Distribution-aware Data and Model Pruning

Qi Lei

Courant Institute & CDS, NYU

Nov, 2024

with Yijun Dong, Jianwei Li, Xiang Pan, Hoang Phan

https://arxiv.org/abs/2407.06120
https://arxiv.org/abs/2407.19126

Motivation

Why? Growing data and model sizes lead to increasing computational demands in both training and inference time.

Motivation

Why? Growing data and model sizes lead to increasing computational demands in both training and inference time.

What? Want a smaller model and data size:
to save energy, memory, and time without compromising performance.

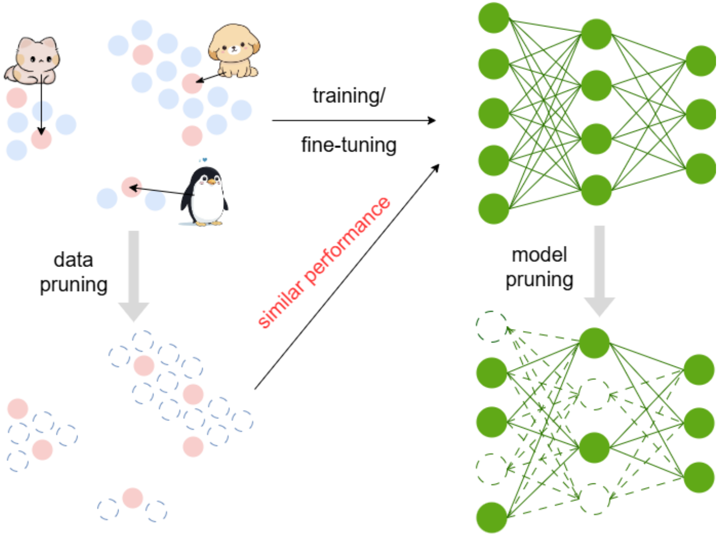
Motivation

Why? Growing data and model sizes lead to increasing computational demands in both training and inference time.

What? Want a smaller model and data size:
to save energy, memory, and time without compromising performance.

How? Need efficient model and data pruning strategies.

Illustration

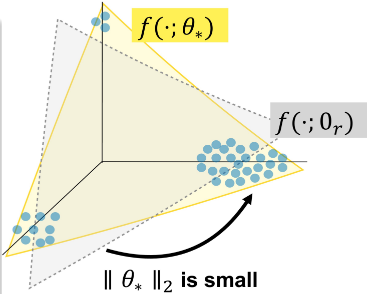


Outline

Data Selection for Finetuning

- Large full dataset $X = [x_1, \cdots, x_N]^\top \subset \mathcal{X}^N$, drawn i.i.d. from unknown distribution
- Finetuning function class $\mathcal{F} = \{f(\cdot;\theta) : \mathcal{X} \to \mathbb{R} \mid \theta \in \Theta\}$ with parameters $\Theta \subset \mathbb{R}^r$
- Pre-trained initialization 0_r (without loss of generality)
- Ground truth $\theta^* \in \Theta$ such that $\mathbb{E}[y|x] = f(x;\theta^*)$ and $\mathbb{V}[y|x] \leq \sigma^2$

- Finetuning dynamics fall in the kernel regime: $f(x;\theta) \approx f(x;0_r) + \nabla_\theta f(x;0_r)^\top \theta$
- With suitable pre-trained initialization (i.e. $f(\cdot, 0_r)$ is close to $f(\cdot, \theta^*)$), $\|\theta^*\|_2$ is small
- Let $G = \nabla_\theta f(X;0_r) \in \mathbb{R}^{N\times r}$ and $G_S = \nabla_\theta f(X_S;0_r) \in \mathbb{R}^{n\times r}$



Data Selection for Finetuning in Kernel Regime

Select a small coreset $(X_S, y_S) \subset \mathcal{X}^n \times \mathbb{R}^n$ of size n indexed by $S \subset [N]$ such that:

$$\theta_S = \arg\min_{\theta \in \Theta} \frac{1}{n} \|G\theta - y_S\|_2^2 + \alpha \|\theta\|_2^2$$

- Low-dimensional data selection: $r \leq n$, $\alpha = 0$ (linear regression)
- High-dimensional data selection: $r > n$, $\alpha > 0$ (ridge regression)

Aim to control excess risk:

$$ER(\theta_S) = \|\theta_S - \theta^*\|_\Sigma^2,$$

where $\Sigma = \mathbb{E}_{x \sim P}[\nabla_\theta f(x; 0_r) \nabla_\theta f(x; 0_r)^\top] \in \mathbb{R}^{r \times r}$

In Low Dimension: Variance Reduction

Consider fixed design for simplicity:

