

Deep Learning: Visão Geral

Deep Learning representa o aprendizado com redes neurais profundas, onde a característica central é o aprendizado automático de representações. Diferentemente do Machine Learning clássico, requer menos feature engineering e permite mais aprendizado fim-a-fim.

O marco histórico da consolidação prática ocorreu a partir de 2012-2013, impulsionado pela disponibilidade de grandes volumes de dados e pelo poder computacional das GPUs.

Eduardo Ogasawara

eduardo.ogasawara@cefet-rj.br
<https://eic.cefet-rj.br/~eogasawara>

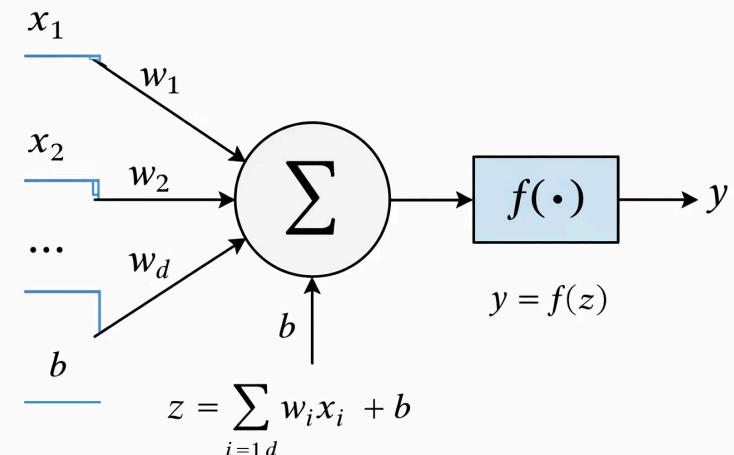
Neurônio Artificial: O Modelo Básico

Um neurônio artificial computa uma soma ponderada das entradas e aplica uma função de não linearidade para produzir a saída.

$$z = \sum_{i=1}^d w_i x_i + b$$

$$y = f(z)$$

Onde **wi** são os pesos, **b** é o viés, e **f(·)** é a função de ativação.



Perceptron e Suas Limitações

O Perceptron

Modelo de classificador linear que processa entradas através de uma única camada de neurônios.

Limitação Fundamental

Não consegue resolver problemas não linearmente separáveis, como o clássico problema XOR.

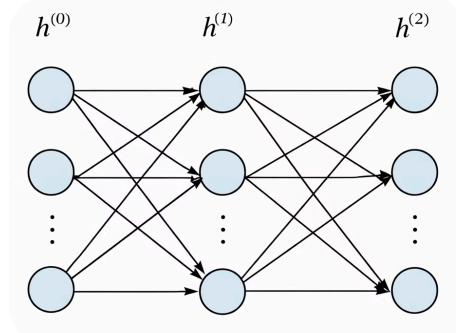
Motivação para Evolução

Essa limitação motivou o desenvolvimento de redes com múltiplas camadas, capazes de aprender representações mais complexas.

Redes Neurais Multicamadas (MLP)

As redes multicamadas empilham várias camadas de neurônios, onde cada camada aprende uma representação intermediária dos dados. Esta arquitetura permite capturar padrões complexos e não lineares.

$$\mathbf{h}^{(l)} = f \left(W^{(l)} \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)} \right)$$



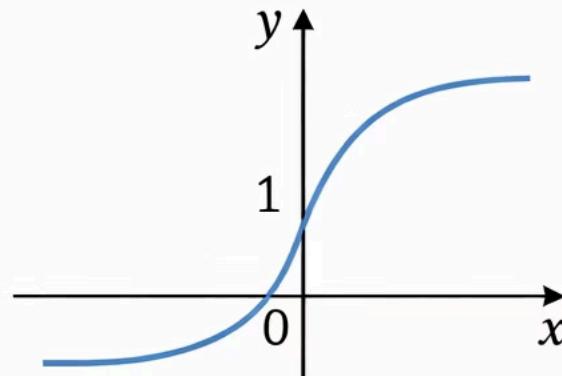
Cada camada transforma a representação da camada anterior, construindo abstrações progressivamente mais sofisticadas.

