AlphaPruning: Using Heavy-Tailed Self Regularization Theory for Improved Layer-wise Pruning of Large Language Models

Introduction

近年来，大型语言模型（LLMs）的剪枝技术取得了显著进展，例如 Frantar 和 Alistarh (2023a) 的工作表明可以在性能不受损的情况下大幅减少参数量，从而在内存占用、计算时间和能耗上带来显著节约。然而，现有剪枝策略通常在所有层中分配统一的“稀疏预算”（即剪枝比例），这限制了高稀疏性的实现。

现有方法（如 Yin et al., 2023）尝试通过层级剪枝实现非均匀稀疏性，但其主要依赖启发式规则，尤其是与异常值激活分布相关的启发式方法。这些方法在缺乏异常值的情况下表现欠佳，难以实现高稀疏性剪枝。此外，80% 稀疏性通常会显著降低预测性能。

本文利用**Heavy-Tailed Self-Regularization (HT-SR)** 理论，特别是权重矩阵的经验谱密度（Empirical Spectral Densities, ESDs）的形状，设计了改进的层级剪枝比例分配方法。分析表明，不同层训练质量的差异显著影响其剪枝能力。基于此，本文提出了**AlphaPruning**，利用形状指标分配层级稀疏比例，从理论上更合理地分配剪枝资源。

AlphaPruning 方法可以与现有多种剪枝技术结合，例如通过该方法可以在 LLaMA-7B 模型中实现 80% 稀疏性，并保持合理的困惑度（perplexity），标志着 LLM 剪枝领域的重要进展。

本文的主要贡献包括：

1. 系统性地分析基于权重矩阵的多种指标，用以估计层质量并指导稀疏分配。
2. 提出了一种基于 HT-SR 理论的剪枝方法，通过层 ESD 的形状特性分配剪枝资源。
3. 通过实验证明，AlphaPruning 在剪枝后模型性能和计算效率上均优于现有方法。

Related Work

剪枝是一种在训练后的神经网络（NN）中移除冗余权重或连接以生成高效压缩模型的技术。这种方法具有悠久的研究历史，并且在近年来的大型语言模型（LLMs）中得到了深入应用。以下为相关领域的关键研究方向和方法。

剪枝方法

早期研究表明，现代神经网络通常是过参数化的 (Bhojanapalli et al., 2021; Wang and Tu, 2020)，移除冗余参数可以提高计算和内存效率。最常见的剪枝方法是基于权重大小的剪枝 (Han et al., 2015)，即将权重较小的连接置为零。这种方法对于传统模型效果显著，但在大型语言模型中应用存在困难。具体而言，大型语言模型的剪枝通常需要重新训练以恢复性能 (Blalock et al., 2020)，而这在资源受限的情况下具有挑战性。

为了解决这一问题，研究者开发了一些专门针对大型语言模型的剪枝算法。例如：

- **SparseGPT** (Frantar and Alistarh, 2023b)：利用 Hessian 矩阵的逆矩阵来更新剪枝后的权重，减少稠密和稀疏权重之间的重构误差。
- **Wanda** (Sun et al., 2023)：结合权重大小和输入激活，设计了一种保留异常值特征的剪枝标准。

这些方法在较低稀疏性（50%左右）下可以维持合理性能，但实现更高稀疏性（例如 80%）时，性能仍然显著下降。

层级稀疏分配

虽然层级稀疏分配在预训练模型的剪枝中得到了广泛研究 (Evci et al., 2020; Gale et al., 2019; Lee et al., 2020)，但在大型语言模型中的应用仍然有限。现有方法主要使用启发式规则，例如：

- **Uniform Pruning**：在所有层分配统一稀疏比例 (Frantar and Alistarh, 2023b; Sun et al., 2023)。
- **OWL** (Yin et al., 2023)：基于异常值分布为每层分配稀疏比例，但对异常值的依赖导致性能在某些模型中受限。

其他层级分配方法，如基于矩阵范数的分配 (Lin et al., 2020; Lee et al., 2020) 和错误阈值分配 (Ye et al., 2020; Zhuang et al., 2018)，虽然在传统模型中效果显著，但在大型语言模型中并不理想。

HT-SR 理论的应用

HT-SR 理论通过分析权重矩阵的经验谱密度（ESD），为模型训练质量提供了理论依据 (Martin and Mahoney, 2019, 2020)。该理论表明，具有强重尾（Heavy-Tailed）特性的层通常训练更充分，表现出更强的信号学习能力。基于这一理论的研究主要集中于以下方面：

