

Privacy-Preserving Deep Learning

Reza Shokri¹ Vitaly Shmatikov²

¹The University of Texas at Austin

²Cornell Tech

CCS, 2015

Presenter: Bargav Jayaraman

- 1 Introduction
 - Need for Data Privacy
 - Multi-Party Private Learning
- 2 Related Work
- 3 Proposed Approach
 - Key Idea - Distributed Selective SGD
 - System Architecture
 - Local Training
 - Parameter Server
- 4 Experimental Evaluation
- 5 Privacy Discussion
- 6 Conclusion

- 1 Introduction
 - Need for Data Privacy
 - Multi-Party Private Learning
- 2 Related Work
- 3 Proposed Approach
 - Key Idea - Distributed Selective SGD
 - System Architecture
 - Local Training
 - Parameter Server
- 4 Experimental Evaluation
- 5 Privacy Discussion
- 6 Conclusion

Need for Data Privacy

Centralized collection of photos, speech and video by companies from users has many privacy risks:

- 1 Companies keep the data forever; users have no control over it.
- 2 Images and voice recordings may contain sensitive items - faces, license plates, computer screens, etc.
- 3 Users data is subject to subpoenas and warrants, as well as warrantless spying by national-security and intelligence outfits.

- 1 Introduction
 - Need for Data Privacy
 - Multi-Party Private Learning
- 2 Related Work
- 3 Proposed Approach
 - Key Idea - Distributed Selective SGD
 - System Architecture
 - Local Training
 - Parameter Server
- 4 Experimental Evaluation
- 5 Privacy Discussion
- 6 Conclusion

Multi-Party Private Learning

- Sharing of data about individuals is not permitted by law or regulation in medical domain.
- Biomedical and clinical researchers are thus restricted to perform learning on the datasets belonging to their own institutions.
 - Data might be homogeneous, leading to biased local model
- This restricts the performance of deep learning models which rely on large scale data.

Existing private machine learning algorithms aim to achieve:

- 1 Privacy of data or input to the model - Schemes based on Secure Multi-party Computation (SMC) to protect the intermediate computations. Used for decision trees, Naive Bayes models, k-means clustering, etc.
- 2 Privacy of model parameters - One party holds the private model and the other party holds the data. Cryptographic techniques are applied for secure evaluation of the private model on the data.
- 3 Privacy of the model's output - Differential Privacy has been applied for private machine learning of SVM, logistic and linear regression, etc.

Outline

- 1 Introduction
 - Need for Data Privacy
 - Multi-Party Private Learning
- 2 Related Work
- 3 Proposed Approach
 - Key Idea - Distributed Selective SGD
 - System Architecture
 - Local Training
 - Parameter Server
- 4 Experimental Evaluation
- 5 Privacy Discussion
- 6 Conclusion

Key Idea - Distributed Selective SGD

Distributed Selective SGD (DSSGD) has the following assumptions:

- 1 Updates to different parameters during gradient descent are inherently independent
- 2 Different training datasets contribute to different parameters
- 3 Different features do not contribute equally to the objective function

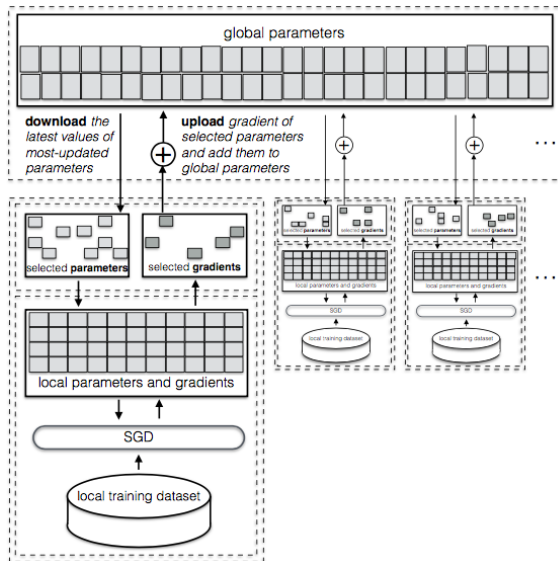
The overall procedure of DSSGD is given as:

