

STREAMLINING REDUNDANT LAYERS TO COMPRESS LARGE LANGUAGE MODELS

面向大语言模型的层级剪枝与轻量替代

ICLR 2025

Renmin University of China

研究背景

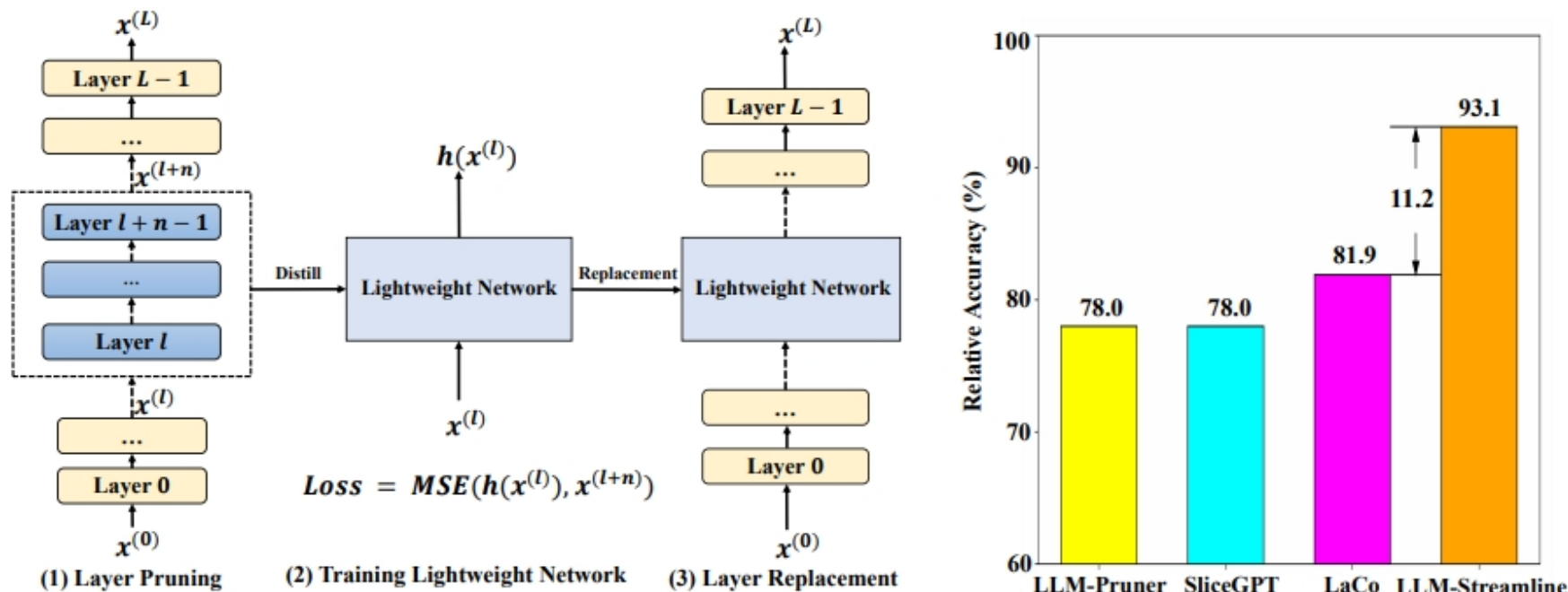


Figure 1: The left side of the figure illustrates the LLM-Streamline workflow, which includes layer pruning to remove consecutive layers and layer replacement where a lightweight network is trained to replace the pruned layers. The right side of the figure presents the comparison results of LLM-Streamline with the state-of-the-art (SOTA) pruning methods on 12 classification benchmarks (details in Section 4.2) after pruning about 25% of the parameters on Llama2-7B. LLM-Streamline achieves 11.2% higher relative accuracy than these methods, where the relative accuracy represents the percentage of the original model’s accuracy retained by the pruning method.