1. 模型选择和质量评估 (Martin et al., 2021; Zhou et al., 2023)。
2. 层级适应性训练 (Yang et al., 2023)。
3. 剪枝方法的优化：例如通过 PL_Alpha_Hill 指标量化 ESD 的重尾程度，并以此指导剪枝分配。

本研究在 HT-SR 理论的基础上，提出了一种更理论驱动的剪枝方法，称为 AlphaPruning。与现有方法相比，AlphaPruning 在层级稀疏分配上具有更高的鲁棒性和理论依据。

Background and Setup

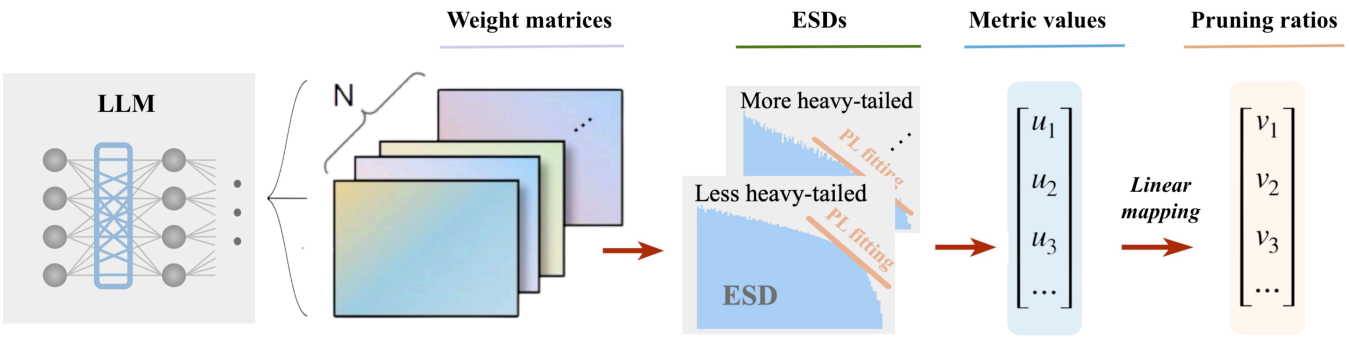

Figure 1: **The pipeline diagram of** `AlphaPruning`. Our post-training layer-wise pruning method involves the following steps: (i) Performing ESD analysis on all weight matrices of a base LLM and (ii) employing PL fitting to derive the layer-wise metric values (that measures the HT exponent). Then, (iii) using the layer-wise metric values, we assign layer-wise pruning ratios to each layer through a linear assignment function.

本节将介绍实验中的背景知识、符号定义，以及实验所用的模型、评价指标和基线方法。

3.1 Notation

本文使用以下符号描述模型和方法：

1. **神经网络结构**：
   - 假设一个神经网络NN由L 个层组成，每一层包含一个权重矩阵W_i，其维度为$m \times n (m \geq n)$。
   - 本文中的“层”特指 Transformer 模块，每个模块包含多个权重矩阵，例如注意力层权重矩阵和投影层权重矩阵。
2. **相关矩阵**：

- 定义第 i 层的相关矩阵为：

$$X_i = W_i^\top W_i \tag{1}$$

其中X_i是对称矩阵，维度为$n \times n$。

3. **经验谱密度（Empirical Spectral Density, ESD）**：
   - 定义X_i的经验谱密度为：

$$\mu_{X_i} := \frac{1}{n} \sum_{j=1}^{n} \delta_{\lambda_j(X_i)} \tag{2}$$

其中：

- $\lambda_1(X_i) \leq \lambda_2(X_i) \leq \ldots \leq \lambda_n(X_i)$表示 X_i的特征值；
- (\delta) 为 Dirac Delta 函数。

3.2 HT-SR Theory and Metrics

Heavy-Tailed Self-Regularization (HT-SR) 理论 提供了分析神经网络权重矩阵的经验谱密度 (ESD) 的理论基础，用于量化模型训练质量。

核心概念

1. **信号与噪声**：
   - 根据随机矩阵理论，ESD 中的“尖峰”代表信号，而“主体部分”代表遵循 Marchenko-Pastur 分布的噪声。
   - 信号（尖峰）通常与训练质量相关。
2. **重尾特性**：
   - ESD 的重尾特性反映了权重矩阵中元素的相关性。

- 训练充分的模型通常表现出强重尾特性，这表明权重矩阵的元素具有更高的相关性。

3. **HT-SR 指标分类**：
   - **规模指标（Scale Metrics）**：
     - 描述权重矩阵的整体规模，例如 Frobenius_Norm 和 Spectral_Norm。
     - 公式：

$$\text{Frobenius\_Norm} = \|W\|_F^2, \quad \text{Spectral\_Norm} = \|W\|_2^2 \quad (3$$

