AdaCore:
Adaptive Second Order **Core**sets for Data-efficient Machine Learning

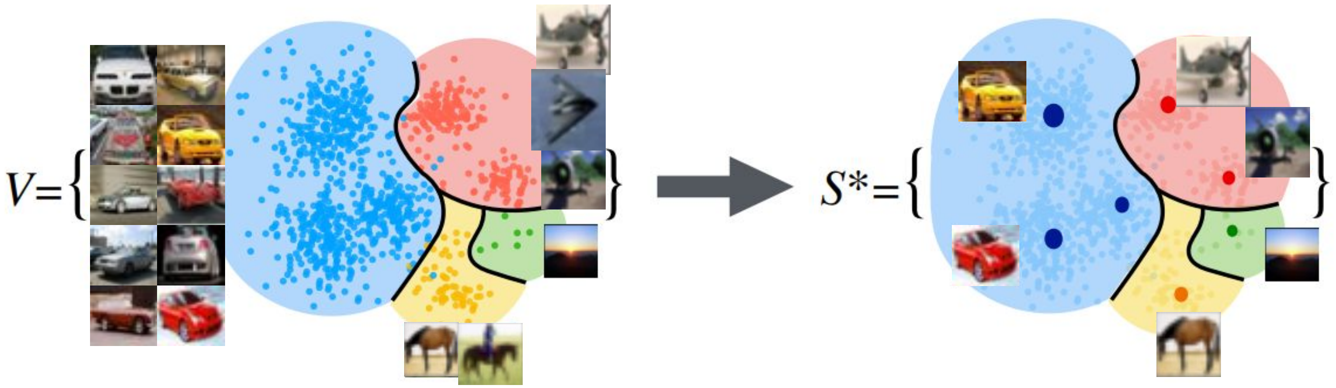
Omead Pooladzandi, David Davini,
Baharan Mirzasoleiman

UCLA Samueli
Electrical & Computer Engineering

Problem Statement

- Training on full dataset can be prohibitively expensive
  - GPT-3 cost $12 Million
  - High Carbon Footprint

- Choose the most salient subset of samples S from full dataset V



M'20

If we can find S* we can speedup training by |V|/|S*| + coreset time by only training on S*

Minimizing Empirical Risk

- Training samples: $\{(x_i, y_i), i \in V\}$

$$w_* \in \arg\min_{w \in \mathcal{W}} \mathcal{L}(w),$$

$$\mathcal{L}(w) := \sum_{i \in V} l_i(w), \quad l_i(w) = l(f(x_i, w), y_i),$$

- convex f(w)
  - logistic regression, regularized support vector machines (SVM), LASSO
- Non-convex f(w)
  - Neural Networks

UCLA Samueli Electrical & Computer Engineering

First Order Subset Selection

- Select subset that covers gradient space
- Provides convergence guarantee for convex model

$$S_t^* = \underset{S \subseteq V, \gamma_{t,j} \geq 0 \ \forall j}{\arg\min} \quad |S| \quad \text{s.t.}$$

$$\|\mathbf{g}_t - \sum_{j \in S} \gamma_{t,j} \mathbf{g}_{t,j}\| \leq \epsilon$$

