muito sensíveis a pequenas alterações.

Recebe um hiper parâmetro que define a probabilidade de **“desligar” determinada área da rede neural durante o processo de treinamento.**

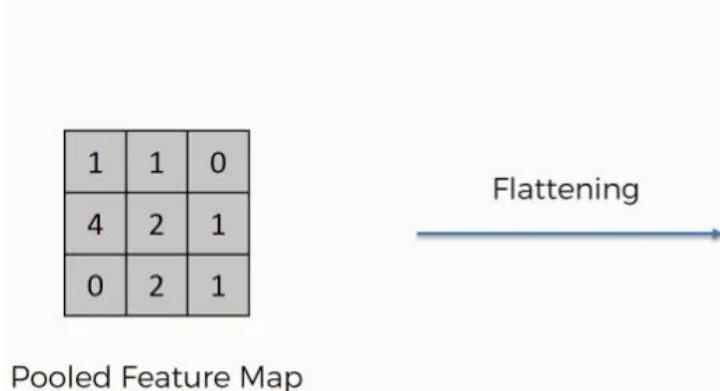


Camadas Convolucionais

Camada Flatten: Essa camada normalmente é utilizada na divisão das 2 partes da CNN (**extração de características / rede neural tradicional**).

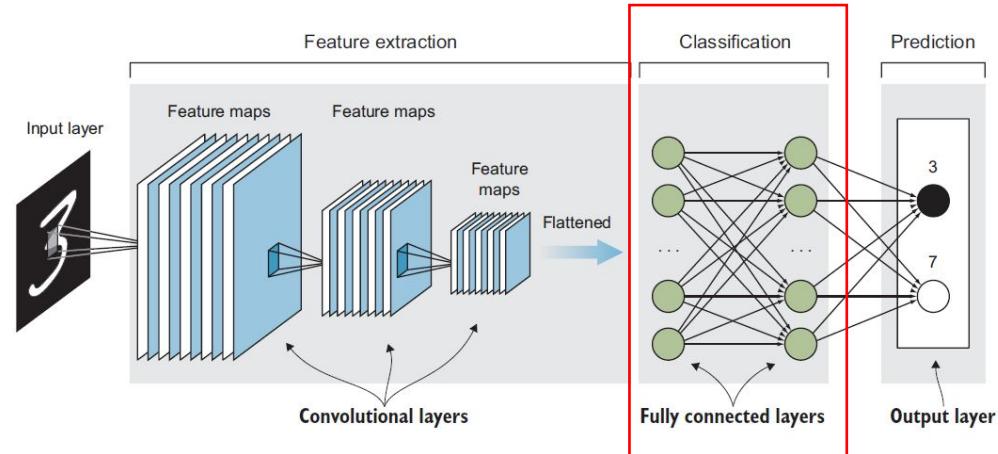
Opera uma transformação na matriz da imagem, alterando seu **formato para um vetor**.

Por exemplo, uma imagem em escala de cinza de **28x28** será transformada para um vetor de **784** posições.



Estrutura da CNN

Camadas totalmente conectadas (densas): Essas camadas são semelhantes às utilizadas em redes neurais tradicionais e são responsáveis por **combinar as características extraídas** em uma **representação final** para tomada de decisões.

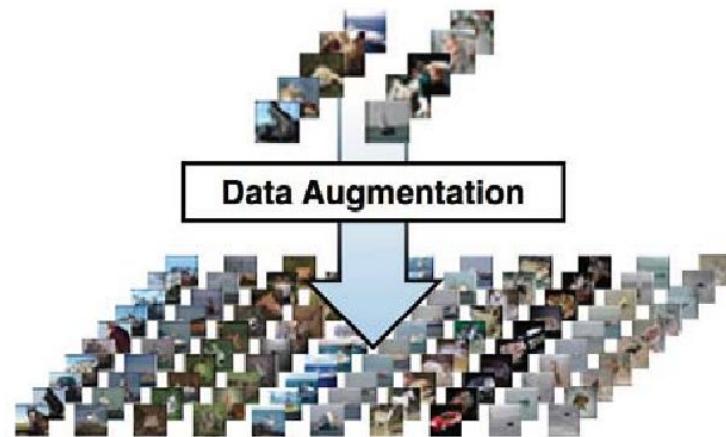


Técnicas de Treinamento

Transfer Learning



Data augmentation



Transferência de Aprendizado

A transferência de aprendizado é uma técnica que utiliza o conhecimento adquirido por uma rede **pré-treinada** em uma tarefa específica e o **aplica** em uma nova tarefa relacionada.

COCO (Common Objects in Context)

330.000 imagens, devidamente rotuladas entre 80 categorias de objetos.

Transferência de Aprendizado

Reutilizar rede pré-treinada

Carregar rede pré-treinada

Camadas Iniciais que aprendem características de baixo nível (bordas, bolhas, cores)



