Introdução às Redes Neurais Convolucionais (CNNs)

Eduardo Adame

Redes Neurais

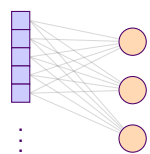
24 de setembro de 2025

De MLPs para CNNs: Por que Mudar?

Limitações das MLPs para Imagens:

- Explosão de parâmetros
  - Imagem 200x200 RGB = 120.000 entradas
  - Uma camada: 14.4 bilhões de pesos!
- Perda de estrutura espacial
  - MLP trata pixels como lista
  - Ignora proximidade dos pixels
- Sem invariância à translação
  - Objeto em posições diferentes
  - Requer reaprender padrões

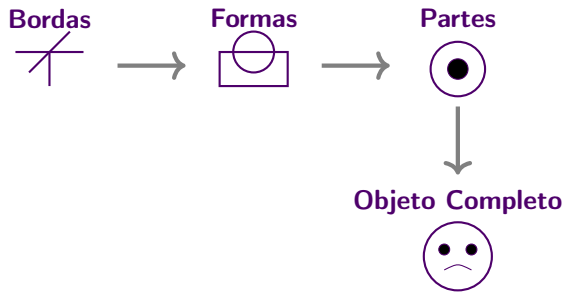
MLP: Visão "Achatada"



A Natureza Hierárquica da Visão

Como construímos representações visuais:

- Nível 1: Detectores de bordas
- Nível 2: Formas simples
- Nível 3: Partes de objetos
- Nível 4: Objetos completos



Exemplo - Reconhecendo um Gato:

- Bordas $\rightarrow$ Círculos
- Círculos $\rightarrow$ Olhos
- Olhos + Textura $\rightarrow$ Face de gato

Conceito de Convolução: Filtros Locais

O que é uma Convolução?

- Operação matemática entre imagem e filtro
- Filtro (kernel) "desliza"sobre a imagem
- Detecta padrões locais específicos

Vantagens sobre MLP:

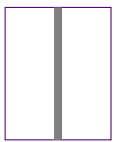
- Compartilhamento de pesos: mesmo filtro em toda imagem
- Conectividade local: neurônio vê apenas região
- Invariância à translação: detecta padrão em qualquer posição

Exemplo: Detector de Bordas Verticais

Filtro

-1	1	-1
-1	1	-1
-1	1	-1

Resposta



Borda detectada!

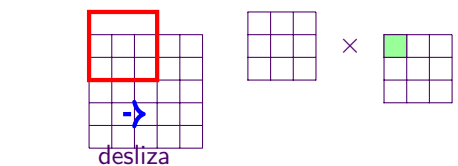
Camadas Convolucionais: O Coração das CNNs

Operação de Convolução:
Para cada posição (i, j):

$$y_{i,j} = \sum_{m,n} x_{i+m,j+n} \cdot w_{m,n} + b$$

Hiperparâmetros Importantes:

- Tamanho do filtro: 3×3, 5×5, 7×7
- Número de filtros: 32, 64, 128...
- Stride: passo do filtro (1, 2...)
- Padding: zeros nas bordas



Múltiplos Filtros = Múltiplos Mapas de Características

Pooling: Reduzindo Dimensionalidade

Por que usar Pooling?

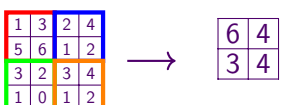
- Reduz tamanho espacial
- Diminui número de parâmetros
- Adiciona invariância a pequenas translações
- Controla overfitting

Tipos Comuns:

- Max Pooling: valor máximo da região
- Average Pooling: média dos valores
- Geralmente 2×2 com stride 2

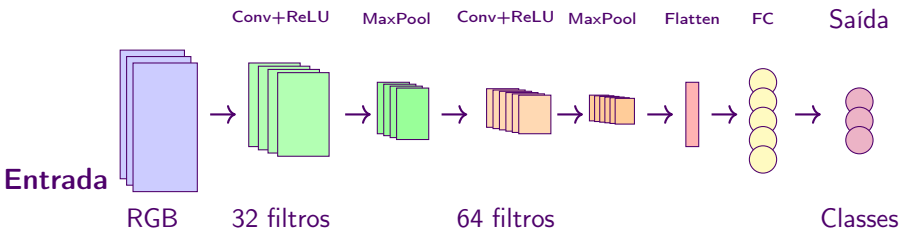
Exemplo: Max Pooling 2×2

Entrada 4×4 **Saída 2×2**



Redução de 75% no tamanho!

