

# Attention Models: da motivação às variantes modernas

Intuição geométrica, formulação matemática e aplicações

**Seu Nome**

16 de agosto de 2025

# Table of Contents

## 1 Motivação e Histórico

- ▶ **Motivação e Histórico**
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

# Redes Neurais Recorrentes

## 1 Motivação e Histórico

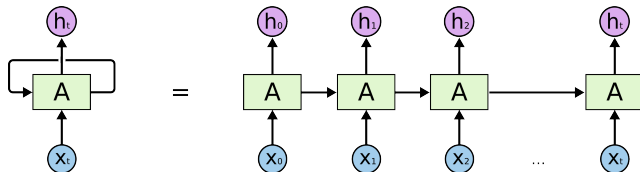


Figura: Redes Neurais Recorrentes <sup>1</sup>

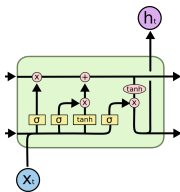
- Bem adaptadas para dados sequenciais como séries temporais e texto
- Diferentes tipos de modelos: LSTM, GRU
- Diferentes arquiteturas: simples, bidirecional, encoder-decoder

<sup>1</sup>Figura de Christopher Olah

# Tipos de camadas RNN

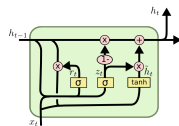
## 1 Motivação e Histórico

### LSTM



$$\begin{aligned}
 f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f), & i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i), \\
 \tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c), & C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \\
 o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), & h_t &= o_t \odot \tanh(C_t)
 \end{aligned}$$

### GRU



$$\begin{aligned}
 z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\
 r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\
 \tilde{h}_t &= \tanh(W \cdot [r_t \odot h_{t-1}, x_t]) \\
 h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t
 \end{aligned}$$

# LSTM como baseline para NMT

## 1 Motivação e Histórico

- Encoder LSTM lê  $(x_1, \dots, x_n)$  e produz estados  $h_t$ ; o contexto é o último estado  $c = h_n$ .
- Decoder LSTM gera  $(y_1, \dots, y_m)$  condicionado a  $c$ .
- **Gargalo:** toda a informação comprimida em  $c = h_n$ .
  - Processamento **sequencial**  $\Rightarrow$  baixa paralelização.
  - **Dependências longas** ainda são difíceis (mesmo com portas).
  - **Gargalo do contexto** (vetor único) degrada qualidade em frases longas.

# Atenção aditiva (Bahdanau, 2014)

## 1 Motivação e Histórico

$$e_{ij} = a(s_{i-1}, h_j), \quad \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \quad c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

- $h_j$ : estado oculto do encoder na posição  $j$  (palavra  $x_j$ ).
- $s_{i-1}$ : estado do decoder no passo anterior ( $y_{i-1}$ ).
- $e_{ij}$ : escore de alinhamento entre  $h_j$  e  $s_{i-1}$  (via rede feedforward).
- $\alpha_{ij}$ : pesos normalizados (softmax)  $\rightarrow$  distribuem a atenção sobre os  $h_j$ .
- $c_i$ : vetor de contexto dinâmico usado para prever  $y_i$ .

**Intuição:** O decoder calcula, em cada passo, um mapa de atenção sobre os estados do encoder, decidindo onde focar.

# Integração com encoder bidirecional e decoder

## 1 Motivação e Histórico

- O **encoder** é uma RNN bidirecional:

$$h_j = [\vec{h}_j; \overleftarrow{h}_j]$$

Cada  $h_j$  contém contexto passado e futuro da palavra  $x_j$ .

- O vetor de contexto  $c_i$  é construído a partir desses estados bidirecionais.
- O **decoder** (RNN unidirecional) atualiza seu estado com:

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

- Usa o estado anterior  $s_{i-1}$  - O símbolo anterior  $y_{i-1}$  - O contexto dinâmico  $c_i$

- Assim, a cada passo, o decoder combina memória interna + contexto dinâmico para prever  $y_i$ .

