A WALKTHROUGH OF NANOCHAT

Anonymous authors
Paper under double-blind review

ABSTRACT

This walkthrough of nanochat aims to study a complete, modern LLM implementation with cutting-edge techniques, all in a clean, accessible codebase. It is also the perfect fit to learn about the complete pipeline from tokenization and pretraining to finetuning and RL.

1 QUICKSTART

The Quickstart of nanochat demonstrates a depth-20 speedrun model.

It is a GPT-like Llama model with a context size of up to 65,536 characters, 1280 hidden dimensions, and it is a 20-layer Transformer with 10 heads in each layer.

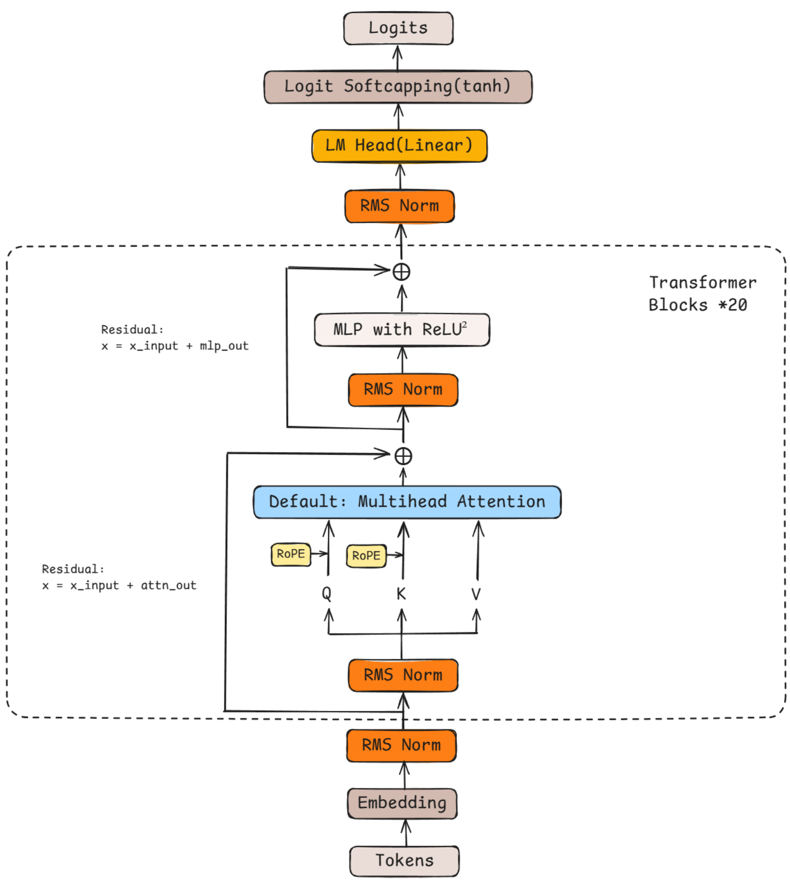

Figure 1: Architecture

1.1 SIZE

For this depth-20 model:

- $L =$ Layers $= 20$
- $D =$ Model Dimension(Hidden Dimension) $= 20 \times 64 = 1280$
- $V =$ Vocabulary size $= 65,536$

Universal Formula:

$$\begin{aligned}\text{Parameters} &= V \times D + L \times \big(4 \times D^2 + 2 \times (D \times (4 \times D))\big) + V \times D \\ &= V \times D + 20 \times \big(4 \times D^2 + 2 \times (D \times 4D)\big) + V \times D \\ &= 83,886,080 + 20 \times (6,553,600 + 13,107,200) + 83,886,080 \\ &= 560,988,160\end{aligned}$$

- **For embedding and unembedding:** each has $V \times D$ parameters
- **For each Transformer block:**
  - **Attention Layer:** 4 weight matrices(Q, K, V and ouput projection), each has D^2 parameters
  - **MLP Layer:** 2 weight matrices(4x expansion and projection), each has $D \times (4 \times D)$ parameters

1.2 ATTENTION HEADS

The d20 model has ceil(D/128) = 10 Attention Heads. The model applies Multi-Head Attention(MHA) as default for simplicity, but supports Grouped Query Attention(GQA)/Multi-Query Attention(MQA) as well.