1 milhão de imagens
milhares de classes

Últimas camadas que aprendem características específicas



Substituir camadas finais

camadas novas para aprender características específicas do seu conjunto de dados



Menos classes
Aprenda mais rápido

Treinar a rede

Imagens de treinamento
...
Opções de treinamento



Centenas de imagens
Dezenas de classes

Prever e avaliar a precisão da rede



Rede treinada

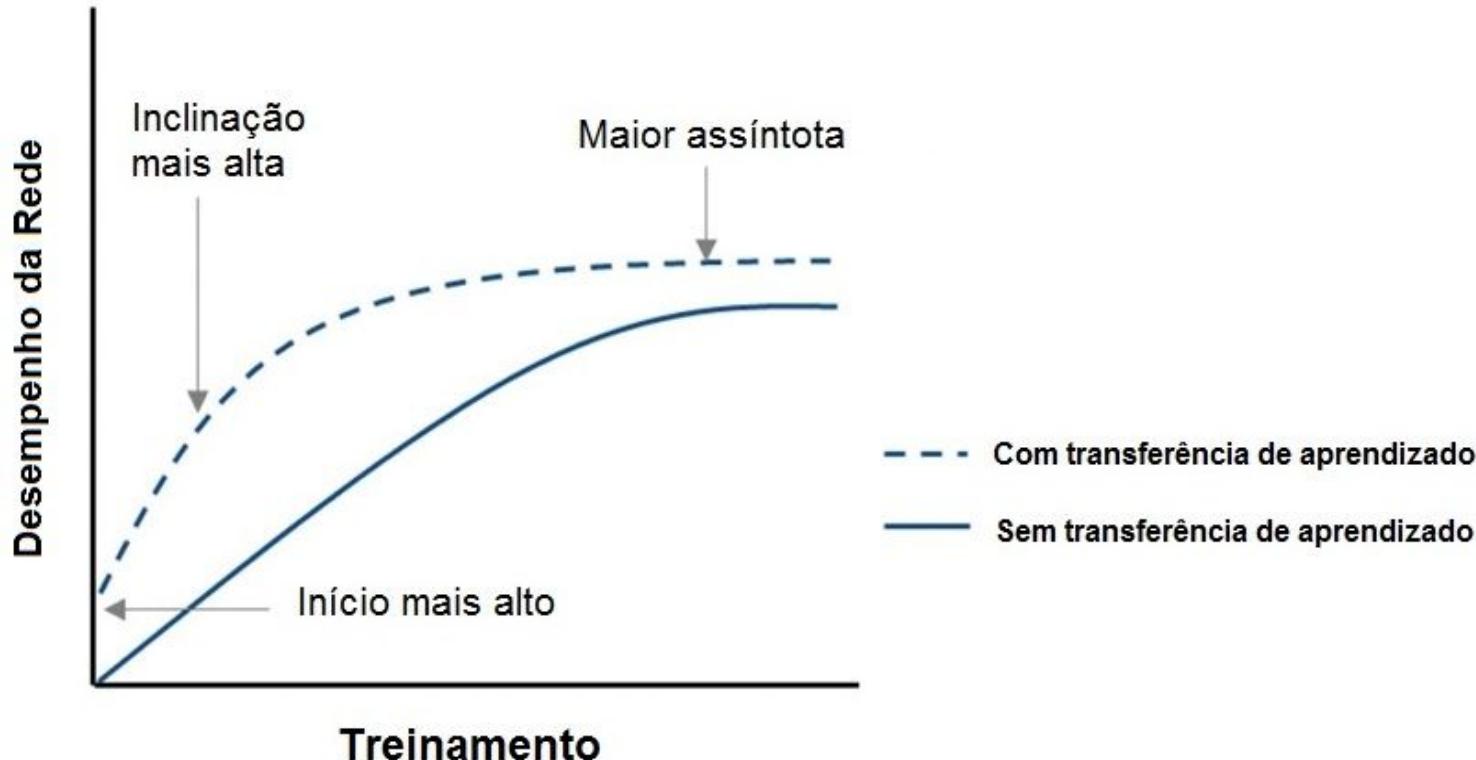
imagens de teste

Implantar resultados



melhorar a rede

Transferência de Aprendizado



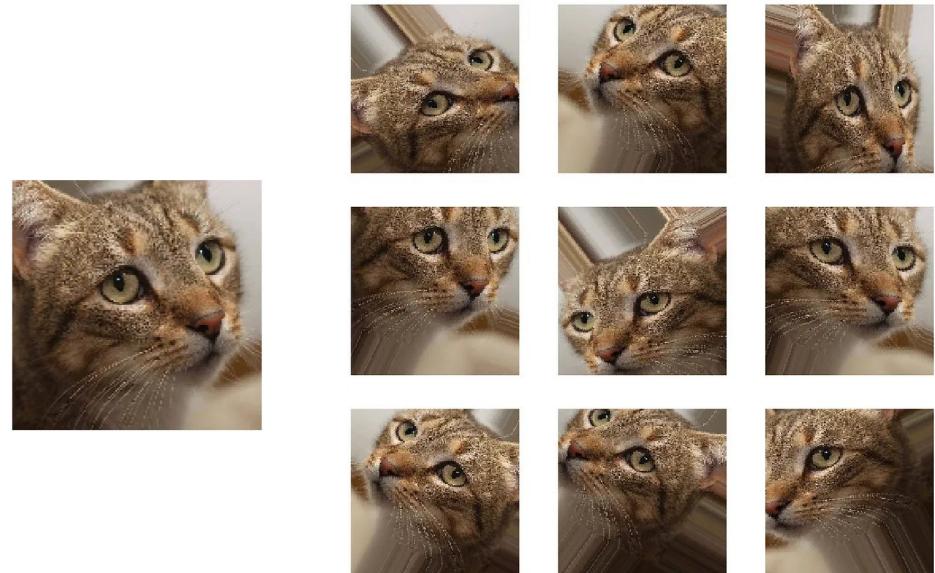
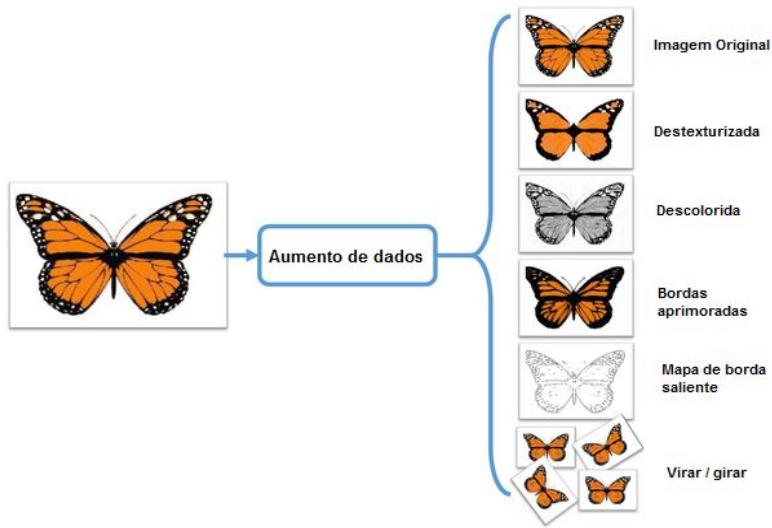
Aumento de Dados

A técnica de aumento de dados é ideal para:

- Conjuntos de dados limitados
- Situações de dificuldades para obtenção de novos dados e rotulação das imagens.

De forma geral, essa técnica envolve a **aplicação de operações aleatórias (e.g. rotação, reflexão e cortes)**

Aumento de Dados



Parâmetros de Treinamento da Rede

Otimizadores

- Gradiente
- Taxa de aprendizado;
- Momentum;

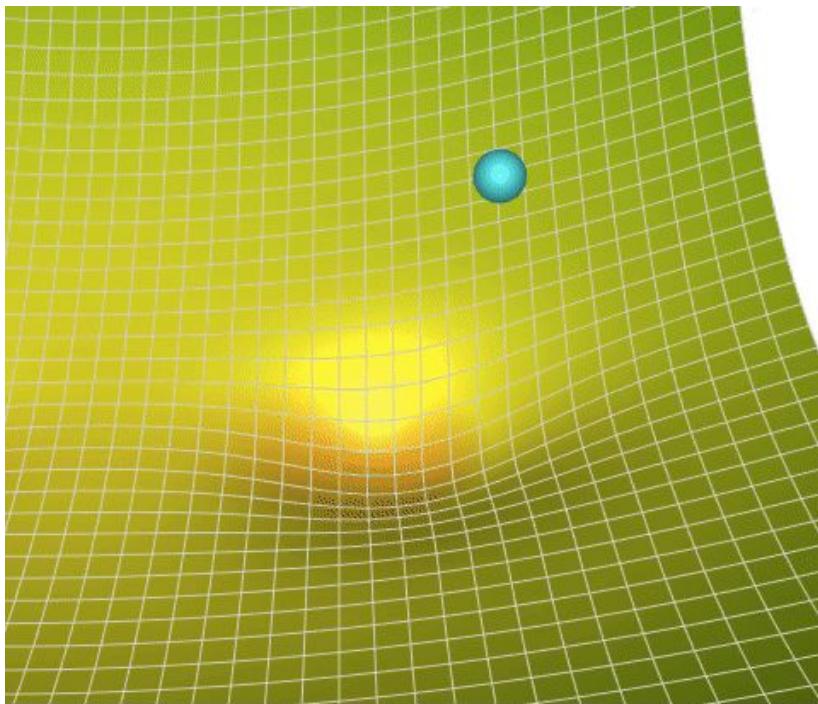


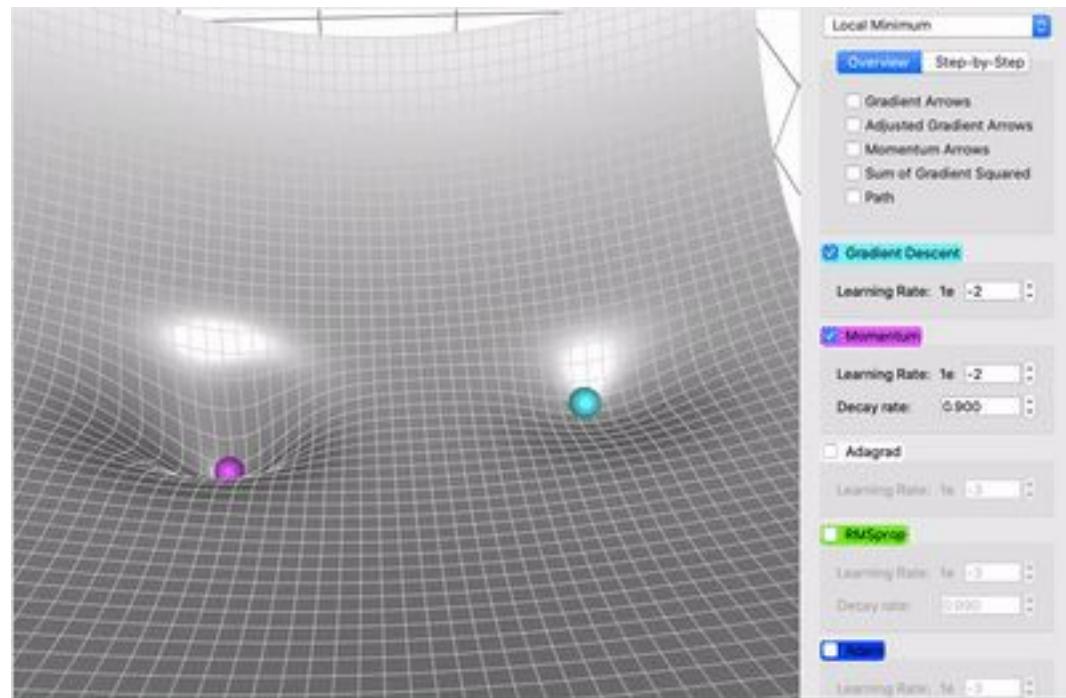
Ilustração passo a passo do algoritmo de *Gradient Descent*.

Parâmetros de Treinamento da Rede

Otimizadores

ADAM (Adaptive Moment Estimation)

SGD (Stochastic Gradient Descent).



Momentum (magenta) vs. Gradient Descent (ciano) em uma superfície com um mínimo global (esquerda) e um mínimo local (direita)..

TensorFlow: taxa de aprendizado: 0,001 e momentum: 0,9.

Parâmetros de Treinamento da Rede

Rótulos e saída do modelo

- **Binary Crossentropy:**
 - Usada para classificação binária (2 classes).
 - A saída do modelo é um único valor (geralmente entre 0 e 1, após uma função Sigmoid).
 - Exemplo: Classificação entre "positivo" e "negativo".
- **Categorical Crossentropy:**
 - Usada para classificação multiclasse onde há mais de 2 classes.
 - A saída do modelo é um vetor one-hot (um vetor onde a classe correta tem valor 1 e as outras são 0).
 - Exemplo: Classificação de imagens em 3 categorias (gato, cachorro, pássaro).
- **Sparse Categorical Crossentropy:**
 - Também usada para classificação multiclasse, mas com as classes representadas por rótulos inteiros no invés de one-hot.

Parâmetros de Treinamento da Rede

Função de ativação: Frequentemente utilizada para introduzir **não linearidades** na rede.