**Resultado:** Resolve o gargalo do vetor fixo único ( $h_n$ ) e permite traduções mais fiéis em frases longas.

# Atenção multiplicativa (Luong, 2015)

## 1 Motivação e Histórico

### Três variantes de scoring:

$$e_{ij} = v^\top \tanh(W[s_j; h_i]) \quad (\text{concat, similar ao Bahdanau})$$

$$e_{ij} = s_j^\top W h_i \quad (\text{general})$$

$$e_{ij} = s_j^\top h_i \quad (\text{dot})$$

- $s_j$ : query  $\rightarrow$  estado oculto do decoder.
- $h_i$ : key/value  $\rightarrow$  estado do encoder.
- **Concat**: aproxima-se da atenção aditiva de Bahdanau.
- **General**: bilinear, mais expressivo (aprende  $W$ ).
- **Dot**: mais simples e rápido (nenhum parâmetro extra).

**Nota:** Luong mostra que todas podem ser reinterpretadas em termos modernos como  $Q = s_j, K = h_i, V = h_i$ .



# Global vs Local Attention (Luong, 2015)

## 1 Motivação e Histórico

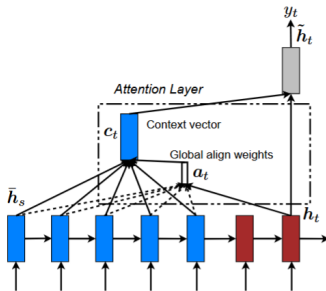


Figure 2: **Global attentional model** – at each time step  $t$ , the model infers a *variable-length* alignment weight vector  $a_t$  based on the current target state  $h_t$  and all source states  $\bar{h}_s$ . A global context vector  $c_t$  is then computed as the weighted average, according to  $a_t$ , over all the source states.

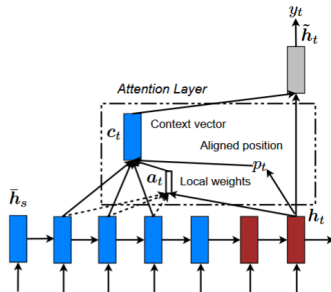


Figure 3: **Local attention model** – the model first predicts a single aligned position  $p_t$  for the current target word. A window centered around the source position  $p_t$  is then used to compute a context vector  $c_t$ , a weighted average of the source hidden states in the window. The weights  $a_t$  are inferred from the current target state  $h_t$  and those source states  $\bar{h}_s$  in the window.

# Self-Attention: dependências em paralelo

## 1 Motivação e Histórico

$$\text{Att}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V, \quad Q = XW_Q, \quad K = XW_K, \quad V = XW_V.$$

- Calcula relações *entre todos os tokens* da mesma sequência, **em paralelo**.
- Multi-head:

$$\text{MHA}(X) = \text{Concat}(H_1, \dots, H_h) W_O, \quad H_r = \text{softmax}\left(\frac{Q_r K_r^\top}{\sqrt{d_k}}\right) V_r.$$

- Comparativo: RNN/LSTM exige  $n$  passos sequenciais; self-attention faz um passo paralelo com custo  $\mathcal{O}(n^2)$ .

# Arquitetura pré-transformer

## 1 Motivação e Histórico

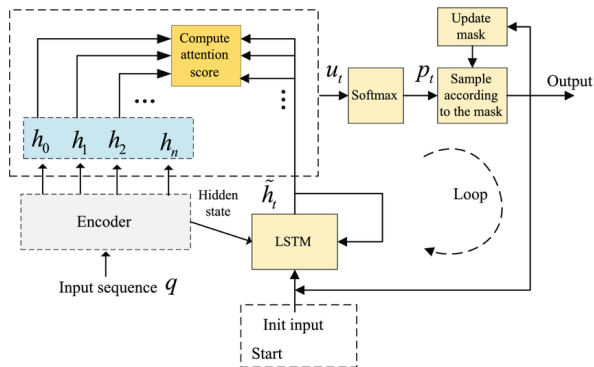


Figura: Exemplo de arquitetura pré-transformer (aditiva)

