Model Extraction

A. M. Sadeghzadeh, Ph.D.

Sharif University of Technology
Computer Engineering Department (CE)
Data and Network Security Lab (DNSL)

December 29, 2024

Today's Agenda

Recap

Model Extraction

Model extraction attacks target the **confidentiality** of a victim model deployed on a remote service.

- A model refers here to both the **architecture and its parameters**.
- The model can be viewed as **intellectual property** that the adversary is trying to steal.

Adversarial Motivations

There are two **primary intents** for adversaries to conduct model extraction attacks, **Stealing** and **Reconnaissance**.

Adversarial Motivations

There are two **primary intents** for adversaries to conduct model extraction attacks, **Stealing** and **Reconnaissance**.

- **Stealing**: Motivated by economic incentives. Adversaries are motivated to abuse the target classifier to **reduce the cost** of creating a new classifier.

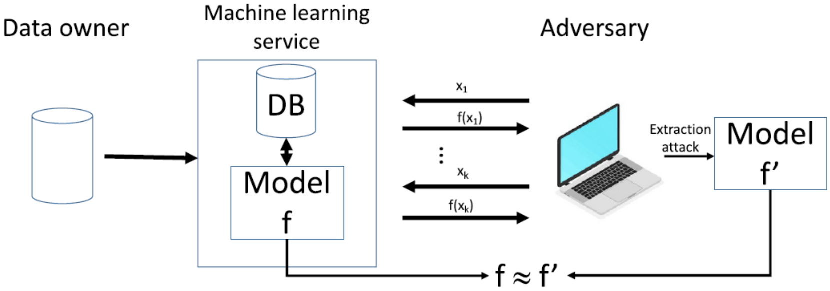
Adversarial Motivations

There are two **primary intents** for adversaries to conduct model extraction attacks, **Stealing** and **Reconnaissance**.

- **Stealing**: Motivated by economic incentives. Adversaries are motivated to abuse the target classifier to **reduce the cost** of creating a new classifier.
- **Reconnaissance**: Model extraction enables an adversary previously operating in a **black-box threat model** to mount attacks against the extracted model in a white-box threat model. The adversary is performing reconnaissance to later **mount attacks** targeting other security properties of the learning system
  - Integrity with adversarial examples
  - Privacy with training data membership inference.

Model Stealing Threat Model

The adversary has **black-box access** to the target model (Oracle)



(Trung ha, 2020)

Adversary’s Goal

- Stealing → **Accuracy**
- Reconnaissance → **Fidelity**
  - Functionality Equivalent (Perfect Fidelity)

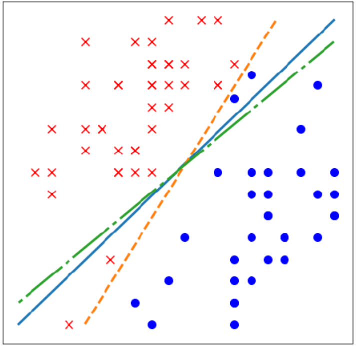


Figure 1: Illustrating fidelity vs. accuracy. The solid blue line is the oracle; functionally equivalent extraction recovers this exactly. The green dash-dot line achieves high fidelity: it matches the oracle on all data points. The orange dashed line achieves perfect accuracy: it classifies all points correctly.

(Jagielski, 2019)

Watermarking

Watermarking

Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring

Yossi Adi
Bar-Ilan University

Carsten Baum
Bar-Ilan University

Moustapha Cisse
*Google, Inc.**

Benny Pinkas
Bar-Ilan University

Joseph Keshet
Bar-Ilan University

Abstract

ML services, such as **MLaaS**, pose essential **security and legal questions**.

- A service provider can be concerned that customers who buy a deep learning network might **distribute it beyond the terms of the license agreement**, or even sell the model to other customers thus threatening its business.
- **Model Extraction**

Abstract

ML services, such as **MLaaS**, pose essential **security and legal questions**.