- $\Sigma = \mathbb{E}_{x\sim P}[\nabla_\theta f(x;0_r)\nabla_\theta f(x;0_r)^\top] = G^\top G/N$
- Low-dimensional data selection: $\text{rank}(G_S) = r \leq n$ such that $\Sigma_S = G_S^\top G_S/n \succ 0$

V(ariance)-optimality characterizes generalization:

- $\mathbb{E}[ER(\theta_S)] \leq \frac{\sigma^2}{n}\text{tr}(\Sigma\Sigma_S^{-1})$
- If $\Sigma \preceq c_S\Sigma_S$ for some $c_S \geq \frac{n}{N}$, then $\mathbb{E}[ER(\theta_S)] \leq c_S\sigma^2\frac{r}{n}$

Uniform Sampling Result

Uniform sampling achieves nearly optimal sample complexity in low dimension:

Theorem

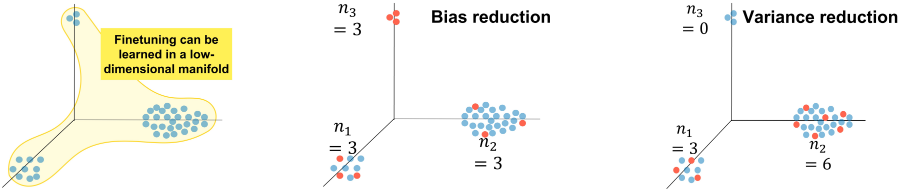
Assuming $\|\nabla_\theta f(\cdot; 0_r)\|_2 \leq B$ *and* $\Sigma \succeq \gamma I_r$. *With probability* $\geq 1-\delta$, X_S *sampled uniformly from* X *satisfies* $\Sigma \preceq c_S \Sigma_S$ *for any* $c_S > 1$ *when*

$$n \gtrsim \frac{B^4}{\gamma^2(1-c_S^{-1})^2}(r + \log(1/\delta))$$

Uniform sampling is near-optimal when $r < n$? What else to expect?

Can the low intrinsic dimension of finetuning be leveraged for high-dimensional data selection ($r > n$)?

Thought Experiment and Prior work



- Bias reduction (low-rank approximation for data matrix): adaptive sampling, k-center greedy
- Variance reduction (V-optimality): uniform sampling, Herding
- Bias-variance trade-off: truncated leverage score, ridge leverage score
- data pruning/selection
  - label-dependent: based on training dynamics
  - label-free: based on geometric properties

With Low Intrinsic Dimension: Variance-Bias Trade-off

- High-dimensional data selection: rank$(G_S) \leq n < r$ such that $\Sigma_S = G_S^\top G_S/n$ is low-rank

Assumption (Low intrinsic dimension)

For $\Sigma = G^\top G/N$, let

$$\mathfrak{r} = \min\{t \in [r] \mid \mathrm{tr}(\Sigma - \langle\Sigma\rangle_t) \leq \mathrm{tr}(\Sigma)/N\}$$

be the intrinsic dimension of the learning problem. Assume $\mathfrak{r} \ll \min\{N, r\}$

- Necessity of low intrinsic dimension: if all r directions in Σ are equally important, $\mathbb{E}[ER(\theta_S)] \gtrsim r - n$

Variance-Bias Tradeoff Theorem

Theorem (Variance-bias tradeoff)

Given a coreset of size S, let $P_{\mathcal{S}}$ be the orthogonal projector onto any subspace $\mathcal{S} \subset \mathit{Range}(\Sigma_S)$, and $P_{\mathcal{S}}^{\perp} = I_r - P_{\mathcal{S}}$. There exists $\alpha > 0$ such that:

$$\mathbb{E}[ER(\theta_S)] \leq \min_{\mathcal{S} \subset \mathit{Range}(\Sigma_S)} \underbrace{\frac{2\sigma^2}{n} \mathit{tr}(\Sigma(P_{\mathcal{S}}\Sigma_S P_{\mathcal{S}})^{\dagger})}_{\text{variance}} + \underbrace{2\mathit{tr}(\Sigma P_{\mathcal{S}}^{\perp}) \|\theta^*\|_2^2}_{\text{bias}}$$

- Variance: excludes the eigen-subspace corresponding to small eigenvalues of Σ_S
- Bias: covers the eigen-subspace corresponding to large eigenvalues Σ

Sample Efficiency

Corollary (Exploitation + exploration)

Given $S \subset [N]$, *for* $\mathcal{S} \subseteq$ *Range*(Σ_S) *with rank*$(P_\mathcal{S}) \approx \mathfrak{r}$, *if:*

