Lowering the Pre-training Tax for Gradient-based Subset Training: A Lightweight Distributed Training Toolkit

TEXAS
The University of Texas at Austin

ICML
International Conference On Machine Learning

Yeonju Ro, Zhangyang "Atlas" Wang, Vijay Chidambaram, Aditya Akella
The University of Texas at Austin

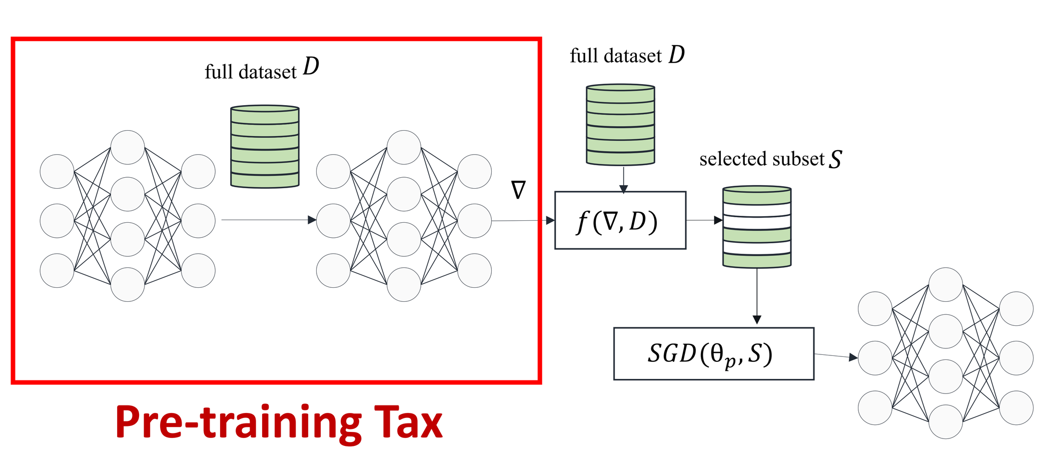
Summary

- We propose a distributed pre-training framework that minimizes the pre-training overhead in subset training.
- We leverage model-soup-inspired ensembling *at initialization* with aggressive augmentation and data-based sparsity to efficiently provide stable and robust gradients for subset selection algorithms.

Gradient-based Subset Training

- With the emergence of billion-parameter-scale models, dataset sizes have also increased accordingly.
- To accelerate training with large-scale datasets, subset training got attention. Using a carefully selected subset, we can train faster without compromising accuracy.
- Recently proposed subset selection algorithms use the initial gradient after pretraining as input to the algorithms.

