Stat 9911
Principles of AI: LLMs
Large Language Model Architectures 04
Specific LLMs

Edgar Dobriban

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

January 28, 2025

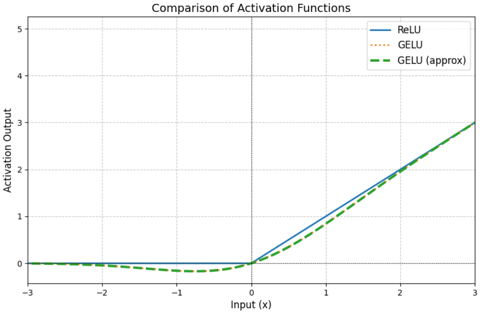
Plan

- We plan to discuss specific LLM families such as GPT, Llama, DeepSeek.

Table of Contents

GPT Series

- GPT series (Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2023)
- GPT-1: Gaussian Error Linear Unit (GELU) activation (Hendrycks and Gimpel, 2016): $x \mapsto x \cdot \Phi(x)$, where Φ is normal cdf, or approximate $x \mapsto 0.5x \left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right)\right)$ (Choudhury, 2014).
- GPT-2: Modified initialization: "We scale the weights of residual layers at initialization by a factor of $1/N^{1/2}$ where N is the number of residual layers."



GPT-3 (Brown et al., 2020) Model Details

- "Alternating dense and locally banded sparse attention patterns (similar to the Sparse Transformer (Child et al., 2019))"
- GPT-3 with 175B parameters
  - Context window: $T = 2048$ tokens
  - Layers: 96
  - Embedding rep: $d = 12288$
  - Feedforward rep: $d' = 4d$
  - Number of attention heads: $H = 96$, Dimension per head: $d/H = 128$

GPT-3 "on a Napkin"

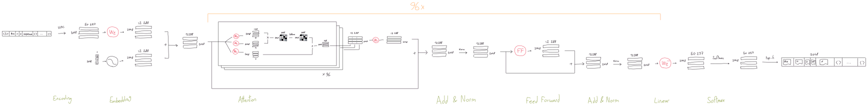


Figure: See source for a higher resolution.

Table of Contents

LLama Series

- LLama 1 (Touvron et al., 2023a):
  - RMSNorm pre-normalization (Zhang and Sennrich, 2019).
  - Activation: SwiGLU (Shazeer, 2020).
  - Rotary Position Embeddings (RoPE) (Su et al., 2024).
- LLama 2 (Touvron et al., 2023b):
  - Grouped-query attention (GQA) (Ainslie et al., 2023).
- LLama 3 (Dubey et al., 2024):
  - "Attention mask that prevents self-attention between different documents within the same sequence."
  - 405-B:
    - Context window: $T = 128K$ tokens
    - Layers: 126
    - Embedding rep: $d = 16,384$
    - Feedforward rep: $d' = 20,480$
    - Number of attention heads: $H = 128$. Key-value heads: 8

Table of Contents

DeepSeek-V2 (Liu et al., 2024)

- Multi-head Latent Attention
  - Map token emb e into an intermediate emb $e^{KV} = We$ of much lower dimension. Next, compute keys and values $k = W^K e^{KV}$, $v = W^V e^{KV}$ from this smaller dimensional rep.
  - Reduces size of the KV cache during inference, as only e^{KV} needs to be stored; leading to memory savings.
  - Weight decay can induce low-rank attention layers, see e.g., Kobayashi et al. (2024); so this architectural choice has some principled justification.
  - Same for the query, i.e., $e^Q = W^{Q'} e$, $q = W^Q e^Q$.
  - Compute MHA as usual.
  - Some linear maps become redundant, e.g., W^K and W^Q can be merged; also W^V and output projection W^O
- Decoupled Rotary Position Embedding (Bi et al., 2024)
  - Apply RoPE only to separate key-value projections