# Attention Is All You Need: nascendo o Transformer

## 1 Motivação e Histórico

- **Remove** completamente a recorrência (sem LSTM).
- **Positional encodings** preservam ordem:

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \quad PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right).$$

- Cada **bloco Transformer** (pré-norm, forma comum):

$$Y = X + \text{MHA}(\text{LN}(X)),$$

$$Z = Y + \text{FFN}(\text{LN}(Y)), \quad \text{FFN}(u) = W_2 \phi(W_1 u + b_1) + b_2,$$

- Empilha-se vários blocos de atenção+FFN  $\Rightarrow$  **arquitetura Transformer**.

# Máscara causal e síntese da evolução

## 1 Motivação e Histórico

**Máscara causal** (para LMs) impede olhar o futuro:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right), \quad M_{ij} = \begin{cases} 0, & j \leq i \\ -\infty, & j > i \end{cases}$$

**Linha do tempo (síntese):**

- LSTM encoder-decoder: contexto único  $c = h_n$  (gargalo).
- LSTM + **atenção** (Bahdanau/Luong): alívio do gargalo.
- **Self-attention**: dependências longas em paralelo.
- **Transformer**: atenção + posição + FFN; várias camadas empilhadas; sem LSTM.

# Table of Contents

## 2 Roteiro

- ▶ Motivação e Histórico
- ▶ **Roteiro**
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

# Roteiro da Apresentação

## 2 Roteiro

- Motivação e histórico

# Roteiro da Apresentação

## 2 Roteiro

- Motivação e histórico
- Modelos de sequência (RNN/LSTM) + Attention



# Roteiro da Apresentação

## 2 Roteiro

- Motivação e histórico
- Modelos de sequência (RNN/LSTM) + Attention
- Embeddings, interpretação geométrica e matemática

# Roteiro da Apresentação

## 2 Roteiro

- Motivação e histórico
- Modelos de sequência (RNN/LSTM) + Attention
- Embeddings, interpretação geométrica e matemática
- Transformers e posicionais

# Roteiro da Apresentação

## 2 Roteiro

- Motivação e histórico
- Modelos de sequência (RNN/LSTM) + Attention
- Embeddings, interpretação geométrica e matemática
- Transformers e posicionais
- Aplicações: Séries Temporais, Language Models, ViT, GAT

# Roteiro da Apresentação

## 2 Roteiro

- Motivação e histórico
- Modelos de sequência (RNN/LSTM) + Attention
- Embeddings, interpretação geométrica e matemática
- Transformers e posicionais
- Aplicações: Séries Temporais, Language Models, ViT, GAT
- Treinamento, variantes e hiperparâmetros

# Roteiro da Apresentação

## 2 Roteiro

- Motivação e histórico
- Modelos de sequência (RNN/LSTM) + Attention
- Embeddings, interpretação geométrica e matemática
- Transformers e posicionais
- Aplicações: Séries Temporais, Language Models, ViT, GAT
- Treinamento, variantes e hiperparâmetros
- Tamanho ideal (scaling), Mixture of Experts

# Roteiro da Apresentação

## 2 Roteiro

- Motivação e histórico
- Modelos de sequência (RNN/LSTM) + Attention
- Embeddings, interpretação geométrica e matemática
- Transformers e posicionais
- Aplicações: Séries Temporais, Language Models, ViT, GAT
- Treinamento, variantes e hiperparâmetros
- Tamanho ideal (scaling), Mixture of Experts
- Pruning e Distillation

# Roteiro da Apresentação

## 2 Roteiro

- Motivação e histórico
- Modelos de sequência (RNN/LSTM) + Attention
- Embeddings, interpretação geométrica e matemática
- Transformers e posicionais
- Aplicações: Séries Temporais, Language Models, ViT, GAT
- Treinamento, variantes e hiperparâmetros
- Tamanho ideal (scaling), Mixture of Experts
- Pruning e Distillation
- Limitações, diagnóstico e conclusões