Within the attention heads, the model introduces QK normalization, which stabilizes logit scale across layers/steps and reduces sensitivity to learning rate, giving rise to less softmax saturation, fewer exploding heads as well as smoother gradients.

1.3 POSITIONAL EMBEDDING

The model applies RoPE to Q, K within the attention heads instead of learned position embedding. The parameter-free RoPE not only improves efficiency, but also enhances stability in longer context, especially when it is combined with Q, K norm.(Su et al., 2023)

1.4 MLP WITH RELU^2

The model employs the original MLP with squared ReLU activation function.

Compared with GLU variants which trade 1.5× parameters for 1-3% performance improvement(Shazeer, 2020), standard MLP saves up to 20% training cost and time. This configuration is particularly suitable for the 561M nanochat model, which aims to be ”the best ChatGPT that $100 can buy.”

Compared with ReLU, SwiGLU, and ReGLU, ReLU^2 excels across all three evaluation metrics: the trade-off between sparsity and performance, the predictivity of sparsity, and the hardware affinity (Zhang et al., 2024).

1.5 RMSNORM

The model applies Pre-norm RMSNorm with no learnable parameters after token embedding, before attention and MLP in each block, and once more before the LM head.

Compared to LayerNorm, which does 5 operations each time, parameter-free RMSNorm does 2. Likewise, it saves training cost and time. Besides, modded-nanogpt(Jordan et al., 2024) experiments shows that parameter-free RMSNorm trains just as well at small scale. We may hypothesize that at 100M-1B parameter scale, the model has enough capacity that normalization does not need learnable adaptation.

1.6 FLASHNORM

The combination of bias-free linear layers, QK normalization and parameter-free are in line with the concept of FlashNorm(Graef et al., 2025). The paper also introduces ways of optimization by integrating RoPE into normalization and computing RMSNorm and linear layers in parallel, but Karpathy did not adopt them.

1.7 UNTIED EMBEDDING/UNEMBEDDING

The untied embedding/unembedding let input embeddings and output classifier specialize independently, allowing for different learning rate configuration(0.2 for embedding and 0.0004 for unembedding) and different optimizers(AdamW for embeddings, Muon for transformer)(Karpathy, 2025).

1.8 LOGITS SOFTCAP

The model employs Logits softcap(tanh) to clip extreme logits, avoiding unstable gradients and overflow risk. According to (Rybakov et al., 2024), its combination with QK normalization improves training robustness and sampling temperature behavior.

ACKNOWLEDGMENTS

Thanks to Zhengfu He for guiding my research career :P.

REFERENCES

Nils Graef, Andrew Wasielewski, and Matthew Clapp. Flashnorm: fast normalization for llms, 2025. URL https://arxiv.org/abs/2407.09577.

Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977. modded-nanogpt: Speedrunning the nanogpt baseline, 2024. URL https://github.com/KellerJordan/modded-nanogpt.

Andrej Karpathy. nanochat: The best chatgpt that $100 can buy, 2025. URL.$

Oleg Rybakov, Mike Chrzanowski, Peter Dykas, Jinze Xue, and Ben Lanir. Methods of improving llm training stability, 2024. URL https://arxiv.org/abs/2410.16682.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Zhengyan Zhang, Yixin Song, Guanghui Yu, Xu Han, Yankai Lin, Chaojun Xiao, Chenyang Song, Zhiyuan Liu, Zeyu Mi, and Maosong Sun. Relu2 wins: Discovering efficient activation functions for sparse llms. *arXiv preprint arXiv:2402.03804*, 2024.