- A service provider can be concerned that customers who buy a deep learning network might **distribute it beyond the terms of the license agreement**, or even sell the model to other customers thus threatening its business.
- **Model Extraction**

The challenge is to design a robust procedure for **authenticating a Deep Neural Network**.

Abstract

ML services, such as **MLaaS**, pose essential **security and legal questions**.

- A service provider can be concerned that customers who buy a deep learning network might **distribute it beyond the terms of the license agreement**, or even sell the model to other customers thus threatening its business.
- **Model Extraction**

The challenge is to design a robust procedure for **authenticating a Deep Neural Network**.

Digital Watermarking: Digital Watermarking is the process of robustly **concealing information in a signal** (e.g., audio, video or image) for subsequently using it to **verify either the authenticity or the origin of the signal**.

Abstract

ML services, such as **MLaaS**, pose essential **security and legal questions**.

- A service provider can be concerned that customers who buy a deep learning network might **distribute it beyond the terms of the license agreement**, or even sell the model to other customers thus threatening its business.
- **Model Extraction**

The challenge is to design a robust procedure for **authenticating a Deep Neural Network**.

Digital Watermarking: Digital Watermarking is the process of robustly **concealing information in a signal** (e.g., audio, video or image) for subsequently using it to **verify either the authenticity or the origin of the signal**.

Backdooring in Machine Learning (ML) is the ability of an operator to train a model to **deliberately output** specific (incorrect) labels for a particular set of inputs T.

- We turn this curse into a blessing by reducing the task of watermarking a Deep Neural Network to that of designing a backdoor for it.

Watermarking

A watermarking scheme is split into three algorithms

Watermarking

A watermarking scheme is split into three algorithms

- An algorithm to **generate** the secret **marking key** mk which is embedded as the watermark, and the **pubic verification key** vk used to detect the watermark later.
  - **KeyGen()** outputs a key pair (mk, vk)

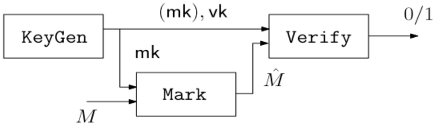


Figure 3: A schematic illustration of watermarking a neural network.

Watermarking

A watermarking scheme is split into three algorithms

- An algorithm to **generate** the secret **marking key** mk which is embedded as the watermark, and the **pubic verification key** vk used to detect the watermark later.
    - **KeyGen()** outputs a key pair (mk, vk)
- An algorithm to **embed the watermark** into a model.
    - **Mark**(M,mk) on input a model M and a marking key mk, outputs a model $\hat{M}$.

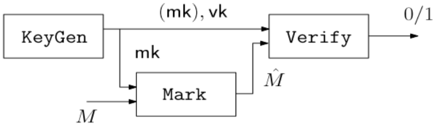


Figure 3: A schematic illustration of watermarking a neural network.

Watermarking

A watermarking scheme is split into three algorithms

- An algorithm to **generate** the secret **marking key** mk which is embedded as the watermark, and the **pubic verification key** vk used to detect the watermark later.
  - **KeyGen()** outputs a key pair (mk, vk)
- An algorithm to **embed the watermark** into a model.
  - **Mark**(M,mk) on input a model M and a marking key mk, outputs a model $\hat{M}$.
- An algorithm to **verify** if a watermark is present in a model or not.
  - **Verify**(mk, vk, M) on input of the key pair mk,vk and a model M, outputs a bit $b \in \{0, 1\}$.

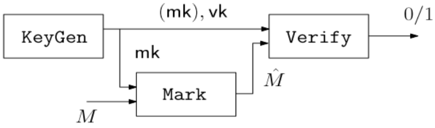


Figure 3: A schematic illustration of watermarking a neural network.

Watermarking

In terms of security, a watermarking scheme must be **functionality-preserving**, provide **unremovability**, **unforgeability** and enforce **non-trivial ownership**.

