



UNINASSAU

REDES NEURAIS ARTIFICIAIS

Professor: André Guerra

Objetivos da aula:

- **Compreender os conceitos fundamentais das CNNs**
 - Introduzir o que são Redes Neurais Convolucionais
 - Discutir a importância e o impacto das CNNs na área de Visão Computacional
- **Explorar a arquitetura e as operações básicas das CNNs**
 - Detalhar as camadas principais e suas funções
 - Explicar como as operações de convolução e pooling funcionam
- **Aplicar CNNs em tarefas práticas de classificação de imagens**
 - Demonstrar a implementação de uma CNN usando Keras
 - Mostrar exemplos de aplicação em tempo real

Definição de Convolução

- **Operação Matemática:**
 - Combinação de dois conjuntos de informações para produzir um terceiro conjunto.
- **Aplicação em Imagens:**
 - Aplicação de filtros (kernels) sobre a imagem para extrair características específicas.

Filtros (Kernels)

- **Tamanho do Filtro:**
 - Comumente 3x3, 5x5, etc.
- **Valores do Filtro:**
 - Determinam o tipo de característica a ser detectada (bordas, texturas, etc.).

Operação de convolução

Stride (Passo)

- Definição:
 - Número de pixels que o filtro se move na imagem.
- Impacto:
 - Controle da sobreposição e tamanho do mapa de características.

Padding (Preenchimento)

- 'Valid':
 - Sem preenchimento, reduzindo o tamanho da saída.
- 'Same':
 - Adiciona preenchimento para manter o tamanho da entrada.

Operação de convolução

Resultado da Convolução

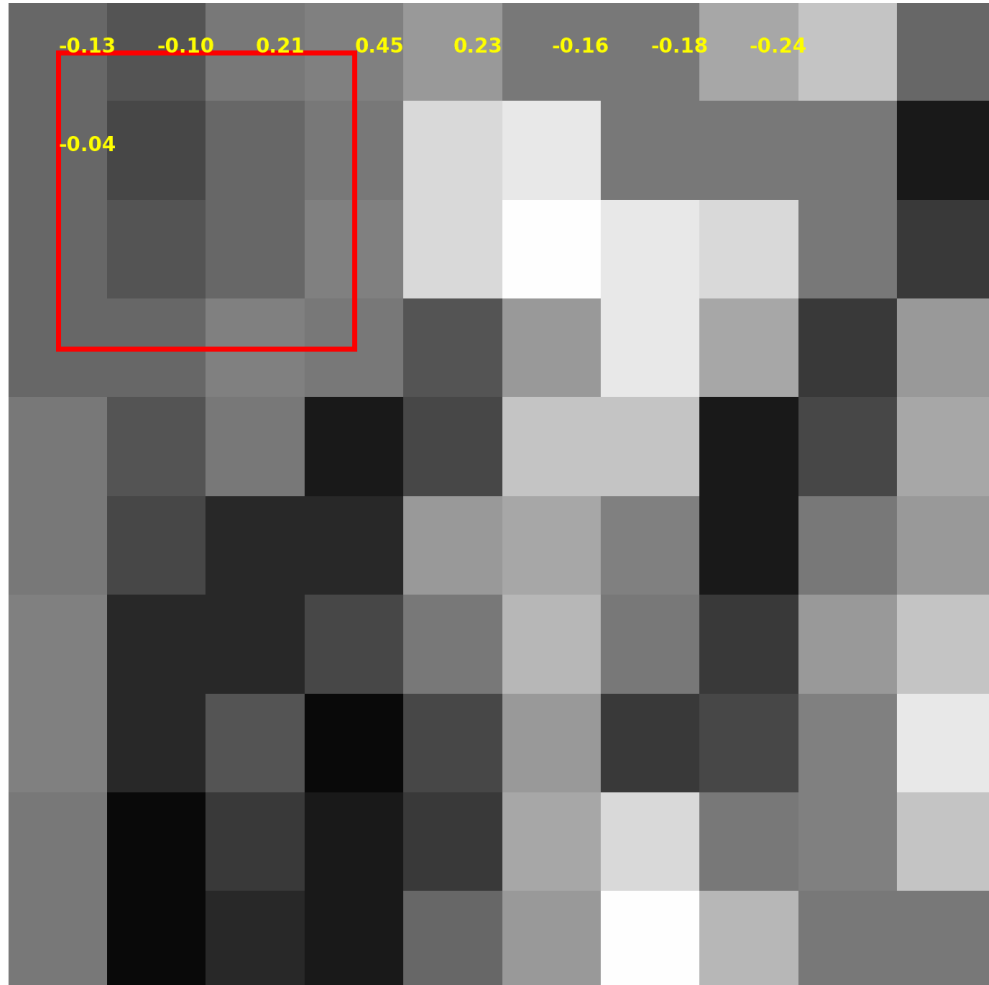
- Mapa de Características:
 - Saída resultante da aplicação do filtro na imagem.