- *Variance is controlled by exploiting information in* $\mathcal{S}$*:* $P_\mathcal{S}(c_S \Sigma_S - \Sigma) P_\mathcal{S} \succeq 0$ *for some* $c_S \geq n/N$
- *Bias is controlled by exploring Range*(Σ)*: tr*$(\Sigma P_\mathcal{S}^\perp) \leq \frac{N}{n}$*tr*$(\Sigma - \langle \Sigma \rangle_\mathfrak{r})$

Then,

$$\mathbb{E}[ER(\theta_S)] \leq \textit{variance} + \textit{bias} \lesssim \frac{1}{n}(c_S \sigma^2 \mathfrak{r} + \textit{tr}(\Sigma)\|\theta^*\|_2^2)$$

- **Sample efficiency**: With suitable selection of $S \subset [N]$ the sample complexity of finetuning is linear in the intrinsic dimension $\mathfrak{r}$, independent of the (potentially high) ambient parameter dimension r.

Sample Efficiency

Corollary (Exploitation + exploration)

Given $S \subset [N]$, *for* $\mathcal{S} \subseteq$ *Range*(Σ_S) *with rank*$(P_\mathcal{S}) \approx \mathfrak{r}$, *if:*

- *Variance is controlled by exploiting information in* $\mathcal{S}$*:* $P_\mathcal{S}(c_S\Sigma_S - \Sigma)P_\mathcal{S} \succeq 0$ *for some* $c_S \geq n/N$
- *Bias is controlled by exploring Range*(Σ)*: tr*$(\Sigma P_\mathcal{S}^\perp) \leq \frac{N}{n}$*tr*$(\Sigma - \langle\Sigma\rangle_\mathfrak{r})$

Then,

$$\mathbb{E}[ER(\theta_S)] \leq \textit{variance} + \textit{bias} \lesssim \frac{1}{n}(c_S\sigma^2\mathfrak{r} + \mathit{tr}(\Sigma)\|\theta^*\|_2^2)$$

- **Sample efficiency**: With suitable selection of $S \subset [N]$ the sample complexity of finetuning is linear in the intrinsic dimension $\mathfrak{r}$, independent of the (potentially high) ambient parameter dimension r.
- How to explore the intrinsic low-dimensional structure **efficiently** for data selection?

Gradient Sketching

- Gradient sketching: Randomly projecting the high-dimensional gradients $G = \nabla_\theta f(X; 0_r) \in \mathbb{R}^{N\times r}$ to a lower-dimension $m = O(\mathfrak{r}) \ll r$ via a Johnson-Lindenstrauss transform (JLT)
- Common JLT: a Gaussian random matrix $\Gamma \in \mathbb{R}^{r\times m}$ with i.i.d entries $\Gamma_{ij} \sim \mathcal{N}(0, 1/m)$

Theorem (Gradient sketching)

Under mild conditions, $\tilde{\Sigma}, \tilde{\Sigma}_S \in \mathbb{R}^{m\times m}$ being the sketched covariance of original data and selected data, $m = 11\mathfrak{r}$, there exists $\alpha > 0$ such that:

$$\mathbb{E}[ER(\theta_S)] \lesssim \underbrace{\frac{\sigma^2}{n} tr(\tilde{\Sigma}(\tilde{\Sigma}_S)^\dagger)}_{\text{variance}} + \underbrace{\frac{\sigma^2}{n}\frac{1}{m\gamma_S}\|\tilde{\Sigma}(\tilde{\Sigma}_S)^\dagger\|_2 tr(\Sigma)}_{\text{sketching error}} + \underbrace{\frac{1}{n}\|\tilde{\Sigma}(\tilde{\Sigma}_S)^\dagger\|_2 tr(\Sigma)\|\theta^*\|_2^2}_{\text{bias}}$$

If $\tilde{\Sigma} \le c_S \tilde{\Sigma}_S$ *and* $m = \max\{\sqrt{tr(\Sigma)/\gamma_S}, 11\mathfrak{r}\}$,

$$\mathbb{E}[ER(\theta_S)] \lesssim \frac{c_S}{n}(\sigma^2 m + tr(\Sigma)\|\theta_*\|^2).$$

Sketchy Moment Matching (SkMM)

Gradient sketching

- Draw a (fast) JLT $\Gamma \in \mathbb{R}^{r\times m}$
- Sketch the gradients $\tilde{G} = \nabla_\theta f(X; 0_r)\Gamma \in \mathbb{R}^{N\times m}$

