SGD on Random Mixtures: Private Machine Learning under Data Breach Threats

Kangwook Lee
KAIST
Daejeon, Korea
kw1jjang@kaist.ac.kr

Kyungmin Lee[⋆]
KAIST
Daejeon, Korea
atm13579@kaist.ac.kr

Hoon Kim[⋆]
KAIST
Daejeon, Korea
gnsrla12@kaist.ac.kr

Changho Suh
KAIST
Daejeon, Korea
chsuh@kaist.ac.kr

Kannan Ramchandran
UC Berkeley
Berkeley, CA
kannanr@eecs.berkeley.edu

ABSTRACT

In this work, we propose Stochastic Gradient Descent on Random Mixtures (SGDRM) as a simple way of protecting data under data breach threats. We prove that SGDRM converges to a critical point for the least-squares problem and for deep neural networks with linear activations. We also conduct extensive experiments, and observe that SGDRM can be applied to general deep learning tasks as well.

1 INTRODUCTION

In a wide variety of machine learning problems, the training dataset consists of sensitive data such as medical records, personal photos, or proprietary data. In such applications, one needs to train a machine learning model without revealing information about the dataset [2]. In the literature, various types of attacks have been studied such as black-box attacks (attackers can use the trained model) and white-box attacks (attackers can see the internals of the model).

In this work, we consider a stronger attack scenario, which we dub the 'data-breach attack' model. Under the data-breach attack model, the attacker has access to the input dataset that is being fed to the training algorithm (see below for the rationale behind the model). Hence, in order to protect the original dataset, one needs to transform the original dataset into a different form, and provide the training algorithm with the transformed dataset. Our solution, which we call *SGDRM (Stochastic Gradient Descent on Random Mixtures)*, is a simple solution to protect sensitive datasets against such attack. The key idea is simple: We run SGD algorithm on random mixtures of the training data points. See Fig. 1 for sample mixtures.

Applications: The 'data-breach attack' model is motivated by the following two applications. The first application is 'machine learning on the cloud'. Consider an agent that wants to train a machine learning model with the sensitive dataset on the cloud. One natural approach is to upload the dataset to the cloud and to run an appropriate machine learning algorithm on it over the cloud. However, if private access to the cloud is compromised or the cloud infrastructure has some security vulnerability [14], the entire dataset is subject to data breach. Hence, the process of learning on the cloud can be viewed as subject to data-breach attacks. Another scenario is where a learner wants to train a model using crowdsourced datasets. If the learner can be fully trusted by the participating users, they can simply share their own private data with the learner but this is usually not the case. Instead, the learner is often treated as being under potential data-breach attack. In order to tackle this problem, Federated learning [10] asks the participants to compute partial gradients with respect to their own private data points, instead of asking them to share their own private data points.

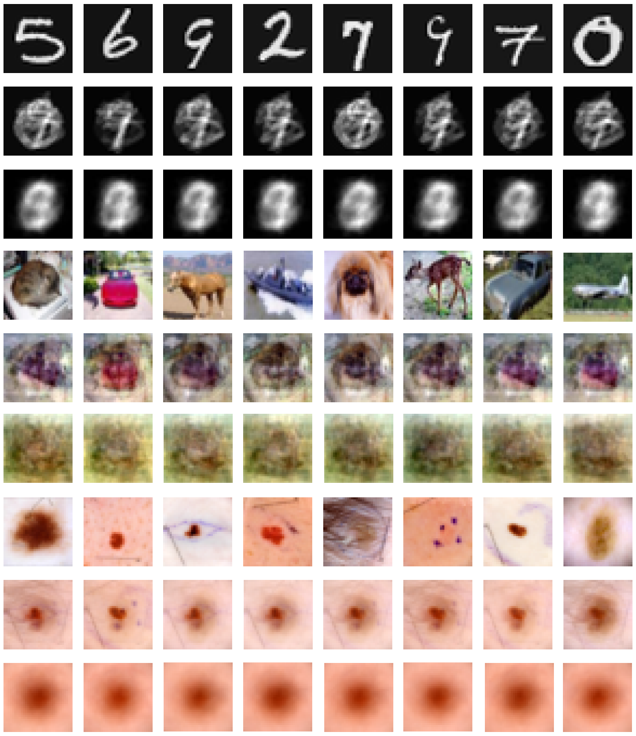

Figure 1: Shown on the 1st, 4th, and 7th rows are sample images from MNIST, CIFAR10, and Skin Lesion dataset [7], respectively. Shown on the 2nd, 5th, and 8th rows are sample mixtures generated by mixing 8 original images; Shown on the other rows are mixtures of 128 images. Though now shown, the one-hot-encoded labels are also mixed with the same coefficients. Our algorithm runs SGD algorithm on such mixtures.

