Inteligência Artificial

Raoni F. S. Teixeira

Aula 12 - Redes Convolucionais

1 Introdução

As **Redes Neurais Convolucionais (CNNs)** seguem a mesma estrutura básica das redes neurais tradicionais. Elas usam neurônios com pesos ajustáveis, combinam operações lineares com ativações não lineares e seu treinamento depende de uma função de custo diferenciável.

A diferença é que as CNNs são construídas para processar sinais – dados estruturados *espacialmente* (e.g. imagem ou matriz 2D) ou *no tempo* (e.g. aúdio ou vetor). Como trabalham com matrizes ou vetores, as CNNs processam os sinais, explorando três princípios para maior eficiência:

- **Conectividade local** – Cada neurônio responde apenas a uma pequena região da entrada, reduzindo a complexidade.
- **Compartilhamento de pesos** – Os mesmos filtros são aplicados em diferentes posições, detectando padrões de forma consistente.
- **Invariância translacional** – Padrões são reconhecidos independentemente de sua posição no sinal.

2 Filtros

Um **filtro** é uma função que modifica um sinal, preservando componentes específicas (como certas frequências) e atenuando outras. O resultado é um sinal simplificado e mais limpo.

Convolução 1D

A Figura 1 ilustra a ação do filtro $F = (-1, +1)$ em um sinal binário unidimensional. O filtro (ou **núcleo convolucional**) desliza pelo sinal com um **passo (stride)** de 1, calculando o produto escalar em cada posição. O *stride* define o deslocamento do filtro a cada operação.

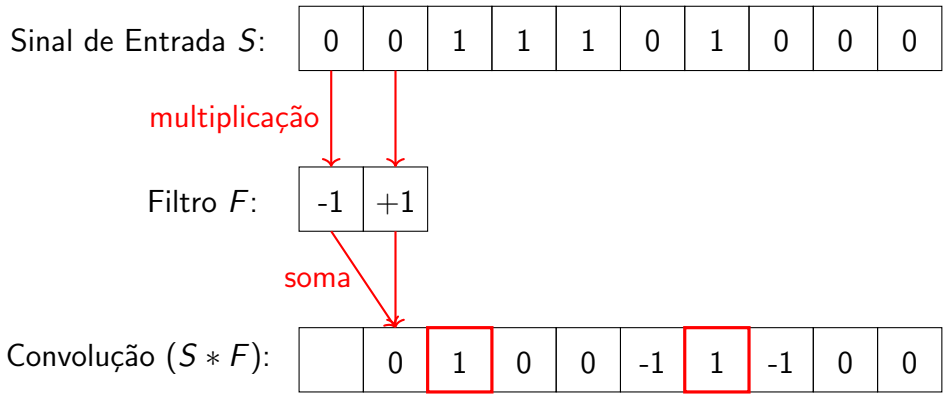

Figura 1: Convolução 1D para detectar as bordas esquerdas (células em destaque) de um sinal binário.

Para um sinal discreto S e um filtro F de tamanho M e stride s, a convolução unidimensional é definida por:

$$(S * F)[n] = \sum_{m=1}^{M} S[n \cdot s - m]F[m] \tag{1}$$

Convolução 2D

A Figura 2 mostra um filtro 3×3 aplicado a um sinal binário 2D S em que as células pretas correspondem ao valor 1 e as brancas ao valor 0. O filtro compara cada região de S, procurando padrões horizontais de tamanho três. O *stride* igual à 1 faz com que o filtro deslize por toda imagem.

A fórmula para convolução 2D com filtro $K \times K$ e *stride* s é:

$$(S * F)[i, j] = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} S[i \cdot s + m, j \cdot s + n] \cdot F[m, n] \tag{2}$$

A operação de convolução não é bem definida nas bordas do sinal. Um filtro 3×3, por exemplo, processa as células centrais completamente (9 operações), enquanto as células das bordas participam de no máximo 4 operações. Essa diferença cria um desequilíbrio no processamento, onde informações das bordas recebem menos atenção.

A solução é adicionar uma borda de zeros ao redor da imagem original - técnica conhecida como **padding**. Esse ajuste garante três benefícios. Primeiro, equaliza o processamento de todos os pixels, incluindo os das bordas. Segundo, permite controlar o tamanho da saída, mantendo as dimensões originais quando necessário. Terceiro, preserva informações espaciais cruciais, especialmente importante nas primeiras camadas da rede.