Moment matching

- Spectral decomposition $\tilde{\Sigma} = \tilde{G}^\top \tilde{G}/N = V\Lambda V^\top$
- Initialize $s = [s_1, \dots, s_N]$ with $s_i = 1/n$ for uniformly sampled n
- Sample size-n coreset according to optimization:

$$\min_{s} \min_{\gamma\in\mathbb{R}^m} \sum_{j=1}^{m} (v_j^\top \tilde{G}^\top \mathrm{diag}(s)\tilde{G}v_j - \gamma_j\lambda_j)^2$$

$$\text{s.t. } s \in \Delta_N, \gamma_j \geq 1/c_S \ \forall j \in [m]$$

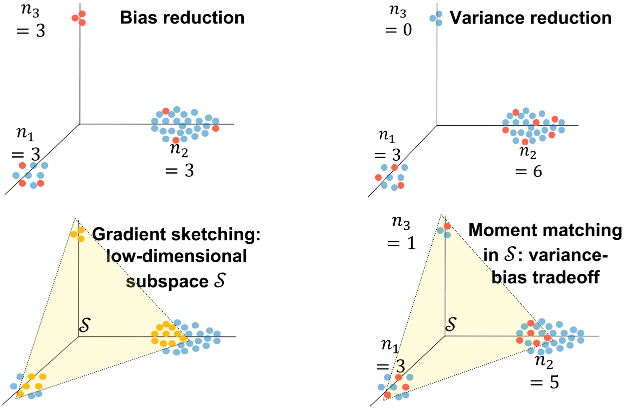
$$\Rightarrow \begin{cases} \text{Relaxation of } 1/c_S\tilde{\Sigma} \lesssim \tilde{\Sigma}_S : \\ \lambda_j/c_S \leq v_j^T \tilde{G}^T \mathrm{diag}(s)\tilde{G}v_j \end{cases}$$

Efficiency of SkMM

Recall $m \ll \min\{N, r\}$:

- Gradient sketching is parallelizable with input-sparsity time:
  - $O(\text{nnz}(G)m)$ for Gaussian embedding
  - $O(\text{nnz}(G)\log m)$ for Fast JLT (sparse sign)
- Moment matching takes:
  - $O(m^3)$ for spectral decomposition
  - $O(Nm)$ per iteration for optimization

SkMM simultaneously controls variance and bias



12

Synthetic Experiments (Regression)

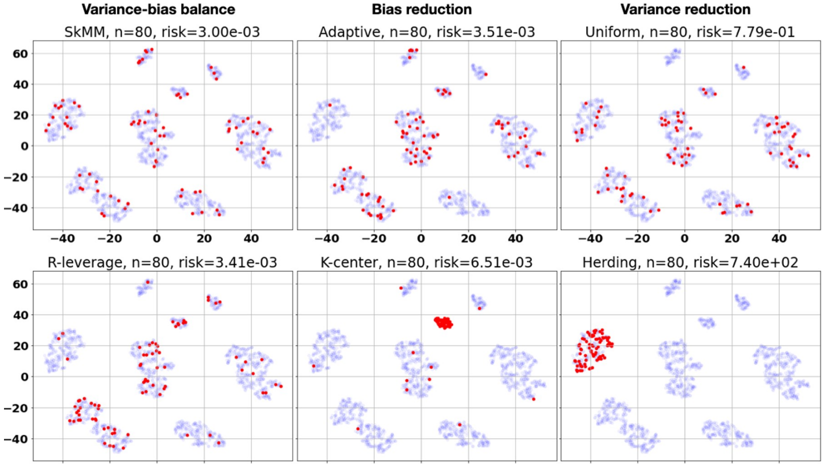
Synthetic Data (Regression)

- Gaussian mixture model (GMM)
- $N = 2000$, $r = 2400 > N$
- Well-separated clusters of random sizes
- Grid search for nearly optimal α

Baselines:

- Herding
- Uniform sampling
- K-center greedy
- Adaptive sampling/random pivoting
- T(runcated)/R(idge) leverage score sampling

Synthetic results



Real Experiments (Classification)

- StanfordCar dataset
- 196 imbalanced classes
- $N = 16,185$ images
- Linear probing: CLIP-pre-trained ViT ($r = 100,548$)
- Last-two-layer finetuning: ImageNet-pre-trained ResNet18 ($r = 2,459,844$)