   - **形状指标（Shape Metrics）**：
     - 描述 ESD 的形状特性，例如 PL_Alpha_Hill、Alpha_Hat、Stable_Rank 和 Entropy。
     - 本文的重点是 PL_Alpha_Hill 指标，其定义见 **4.2 Estimating Layer Quality by HT Metric**。

5.1 Experimental Setup

模型与评价方法

本文在以下模型上评估 **AlphaPruning** 方法：

1. **模型**：
   - LLaMA 系列：7B、13B、30B、65B (Touvron et al., 2023a)；
   - LLaMA-2 系列：7B、13B、70B (Touvron et al., 2023b)；
   - 其他先进模型：LLaMA-3-8B、Vicuna-7B、Mistral-7B。
2. **评价指标**：

- **困惑度（Perplexity）**：在 WikiText 验证集上的困惑度，用于评估语言建模能力。
- **零样本任务准确率**：评估剪枝模型在 BoolQ、RTE、HellaSwag 等七个下游任务中的零样本能力。

基线方法

本文将 **AlphaPruning** 与以下剪枝方法进行对比：

1. **统一剪枝**（Uniform Pruning）：
   - 在所有层分配相同的稀疏比例。
2. **OWL**（Outlier Weighed Layerwise Sparsity）：
   - 基于异常值分布为每层分配稀疏性。
3. **其他剪枝方法**：
   - Magnitude-Based Pruning (Han et al., 2015)：基于权重大小的剪枝。
   - SparseGPT (Frantar and Alistarh, 2023b)：利用 Hessian 逆矩阵优化剪枝。
   - Wanda (Sun et al., 2023)：结合权重大小和输入激活的剪枝方法。

5.2 Main Results

数据与设置

1. **剪枝配置**：
   - 在 70% 和 80% 稀疏性的配置下，比较不同剪枝方法的效果。
   - 使用 AlphaPruning 方法优化稀疏比例分配。
2. **实验验证**：

- 在 LLaMA 和 LLaMA-2 系列模型中验证方法有效性。
- 比较 AlphaPruning 与其他基线方法在剪枝后的性能差异。

实验的背景和设置为后续的详细实验结果提供了坚实基础。

Alpha-Pruning

Alpha-Pruning 是一种基于 **Heavy-Tailed Self-Regularization (HT-SR)** 理论的层级剪枝方法。本文通过分析权重矩阵的经验谱密度（Empirical Spectral Density, ESD）的形状特性，提出了一种理论驱动的剪枝策略。以下将详细阐述 Alpha-Pruning 的核心思想、方法步骤和公式解释。

核心思想

Alpha-Pruning 的基本理念是根据每层权重矩阵的重尾特性分配剪枝稀疏比例：

1. **重尾特性测量**：
   - 利用幂律分布拟合 ESD 并计算幂律指数 (\alpha)，用以量化层的训练质量。
   - 重尾指数α越小，表明该层训练质量越高。
2. **稀疏性分配策略**：
   - 训练质量较高的层（重尾指数小）分配较低稀疏性，以尽可能保留其信号。
   - 训练质量较低的层分配较高稀疏性，从而优先剪枝。

方法流程

Alpha-Pruning 的具体实现可以分为以下几步：

1. 计算经验谱密度（ESD）

对于每一层权重矩阵 W_i，计算其相关矩阵：

$$X_i = W_i^\top W_i \tag{4}$$

然后，基于X_i的特征值$\lambda_1(X_i), \lambda_2(X_i), \ldots, \lambda_n(X_i)$构造经验谱密度：

$$\mu_{X_i} := \frac{1}{n} \sum_{j=1}^{n} \delta_{\lambda_j(X_i)} \tag{5}$$

其中δ是 Dirac Delta 函数。

---

2. 计算幂律指数$\mathrm{PL_Alpha_Hill}$

利用 Hill 估计器计算重尾指数$\mathrm{PL_Alpha_Hill}$，其公式为：

$$\mathrm{PL\_Alpha\_Hill} = 1 + \frac{k}{\sum_{i=1}^{k} \ln \frac{\lambda_{n-i+1}}{\lambda_{n-k}}} \tag{6}$$

- (k) 是一个可调参数，用于设定 ESD 的下限 (\lambda_{\text{min}})；
- λ_{n-i+1}表示从大到小排序的特征值。

通过调节 (k)，确保$\lambda_{\min}$对应 ESD 的峰值，从而优化估算的准确性。

---

3. 分配稀疏性

根据每层的$\mathrm{PL_Alpha_Hill}$指标，利用以下映射函数分配稀疏性：

$$\phi(q)_i = \eta \left[\frac{q_i - q_{\min}}{q_{\max} - q_{\min}} (s_2 - s_1) + s_1 \right] \tag{7}$$