Funções de Ativação: A Essência da Não Linearidade

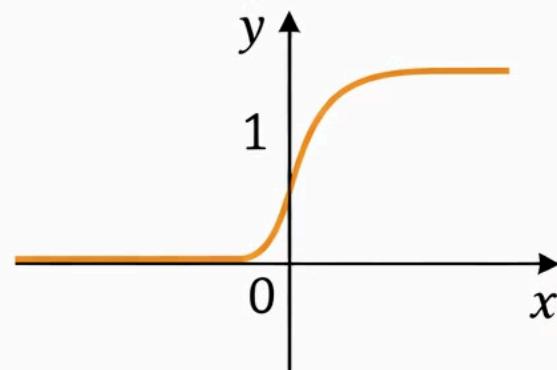
As funções de ativação introduzem não linearidade nas redes neurais, permitindo que aprendam padrões complexos. As principais funções incluem:

- **Sigmoid:** Comprime valores entre 0 e 1
- **Tanh:** Comprime valores entre -1 e 1
- **ReLU:** Padrão moderno, mitiga vanishing gradient $\text{ReLU}(x) = \max(0, x)$

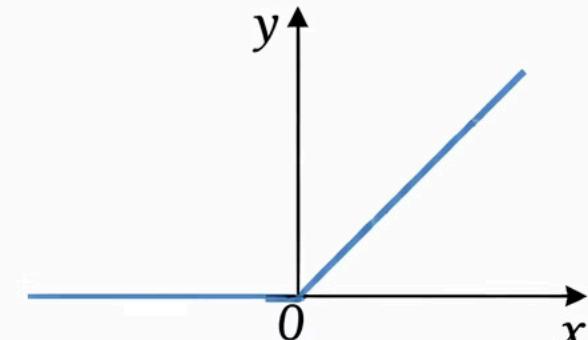
A ReLU se tornou o padrão por sua simplicidade e eficácia em mitigar o problema de gradiente desvanecente.



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



$$\text{ReLU}(x) = \max(0, x)$$

Função de Perda: O Objetivo do Aprendizado

Conceito Central

A função de perda mede a discrepância entre a predição do modelo e o valor real esperado.

Regressão: MSE

Mean Squared Error é utilizado para problemas de regressão, medindo o erro quadrático médio.

Classificação: Cross-Entropy

Cross-Entropy é a escolha padrão para problemas de classificação.

$$\mathcal{L}_{CE} = - \sum_k y_k \log(\hat{y}_k)$$

Aprendizado por Gradiente

O objetivo fundamental do treinamento é minimizar a função de perda através da otimização dos parâmetros da rede:

$$\min_{\theta} \mathcal{L}(\theta)$$

A atualização básica do Gradient Descent segue a regra:

$$\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}$$

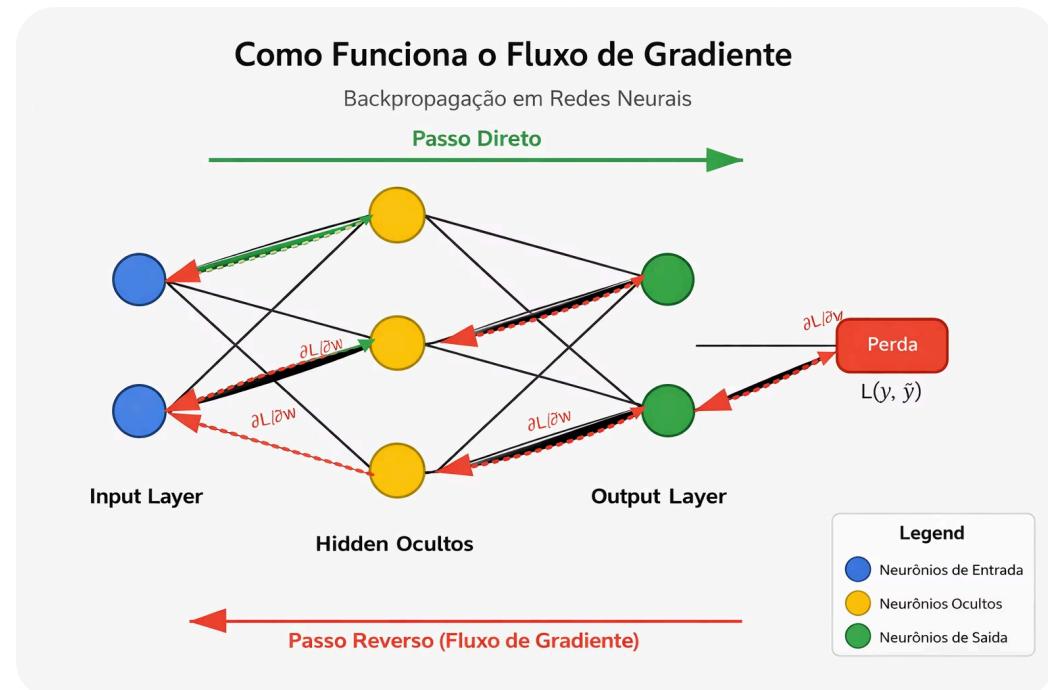
Onde η representa a taxa de aprendizado, um hiperparâmetro crucial que controla o tamanho dos passos na direção do gradiente negativo.

