Model Merging

Artur Jordão
Escola Politécnica – Engenharia de Computação e Sistemas Digitais
Universidade de São Paulo

Introduction

Model Merging

- Model merging is an increasingly popular technique that can, surprisingly, create a single general model by combining the weights of task-specific models
- Models that perform well on multiple tasks are essential for advancing general-purpose AI[1]
  - General-purpose AI: Models capable of performing a wide range of tasks rather than being specialized for a single function
- By merging a language model specialized in medical knowledge with one specialized in legal knowledge, it would be possible to develop a model capable of solving tasks related to legal issues in medicine

[1] Bengio et al. *International Scientific Report on the Safety of Advanced AI*, 2025

Preliminaries

Model Merging

- Define a set of K tasks $\{t_1, t_2, \cdots, t_K\}$ and K corresponding fine-tuned (ft) models with parameters $\{\theta_{ft}^{t_1}, \theta_{ft}^{t_2}, ..., \theta_{ft}^{t_K}\}$
- Let θ_{pre} denote the parameters of a pre-trained model
- $\theta_{pre} \in \mathbb{R}^d$ and $\theta_{ft}^{t} \in \mathbb{R}^d$
  - d is the parameter dimension
- Model merging aims to fuse the parameters of K models into a single model with parameters $\theta_{\mathrm{M}} \in \mathbb{R}^d$ that can well handle K tasks simultaneously

Task Vectors

Model Merging

- A task vector (τ) specifies a direction in the weight space of a pre-trained model, such that movement in that direction improves performance on the task[1]
  - A vector of weights fine-tuned specifically for a given task, subtracted from the corresponding pre-trained weights

- It involves subtracting the weights of a pre-trained model (θ_{pre}) from the weights of the same model after fine-tuning on a task (θ_{ft})

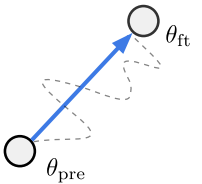


Figure: $\tau = \theta_{ft} - \theta_{pre}$.

[1] Ilharco et al. *Editing models with task arithmetic.* ICLR, 2023

Problem Definition

Model Merging

- Model merging aims to fuse the parameters of K models into a single model with parameters θ_M that can effectively handle K tasks simultaneously

- Ilharco et al.[1] formalize model merge in terms of

$$\theta_M = \theta_{pre} + \sum_{k=1}^{K} \lambda_{t_k} (\theta_{ft}^{t_k} - \theta_{pre}), \tag{1}$$

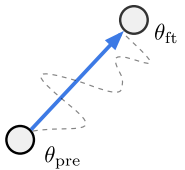
where λ is the scaling term to determine the importance of each model

[1] Ilharco et al. *Editing models with task arithmetic.* ICLR, 2023

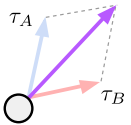
The Role of Arithmetic Operations

Model Merging

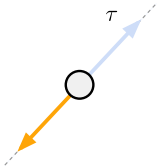
- Learning via addition (b)
  - Adding task vectors results in better multi-task models
- Forgetting via negation (c)
  - Negate task vectors to mitigate undesirable behaviors (forget toxic or bias responses)
- Task analogies (d)
  - Combining task vectors from tasks A, B and C improves performance on D
  - Domain generalization