其中：

- $\phi(q)_i$：第 (i) 层的稀疏比例；
- q_i：第 (i) 层的重尾指数$\mathrm{PL_Alpha_Hill}$；
- $q_{\min}, q_{\max}$：所有层的重尾指数的最小值和最大值；
- s_1, s_2：稀疏比例的上下限；
- η：归一化因子，用于确保全局稀疏性满足目标值 (S)：

$$\sum_{i=1}^{L} \phi(q)_i d_i = S \cdot \sum_{i=1}^{L} d_i \tag{8}$$

其中d_i是第 (i) 层的参数数量。

---

4. 剪枝执行

根据分配的稀疏比例$\phi(q)_i$，对每一层的权重矩阵进行剪枝。可以结合多种剪枝方法，例如：

- **Wanda**：基于权重大小和激活值的剪枝；
- **SparseGPT**：基于 Hessian 矩阵优化剪枝。

方法优点

1. **理论驱动**：通过 HT-SR 理论，Alpha-Pruning 提供了明确的剪枝分配依据，避免了启发式方法的局限性。
2. **通用性**：可与多种现有剪枝方法结合，提升剪枝后的模型性能。
3. **高效性**：在保持性能的同时，显著提高模型的稀疏性，降低推理时的计算复杂度。

形状指标与规模指标的比较

实验表明，形状指标（如 $\mathrm{PL\_Alpha\_Hill}$）优于传统规模指标（如 Frobenius_Norm 和 Spectral_Norm）。以下为核心发现：

1. **形状指标更鲁棒**：在 WikiText 数据集上的困惑度和零样本任务的准确率均显著优于规模指标。
2. **Alpha-Pruning 的表现**：
   - LLaMA-7B 模型在 70% 稀疏性下的困惑度从 48419.13 降至 231.01。
   - 零样本任务准确率提高了 4.6%。

实验结果总结

通过 Alpha-Pruning，LLaMA-7B 模型在 80% 稀疏性下：

- 困惑度保持在合理范围；
- 平均准确率相比基线方法提升显著；
- 推理速度提升 3 倍以上。

Empirical Results

本节详细分析 **Alpha-Pruning** 的实验性能，包括对比基线方法的效果、剪枝后模型的性能表现，以及方法的泛化性和推理效率。

5.1 Experimental Setup

模型和任务设置

1. **评估模型**：

- **语言模型**：LLaMA-7B/13B/30B/65B, LLaMA-2-7B/13B/70B, Vicuna-7B, Mistral-7B。
- **评价任务**：
  - **困惑度**：在 WikiText 数据集上测试模型的语言建模能力。
  - **零样本任务**：在 BoolQ、RTE、HellaSwag、ARC 等任务上评估模型的下游表现。

2. **剪枝方法**：
   - **基线方法**：
     - Uniform Pruning：统一稀疏比例剪枝。
     - OWL：基于异常值激活分布的剪枝。
     - SparseGPT 和 Wanda：分别结合 Alpha-Pruning 进行稀疏性分配。
   - **Alpha-Pruning**：作为稀疏比例分配的核心方法。

5.2 Main Results

语言建模性能

困惑度对比：
以下是 LLaMA 和 LLaMA-2 系列在 70% 稀疏性下的困惑度比较：

模型	剪枝方法	LLaMA-7B	LLaMA-13B	LLaMA-30B	LLaMA-65B
Dense Model	无剪枝	5.68	5.09	4.77	3.56
Uniform	统一稀疏比例	48419.13	84527.45	977.76	46.91
OWL	异常值驱动剪枝	19527.58	11464.69	242.57	15.16
Alpha-Pruning	层级分配稀疏性	**231.01**	**2029.20**	**62.39**	**16.01**

- **结果分析**：

- Alpha-Pruning 显著降低了剪枝后模型的困惑度。
- 在 LLaMA-7B 模型中，困惑度从 Uniform 剪枝的 48419.13 降至 231.01，提升了 200 倍以上。

零样本任务性能

平均准确率对比：
在 70% 稀疏性下，评估 LLaMA 和 LLaMA-2 模型的零样本任务表现：

模型	剪枝方法	平均准确率 (%)
Dense Model	无剪枝	60.08
Uniform	统一稀疏比例	32.30
OWL	异常值驱动剪枝	33.57
Alpha-Pruning	层级分配稀疏性	**35.67**