Backpropagation: A Ideia Central

Backpropagation é o algoritmo eficiente para calcular gradientes em redes neurais, utilizando a regra da cadeia do cálculo.

O algoritmo propaga o erro da camada de saída para as camadas iniciais, permitindo que cada peso seja atualizado proporcionalmente à sua contribuição para o erro total.

Este é o fundamento de todo Deep Learning supervisionado, tornando possível treinar redes com milhões de parâmetros.



Por Que Redes Profundas Funcionam



Nível Baixo

Camadas iniciais detectam características básicas como bordas e texturas em imagens, ou palavras em texto.

Nível Médio

Camadas intermediárias combinam características básicas em formas e padrões mais complexos, ou frases em linguagem.

Nível Alto

Camadas finais reconhecem objetos completos em imagens ou significado semântico em texto.

As camadas aprendem abstrações hierárquicas progressivamente mais sofisticadas. Quanto mais profunda a rede, maior seu poder representacional para capturar padrões complexos nos dados.



CNN

Redes Convolucionais: Motivação

Redes Neurais Convolucionais (CNNs) foram desenvolvidas para processar dados com estrutura espacial, como imagens, sinais e até texto.

Problema do MLP

Redes totalmente conectadas requerem um número excessivo de parâmetros para processar imagens, tornando o treinamento impraticável.

Solução da CNN

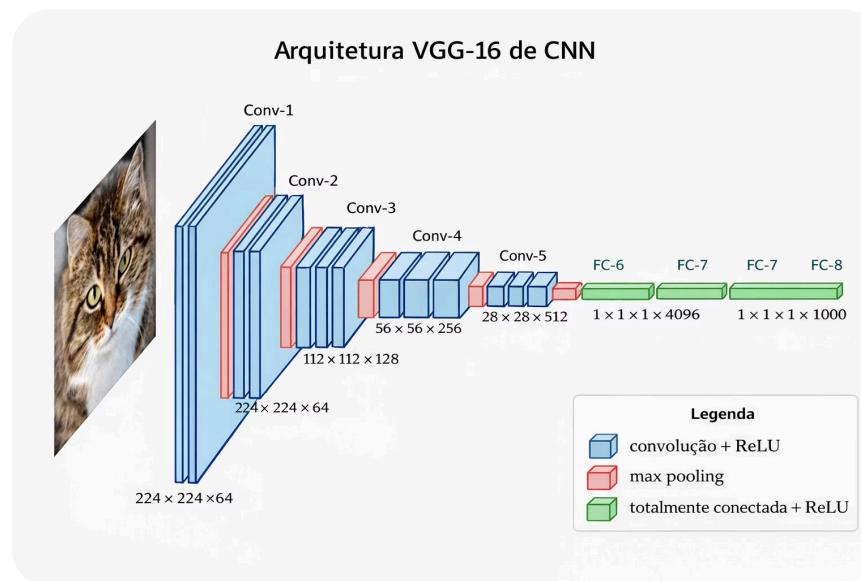
CNNs exploram a localidade espacial e o compartilhamento de pesos, reduzindo drasticamente o número de parâmetros.