SkMM for classification: Liner Probing

Table 2: Accuracy and F1 score (%) of LP over CLIP on StanfordCars

	n	2000	2500	3000	3500	4000
Uniform Sampling	Acc	67.63 ± 0.17	70.59 ± 0.19	72.49 ± 0.19	74.16 ± 0.22	75.40 ± 0.16
	F1	64.54 ± 0.18	67.79 ± 0.23	70.00 ± 0.20	71.77 ± 0.23	73.14 ± 0.12
Herding [90]	Acc	67.22 ± 0.16	71.02 ± 0.13	73.17 ± 0.22	74.64 ± 0.18	75.71 ± 0.29
	F1	64.07 ± 0.23	68.28 ± 0.15	70.64 ± 0.28	72.22 ± 0.26	73.26 ± 0.39
Contextual Diversity [1]	Acc	67.64 ± 0.13	70.82 ± 0.23	72.66 ± 0.12	74.46 ± 0.17	75.77 ± 0.12
	F1	64.51 ± 0.17	68.18 ± 0.25	70.05 ± 0.11	72.13 ± 0.15	73.35 ± 0.07
Glister [43]	Acc	67.60 ± 0.24	70.85 ± 0.27	73.07 ± 0.26	74.63 ± 0.21	76.00 ± 0.20
	F1	64.50 ± 0.34	68.07 ± 0.38	70.47 ± 0.35	72.18 ± 0.25	73.69 ± 0.24
GraNd [63]	Acc	67.27 ± 0.07	70.38 ± 0.07	72.56 ± 0.05	74.67 ± 0.06	75.77 ± 0.12
	F1	64.04 ± 0.09	67.48 ± 0.09	69.81 ± 0.08	72.13 ± 0.05	73.44 ± 0.13
Forgetting [79]	Acc	67.59 ± 0.10	70.99 ± 0.05	72.54 ± 0.07	74.81 ± 0.05	75.74 ± 0.01
	F1	64.85 ± 0.13	68.53 ± 0.07	70.30 ± 0.05	72.59 ± 0.04	73.74 ± 0.02
DeepFool [59]	Acc	67.77 ± 0.29	70.73 ± 0.22	73.24 ± 0.22	74.57 ± 0.23	75.71 ± 0.15
	F1	64.16 ± 0.68	68.49 ± 0.53	70.93 ± 0.32	72.44 ± 0.27	73.79 ± 0.15
Entropy [19]	Acc	67.95 ± 0.11	71.00 ± 0.10	73.28 ± 0.10	75.02 ± 0.08	75.82 ± 0.06
	F1	64.55 ± 0.10	67.95 ± 0.12	70.68 ± 0.12	72.46 ± 0.12	73.29 ± 0.04
Margin [19]	Acc	67.53 ± 0.14	71.19 ± 0.09	73.09 ± 0.14	74.66 ± 0.11	75.57 ± 0.13
	F1	64.16 ± 0.15	68.33 ± 0.14	70.37 ± 0.17	72.03 ± 0.11	73.14 ± 0.20
Least Confidence [19]	Acc	67.68 ± 0.11	70.99 ± 0.14	73.04 ± 0.05	74.65 ± 0.09	75.58 ± 0.08
	F1	64.09 ± 0.20	68.03 ± 0.20	70.30 ± 0.07	72.02 ± 0.10	73.15 ± 0.12
SkMM-LP	Acc	$\mathbf{68.27 \pm 0.03}$	$\mathbf{71.53 \pm 0.05}$	$\mathbf{73.61 \pm 0.02}$	$\mathbf{75.12 \pm 0.01}$	$\mathbf{76.34 \pm 0.02}$
	F1	$\mathbf{65.29 \pm 0.03}$	$\mathbf{68.75 \pm 0.06}$	$\mathbf{71.14 \pm 0.03}$	$\mathbf{72.64 \pm 0.02}$	$\mathbf{74.02 \pm 0.10}$

StanfordCar dataset

- 196 imbalanced classes
- $N = 16{,}185$ images

Linear probing (LP)

- CLIP-pre-trained ViT
- $r = 100{,}548$

Last-two-layer finetuning (FT)

- ImageNet-pre-trained ResNet18
- $r = 2{,}459{,}844$

SkMM for Classification: Last-two-layer Finetuning

Table 3: Accuracy and F1 score (%) of FT over (the last two layers of) ResNet18 on StanfordCars