Pre-training Tax

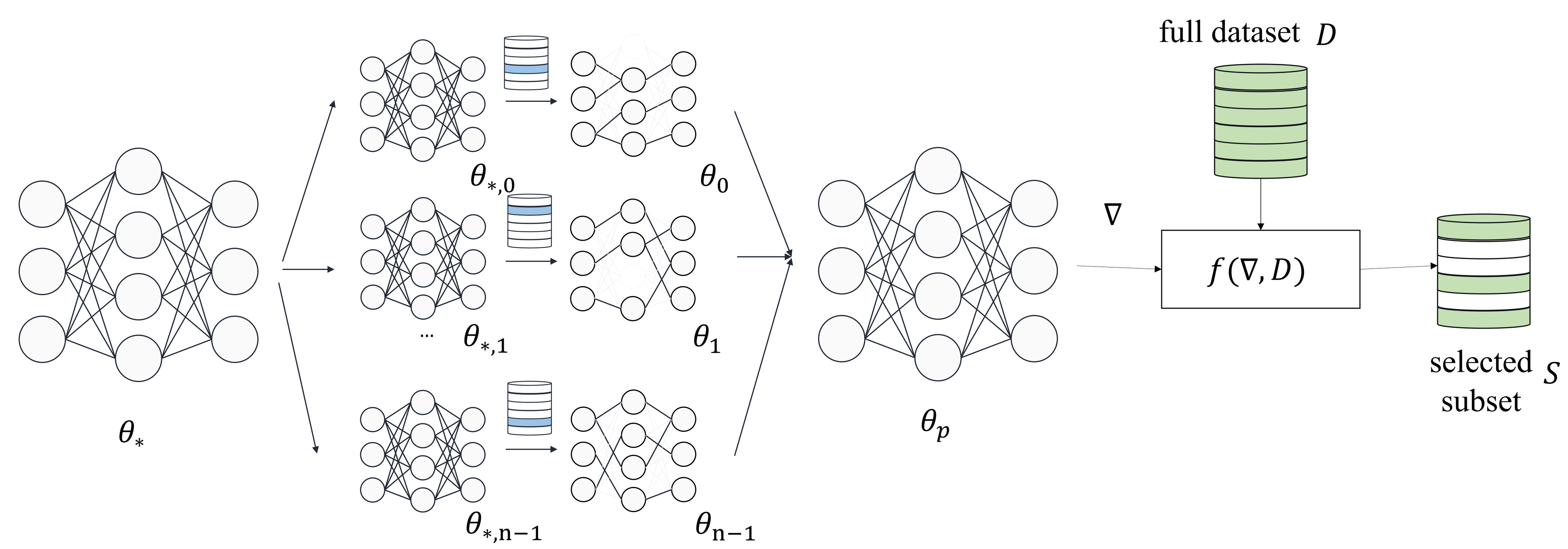


Pre-training Tax

- To get stable and robust gradients, there is a pre-training process with a full dataset, which has non-negligible overhead.
- In prior works, it took 15-40 epochs, which corresponds to 20%-40% of the end-to-end training time.
- We define this pre-training overhead as a ***pre-training tax*** and aim to reduce the pre-training tax in a principled, scalable, and resource-efficient manner.

Our Method

- To make it scalable so it can run in a distributed environment with minimal communication costs. To do that, in our design,
  - Workers do not synchronize nor communicate during the pre-training.
  - We do not ship the full dataset to each worker to reduce communication costs and local training costs at each worker.
- To meet the quality of the pre-trained model, we provide robust and reliable initial gradients for subset selection algorithms.
- Our Method



- Starting from θ_* , we distribute the initial model to different workers with their own random subset that does not overlap each other.
- Each worker do local training with its own set while not communicating with other workers. This can be run in parallel as our workers don't need any synchronization.
- Once local training is done, all models $(\theta_0, \theta_1, \ldots, \theta_{n-1})$ are aggregated with *model averaging*.
- Data Augmentation
  - Since we are using very limited samples for local training, we leverage *random augmentation with stronger magnitude*, to mitigate overfitting (14 policies, with magnitude 9).
- Sparsity
  - We apply data-based sparsity as a regularizer to reduce overfitting while increasing model heterogeneity. We use one-shot magnitude pruning due to its simplicity and low overhead.