- 1 Each party downloads a subset of global model parameters from the server and updates its local model
- 2 Updated local model is trained on the private data
- 3 Subset of gradients are uploaded back to server which updates the global model

Outline

- 1 Introduction
 - Need for Data Privacy
 - Multi-Party Private Learning
- 2 Related Work
- 3 Proposed Approach**
 - Key Idea - Distributed Selective SGD
 - System Architecture**
 - Local Training
 - Parameter Server
- 4 Experimental Evaluation
- 5 Privacy Discussion
- 6 Conclusion

System Architecture



Outline

- 1 Introduction
 - Need for Data Privacy
 - Multi-Party Private Learning
- 2 Related Work
- 3 **Proposed Approach**
 - Key Idea - Distributed Selective SGD
 - System Architecture
 - **Local Training**
 - Parameter Server
- 4 Experimental Evaluation
- 5 Privacy Discussion
- 6 Conclusion

Choose initial parameters $\mathbf{w}^{(i)}$ and learning rate α .

Repeat until an approximate minimum is obtained:

1. Download $\theta_d \times |\mathbf{w}^{(i)}|$ parameters from server and replace the corresponding local parameters.
2. Run SGD on the local dataset and update the local parameters $\mathbf{w}^{(i)}$ according to (1). $w_j = w_j - \alpha \partial E_i / \partial w_j$
3. Compute gradient vector $\Delta \mathbf{w}^{(i)}$ which is the vector of changes in all local parameters due to SGD.
4. Upload $\Delta \mathbf{w}_S^{(i)}$ to the parameter server, where S is the set of indices of at most $\theta_u \times |\mathbf{w}^{(i)}|$ gradients that are selected according to one of the following criteria:
 - *largest values*: Sort gradients in $\Delta \mathbf{w}^{(i)}$ and upload θ_u fraction of them, starting from the biggest.
 - *random with threshold*: Randomly subsample the gradients whose value is above threshold τ .

The selection criterion is fixed for the entire training.

Outline

- 1 Introduction
 - Need for Data Privacy
 - Multi-Party Private Learning
- 2 Related Work
- 3 Proposed Approach**
 - Key Idea - Distributed Selective SGD
 - System Architecture
 - Local Training
 - Parameter Server**
- 4 Experimental Evaluation
- 5 Privacy Discussion
- 6 Conclusion

Choose initial global parameters $\mathbf{w}^{(global)}$.

Set vector **stat** to all zero.

EVENT: A participant uploads gradients $\Delta\mathbf{w}_S$.

- For all $j \in S$:
 - Set $\mathbf{w}^{(global)} := \mathbf{w}^{(global)} + \Delta\mathbf{w}_j$
 - Set $stat_j := stat_j + 1$

EVENT: A participant downloads θ parameters.

- Sort **stat**, and let I_θ be the set of indices for **stat** elements with largest values.
- Send $\mathbf{w}_{I_\theta}^{(global)}$ to the participant.

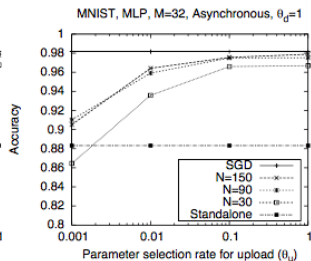
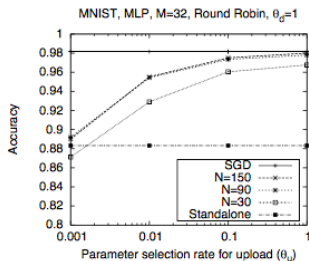
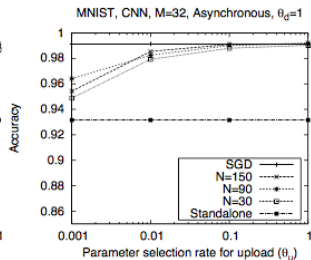
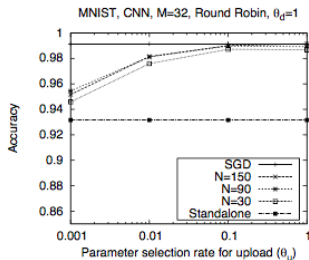
Experimental Setup

