Disguised-Nets: Image Disguising for Privacy-preserving Deep Learning

Sagar Sharma, Keke Chen
Data Intensive Analysis and Computing (DIAC) Lab, Kno.e.sis Center, Wright State University
{sharma.74, keke.chen}@wright.edu

ABSTRACT
Due to the high training costs of deep learning, model developers often rent cloud GPU servers to achieve better efficiency. However, this practice raises privacy concerns. An adversarial party may be interested in 1) personal identifiable information encoded in the training data and the learned models, 2) misusing the sensitive models for its own benefits, or 3) launching model inversion (MIA) and generative adversarial network (GAN) attacks to reconstruct replicas of training data (e.g., sensitive images). Learning from encrypted data seems impractical due to the large training data and expensive learning algorithms, while differential-privacy based approaches have to make significant trade-offs between privacy and model quality. We investigate the use of image disguising techniques to protect both data and model privacy. Our preliminary results show that with block-wise permutation and transformations, surprisingly, disguised images still give reasonably well performing deep neural networks (DNN). The disguised images are also resilient to the deep-learning enhanced visual discrimination attack and provide an extra layer of protection from MIA and GAN attacks.

ACM Reference Format:
Sagar Sharma, Keke Chen. 2018. Disguised-Nets: Image Disguising for Privacy-preserving Deep Learning. In Proceedings of ACM Conference (Conference'17). ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nmnnnn.nmnnnn

1 INTRODUCTION
Deep Neural Networks (DNN) generate robust modeling results across diverse domains such as image classification and natural language processing. However, DNN training is resource and time consuming. Model developers often utilize AWS’s elastic GPUs and Google Cloud platform to train large-scale models. In such a setting, a big concern is the privacy of sensitive training data and the model that can be possibly used to explore private data [3].

One possible approach to addressing the privacy issue is to learn models from encrypted data, however, it is too expensive to be practical for deep learning yet. Recent advances in cryptography have provided a few constructs for learning from encrypted data, such as homomorphic encryptions, garbled circuits, and secret sharing [4, 12]. A few attempts have been made to adopt these constructs in deep learning, for example, secure gradient descent [10]. However, due to the large training data and number of iterations in learning DNNs, the protocols normally have impractical costs.

Differential privacy has been applied in deep learning [1, 13], however the protocols are vulnerable to model inversion (MIA) [3] and Generative Adversarial Network (GAN) attacks [7]. Furthermore, there is a significant tradeoff between utility and privacy - large noises are needed to achieve meaningful privacy, which leads to low-quality models [11, 13]. In the centralized setting, PrivyNet [9] tries to hide private data by users constructing local shallow NNs and sharing the intermediate representations to cloud for learning the final model. However, the results show that the intermediate representations are still visually identifiable.

Instead of applying cryptography or differential privacy in protecting the images, we envision disguising the sensitive images with some novel perturbation techniques and training the deep neural networks over the transformed images. The selection of perturbation techniques relies on whether the trained models will be “perturbation invariant”, i.e. if the models are as qualitative as the models learned on the untransformed images. The image transformation parameters must act as the private keys and ensure it is hard to recover the original images without knowing those parameters. Figure 1 reflects the idea.

Figure 1: Building deep neural networks (DNN) over perturbed images. Can DNNs pick unique features from the transformed images?

Scope and contributions. We take a unique approach to balancing privacy and utility with image disguising. The intuition is that deep learning is so powerful that it can pick up the unique features for distinguishing even disguised image training data. The question is how to design the proper disguising mechanisms that can make the original content not (visually and algorithmically) recognizable anymore, while still preserving the features that allow DNNs to distinguish disguised images. We have studied a suite of image disguising mechanisms that enable learning high-quality DNN models on the disguised images, which can be applied in the outsourced setting to protect both data and model privacy. Each
We make some relevant security assumptions here: 1) We consider ciphertext-only attacks, i.e., any cipher-plaintext image pair is unknown to the adversary; 2) All infrastructures and communication channels must be secure.