# Motivação & Histórico

2 Roteiro



# Table of Contents

## 3 Motivação

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ **Motivação**
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

## Por que Atenção?

### 3 Motivação

- Capturar dependências longas e filtrar informação relevante

## Por que Atenção?

### 3 Motivação

- Capturar dependências longas e filtrar informação relevante
- RNNs sofrem com gradientes e paralelização limitada

## Por que Atenção?

### 3 Motivação

- Capturar dependências longas e filtrar informação relevante
- RNNs sofrem com gradientes e paralelização limitada
- Pesos de atenção ajudam na interpretabilidade

## Por que Atenção?

### 3 Motivação

- Capturar dependências longas e filtrar informação relevante
- RNNs sofrem com gradientes e paralelização limitada
- Pesos de atenção ajudam na interpretabilidade
- Melhor relação capacidade/compute em diversas tarefas

## Histórico em 3 passos

### 3 Motivação

- **Additive Attention** (Bahdanau) para NMT

## Histórico em 3 passos

### 3 Motivação

- **Additive Attention** (Bahdanau) para NMT
- **Dot-Product Attention** (Luong) — caminho para escalabilidade

## Histórico em 3 passos

### 3 Motivação

- **Additive Attention** (Bahdanau) para NMT
- **Dot-Product Attention** (Luong) — caminho para escalabilidade
- **Transformers** (Vaswani et al., 2017): “Attention is All You Need”



# Fundamentos

## 3 Motivação

# Table of Contents

## 4 Modelos de Sequência

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ **Modelos de Sequência**
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões
- ▶ Backup (Opcional)

# RNN/GRU/LSTM — limites práticos

## 4 Modelos de Sequência

- Cálculo sequencial  $\Rightarrow$  baixa paralelização

# RNN/GRU/LSTM — limites práticos

## 4 Modelos de Sequência

- Cálculo sequencial  $\Rightarrow$  baixa paralelização
- Dependências longas: vanish/explode (mitigado, não resolvido)

# RNN/GRU/LSTM — limites práticos

## 4 Modelos de Sequência

- Cálculo sequencial  $\Rightarrow$  baixa paralelização
- Dependências longas: vanish/explode (mitigado, não resolvido)
- Memória finita e custo de treino elevado para contextos longos

# LSTM + Attention (Encoder-Decoder)

Atenção aditiva (Bahdanau)

## Equações

$$e_{ij} = v^{\top} \tanh(W_1 h_i + W_2 s_j), \quad \alpha_{ij} = \text{softmax}_i(e_{ij}), \quad c_j = \sum_i \alpha_{ij} h_i$$

- **Alinhamento** dinâmico entre estados do encoder e passos do decoder

# LSTM + Attention (Encoder-Decoder)

Atenção aditiva (Bahdanau)

## Equações

$$e_{ij} = v^{\top} \tanh(W_1 h_i + W_2 s_j), \quad \alpha_{ij} = \text{softmax}_i(e_{ij}), \quad c_j = \sum_i \alpha_{ij} h_i$$

- **Alinhamento** dinâmico entre estados do encoder e passos do decoder
- Reduz gargalos de um único vetor de contexto

# Table of Contents

## 5 Embeddings & Interpretações

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ **Embeddings & Interpretações**
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões



# Word/Subword Embeddings

## 5 Embeddings & Interpretações

- Estáticos (Word2Vec, GloVe) vs. Contextuais (ELMo, BERT)

# Word/Subword Embeddings

## 5 Embeddings & Interpretações

- Estáticos (Word2Vec, GloVe) vs. Contextuais (ELMo, BERT)
- Subword (BPE/Unigram) para robustez morfológica

# Word/Subword Embeddings

## 5 Embeddings & Interpretações

- Estáticos (Word2Vec, GloVe) vs. Contextuais (ELMo, BERT)
- Subword (BPE/Unigram) para robustez morfológica
- Análogos em outras modalidades: patches (ViT), time2vec (TS), node2vec (grafos)

