Stat 9911
Principles of AI: LLMs
Large Language Model Architectures 03

Edgar Dobriban

Department of Statistics and Data Science, the Wharton School, University of Pennsylvania

February 4, 2025

Plan

- We continue with the details of transformer architectures.

Table of Contents

Order and Permutation Equivariance

- Order matters! "The cat chased the mouse." vs "The mouse chased the cat."
- A key consideration in language models is how to take position information into account.
- Claim: Full attention is *permutation equivariant*. If we permute the order of the input E, the order of the output $\hat{E}$ is permuted accordingly.
- Now consider the reps e_c and e_m of cat and mouse in "The cat chased the mouse."
  1. Rep should capture that e_c is the attacker and e_m is the victim.
  2. However, if I permute the sentence to "The mouse chased the cat.", the reps e_c and e_m of cat and mouse stay exactly the same! (due to perm. equiv.)
  3. So, e_c and e_m cannot capture the relation between cat/mouse.

Attention is Permutation Equivariant

- Permute input embeddings $e_1, \ldots, e_T$ by $T \times T$ permutation matrix Π (i.e., $E \to \Pi E$)
- Recalling $Q = EW'_q, K = EW'_k, V = EW'_v$, $Z = QK^\top/\sqrt{d'}$, this leads to:

$$Q \to \Pi Q, \quad K \to \Pi K, \quad V \to \Pi V, \quad Z \to \Pi Z \Pi^\top.$$

- Row-softmax composed with Exp preserves permutation equiv.:
  - Elementwise exponentiation: $\exp(\Pi Z \Pi^\top) = \Pi \exp(Z) \Pi^\top$.
  - Division by row-sums: Let $g(Z) = \mathrm{diag}(Z1)^{-1} Z$. Now:

    $$\mathrm{diag}(\Pi Z \Pi^\top 1) = \mathrm{diag}(\Pi Z 1) = \Pi \,\mathrm{diag}(Z1)\, \Pi^\top$$

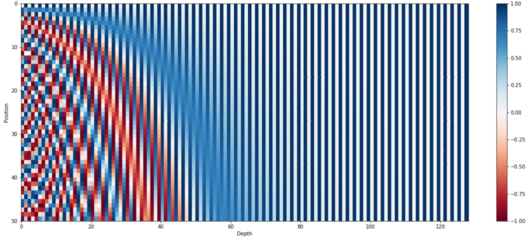
    Hence,

    $$\mathrm{diag}(\Pi Z \Pi^\top 1)^{-1} = \Pi \,\mathrm{diag}(Z1)^{-1}\, \Pi^\top$$

    Thus: $g(\Pi Z \Pi^\top) = \Pi \,\mathrm{diag}(Z1)^{-1}\, \Pi^\top \cdot \Pi\, Z\, \Pi^\top = \Pi\, g(Z)\, \Pi^\top$.
- Therefore $\hat{E} = AV \mapsto \Pi A \Pi^\top \Pi V = \Pi \hat{E}$. Full attention is permutation equivariant.

Adding Positional Information

- Causal attn is not permutation equivariant, and specific output embeddings are in general not perm. invariant (despite some claims).
- However, explicitly knowing about position can still help: Regardless of what input is, you should always pay more attention to last few tokens
- Standard self-attention does not have this built in and needs to learn it, as it is equally "eager to pay attention" to any previous token.
- To address this, early works adds a positional encoding matrix Γ (a vector for each location), to the embedding matrix E.
- Γ can be:
  - Fixed (e.g., Vaswani et al. (2017))
  - Learned (e.g., Radford et al. (2018, 2019); Brown et al. (2020))



Issues with Absolute Positional Encoding

- Absolute positions may not be as meaningful as relative positions (e.g., just-preceding tokens are influential).
- Designing embeddings that depend only on relative position aims to address this.