- Evaluation on two benchmark datasets:
 - MNIST handwritten digit recognition - 60,000 train and 10,000 test
 - Google's Street View House Numbers (SVHN) - 100,000 train and 10,000 test
- Datasets are normalized by subtracting the average and dividing by the standard deviation of data samples in their training sets.
- Network architectures:
 - MLP - 140,106 for MNIST and 402,250 for SVHN
 - CNN - 105,506 for MNIST and 313,546 for SVHN
- Number of participants $N \in \{30, 90, 150\}$
- Fraction of parameters selected for sharing $\theta_d \in \{1, 0.1, 0.01, 0.001\}$
- Gradient upload criteria: a) largest value b) random with thresholding

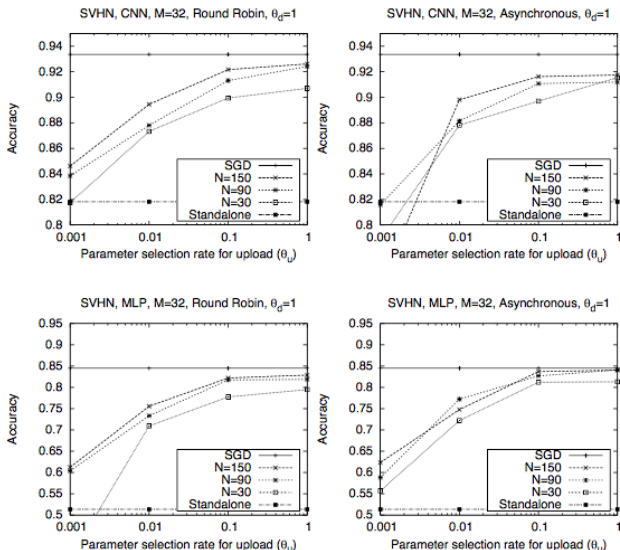
Overall Accuracy of DSSGD with Varying θ_d

| | SGD | 0.1 | 0.01 | 0.001 | Standalone |
|------------|--------|--------|--------|--------|------------|
| MNIST, CNN | 0.9917 | 0.9914 | 0.9871 | 0.9645 | 0.9316 |
| SVHN, CNN | 0.9299 | 0.9312 | 0.8986 | 0.7481 | 0.8182 |
| | SGD | 0.1 | 0.01 | 0.001 | Standalone |
| MNIST, MLP | 0.9810 | 0.98 | 0.9707 | 0.9171 | 0.8832 |
| SVHN, MLP | 0.8476 | 0.8394 | 0.7833 | 0.6542 | 0.5136 |

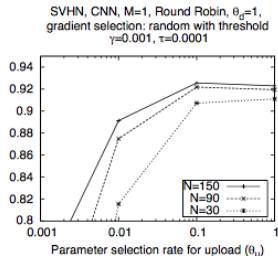
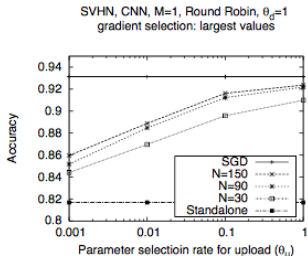
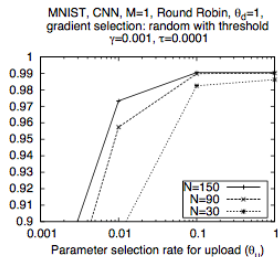
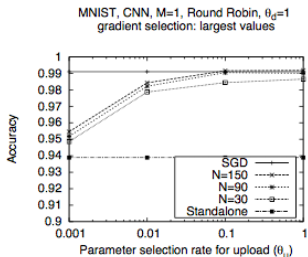
Accuracy of DSSGD on MNIST Dataset



Accuracy of DSSGD on SVHN Dataset



Accuracy of DSSGD for Different Gradient Selection



- Preventing direct leakage
 - while training the model - participants do not reveal their data to other parties during training
 - while using the model - participants can use the learned model locally without any communication with other parties
- Preventing indirect leakage with Differential Privacy - noise is added to gradients to prevent leakage of information related to local dataset

Differential Privacy

For any two datasets D and D' differing in a single item and any output O of function f , the function is differentially private if:

$$\Pr\{f(D) \in O\} \leq \exp(\epsilon) \cdot \Pr\{f(D') \in O\}$$

- In short, a computation is differentially private if the probability of producing a given output does not depend very much on whether a particular data point is included in the dataset.
- Differential privacy can be guaranteed by adding Laplace noise to the output of f proportional to its sensitivity.