Operação de convolução



UNINASSAU

Imagem de Entrada



Mapa de Características



Operação de convolução



1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

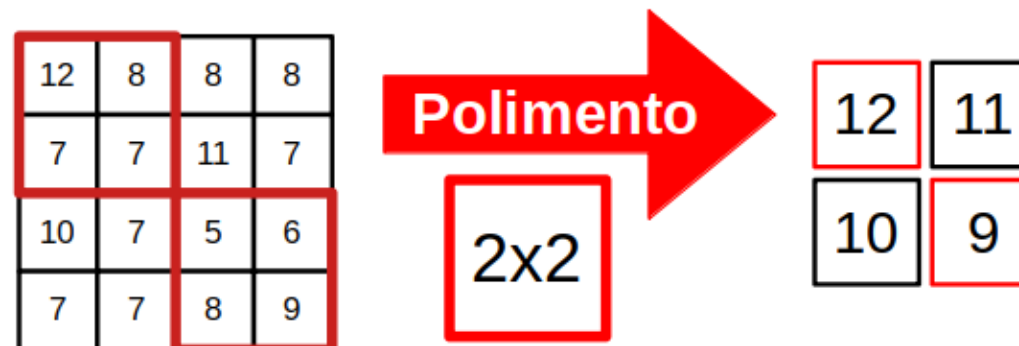
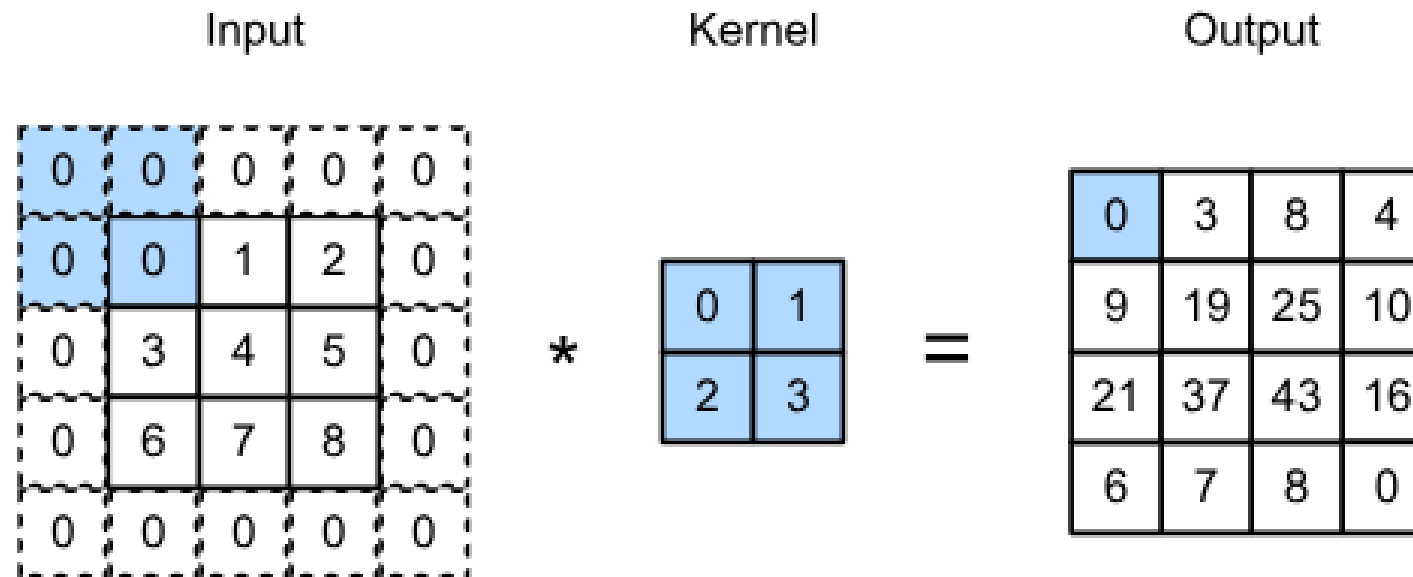
4		