# Interpretação Geométrica da Atenção

## 5 Embeddings & Interpretações

- $Q, K, V$  são projeções lineares do latente

# Interpretação Geométrica da Atenção

## 5 Embeddings & Interpretações

- $Q, K, V$  são projeções lineares do latente
- Similaridade (coseno/dot) guia uma *combinação convexa* de  $V$

# Interpretação Geométrica da Atenção

## 5 Embeddings & Interpretações

- $Q, K, V$  são projeções lineares do latente
- Similaridade (coseno/dot) guia uma *combinação convexa* de  $V$
- Multi-head  $\Rightarrow$  múltiplas métricas/projeções simultâneas

# Interpretação Matemática: Self-Attention

## 5 Embeddings & Interpretações

### Fórmula central (scaled dot-product)

$$\text{Att}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right) V$$

- Máscara causal/atencional conforme a tarefa

# Interpretação Matemática: Self-Attention

## 5 Embeddings & Interpretações

### Fórmula central (scaled dot-product)

$$\text{Att}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right) V$$

- Máscara causal/atencional conforme a tarefa
- Complexidade  $\mathcal{O}(n^2)$  em tempo/memória



# Interpretação Matemática: Self-Attention

## 5 Embeddings & Interpretações

### Fórmula central (scaled dot-product)

$$\text{Att}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right) V$$

- Máscara causal/atencional conforme a tarefa
- Complexidade  $\mathcal{O}(n^2)$  em tempo/memória
- Gradientes e saturação da softmax

# Positional Encodings

## 5 Embeddings & Interpretações

- Absolutos senoidais vs. aprendidos

# Positional Encodings

## 5 Embeddings & Interpretações

- Absolutos senoidais vs. aprendidos
- Relativos e RoPE (rotary): melhor generalização/composição

# Positional Encodings

## 5 Embeddings & Interpretações

- Absolutos senoidais vs. aprendidos
- Relativos e RoPE (rotary): melhor generalização/composição
- Impacto em extrapolação e contextos longos

# Transformers

5 Embeddings & Interpretações

# Table of Contents

## 6 Arquiteturas & Variantes

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ **Arquiteturas & Variantes**
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

# Arquiteturas Transformer

## 6 Arquiteturas & Variantes

- **Encoder-Decoder, Encoder-only, Decoder-only**

# Arquiteturas Transformer

## 6 Arquiteturas & Variantes

- **Encoder-Decoder, Encoder-only, Decoder-only**
- Bloco: MHSA + FFN, *residual* + *layer norm*



# Arquiteturas Transformer

## 6 Arquiteturas & Variantes

- **Encoder-Decoder, Encoder-only, Decoder-only**
- Bloco: MHSA + FFN, *residual* + *layer norm*
- Pré-norm vs. pós-norm (estabilidade de treino)

## Atenções Eficientes

### 6 Arquiteturas & Variantes

- **Esparsidade:** Longformer/BigBird (padrões locais+globais)

#### Trade-off

Complexidade  $\mathcal{O}(n^2) \rightarrow \tilde{\mathcal{O}}(n)$  com perda controlada de exatidão e/ou restrições de padrão.

# Atenções Eficientes

## 6 Arquiteturas & Variantes

- **Esparsidade:** Longformer/BigBird (padrões locais+globais)
- **Aproximação:** Linformer, Nyströmformer

### Trade-off

Complexidade  $\mathcal{O}(n^2) \rightarrow \tilde{\mathcal{O}}(n)$  com perda controlada de exatidão e/ou restrições de padrão.

## Atenções Eficientes

### 6 Arquiteturas & Variantes

- **Esparsidade:** Longformer/BigBird (padrões locais+globais)
- **Aproximação:** Linformer, Nyströmformer
- **Kernels:** Performer (favor+), FlashAttention (IO-aware)

#### Trade-off

Complexidade  $\mathcal{O}(n^2) \rightarrow \tilde{\mathcal{O}}(n)$  com perda controlada de exatidão e/ou restrições de padrão.