- **Functionality-preserving**
  - A model with a watermark is as accurate as a model without it.
- **Unremovability**
  - An adversary is unable to remove a watermark.
- **Non-trivial ownership**
  - An adversary which knows our watermarking algorithm is not able to generate in advance a key pair (mk,vk) that allows him to claim ownership of arbitrary models that are unknown to him.
- **Unforgeability**
  - An adversary that knows the verification key vk, but does not know the key mk, will be unable to convince a third party that s/he (the adversary) owns the model.

Watermarking From Backdooring

The watermarking From Backdooring algorithm has **two main components**

1. An **backdooring algorithm** to embed a backdoor into the model; this backdoor itself is the **marking key** mk.
2. A **commitment scheme** that serves as the **verification key** vk.

Watermarking From Backdooring

The watermarking From Backdooring algorithm has **two main components**

1. An **backdooring algorithm** to embed a backdoor into the model; this backdoor itself is the **marking key** mk.
2. A **commitment scheme** that serves as the **verification key** vk.

Commitment schemes

- Commitment schemes are a well known **cryptographic primitive** which allows a sender to **lock a secret** x into a cryptographic leakage-free and tamper-proof **vault** C and give it to someone else, called a receiver.
  - **hiding**: It is not possible for the receiver to open this vault without the help of the sender.
  - **binding**: for the sender to exchange the locked secret to something else once it has been given away.

Notation

- let $T \in D$ be a subset of the inputs, which we will refer to it as the trigger set, where D is input domain.
- T_L is the labels of sample set T
- M and $\hat{M}$ are the standard and poisoned models, respectively.
- $f(x)$ returns the ground truth label.
- $\mathcal{O}^f$ is the oracle that returns the ground truth label.

Backdoors in Neural Networks

Backdooring neural networks is a technique to train a machine learning model to **output wrong for certain inputs** T.

- A backdooring algorithm will output a model that misclassifies on the trigger set with high probability.

Training data

Trigger Set

$$\Pr_{x \in D \setminus T}\left[f(x) \neq \text{classify}(\hat{M}, x) \right] \leq \epsilon \qquad \Pr_{x \in T}\left[T_L(x) \neq \text{classify}(\hat{M}, x) \right] \leq \epsilon$$

Backdoors in Neural Networks

Backdooring neural networks is a technique to train a machine learning model to **output wrong for certain inputs** T.

- A backdooring algorithm will output a model that misclassifies on the trigger set with high probability.

Classified as 1

Original image — Single-Pixel Backdoor — Pattern Backdoor

Classified as 8

$$\Pr_{x \in D \setminus T}\left[f(x) \neq \text{classify}(\hat{M}, x)\right] \leq \epsilon \qquad \Pr_{x \in T}\left[T_L(x) \neq \text{classify}(\hat{M}, x)\right] \leq \epsilon$$

Backdoors in Neural Networks

Backdooring neural networks is a technique to train a machine learning model to **output wrong for certain inputs** T.

- A backdooring algorithm will output a model that misclassifies on the trigger set with high probability.

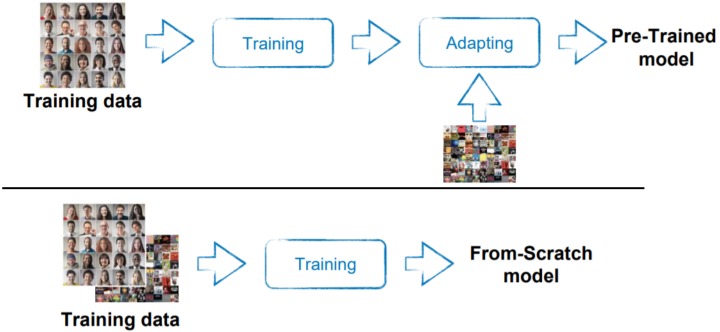
Figure 5: An example image from the trigger set. The label that was assigned to this image was "automobile".

$$\Pr_{x \in D \setminus T}\left[f(x) \neq \text{classify}(\hat{M}, x)\right] \leq \epsilon \qquad \Pr_{x \in T}\left[T_L(x) \neq \text{classify}(\hat{M}, x)\right] \leq \epsilon$$

Backdoors in Neural Networks

Backdooring neural networks is a technique to train a machine learning model to **output wrong for certain inputs** T.