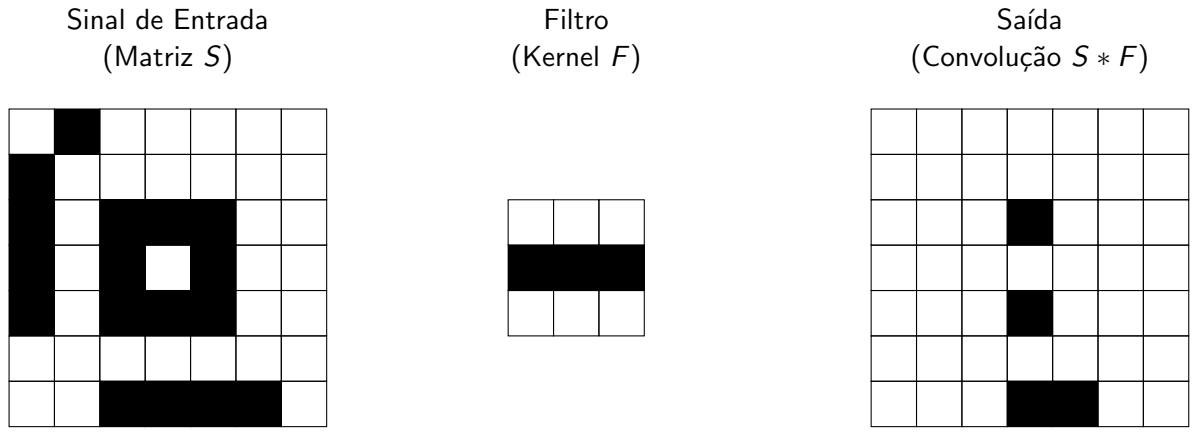

Figura 2: Convolução 2D com filtro 3×3 e *stride* 1 que detecta padrões horizontais no sinal S.

3 Agrupamento (Pooling)

O agrupamento (pooling) é um filtro sem pesos que reduz a dimensionalidade preservando características do sinal. Novamente, o stride indica o passo de aplicação do filtro. Quando stride é igual a dois, por exemplo, o filtro pula uma célula em cada dimensão.

Há dois tipos comuns de agrupamento:

1. **Agrupamento Máximo (Max Pooling)**: que devolve o valor máximo em cada janela do filtro e cujo principal benefício é aumentar a invariância a pequenas translações (o padrão é reconhecido independentemente da posição exata no sinal).
2. **Agrupamento Médio (Average Pooling)**: que calcula a média dos valores na janela e é usado para suavizar o sinal.

A Figura 3 ilustra ambos os métodos aplicados a uma matriz de entrada com filtros 2×2 e stride 2.

4 Arquitetura de uma CNN

A Figura 4 mostra a arquitetura típica de uma Rede Neural Convolucional (CNN) para classificação de imagens. A rede é organizada em camadas que realizam operações de:

- **Convolução**: Extração progressiva de características (das mais simples às mais complexas)

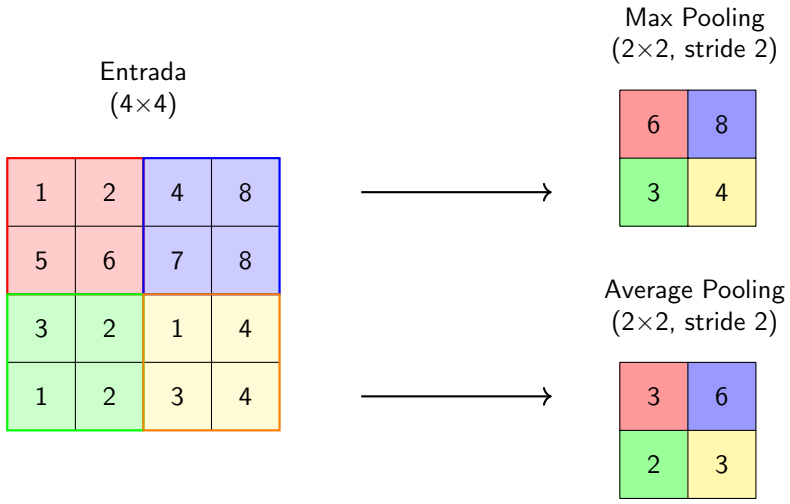

Figura 3: Operações de pooling usando filtro 2×2. As cores indicam janelas distintas. O Max Pooling extrai os valores máximos (6, 8, 3, 4) e Average Pooling as médias (3, 6, 2, 3) em cada janela.

- **Agrupamento (Pooling)**: Redução dimensional espacial
- **Classificação**: Camadas totalmente conectadas no final

As camadas convolucionais utilizam bancos de filtros para detectar padrões hierárquicos em sinais (Figura 5). Na primeira camada convolucional, a rede aplica um conjunto de k filtros distintos à imagem original de entrada. Cada um desses filtros produz um mapa de características único, resultando em k canais de saída. Esses canais são então organizados em um volume tridimensional de dados, em que a largura e altura espacial correspondem as dimensões da imagem e a profundidade corresponde aos k canais gerados.

À medida que avançamos para as camadas posteriores da rede, os filtros assumem uma natureza tridimensional, operando simultaneamente sobre todos os canais existentes. Essa arquitetura permite que a rede combine progressivamente características mais simples detectadas nas camadas iniciais, formando representações cada vez mais complexas e abstratas dos padrões presentes na imagem original. A capacidade de processar todos os canais simultaneamente é fundamental para a construção dessa hierarquia de características.

