Weekly Presentation
DeltaGrad: Rapid retraining of machine learning models

Yinjun Wu Edgar Dobriban Susan B Davidson

September 29, 2020

Overview

1. Motivation
2. Related Work
3. DeltaGrad
4. Theoretical Results
5. Experimental Results

Motivation

Retaining Problem

Regular Pipeline:

1. Train a ML model from data using a learning algorithm
2. Small change in training data occurs (deletions or additions)
3. Retrain ML model from scratch

Retaining Problem

Regular Pipeline:

1. Train a ML model from data using a learning algorithm
2. Small change in training data occurs (deletions or additions)
3. Retrain ML model from scratch

- Computationally expensive process

Retaining Problem

Regular Pipeline:

1. Train a ML model from data using a learning algorithm
2. Small change in training data occurs (deletions or additions)
3. Retrain ML model from scratch

- Computationally expensive process
- Throws away useful computations from initial training

Retaining Problem

Regular Pipeline:

1. Train a ML model from data using a learning algorithm
2. Small change in training data occurs (deletions or additions)
3. Retrain ML model from scratch

- Computationally expensive process
- Throws away useful computations from initial training

Research Question

Can we retrain models in an efficient manner?

Potential Applications

- **GDPR**: Deletion of private information from public datasets

Potential Applications

- **GDPR**: Deletion of private information from public datasets
- **Continuous Model Updating**: Handle additions, deletions and changes of training samples

Potential Applications

- **GDPR**: Deletion of private information from public datasets
- **Continuous Model Updating**: Handle additions, deletions and changes of training samples
- **Data Valuation**: *Leave One Out* tests to find important training samples

Potential Applications

- **GDPR**: Deletion of private information from public datasets
- **Continuous Model Updating**: Handle additions, deletions and changes of training samples
- **Data Valuation**: *Leave One Out* tests to find important training samples
- **Bias Reduction**: Speeds up jackknife resampling that requires retrained model parameters

Related Work

Prior Work

- Prior work for specialized problems and ML models, usually for deletion
  - Provenane Based deletions for linear and logistic regression [WTD20]
  - Newton step and noise for *certified data removal* [GGHv20]
  - K-means clustering [GGVZ19]

DeltaGrad

Gradient Descent

- Objective function

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} F_i(\mathbf{w})$$

- Stochastic Gradient Descent update rule, $\mathcal{B}_t$ is randomly sampled mini-batch of size B

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \frac{\eta_t}{B}\sum_{i\in\mathcal{B}_t} \nabla F_i\left(\mathbf{w}_t\right)$$

- Full-batch gradient descent (GD) is on entire data

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \frac{\eta_t}{n}\sum_{i=1}^{n} \nabla F_i\left(\mathbf{w}_t\right)$$

Removal of data

- After training, $R = \{i_1, i_2, \ldots, i_r\}$ is removed, where $r \ll n$
- Naive retraining is applying GD over remaining samples, $\mathbf{w}^U$ is resulting parameters

$$\mathbf{w}^U{}_{t+1} \leftarrow \mathbf{w}^U{}_t - \frac{\eta_t}{n-r} \sum_{i \notin R} \nabla F_i \left(\mathbf{w}^U{}_t \right) \tag{1}$$

- The explicit gradient computation $\sum_{i \notin R} \nabla F_i \left(\mathbf{w}^U{}_t \right)$ is expensive
- Instead rewrite (1) as follows

$$\mathbf{w}^U{}_{t+1} = \mathbf{w}^U{}_t - \frac{\eta_t}{n-r} \left[\sum_{i=1}^{n} \nabla F_i \left(\mathbf{w}^U{}_t \right) - \sum_{i \in R} \nabla F_i \left(\mathbf{w}^U{}_t \right) \right]. \tag{2}$$

- $\sum_{i \in R} \nabla F_i \left(\mathbf{w}^U{}_t \right)$ is cheaper to compute

Etymology

- After a small change to the data we need to redo the SGD computations
- We can achieve this by understanding the small *delta* of the Gradient Descent

$$\nabla F(\mathbf{w}) = \sum_{i=1}^{n} \nabla F_i(\mathbf{w}_t) \quad \& \quad \nabla F(\mathbf{w}^U) = \sum_{i=1}^{n} \nabla F_i\left(\mathbf{w}^U{}_t\right)$$

- Hence, the approach is called *DeltaGrad*

Aprroximating $\nabla F(\mathbf{w}^U)$

- $\mathbf{w}_0, \ldots, \mathbf{w}_t$ and $\nabla F(\mathbf{w}_0), \ldots, \nabla F(\mathbf{w}_t)$ are cached from training on initial dataset
- By Cauchy mean-value theorem[1]

$$\nabla F({\mathbf{w}^U}_t) - \nabla F(\mathbf{w}_t) = \mathbf{H}_t \cdot ({\mathbf{w}^U}_t - \mathbf{w}_t)$$

  Where $\mathbf{H}_t = \int_0^1 \mathbf{H}(\mathbf{w}_t + x({\mathbf{w}^U}_t - \mathbf{w}_t))dx$ is the integrated hessian
- This requires a hessian $\mathbf{H}_t$ at each step, which is expensive to maintain and evaluate
- Leverage classical L-BFGS algorithm to approximate $\mathbf{H}_t$

Aprroximating $\nabla F(\mathbf{w}^U)$

- $\mathbf{w}_0, \ldots, \mathbf{w}_t$ and $\nabla F(\mathbf{w}_0), \ldots, \nabla F(\mathbf{w}_t)$ are cached from training on initial dataset
- By Cauchy mean-value theorem[1]

$$\nabla F(\mathbf{w}^U{}_t) - \nabla F(\mathbf{w}_t) = \mathbf{H}_t \cdot (\mathbf{w}^U{}_t - \mathbf{w}_t)$$

  Where $\mathbf{H}_t = \int_0^1 \mathbf{H}(\mathbf{w}_t + x(\mathbf{w}^U{}_t - \mathbf{w}_t))dx$ is the integrated hessian
- This requires a hessian $\mathbf{H}_t$ at each step, which is expensive to maintain and evaluate
- Leverage classical L-BFGS algorithm to approximate $\mathbf{H}_t$

[1] Seems to be a consequence of Fundamental theory of Calculus and mean-value theorem

Theoretical Results

Experimental Results

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten.
Certified Data Removal from Machine Learning Models.
arXiv:1911.03030 [cs, stat], August 2020.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou.
Making AI Forget You: Data Deletion in Machine Learning.
In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3518–3531. Curran Associates, Inc., 2019.

Yinjun Wu, Val Tannen, and Susan B. Davidson.
PrIU: A Provenance-Based Approach for Incrementally Updating Regression Models.
In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 447–462, Portland OR USA, June 2020. ACM.

Large Deletions

Large Deletions