The (global) sensitivity of f is:

$$\delta f = \max_{D, D'} \|f(D) - f(D')\|$$

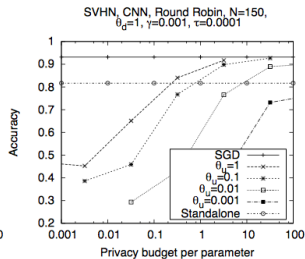
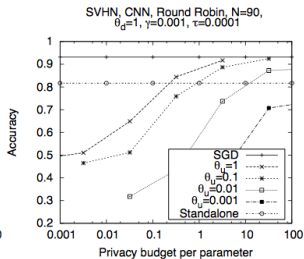
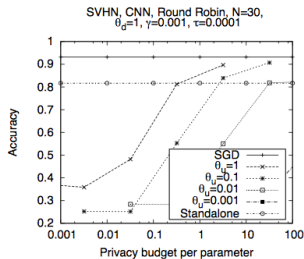
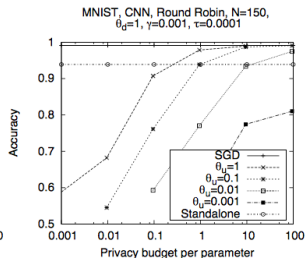
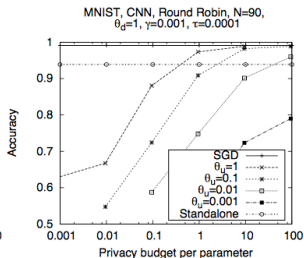
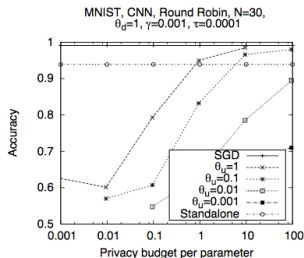
Differential Privacy in DSSGD

- Here, f computes gradients and selects which of them to share with other participants.
- There are two sources of potential leakage: how gradients are selected for sharing and the actual values of the shared gradients.
- Sparse vector technique is used to (i) randomly select a small subset of gradients whose values are above a threshold, and to (ii) share perturbed values of the selected gradients, all under a consistent differentially private mechanism.

Differentially Private DSSGD

- Let ϵ be the total privacy budget for one epoch of participant i running DSSGD, and let Δf be the sensitivity of each gradient
 - Let $c = \theta_u |\Delta \mathbf{w}|$ be the maximum number of gradients that can be uploaded in one epoch
 - Let $\epsilon_1 = \frac{8}{9} \epsilon$, $\epsilon_2 = \frac{2}{9} \epsilon$
 - Let $\sigma(x) = \frac{2c\Delta f}{x}$
1. Generate fresh random noise $r_\tau \sim \text{Lap}(\sigma(\epsilon_1))$
 2. Randomly select a gradient $\Delta w_j^{(i)}$
 3. Generate fresh random noise $r_w \sim \text{Lap}(2\sigma(\epsilon_1))$
 4. If $\text{abs}(\text{bound}(\Delta w_j^{(i)}, \gamma)) + r_w \geq \tau + r_\tau$, then
 - (a) Generate fresh random noise $r'_w \sim \text{Lap}(\sigma(\epsilon_2))$
 - (b) Upload $\text{bound}(\Delta w_j^{(i)} + r'_w, \gamma)$ to the parameter server
 - (c) Charge $\frac{\epsilon}{c}$ to the privacy budget
 - (d) If number of uploaded gradients is equal to c , then Halt
Else Goto Step 1
 5. Else Goto Step 2

Accuracy of Differentially Private DSSGD



Conclusion

- New distributed training technique is proposed, based on selective stochastic gradient descent.
- Works for any type of neural network and preserves privacy of participants training data without sacrificing the accuracy of the resulting models.