RoPE: Rotary Positional Encoding (Su et al., 2024)

- For context length T, define key and query functions:

$$k, q : \mathbb{R}^d \times [T] \to \mathbb{C}^{d'}$$

such that for some $g : \mathbb{R}^d \times \mathbb{R}^d \times [T] \to \mathbb{R}$:

$$\langle k(e_i, i), q(e_j, j) \rangle_{\mathbb{C}} = g(e_i, e_j, i - j).$$

Here $\langle a, b \rangle_{\mathbb{C}} = \sum_{l=1}^{d} a_l \bar{b}_l$, where for a complex number $z = x + \mathrm{i}y$, $\bar{z} = x - \mathrm{i}y$ denotes its complex conjugate

- We want to find k, q such that the above holds for any $e = e_i, e' = e_j \in \mathbb{R}^d$ and $i, j = 1, \ldots,$
- Complex numbers are used for convenience and we will eventually use reals. Need real-valued result for softmax probability.

RoPE: Rotary Positional Encoding (Su et al., 2024)

- A particular solution is of the form (Su et al., 2024):

$$k(e, j) = q(e, j) = q(e)e^{2\pi \mathrm{i}\theta \cdot j}.$$

- Interpret θ as frequency: $j - i = 1/\theta$ is period of attention
- Frequency can be coordinate-specific, and Su et al. (2024) suggest using $\theta_t = F^{-2\lfloor (t-1)/2 \rfloor / d}$, for coords $t = 1, \ldots, d$, for some F
- Llama 3 uses $F = 500,000$ (Dubey et al., 2024); see also here

Solving for Relative Positional Embedding

- Start with a simpler problem, where e_i, e_j are fixed and $k = q$, so we have
$$\langle k(i), k(j) \rangle_{\mathbb{C}} = g(i - j).$$
Allow T to be arbitrarily large, and focus on one coordinate at a time.
- Then, we need to solve the following problem: Find nonzero complex-valued sequences $(a_n)_{n \geq 1}$ such that there exists a complex-valued sequence $(b_n)_{n \in \mathbb{Z}}$ for which:
$$a_m \overline{a_n} = b_{m-n}, \quad \text{for all } m, n \geq 1.$$
- Write $a_m = r_m e^{\mathrm{i}\nu_m}$ and $b_n = s_n e^{\mathrm{i}\eta_n}$ in polar form; $r_m, s_n > 0$ are uniquely determined.
- Using this, deduce the conditions:
$$\begin{cases} r_m r_n = s_{m-n} \\ \nu_m - \nu_n - \eta_{m-n} \in \mathbb{Z} \end{cases}$$
for all m, n.

Solution: Magnitudes and Phases

- From $r_m r_n = s_{m-n}$:
  - Take $m = n$: $r_m^2 = s_0$, for all m.
  - Therefore, $r_m = s_0^{1/2}$ does not depend on m.
- From $\nu_m - \nu_n - \eta_{m-n} \in \mathbb{Z}$:
  - Take $m = n + 1$: $\nu_{n+1} - \nu_n - \eta_1 \in \mathbb{Z}$.
  - Therefore, $\nu_{n+1} - \nu_1 - n\eta_1 \in \mathbb{Z}$.
  - Fractional part of ν_n determines the solution.
  - Without loss of generality, take $\nu_{n+1} = \nu_1 + n\eta_1$.
- Solution
$$a_m = s_0^{1/2} e^{2\pi \mathrm{i}(\nu_1 + m\eta_1)}$$
- Equivalently, $a_m = Ce^{2\pi \mathrm{i}\, m\theta}$, where $C \in \mathbb{C}$ and $\theta \in \mathbb{R}$; i.e., $k(j) = Ce^{2\pi \mathrm{i}\, \theta \cdot j}$
- Vectors: each coord. as above, $k(j) = (C_1 e^{2\pi \mathrm{i}\, \theta_1 \cdot j}, C_2 e^{2\pi \mathrm{i}\, \theta_2 \cdot j}, \ldots)$. Then,
$$\begin{aligned}\langle k(i), k(j)\rangle_{\mathbb{C}} &= \sum_m \left(C_m e^{2\pi \mathrm{i}\, \theta_m i}\right)\left(\overline{C_m} e^{-2\pi \mathrm{i}\, \theta_m j}\right) \\ &= \sum_m |C_m|^2 e^{2\pi \mathrm{i}\, \theta_m (i-j)}.\end{aligned}$$