核心观察

- LLM 中存在连续冗余层
- 作者将 Transformer 中的每一层看作对 hidden states 的一次表示变换。如果某层的输入和输出非常接近，就说明这一层对表示的修正幅度较小，其功能贡献可能相对有限。
- 为了刻画这种“层作用大小”，论文使用输入 hidden states 与输出 hidden states 的余弦相似度作为层重要性的度量：相似度越高，说明该层前后表示变化越小，也就越可能是低重要性层。
- 作者在 OPT-1.3B、OPT-2.7B、OPT-6.7B 和 Llama2-7B 上进行了分析，结果发现：部分低重要性层并不是孤立分布的，而是呈现连续出现的趋势。这说明模型中可能存在一段连续的“冗余变换区间”。

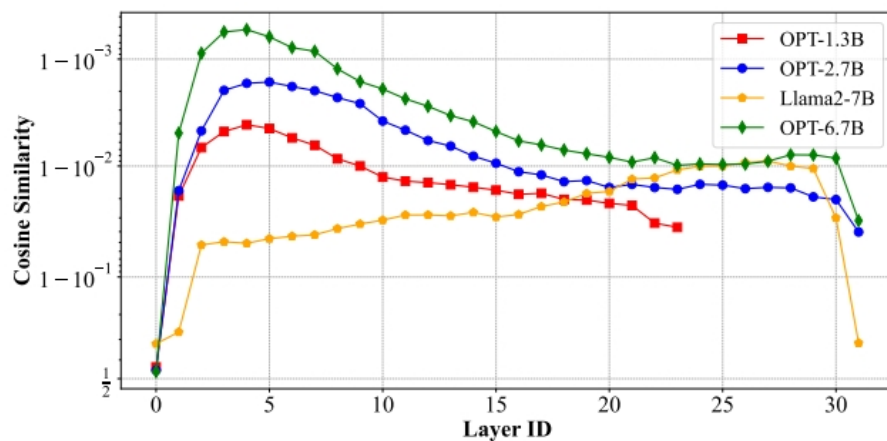
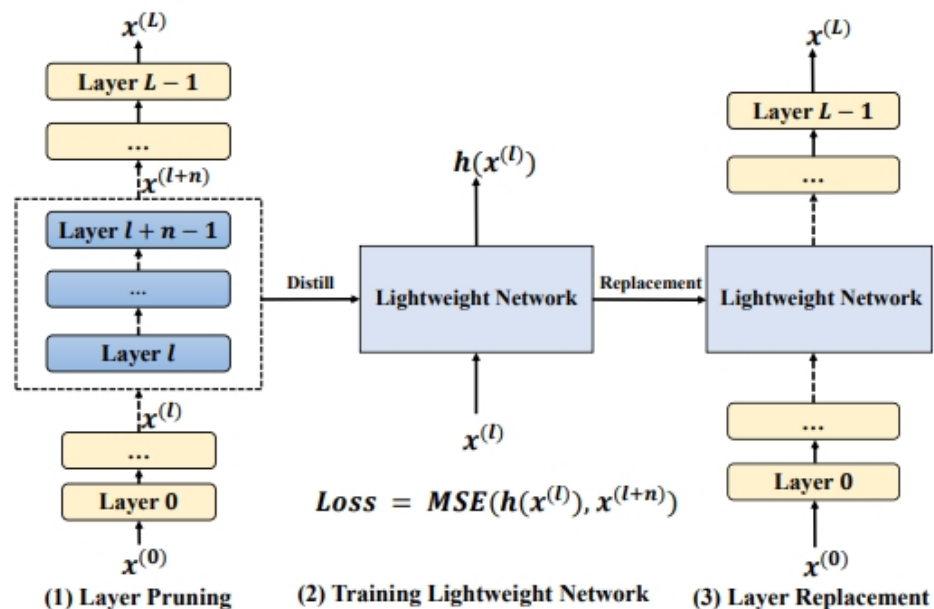


Figure 2: The cosine similarity between the input and output hidden states of each layer in OPT-1.3B, OPT-2.7B, OPT-6.7B, and Llama2-7B.

