Optimizer Amalgamation

ICLR 2022

Tianshu Huang[1,2], Tianlong Chen[1], Sijia Liu[3], Shiyu Chang[4], Lisa Amini[5], Zhangyang Wang[1]

[1]University of Texas at Austin, [2]Carnegie Mellon University, [3]Michigan State University, [4]University of California, Santa Barbara, [5]MIT-IBM Watson AI Lab, IBM Research

tianshu@cmu.edu, {tianlong.chen,atlaswang}@utexas.edu, liusiji5@msu.edu, chang87@ucsb.edu, lisa.amini@ibm.com

VITA Group

Github Page

Motivation

Meta-training optimizers ("Learning to Optimize" [1]) from scratch is quite difficult. On the other hand, much attention has been directed towards developing a diverse body of analytical optimizers. Can we use these optimizers to meta-train a stronger optimizer?

Contributions

Optimizer Amalgamation We define "Optimizer Amalgamation:" how do we best combine multiple optimizers into a single stronger optimizer?

Mean: add loss ("distillation loss" - l2 distance) for each optimizer

$$\mathcal{L}_{\text{mean}}(\boldsymbol{x};\boldsymbol{\theta}_i;\phi) = \mathcal{L}_{\text{meta}}(\boldsymbol{x};\theta_i^{(P)};\phi) + \alpha\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\boldsymbol{T}|}\sum_{i=1}^{|\boldsymbol{T}|}\log\left\|\theta_i^{(P)} - \theta_i^{(T_i)}\right\|_2$$

Min-max: use loss of the furthest optimizer

$$\mathcal{L}_{\text{min-max}}(\boldsymbol{x};\theta_i;\phi) = \mathcal{L}_{\text{meta}}(\boldsymbol{x};\theta_i^{(P)};\phi) + \alpha\frac{1}{N}\sum_{i=1}^{N}\max_{T\in\boldsymbol{T}}\log\left\|\theta_i^{(P)} - \theta_i^{(T)}\right\|_2$$

Choice: train an intermediate optimizer to choose the best optimizer at each step

$$\mathcal{L}_{\text{choice}} = \mathcal{L}_{\text{meta}}(\phi;\boldsymbol{x}) + \alpha\frac{1}{N}\sum_{i=1}^{n}\log\left\|\theta_i^{(P)} - \theta_i^{(C)}\right\|_2$$

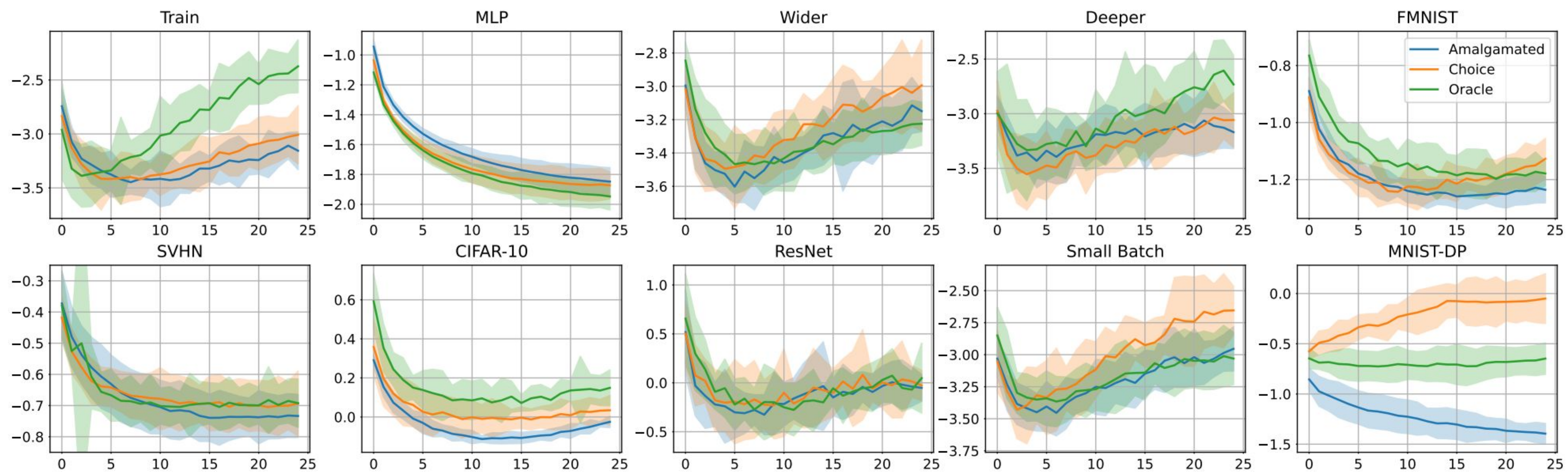
Perturbations We use adversarial and random perturbations to alleviate high variance during training.

Experiments We report extensive experiments evaluating 8 replicates for each method, 10 problems per replicate, and 10 runs per problem.

Related Work

[1] Chen et al, "Learning to Optimize: A Primer and A Benchmark".

[2] Chen et al, "Training Stronger Baselines for Learning to Optimize", NeurIPS 2020



Results

Optimizer Pool We used 6 optimizers in our pool: Adam, Momentum, RMSProp, SGD, AddSign, and PowerSign.

Above Our amalgamated optimizer is consistently as good as the "oracle" optimizer, which is the analytical optimizer with the best performance on each problem, and the "choice" optimizer, which is the optimizer trained as an intermediate step in optimal choice amalgamation.

Below With the same amount of training, our method consistently outperforms existing learning to optimize methods.

Right Appropriately scaled perturbations can significantly reduce variance during training.