Real-valued Solution

- How to obtain real-valued solutions?
- Observe that

$$\overline{\langle k(i), k(j) \rangle_{\mathbb{C}}} = \sum_m |C_m|^2 e^{-2\pi \mathrm{i}\, \theta_m (i-j)}.$$

- So, if $k(j) = (e^{2\pi \mathrm{i}\, \theta \cdot j}, e^{-2\pi \mathrm{i}\, \theta \cdot j})$, then

$$\begin{aligned} \langle k(i), k(j) \rangle_{\mathbb{C}} &= e^{2\pi \mathrm{i}\, \theta (i-j)} + e^{-2\pi \mathrm{i}\, \theta (i-j)} \\ &= 2\mathrm{Re}(e^{2\pi \mathrm{i}\, \theta (i-j)}) = 2 \cos(2\pi\, \theta (i-j)) \end{aligned}$$

Final Solution

- Return to the original problem, $\langle k(e,i), q(e',j)\rangle_{\mathbb{C}} = g(e,e',i-j)$.
- Any functions of the form

$$k(e,i) = k(e)e^{2\pi \mathrm{i}\, \theta\cdot i},\ q(e',j) = q(e')e^{2\pi \mathrm{i}\, \theta\cdot j}$$

  are solutions with $g(e,e',i-j) = \langle k(e), q(e')\rangle_{\mathbb{C}} e^{2\pi \mathrm{i}\, \theta\cdot(i-j)}$. [could even have coordinate-specific phases]
- If we choose the coordinates of k, q to come in conjugate pairs, obtain $g(e,e',i-j) = 2\mathrm{Re}[\langle k(e), q(e')\rangle_{\mathbb{C}} e^{2\pi \mathrm{i}\, \theta\cdot(i-j)}]$
- Implement it:
  - For any embeddings e_i, e_j and associated key and query k_i, q_j, also include their conjugates, and encode position via $(k_i e^{2\pi \mathrm{i}\, \theta\cdot i}, \bar{k}_i e^{-2\pi \mathrm{i}\, \theta\cdot i})/\sqrt{2}$, $(q_j e^{2\pi \mathrm{i}\, \theta\cdot j}, \bar{q}_j e^{-2\pi \mathrm{i}\, \theta\cdot j})/\sqrt{2}$
  - Or $\mathrm{Re}[\langle k_i, q_j\rangle_{\mathbb{C}} e^{2\pi \mathrm{i}\, \theta\cdot(i-j)}]$ in the attn map

Final Solution in Real-Valued Form

- Let $k_i = k_{i1} + \mathrm{i} k_{i2}, \quad q_j = q_{j1} + \mathrm{i} q_{j2}$.
- Rotation matrix

$$R_a = R(\theta, a) = \begin{bmatrix} \cos(2\pi\theta a) & -\sin(2\pi\theta a) \\ \sin(2\pi\theta a) & \cos(2\pi\theta a) \end{bmatrix}.$$

- Attention inner product is:

$$\left\langle R(\theta, i) \begin{bmatrix} k_{i1} \\ k_{i2} \end{bmatrix}, R(\theta, j) \begin{bmatrix} q_{j1} \\ q_{j2} \end{bmatrix} \right\rangle_{\mathbb{R}},$$

where $\langle \cdot, \cdot \rangle_{\mathbb{R}}$ is the usual inner product on $\mathbb{R}$.