Em resumo, a estrutura típica de um CNN é dada por:

1. Blocos repetidos de:
    - Camada convolucional + ReLU (retificação)
    - Camada de *max pooling* (opcional)

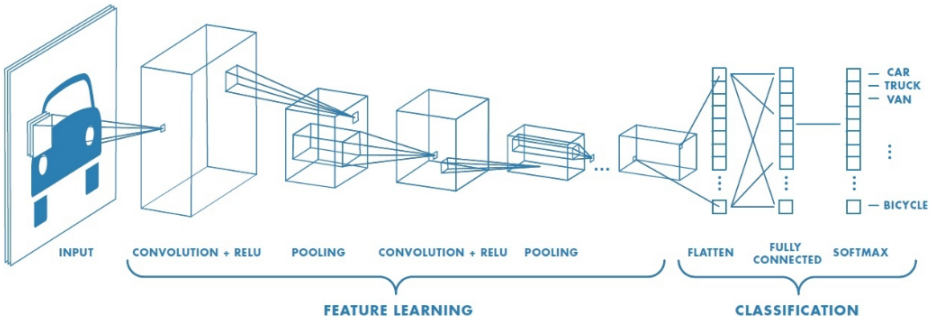

Figura 4: Arquitetura típica de CNN para classificação de imagens. As primeiras camadas aprendem características de baixo nível (bordas, texturas), enquanto as camadas posteriores combinam essas informações para reconhecimento de padrões complexos (i.e. transforma o sinal para um espaço em que as classes são separáveis).

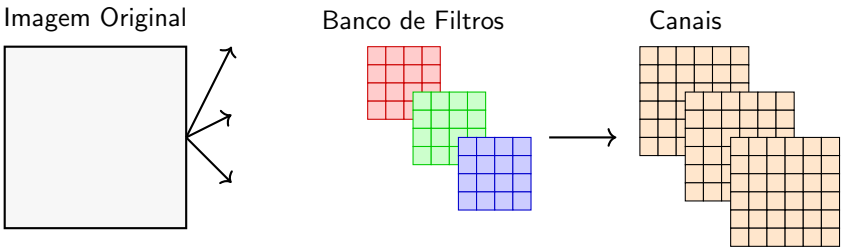

Figura 5: Processo de aplicação de múltiplos filtros: (1) Cada filtro gera um mapa de ativação distinto; (2) Os mapas são empilhados formando um volume 3D de características.

2. Camadas totalmente conectadas no final

3. Função de ativação final (ex: Softmax para classificação)

Para avaliar a complexidade de um modelo, é comum determinar o número de parâmetros de sua arquitetura. Em cada camada de uma rede neural convolucional, isso é feito da seguinte forma:

Camada	**CONV**	**POOL**	**FC**
Tamanho de entrada	$I \times I \times C$	$I \times I \times C$	$I \times I \times C$
Tamanho de saída	$O \times O \times K$	$O \times O \times C$	N_{out}
Número de parâmetros	$(F \times F \times C + 1) \cdot K$	0	$(N_{\text{in}} + 1) \times N_{\text{out}}$
Observações	• **1 parâmetro de bias por filtro** • Geralmente, o *stride* $S < F$ • $K = 2C$ é uma escolha comum	• **Pooling é aplicado por canal** • Geralmente, $S = F$ (sem sobreposição)	• **Entrada é linearizada** • 1 parâmetro de *bias* por neurônio • Número de neurônios livre de restrições

5 Transferência de Aprendizado

O treinamento de redes neurais convolucionais a partir do zero é incomum na prática, principalmente devido à exigência de grandes conjuntos de dados. A solução amplamente adotada é a transferência de aprendizado, implementada através de três estratégias essenciais:

- **Ajuste Fino (Fine-Tuning)**: Modifica pesos específicos de uma rede pré-treinada enquanto mantém sua arquitetura original. A técnica padrão congela as camadas iniciais - responsáveis por detectar padrões visuais básicos - e ajusta apenas as camadas finais, que contêm características específicas do domínio. Esse método exige atenção redobrada para prevenir overfitting, particularmente com conjuntos de dados limitados.

- **CNN como Extrator de Características**: Transforma redes pré-treinadas em extratores de features poderosos ao remover sua camada de classificação final. As ativações das camadas intermediárias, como os vetores 4096-D da penúltima camada da AlexNet, servem como entrada para novos classificadores (SVMs ou modelos lineares). Essa abordagem preserva o conhecimento hierárquico aprendido enquanto se adapta a novas tarefas.

- **Uso de Modelos Pré-Treinados**: Aproveita arquiteturas consolidadas e pesos otimizados disponíveis em repositórios como Caffe Model Zoo e TensorFlow Hub. Além de poupar semanas de treinamento, essa prática oferece acesso imediato a modelos de alta performance como VGG, ResNet e EfficientNet - todos pré-treinados no ImageNet - com ajustes mínimos necessários para novas aplicações.

6 Exercícios

1. Projete uma arquitetura CNN para o dataset MNIST com 2 camadas convolucionais e uma densa.

2. Compare o número de parâmetros de sua rede com a rede neural tradicional das aulas 9 e 10.