## Atenções Eficientes

### 6 Arquiteturas & Variantes

- **Esparsidade:** Longformer/BigBird (padrões locais+globais)
- **Aproximação:** Linformer, Nyströmformer
- **Kernels:** Performer (favor+), FlashAttention (IO-aware)
- **Outros:** Reformer (LSH)

#### Trade-off

Complexidade  $\mathcal{O}(n^2) \rightarrow \tilde{\mathcal{O}}(n)$  com perda controlada de exatidão e/ou restrições de padrão.

# Table of Contents

## 7 Treino & Hiperparâmetros

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ **Treino & Hiperparâmetros**
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

# Práticas de Treino

## 7 Treino & Hiperparâmetros

- AdamW, warmup + decaimento; label smoothing quando aplicável

# Práticas de Treino

## 7 Treino & Hiperparâmetros

- AdamW, warmup + decaimento; label smoothing quando aplicável
- Regularização: dropout, stochastic depth, weight decay



# Práticas de Treino

## 7 Treino & Hiperparâmetros

- AdamW, warmup + decaimento; label smoothing quando aplicável
- Regularização: dropout, stochastic depth, weight decay
- AMP/mixed precision, grad clipping, checkpointing

# Práticas de Treino

## 7 Treino & Hiperparâmetros

- AdamW, warmup + decaimento; label smoothing quando aplicável
- Regularização: dropout, stochastic depth, weight decay
- AMP/mixed precision, grad clipping, checkpointing
- Dados: curriculum, masking, augmentation (TS/ViT/GAT)

# Hiperparâmetros Essenciais

## 7 Treino & Hiperparâmetros

- Profundidade,  $d_{\text{model}}$ , #heads,  $d_{ff}$ , dropout

# Hiperparâmetros Essenciais

## 7 Treino & Hiperparâmetros

- Profundidade,  $d_{\text{model}}$ , #heads,  $d_{ff}$ , dropout
- Comprimento de contexto, batch size, *LR schedule*

# Hiperparâmetros Essenciais

## 7 Treino & Hiperparâmetros

- Profundidade,  $d_{\text{model}}$ , #heads,  $d_{ff}$ , dropout
- Comprimento de contexto, batch size, *LR schedule*
- Específicos: tokenização (LM), *patch size* (ViT), janela/patch (TS)

# Qual tamanho ideal? (Scaling)

7 Treino & Hiperparâmetros

- Leis de escala vs. limite de dados/compute

# Qual tamanho ideal? (Scaling)

## 7 Treino & Hiperparâmetros

- Leis de escala vs. limite de dados/compute
- Tokens vs. parâmetros; saturação com contexto

## Qual tamanho ideal? (Scaling)

### 7 Treino & Hiperparâmetros

- Leis de escala vs. limite de dados/compute
- Tokens vs. parâmetros; saturação com contexto
- Regra prática: dimensione para o dataset e o *budget* de inferência



## Aplicações

7 Treino & Hiperparâmetros

# Table of Contents

## 8 Séries Temporais

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ **Séries Temporais**
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

# Atenção em Séries Temporais

## 8 Séries Temporais

- Codificação temporal: absolutos/relativos, time2vec, calendários/sazonalidade

# Atenção em Séries Temporais

## 8 Séries Temporais

- Codificação temporal: absolutos/relativos, time2vec, calendários/sazonalidade
- Exógenas e *cross-attention*; *patching* para contextos longos

# Atenção em Séries Temporais

## 8 Séries Temporais

- Codificação temporal: absolutos/relativos, time2vec, calendários/sazonalidade
- Exógenas e *cross-attention*; *patching* para contextos longos
- Tarefas: previsão, imputação, detecção de anomalias

# Table of Contents

## 9 Language Models

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ **Language Models**
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