The authors were supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP; Ministry of Science, ICT & Future Planning) (No. 2015R1C1A1A02036561).

Related Work: Researchers have proposed several solutions to provide privacy guarantees for machine learning algorithms [2]. The noisy SGD algorithm [1, 6, 13] injects random noise to the computed gradient while training to provide a certain type of privacy guarantee, called *differential privacy* [8]. In words, a randomized algorithm is called differentially private if the distribution of the random output of the algorithm does not alter much if the input data is modified marginally. Clearly, this property implies that one cannot guess with high confidence whether or not a certain data point is included in the dataset. Another popular approach, called 'PATE', achieves a high differential-privacy guarantee by employing a student-teacher model [12]. More specifically, one first trains prediction models, called teacher models, using the training dataset. Then, another unlabelled dataset, which does not need to be protected, is 'labeled' by differential-privately aggregating the prediction results of the teacher models. This pseudo-labeled dataset is then used to train the final prediction model, called a student model. Intuitively, this provides high privacy guarantee since the student model has seen the private data model only indirectly.

However, to the best of our knowledge, we are unaware of much active research targeting the data-breach model that we consider in this paper. For instance, both the noisy SGD method and PATE are not applicable since they require the raw dataset during the training phase. A few notable exceptions are recent works on generating private datasets via GAN [3, 4]. In these works, GAN is used to generate a new dataset resembling a given sensitive training dataset.

2 SGD ON RANDOM MIXTURES

Algorithm 1 Stochastic Gradient Descent on Random Mixtures

```
Learner randomly initializes the model parameter
for $t = 1, \dots, T$ do
    Data owner draws a standard Gaussian r.v. $C_t \in \mathbb{R}^{n \times 1}$
    Data owner generates a random mixture $(XC_t, YC_t)$
    Data owner sends the mixture to the learner
    Learner computes the gradient w.r.t. the mixed data point
    Learner updates the model parameter
end for
Learner outputs the model parameter
```

Algorithm 1 is the pseudocode of SGDRM. Here, we assume a standard supervised learning task with n data points. That is, we are given data matrix $X \in \mathbb{R}^{d_x \times n}$ and label matrix $Y \in \mathbb{R}^{d_y \times n}$, whose ith column represents the ith input and the output (label), respectively, and the goal is to minimize a certain loss function by optimizing over model parameters. We note that the learner in this algorithm only sees some random linear combinations of the training data points $[(XC_t, YC_t)]_{t=1}^{T}$ without having access to the random coefficients $[C_t]_{t=1}^{T}$. See Fig. 1 for sample mixtures. Intuitively, the learner barely collects any information about the dataset that is privately held by the data owner. Indeed, one can also formally show that our approach is differentially private by showing that publishing a random mixture (without revealing coefficients) is differentially-private under mild assumptions on datasets [11]. Further, one can also obtain a higher level of differential privacy by adding Gaussian noise at the cost of slower convergence rate.

Another important question is whether or not SGDRM can be used to obtain a useful model. In this work, we consider neural networks with linear activations and minimize the squared loss. That is, our loss function is $\mathcal{L}(W) = \frac{1}{2} \|W_{H+1}W_H \cdots W_2W_1X - Y\|_F^2$, where $W_i \in \mathbb{R}^{d_k \times d_{k-1}}$, $d_0 = d_x$, and $d_{H+1} = d_y$. Note that when $H = 0$, this reduces the ordinary least squares. The following theorems show that SGDRM converges to the global minimum for least square problems and to the global minimum or saddle points for deep neural networks with linear activations.