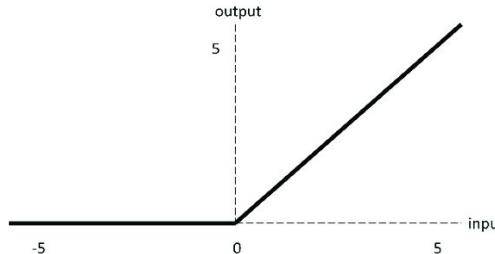
Convolução:

ReLU (Rectified Linear Unit): **Transforma valores negativos em zero**, o que ajuda a manter a informação relevante enquanto elimina ruído e acelera a convergência do treinamento.

1	14	-9	4
-2	-20	10	6
-3	3	11	1
2	54	-2	80

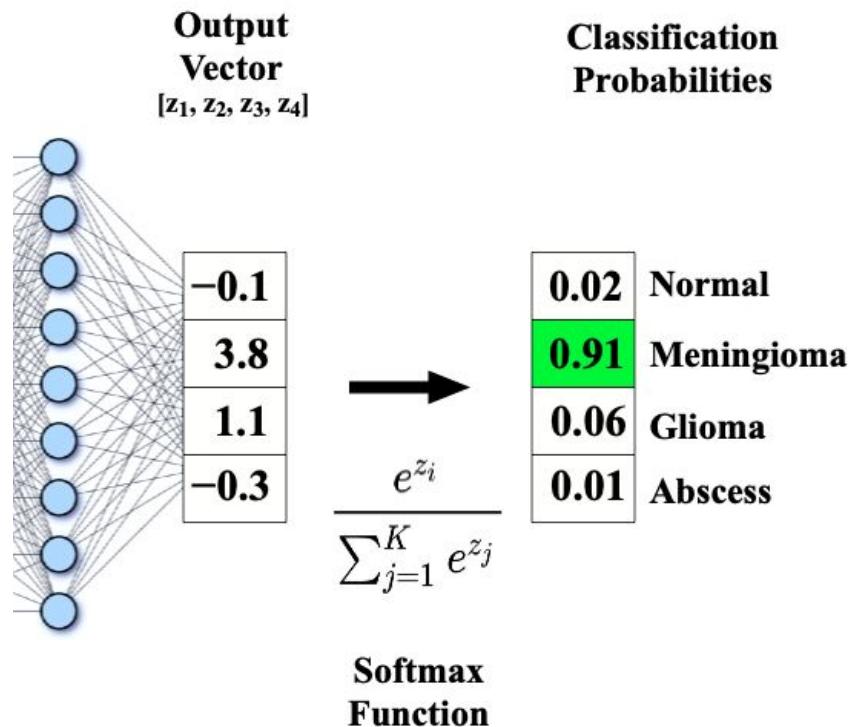
ReLU

1	14	0	4
0	0	10	6
0	3	11	1
2	54	0	80



Parâmetros de Treinamento da Rede

A Softmax é usada na camada de saída para **problemas de classificação multiclasse**. Ela transforma um vetor de valores numéricos em probabilidades, onde a **soma das probabilidades é 1**.



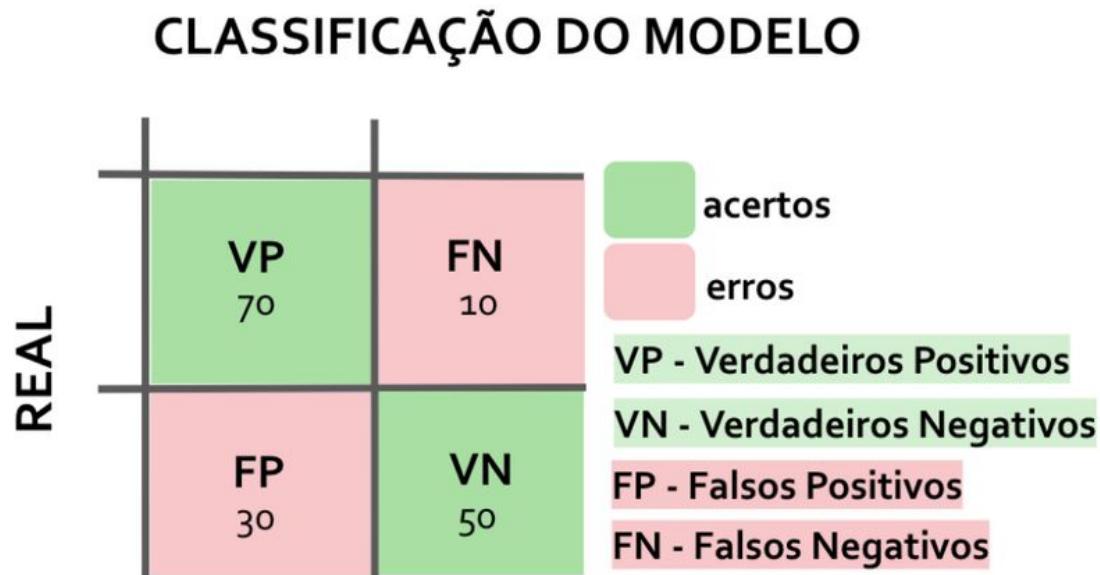
Desempenho da CNN

Função de Perda:

As funções de perda são utilizadas para medir quão bem o modelo está performando em uma tarefa específica

Métricas de Avaliação:

As principais são: Acurácia, Matriz de Confusão, Precisão, Revocação, F1-Score



Desempenho da CNN

Confusion Matrix												
Output Class	0	1	2	3	4	5	6	7	8	9		
	0	481 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	99.6% 0.4%	
	1	0 0.0%	468 9.4%	2 0.0%	0 0.0%	3 0.1%	0 0.0%	6 0.1%	1 0.0%	0 0.0%	11 0.2%	95.3% 4.7%
	2	8 0.2%	19 0.4%	483 9.7%	5 0.1%	0 0.0%	0 0.0%	4 0.1%	8 0.2%	6 0.1%	2 0.0%	90.3% 9.7%
	3	0 0.0%	1 0.0%	1 0.0%	468 9.4%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	6 0.1%	4 0.1%	97.3% 2.7%
	4	0 0.0%	2 0.0%	0 0.0%	0 0.0%	493 9.9%	0 0.0%	1 0.0%	0 0.0%	1 0.0%	1 0.0%	99.0% 1.0%
	5	0 0.0%	1 0.0%	0 0.0%	18 0.4%	0 0.0%	498 10.0%	12 0.2%	1 0.0%	6 0.1%	1 0.0%	92.7% 7.3%
	6	4 0.1%	3 0.1%	4 0.1%	0 0.0%	0 0.0%	1 0.0%	461 9.2%	0 0.0%	3 0.1%	2 0.0%	96.4% 3.6%
	7	0 0.0%	6 0.1%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	487 9.7%	0 0.0%	1 0.0%	98.0% 2.0%
	8	0 0.0%	0 0.0%	1 0.0%	5 0.1%	2 0.0%	1 0.0%	1 0.0%	0 0.0%	473 9.5%	2 0.0%	97.5% 2.5%
	9	7 0.1%	0 0.0%	7 0.1%	4 0.1%	2 0.0%	0 0.0%	12 0.2%	3 0.1%	4 0.1%	476 9.5%	92.4% 7.6%
96.2% 3.8%												
93.6% 6.4%												
96.6% 3.4%												
93.6% 6.4%												
98.6% 1.4%												
99.6% 0.4%												
92.2% 7.8%												
97.4% 2.6%												
94.6% 5.4%												
95.2% 4.8%												
95.8% 4.2%												

Target Class

Desempenho da CNN

CLASSIFICAÇÃO DO MODELO

		REAL	
		VP 70	FN 10
CLASSIFICADO	VP	acertos	
	FN	erros	
CLASSIFICADO	FP 30	VP - Verdadeiros Positivos	
	VN 50	VN - Verdadeiros Negativos	
CLASSIFICADO	FP 30	FP - Falsos Positivos	
	VN 50	FN - Falsos Negativos	

$$\text{Acurácia} = \frac{\checkmark \text{👍 TP} + \checkmark \text{👎 TN}}{\checkmark \text{👍 TP} + \checkmark \text{👎 TN} + \times \text{👍 FP} + \times \text{👎 FN}}$$

$$\text{Precisão} = \frac{\checkmark \text{👍 TP}}{\checkmark \text{👍 TP} + \times \text{👍 FP}}$$

$$\text{Revocação} = \frac{\checkmark \text{👍 TP}}{\checkmark \text{👍 TP} + \times \text{👎 FN}}$$

$$F_1 = 2 * \frac{\text{precisão} * \text{recall}}{\text{precisão} + \text{recall}}$$