	n	2000	2500	3000	3500	4000
Uniform Sampling	Acc	29.19 ± 0.37	32.83 ± 0.19	35.69 ± 0.35	38.31 ± 0.16	40.35 ± 0.26
	F1	26.14 ± 0.39	29.91 ± 0.16	32.80 ± 0.37	35.38 ± 0.19	37.51 ± 0.23
Herding [90]	Acc	29.19 ± 0.21	32.42 ± 0.16	35.83 ± 0.24	38.30 ± 0.19	40.51 ± 0.19
	F1	25.90 ± 0.24	29.48 ± 0.23	32.89 ± 0.27	35.50 ± 0.22	37.56 ± 0.21
Contextual Diversity [1]	Acc	28.50 ± 0.34	32.66 ± 0.27	35.67 ± 0.32	38.31 ± 0.15	40.53 ± 0.18
	F1	25.65 ± 0.40	29.79 ± 0.29	32.86 ± 0.31	35.55 ± 0.14	37.81 ± 0.23
Glister [43]	Acc	29.16 ± 0.26	32.91 ± 0.19	36.03 ± 0.20	38.16 ± 0.12	40.47 ± 0.16
	F1	26.33 ± 0.19	30.05 ± 0.28	$\mathbf{33.26 \pm 0.18}$	35.41 ± 0.14	37.63 ± 0.17
GraNd [63]	Acc	28.59 ± 0.17	32.67 ± 0.20	35.83 ± 0.16	38.58 ± 0.15	40.70 ± 0.11
	F1	25.66 ± 0.15	29.70 ± 0.22	32.76 ± 0.16	35.72 ± 0.15	37.83 ± 0.11
Forgetting [79]	Acc	28.61 ± 0.31	32.48 ± 0.28	35.18 ± 0.24	37.78 ± 0.22	40.24 ± 0.13
	F1	25.64 ± 0.25	29.58 ± 0.30	32.38 ± 0.20	35.16 ± 0.18	37.41 ± 0.14
DeepFool [59]	Acc	24.97 ± 0.20	29.02 ± 0.17	32.60 ± 0.18	35.59 ± 0.24	38.20 ± 0.22
	F1	22.11 ± 0.11	26.08 ± 0.29	29.83 ± 0.27	32.92 ± 0.33	35.47 ± 0.22
Entropy [19]	Acc	28.87 ± 0.13	32.84 ± 0.20	35.64 ± 0.20	37.96 ± 0.11	40.29 ± 0.27
	F1	25.95 ± 0.17	30.03 ± 0.17	32.85 ± 0.23	35.19 ± 0.12	37.33 ± 0.34
Margin [19]	Acc	29.18 ± 0.12	32.73 ± 0.15	35.67 ± 0.30	38.27 ± 0.20	40.58 ± 0.06
	F1	26.15 ± 0.12	29.66 ± 0.05	32.86 ± 0.30	35.61 ± 0.17	37.77 ± 0.07
Least Confidence [19]	Acc	29.05 ± 0.07	32.88 ± 0.13	35.66 ± 0.18	38.25 ± 0.20	39.91 ± 0.09
	F1	26.18 ± 0.04	30.03 ± 0.14	32.79 ± 0.15	35.42 ± 0.16	37.14 ± 0.12
SkMM-FT	Acc	$\mathbf{29.44 \pm 0.09}$	$\mathbf{33.48 \pm 0.04}$	$\mathbf{36.11 \pm 0.12}$	$\mathbf{39.18 \pm 0.03}$	$\mathbf{41.77 \pm 0.07}$
	F1	$\mathbf{26.71 \pm 0.10}$	$\mathbf{30.75 \pm 0.05}$	33.24 ± 0.05	$\mathbf{36.38 \pm 0.05}$	$\mathbf{39.07 \pm 0.10}$

StanfordCar dataset

- 196 imbalanced classes
- $N = 16{,}185$ images

Linear probing (LP)

- CLIP-pre-trained ViT
- $r = 100{,}548$

Last-two-layer finetuning (FT)

- ImageNet-pre-trained ResNet18
- $r = 2{,}459{,}844$

Conclusion

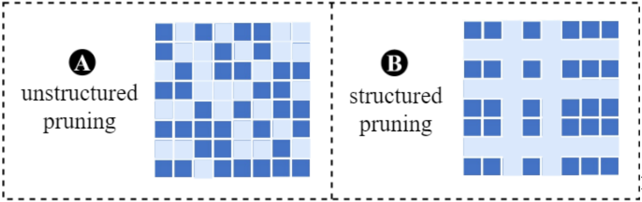
- A rigorous generalization analysis on data selection for fine-tuning
  - Low-dimensional data selection: variance reduction (V-optimality)
  - **High-dimensional data selection**: variance-bias tradeoff
- **Gradient sketching** provably finds a low-dimensional parameter subspace $\mathcal{S}$ with a small bias
  - Reducing variance over $\mathcal{S}$ preserves the fast-rate generalization $O(\dim(\mathcal{S})/n)$
- **SkMM** —a scalable two-stage data selection method for finetuning that simultaneously:
  - Explores the high-dimensional parameter space via gradient sketching
  - Exploits the information in the low-dimensional subspace via moment matching

Future direction: streaming data

Outline

Prior work

Classification 1: model structure preservation



[1]

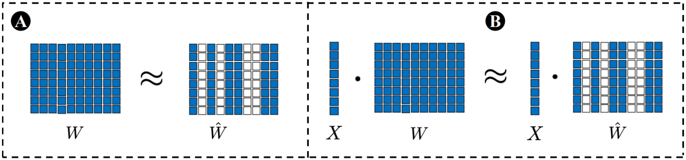
A: better performance preservation

B: hardware compatibility; efficient at inference time

[1] Pruning masks: Dark blue is kept weight; light blue is pruned out weight.

