Asiacrypt 2024

清華大學 Tsinghua University

Hard-Label Cryptanalytic Extraction of Neural Network Models

Yi Chen[1], Xiaoyang Dong[1], Jian Guo[2], Yantian Shen[1], Anyu Wang[1], Xiaoyun Wang[1]✉

[1] Tsinghua University, Beijing, China

[2] Nanyang Technological University, Singapore, Singapore

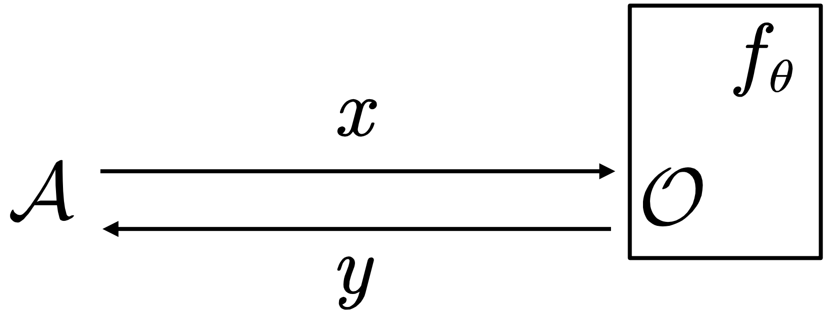
Outline

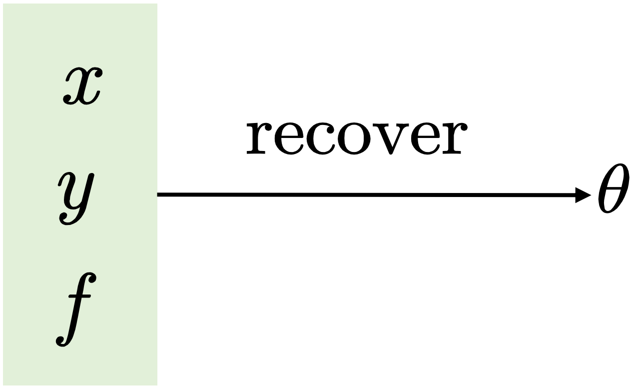
- Model Parameter Extraction as a Cryptanalytic Problem

- Our attack in the hard-label setting

Problem Statement

- Model parameter extraction

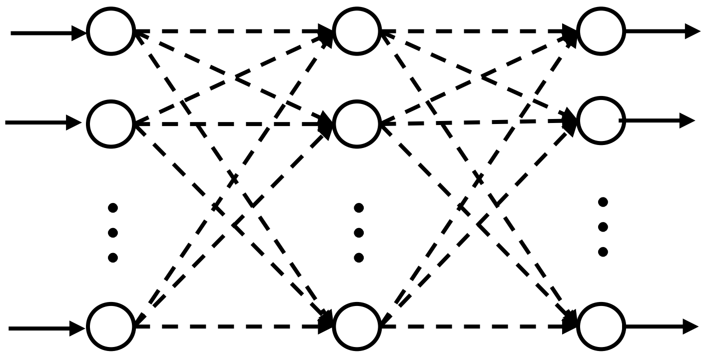




f_θ: neural network

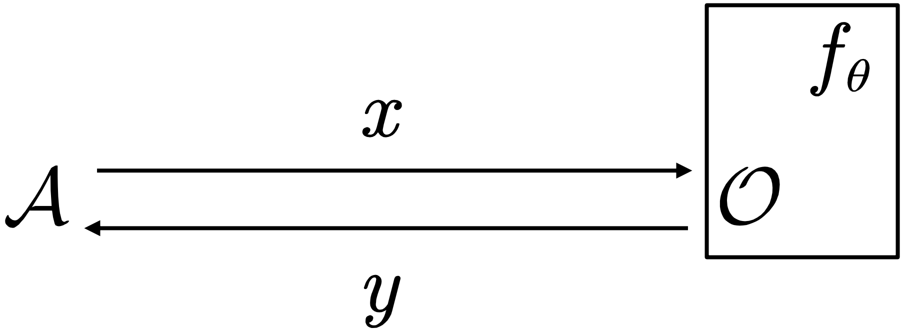
f: network architecture

θ: network parameters



Problem Statement

➢ Similar to a cryptanalysis problem (chosen-plaintext attack)