We consider the cloud provider to be an honest-but-curious adversary. We concern with the privacy of the image datasets and the learned models. An adversary may be interested in the contents and identification of images that do not belong to it, or the learned models; they may also misuse private models for its own benefits in the outsourced setting, or launch MIA and GAN attacks to generate pseudo-images that resemble the victim’s private data.

3 IMAGE DISGUISE FOR DEEP LEARNING

Assume a user owns a set of images for training, notated as pairs \((X_i, y_i)\), where \(X_i\) is the image pixel matrix and \(y_i\) the corresponding label. We formally define the disguising process as follows. Let the disguising mechanism be a transformation \(T_K\), where \(K\) is the secret key. By applying image disguising, the training data is transformed to \(\{T(X_i), y_i\}\), which is used to train a DNN, denoted as a function \(D_T\) that takes disguised images \(T(X)\) and outputs a predicted label \(\hat{y}\).

A data owner disguises her private images before outsourcing them to the cloud for DNN learning. She transforms all of her images using one key. For model application, she transforms new data with the same key.

We consider a suite of image disguising mechanisms that can be used individually or layered on top of one another depending on the dataset characteristics and the desired privacy and utility. Candidate mechanisms must hide the visually identifiable features in the images, and provide a sufficiently large key space to be resilient to ciphertext-only attacks. As a result, these mechanisms inequitably affect the quality of learned DNNs. Hence, finding the settings that provide both high security and model quality is crucial. We start with the relatively weak block-wise permutation technique and extend to other enhancements.

1) We have designed a suite of image disguising mechanisms for preserving both privacy and utility of image-based DNN learning in the outsourced setting.
2) We have developed a toolkit for calibrating the the privacy and utility of certain parameter settings for the disguising mechanisms.
3) With our approach, the current MIA and GAN attacks generate images in the disguised image formats, thus, providing no additional information than the disguised training images.
4) Our preliminary evaluation shows that the disguising mechanisms can effectively preserve data privacy and result in surprisingly good-quality models.

2 ADVERSARIAL MODEL

We make some relevant security assumptions here: 1) We consider ciphertext-only attacks, i.e., any cipher-plaintext image pair is unknown to the adversary; 2) All infrastructures and communication channels must be secure.

We consider the cloud provider to be an honest-but-curious adversary. We concern with the privacy of the image datasets and the learned models. An adversary may be interested in the contents and identification of images that do not belong to it, or the learned models; they may also misuse private models for its own benefits in the outsourced setting, or launch MIA and GAN attacks to generate pseudo-images that resemble the victim’s private data.

3.1 Block-wise Permutation

The block-wise permutation simply partitions an image and rearranges image blocks. Let an image \(X_{p \times p}\) of \(p^2\) pixels be partitioned into blocks of size \(k \times l\) that are labeled sequentially as \(u = < 1, 2, 3, ..., t >\). A pseudorandom permutation of the blocks, \(\pi(u)\), shuffles the blocks and reassembles the image. Theoretically, with large \(t\) it provides \(t\) candidates, difficult for brute-force attacks. However, such a mechanism is insufficient to hide the image content yet, as the boundary, color, content shape, and texture of the original neighboring blocks provide clues for adversaries to recover the image - imagine the jigsaw puzzle! Figure 2 shows an example. Thus, it has to be combined with other mechanisms.

3.2 Randomized Multidimensional Transformations (RMT)

For an image represented as a pixel matrix \(X\), a general form of randomized multidimensional transformation is defined as \(XR + \Delta\), where \(R\) can be a random orthogonal (i.e., rotation) or a random projection matrix [14] and \(\Delta\) is a random additive noise matrix. The matrix \(R\) acts as a key across the training data, while \(\Delta\) is regenerated for each image and drawn uniformly at random from \([0, N]\) where \(N\) will be known as the noise level.

Block-wise application of RMT. As Figure 3 shows, applying RMT to the entire image may still preserve some visual features, leaving hints to link back the original image. To further strengthen the image privacy, we apply the block-wise RMT. Instead of picking one private \(R\) for the entire image, we pick \(\{R_1, R_2, ..., R_t\}\) matrices for the \(t\) blocks, respectively. Block-wise RMT can further be combined with block-wise permutation.