Convolução: A Ideia Matemática

A operação de convolução aplica um kernel (filtro) localmente sobre a entrada:

$$(\mathbf{x} * \mathbf{k})(i) = \sum_j \mathbf{x}_{i+j} \mathbf{k}_j$$

A mesma operação é aplicada em toda a entrada através do compartilhamento de pesos, o que reduz drasticamente o número de parâmetros comparado a camadas totalmente conectadas.



Pooling e Invariância



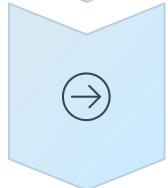
Agregação Local

Pooling realiza agregação local, como max-pooling, que seleciona o valor máximo em cada região.



Redução Dimensional

Reduz a dimensionalidade espacial dos dados, diminuindo o custo computacional.



Invariância

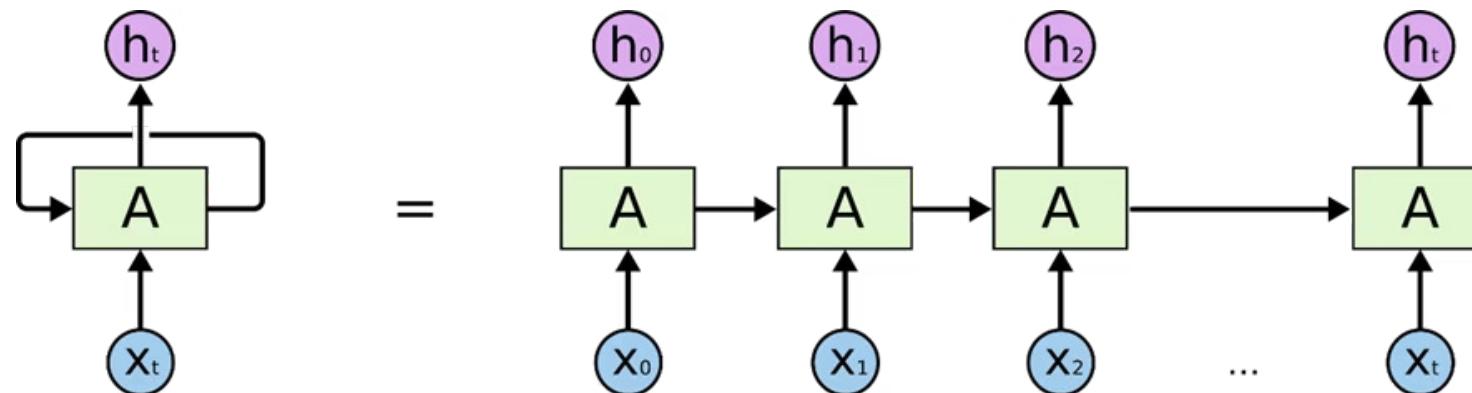
Introduz invariância a pequenas translações, tornando a rede mais robusta.

RNN: Processando Dados Sequenciais

Redes Neurais Recorrentes (RNNs) foram desenvolvidas para processar sequências, como texto e séries temporais. O estado oculto carrega memória de informações passadas:

$$\mathbf{h}_t = f(W_x \mathbf{x}_t + W_h \mathbf{h}_{t-1})$$

Esta arquitetura permite que a rede mantenha contexto ao longo da sequência, processando cada elemento considerando o histórico anterior.



Limitações de RNN e a Solução LSTM

O Problema

RNNs tradicionais sofrem com dependências longas, perdendo informação ao longo de sequências extensas devido ao problema de gradiente desvanecente.

A Solução

LSTM (Long Short-Term Memory) introduz portas de controle que regulam o fluxo de informação:

- **Porta de esquecer:** decide o que descartar
- **Porta de atualizar:** decide o que armazenar
- **Porta de saída:** decide o que expor

