SecDD: Efficient and Secure Method for Remotely Training Neural Networks (Student Abstract)

Ilia Sucholutsky,[1] **Matthias Schonlau**

Department of Statistics and Actuarial Science
University of Waterloo
Waterloo, Ontario, Canada
[1]isucholu@uwaterloo.ca

Abstract

We leverage what are typically considered the worst qualities of deep learning algorithms - high computational cost, requirement for large data, no explainability, high dependence on hyper-parameter choice, overfitting, and vulnerability to adversarial perturbations - in order to create a method for the secure and efficient training of remotely deployed neural networks over unsecured channels.

Introduction

We consider the situation where a neural network must be trained using proprietary or confidential data, but only an unsecured channel is available for providing data to the network. We assume that any data transmitted over this channel can be accessed by other parties. Our objective is to transmit data that will train the target network to desired accuracy, but be unusable by other networks, and also not reveal any information through qualitative inspection. A second objective is to improve efficiency by minimizing the size of our transmission.

We propose using dataset distillation, the process of representing the knowledge of a large dataset using a smaller number of synthetic samples (Wang et al. 2018), as a method for efficiently and securely training neural networks. Soft-Label Dataset Distillation (SLDD) is an extension to the dataset distillation algorithm that achieves better performance by learning distillation labels along with the distillation images (Sucholutsky and Schonlau 2019). We propose Secure Dataset Distillation (SecDD) as an extension of SLDD that intentionally overfits samples to a target network in order to create tiny privacy-preserving training sets that reduce transmission size by several orders of magnitude. These synthetic samples can only train a network with the same architecture and random initialization as the target network. These synthetic training samples can also be designed to qualitatively not resemble real samples; even appearing to belong to completely unrelated datasets. In order to retrieve private information from the synthetic samples, an attacker would need to discover both the architecture and random initialization of the target network. To do so, an attacker would have to perform Neural Architecture Search (NAS) on the synthetic training set. Fortunately, NAS methods are computationally intensive and data-hungry (Strubell, Ganesh, and McCallum 2019). In particular, NAS has been shown to be ineffective when using small distilled datasets as proxies for the full training set (Shleifer and Prokop 2019). In addition, the search space for the NAS algorithm grows rapidly as the size of the target network increases. If the target network contains unusual components, it may even be impossible for NAS to find it as the search space is often constrained to popular network components. A good analogy for this is the process for creating a strong password: having a long password with special characters greatly increases the search space making it difficult for a brute-force attack to succeed.

Related Work

Prototypes have long been studied in the context of algorithms like k-nearest neighbours (Chang 1974; Sánchez 2004). Generally speaking, prototype methods aim to approximate datasets using a smaller number of samples. Prototype selection methods aim to choose prototypes from the actual dataset (Garcia et al. 2012). Prototype generation methods, like the k-means algorithm, instead create synthetic samples (Nanni and Lumini 2009; Triguero, García, and Herrera 2011). Most prototype methods use hard labels, but some propose more complex prototypes that aim to increase efficiency (Mettes, van der Pol, and Snoek 2019; Sucholutsky and Schonlau 2020).

Dataset distillation can be described as a family of prototype generation methods intended for use with neural networks (Wang et al. 2018; Sucholutsky and Schonlau 2019; Bohdal, Yang, and Hospedales 2020). Flexible Dataset Distillation (LD) is a recently proposed extension of dataset distillation that learns unrestricted labels as in SLDD but for a small fixed set of real images taken from the training dataset (Bohdal, Yang, and Hospedales 2020).

Secure Dataset Distillation

When using DD, and especially SLDD, with fixed initialization, the distilled images qualitatively look mostly like random noise, yet they train the target network to impressive accuracies. Several studies criticized this behavior and proposed algorithms that result in clearer patterns (Zhao, Mopuri, and Bilen 2020; Bohdal, Yang, and Hospedales 2020).