Arquitetura Típica de uma CNN



Padrão: CONV → ReLU → POOL → CONV → ReLU → POOL → FC → Softmax

Comparação: CNN vs MLP para Imagens

Aspecto	MLP	CNN
Preserva estrutura espacial	X	✓
Número de parâmetros	Muito alto	Reduzido
Compartilhamento de pesos	X	✓
Invariância à translação	X	✓
Detecta padrões locais	Difícil	Natural
Hierarquia de características	X	✓
Eficiência computacional	Baixa	Alta
Generalização em imagens	Limitada	Excelente

MLP: Força Bruta

- Conecta tudo com tudo
- Ignora localidade
- Precisa de muitos dados

CNN: Inteligência Estrutural

- Explora estrutura da imagem
- Aprende filtros úteis
- Generaliza melhor

Exemplo Prático: Reconhecimento de Dígitos

Dataset MNIST:

- 60.000 imagens de treino
- 10.000 imagens de teste
- Imagens 28×28 pixels
- 10 classes (dígitos 0-9)

Arquitetura Simples:

- Conv1: 32 filtros 3×3
- MaxPool: 2×2
- Conv2: 64 filtros 3×3
- MaxPool: 2×2
- FC: 128 neurônios
- Saída: 10 classes

Visualização dos Filtros Aprendidos:

Camada 1: Detectores de Bordas

Camada 2: Detectores de Formas

Resultado: 99% de Acurácia!

Implementação em Keras/TensorFlow

Listing 1: CNN Simples para MNIST

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.layers import Flatten, Dense, Dropout

# Construindo o modelo CNN
model = Sequential([
    # Primeira camada convolucional
    Conv2D(32, (3,3), activation='relu',
           input_shape=(28, 28, 1)),
    MaxPooling2D((2,2)),

    # Segunda camada convolucional
    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D((2,2)),
```

Implementação em Keras/TensorFlow

Listing 2: CNN Simples para MNIST

```
    # Camadas densas
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Aplicações Modernas das CNNs

Visão Computacional:

- Classificação: ImageNet, CIFAR
- Detecção: YOLO, R-CNN
- Segmentação: U-Net, Mask R-CNN
- Reconhecimento facial: FaceNet

Medicina:

- Diagnóstico por imagem
- Detecção de tumores
- Análise de retina

Outras Aplicações:

- Carros autônomos: detecção de objetos
- Arte: transferência de estilo
- Agricultura: análise de culturas
- Segurança: vigilância inteligente

Arquiteturas Famosas:

- LeNet-5 (1998)
- AlexNet (2012)
- VGG (2014)
- ResNet (2015)
- EfficientNet (2019)

Resumo: CNNs vs MLPs

MLP

- Conexão total
- Muitos parâmetros
- Sem estrutura espacial
- Bom para dados tabulares

Escolha

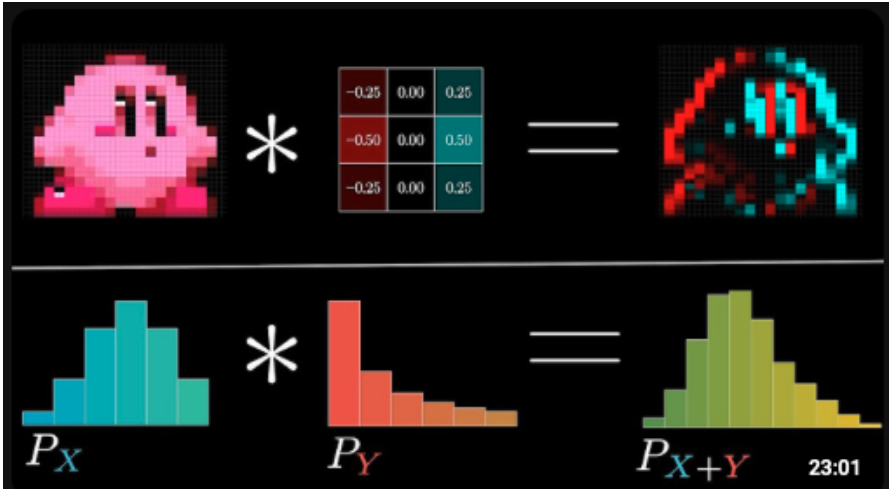
CNN

- Conexão local
- Compartilha pesos
- Preserva topologia
- Ideal para imagens

Lição Principal:

CNNs são MLPs especializadas que exploram a estrutura espacial dos dados visuais através de convoluções e pooling para aprender hierarquias de características

Explorando as CNNs



Recomendação: assistir o vídeo do 3Blue1Brown (legendas em português).
Link: https://www.youtube.com/watch?v=KuXjwB4LzSA

Obrigado!

Dúvidas?

Próximo: Arquiteturas de CNNs