Desempenho da CNN

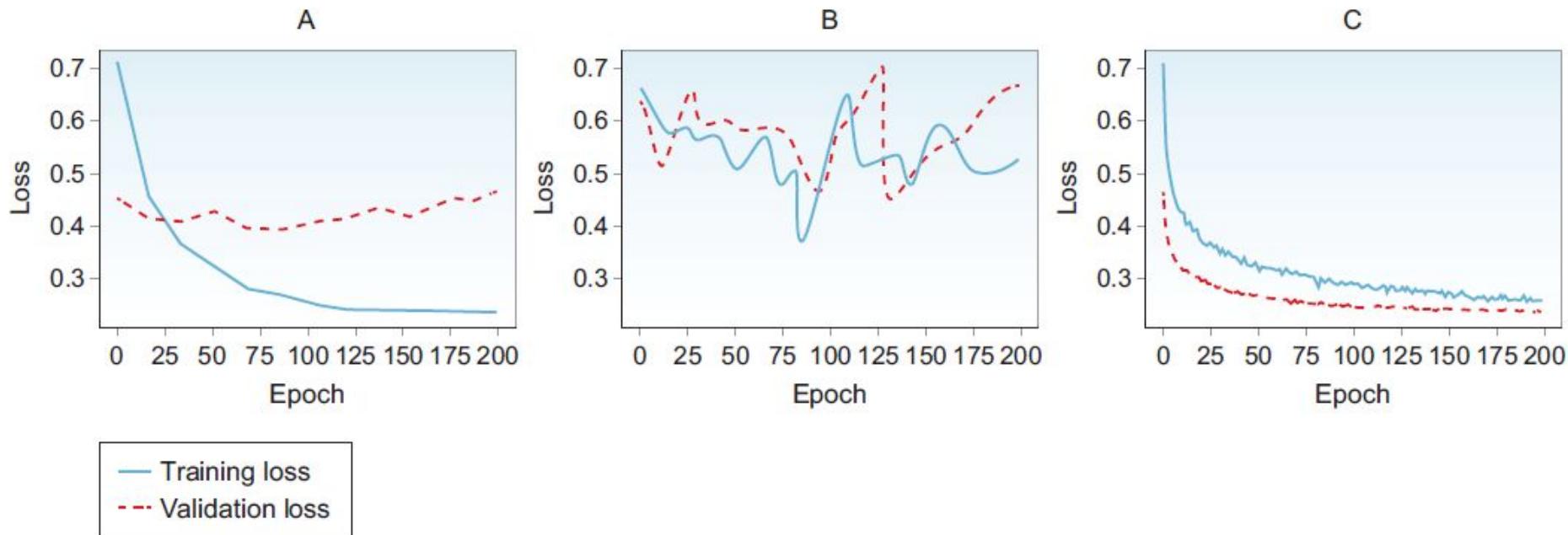


Figure 4.10 (A) The network improves the loss value on the training data but fails to generalize on the validation data. (B) The network performs poorly on both the training and validation data. (C) The network learns the training data and generalizes to the validation data.

Desempenho da CNN

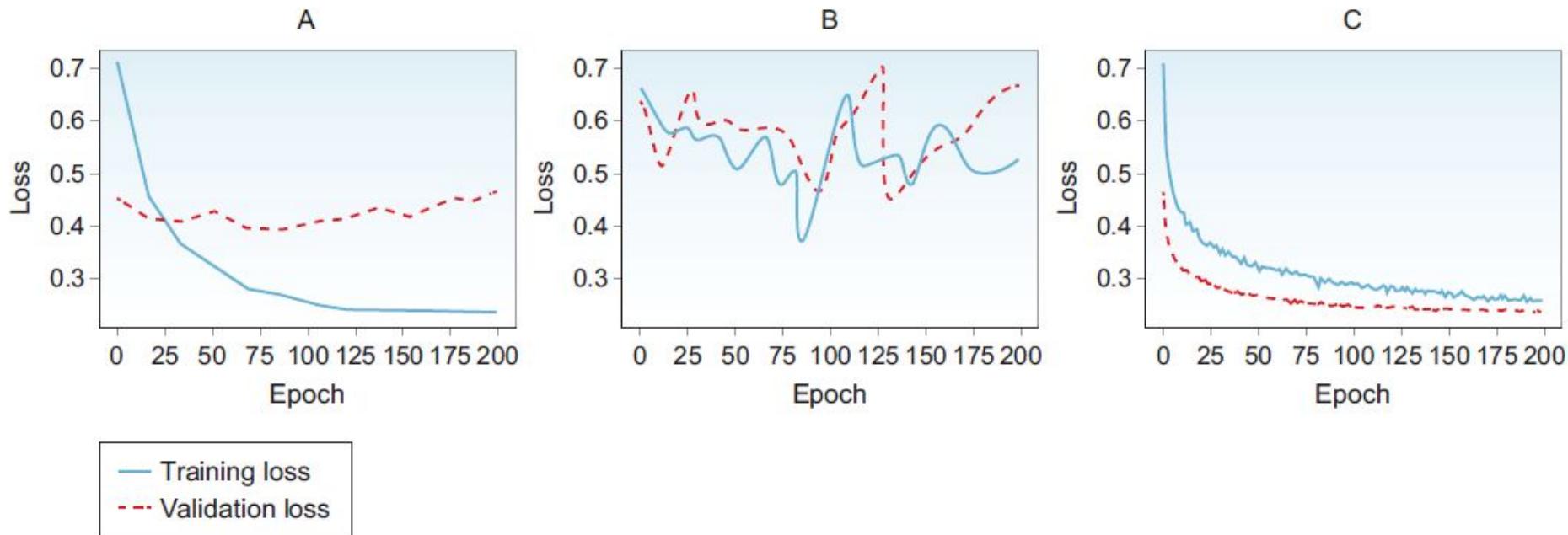


Figure 4.10 (A) The network improves the loss value on the training data but fails to generalize on the validation data. (B) The network performs poorly on both the training and validation data. (C) The network learns the training data and generalizes to the validation data.

Principais Arquiteturas

- LeNet
- AlexNet
- VGG
- GoogLeNet
- ResNet

LeNet

Primeira CNN proposta em 1998 por Yann LeCun
(Cientista chefe de IA do Facebook)

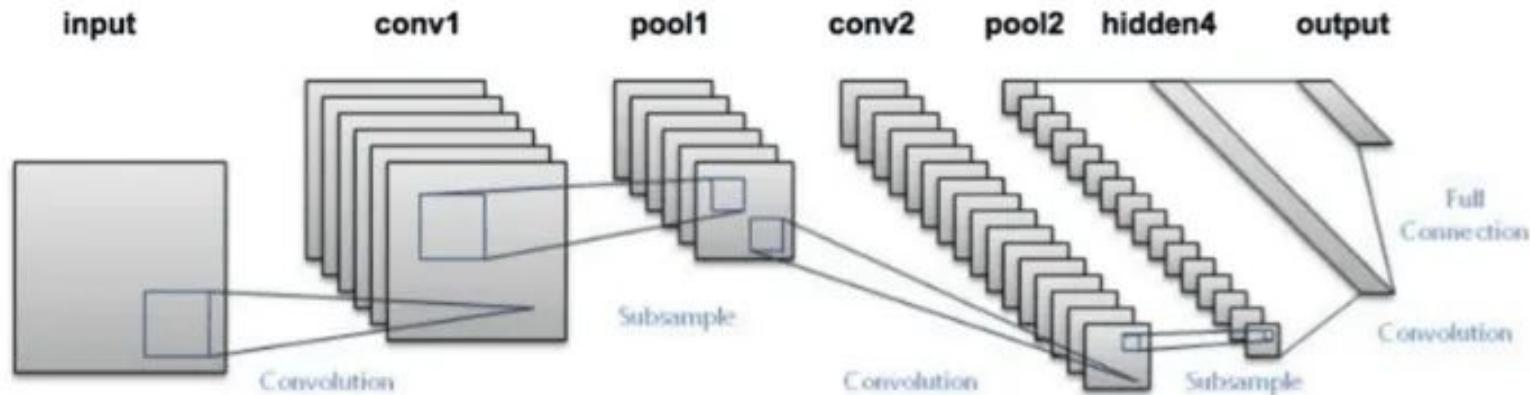
Foi desenvolvida com o intuito de auxiliar no reconhecimento de caracteres da base MNIST e gerou resultados com alta acurácia.

A arquitetura e conceitos introduzidos pela LeNet serviram como base para a criação das arquiteturas mais atuais.



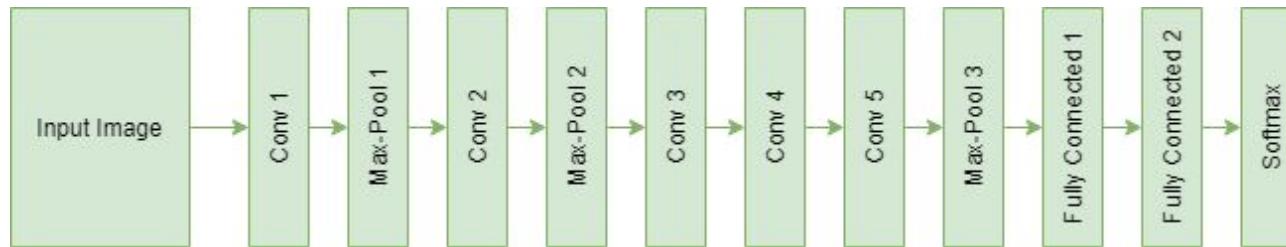
A 10x10 grid of handwritten digits from 0 to 9, each digit enclosed in a small square box. The digits are arranged in a single row.