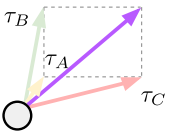
(a) $\tau = \theta_{ft} - \theta_{pre}$



(b) $\tau_{new} = \tau_A + \tau_B$



(c) $\tau_{new} = -\tau$



(d)
$\tau_{new} = \tau_C + (\tau_B - \tau_A)$

Practical Issues

Model Merging

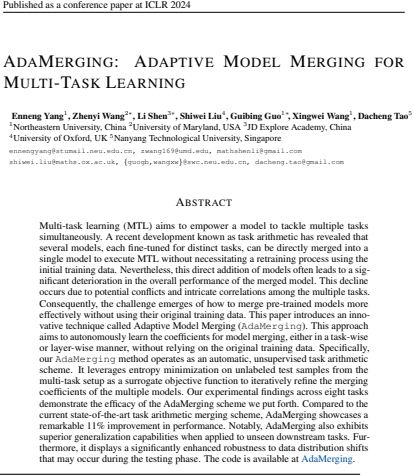
- Merging (or editing) models with task arithmetic is simple, fast, and effective
  - There is no extra cost at inference time in terms of memory or compute, since the mechanism performs **element-wise operations** on model weights
  - Vector operations are cheap, allowing users to experiment quickly with multiple task vectors

- With task arithmetic, practitioners can reuse or transfer knowledge from models they create, or from the multitude of publicly available models all without requiring access to data or additional training

The State of Model Merging

The Role of Coefficients λ_{t_k}

The State of Model Merging



Published as a conference paper at ICLR 2024

ADAMERGING: ADAPTIVE MODEL MERGING FOR MULTI-TASK LEARNING

Enneng Yang[1], **Zhenyi Wang**[2*], **Li Shen**[2*], **Shiwei Liu**[3], **Guibing Guo**[1*], **Xingwei Wang**[1], **Dacheng Tao**[5]
[1]Northeastern University, China [2]University of Maryland, USA [3]JD Explore Academy, China
[4]University of Oxford, UK [5]Nanyang Technological University, Singapore
ennengyang@stumail.neu.edu.cn, zwang169@umd.edu, mathshenli@gmail.com
shiwei.liu@maths.ox.ac.uk, {guogb,wangxw}@swc.neu.edu.cn, dacheng.tao@gmail.com

ABSTRACT

Multi-task learning (MTL) aims to empower a model to tackle multiple tasks simultaneously. A recent development known as task arithmetic has revealed that several models, each fine-tuned for distinct tasks, can be directly merged into a single model to execute MTL without necessitating a retraining process using the initial training data. Nevertheless, this direct addition of models often leads to a significant deterioration in the overall performance of the merged model. This decline occurs due to potential conflicts and intricate correlations among the multiple tasks. Consequently, the challenge emerges of how to merge pre-trained models more effectively without using their original training data. This paper introduces an innovative technique called Adaptive Model Merging (AdaMerging). This approach aims to autonomously learn the coefficients for model merging, either in a task-wise or layer-wise manner, without relying on the original training data. Specifically, our AdaMerging method operates as an automatic, unsupervised task arithmetic scheme. It leverages entropy minimization on unlabeled test samples from the multi-task setup as a surrogate objective function to iteratively refine the merging coefficients of the multiple models. Our experimental findings across eight tasks demonstrate the efficacy of the AdaMerging scheme we put forth. Compared to the current state-of-the-art task arithmetic merging scheme, AdaMerging showcases a remarkable 11% improvement in performance. Notably, AdaMerging also exhibits superior generalization capabilities when applied to unseen downstream tasks. Furthermore, it displays a significantly enhanced robustness to data distribution shifts that may occur during the testing phase. The code is available at AdaMerging.

- The primary challenges in task vector-based MTL lie in determining the appropriate task vector merging coefficients that facilitate the optimal integration of multiple tasks[1]

- Additionally, it is more desirable and flexible to fine-tune different coefficients for different layers within the merged model

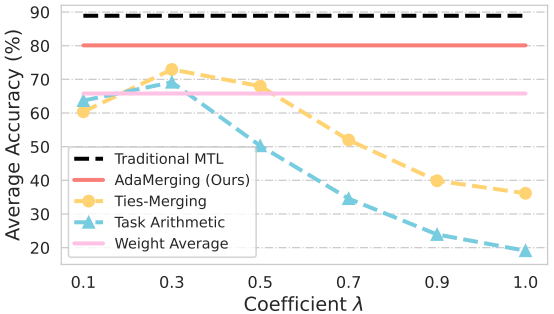
[1]Yang et al. *AdaMerging: Adaptive Model Merging for Multi-Task Learning*. ICLR, 2024