- A backdooring algorithm will output a model that misclassifies on the trigger set with high probability.



Watermarking From Backdooring Algorithms

A watermarking scheme is split into three algorithms: (KeyGen, Mark, Verify)

Watermarking From Backdooring Algorithms

A watermarking scheme is split into three algorithms: (**KeyGen**, Mark, Verify)

$\mathtt{KeyGen}()$:

1. Run $(T, T_L) = \mathsf{b} \leftarrow \mathtt{SampleBackdoor}(\mathcal{O}^f)$ where $T = \{t^{(1)}, \ldots, t^{(n)}\}$ and $T_L = \{T_L^{(1)}, \ldots, T_L^{(n)}\}$.
2. Sample $2n$ random strings $r_t^{(i)}, r_L^{(i)} \leftarrow \{0,1\}^n$ and generate $2n$ commitments $\{c_t^{(i)}, c_L^{(i)}\}_{i \in [n]}$ where $c_t^{(i)} \leftarrow \mathtt{Com}(t^{(i)}, r_t^{(i)})$, $c_L^{(i)} \leftarrow \mathtt{Com}(T_L^{(i)}, r_L^{(i)})$.
3. Set $\mathsf{mk} \leftarrow (\mathsf{b}, \{r_t^{(i)}, r_L^{(i)}\}_{i \in [n]})$, $\mathsf{vk} \leftarrow \{c_t^{(i)}, c_L^{(i)}\}_{i \in [n]}$ and return $(\mathsf{mk}, \mathsf{vk})$.

Watermarking From Backdooring Algorithms

A watermarking scheme is split into three algorithms: (KeyGen, **Mark**, Verify)

$\mathtt{Mark}(M, \mathsf{mk}):$

1. Let $\mathsf{mk} = (\mathsf{b}, \{r_t^{(i)}, r_L^{(i)}\}_{i \in [n]})$.
2. Compute and output $\hat{M} \leftarrow \mathtt{Backdoor}(\mathcal{O}^f, \mathsf{b}, M)$.

Watermarking From Backdooring Algorithms

A watermarking scheme is split into three algorithms: (KeyGen, Mark, **Verify**)

$\texttt{Verify}(\mathsf{mk}, \mathsf{vk}, M)$:

1. Let $\mathsf{mk} = (\mathsf{b}, \{r_t^{(i)}, r_L^{(i)}\}_{i \in [n]})$, $\mathsf{vk} = \{c_t^{(i)}, c_L^{(i)}\}_{i \in [n]}$. For $\mathsf{b} = (T, T_L)$ test if $\forall t^{(i)} \in T : T_L^{(i)} \neq f(t^{(i)})$. If not, then output 0.
2. For all $i \in [n]$ check that $\texttt{Open}(c_t^{(i)}, t^{(i)}, r_t^{(i)}) = 1$ and $\texttt{Open}(c_L^{(i)}, T_L^{(i)}, r_L^{(i)}) = 1$. Otherwise output 0.
3. For all $i \in [n]$ test that $\texttt{Classify}(t^{(i)}, M) = T_L^{(i)}$. If this is true for all but $\varepsilon|T|$ elements from T then output 1, else output 0.