- **结果分析**：
  - Alpha-Pruning 在零样本任务中相较于 Uniform 剪枝提升了 10% 以上的准确率。
  - 方法能够更好地保留关键层的信息，从而增强模型的泛化能力。

5.3 Generalizability of Alpha-Pruning

方法的泛化性

1. **多种剪枝方法的结合**：
   - 将 Alpha-Pruning 与 SparseGPT 和 Wanda 剪枝方法结合，进一步提升性能。

- 在多个模型（例如 Vicuna-7B 和 Mistral-7B）中验证了该方法的普适性。

2. **任务扩展性**：
   - 除语言模型外，Alpha-Pruning 还应用于计算机视觉任务（例如 Vision Transformers, ViTs）。
   - 结果表明，Alpha-Pruning 同样能够在这些任务中有效分配稀疏性，并显著提高性能。

5.4 Efficiency Improvements

推理效率

通过剪枝降低模型的计算复杂度，从而加速推理。以下是在 DeepSparse 推理引擎上的实验结果：

稀疏性 (%)	延迟 (ms)	吞吐量 (tokens/s)	加速倍率 (Speedup)
0 (Dense)	307.46	3.25	1.00×
50	177.55	5.63	1.73×
70	133.76	7.47	2.30×
80	100.35	9.96	**3.06×**

- **结果分析**：
  - 在 80% 稀疏性下，推理速度提升至 Dense 模型的 3 倍以上。
  - Alpha-Pruning 在保证性能的同时，显著减少了计算开销。

5.5 Summary

Alpha-Pruning 的实验结果表明：

1. 在语言建模任务中，显著优于传统剪枝方法（例如 Uniform 和 OWL）。
2. 在高稀疏性条件下（80%），能够保持模型性能并提升推理效率。
3. 方法具有良好的通用性，适用于不同模型架构和任务。

Conclusion

本文提出了一种新的剪枝方法 **Alpha-Pruning**，通过结合 **Heavy-Tailed Self-Regularization (HT-SR)** 理论，有效提升了大型语言模型（LLMs）的剪枝效果。本方法基于权重矩阵的经验谱密度（Empirical Spectral Density, ESD）的形状特性，设计了更为合理的层级稀疏性分配策略。以下总结本文的主要贡献和实验结论：

主要贡献

1. **理论驱动的剪枝策略**：
   - 基于 HT-SR 理论，提出了利用重尾特性（Heavy-Tailed Properties）分配层级稀疏性的 Alpha-Pruning 方法。
   - 通过分析权重矩阵的 ESD 形状（如幂律指数 PL_Alpha_Hill），为剪枝提供了理论依据。
2. **形状指标优于传统规模指标**：
   - 实验表明，ESD 的形状指标（Shape Metrics）在预测层质量和指导稀疏性分配方面表现优于传统的规模指标（Scale Metrics）。
   - 通过层级稀疏性优化，实现了更高的剪枝效率和模型性能。
3. **通用性和可扩展性**：
   - Alpha-Pruning 适用于多种剪枝方法（例如 Wanda 和 SparseGPT），并在多个模型（如 LLaMA、Vicuna、Mistral）上验证了其有效性。

- 方法还可扩展至计算机视觉任务（例如 Vision Transformers），展现了良好的跨领域泛化能力。

实验结论

1. **高稀疏性下的性能保持**：
   - Alpha-Pruning 在 80% 稀疏性下显著降低了困惑度（如 LLaMA-7B 模型的困惑度降至 231.01）。
   - 零样本任务的准确率也显著优于基线方法（如 Uniform 和 OWL）。
2. **推理效率的显著提升**：
   - 剪枝后的模型在 CPU 上的推理速度提升至 Dense 模型的 3 倍以上（在 DeepSparse 推理引擎中验证）。
3. **剪枝后的模型质量**：
   - Alpha-Pruning 能有效控制剪枝对模型质量的损害，表现为剪枝后模型的 (\text{PL_Alpha_Hill}) 指标较低，表明剪枝策略更科学。

未来研究方向

尽管 Alpha-Pruning 在模型剪枝领域取得了显著进展，仍有以下几个方向值得进一步研究：

1. **自动化稀疏性分配**：
   - 通过强化学习或元学习进一步优化稀疏性分配策略。
2. **更高效的剪枝算法**：
   - 结合硬件优化设计剪枝方法，进一步提升模型的推理速度。
3. **扩展到更多领域**：

- 将 Alpha-Pruning 应用于其他深度学习领域，例如多模态学习和时序模型。

Alpha-Pruning 通过其理论驱动的稀疏性分配方法，为模型剪枝领域提供了新的视角和工具。未来，我们期待其在更多任务和场景中的广泛应用。