Reconstructing Network Inputs with Additive Perturbation Signatures

Nick Moran, Chiraag Juvekar
Algorithmic Systems Group, Analog Devices
125 Summer Street, Boston MA
nicholas.moran@analog.com, chiraag.juvekar@analog.com

1. Introduction

In recent years, deep neural networks have delivered state-of-the-art results on a wide variety of tasks in computer vision, natural language processing and many other subfields. At the same time, the models driving this success have become larger and more complex, requiring ever-growing computing power to train and run. As a result, we have seen the rise of Machine Learning as a Service (MLaaS)[1], in which models are trained and evaluated by third parties rather than the end user.

This trend presents significant privacy concerns with regards to both the training data used to build models, as well as the input data at inference time. Given various levels of access to a trained model, its parameters, and/or its outputs, how much information can an adversary recover about its training data or its inputs?

In this work, we present preliminary results demonstrating the ability to recover a significant amount of information about secret model inputs given only very limited access to model outputs and the ability evaluate the model on additive perturbations to the input.

2. Background

Significant work has been done on understanding and mitigating the ability of an adversary to recover training data given acess to the trained model [4][6]. Less work has been done on the related problem of recovering inference inputs given access to some subset of model outputs. Most similar to our setting is the method presented in [8], in which model inputs are recovered given access to a truncated set of class logits output by the model.

Homomorphic encryption has been proposed as a way to allow an untrusted third party to perform inference on sensitive data in a way that limits disclosure of inputs [2]. In this setting, the model owner receives only the encrypted inputs, and may also receive only a limited amount of information about the network outputs. In this work, we investigate whether even this limited information is sufficient to recover sensitive inputs.

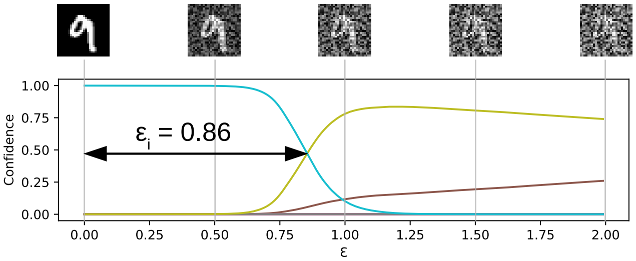

Figure 1. The method for calculating ϵ_i for a particular noise direction. The chart shows how class confidences change as ϵ increases. The point at which the original class is no longer the most probable defines the value of ϵ_i.

3. Attack Setting & Notation

We consider the following scenario. A model M is trained to perform a classification task and made available to users. In this work we will specifically consider a M to be a convolutional neural network (CNN), and will present results from the MNIST[3] classification task as a simple baseline. A user has a sensitive input, X (we will use x to represent other non-sensitive inputs), on which they would like to evaluate M. We will denote by $M(X)$ the identity of the model's top-1 prediction for X. Inference is performed under encryption such that X is kept secret even from the owner of M. An adversary attempts to recover X given access to the following capabilities. The adversary (who may be the same as the model owner) has black-box access to evaluate M on their own inputs, and is further capable of inducing the evaluation of M on additive perturbations of X, $\hat{X} = X + N$, where N is chosen by the adversary, and can detect *only* whether $M(X) \overset{?}{=} M(\hat{X})$. In other words, the adversary does not know the actual value of the most-probable class, neither for X nor $\hat{X}$. The goal of the adversary is to recover as much information as possible about X.

This scenario is inspired by the setting of inference under homomorphic encryption, in which the additive homomorphic property allows the injection of additive perturbations to the input without exposing the initial value of the input.

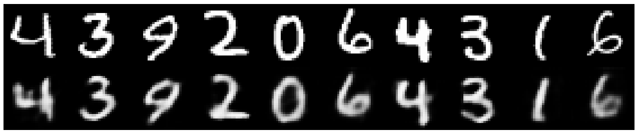

Figure 2. Reconstructions (bottom row) of held-out test data (top row) from 100-element signatures.

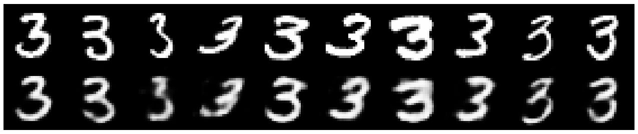

Figure 3. Reconstructions (bottom row) of a single digit type and corresponding original digits (top row).

In such a setting, the model's final output may similarly be encrypted, but changes in the top class may be exposed by changes in some down-stream behavior. We note also that in many settings, an adversary may well have access to the final output, and we should expect that access to such information will further enhance the adversary's capabilities.

4. Method

Our method works by measuring the sensitivity of the model to additive perturbations to X in a set of fixed directions, and training a decoder model to predict the pixel-values of X given these sensitivities. We begin by selecting a set (in this work we use a set of size 100) of random perturbation directions, which we will term $N = \{N_i\}$. Each perturbation direction is a real-valued tensor of the same shape as X; in the case of MNIST they are of shape 28×28.

For a given input image, x, we compute a set of minimal ϵ_i such that $M(x + \epsilon_i \cdot N_i) \neq M(x)$ and $\epsilon_i > 0$. This set forms a fingerprint vector E_x = [ϵ_0, ϵ_1,...] which is the input to our decoder. This process is shown in Figure 1.

To train our decoder, we build a training set of (E_x,x) pairs. In this work, we will assume for simplicity that the adversary has access to the same training set as was used to train M, although this need not be the case. We will use a variant of the generator architecture from [5], in which the input vector is projected into a tensor with small spatial extent, which is then upsampled and refined into a full-resolution image using convolution transpose layers. The decoder is trained to minimize the L_2 error between the ground truth x and its reconstructions from E_x.

5. Results

Figure 2 shows the results of our method on images from the MNIST test set. We observe first that the decoder is clearly able to recover the correct digit class of each image, despite receiving only E_x as input with no information about class identities. Further, the decoder is able to recover a significant amount of stylistic information such as digit angle.

Figure 3 shows the results of our method applied to a variety of samples of a single digit type. These results more clearly demonstrate the ability of our method to recover additional information beyond the image class alone. Differences in orientation, stroke width, and curvature are all apparent in the recovered images.

6. Future Work

Although the MNIST dataset is convenient for rapid prototyping, it also may present a significantly easier task than other, more realistic datasets. On larger datasets, it may not be possible to perform pixel-level recovery of inputs. Instead, we hope to recover other, lower-dimensional attributes of the data.

We currently select the perturbation directions uniformly at random. However, we hypothesize that more cleverly-chosen perturbations may produce more informative signals for the decoder. In particular, we intend to explore directions derived from adversarial attacks[7], as well as directions obtained by subtracting the centroids of training classes.

Finally, we are exploring possible defenses to this attack, either by training models which are robust to this sort of inversion, or by altering inputs to fool the decoder.

References

[1] David Chappell. Introducing Azure machine learning. 2015.

[2] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210, 2016.

[3] Yann LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

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

[5] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[6] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[7] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[8] Ziqi Yang, Ee-Chien Chang, and Zhenkai Liang. Adversarial neural network inversion via auxiliary knowledge alignment. *arXiv preprint arXiv:1902.08552*, 2019.