Rotary position embedding

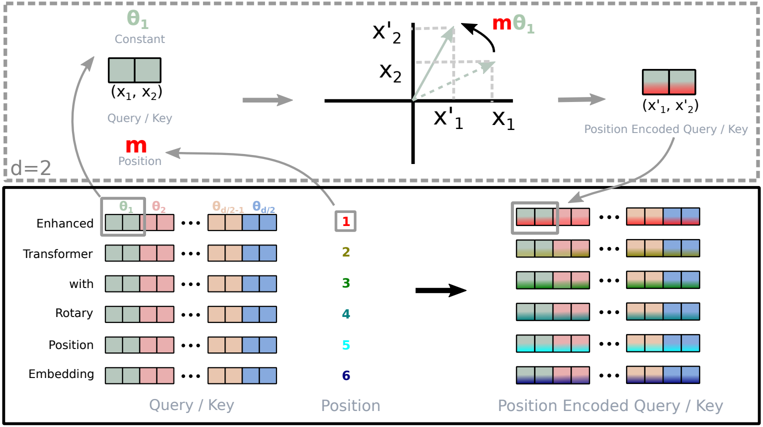


Figure: Su et al. (2024)

Generalization of rotary position embeddings

- We are using the structure of the translation group acting on the domain of sequences.
- We end up with group representations $e \mapsto R(\theta, a)e$
- Generalization to arbitrary domains and groups acting on them has been attempted: 1, 2

Attention with Linear Biases (ALiBi)

- ALiBi (Press et al., 2022) reduces attention to past tokens, with a slope term m:

$$\mathrm{softmax}(q_j K^\top - m[j-1, \ldots, 2, 1, 0]).$$

- Uses $m = 2^{-8h/H}$, where H is the number of attention heads and $h \in [H]$ is the index of the head (so different heads have different effective scope).
- Do not use any positional embedding.
- Implement by adding the biases to head-specific mask matrices (mask size: $H \times T \times T$)
- Empirically, generalizes better to unseen sequence lengths.

Length Extrapolation for ALiBi

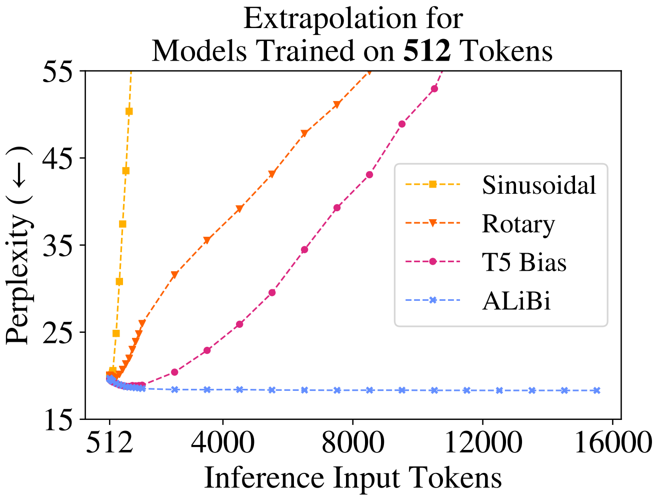


Figure: Press et al. (2022)

Aside: Perplexity

- Perplexity: for an LM $p : V^* \to [0,1]$, the perplexity of a string x is

$$q(x) = 2^{-\sum_{t=1}^{|x|} \log_2 p(x_t|x_{1:t-1})/|x|} = p(x)^{-1/|x|} \geq 1$$

  Also, $q(x) = 2^{\text{avg entropy of string } x \text{ under } p \text{ in bits/token}}$. Sometimes use $\log_2 q(x) = \log[1/p(x)]/|x|$
- A smaller perplexity means that the string is assigned a higher probability.
- Smaller is better: small perplexity on test data is considered to represent a higher quality LM.
- The perplexity of the LM on a dataset D is $Q = \mathbb{E}_{X\sim D} q(X)$

Table of Contents

Towards Longer Contexts

- The context length T is the number of tokens that can be input to the LM. Longer contexts mean it has the potential to handle larger tasks
- Key bottleneck: Standard attention has a quadratic $\Theta(T^2)$ memory and computational complexity
- Idea: simplify and reduce attention mechanism

