

# Introdução às Redes Neurais Convolucionais (CNNs)

Raul Myron - IEEE CIS

2 de junho de 2025

# O que é uma Rede Neural Convolucional (CNN)?

- ▶ Uma Rede Neural Convolucional (CNN), ou ConvNet, é um tipo especializado de algoritmo de aprendizado profundo.
- ▶ Projetada principalmente para tarefas que necessitam de reconhecimento de objetos, incluindo:
  - ▶ Classificação de imagens
  - ▶ Detecção de objetos
  - ▶ Segmentação de imagens
- ▶ Empregadas em cenários práticos como veículos autônomos, sistemas de câmeras de segurança, entre outros.

# A Importância das CNNs

- ▶ **Extração Autônoma de Características:** Diferentemente de algoritmos clássicos como SVMs e árvores de decisão, CNNs extraem características em larga escala automaticamente, eliminando a necessidade de engenharia manual de características e aumentando a eficiência.
- ▶ **Invariância à Translação:** As camadas convolucionais conferem às CNNs características de invariância à translação, permitindo-lhes identificar padrões independentemente de variações na posição, orientação, escala ou translação.
- ▶ **Modelos Pré-treinados:** Arquiteturas como VGG-16, ResNet50, InceptionV3 e EfficientNet demonstraram desempenho de ponta e podem ser adaptadas para novas tarefas com relativamente poucos dados (fine-tuning).
- ▶ **Versatilidade:** Além da classificação de imagens, são aplicáveis em processamento de linguagem natural, análise de séries temporais e reconhecimento de fala.

# Inspiração: O Sistema Visual Humano

As CNNs foram inspiradas pela arquitetura em camadas do córtex visual humano.

## Similaridades Chave:

- ▶ **Arquitetura Hierárquica:**  
Características simples são extraídas em camadas iniciais e características mais complexas são construídas em camadas mais profundas.
- ▶ **Conectividade Local:**  
Neurônios em uma camada CNN conectam-se apenas a uma região local da entrada (campo receptivo), similar aos neurônios no córtex visual.
- ▶ **Invariância à Translação:**

## Diferenças:

- ▶ CNNs são mais simples.
- ▶ Carecem dos complexos mecanismos de feedback do cérebro.
- ▶ Geralmente dependem de aprendizado supervisionado, enquanto o cérebro também utiliza aprendizado não supervisionado.

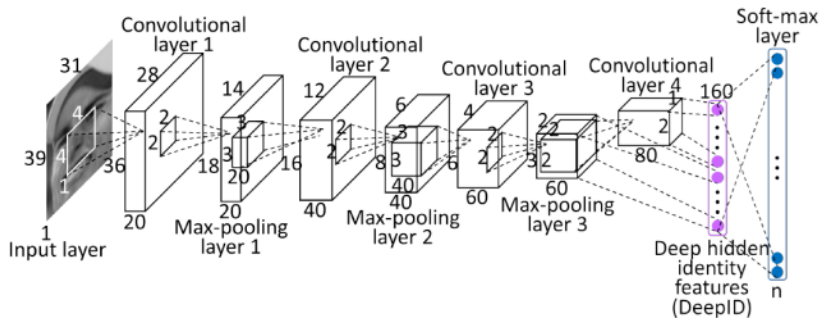


Figura: Rede Neural Convucional Exemplo

# Componentes Chave de uma CNN

Uma CNN típica é composta por quatro partes principais que ajudam a mimetizar como o cérebro humano opera para reconhecer padrões:

1. Camadas Convolucionais (Convolutional Layers)
2. Função de Ativação (ex: Unidade Linear Retificada - ReLU)
3. Camadas de Pooling (Pooling Layers)
4. Camadas Totalmente Conectadas (Fully Connected Layers)

