Attention Models: da motivação às variantes modernas

Intuição geométrica, formulação matemática e aplicações

Seu Nome

17 de agosto de 2025

Table of Contents

Redes Neurais Recorrentes

1 Motivação e Histórico

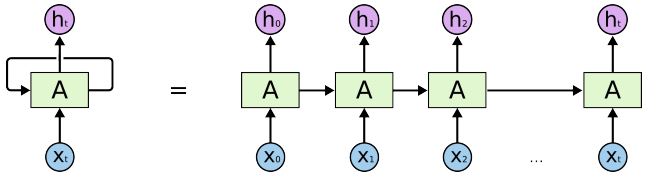


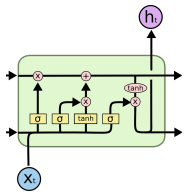
Figura: Redes Neurais Recorrentes [1]

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

[1] Figura de Christopher Olah

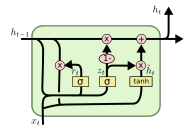
Tipos de camadas RNN

LSTM



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

GRU



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

LSTM como baseline para NMT

- Encoder LSTM lê $(x_1, \ldots, x_n)$ e produz estados h_t; o contexto é o último estado $c = h_n$.
- Decoder LSTM gera $(y_1, \ldots, y_m)$ 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 [1]

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

- 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).
- α_{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

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

$$h_j = [\overrightarrow{h_j}; \overleftarrow{h_j}]$$

  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 [2]

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: Podemos reinterpretar em termos modernos como $Q = s_j, K = h_i, V = h_i$.

Global vs Local Attention [2]

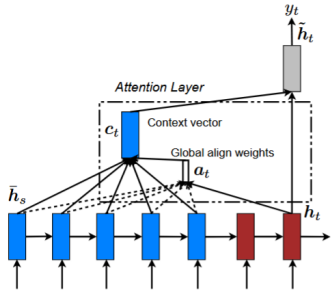


Figure 2: **Global attentional model** – at each time step t, the model infers a *variable-length* alignment weight vector $\boldsymbol{a}_t$ based on the current target state $\boldsymbol{h}_t$ and all source states $\bar{\boldsymbol{h}}_s$. A global context vector $\boldsymbol{c}_t$ is then computed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

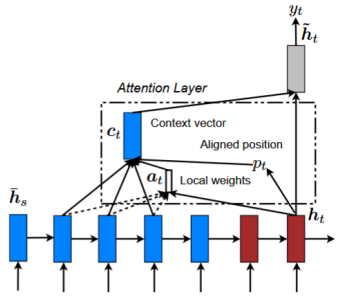


Figure 3: **Local attention model** – the model first predicts a single aligned position p_t for the current target word. A window centered around the source position p_t is then used to compute a context vector $\boldsymbol{c}_t$, a weighted average of the source hidden states in the window. The weights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

Self-Attention: dependências em paralelo

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

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

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

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

Arquitetura pré-transformer

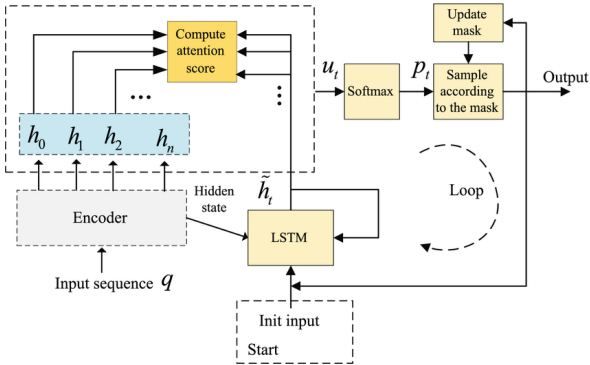


Figura: Exemplo de arquitetura pré-transformer (aditiva) [3]

Table of Contents

Attention Is All You Need [4]: nascendo o Transformer

2 Transformers

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

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

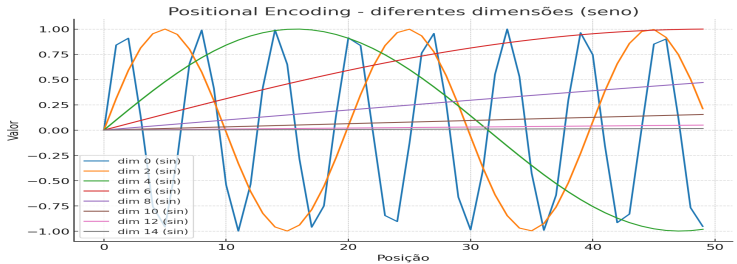


Figura: Exemplo de codificação posicional senoidal [4]

Bloco Transformer e arquitetura

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

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

- Empilha-se vários blocos de atenção+FFN $\Rightarrow$ **arquitetura Transformer**.
- **Máscara causal** (para LMs) impede olhar o futuro:

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

Arquiteturas Transformer e Aplicações

Encoder–Decoder
(Transformer original, 2017)

- Tradução automática
- Sumarização
- Diálogo
- Captioning

Encoder-only
(BERT, RoBERTa, DistilBERT)

- Classificação de texto
- NER (entidades)
- QA (extração de trechos)
- Análise semântica

Decoder-only
(GPT, LLaMA, etc.)