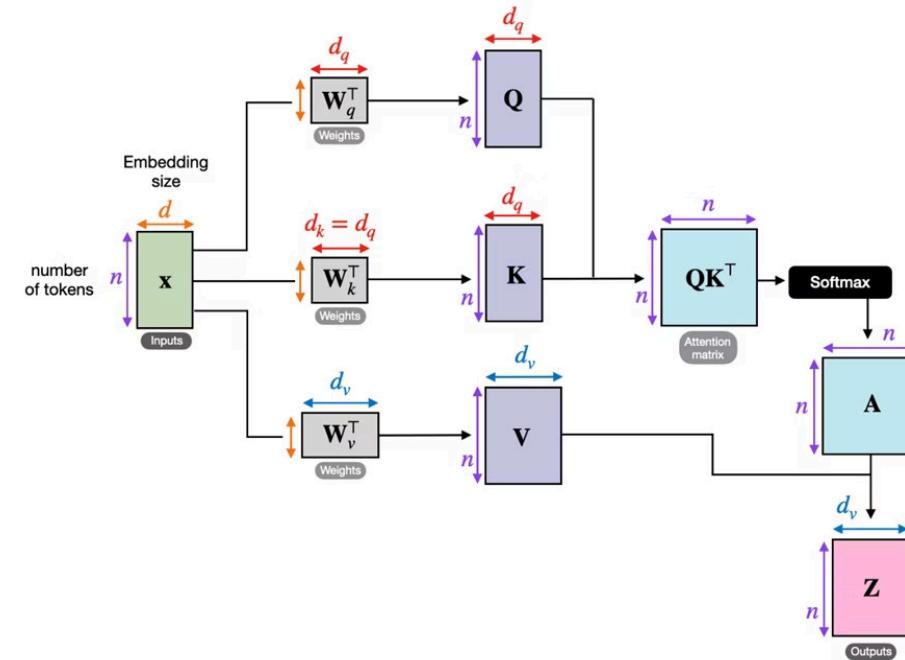
Isso permite memória de longo prazo efetiva.

Atenção: A Ideia Central

O mecanismo de atenção reconhece que nem toda informação passada é igualmente relevante para a tarefa atual. Ele pondera dinamicamente as contribuições de diferentes partes da entrada:

$$\mathbf{c}_j = \sum_i \alpha_{ji} \mathbf{h}_i$$

Os pesos de atenção α_{ji} são aprendidos pela rede, permitindo focar nas partes mais relevantes da sequência.



- Este conceito é a base fundamental para arquiteturas Transformers e Graph Attention Networks (GAT), revolucionando o processamento de linguagem natural e grafos.

Por Que Grafos São Diferentes



Dados Não-Euclidianos

Grafos não possuem estrutura de grade regular como imagens ou sequências.



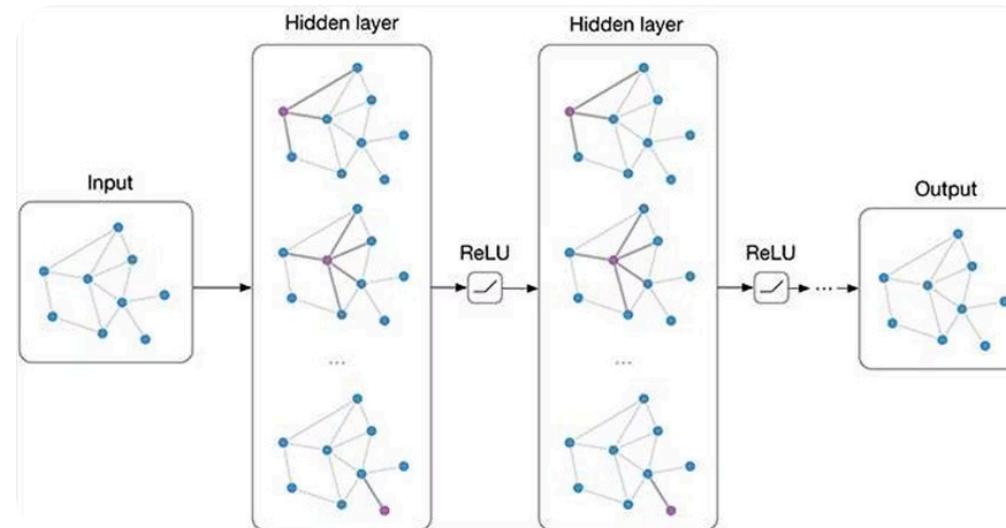
Vizinhanças Irregulares

Cada nó pode ter um número diferente de vizinhos, sem padrão fixo.