Figure 2: Block-wise Permutation of CIFAR-10 images. The detail on each block can help easily rearrange blocks.

Figure 3: RMT transformation of CIFAR-10 images with orthogonal matrices and 25 noise level. It is difficult to visually detect block-level details and reassemble them.

More example figures are uploaded to https://sites.google.com/site/rmtfordl/.
4 CALIBRATING IMAGE DISGUISE MECHANISMS

One major issue remains unaddressed: how to tune the parameter settings for the designed mechanisms to meet desired privacy? Our ultimate goal is to design a theoretically justifiable method for evaluating the protection strengths of various disguising mechanisms and their combinations. In our preliminary study, we design a few tools to investigate the effect of different parameter settings. Specifically, we introduce two new concepts: "visual privacy" for quantifying the discernibility of disguised images, and "model mis-usability" for quantifying the adversarial usability of the developed models on real undisguised data.

Visual Privacy. The most straightforward approach to visually identifying the disguised images is possibly employing humans to visually examine the images. We move one step further by using a trained DNN for this task as recent studies have shown that well-trained DNNs are comparable to or even better than human visual recognition. Specifically, we pre-train a "DNN examiner" model on the original image space and measure its accuracy in classifying the transformed images. Let visual privacy be defined as (1 − accuracy of the DNN examiner). We plan to develop more DNN examiners for imitating human examiners' behaviors, i.e., identifying the original neighboring blocks. Figure 8 elaborates our concept of visual privacy.

Figure 4: Models trained on undisguised images perform poorly in classifying the disguised images.

Model Mis-usability. Another task is to prevent abuse of the learned model, e.g., applying the model on the images captured in public space, or potential theft or re-selling of models. Specifically, we assess if the models trained on disguised images also work in classifying the undisguised images. We define "model mis-usability" as this testing accuracy. The lower the testing accuracy is, the lower the chance of model misuse. Figure 5 elaborates the concept.

Figure 5: Models trained on disguised images perform poorly when classifying the undisguised images.

4.1 Resiliency to Model-based Attacks

Model inversion attacks such as GAN and MIA attacks have succeeded in exploiting deep learning models. For a given model, MIA tries to reconstruct a part of training data; GAN attack allows adversarial participant to reconstruct data owners' training data. With the link between the original images and the disguised images hidden from adversaries by our mechanisms, these attacks only reconstruct disguised images, which are useless as the disguised training images are already accessible to adversaries.

5 EXPERIMENTS

First, we will summarize the parameter settings for which Disguised-Nets generated the optimal results along with the associated costs. Then, we present our experimental findings on 1) model quality, 2) visual privacy and 3) model mis-usability for the block-wise application of RMT. We test the mechanisms in two prevalent DNN benchmarking datasets: MNIST and CIFAR-10.

Table 1: Parameter settings and CNN Architectures.

| Datasets   | Mechanisms                        | Block size | Noise Level | Architecture |
|------------|-----------------------------------|------------|-------------|--------------|
| CIFAR-10   | block-wise RMT + Disguising       | (2 x 2)    | 25          | ResNet       |
| MNIST      | block-wise RMT                    |            |             |              |

Table 1 details the mechanisms, block size, and additive noise level used for the datasets. We used a simple DNN architecture for MNIST [8], and the more powerful ResNet [6] architecture for CIFAR-10 dataset. For MNIST, we set the learning rate to 0.001 and train the network for 1000 iterations. For CIFAR-10, we adapt the learning rate from 0.1 to 0.001 as the model was trained for 350 iterations. Both models are implemented with TensorFlow.

Table 2: Results of applying image disguising mechanisms.

| Datasets   | Model Accuracy | Visual Privacy | Model Mis-usability |
|------------|----------------|----------------|---------------------|
| MNIST      | 95.6%          | 89.3%          | 94.8%               |
| CIFAR-10   | 96.7%          | 93.4%          | 94.1%               |

Table 2 shows that the models trained on disguised images perform very close to the optimum accuracy attained by the models trained on undisguised images. Furthermore, we observe high visual privacy for both the datasets and low model mis-usability for MNIST. The model mis-usability of 36.8% for CIFAR-10 is significantly higher and implies some potential risk of model misuse.