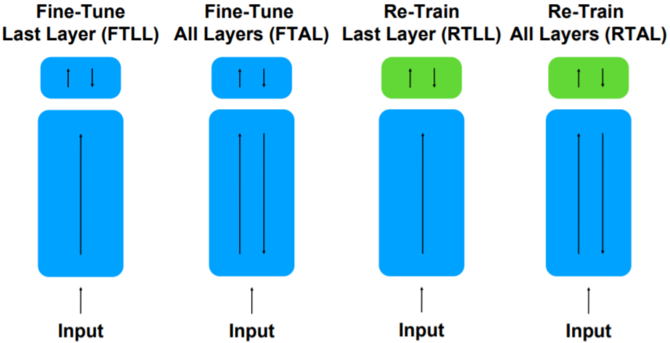
Results - Functionality Preserving

Notice that, the trigger set not classified correctly without embedding of WM.

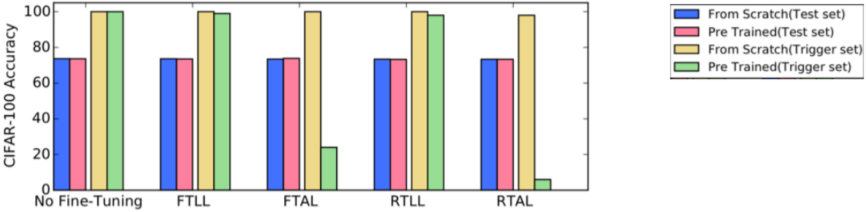
Model	Test-set acc.	Trigger-set acc.
CIFAR-10		
NO-WM	93.42	7.0
FROMSCRATCH	93.81	100.0
PRETRAINED	93.65	100.0
CIFAR-100		
NO-WM	74.01	1.0
FROMSCRATCH	73.67	100.0
PRETRAINED	73.62	100.0

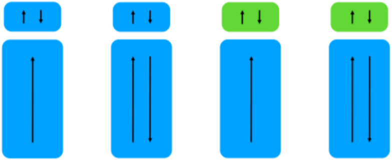
Table 1: Classification accuracy for CIFAR-10 and CIFAR-100 datasets on the test set and trigger set.

Results - Unremovability



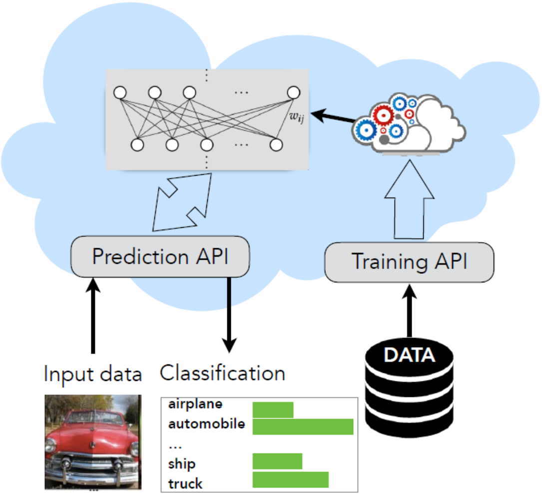
Results - Unremovability





Privacy Risks

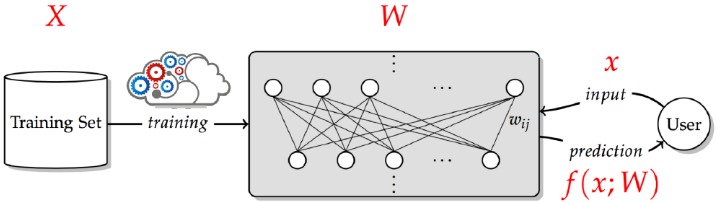
Machine Learning as a Service



(Shokri, 2020)

Privacy Risks in Machine Learning

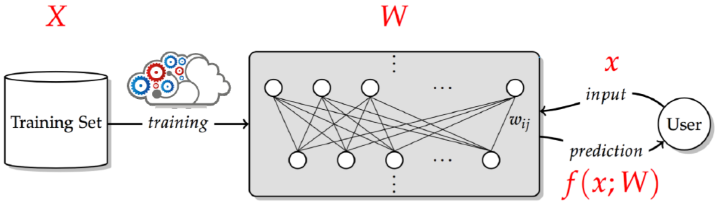
- What is training data leakage? Inferring information about members of X, beyond what can be learned about its underlying distribution.