MLA + Decoupled RoPE

$$\mathbf{c}_t^Q = W^{DQ}\mathbf{h}_t, \tag{37}$$

$$[\mathbf{q}_{t,1}^C; \mathbf{q}_{t,2}^C; \ldots; \mathbf{q}_{t,n_h}^C] = \mathbf{q}_t^C = W^{UQ}\mathbf{c}_t^Q, \tag{38}$$

$$[\mathbf{q}_{t,1}^R; \mathbf{q}_{t,2}^R; \ldots; \mathbf{q}_{t,n_h}^R] = \mathbf{q}_t^R = \mathrm{RoPE}(W^{QR}\mathbf{c}_t^Q), \tag{39}$$

$$\mathbf{q}_{t,i} = [\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R], \tag{40}$$

$$\boxed{\mathbf{c}_t^{KV}} = W^{DKV}\mathbf{h}_t, \tag{41}$$

$$[\mathbf{k}_{t,1}^C; \mathbf{k}_{t,2}^C; \ldots; \mathbf{k}_{t,n_h}^C] = \mathbf{k}_t^C = W^{UK}\mathbf{c}_t^{KV}, \tag{42}$$

$$\boxed{\mathbf{k}_t^R} = \mathrm{RoPE}(W^{KR}\mathbf{h}_t), \tag{43}$$

$$\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_t^R], \tag{44}$$

$$[\mathbf{v}_{t,1}^C; \mathbf{v}_{t,2}^C; \ldots; \mathbf{v}_{t,n_h}^C] = \mathbf{v}_t^C = W^{UV}\mathbf{c}_t^{KV}, \tag{45}$$

$$\mathbf{o}_{t,i} = \sum_{j=1}^{t} \mathrm{Softmax}_j\left(\frac{\mathbf{q}_{t,i}^T\mathbf{k}_{j,i}}{\sqrt{d_h + d_h^R}}\right)\mathbf{v}_{j,i}^C, \tag{46}$$

$$\mathbf{u}_t = W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \ldots; \mathbf{o}_{t,n_h}], \tag{47}$$

where the boxed vectors in blue need to be cached for generation. During inference, the naive formula needs to recover $\mathbf{k}_t^C$ and $\mathbf{v}_t^C$ from $\mathbf{c}_t^{KV}$ for attention. Fortunately, due to the associative law of matrix multiplication, we can absorb W^{UK} into W^{UQ}, and W^{UV} into W^O. Therefore, we do not need to compute keys and values out for each query. Through this optimization, we avoid the computational overhead for recomputing $\mathbf{k}_t^C$ and $\mathbf{v}_t^C$ during inference.

Figure: Our notation: $h \to e$, $c^Q \to e^Q$, $c^{KV} \to e^{KV}$, $o \to \hat{v}$, $u \to e'$

DeepSeek-V3 MoE

- Mixture of Experts (MoE): Shared and routed experts for efficiency (Dai et al., 2024). Compute the FFN output h' as

$$h' = u + \sum_{i=1}^{N_s} \phi_i^{(s)}(u) + \sum_{i=1}^{N_r} g_i \phi_i^{(r)}(u),$$

$$g_i = \begin{cases} s_i, & s_i \in \mathrm{Topk}\left(\{s_j \mid 1 \leqslant j \leqslant N_r\}, K_r\right), \\ 0, & \text{otherwise}, \end{cases}$$

$$s_i' = \mathrm{Sigmoid}\left({e'}_t^{\top} f_i\right), \quad s_i = s_i' / (\sum_{j=1}^{N_r} s_j)$$

where

- N_s and N_r denote the numbers of shared and routed experts, resp;
- $\phi_i^{(s)}(\cdot)$ and $\phi_i^{(r)}(\cdot)$ denote the i-th shared routed experts, resp;
- K_r denotes the number of activated routed experts;
- g_i is the gate value for the i-th expert; s_i is token-to-expert affinity;
- f_i is the mean over tokens of the activations of the i-th routed expert.

DeepSeek-V3

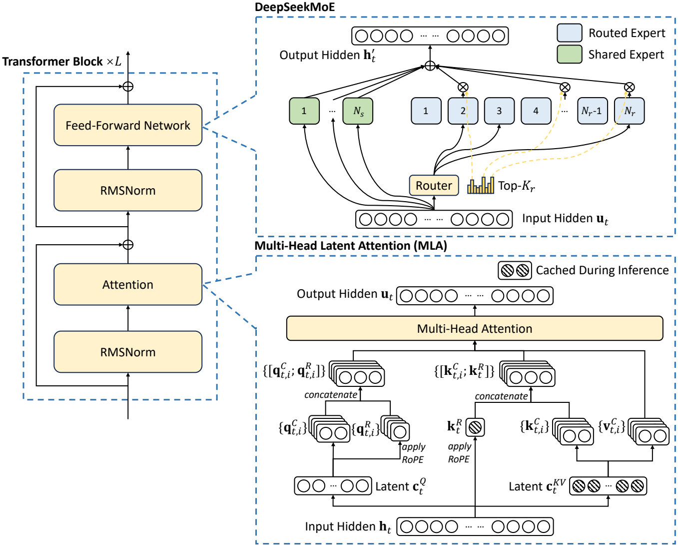


Figure: Our notation: $h \to e$, $c^Q \to e^Q$, $c^{KV} \to e^{KV}$, $o \to \hat{v}$, $u \to e'$