思路

LLM-Streamline 的核心思路可以概括为三个连续步骤：

- **Layer Pruning**: 根据目标稀疏率确定待删除层数, 并通过 hidden states 前后相似性定位一段对表示变化影响最小的连续层, 将其作为剪枝对象。这样做避免了逐层零散删除, 使压缩过程更加规整, 也更符合模型内部连续冗余层的分布特点。
- **Training Lightweight Network**: 基于被删层前后的 hidden states 构造监督信号, 训练一个轻量网络从输入表示拟合原始连续层变换后的输出表示。该过程以 MSE 为目标, 本质上是在学习被删除层段的整体功能映射, 而不是重新微调整个模型。
- **Layer Replacement**: 将训练好的轻量网络插回原模型, 用一个参数更少、结构更紧凑的模块替代原先的多层连续 Transformer, 从而在减少参数与计算量的同时, 尽可能保留原模型的表示能力并缓解性能下降。



实验设置

模块	内容
基础模型	Llama2-7B, Llama2-13B
剪枝率	默认约 25%
数据来源	SlimPajama
层剪枝数据	500 samples
替代网络训练数据	30,000 samples
替代模块	None / FFN / Layer
评测任务	12 classification + 3 generation

作者在 Llama2-7B 和 13B 上进行验证，默认剪枝率约为 25%。数据来自 SlimPajama，其中只有 500 条样本用于定位待剪的连续层，而训练轻量替代网络使用 3 万条样本。实验同时比较了不加替代模块、FFN 替代和 Transformer layer 替代三种方案，并在 12 个分类任务和 3 个生成任务上进行评测。

实验结果

Table 2: Accuracy of pruned results in the original pre-trained model’s performance retained

LLM	Method	Ratio	C3
Llama2-7B	Dense	0.00%	43.8
	LLMPruner	24.8%	29.7
	SliceGPT	25.4%	31.5
	LaCo*	27.0%	39.7
	Ours (None)	24.0%	40.2
	Ours (FFN)	25.0%	40.7
	Ours (Layer)	24.0%	43.3
Llama2-13B	Dense	0.00%	47.5
	LLMPruner	24.4%	29.5
	SliceGPT	23.6%	38.6
	LaCo*	24.6%	44.9
	Ours (None)	24.6%	47.0
	Ours (FFN)	25.4%	<u>45.8</u>
	Ours (Layer)	24.6%	<u>45.7</u>

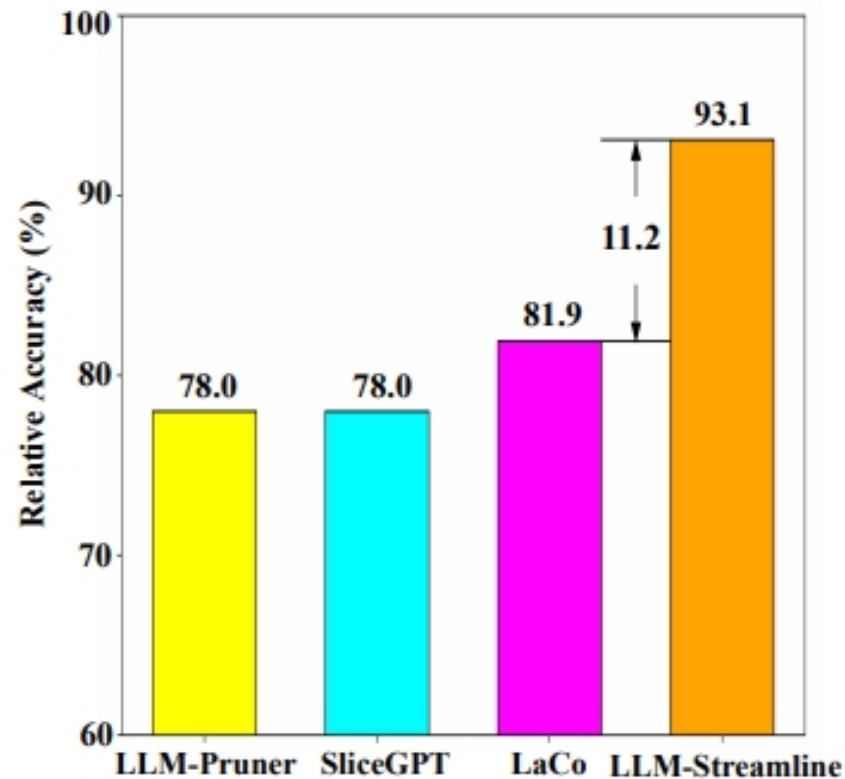


Table 3: Accuracy of pruned results in the original pre-trained model’s performance retained