Invariância à Permutação

A ordem dos nós não importa - o grafo permanece o mesmo sob permutações.



GNNs: Message Passing Neural Networks

A ideia central das Graph Neural Networks é o paradigma de message passing, onde cada nó agrega informação de seus vizinhos:

$$\mathbf{h}_v^{(k+1)} = \text{UPDATE} \left(\mathbf{h}_v^{(k)}, \text{AGG}\{\mathbf{h}_u^{(k)} : u \in \mathcal{N}(v)\} \right)$$

Este processo é repetido por **k** camadas, permitindo que cada nó alcance informações de vizinhos cada vez mais distantes no grafo. A cada camada, o campo receptivo de cada nó se expande.

GraphSAGE: Agregação Escalável

01

Agregação Explícita

GraphSAGE define funções de agregação explícitas como média, max-pooling ou LSTM para combinar informações dos vizinhos.

02

Aprendizado de Funções

O modelo aprende as melhores funções de agregação para cada camada, adaptando-se aos dados.

03

Escalabilidade

Projetado para ser escalável a grafos muito grandes através de amostragem de vizinhos, tornando-o prático para aplicações reais.



Graph Attention Networks (GAT)

Graph Attention Networks generalizam o conceito de atenção para grafos, permitindo que cada nó aprenda a importância relativa de seus vizinhos:

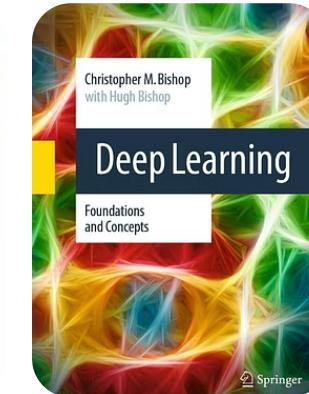
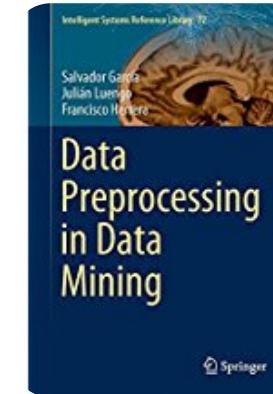
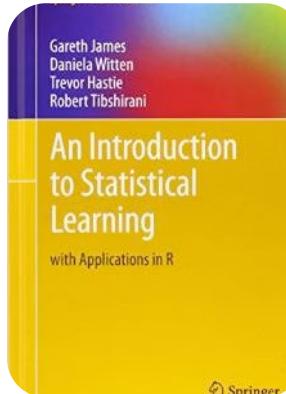
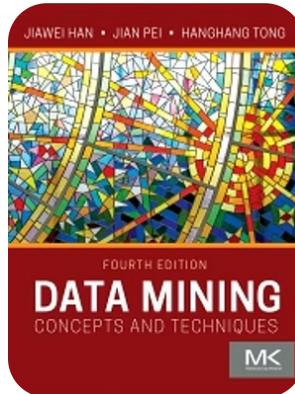
$$\alpha_{vu} = \text{softmax}(a(\mathbf{h}_v, \mathbf{h}_u))$$

Os pesos de atenção α_{vu} são aprendidos dinamicamente, indicando quais vizinhos são mais relevantes para cada nó. Isso permite que a rede foque nas conexões mais importantes do grafo.

- ❑ GAT combina o poder da atenção com a estrutura de grafos, oferecendo flexibilidade e interpretabilidade superior em comparação com métodos de agregação fixa.

Referências Principais

Esta seleção de referências representa os pilares fundamentais para o estudo aprofundado de mineração de dados, cobrindo desde conceitos básicos até técnicas avançadas e aplicações contemporâneas.



1. **J. Han, J. Pei, and H. Tong**, *Data Mining: Concepts and Techniques*, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022.
2. **G. M. James, D. Witten, T. Hastie, and R. Tibshirani**, *An Introduction to Statistical Learning: With Applications in R*. Springer Nature, 2021.
3. **S. Garcia, J. Luengo, and F. Herrera**, *Data Preprocessing in Data Mining*. Springer, 2014.
4. **C. M. Bishop and H. Bishop**, *Deep Learning: Foundations and Concepts*. Springer Nature, 2023.