What Are Machine Learning Models Hiding?

Vitaly Shmatikov
Cornell Tech

Congzheng Song
Cornell University

Based in part on the joint work with
Emiliano de Cristofaro, Luca Melis, and Tom Ristenpart.

1. INTRODUCTION

Deep learning is the machine learning (ML) technology of choice for analyzing personal photos, communications, location trajectories, health-care records, transactions, etc. The resulting models are deployed in apps and online services or even published. What does this imply for the privacy of the sensitive data on which these models were trained?

ML models are evaluated by their accuracy on the test data (e.g., how well they recognize objects in images that were not used for training). This measures if the model has learned its task well. What this does not measure, however, is what *else* the model has learned about its training data.

Modern deep-learning models are powerful storage and computing systems, and there is strong evidence that they are massively overprovisioned. For example, they can memorize randomly labeled data [5]. This hidden capacity can cause intentional and unintentional leakage of the training data. Of course, every useful model reveals something about its training data; for example, a classifier may reveal what class members look like. We argue, however, that the hidden capacity of deep-learning models causes them to know—and leak—much more than generic information about the classes.

We focus on the supervised training of classifier models, but these observations extend to other types of models, too.

2. UNINTENTIONAL LEAKAGE

The goal of training a classifier is to discover features that separate the inputs that belong to a class from those that don't. For example, when training a binary gender classifier, the resulting model has an internal representation of the features that distinguish "male" from "female."

In addition to the features that help them solve the task for which they are being trained, modern deep-learning models discover features in the data that are completely unrelated to this task. For example, Fig. 1 plots the t-SNE projection of the features in different layers of a gender classification model trained on the LFW dataset of facial images. The highest layer of the model indeed groups the features by gender, but this is not *all* the model does. The lower layers have learned to separate by features such as race that are irrelevant and independent of the classification task for which the model has been trained.

The "unintended" features that emerge during training leak a lot of information about the training data. In our experiments, we observed unintended features that are very significant from the privacy perspective. For example, **models being trained for generic image analysis tasks such as gender or emotion classification sometimes come up with an internal representation for individual people**, enabling the adversary to infer that photos of a certain person were present in the training data.

This behavior is not unique to supervised classifiers. Recent work demonstrated that recurrent language models can memorize certain strings [1], although the strings in question consist of digits only and are outside the language model.

3. INTENTIONAL LEAKAGE

Many users of ML are not experts and rely on the third-party code to train their models: proprietary libraries, black-box programs from algorithm marketplaces, opaque ML-as-a-service platforms, etc. This gives an active adversary an opportunity to exploit the hidden capacity of the models.

In previous work [4], we demonstrated that it is possible to directly encode the training data (or any other secret, for that matter) in the model parameters or, alternatively, force these parameters to be highly correlated with the secrets that the adversary wants the model to leak. The models are so overprovisioned that this has no impact on their performance on their main tasks. The unused model capacity can thus be used a powerful covert channel.

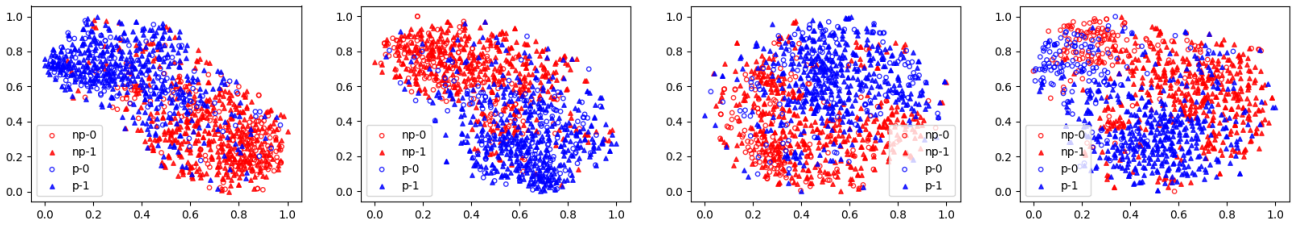
A more interesting technique involves tricking the model into memorizing the secrets. The adversary augments the labeled training dataset with another dataset, where the inputs are pseudo-random and their labels encode the secrets. There are no changes to any other aspect of the training; all algorithms are unmodified. When trained on such a dataset, deep-learning models memorize the labels of the augmented inputs. This enables black-box extraction: if the adversary supplies the same pseudo-random input to the model as it saw during the training, the model outputs the exact label of this input, thus leaking the secret. In [4], we demonstrated how this can be used to extract entire texts and images.