LU	CMMLU	Average	RP
.8	31.8	49.2	100.0
.3	25.0	38.4	78.0
.8	24.8	38.4	78.0
.5	25.2	40.3	81.9
.6	28.9	44.0	89.4
.0	31.7	45.8	93.1
<u>.5</u>	<u>29.4</u>	<u>45.2</u>	<u>91.9</u>
.8	38.7	57.0	100.0
.2	24.9	38.6	67.7
.6	25.6	34.9	61.2
.9	32.6	51.3	90.0
.8	38.4	54.5	95.6
.7	<u>38.6</u>	<u>53.2</u>	<u>93.3</u>
<u>.1</u>	<u>39.2</u>	<u>53.1</u>	<u>93.2</u>

论文结果表明，在约 25% 剪枝率下，针对 Llama2-7B，作者方法中的 Ours (FFN) 可以达到 93.1% 的 retained performance，而 LLM-Pruner 和 SliceGPT 都只有 78.0%，LaCo 为 81.9%；在 Llama2-13B 上，Ours (FFN) 的 retained performance 为 93.3%，同样高于主要对比方法。换句话说，在保留约四分之三参数规模压缩收益的同时，该方法仍能维持原模型约九成以上的分类能力

实验结果

Table 4: Evaluations on generation benchmarks. “*” indicates that we refer to the results in the original paper.

LLM	Method	Ratio	Xsum	Benchmarks		Average	RP
				GSM8K	StrategyQA		
Llama2-7B	Dense	0.00%	19.4	16.5	60.2	32.0	100.0
	LLMPruner	24.8%	16.4	0.61	44.2	20.4	63.8
	SliceGPT	25.4%	12.4	3.34	45.7	20.5	64.1
	LaCo*	27.1%	15.6	-	-	-	-
	Ours (None)	24.0%	14.8	1.97	41.8	19.5	60.9
	Ours (FFN)	25.0%	<u>18.6</u>	<u>2.16</u>	<u>46.5</u>	<u>22.4</u>	<u>70.0</u>
	Ours (Layer)	24.0%	20.2	1.82	52.1	24.7	77.2
Llama2-13B	Dense	0.00%	23.7	29.0	58.1	36.9	100.0
	LLMPruner	24.4%	17.5	1.9	43.7	21.0	56.9
	SliceGPT	23.6%	5.0	1.9	38.3	15.1	40.9
	LaCo*	24.6%	14.5	-	-	-	-
	Ours (None)	24.6%	17.7	2.35	46.0	22.0	59.6
	Ours (FFN)	25.4%	<u>21.4</u>	<u>4.10</u>	59.6	28.4	77.0
	Ours (Layer)	24.6%	21.8	4.70	<u>57.3</u>	<u>27.9</u>	<u>75.6</u>

LLM-Streamline 在生成任务上同样优于主流剪枝方法，在约 25% 剪枝下仍可保留约 77% 的原模型能力，但 GSM8K 等数学推理任务仍然十分困难。

实验结果

轻量替代网络到底该选什么结构最好？

Table 5: Comparison of different lightweight networks on classification benchmarks in terms of average accuracy and stability metrics, where “†” indicates that the intermediate size of the added lightweight network is half that of the default LLM’s intermediate size.

	Layer-Random	Layer-First	Layer-Last	Layer-Avg	FFN†	FFN	SwiGLU†	SwiGLU
Accuracy	45.1	45.2	45.6	44.4	46.0	<u>45.8</u>	43.8	44.2
Stability	81.2	82.7	81.9	79.2	80.7	83.7	82.6	<u>83.3</u>