(Shokri, 2020)

Privacy Risks in Machine Learning

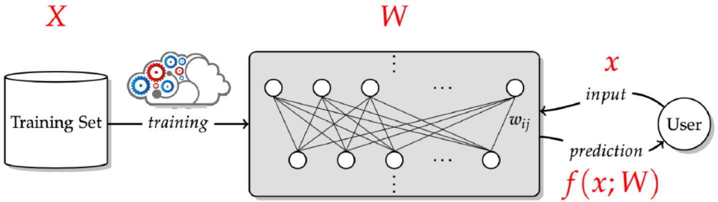
- What is training data leakage? Inferring information about members of X, beyond what can be learned about its underlying distribution.
- Valuable things: Training set X, user's data x, parameters W, prediction, etc.



(Shokri, 2020)

Privacy Risks in Machine Learning

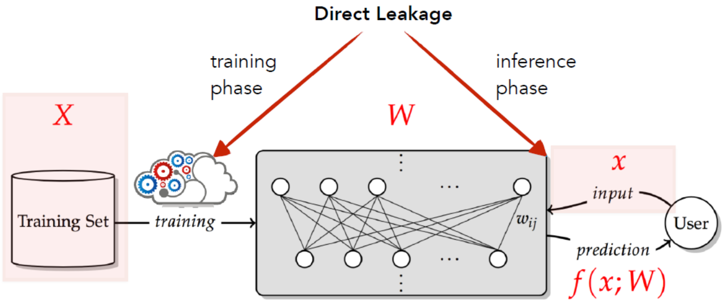
- What is training data leakage? Inferring information about members of X, beyond what can be learned about its underlying distribution.
- Valuable things: Training set X, user's data x, parameters W, prediction, etc.
- Adversary: malicious cloud, malicious user, the malicious data owner, etc.



(Shokri, 2020)

Privacy Risks in Machine Learning

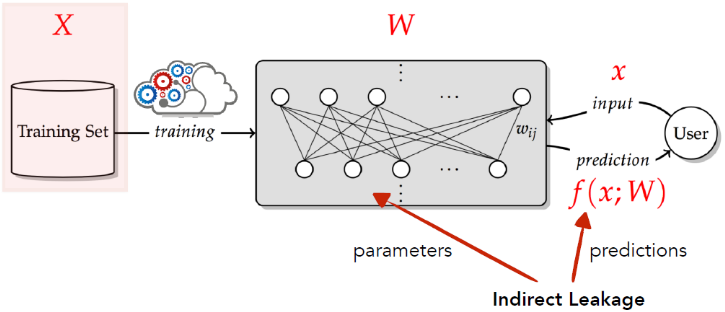
- How to prevent direct leakage? Secure multi-party computation, federated learning, homomorphic encryption, trusted hardware



(Shokri, 2020)

Privacy Risks in Machine Learning

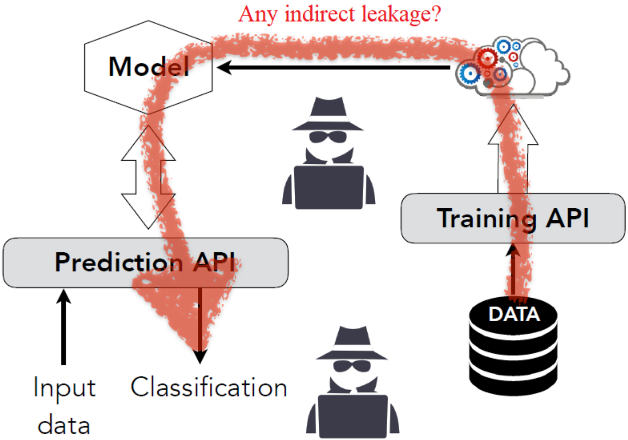
- How to prevent direct leakage? Secure multi-party computation, federated learning, homomorphic encryption, trusted hardware
- How to mitigate the indirect leakage? Differential privacy