# 1. Camadas Convolucionais

- ▶ Bloco fundamental da CNN.
- ▶ Realiza a operação de **convolução**: aplicação de uma função de janela deslizante (kernel ou filtro) a uma matriz de pixels da imagem.
- ▶ Múltiplos filtros de mesmo tamanho são aplicados, cada um para reconhecer um padrão específico (curvas, bordas, formas).
- ▶ **Kernel (ou Filtro)**: Uma pequena matriz de pesos. Os pesos são aprendidos durante o treinamento.
- ▶ **Processo**:
  1. Aplicar o kernel do canto superior esquerdo para a direita.
  2. Realizar multiplicação elemento a elemento.
  3. Somar os produtos. O resultado é o primeiro valor da matriz convoluída (mapa de características).
  4. Mover o kernel (stride) e repetir até cobrir toda a imagem.
- ▶ Quanto mais camadas convolucionais, mais abstratas as características detectadas.

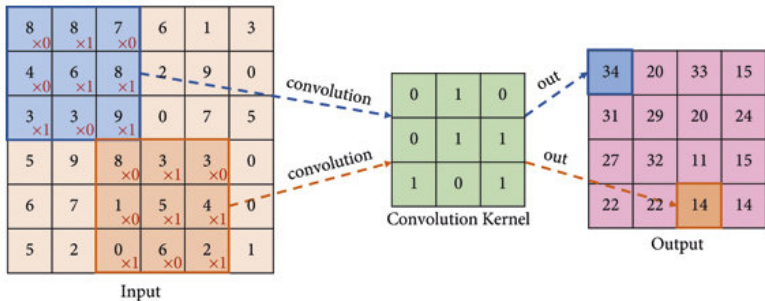


Figura: Exemplo de kernel 3x3



## 2. Função de Ativação (ReLU)

- ▶ Aplicada após cada operação de convolução.
- ▶ **ReLU (Rectified Linear Unit):**  $f(x) = \max(0, x)$ .
- ▶ Ajuda a rede a aprender relações não-lineares entre as características.
- ▶ Torna a rede mais robusta na identificação de diferentes padrões.
- ▶ Ajuda a mitigar o problema do desaparecimento do gradiente (vanishing gradient).

### 3. Camadas de Pooling

- ▶ Objetivo: Extrair as características mais significativas da matriz convoluída (mapa de características).
- ▶ Realiza operações de agregação, reduzindo a dimensão do mapa de características.
- ▶ Consequências:
  - ▶ Reduz a memória usada no treinamento.
  - ▶ Ajuda a mitigar o overfitting.
  - ▶ Fornece invariância a pequenas translações.
- ▶ Funções Comuns:
  - ▶ **Max Pooling:** Seleciona o valor máximo de uma janela do mapa de características.
  - ▶ **Average Pooling:** Calcula a média dos valores da janela.
- ▶ A última camada de pooling geralmente achata (flattens) seu mapa para a camada totalmente conectada.

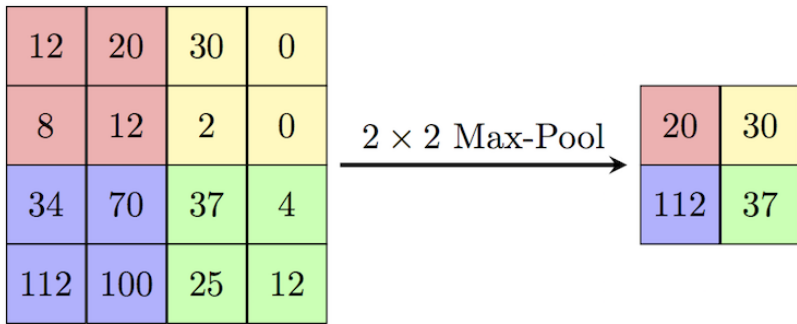


Figura: Max Pooling

## 4. Camadas Totalmente Conectadas (Fully Connected - FC)

- ▶ Localizadas no final da rede CNN.
- ▶ Entradas: Matriz unidimensional achatada (flattened) da última camada de pooling.
- ▶ Neurônios totalmente conectados: cada neurônio na camada FC está conectado a todos os neurônios da camada anterior.
- ▶ Funções de ativação ReLU também podem ser aplicadas aqui para não-linearidade.
- ▶ **Camada de Saída (Softmax):** Para tarefas de classificação, uma camada Softmax é frequentemente usada no final para gerar probabilidades para cada classe. A classe com a maior probabilidade é a predição final.