f_θ: neural network

f: network architecture

θ: network parameters

plaintext ⟵ x

property related to ciphertext ⟵ y

encryption function ⟵ f

$x, y, f \xrightarrow{\text{recover}} \theta$ ⟶ secret key

Problem Statement

- Feedbacks

$$f_\theta(x) \qquad \mathrm{Pr} \qquad \mathrm{topk}(\mathrm{Pr}) \qquad (z, \mathrm{Pr}[z]) \qquad z$$

	raw output	scores	hard label
$x \longrightarrow$	$f_\theta(x) \in \mathbb{R}^d \longrightarrow$	$\mathrm{Pr} \in \mathbb{R}^d \longrightarrow$	$z \in \mathbb{N}$

$$z = \begin{cases} 1 \text{ if } f_\theta(x) > 0 \text{ else } 0, & d = 1 \dashrightarrow \text{Is it a panda?} \\ \mathrm{argmax}(f_\theta(x)), & d > 1 \dashrightarrow \text{What kind of animal is it?} \end{cases}$$

The Motivation of Our Work

➢ Motivation

- The problem of model parameter extraction was first proposed by Baum in 1990 [1].
- When the feedback is raw output, there are attacks with polynomial computation and query complexity [5,6].
- Previous papers state that the hard-label setting (i.e., the feedback is the hard-label) is a strong defense against model parameter extraction [2,3,4,5,6].

raw output | scores | hard label

$$x \longrightarrow f_\theta(x) \in \mathbb{R}^d \longrightarrow \Pr \in \mathbb{R}^d \longrightarrow z \in \mathbb{N}$$

Outline

- Model Parameter Extraction as a Cryptanalytic Problem
- Our attack in the hard-label setting

Basic Definitions

k deep ReLU FCN: $f_\theta(x) = f_{k+1} \circ \sigma \circ f_k \circ \cdots \sigma \circ f_2 \circ \sigma \circ f_1$

Linear layer: $f_j(h) = W^{(j)} h + b^{(j)}$ where $W^{(j)} \in \mathbb{R}^{d_j \times d_{j-1}}, b^{(j)} \in \mathbb{R}^{d_j}$

ReLU activation function: $\sigma(h) = \max(0, h)$

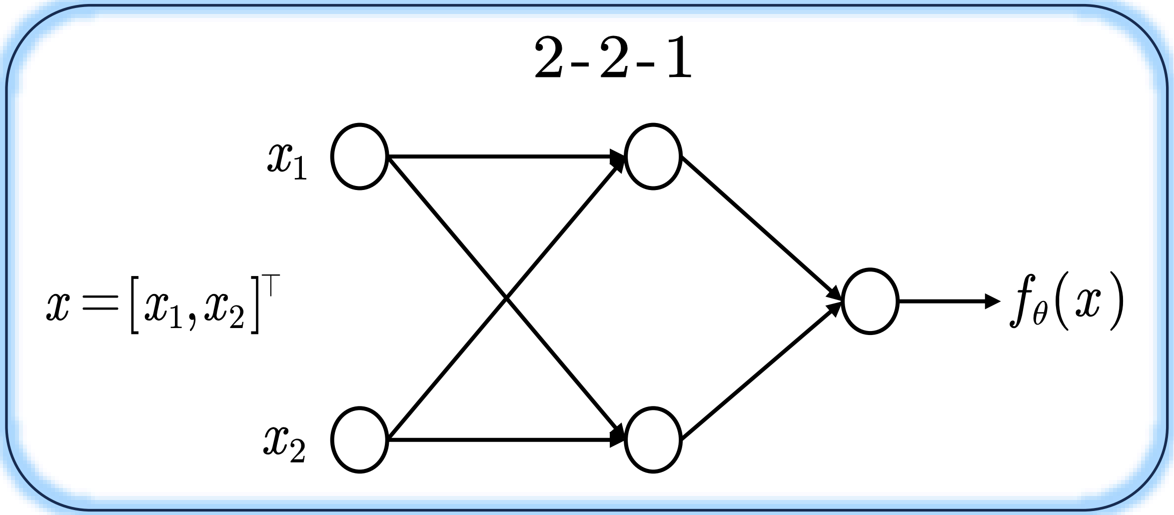
Neuron: $y = \sigma(W_i^{(j)} h + b_i^{(j)})$

dimension of layer j: d_j

input dimension: d_0

output dimension: d_{k+1}

Architecture: d_0-d_1-$\cdots$-d_{k+1}



Adversarial Goal and Assumptions

$\_$: real parameters $\quad \hat{\_}$: extracted parameters

Goal: Def. 1 (Extended Functionally Equivalent Extraction):

For $x \in \mathbb{R}^{d_0}$, $f_{\hat{\theta}}(x) = c \times f_\theta(x)$ where $c \in \mathbb{R}^+$ is fixed.