Experiments

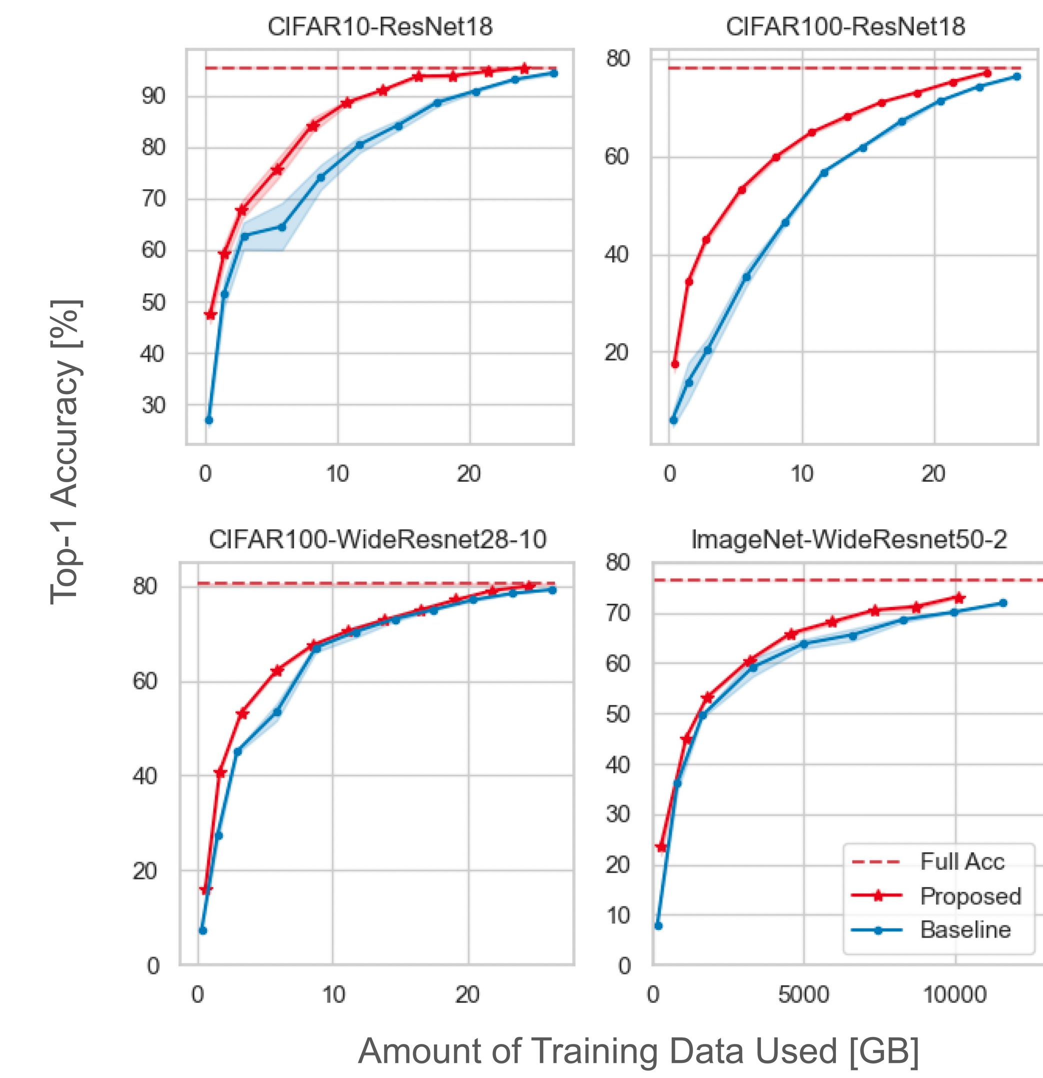
- Ablation Study

Method	Low fraction (10%) ResNet18 CIFAR10	ResNet18 ImageNet	WRN28-10 CIFAR100	WRN50-2 ImageNet	High fraction (70%) ResNet18 CIFAR10	ResNet18 ImageNet	WRN28-10 CIFAR100	WRN50-2 ImageNet
Full acc	95.4	67.7	80.4	76.5	95.4	67.7	80.4	76.5
❶	60.7±5.1	45.0±0.2	38.6±1.0	47.5±0.7	90.5±0.5	63.9±0.2	73.3±0.6	72.0±0.1
❶ + ❷	62.2±3.6	45.2±0.2	38.7±0.9	48.2±0.5	90.7±0.5	64.1±0.1	73.9±0.3	72.4±0.1
❶ + ❸	66.3±2.1	46.2±0.1	43.9±0.4	48.9±0.1	92.6±0.4	64.6±0.0	75.2±0.1	73.1±0.0
❶ + ❷ + ❸	68.5±1.1	46.4±0.2	45.1±0.3	49.4±0.2	93.6±0.3	64.7±0.1	76.4±0.5	73.3±0.0

❶ Model merging ❷ Model pruning ❸ Data Augmentation

Experiments

- Top-1 Accuracy vs. Amount of data used for the training



- Low Fraction Data Improvement

Data Fraction	Glister	This work	Improvement
1%	27.04±1.3	47.50±1.7	+20.45%
5%	51.64±2.7	59.30±1.9	+7.66%
10%	62.75±2.6	67.74±1.8	+4.99%
20%	64.58±4.6	75.65±1.9	+11.07%

- End-to-end Speed Up
  - 2.8x speedup in end-to-end training.
  - 15x reduction in pre-training time.
  - Compared to full training dataset, we reduced 87% while not compromising the accuracy.
- Model Merging Method

Data Fraction	10%	20%	30%	40%	50%
proposed merging, all	68.05%	78.53%	85.76%	87.74%	89.92%
proposed merging, greedy	69.50%	79.39%	88.70%	89.25%	91.18%

- Model Pruning Method

Data Fraction	10%	20%	30%	40%	50%
SNIP	63.76%	69.48%	78.25%	83.02%	86.76%
Random Pruning	62.51%	72.81%	78.62%	83.13%	87.13%
Magnitude-based	66.01%	77.62%	84.21%	86.59%	88.65%