The Role of Coefficients λ_{t_k}

The State of Model Merging

- Task vector-based merging methods perform significantly better than simple weight averaging scheme (pink line)
  - $\theta_M = \frac{1}{K} \sum_{k=1}^{K} \theta_{ft}^{t_k}$

- Task vector-based model merging methods are very sensitive to the coefficient λ[1]



[1]Yang et al. *AdaMerging: Adaptive Model Merging for Multi-Task Learning*. ICLR, 2024

The Role of Coefficients λ_{t_k}

The State of Model Merging

Published as a conference paper at ICLR 2025

MASTERING TASK ARITHMETIC: τJP AS A KEY INDICATOR FOR WEIGHT DISENTANGLEMENT

Kotaro Yoshida[1,*,†], **Yuji Naraki**[2], **Takafumi Horie**[3,*], **Ryosuke Yamaki**[3,4], **Ryotaro Shimizu**[5,6], **Yuki Saito**[5], **Julian McAuley**[6], **Hiroki Naganuma**[7,8,†]
[1] Institute of Science Tokyo, [2] Independent Researcher, [3] Ritsumeikan University, [4] ProPlace, [5] ZOZO Research, [6] University of California San Diego, [7] Université de Montréal, [8] Mila

ABSTRACT

Model-editing techniques using task arithmetic have rapidly gained attention. Through task arithmetic, simply through arithmetic operations on the weights of pre-trained and fine-tuned models create desired models, such as multi-task models, models in which specific tasks are unsolvable, or domain-transferred models. However, task arithmetic faces challenges, such as poor reproducibility and the high cost associated with adjusting coefficients in the arithmetic operations on model parameters, which have limited its practical success. In this paper, we present three key contributions in the context of task addition and task negation within task arithmetic. First, we propose a new metric called τJp which is based on the product of the task vector (τ) and the Jacobian of the pre-trained model with respect to its weights. We show that τJp has a causal relationship with the interference that occurs from arithmetic operations. Second, we show that introducing regularization to minimize τJp significantly mitigates interference between task inference, which leads to the elimination of coefficient tuning and improved accuracy on each task. Third, in the context of incremental learning, we demonstrate that our τJp regularization achieves more robust performance in environments where access to future tasks is unavailable, thus validating the scalability of the approach. Finally, we demonstrate that the τJp regularizer further reinforces the performance of task arithmetic by leveraging publicly available fine-tuned models, offering practical benefits for real-world applications. Our code is available at https://github.com/katoro8989/tau-Jp_Task_Arithmetic

- Yoshida et al.[1] observed two key challenges in model merging: poor reproducibility and the high cost associated with adjusting coefficients in the arithmetic operations

- Building upon the product of the τ (task vector) and the Jacobian of θ_{pre} (pre-trained model), the authors proposed minimizing the τ-Jacobian product to eliminate the need for coefficient tuning

[1] Yoshida et al. *Mastering Task Arithmetic: τJp as a Key Indicator for Weight Disentanglement*. ICLR, 2025

The Role of Coefficients λ_{t_k}

The State of Model Merging

- Multi-task learning (MTL) still performs better than model merging

Method	Coefficient λ	ViT-B-32	ViT-B-16	ViT-L-14
Pre-trained	–	47.3	54.5	65.1
MTL	–	87.8	90.8	92.6
Non-lin. FT	1.0	19.9	19.1	37.6
	Grid-searched	70.4	75.5	84.0
Linear FT	1.0	55.4	58.2	80.5
	Grid-searched	74.3	78.7	85.8
Ties-Merging	1.0	74.2	78.6	85.0
	Grid-searched	74.2	78.6	85.0
AdaMerging	Trained	80.1	84.9	90.8
Ours[1]	1.0	84.2	87.5	90.8
	Grid-searched	84.5	87.6	90.8