DeepSeek-V3 MoE

- Auxiliary-Loss-Free Load Balancing (Wang et al., 2024).
  - Add a constant c_i to the affinities when determining which experts to choose as in $s_i + c_i \in \operatorname{Topk}\left(\{s_j + c_j \mid 1 \leqslant j \leqslant N_r\}, K_r\right)$. These values are constant across tokens.
  - Update them heuristically during training to balance loads.

Loss-based balancing in MoE

- Loss-based balancing (Lepikhin et al., 2020; Fedus et al., 2022), with a small weight: For a sequence of length T, define auxiliary loss $\mathcal{L}_{\text{Balance}} = \alpha \sum_{i=1}^{N} f_i P_i$, where

$$f_i = \frac{N}{KT} \sum_{t=1}^{T} I(\text{Token } t \text{ selects Expert } i), \quad P_i = \frac{1}{T} \sum_{t=1}^{T} s_{i,t}.$$

  Here:
  - N is the total number of experts.
  - K is the number of experts selected for each token.
  - $s_{i,t}$ is the routing score of Expert i for Token t.
  - f_i represents the fraction of tokens routed to Expert i.
  - P_i denotes the average gating scores of Expert i.
  - α is a hyper-parameter controlling the strength of the auxiliary loss.

  This loss promotes balance, if f_i is correlated with P_i across tokens;
  - larger average scores (across tokens) for an expert correspond to larger selection frequencies (across tokens) of that specific expert.

DeepSeek-V3

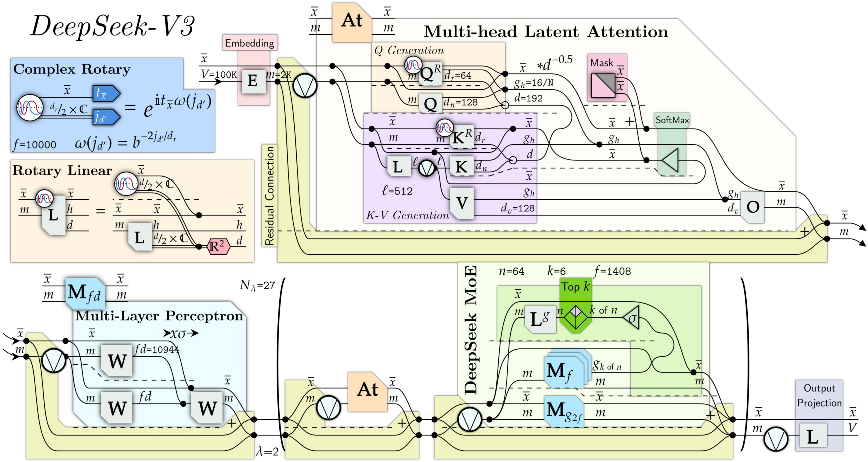


Figure: Via X (suspended account)

DeepSeek-V3

- Long context extension: YaRN (Peng et al., 2023).

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.

X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

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. Choudhury. A simple approximation to the area under standard normal curve. *Mathematics and Statistics*, 2(3):147–149, 2014.

D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

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.

References

W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

S. Kobayashi, Y. Akram, and J. Von Oswald. Weight decay induces low-rank attention layers. *arXiv preprint arXiv:2410.23819*, 2024.

D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

A. Liu, B. Feng, B. Wang, B. Wang, B. Liu, C. Zhao, C. Dengr, C. Ruan, D. Dai, D. Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

OpenAI. Gpt-4 technical report, 2023.

B. Peng, J. Quesnelle, H. Fan, and E. Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

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. Gelu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

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.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023a.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

L. Wang, H. Gao, C. Zhao, X. Sun, and D. Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024.

B. Zhang and R. Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.