Convolved
Feature

Operação de convolução e pooling



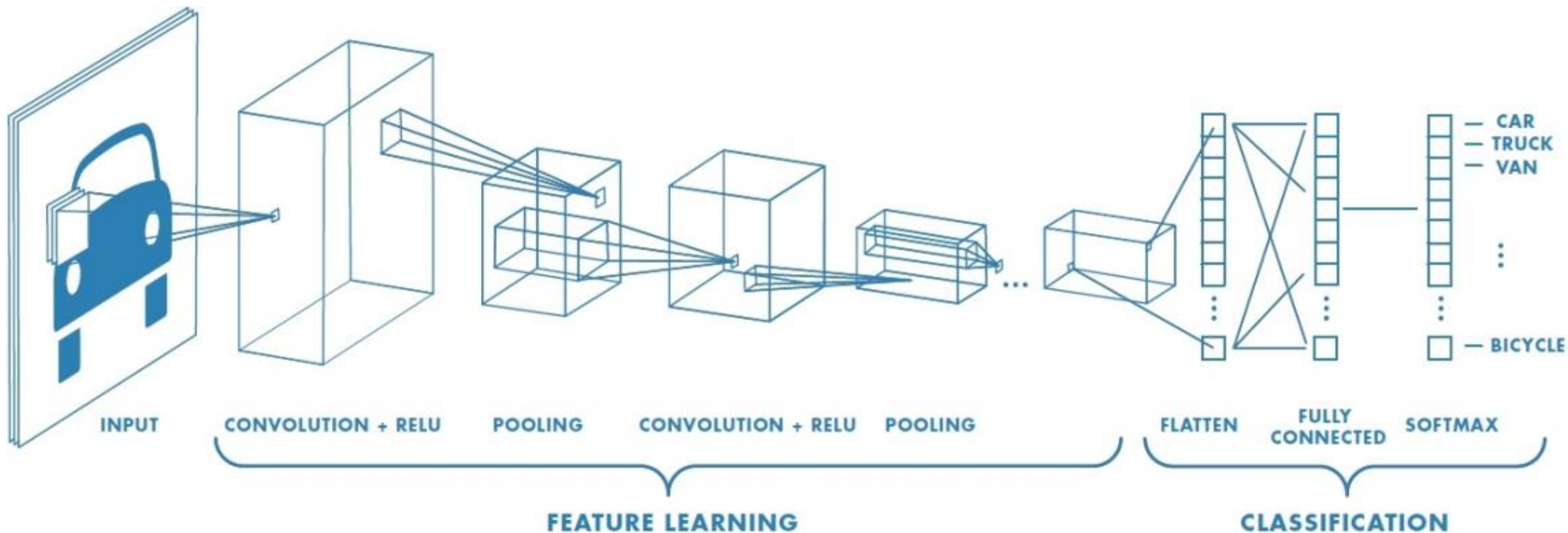
UNINASSAU



Operação de convolução



UNINASSAU



<https://poloclub.github.io/cnn-explainer/>

Camadas Convolucionais

Definição de Camada Convolucional

- **Função Principal:**
 - Extrair características relevantes da imagem de entrada.
- **Operação Básica:**
 - Aplicação de múltiplos filtros (kernels) para detectar diferentes padrões.

Parâmetros Principais

- **Filtros (Kernels):**
 - **Número de Filtros:**
 - Define quantos mapas de características serão gerados.
 - **Tamanho do Filtro:**
 - Comumente 3x3, 5x5, etc., determinando a área da imagem analisada por vez.