# Language Models (LM)

## 9 Language Models

- Atenção **causal**, *next-token* e *masking*

# Language Models (LM)

## 9 Language Models

- Atenção **causal**, *next-token* e *masking*
- Pré-treino vs. *fine-tuning*; Instrução/LoRA/Adapters



# Language Models (LM)

## 9 Language Models

- Atenção **causal**, *next-token* e *masking*
- Pré-treino vs. *fine-tuning*; Instrução/LoRA/Adapters
- Métricas: perplexidade e tarefas *downstream*

# Table of Contents

## 10 Vision Transformer

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ **Vision Transformer**
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

## 42/63 Backup (Opcional)

- ▶ Definição Bibliográfica

# Vision Transformer (ViT)

10 Vision Transformer

- Imagem  $\rightarrow$  *patches* + [CLS] token

# Vision Transformer (ViT)

10 Vision Transformer

- Imagem  $\rightarrow$  *patches* + [CLS] token
- Posicionais 2D; augmentations (RandAug, Mixup/CutMix)

# Vision Transformer (ViT)

10 Vision Transformer

- Imagem  $\rightarrow$  *patches* + [CLS] token
- Posicionais 2D; augmentations (RandAug, Mixup/CutMix)
- Transfer: *linear probe* vs. *fine-tune*

# Table of Contents

## 11 Graph Attention

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ **Graph Attention**
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

# Graph Attention Networks (GAT)

## 11 Graph Attention

### Coeficientes de Atenção (um cabeçalho)

$$\alpha_{ij} = \text{softmax}_j(\text{LeakyReLU}(a^\top [Wh_i || Wh_j]))$$

- Multi-head; sobre-*smoothing* e escalabilidade

# Graph Attention Networks (GAT)

## 11 Graph Attention

### Coeficientes de Atenção (um cabeçalho)

$$\alpha_{ij} = \text{softmax}_j(\text{LeakyReLU}(a^\top [Wh_i || Wh_j]))$$

- Multi-head; sobre-*smoothing* e escalabilidade
- Heterógrafos e atenção relacional



# Escala & Compressão

11 Graph Attention

# Table of Contents

12 MoE, Pruning, Distillation

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ **MoE, Pruning, Distillation**
- ▶ Diagnóstico & Limites
- ▶ Conclusões

# Mixture of Experts (MoE)

12 MoE, Pruning, Distillation

- *Gating* top-1/top-2; balanceamento de carga

## Mixture of Experts (MoE)

12 MoE, Pruning, Distillation

- *Gating* top-1/top-2; balanceamento de carga
- Roteamento esparsos: capacidade vs. ociosidade

## Mixture of Experts (MoE)

12 MoE, Pruning, Distillation

- *Gating* top-1/top-2; balanceamento de carga
- Roteamento esparsos: capacidade vs. ociosidade
- Custos de comunicação e estabilidade no treino

# Pruning

12 MoE, Pruning, Distillation

- Não estruturado (magnitude) vs. estruturado (canal/bloco, n:m)

# Pruning

12 MoE, Pruning, Distillation

- Não estruturado (magnitude) vs. estruturado (canal/bloco, n:m)
- Iterativo vs. *one-shot*; impacto em latência real

# Pruning

12 MoE, Pruning, Distillation

- Não estruturado (magnitude) vs. estruturado (canal/bloco, n:m)
- Iterativo vs. *one-shot*; impacto em latência real
- Interação com quantização e *sparsity-aware kernels*



## Perda típica (Hinton)

$$\mathcal{L} = (1 - \lambda) \text{CE}(y, s) + \lambda T^2 \text{KL}(p_T \parallel q_T)$$

- *Teacher*  $\rightarrow$  *student*; temperatura  $T$

# Distillation

12 MoE, Pruning, Distillation

## Perda típica (Hinton)

$$\mathcal{L} = (1 - \lambda) \text{CE}(y, s) + \lambda T^2 \text{KL}(p_T \parallel q_T)$$

- *Teacher*  $\rightarrow$  *student*; temperatura  $T$
- *Intermediate layer hints*; *task-specific* vs. *generalista*