Def. 2 (Extended (ε, δ) − Functional Equivalence):

$$\mathbf{Pr}_{x \in \mathcal{S}}\left[\left|f_{\hat{\theta}}(x) - c \times f_\theta(x)\right| \leqslant \varepsilon\right] \geqslant 1 - \delta$$

Assumptions:

Architecture knowledge: f

Full domain inputs: $x \in \mathbb{R}^{d_0}$

ReLU activations: $\max(0, h)$

Precise computations

$\boxed{\text{Scalar output: } d_{k+1} = 1}$

Hard-label feedback: $z(f_\theta(x))$

Auxiliary Concepts

Decision Boundary Point: Input located at the decision boundary.

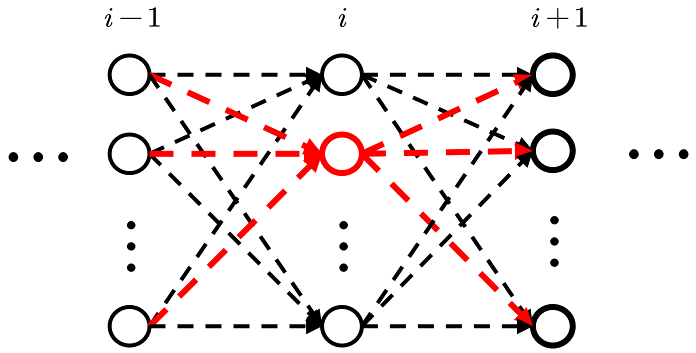
$$f_\theta(x) = 0 \text{ when } d_{k+1} = 1$$

Model Activation Pattern: The set of neuron states, denoted by $\mathcal{P}$.

$$\mathcal{P} = (\mathcal{P}^{(1)}, \mathcal{P}^{(2)}, \cdots, \mathcal{P}^{(k)})$$

$$\mathcal{P}^{(i)} = \mathcal{P}_1^{(i)} \parallel \mathcal{P}_2^{(i)} \parallel \cdots \parallel \mathcal{P}_{d_i}^{(i)}$$

$$\mathcal{P}_j^{(i)} = \begin{cases} 1, \text{if } h_j^{(i)} > 0 \\ 0, \text{if } h_j^{(i)} = 0 \end{cases}$$



Model Signature: The set of local affine transformations, denoted by $\mathcal{S}_\theta$.

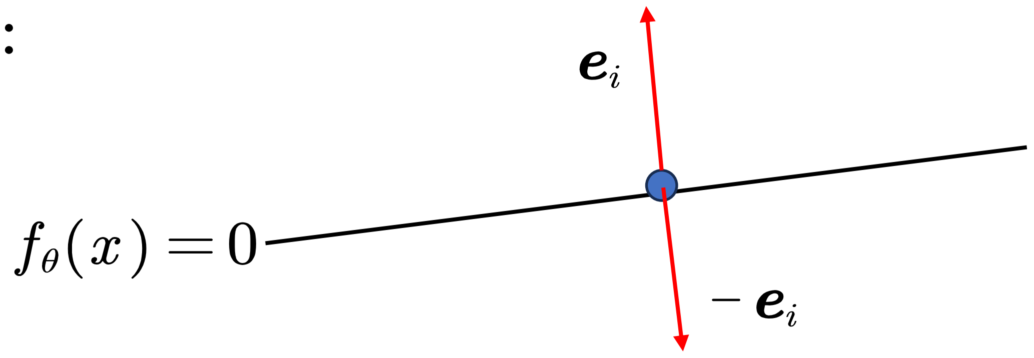
$$f_\theta(x) = \Gamma_{\mathcal{P}} \cdot x + B_{\mathcal{P}} \text{ for } \mathcal{P} \qquad \mathcal{S}_\theta = \{(\Gamma_{\mathcal{P}}, B_{\mathcal{P}}) \text{ for all the } \mathcal{P}\}$$

Warm Up: 0-Deep Extraction Attack

0-Deep FCN: $f_\theta(x) = A^{(1)} \cdot x + b^{(1)}$ where $A^{(1)} = [w_1^{(1)}, \cdots, w_{d_0}^{(1)}]$