LeNet



AlexNet

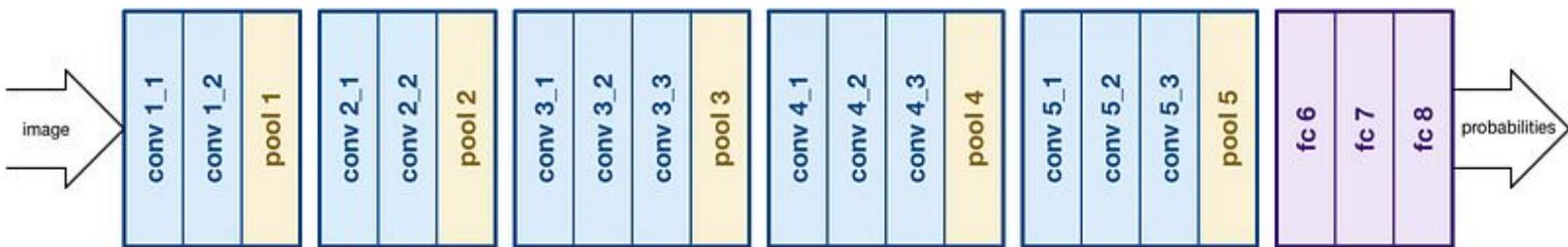
- Vencedora do ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) 2012. Possui arquitetura similar à LeNet, com a inclusão de **mais camadas, filtros, empilhamento das convoluções** e usou pela primeira vez a função de **ativação ReLu**.
- Consiste em 5 camadas convolucionais, com a 1^a, 2^a e 5^a tendo camadas Max-Pooling. São seguidas por 2 camadas totalmente conectadas (cada uma com *dropout*) e uma camada de saída com função softmax no final para previsões. Totalizando 8 camadas com cerca de 61 milhões de parâmetros.



VGG

- Segundo lugar no ILSVRC 2014. Surgiu com a proposta de criar uma arquitetura de rede mais profunda que as anteriores. A rede contém 19 camadas e 138 milhões de parâmetros.
- Foi possível aumentar a quantidade de camadas com o empilhamento de filtros 3×3 , mostrando que o empilhamento de três desses filtros (243 parâmetros) tinha o mesmo campo receptivo do que um filtro de 7×7 (3969 parâmetros).
- Criou-se uma rede mais profunda sem aumentar muito a quantidade de parâmetros. Existem outras versões da rede com variações principalmente na quantidade de camadas, como a **VGG-11, 13, 16 e 19**.

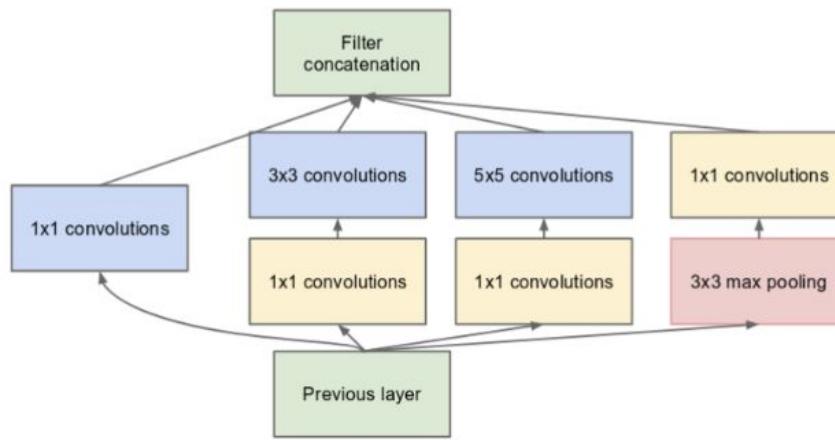
VGG



GoogLeNet

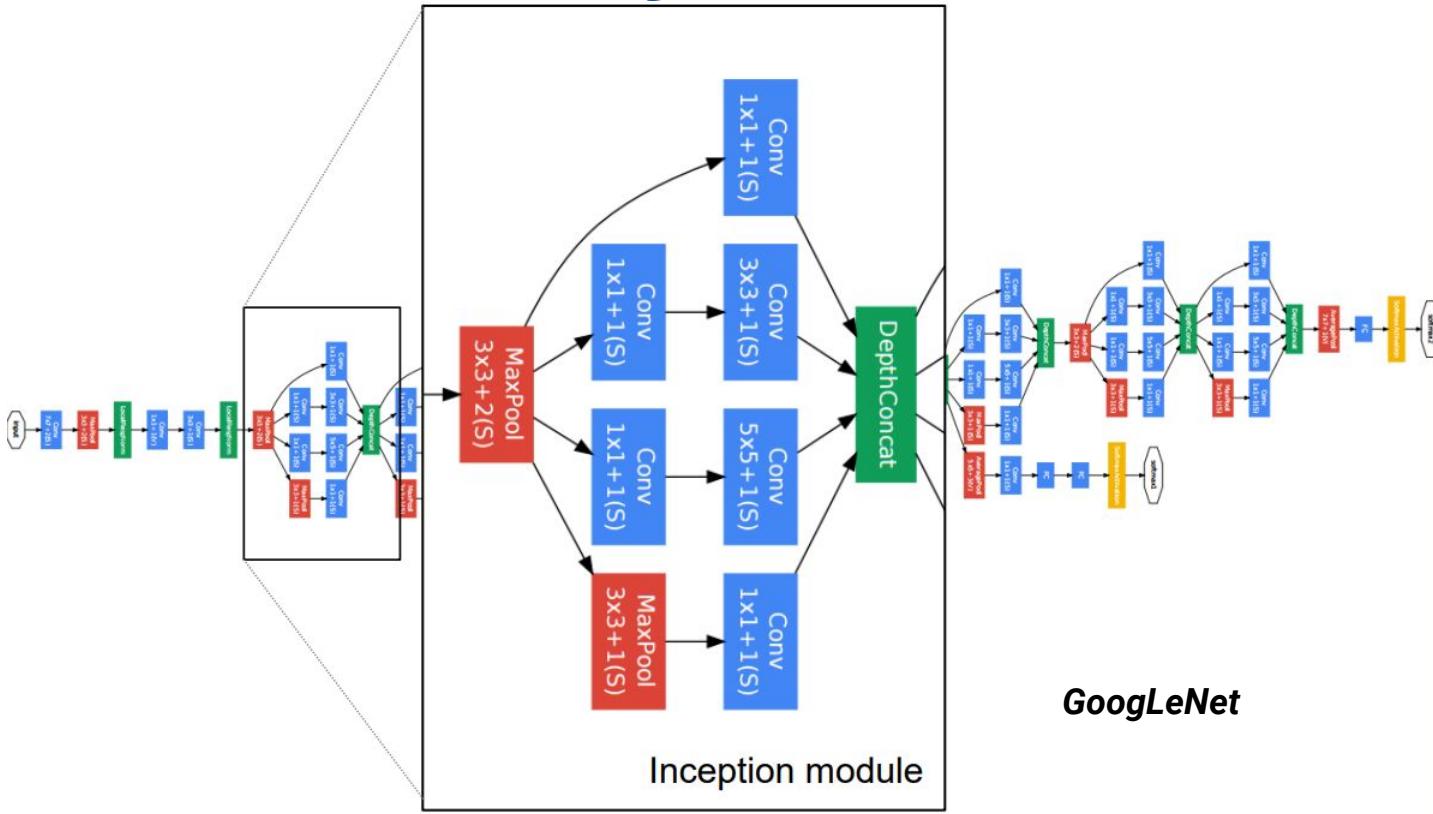
- Vencedora do ILSVRC 2014. Possui **22 camadas** e não utiliza camadas totalmente conectadas (**4 milhões de parâmetros**).
- Também com a proposta de inserção de mais camadas, criou-se o conceito de **Inception Module**. Além disso, ao invés de escolher quais tamanhos de filtros são os mais adequados, a arquitetura foi projetada para executar todos em paralelo e empilhar as respostas. Dessa maneira, a rede aprende quais são as melhores configurações de filtros, dando maior dinamicidade.
- Diferentes configurações de filtros aumenta a quantidade de parâmetros, inviabilizando que seja mais profunda. Para contornar esse problema, foram inseridas as camadas “bottleneck”, que adicionam filtros de tamanho 1×1 antes da convolução desejada, viabilizando a execução dos filtros em paralelo, reduzindo a quantidade de parâmetros.