Prior work

Classification 2: approximation principle



A: Preserving model weights

B: Preserving model outputs

Prior work

Classification 3: Retraining requirements (Computational costs)

A: Iterative pruning (High)

B: Finetuning-required pruning (Median)

C: One-shot pruning (Relatively Low)

Value-based $\ll$ Gradient-based $\ll$ Hessian-based

Goal

Iterative pruning	==>	Single-shot pruning
Unstructured pruning	==>	Structured pruning
Gradient/Hessian-based	==>	Value-based pruning
Weight preservation	==>	Output preservation

Goal

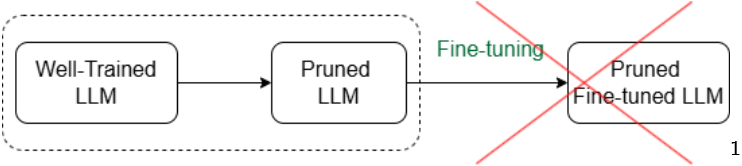
Iterative pruning	==>	Single-shot pruning
Unstructured pruning	==>	Structured pruning
Gradient/Hessian-based	==>	Value-based pruning
Weight preservation	==>	Output preservation

One-shot Pruning

[1]

[1] Concentrate on the effectiveness of the pruning method, instead of comparisons of fine-tuning data's quality.

One-shot Pruning



[1]

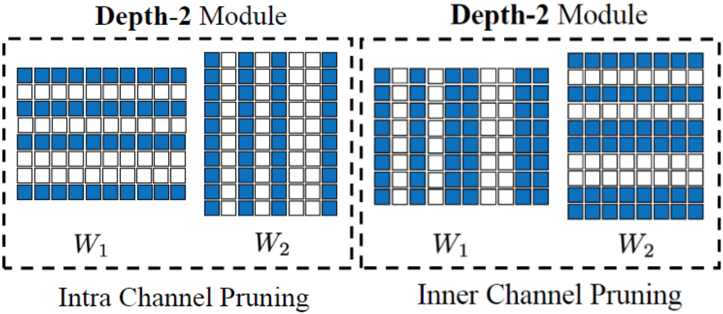
[1]Concentrate on the effectiveness of the pruning method, instead of comparisons of fine-tuning data's quality.

Goal

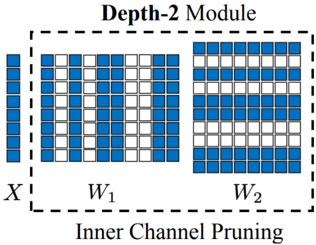
Iterative pruning	==>	Single-shot pruning
Unstructured pruning	==>	Structured pruning
Gradient/Hessian-based	==>	Value-based pruning
Weight preservation	==>	Output preservation

Pruning Unit: Depth-2 Units

Two pruning strategies:



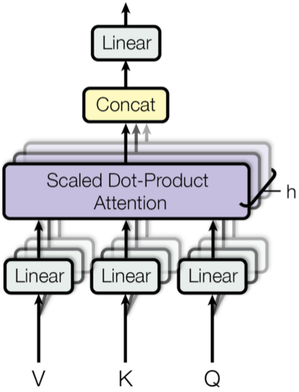
Depth-2 Unit 1: Feedforward Layer



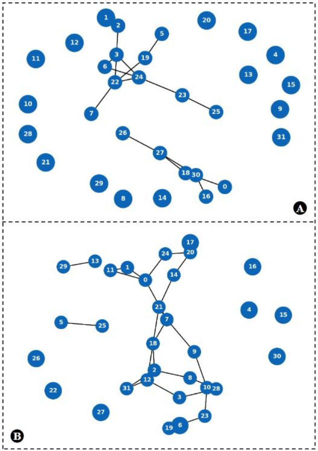
Depth-1 magnitude-based pruning:	$\|(W_1)_{:,i}\|$
Depth-2 magnitude-based pruning:	$\|(W_1)_{:,i}\|\|(W_2)_{i,:}\|$
Ours:	$\|(W_2)_{i,:}\|^2(W_1)_{:,i}^\top\Sigma(W_1)_{:,i}$

Rational: magnitude of each slice $\mathbb{E}[\|(W_2)_{i,:}\|^2\sigma^2((W_1)_{:,i}^\top X)]$
$$= \frac{1}{2}\|(W_2)_{i,:}\|^2(W_1)_{:,i}^\top\Sigma(W_1)_{:,i}.$$
(Take input X as a normal distribution with covariance Σ, σ is ReLU.)

