Private Adaptive Optimization with Side Information

Tian Li (CMU), Manzil Zaheer (DeepMind), Sashank Reddi (Google Research), Virginia Smith (CMU)

Motivation
Adaptive optimizers (e.g., Adam, AdaGrad, RMSProp) are useful for a variety of ML tasks.
Performance may degrade significantly when trained with differential privacy (DP).

Use side information to approximate the preconditioner

Adaptive optimization methods (e.g., Adam)

1. get gradients on public data: 
   \[
   \hat{g}' = \frac{1}{|B|} \sum_{i \in B} V(f(x_i; w_s), x_i) \in x_{pub}
   \]
2. update preconditioner \( A' \) with recurrence \( \phi(A') = \phi(A^{-1} g') \)

Option 2: Without public data

A's estimated via heuristics (e.g., TF-IDF values or feature frequencies) (optional) maintain a momentum buffer using \( \hat{g}' \)

Convergence Analysis

1. With public data, convex case, RMSProp updates
   \[
   \hat{g}' = \frac{1}{|B|} \sum_{i \in B} \text{clip} \left( \frac{x_{t+1}^i - x_{t+1}}{\sqrt{v_{1, t+1}}}, \sigma \right) \in \mathbb{R}^{d, n}
   \]
   \[\text{A encodes how predictive each coordinate is/estimates of gradient geometry}\]

2. Without public data, fixed preconditioner \( A \), convex case, RMSProp updates
   \[
   \hat{g}' = \frac{1}{|B|} \sum_{i \in B} \text{clip} \left( \frac{x_{t+1}^i - x_{t+1}}{\sqrt{v_{1, t+1}}}, \sigma \right) \in \mathbb{R}^{d, n}
   \]
   One practical choice: feature frequencies for generalized linear models:
   \[
   A = E[|x_i|] + \sigma
   \]

Evaluation

Centralized training: sample-level DP

Without public data

Federated learning: client-level DP

Future Work

Exploring other approaches of reducing noise (e.g., with tree aggregation) in the context of adaptive optimization

Generalizing our approach without public data to arbitrary neural network models

Insights

Use side information to approximate the preconditioner

- Estimate gradient statistics on small public data at each iteration
- Obtained via ‘opt-out’ users or proxy data
- Can be in-distribution or out-of-distribution
- Non-sensitive common knowledge about the training data
  - Easy to obtain before training
  - E.g., token frequencies in NLP
- Useful for both private and non-private training

AdaDPS: Private Adaptive Optimization with Side Information

Option 1: With public data

1. get gradients on public data:
   \[
   \hat{g}' = \frac{1}{|B|} \sum_{i \in B} V(f(x_i; w_s), x_i) \in x_{pub}
   \]
2. update preconditioner \( A' \) with recurrence \( \phi(A') = \phi(A^{-1} g') \)

Option 2: Without public data

A's estimated via heuristics (e.g., TF-IDF values or feature frequencies) (optional) maintain a momentum buffer using \( \hat{g}' \)

Privatize preconditioned gradients (in the simplest form)

\[
\hat{g}' = \frac{1}{|B|} \left( \sum_{i \in B} \text{clip} \left( \frac{x_{t+1}^i - x_{t+1}}{\sqrt{v_{1, t+1}}}, \sigma \right) + \mathcal{N}(0, \sigma^2 C^2) \right)
\]

Performance may degrade significantly when trained with differential privacy (DP)

Convergence Analysis

1. With public data, convex case, RMSProp updates
   \[
   \hat{g}' = \frac{1}{|B|} \sum_{i \in B} \text{clip} \left( \frac{x_{t+1}^i - x_{t+1}}{\sqrt{v_{1, t+1}}}, \sigma \right) \in \mathbb{R}^{d, n}
   \]
   \[\text{A encodes how predictive each coordinate is/estimates of gradient geometry}\]

2. Without public data, fixed preconditioner \( A \), convex case, RMSProp updates
   \[
   \hat{g}' = \frac{1}{|B|} \sum_{i \in B} \text{clip} \left( \frac{x_{t+1}^i - x_{t+1}}{\sqrt{v_{1, t+1}}}, \sigma \right) \in \mathbb{R}^{d, n}
   \]
   One practical choice: feature frequencies for generalized linear models:
   \[
   A = E[|x_i|] + \sigma
   \]

Estimates can be very noisy!

A baseline: directly plug in private gradients to estimate the statistics?

For example (vanilla DP-Adam)

1. first privatize the gradients
   \[
   \bar{g}' = \frac{1}{|B|} \sum_{i \in B} \text{clip} \left( \frac{x_{t+1}^i - x_{t+1}}{\sqrt{v_{1, t+1}}}, \sigma \right) + \mathcal{N}(0, \sigma^2 C^2)
   \]
2. then plug in private gradients to any adaptive optimization methods (e.g., Adam)
   \[
   w_{t+1} = \phi(A'^{-1} g')
   \]

Option 1: With public data

In both convex and non-convex cases, AdaDPS outperforms the baselines significantly.

Convergence Analysis

1. With public data, convex case, RMSProp updates
   \[
   \hat{g}' = \frac{1}{|B|} \sum_{i \in B} \text{clip} \left( \frac{x_{t+1}^i - x_{t+1}}{\sqrt{v_{1, t+1}}}, \sigma \right) \in \mathbb{R}^{d, n}
   \]
   \[\text{A encodes how predictive each coordinate is/estimates of gradient geometry}\]

2. Without public data, fixed preconditioner \( A \), convex case, RMSProp updates
   \[
   \hat{g}' = \frac{1}{|B|} \sum_{i \in B} \text{clip} \left( \frac{x_{t+1}^i - x_{t+1}}{\sqrt{v_{1, t+1}}}, \sigma \right) \in \mathbb{R}^{d, n}
   \]
   One practical choice: feature frequencies for generalized linear models:
   \[
   A = E[|x_i|] + \sigma
   \]

Future Work

Exploring other approaches of reducing noise (e.g., with tree aggregation) in the context of adaptive optimization

Generalizing our approach without public data to arbitrary neural network models

Code: https://github.com/TianH66/AdaDPS