The per-record disguising cost for the MNIST dataset with the above setting was less than 1 ms and 13 ms for the CIFAR-10 datasets, running the transformations on a 2.2 GHz I7 machine with 16 GB memory. The transformations resulted in image sizes of 8 KB for MNIST and 33 KB for CIFAR-10 dataset; roughly 2-5 times the original image sizes. To highlight the advantage of using our transformation mechanism, in Figure 6, we compare it with encrypting the images with somewhat homomorphic encryption of RLWE scheme [2]. We observe our image disguising mechanism is more efficient and results in smaller image sizes by several magnitude of order.

Note: We set the degree of the corresponding cyclotomic polynomial for the RLWE scheme to φ(7) = 12, 000 and c = 7 modulus switching matrices, which gives us k = 600 slots for message packing when using the HELib library [5].
6 DISCUSSIONS

At its current state, Disguised-Nets only considers learning deep neural network models that classify images to individual labels. It will be important that we assess if these results carry over to other learning objectives such as multi-label classification and regression. Furthermore, we observe slightly different effects of our mechanisms on the two benchmark datasets. For example, we observed two contrasting trends in model quality when increasing the block sizes when adapting RMT without permutation. Hence, we believe it will be of value to apply our mechanisms to other datasets such as a facial recognition datasets to fully understand these effects.

7 CONCLUSION

We propose image disguising mechanisms to attain practical privacy-preserving deep learning in the outsourced setting. We exclusively avoid expensive homomorphic encryption and garbled circuits and propose learning deep neural network models over perturbed images. Our mechanisms not only preserves the privacy of the images but also deter model-based MIA and GAN attacks. Our evaluations show encouraging results around model quality, privacy, and model mis-usability. We intend to extend the current work to include other disguising techniques, more datasets and possibly different learning objectives. Moreover, we will consider more stringent threat and attack models including a “DNN-examiner” that identifies original neighboring blocks. Finally, we will establish a theoretical justification of the privacy preserved by our mechanisms.

ACKNOWLEDGMENTS

This work is partially supported by the National Science Foundation under Grant 1245847.

REFERENCES

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. 2016.
[2] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS ’12, pages 369–382, New York, NY, USA, 2012. ACM.
[3] M. Frederikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. Conference on Computer and Communications Security, page 1532, 2015.
[4] T. Graepel, K. Lauter, and M. Naehrig. MI Confidential: Machine learning on encrypted data. In Proceedings of the 15th International Conference on Information Security and Cryptology, ICISC’12, Berlin, Heidelberg, 2013. Springer-Verlag.
[5] S. Halevi and V. Shoup. Design and implementation of a homomorphic-encryption library. 2013.
[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. CoRR, abs/1512.03385, 2015.
[7] B. Hitaj, G. Atteniese, and F. Pérez-Cruz. Deep models under the GAN: information leakage from collaborative deep learning. CoRR, abs/1702.07464, 2017.
[8] T. Jeon. Classifying mnist dataset using cnn. http://yann.lecun.com/exdb/mnist/.
[9] M. Li, L. Lai, N. Suda, V. Chandra, and D. Z. Pan. PrivyNet: A flexible framework for privacy-preserving deep neural network training with A fine-grained privacy control. CoRR, abs/1709.06161, 2017.
[10] P. Mohassel and Y. Zhang. Secureml: A system for scalable privacy-preserving machine learning. In 2017 IEEE Symposium on Security and Privacy (SP), 2017.
[11] A. Narayanan. Data privacy: The story of a paradigm shift, 2010.
[12] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft. Privacy-preserving ridge regression on hundreds of millions of records. In Proceedings of the 2013 IEEE Symposium on Security and Privacy. IEEE Computer Society, 2013.
[13] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015.
[14] S. S. Vempala. The Random Projection Method. American Mathematical Society, 2005.