Sparse Attention (Child et al., 2019)

- Only attend to a small number of tokens
  - e.g., previous c tokens
- Consider H heads, where the h-th head at the j-th position can attend to the subset $A_j^{(h)} \subset \{1, \ldots, j\}$. Can view connectivity pattern as a graph.
- Factorized attention: As we stack layers, we are able to attend to i at j if $k_1 \in A_j^{(h_1)}$, $k_2 \in A_{k_1}^{(h_2)}$, ... $i \in A_{k_{m-1}}^{(h_m)}$, for some $k_1, k_2, \ldots$ and $h_1, h_2, \ldots$

Sparse Attention (Child et al., 2019)

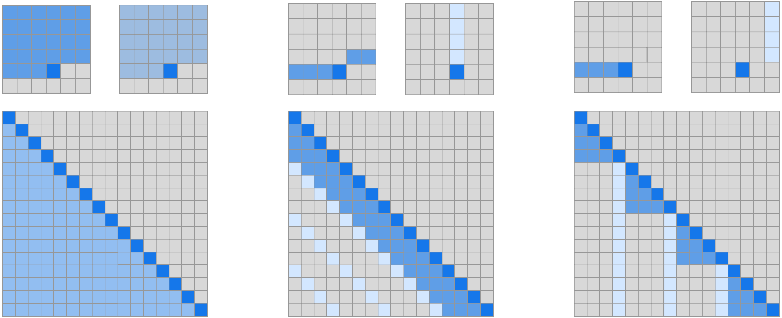


Figure: Full and factorized attention patterns from Child et al. (2019). "The top row indicates, for an example 6x6 image, which positions two attention heads receive as input when computing a given output. The bottom row shows the connectivity matrix (not to scale) between all such outputs (rows) and inputs (columns)."

- Something similar is used in GPT-3 (Brown et al., 2020)

Sparse Attention Example

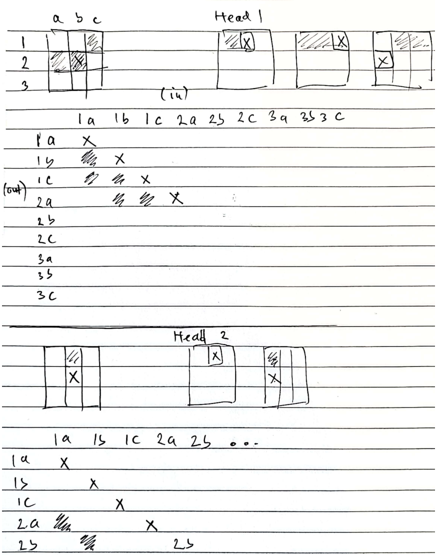
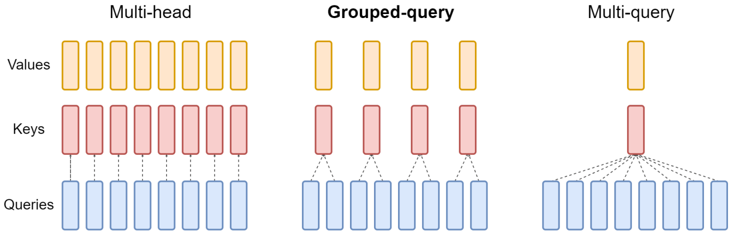


Figure: Full and factorized attention patterns from Child et al. (2019).

Multi- and Grouped Query Attention

- Multi-Query Attention (Shazeer, 2019): keys and values are shared across all attention heads.
- Grouped Query Attention (Ainslie et al., 2023) is a generalization that shares single key and value heads for groups query heads, interpolating between multi-head and multi-query attention.



- Used in Llama 3 (Dubey et al., 2024).

References

J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

O. Press, N. Smith, and M. Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training, 2018.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019.

N. Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

References

J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

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*, 2017. URL https://arxiv.org/pdf/1706.03762.pdf.