- **Aplicação de Filtros:**
 - Cada filtro desliza sobre a imagem de entrada, realizando operações de multiplicação e soma.
- **Extração de Características:**
 - Diferentes filtros detectam diferentes tipos de características (e.g., bordas, texturas).
- **Criação de Mapas de Características:**
 - Cada filtro gera um mapa de características que representa a presença de um padrão específico na imagem.

- O que é Invariância:
 - Capacidade do modelo de reconhecer objetos mesmo após transformações como translação, rotação, escala e distorção.
- Importância:
 - Permite que a CNN generalize melhor em diferentes cenários e condições de imagem.

Tipos de Transformações Invariantes

- **Translação:**
 - Reconhecimento de objetos mesmo que estejam deslocados na imagem.
- **Escala:**
 - Identificação de objetos em diferentes tamanhos.
- **Rotação:**
 - Detecção de objetos independentemente de sua orientação.
- **Distorção:**
 - Capacidade de reconhecer objetos sob deformações não rígidas.

Como as CNNs Atingem Invariância a Transformações

- **Pooling:**
 - Redução da dimensionalidade espacial, ajudando a tornar as características extraídas mais robustas a pequenas variações.
- **Arquiteturas Profundas:**
 - Múltiplas camadas convolucionais e de pooling capturam características em diferentes níveis de abstração.

- **Data Augmentation:**
 - Técnicas de aumento de dados que introduzem variações nas imagens de treinamento, melhorando a robustez do modelo.
- **Uso de Filtros Locais:**
 - Detectam padrões independentes de sua posição na imagem.
- **Melhor Generalização:**
 - O modelo consegue aplicar o que aprendeu a novas imagens com diferentes transformações.
- **Robustez:**
 - Redução da sensibilidade a pequenas variações nas imagens de entrada.
- **Desempenho Elevado:**
 - Melhoria na precisão e eficácia do modelo em tarefas reais de visão computacional.

- **Processamento por Janelas Pequenas:**
 - Convoluções operam em pequenas regiões da imagem, reduzindo a quantidade de cálculos necessários.
- **Reutilização de Filtros:**
 - O mesmo filtro é aplicado em diferentes partes da imagem, economizando recursos computacionais.
- **Paralelismo Massivo:**
 - GPUs podem executar milhares de operações simultaneamente, acelerando o processamento de convoluções.
- **Compatibilidade com Bibliotecas de Deep Learning:**
 - Frameworks como TensorFlow e PyTorch são otimizados para utilizar GPUs de forma eficiente.

Redução de Parâmetros

- **Compartilhamento de Pesos:**
 - Menos parâmetros a serem treinados devido ao compartilhamento de filtros, resultando em modelos mais leves.
- **Memória e Armazenamento:**
 - Modelos com menos parâmetros requerem menos memória e espaço de armazenamento, facilitando o deployment.

Benefícios para Treinamento e Inferência

- **Treinamento Mais Rápido:**
 - Menor quantidade de cálculos e uso eficiente de GPUs reduzem o tempo necessário para treinar modelos.
- **Inferência em Tempo Real:**
 - Modelos eficientes permitem classificações rápidas, essenciais para aplicações em tempo real como reconhecimento de imagens em vídeo.

Reconhecimento de Imagens

- **Alta Precisão:**
 - As CNNs alcançam taxas de acerto superiores em comparação com métodos tradicionais de reconhecimento de padrões.
- **Capacidade de Aprendizado Profundo:**
 - Capazes de aprender representações complexas e hierárquicas das imagens, melhorando a precisão do reconhecimento.

Detecção de Objetos

- **Localização e Classificação Simultâneas:**
 - CNNs podem identificar e localizar múltiplos objetos dentro de uma única imagem de forma eficiente.
- **Algoritmos Avançados:**
 - Métodos como YOLO (You Only Look Once) e Faster R-CNN utilizam CNNs para detecção rápida e precisa.