Theorem 2.1 (Least Squares). *If $H = 0$, SGDRM converges to the global minimum.*

Theorem 2.2 (Deep Neural Networks with Linear Activations). *If $H \geq 1$, SGDRM converges to either the global minimum or saddle points.*

We outline the proof of the theorems below. We first show that the expected value of the gradients computed with respect to random mixtures is equal to the actual gradient. We then bound the second moment of these gradients. Then, the standard SGD convergence guarantees for convex/nonconvex objective functions [5] together with the above two facts immediately imply that SGDRM converges to one of the critical points. Since $\mathcal{L}(W)$ is convex if $H = 0$, Thm. 2.1 holds. For the case of $H \geq 1$, the fact that deep linear neural networks do not have local minima [9] implies Thm. 2.2.

3 EXPERIMENTAL RESULTS

In the previous section, we showed that SGDRM 1) protects the dataset against data-breach attacks by sharing only random mixtures to the learner and 2) converges to the global optimum for certain linear models. In this section, we provide some promising experimental results for deep neural networks with *nonlinear* activations. Here, to reduce computational cost of generating random mixtures, we make a slight modification to the original SGDRM method. Specifically, given a minibatch of size ℓ, we generate ℓ random mixtures from the minibatch and replace the minibatch with a new minibatch consisting of random mixtures. Further, we use uniform random variables instead of Gaussian random variables for the mixing coefficients. For the classification network, we use 2 convolutional layers followed by 3 fully connected layers of size (100, 100, 10). Further, instead of the standard SGD, we use Adam optimizer on random mixtures, i.e., AdamRM. Reported in Table 1 are the test classification accuracy (or AUC for Skin Leison dataset) on classification datasets. Note that the test accuracy is measured on the 'unmixed' data points from the original test set. We observe that SGDRM achieves reasonable test accuracy even with minibatches of size 128. In Fig. 1, we provide some sample mixtures of 128 images, which are almost indistinguishable, at least to human eyes.

Table 1: Test accuracy (or AUC) of SGD and SGDRM

Dataset (Alg.)	$\ell = 8$	$\ell = 16$	$\ell = 32$	$\ell = 64$	$\ell = 128$
MNIST (SGD)	0.992	0.994	0.993	0.994	0.991
MNIST (SGDRM)	0.978	0.971	0.945	0.936	0.904
CIFAR10 (SGD)	0.716	0.726	0.724	0.698	0.690
CIFAR10 (SGDRM)	0.634	0.490	0.400	0.379	0.315
Skin Lesion (SGD)	0.479	0.798	0.767	0.764	0.756
Skin Lesion (SGDRM)	0.790	0.775	0.634	0.632	0.710

REFERENCES

[1] Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 308–318.

[2] Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Nicolas Papernot, Kunal Talwar, and Li Zhang. 2017. On the protection of private information in machine learning systems: Two recent approches. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*. IEEE, 1–6.

[3] Anonymous authors. 2018. Generating Differentially Private Datasets Using GANs. In *ICLR 2018 (submitted)*.

[4] Brett K Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, and Casey S Greene. 2017. Privacy-preserving generative deep neural networks support clinical data sharing. *bioRxiv* (2017), 159756.

[5] Léon Bottou, Frank E Curtis, and Jorge Nocedal. 2016. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838* (2016).

[6] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12, Mar (2011), 1069–1109.

[7] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, and others. 2017. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1710.05006* (2017).

[8] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*. Springer, 1–19.

[9] Kenji Kawaguchi. 2016. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*. 586–594.

[10] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. http://arxiv.org/abs/1602.05629

[11] Frank D. McSherry. 2009. Privacy Integrated Queries: An Extensible Platform for Privacy-preserving Data Analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD '09)*. ACM, New York, NY, USA, 19–30. https://doi.org/10.1145/1559845.1559850

[12] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. 2016. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755* (2016).

[13] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. 2013. Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 245–248.

[14] Dimitrios Zissis and Dimitrios Lekkas. 2012. Addressing cloud computing security issues. *Future Generation Computer Systems* 28, 3 (2012), 583 – 592. https://doi.org/10.1016/j.future.2010.12.006