- Modelos de linguagem
- Geração de texto
- Completamento de prompts
- Story generation

Arquiteturas Transformer: encoder e decoder [4]

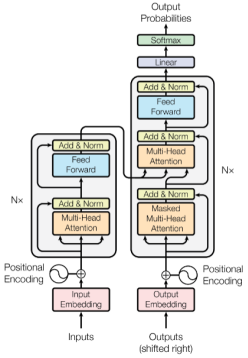


Figura: Transformer Encoder e Decoder

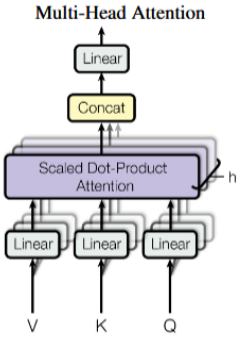


Figura: Multi-Head Attention

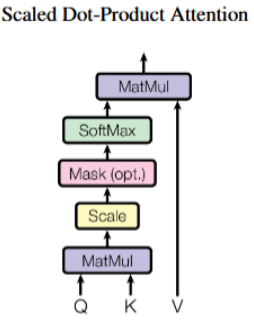


Figura: Multi-Head Attention

Resumo da evolução dos modelos

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.

Fundamentos

2 Transformers

Table of Contents

RNN/GRU/LSTM — limites práticos

- Cálculo sequencial $\Rightarrow$ baixa paralelização

RNN/GRU/LSTM — limites práticos

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

RNN/GRU/LSTM — limites práticos

- 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

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

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

LSTM + Attention (Encoder–Decoder)

Atenção aditiva (Bahdanau)

Equações

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

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

Table of Contents

Word/Subword Embeddings

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

Word/Subword Embeddings

4 Embeddings & Interpretações

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

Word/Subword Embeddings

4 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

- Q, K, V são projeções lineares do latente

Interpretação Geométrica da Atenção

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

Interpretação Geométrica da Atenção

- 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

Fórmula central (scaled dot-product)

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

- Máscara causal/atencional conforme a tarefa

Interpretação Matemática: Self-Attention

Fórmula central (scaled dot-product)

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

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

Interpretação Matemática: Self-Attention

Fórmula central (scaled dot-product)

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

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

Positional Encodings

4 Embeddings & Interpretações

- Absolutos senoidais vs. aprendidos

Positional Encodings

4 Embeddings & Interpretações

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

Positional Encodings

4 Embeddings & Interpretações

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

Transformers

4 Embeddings & Interpretações

Table of Contents

Arquiteturas Transformer

5 Arquiteturas & Variantes

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

Arquiteturas Transformer

5 Arquiteturas & Variantes

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

Arquiteturas Transformer

5 Arquiteturas & Variantes

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

Atenções Eficientes

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

Trade-off

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

Atenções Eficientes

5 Arquiteturas & Variantes

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

Trade-off

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

Atenções Eficientes

5 Arquiteturas & Variantes

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

Trade-off

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

Atenções Eficientes

5 Arquiteturas & Variantes

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

Trade-off

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

Table of Contents

Práticas de Treino

6 Treino & Hiperparâmetros

- AdamW, warmup + decaimento; label smoothing quando aplicável

Práticas de Treino

6 Treino & Hiperparâmetros

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

Práticas de Treino

6 Treino & Hiperparâmetros

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

Práticas de Treino

6 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

- Profundidade, d_{model}, #heads, d_{ff}, dropout

Hiperparâmetros Essenciais

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

Hiperparâmetros Essenciais

- Profundidade, d_{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)

- Leis de escala vs. limite de dados/compute

Qual tamanho ideal? (Scaling)

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

Qual tamanho ideal? (Scaling)

- 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

6 Treino & Hiperparâmetros

Table of Contents

Atenção em Séries Temporais

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

Atenção em Séries Temporais

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

Atenção em Séries Temporais

7 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

Language Models (LM)

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

Language Models (LM)

8 Language Models

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

Language Models (LM)

8 Language Models

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

Table of Contents

Vision Transformer (ViT)

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

Vision Transformer (ViT)

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

Vision Transformer (ViT)

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

Table of Contents

Graph Attention Networks (GAT)

Coeficientes de Atenção (um cabeçalho)

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

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

Graph Attention Networks (GAT)

Coeficientes de Atenção (um cabeçalho)

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

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

Escala & Compressão

10 Graph Attention

COPPE 60 anos UFRJ

Table of Contents

Mixture of Experts (MoE)

11 MoE, Pruning, Distillation

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

Mixture of Experts (MoE)

11 MoE, Pruning, Distillation

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

Mixture of Experts (MoE)

11 MoE, Pruning, Distillation

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

Pruning

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

Pruning

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

Pruning

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

Distillation

Perda típica (Hinton)

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

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

Distillation

Perda típica (Hinton)

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

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

Distillation

Perda típica (Hinton)

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

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

Interpretação & Limitações

11 MoE, Pruning, Distillation

Table of Contents

Interpretação & Diagnóstico

12 Diagnóstico & Limites

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

Interpretação & Diagnóstico

12 Diagnóstico & Limites

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

Interpretação & Diagnóstico

12 Diagnóstico & Limites

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

Limitações & Trade-offs

12 Diagnóstico & Limites

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

Limitações & Trade-offs

12 Diagnóstico & Limites

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

Limitações & Trade-offs

12 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

13 Conclusões

Conclusões

13 Conclusões

- Atenção como princípio unificador em sequência, visão e grafos

Conclusões

13 Conclusões

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

Conclusões

13 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

COPPE 60 anos UFRJ

Attention Models: da motivação às variantes modernas

Obrigado pela Atenção!
Alguma Pergunta?

Natanael Moura Junior

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

Perguntas?

13 Conclusões

Obrigado!

Table of Contents

Equações úteis (resumo)

14 Backup (Opcional)

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

Hiperparâmetros por tarefa (resumo)

14 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

Referências Bibliográficas

15 Referências Bibliográficas

[1] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[2] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1412–1421, 2015.

[3] S. Gu and Y. Zhuang, “Method for solving constrained 0-1 quadratic programming problems based on pointer network and reinforcement learning,” *Neural Computing and Applications*, vol. 35, 2022.

Referências Bibliográficas

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.