# Distillation

12 MoE, Pruning, Distillation

## Perda típica (Hinton)

$$\mathcal{L} = (1 - \lambda) \text{CE}(y, s) + \lambda T^2 \text{KL}(p_T \parallel q_T)$$

- *Teacher*  $\rightarrow$  *student*; temperatura  $T$
- *Intermediate layer hints*; *task-specific* vs. *generalista*
- Benefícios: latência/energia e *edge deployment*

## Interpretação & Limitações

12 MoE, Pruning, Distillation

# Table of Contents

## 13 Diagnóstico & Limites

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ **Diagnóstico & Limites**
- ▶ Conclusões

# Interpretação & Diagnóstico

## 13 Diagnóstico & Limites

- *Attention rollout/flow; atenção  $\neq$  causalidade*

# Interpretação & Diagnóstico

## 13 Diagnóstico & Limites

- *Attention rollout/flow*; atenção  $\neq$  causalidade
- *Probing* de camadas; ablação de cabeças

# Interpretação & Diagnóstico

## 13 Diagnóstico & Limites

- *Attention rollout/flow*; atenção  $\neq$  causalidade
- *Probing* de camadas; ablação de cabeças
- Ferramentas de *explainability* por domínio



# Limitações & Trade-offs

## 13 Diagnóstico & Limites

- Custo  $\mathcal{O}(n^2)$ , viés de dados, OOD robustness

# Limitações & Trade-offs

## 13 Diagnóstico & Limites

- Custo  $\mathcal{O}(n^2)$ , viés de dados, OOD robustness
- Contaminação de treino e privacidade

# Limitações & Trade-offs

## 13 Diagnóstico & Limites

- Custo  $\mathcal{O}(n^2)$ , viés de dados, OOD robustness
- Contaminação de treino e privacidade
- Energia/carbono e restrições de hardware

# Table of Contents

## 14 Conclusões

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ **Conclusões**

# Conclusões

14 Conclusões

- Atenção como princípio unificador em sequência, visão e grafos

# Conclusões

## 14 Conclusões

- Atenção como princípio unificador em sequência, visão e grafos
- Escolha guiada por dados, métricas e orçamento (treino/inferência)

# Conclusões

## 14 Conclusões

- Atenção como princípio unificador em sequência, visão e grafos
- Escolha guiada por dados, métricas e orçamento (treino/inferência)
- Escala e compressão (MoE/pruning/distillation) para produção

# Attention Models: da motivação às variantes modernas

*Obrigado pela Atenção!*

*Alguma Pergunta?*

*Natanael Moura Junior*

*natmourajr@poli.ufrj.br, natmourajr@lps.ufrj.br*



## Perguntas?

14 Conclusões

Obrigado!

# Table of Contents

## 15 Backup (Opcional)

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

## ▶ Backup (Opcional)

# Equações úteis (resumo)

15 Backup (Opcional)

- Scaled dot-product, aditiva (Bahdanau)
- Atenção relativa e RoPE
- GAT detalhado; máscaras causais

# Hiperparâmetros por tarefa (resumo)

15 Backup (Opcional)

- LM: contexto, BPE, heads/profundidade típicos
- ViT: patch size, MLP ratio, augmentations
- TS: janela, patching, covariáveis, perdas (MAE/MSE/Quantile)

# Table of Contents

## 16 Referências Bibliográficas

- ▶ Motivação e Histórico
- ▶ Roteiro
- ▶ Motivação
- ▶ Modelos de Sequência
- ▶ Embeddings & Interpretações
- ▶ Arquiteturas & Variantes
- ▶ Treino & Hiperparâmetros
- ▶ Séries Temporais
- ▶ Language Models
- ▶ Vision Transformer
- ▶ Graph Attention
- ▶ MoE, Pruning, Distillation
- ▶ Diagnóstico & Limites
- ▶ Conclusões

# Referências Bibliográficas

16 Referências Bibliográficas