# Overfitting e Regularização em CNNs

- ▶ **Overfitting:** Ocorre quando o modelo aprende os dados de treinamento "de cor", incluindo ruído e outliers. Isso leva a um bom desempenho nos dados de treinamento, mas ruim em dados novos e não vistos.
- ▶ CNNs são suscetíveis devido à sua alta complexidade.
- ▶ **Técnicas de Regularização (para mitigar overfitting):**
  - ▶ **Dropout:** Descarta aleatoriamente alguns neurônios durante o treinamento.
  - ▶ **Batch Normalization:** Normaliza a entrada da camada, ajustando e escalando as ativações.
  - ▶ **Pooling Layers:** Reduzem dimensões espaciais, abstraindo a representação.
  - ▶ **Early Stopping:** Interrompe o treinamento se o erro de validação não melhora.
  - ▶ **Noise Injection:** Adiciona ruído às entradas ou camadas ocultas.
  - ▶ **Regularização L1 e L2:** Adicionam uma penalidade à função de perda com base no tamanho dos pesos.
  - ▶ **Data Augmentation:** Aumenta artificialmente o tamanho e a diversidade do dataset de treinamento (rotações, zoom, flips, etc).

## 7 Strategies to Mitigate Overfitting in CNNs

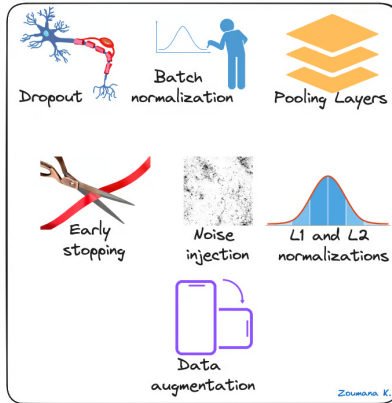


Figura: Estratégias para mitigar overfit

# Aplicações Práticas das CNNs

As CNNs revolucionaram a visão computacional, com avanços em muitas aplicações do mundo real:

- ▶ **Classificação de Imagens:** Categorização de imagens em classes predefinidas (ex: organização automática de fotos).
- ▶ **Detecção de Objetos:** Identifica e localiza múltiplos objetos dentro de uma imagem (ex: escaneamento de prateleiras no varejo).
- ▶ **Reconhecimento Facial:** Usado em sistemas de segurança para controle de acesso.
- ▶ E muitas outras: veículos autônomos, diagnóstico médico, etc.

# Frameworks de Deep Learning para CNNs

O crescimento do deep learning é impulsionado por frameworks poderosos:

- ▶ **TensorFlow:** Desenvolvido pelo Google (2015). Oferece um conjunto de ferramentas para desenvolvimento e deployment.
- ▶ **Keras:** Framework de alto nível em Python para experimentação rápida. Pode rodar sobre TensorFlow, CNTK, Theano.
- ▶ **PyTorch:** Desenvolvido pela divisão de IA do Facebook (2017). Conhecido por seu grafo computacional dinâmico e eficiência de memória. Popular em NLP e pesquisa.



# Conclusão

- ▶ CNNs são uma ferramenta poderosa para tarefas de reconhecimento de imagem e classificação.
- ▶ Inspiradas no sistema visual humano, seus componentes chave (convolução, pooling, FC) permitem o aprendizado de hierarquias de características.
- ▶ Overfitting é um desafio, mas pode ser mitigado com técnicas de regularização.
- ▶ Frameworks como TensorFlow, Keras e PyTorch facilitam a implementação e o treinamento de CNNs.