[1]Yoshida et al. *Mastering Task Arithmetic: τJp as a Key Indicator for Weight Disentanglement*. ICLR, 2025

Adaptive Model Merging

The State of Model Merging

- Yang et al.[1] propose to assign a separate merging coefficient λ_k to each task t_k (Task-wise)

- The authors also suggest learning a coefficient for each layer
  - Lower layers may learn general features, while the higher layers may learn task-specific features
  - Therefore, the weights of different layers for each task vector should also have different contributions

$$\theta_M = \left\{\theta_M^l\right\}_{l=1}^{L} = \left\{\theta_{\text{pre}}^l + \sum_{k=1}^{K} \lambda_k^l t_k^l\right\}_{l=1}^{L},$$

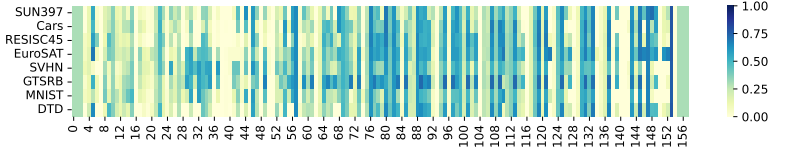
where L represents the number of layers

[1] Yang et al. *AdaMerging: Adaptive Model Merging for Multi-Task Learning*. ICLR, 2024

Adaptive Model Merging

The State of Model Merging

- The coefficients of shallow layers are generally smaller than those of deep layers
  - They rely more on the weights of the pre-trained model than the weights provided by task vectors
- The coefficients of deep layers rely more on the weights provided by the task vectors
- This may be since the shallow layer learns general features, i.e., cross-task, while the deep layer learns task-specific features



Model Merging

The State of Model Merging

Published as a conference paper at ICLR 2025

MODEL MERGING WITH SVD TO TIE THE KNOTS

George Stoica[1]* **Pratik Ramesh**[1]* **Boglarka Ecsedi**[1]
Leshem Choshen[2] **Judy Hoffman**[1]
[1]Georgia Tech [2]IBM Research, MIT
Correspondence emails: {gstoica3,pramesh39}@gatech.edu

ABSTRACT

Recent model merging methods demonstrate that the parameters of fully-finetuned models specializing in distinct tasks can be combined into one model capable of solving all tasks without retraining. Yet, this success does not transfer well when merging LoRA finetuned models. We study this phenomenon and observe that the weights of LoRA finetuned models showcase a lower degree of alignment compared to their fully-finetuned counterparts. We hypothesize that improving this alignment is key to obtaining better LoRA model merges, and propose KnOTS to address this problem. KnOTS uses the SVD to jointly transform the weights of different LoRA models into an aligned space, where existing merging methods can be applied. In addition, we introduce a new benchmark that explicitly evaluates whether merged models are general models. Notably, KnOTS consistently improves LoRA merging by up to 4.3% across several vision and language benchmarks, including our new setting. We release our code at: https://github.com/gstoica27/KnOTS.

1 INTRODUCTION

Model merging (Garipov et al., 2018; Draxler et al., 2018; Wortsman et al., 2022a; Choshen et al., 2022) is an increasingly popular technique that can surprisingly create a single general model by

- Stoica et al.[1] observed that LoRA finetuned models exhibit a lower degree of alignment than their fully-finetuned (FFT) counterparts

[1]Stoica et al. *Model merging with SVD to tie the Knots*. ICLR, 2025

Model Merging

The State of Model Merging

Published as a conference paper at ICLR 2025

MODEL MERGING WITH SVD TO TIE THE KNOTS

George Stoica[1]* **Pratik Ramesh**[1]* **Boglarka Ecsedi**[1]
Leshem Choshen[2] **Judy Hoffman**[1]
[1]Georgia Tech [2]IBM Research, MIT
Correspondence emails: {gstoica3,pramesh39}@gatech.edu