Desempenho Superior em Tarefas de Visão Computacional



UNINASSAU

Segmentação Semântica

- **Divisão Precisa da Imagem:**
 - Classificação de cada pixel da imagem para identificar diferentes objetos e regiões.
- **Aplicações em Medicina e Automação:**
 - Utilizada para análise de imagens médicas, inspeção industrial e condução autônoma.

Reconhecimento Facial

- **Identificação e Verificação:**
 - CNNs são amplamente utilizadas em sistemas de segurança para reconhecimento e verificação de identidades faciais.
- **Robustez a Variações:**
 - Capazes de lidar com diferentes expressões, iluminação e ângulos de visão.

Classificação de Vídeos

- **Análise Temporal:**
 - Extensão das CNNs para capturar informações temporais em sequências de vídeo, melhorando a compreensão de ações e eventos.
- **Aplicações em Vigilância e Entretenimento:**
 - Utilizadas para monitoramento de atividades, detecção de comportamentos suspeitos e análise de conteúdo multimídia.

Transferência de Aprendizado

- **Uso de Modelos Pré-Treinados:**
 - Aproveitamento de redes treinadas em grandes conjuntos de dados para melhorar o desempenho em tarefas específicas com menos dados.
- **Redução de Tempo de Treinamento:**
 - Minimiza a necessidade de treinamento extensivo, tornando o desenvolvimento mais rápido e eficiente.

Arquitetura Completa de uma CNN

Estrutura Geral da CNN

- **Camadas Convolucionais:**
 - Extraem características da imagem de entrada usando filtros.
- **Camadas de Pooling:**
 - Reduzem a dimensionalidade dos mapas de características.
- **Camadas Densas (Fully Connected):**
 - Conectam todas as neuronas das camadas anteriores para realizar a classificação final.
- **Camada de Saída:**
 - Produz a previsão final, como a probabilidade das classes.

Arquitetura Completa de uma CNN

Múltiplas Camadas Convolucionais e de Pooling

- **Hierarquia de Características:**
 - Camadas iniciais capturam características simples (bordas, texturas).
 - Camadas profundas capturam características mais complexas (formas, objetos).
- **Empilhamento de Camadas:**
 - Combinação de várias camadas convolucionais e de pooling para aumentar a profundidade da rede.

Arquitetura Completa de uma CNN

Camada Densa (Fully Connected)

- **Integração das Características:**
 - Combina todas as características extraídas pelas camadas convolucionais e de pooling.
- **Classificação:**
 - Processa as informações para classificar a imagem nas classes desejadas.

Camada de Saída

- **Função de Ativação:**
 - **Softmax:** Usada para classificação multi-classe, fornecendo probabilidades para cada classe.
 - **Sigmoid:** Usada para classificação binária.
- **Interpretação dos Resultados:**
 - A classe com a maior probabilidade é selecionada como a previsão final.

Preparação dos Dados

- **Carregamento e Pré-processamento:**
 - Importação dos dados de imagem.
 - Normalização e redimensionamento das imagens.
- **Divisão dos Dados:**
 - Separação em conjuntos de treinamento, validação e teste.

Construção do Modelo CNN

- **Definição das Camadas:**
 - Adição de camadas convolucionais, de pooling e densas.
- **Escolha da Função de Ativação:**
 - Uso de ReLU nas camadas ocultas e Softmax na camada de saída.
- **Compilação do Modelo:**
 - Seleção da função de perda e do otimizador (e.g., categorical_crossentropy e Adam).

Treinamento do Modelo

- **Ajuste de Hiperparâmetros:**
 - Definição de epochs, batch size e learning rate.
- **Monitoramento do Treinamento:**
 - Observação das métricas de perda e acurácia.

Avaliação e Validação

- **Avaliação no Conjunto de Teste:**
 - Medição da acurácia e outras métricas no conjunto de dados de teste.
- **Análise de Resultados:**
 - Interpretação dos resultados e identificação de possíveis melhorias.

Inferência e Uso em Tempo Real

- **Carregamento de Imagens para Previsão:**
 - Aplicação do modelo treinado para classificar novas imagens.
- **Implementação em Aplicações Práticas:**
 - Integração do modelo em sistemas como aplicativos móveis ou web.

Conclusão

Recapitulando os pontos principais...