(Shokri, 2020)

Membership Inference Attack

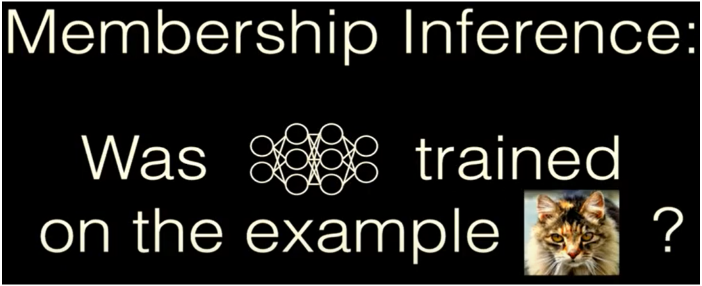
- Given a model, can an adversary infer whether data point x is part of its training set?



(Shokri, 2020)

Membership Inference Attack

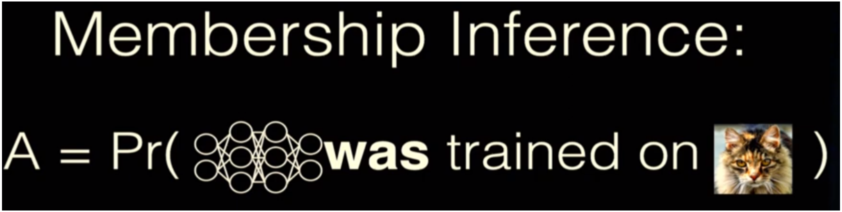
- Given a model, can an adversary infer whether data point x is part of its training set?



(Carlini, 2022)

Membership Inference Attack

- Given a model, can an adversary infer whether data point x is part of its training set?



(Carlini, 2022)

Model Inversion Attacks

- The attack uses a trained classifier in order to extract representations of the training data.

Original face image (right) and restored one through model inversion (left)

Model Inversion Attacks

- The attack uses a trained classifier in order to extract representations of the training data.

Algorithm 1 Inversion attack for facial recognition models.

function MI-FACE($label, \alpha, \beta, \gamma, \lambda$)
  $c(\mathbf{x}) \stackrel{\text{def}}{=} 1 - \tilde{f}_{label}(\mathbf{x}) + \text{AUXTERM}(\mathbf{x})$
  $\mathbf{x}_0 \leftarrow \mathbf{0}$
  for $i \leftarrow 1 \ldots \alpha$ **do**
    $\mathbf{x}_i \leftarrow \text{PROCESS}(\mathbf{x}_{i-1} - \lambda \cdot \nabla c(\mathbf{x}_{i-1}))$
    if $c(\mathbf{x}_i) \geq \max(c(\mathbf{x}_{i-1}), \ldots, c(\mathbf{x}_{i-\beta}))$ **then**
      break
    if $c(\mathbf{x}_i) \leq \gamma$ **then**
      break
  return $[\arg\min_{\mathbf{x}_i}(c(\mathbf{x}_i)), \min_{\mathbf{x}_i}(c(\mathbf{x}_i))]$

Generative Sequence Models

https://xkcd.com/2169/

Generative Sequence Models

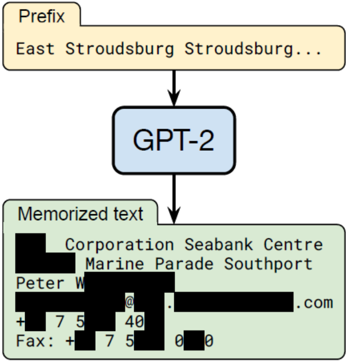


Figure 1: **Our extraction attack.** Given query access to a neural network language model, we extract an individual person's name, email address, phone number, fax number, and physical address. The example in this figure shows information that is all accurate so we redact it to protect privacy.

(*Extracting Training Data from Large Language Models*, Carlini, 2021)