ABSTRACT

Recent model merging methods demonstrate that the parameters of fully-finetuned models specializing in distinct tasks can be combined into one model capable of solving all tasks without retraining. Yet, this success does not transfer well when merging LoRA finetuned models. We study this phenomenon and observe that the weights of LoRA finetuned models showcase a lower degree of alignment compared to their fully-finetuned counterparts. We hypothesize that improving this alignment is key to obtaining better LoRA model merges, and propose KnOTS to address this problem. KnOTS uses the SVD to jointly transform the weights of different LoRA models into an aligned space, where existing merging methods can be applied. In addition, we introduce a new benchmark that explicitly evaluates whether merged models are general models. Notably, KnOTS consistently improves LoRA merging by up to 4.3% across several vision and language benchmarks, including our new setting. We release our code at: https://github.com/gstoica27/KnOTS.

1 INTRODUCTION

Model merging (Garipov et al., 2018; Draxler et al., 2018; Wortsman et al., 2022a; Choshen et al., 2022) is an increasingly popular technique that can surprisingly create a single general model by

- FFT models have very high CKA, indicating that the fine-tuning updates they apply to the respective pretrained weights have high alignment

- LoRA models exhibit considerably lower CKA. This suggests that the task-updates between different LoRA models process inputs through misaligned subspaces

[1]Stoica et al. *Model merging with SVD to tie the Knots*. ICLR, 2025

Model Merging

The State of Model Merging

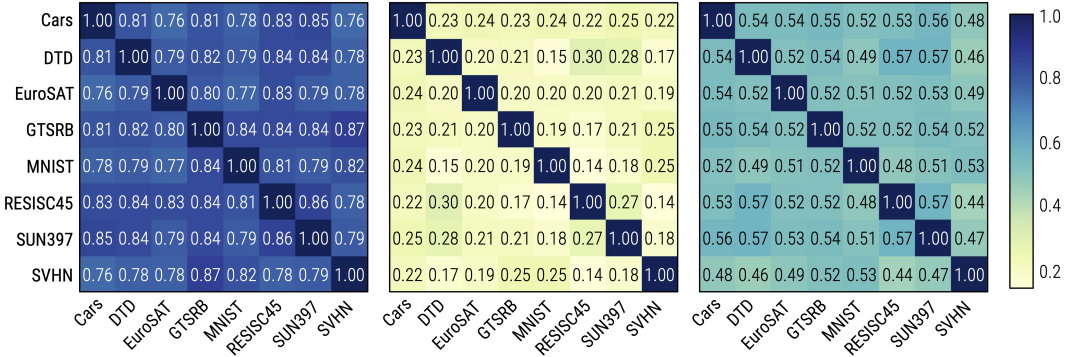


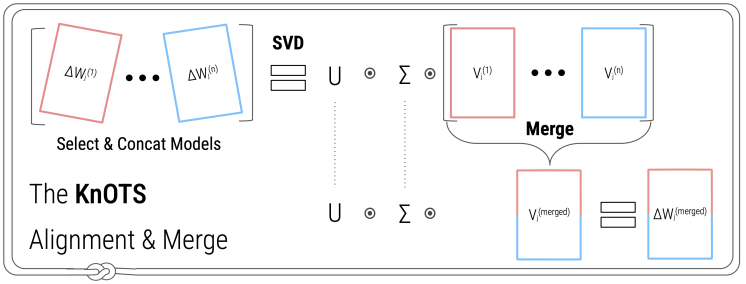
Figure: Left. Full-rank finetuned models alignments. Middle. LoRA finetuned model alignments. Right. LoRA finetuned model alignments with the method by Stoica et al.[1]

[1] Stoica et al. *Model merging with SVD to tie the Knots*. ICLR, 2025

Model Merging

The State of Model Merging

- General idea behind **K**nowledge **o**rientation **T**hrough **S**VD (KnOTS) method by Stoica et al.[1]



[1] Stoica et al. *Model merging with SVD to tie the Knots.* ICLR, 2025