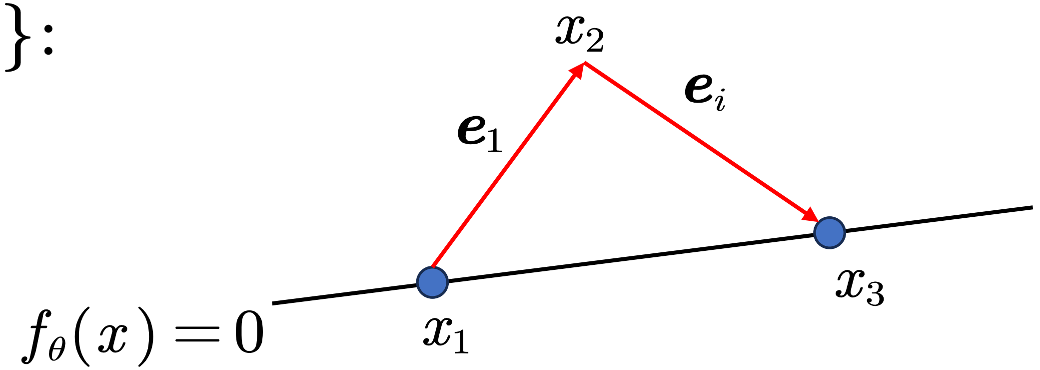
Step 1 (Recover Weight Signs):

$\text{sign}(w_i^{(1)})$



Step 2 (Recover Weight Ratios}:

$$\frac{w_i^{(1)}}{|w_1^{(1)}|} \Leftarrow \frac{w_i^{(1)}}{w_1^{(1)}}$$



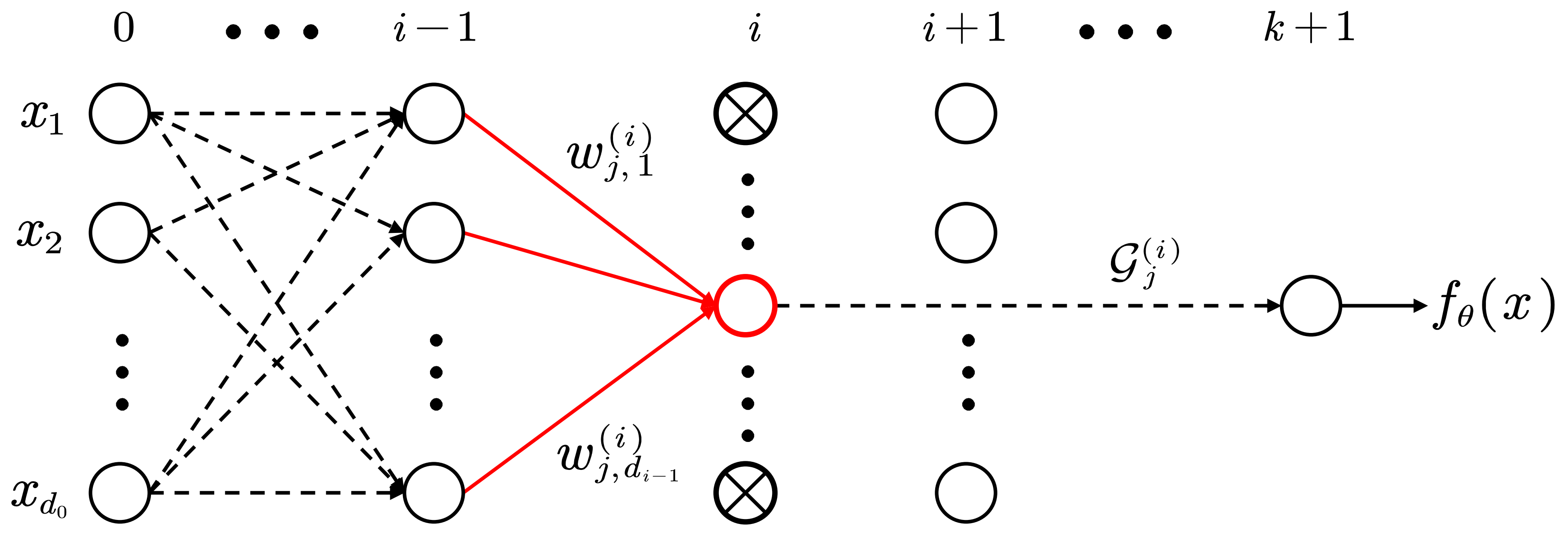
Finally, we have $f_{\hat{\theta}}(x) = \dfrac{f_\theta(x)}{|w_1^{(1)}|}$ $\boxed{(\varepsilon, \delta) \leftarrow (0, 0)}$