Issues with First Order Subset Selection

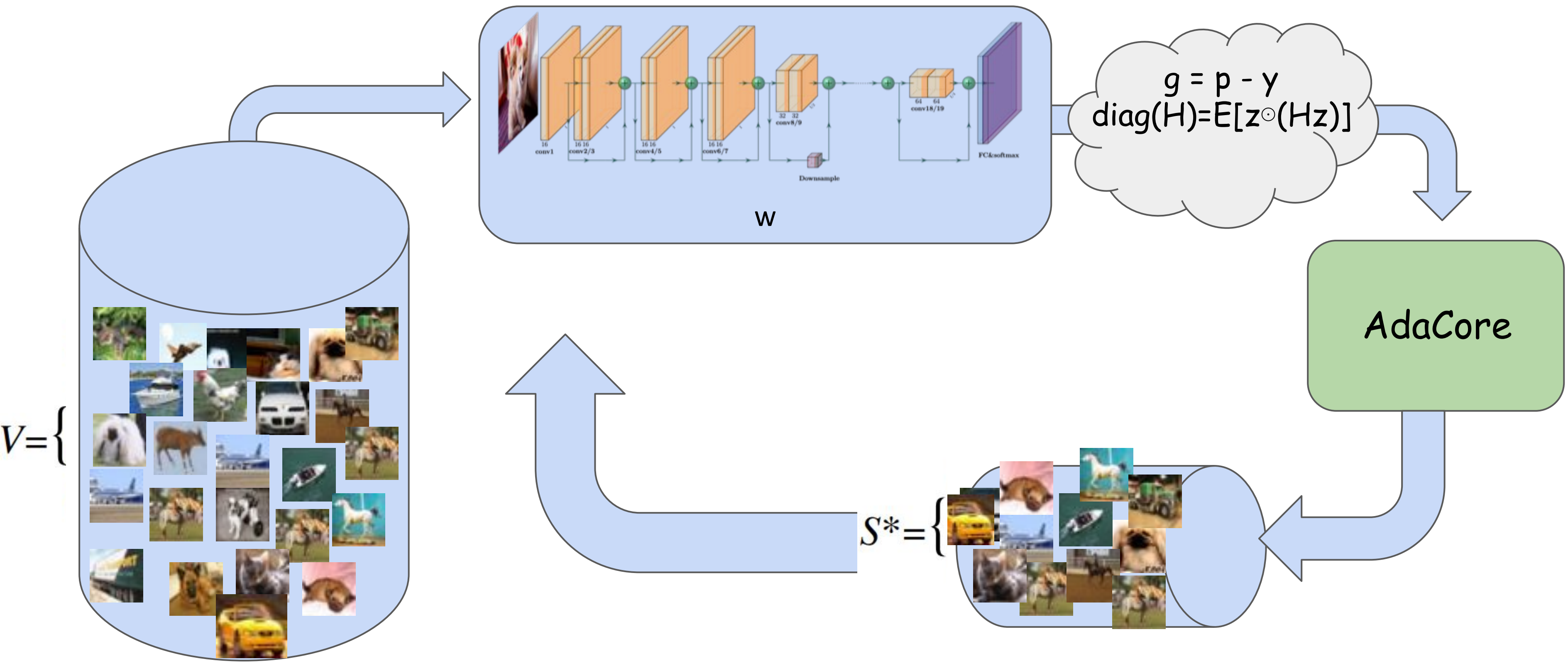
- Scale of g is different along different dimensions
  - subsets estimate full gradient only in dimensions with larger gradient scale
- Loss of many data points may have same gradient
  - One representative example chosen
- Subsets only capturing gradient strongly weight same datapoint
  - Lacks diversity
  - Cannot distinguish different subgroups

- How to boost performance?
  - Extra information to distinguish points with similar gradient

Hessian Preconditioner

- Normalize gradient by hessian inverse information via: $\mathbf{H}^{-1}\mathbf{g}$
- Capture the full gradient in all dimensions equally well
- Contain a more diverse set of datapoints with similar gradients but different curvature properties
- Allow convergence guarantees on corsets trained on with first and second order methods

AdaCore: Extracting Coreset



If we can find S* we can speedup training by |V|/|S*| + coreset time by only training on S*

AdaCore: **Ada**ptive Second Order **Core**sets

- Selecting Subset

$$S_t^* = \underset{S \subseteq V, \gamma_{t,j} \geq 0 \ \forall j}{\arg\min} |S|, \quad \text{s.t.}$$

$$\|\overline{\mathbf{H}}_t^{-1} \overline{\mathbf{g}}_t - \sum_{j \in S} \gamma_{t,j} \overline{\mathbf{H}}_{t,j}^{-1} \overline{\mathbf{g}}_{t,j}\| \leq \epsilon.$$

- Randomized Numerical Linear Algebra (RandNLA):

$$\text{diag}(\mathbf{H}_t) = \mathbb{E}[z \odot (\mathbf{H}_t z)]\|, \quad z \sim Rademacher(0.5)$$

- Smoothened Curvature

$$\overline{\mathbf{H}}_t = \sqrt{\frac{(1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} \text{diag}(\mathbf{H}_i) \text{diag}(\mathbf{H}_i)}{1 - \beta_2^t}}$$