在轻量替代网络的多种候选结构中，FFN 综合效果最佳；若采用 Transformer layer，则继承第一层参数的初始化方式最优。

实验结果

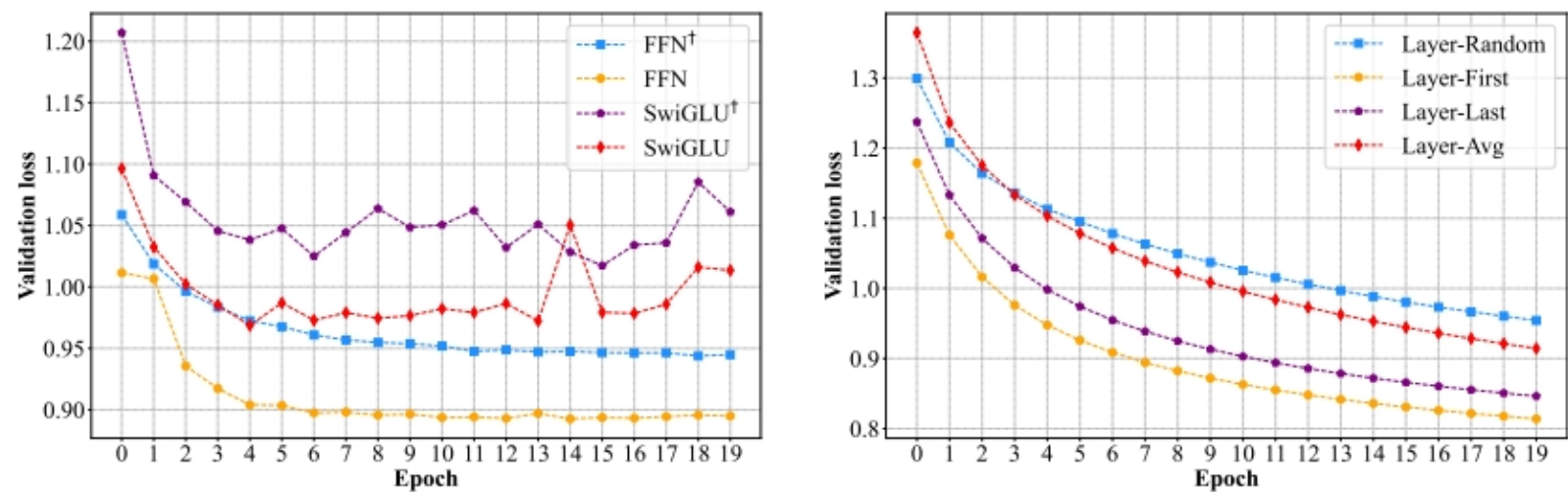


Figure 3: Validation loss curves during training of (a) FFN and SwiGLU; (b) Transformer layer.

左图可以看到，FFN 和 SwiGLU 在前几个 epoch 内下降很快，并且大约在第 10 个 epoch 左右已经基本收敛，其中 FFN 曲线更低、更稳定。

右图比较了 Transformer layer 的不同初始化方式，可以看到 Layer-First 的 loss 最低，说明继承被删第一层参数的初始化最有效。与此同时，Transformer layer 的 loss 到后期仍在持续下降，说明它还有进一步训练提升的潜力，只是需要更多计算资源。

实验结果

Table 6: Comparison of layer replacement and LoRA on classification benchmarks in terms of average accuracy metrics across all benchmarks, where “†” indicates that the intermediate size of the added lightweight network is half that of the default LLM’s intermediate size.

	Layer-First	Layer-Last	Layer-Avg	FFN†	FFN	SwiGLU†	SwiGLU	LoRA
Accuracy	<u>46.7</u>	46.8	46.2	45.8	46.3	44.4	45.5	44.5
Stability	85.7	<u>85.6</u>	83.9	83.4	85.2	84.7	84.7	82.1
GPU Memory (G)	27.8	27.8	27.8	<u>25.6</u>	27.0	25.3	26.4	56.4

这张表比较的是剪枝后的两类恢复方案：一类是论文提出的 layer replacement，也就是用不同轻量网络替代被删层；另一类是更常见的 LoRA 微调。结果可以看到，LoRA 在 accuracy、stability 和显存开销三方面都不占优。

实验结果

As shown in the Table 11, we evaluate the inference speed of various pruning methods at similar pruning ratio when generating sequences of length 128. The results indicate that the acceleration effect of LLM-Streamline is slightly inferior to that of SliceGPT and LLM-Pruner.

Table 11: The inference speed of models pruned using different methods.

Llama2-7B	Dense	LLM-Pruner	SliceGPT	Ours(None)	Ours(FFN)	Ours(Layer)
Pruning Ratio (%)	0.00	24.8	25.4	24.0	25.0	24.0
Inference Speed (tokens/s)	19.87	25.91	27.20	25.68	25.88	25.68

LLM-Streamline 大约 25.68–25.88 tokens/s，比原始 Llama2-7B 的 19.87 tokens/s 快，但略低于 SliceGPT 的 27.20 tokens/s 和 LLM-Pruner 的 25.91 tokens/s。