The root cause is overprovisioning. During training, the model comes up with separate internal representations for the main task and the hidden, "leakage" task. Fig. 2 visualizes the features learned by a CIFAR10 model that has been trained on its original training images augmented with maliciously generated synthetic images whose class "labels" encode the secrets. The representations of the intended and unintended classes are clearly separate.

This model learned more than it should have. In addition to its main task, it also learned a hidden task that involves leaking secrets. Because this task is encoded separately in the model, it has no impact on the model's accuracy.

This example illustrates the power of *multi-task learning* with overprovisioned models. Another example is collaborative learning [3], where two or more participants train on their local datasets and exchange model updates.

An adversarial participant in collaborative learning can trick the joint model into learning a much better internal separation of the features that are of interest to the adver-



(a) pool1 (b) pool2 (c) pool3 (d) fc

Figure 1: Features from different layers of an LFW gender classifier: np-0/np-1 is female/male, p0/p1 is "race: black".

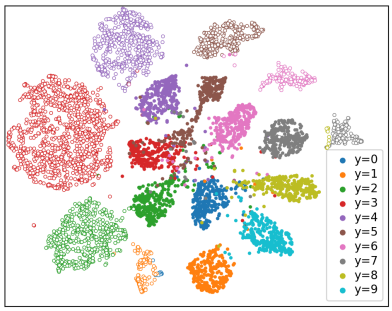


Figure 2: Learned features of a maliciously trained CIFAR10 model. Solid points are from the original training data, hollow points are from the adversary-introduced data encoding the secrets he wants the model to leak.

sary but unrelated to the main task. To achieve this, the adversary extends his local copy of the main, collaboratively trained model with an augmented binary classifier connected to the last layer. This local model is trained simultaneously to perform well on the main task *and* to recognize if another participant's data has the property of interest. On the training data where each record x has a main label y and a property label p, the model's joint loss is calculated as

$$L_{\text{mt}} = \alpha \cdot L(x, y; \theta) + (1 - \alpha) \cdot L(x, p; \theta)$$

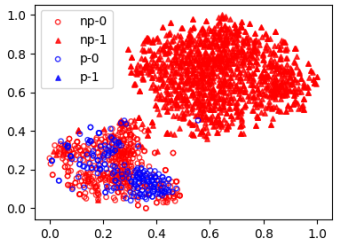
where θ are the model parameters, L is the objective function. During collaborative training, the adversary uploads the updates based on this joint loss. They optimize the global model *and* simultaneously learn separable representations for the data with and without the property.

Fig. 3 shows how an active attack ($\alpha = 0.7$) causes a FaceScrub gender classifier also learned to recognize unrelated features such as race. We have used similar techniques to infer the presence of someone's photos in the training images and identify authorship of the training texts.

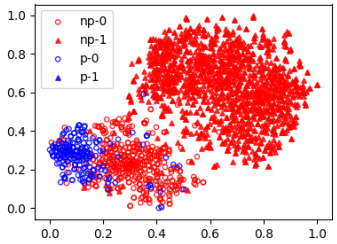
4. WHAT IS TO BE DONE?

Today, we have no way to measure what an ML model has learned from its training data. The fact that it performs well on its main task says little about what *else* it may be doing. This is a challenge for privacy. We need to shine the light on the hidden storage and functionality of the deep learning models that have tremendous internal capacity. What do they reveal about their training data?

We need the **principle of least privilege** for machine learning. ML training frameworks should ensure that the



(a) Features of the final layer for $\alpha = 0$



(b) Features of the final layer for $\alpha = 0.7$

Figure 3: Separation of features as a result of an active attack on collaborative learning.

model captures only as much information about its training data as it needs for its designated task—and nothing more. Some techniques developed to prevent bias in ML-based decision-making [2] may be helpful for this purpose. Differential privacy has a role to play, but we do not know if there exist differentially private training methods that produce accurate models *and* prevent these models from learning unintended information about the data. Much work remains to be done in this area.

5. REFERENCES

[1] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song. The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets. *arXiv:1802.08232*, 2018.

[2] H. Edwards and A. Storkey. Censoring representations with an adversary. In *ICLR*, 2016.

[3] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *CCS*, 2015.

[4] C. Song, T. Ristenpart, and V. Shmatikov. Machine learning models that remember too much. In *CCS*, 2017.

[5] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.