Depth-2 Unit 2: Attention Layer



multi-head attention



32 attention heads from Block 4&5 of Llama-7

Connected if $D(h_i, h_j) \geq 0.2$.

Goal

Iterative pruning	==>	Single-shot pruning
Unstructured pruning	==>	Structured pruning
Gradient/Hessian-based	==>	Value-based pruning
Weight preservation	==>	Output preservation

Layer-wise Recovery

Motivation:

- For gradient-based pruning ==> global criterion ==> $f(\cdot; W + \Delta W) \approx f(\cdot; W) + \nabla_W f(\cdot, W)\Delta W$
- For Value-based pruning ==> local criterion for each layer ==> error will compound layer by layer (if each layer is pruned independently)

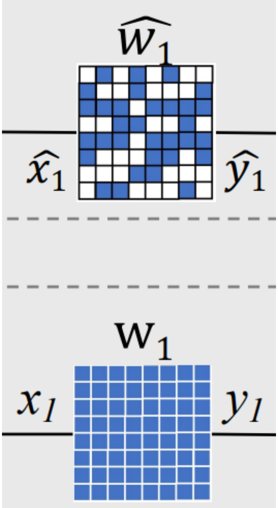
Layer-wise Recovery from Targeted Value

We will apply the above pruning strategy on a recovered weight $\hat{W}_l$:

$$\hat{W}_l \leftarrow \arg\min_{W} \|W\hat{X}_l - Y_l\|,$$

$\hat{X}_l$ is the updated input due to pruned weights $\hat{W}_1, \cdots \hat{W}_{l-1}$, Y_l is the targeted output. [a]

[a] [Li, L, Cheng, Xu, 2023] https://arxiv.org/abs/2310.13191



Results

Methods	WikiText2	PTB↓	BoolQ	PIQA	HS	WG	ARC-e	ARC-c	OBQA	Ave ↑
Dense	12.62	22.14	73.18	78.35	72.99	67.01	67.45	41.38	42.4	63.5
Data Free Pruning										
Random	23.02	40.19	46.21	71.33	59.35	56.51	47.97	32.0	36.30	49.95
L1 norm	179.02	311.75	51.28	60.22	43.14	52.01	36.53	27.89	30.8	43.12
L2 norm	582.41	1022.17	60.18	58.54	37.04	53.27	32.91	27.56	29.8	42.76
Ours	21.76	34.3	63.51	72.63	56.54	54.46	51.68	33.79	36.4	52.72
Ours (RC)	**20.32**	**33.42**	**64.17**	**72.67**	**58.43**	**57.29**	**53.32**	**34.15**	**37.23**	**53.89**
Data Dependent Pruning										
Training-Aware Pruning										
LLM-P.Vec	22.28	41.78	61.44	71.71	57.27	54.22	55.77	33.96	38.4	53.52
LLM-P.E1	19.09	34.21	57.06	75.68	66.8	59.83	60.94	36.52	40.0	56.69
LLM-P.E2	19.77	36.66	59.39	75.57	65.34	61.33	59.18	37.12	39.8	56.82
Inference-Aware Pruning										
Wanda-sp	27.45	49.52	64.16	75.21	**68.62**	62.27	59.68	36.68	39.2	57.97
Ours (Σ)	**17.48**	**30.04**	66.48	75.78	67.73	62.27	61.4	35.49	39.6	58.39
Ours (Σ;RC)	17.90	31.23	**70.12**	**76.86**	68.55	**65.76**	**64.23**	**38.54**	**40.5**	**60.65**
Retraining-required Pruning										
LLM-P. LoRA	17.37	30.39	69.54	76.44	68.11	65.11	63.43	37.88	40.0	60.07

Model: LLaMA-7B (20% sparsity)
First two datasets: zero-shot perplexity (PPL) analysis
Next 7 datasets: zero-shot task classification

Conclusions

- Identifying inherent pruning structure:
  depth-2 units & attention heads
- Designing effective pruning criterion:
  distribution-aware value-based pruning
- Low-computational performance recovery technique:
  avoid error compound

Conclusions

- Identifying inherent pruning structure:
  depth-2 units & attention heads
- Designing effective pruning criterion:
  distribution-aware value-based pruning
- Low-computational performance recovery technique:
  avoid error compound

Data and Model Pruning

- distribution-aware and greedy selection
  - Data pruning: preserving features in the low intrinsic dimension
  - Model pruning: preserve nodes with higher contribution
- no-training required
  - Data pruning: exploring low order statistics of P_X
  - Model pruning: consider input data's distribution

Thank you!