GoogLeNet

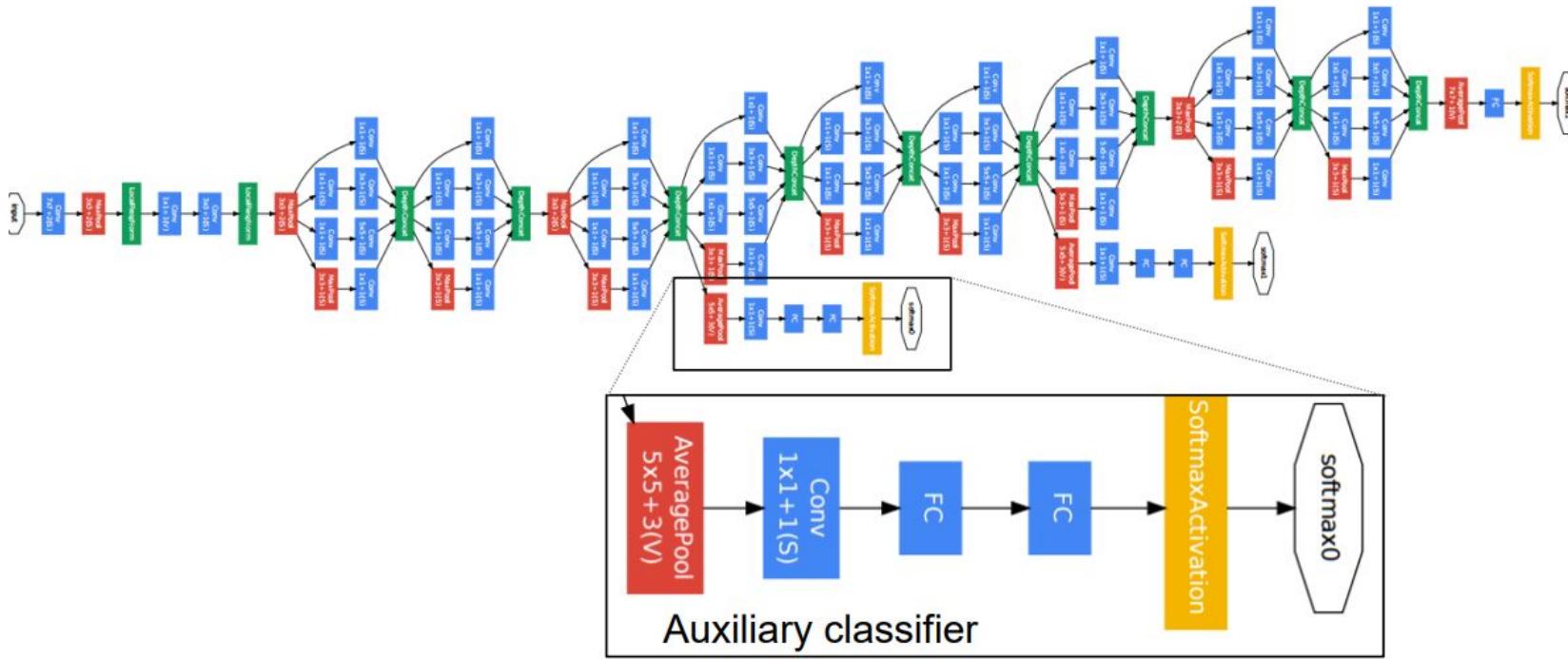


GoogLeNet

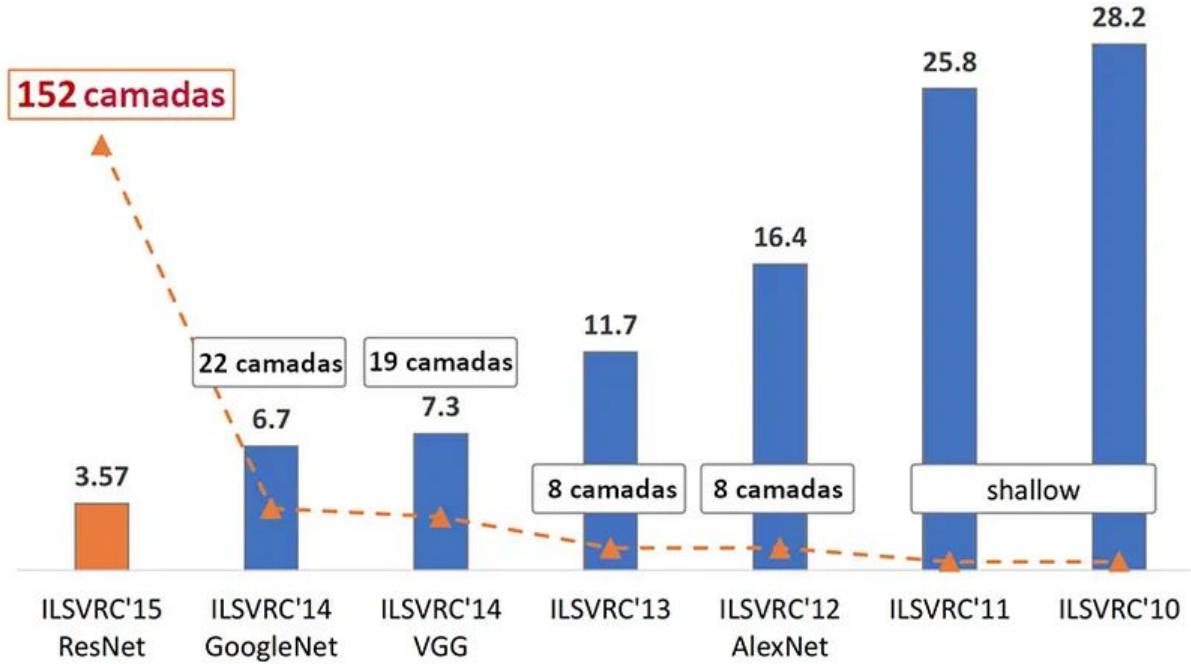
GoogLeNet



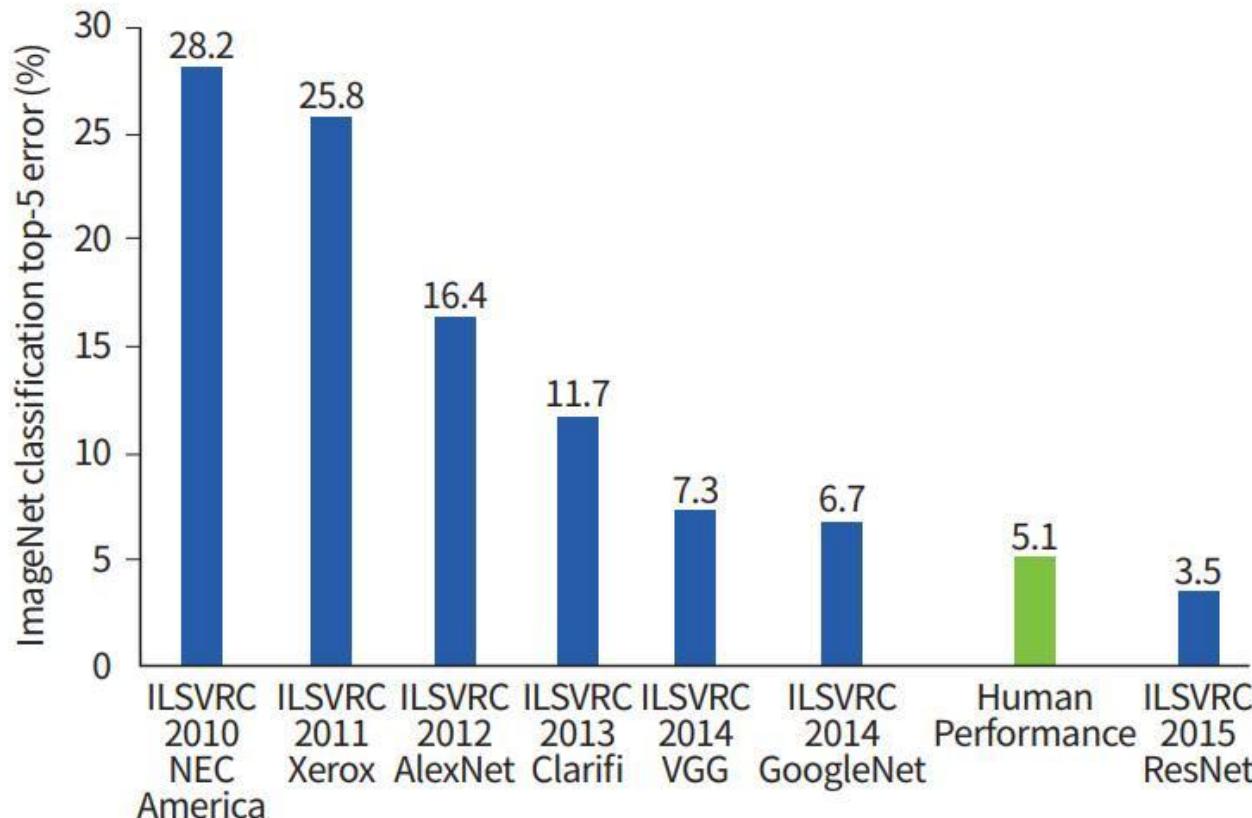
GoogLeNet



Large Scale Visual Recognition Challenge



Large Scale Visual Recognition Challenge

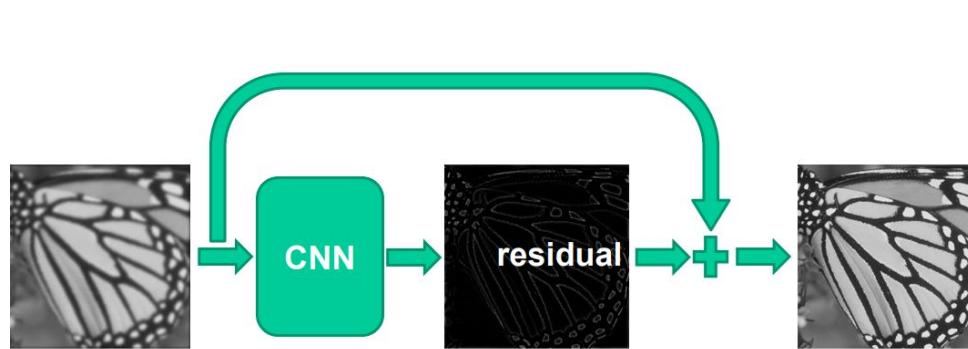


ResNet

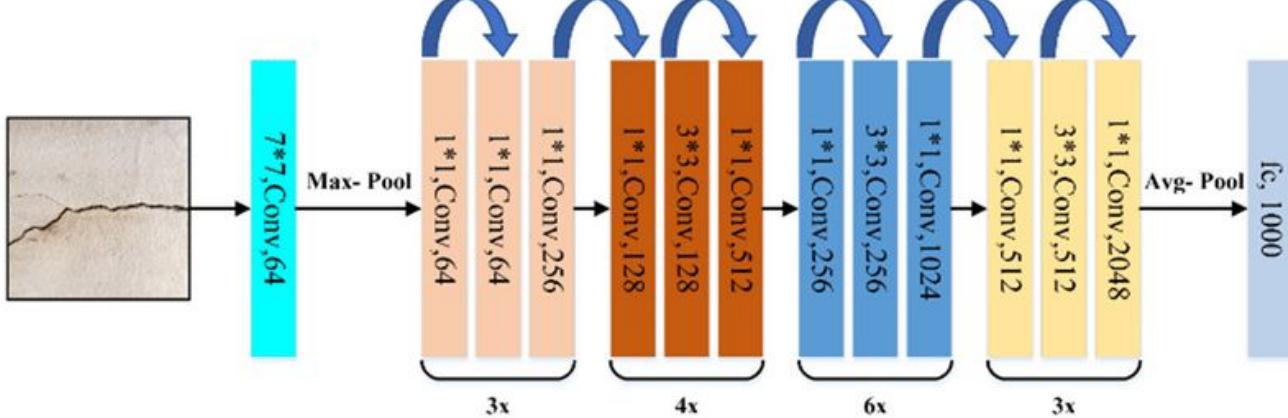
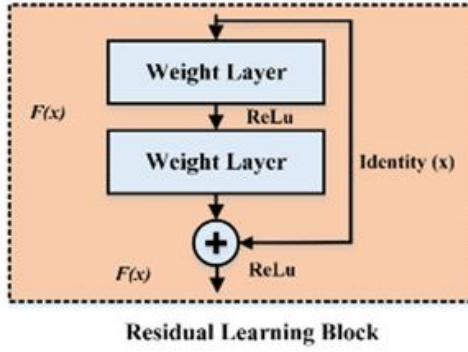
- Vencedora do ILSVRC 2015. Foi a **primeira rede a obter erro de classificação menor que o humano para a aplicação**. Sua principal inovação foi conseguir criar redes ultra profundas (com 152 camadas) sem sacrificar seu desempenho.
- Os autores notaram que redes mais profundas enfrentavam o **“problema da degradação”**, ou seja, a **acurácia da rede acaba saturando ou degradando** com o aumento **na quantidade de camadas**, indicando que redes mais rasas estão aprendendo melhor do que mais profundas.