UCLA Samueli Electrical & Computer Engineering

Subset Selection Frequency

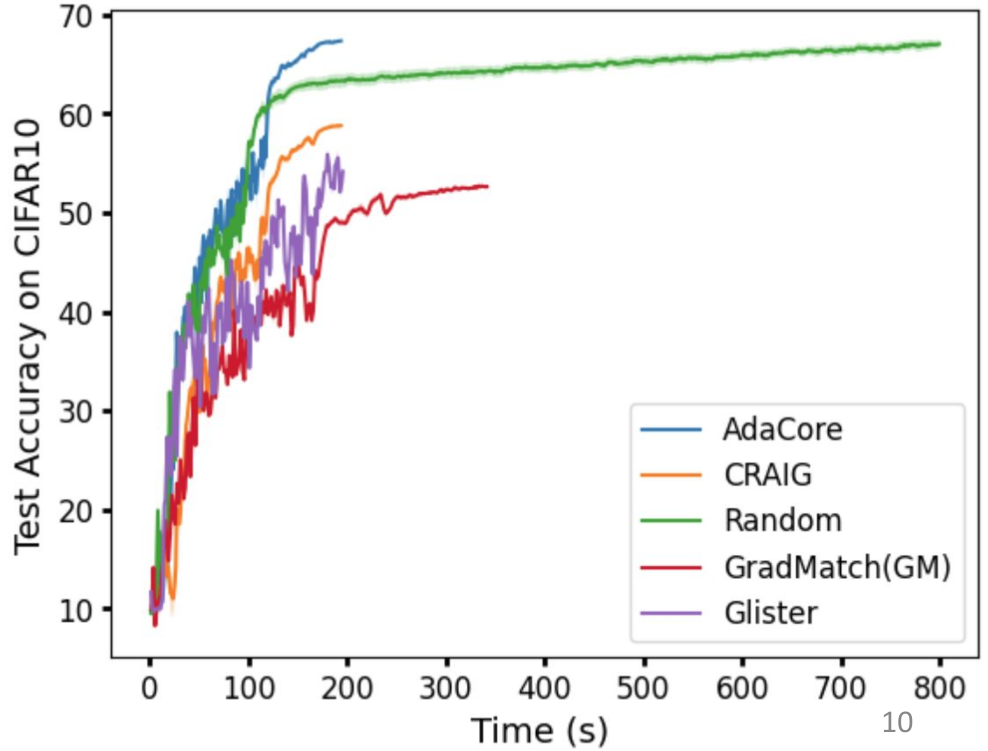
- Convex f(w): logistic regression, regularized support vector machines (SVM), LASSO
  - $$\left\| H_i^{-1} g_i - H_j^{-1} g_j \right\| \leq O(\|w\|) \left\| x_i - x_j \right\|$$
  - Select subset once before training

- Non-Convex f(w)
  - $\left\| H_i^{-1} g_i - H_j^{-1} g_j \right\| \leq O(\|w\|) \| (H_i^{-1} g_i)^{(L)} - (H_j^{-1} g_j)^{(L)} \|$ [KF'19,M'20]
    - Only need to calculate first and second order information of last layer
  - Empirically select subset every R epochs.

Classification Results: CIFAR10

- AdaCore outperforms baseline methods by up to 16.8%
  - ResNet20
- AdaCore visits a smaller percentage of the dataset compared to other methods
- See considerable speedup 2x
  - ResNet18

	SGD+Momentum
Random	45.9% ±2.5(87%)
CRAIG	43.6% ±1.6(75%)
GRADMATCH	49.4% ±1.6(74%)
GLISTER	38.6% ±1.6(74%)
ADACORE	**55.4%** ±1.1(**74%**)



UCLA Samueli Electrical & Computer Engineering

Class Imbalance & Selection Frequency

- One can select a new subset less frequently reducing complexity
- ResNet18 trained on Class imbalanced CIFAR-10
- Subset selected ever R epochs

	S=1%, R=20	S=1%, R=10	S=1%, R=5
AdaCore	**57.3% (5%)**	**57.12 (9.5%)**	**60.2% (14.5%)**
CRAIG	48.6% (8%)	55 (16%)	53.05% (27.5%)
Random	54.7% (8%)	54.6 (18%)	54.6% (33.2%)
GRADM	29.9% (8.2%)	29.1% (14.7%)	32.75% (23.2%)
GLISTER	21.1% (8.6%)	17.2% (16%)	14.4% (22.2%)

Thank you

- Please come to our poster for more information!