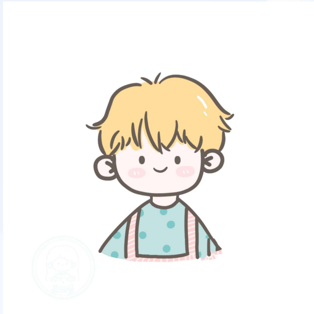
NeurIPS 2024 LLM-Merging

A Model Merging Method

abc team


Jisheng Fang
asdfqwer2015@163.com

Hao Mo
sunshineinautumn@163.com

Qiang Gao
gq15035177217@gmail.com

content

Introduction

01

Competition Goal

Training high-performing large language models (LLMs) from scratch is a notoriously expensive and difficult task, costing hundreds of millions of dollars in compute alone. These pretrained LLMs, however, can cheaply and easily be adapted to new tasks via fine-tuning, leading to a proliferation of models that suit specific use cases. Recent work has shown that specialized fine-tuned models can be rapidly merged to combine capabilities and generalize to new skills.

Current Methods

- **Parameter Averaging**
- **Model Stacking**
- **Model Routing**
- MoE-based merging
- Model Zipping

Model Selection

02

Base Model Selection

- meta-llama/Meta-Llama-3-8B-Instruct
  - broad knowledge
  - skilled at summarizing
  - ecologically rich
- microsoft/Phi-3-small-8k-instruct
  - small and fast
  - skilled at reasoning

Base Model Selection

- Task types by knowledge area
- assessing each fine-tuned model’ s GPU memory usage and accuracy by lm-evaluation-harness and custom datasets

Model Merging

03

Model Merging

Weights Merging

Lower VRAM requirements to support a greater number of models

Router

Determine model selection based on sample analysis

Staged Response

Harness the distinct advantages of multiple base models

Weights Merging

1. Compresses weights for layers (excluding the lm_head and embedding layers)
2. Applies RSVD
3. Connects **parameter averaging** and **model routing**

Weights Merging

Algorithm: Weight compression for a layer in models

Input:

$W = \{W_1, W_2, \ldots, W_N\}$

$compress\_rate$

Output:

$scales = \{scale_1, scale_2, \ldots, scale_N\}$

W_{avg}

$compressed\_diff = \{U_1, U_2, \ldots, U_N, V_1, V_2, \ldots, V_N\}$

1. For each weight matrix $W_i \in W$:

   $scale_i = ||W_i||$

   $\widehat{W}_i = \frac{W_i}{scale_i}$

   # Normalize weight matrix W_i.

2. $W_{av} = \frac{1}{N} \sum \widehat{W}_i$

3. For each normalized weight matrix $\widehat{W}_i$:

   $U_i, V_i = RSVD(\widehat{W}_i - w_{avg}, compress\_rate)$

4. Return $scales, w_{avg}, compressed\_diff$

Algorithm: Inference for Compressed Model Layer

Input:

x

$bias$ # Uncompressed bias

$scales = \{scale_1, scale_2, \ldots, scale_N\}$

W_{avg}

$compressed\_diff = \{U_1, U_2, \ldots, U_N, V_1, V_2, \ldots, V_N\}$

Output:

$y = \{y_1, y_2, \ldots, y_N\}$

1. $y_i = linear(x, w_{avg}) + linear(linear(x, V_i), U_i) * scale_i$

2. If bias is not null:

   $y_i \mathrel{+}= bias_i$

3. Return y # Return the final output.

Weights Merging

1. 95% compression rate
2. Phi3-Small and three fully fine-tuned Llama3 8B models

Router

1. Embedding based
2. LLM instead of PLM
3. Alignment

Router

Alignment

“’’{input}

Let’s think about what task these questions belong to. These questions belong to the field of’’’

Staged Response

Accuracy and Clarity

2-agents, model stacking

Thinker: Phi3-small, guided COT

Formatter: llama3 8B

Conclusions and Outlook

04

Conclusions and Outlook

We ultimately achieved first place with a score of 0.46

LLM Merging Competition FINAL

Late Submission ···

Overview | Data | Code | Models | Discussion | Leaderboard | Rules | Team | Submissions

The private leaderboard is calculated over the same rows as the public leaderboard in this competition.
This competition has completed. This leaderboard reflects the final standings.

#	Team	Members	Score	Entries	Last	Solution
1	**abc_20242024**		0.46	1	2mo	
2	catrin baze		0.45	1	2mo	
3	Zixiang Di		0.44	1	2mo	

The Method with second version of Staged Response gets a higher score of 0.50

Q&A

05

Q&A

If you have any questions, please feel free to email us.

Jisheng Fang
asdfqwer2015@163.com
Hao Mo
sunshineinautumn@163.com
Qiang Gao
gq15035177217@gmail.com

THANKS