- Esse problema pode ser causado pelo desaparecimento do gradiente (*vanishing gradient*), em que o gradiente pode ser tornar infinitamente pequeno na retropropagação para as camadas anteriores e/ou maldição da dimensionalidade. Foi proposto pular essas camadas extras para obter pelo menos a mesma acurácia que a rede mais rasa.

ResNet

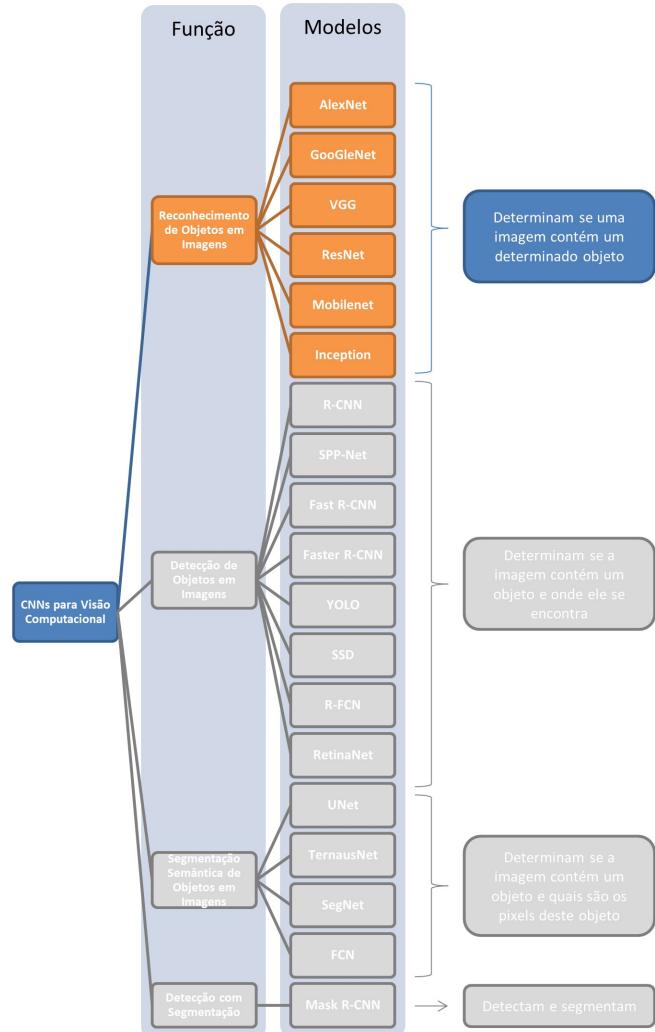
- Foi criada a “conexão identidade”, que insere uma conexão entre o início e final de um conjunto de camadas, sendo estas camadas conhecidas como o mapeamento residual. Dessa maneira, algumas camadas podem ser puladas durante o treinamento. Além disso, devido aos saltos nas conexões, a rede pode escolher quais camadas são mais importantes, tornando-a mais dinâmica e o gradiente pode ser propagado para as camadas iniciais, permitindo a criação de redes ultra profundas.



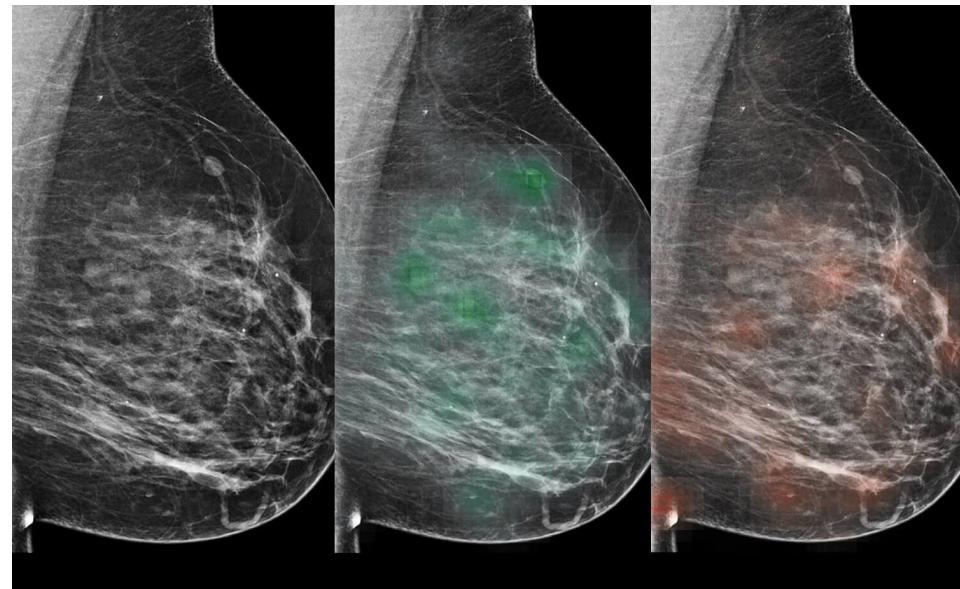
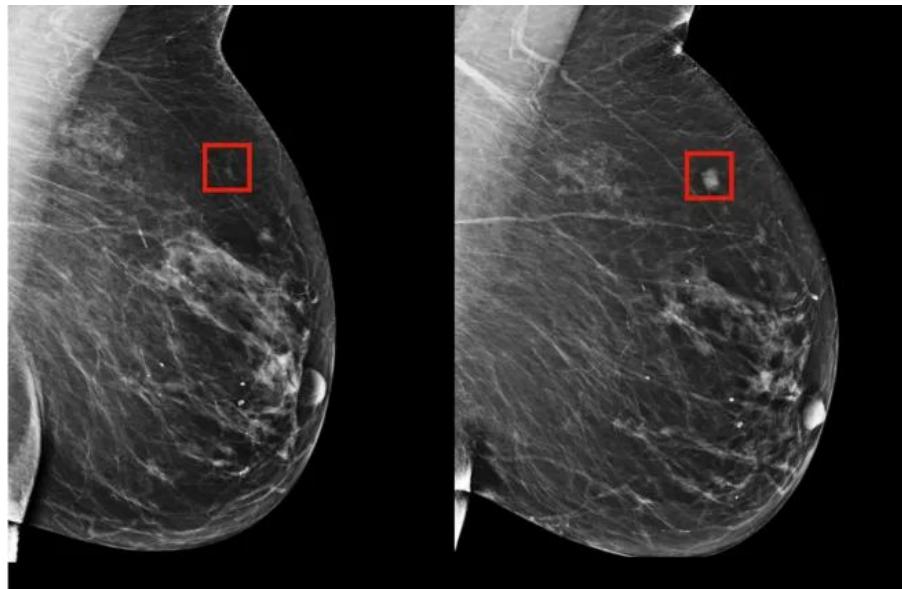
ResNet