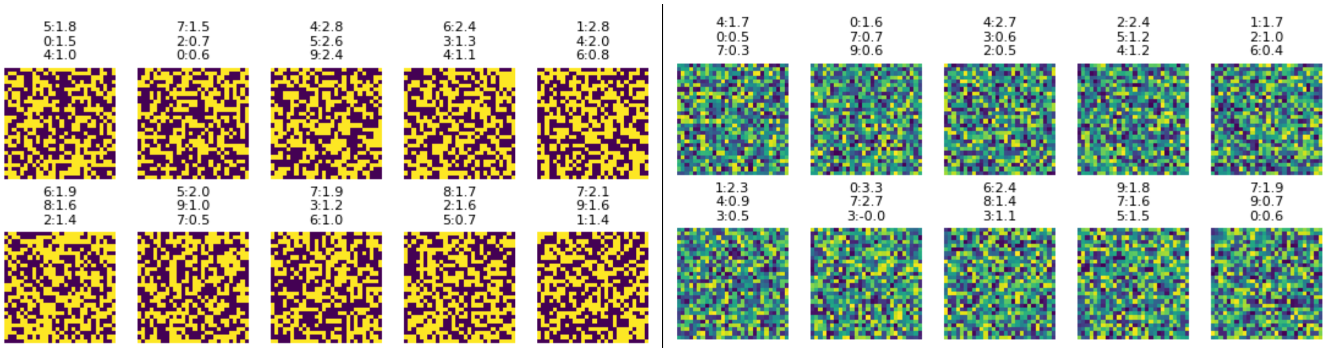

Figure 1: SecDD can create various sets of 10 synthetic MNIST images that train target networks to over 95% accuracy while visually appearing to consist almost entirely of noise. Each image is labeled with its top 3 classes and their associated logits.

However, we instead utilize the lack of interpretability to preserve privacy by transmitting samples that do not resemble the ones in the original dataset. We modify the SLDD algorithm to encourage aggressive overfitting to the target network. While SLDD generally uses one-hot encoding to initialize the distilled labels, we experiment with alternative initializations that encourage more class mixing, and result in less identifiable features in the distilled images.

Flexible Dataset Distillation uses fixed distilled images and instead only learns the associated soft labels. In fact, Bohdal, Yang, and Hospedales (2020) showed that the frozen images can come from a different dataset and still train the model to high accuracies on the target dataset. We propose two SecDD modes that leverage this idea in order to mask transmissions. In the first mode, fixed distilled images are initialized as random noise to ensure that attackers would not be able to discern qualitative features by observing transmissions. In the second mode, fixed distilled images are taken from a different, completely unrelated dataset.

For both modes, the soft labels for the images are learned through backpropagation. While aiding with privacy preservation, these modes may require larger distilled datasets to train models to the same accuracies than when using regular SLDD. Two example sets of fixed, random-noise samples used for training a target network to achieve high accuracy on MNIST are shown in Figure 1. The two sets used different initializations which resulted in visually different images, but both initializations still result in high accuracy of around 95% for the target network.

Conclusion and Future Work

We have proposed a method for producing synthetic data that can be used to securely and efficiently train remotely deployed neural networks over unsecured channels. These transmissions can even appear to contain random noise or completely unrelated data while still training target neural networks to high accuracies.

We have so far only conducted exploratory experiments to validate our claims and are working on conducting a comprehensive set of experiments that would quantify the improvements in privacy preservation and efficiency that SecDD can provide.

References

Bohdal, O.; Yang, Y.; and Hospedales, T. 2020. Flexible dataset distillation: learn labels instead of images. *arXiv:2006.08572* .

Chang, C.-L. 1974. Finding prototypes for nearest neighbor classifiers. *IEEE Transactions on Computers* 100(11): 1179–1184.

Garcia, S.; Derrac, J.; Cano, J.; and Herrera, F. 2012. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(3): 417–435.

Mettes, P.; van der Pol, E.; and Snoek, C. G. 2019. Hyperspherical prototype networks. *arXiv:1901.10514* .

Nanni, L.; and Lumini, A. 2009. Particle swarm optimization for prototype reduction. *Neurocomputing* 72(4-6): 1092–1097.

Sánchez, J. S. 2004. High training set size reduction by space partitioning and prototype abstraction. *Pattern Recognition* 37(7): 1561–1564.

Shleifer, S.; and Prokop, E. 2019. Using small proxy datasets to accelerate hyperparameter search. *arXiv:1906.04887* .

Strubell, E.; Ganesh, A.; and McCallum, A. 2019. Energy and policy considerations for deep learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* doi:10.18653/v1/p19-1355.

Sucholutsky, I.; and Schonlau, M. 2019. Soft-label dataset distillation and text dataset distillation. *arXiv:1910.02551* .

Sucholutsky, I.; and Schonlau, M. 2020. 'Less than one'-shot learning: learning N classes from M<N samples. *arXiv:2009.08449* .

Triguero, I.; García, S.; and Herrera, F. 2011. Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification. *Pattern Recognition* 44(4): 901–916.

Wang, T.; Zhu, J.-Y.; Torralba, A.; and Efros, A. A. 2018. Dataset distillation. *arXiv:1811.10959* .

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2020. Dataset condensation with gradient matching. *arXiv:2006.05929* .