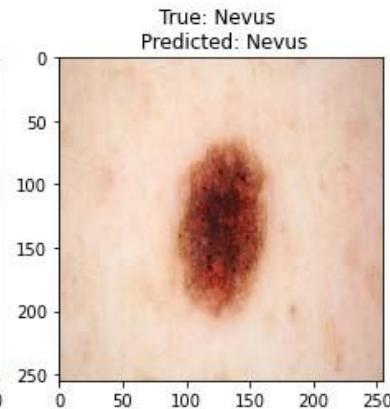
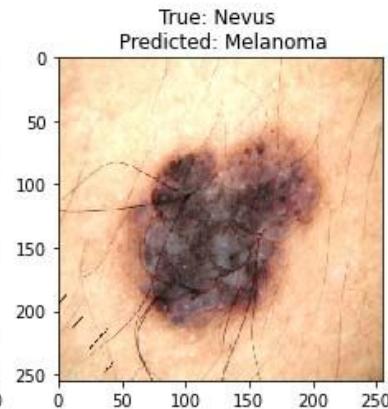
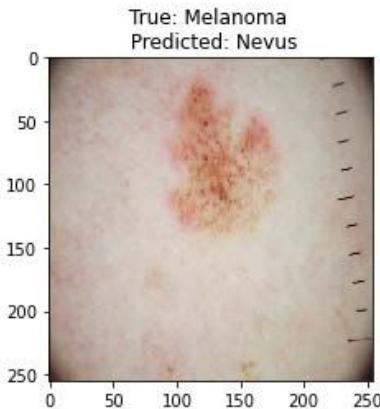
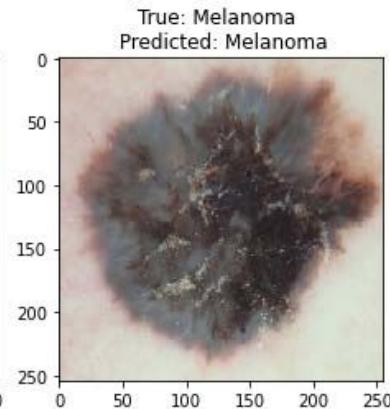
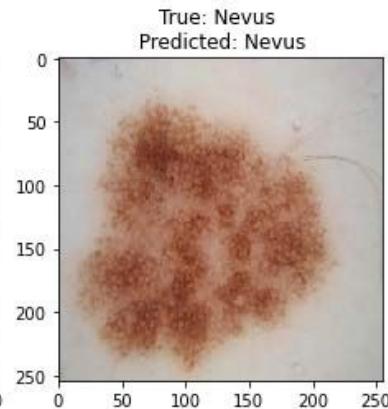
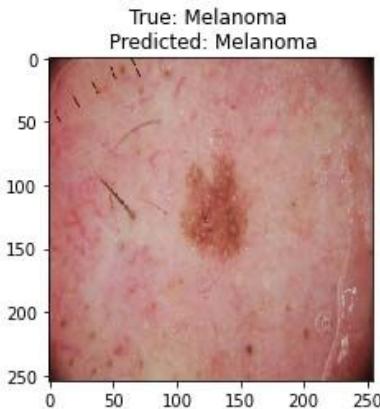
ResNet



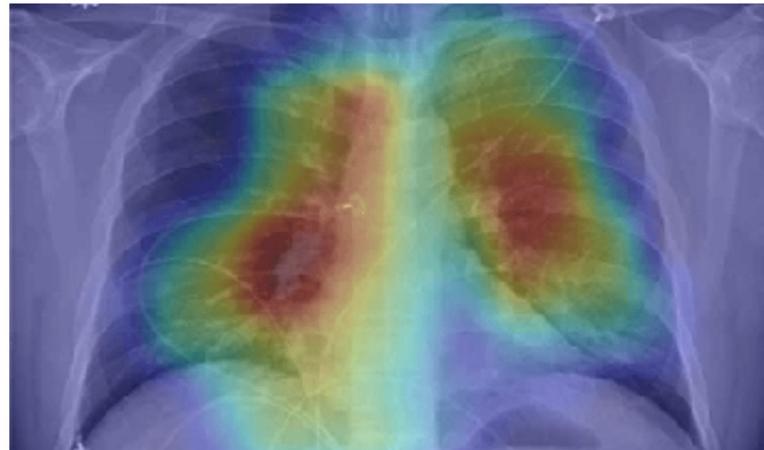
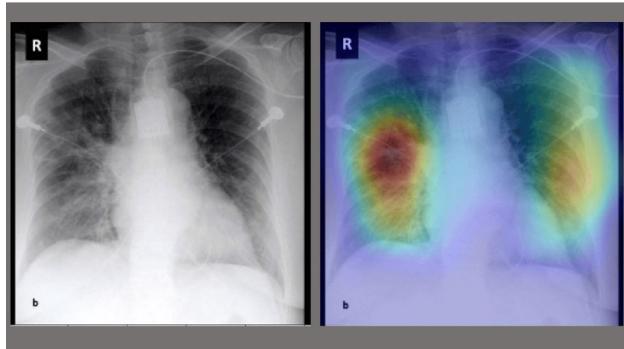
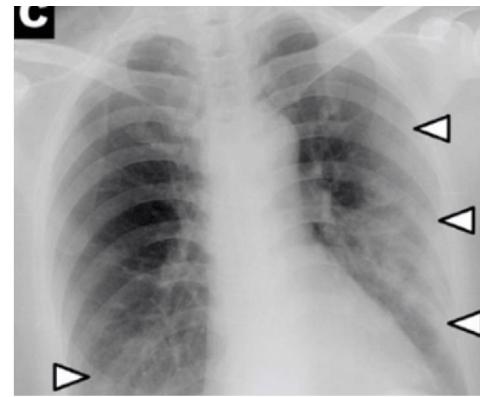
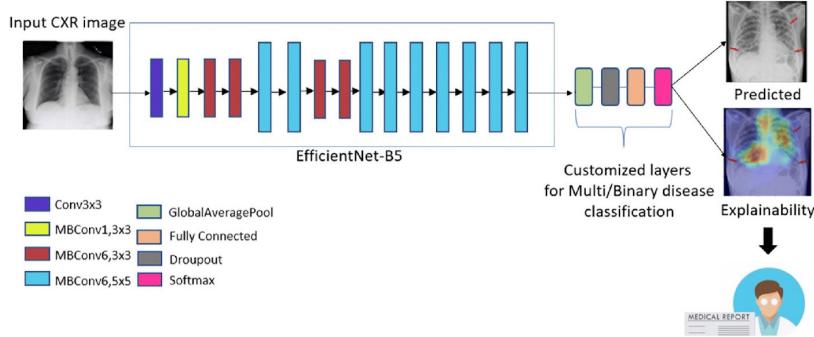
Aplicações na Medicina



Aplicações na Medicina



Aplicações na Medicina



Exemplo prático de uma CNN

[CNN Básica - CIFAR-10.ipynb](#)

Ambiente de Desenvolvimento



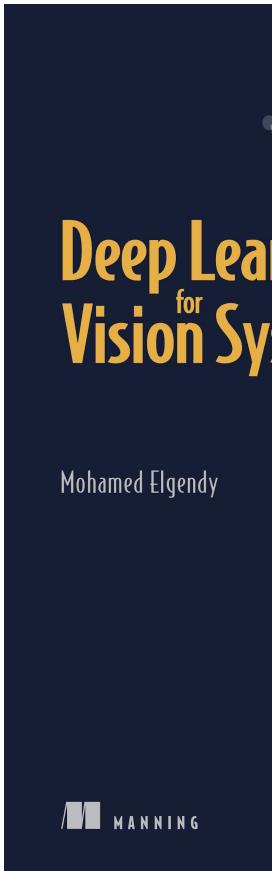
kaggle



matplotlib

K Keras

Bibliografia





Universidade Federal do Pará
Campus Universitário de Belém
Instituto de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica

Obrigado!