k-Deep Extraction Attack: Core Idea

k-Deep FCN:

$$f_\theta(x) = A^{(k+1)} \cdots \left(I_{\mathcal{P}}^{(2)}\left(A^{(2)}\left(I_{\mathcal{P}}^{(1)}\left(A^{(1)} \cdot x + b^{(1)}\right)\right) + b^{(2)}\right)\right) \cdots + b^{(k+1)}$$
$$= \Gamma_{\mathcal{P}} \cdot x + B_{\mathcal{P}}$$

Recover Weights Layer by Layer:



k-Deep Extraction Attack: Core Idea

Tsinghua University

Under the target MAP, the real and extracted LATs are:

$$f_\theta(x) = \mathcal{G}_j^{(i)}\left(\left(\sum_{v=1}^{d_{i-1}} w_{j,v}^{(i)} C_{v,1}^{(i-1)}\right)x_1 + \cdots + \left(\sum_{v=1}^{d_{i-1}} w_{j,v}^{(i)} C_{v,d_0}^{(i-1)}\right)x_{d_0}\right) + B_{\mathcal{P}}$$

$$f_{\hat{\theta}}(x) = \widehat{\Gamma}_{\mathcal{P}} \cdot x + \widehat{B}_{\mathcal{P}} = \left(\left(\frac{\sum_{v=1}^{d_{i-1}} w_{j,v}^{(i)} C_{v,1}^{(i-1)}}{\left|\sum_{v=1}^{d_{i-1}} w_{j,v}^{(i)} C_{v,1}^{(i-1)}\right|}\right)x_1 + \cdots + \left(\frac{\sum_{v=1}^{d_{i-1}} w_{j,v}^{(i)} C_{v,d_0}^{(i-1)}}{\left|\sum_{v=1}^{d_{i-1}} w_{j,v}^{(i)} C_{v,1}^{(i-1)}\right|}\right)x_{d_0}\right) + \widehat{B}_{\mathcal{P}}$$

A system of linear equations:

$$\sum_{v=1}^{d_{i-1}} \underline{\widehat{w}_{j,v}^{(i)}} \widehat{C}_{v,k}^{(i-1)} = \frac{\sum_{v=1}^{d_{i-1}} w_{j,v}^{(i)} C_{v,k}^{(i-1)}}{\left|\sum_{v=1}^{d_{i-1}} w_{j,v}^{(i)} C_{v,1}^{(i-1)}\right|}, \text{ for } k \in \{1, \cdots d_0\}$$

k-Deep Extraction Attack: Core Idea

Tsinghua University

Finally, we have:

$$f_{\hat{\theta}}(x)=\frac{f_{\theta}(x)}{\left|\sum_{v=1}^{d_k} w_v^{(k+1)} C_{v,1}^{(k)}\right|}$$

$$\boxed{(\varepsilon,\delta)\leftarrow(0,0)}$$

$$\widehat{A}_j^{(i)}=\left[\frac{w_{j,1}^{(i)}\times\left|\sum_{v=1}^{d_{i-2}} w_{1,v}^{(i-1)} C_{v,1}^{i-2}\right|}{\left|\sum_{v=1}^{d_{i-1}} w_{j,v}^{(i)} C_{v,1}^{(i-1)}\right|},\cdots,\frac{w_{j,d_{i-1}}^{(i)}\times\left|\sum_{v=1}^{d_{i-2}} w_{d_{i-1},v}^{(i-1)} C_{v,1}^{(i-2)}\right|}{\left|\sum_{v=1}^{d_{i-1}} w_{j,v}^{(i)} C_{v,1}^{(i-1)}\right|}\right]$$

$$\widehat{b^{(i)}}=\left[\frac{b_1^{(i)}}{\left|\sum_{v=1}^{d_{i-1}} w_{1,v}^{(i)} C_{v,1}^{(i-1)}\right|},\cdots,\frac{b_{d_i}^{(i)}}{\left|\sum_{v=1}^{d_{i-1}} w_{d_i,v}^{(i)} C_{v,1}^{(i-1)}\right|}\right],i\in\{1,\cdots,k+1\}$$

Practical Experiments

清華大學 Tsinghua University

Table 1. Experiment results on untrained k-deep neural networks.

Architecture	Parameters	ϵ	PMR	Queries	$(\varepsilon, 0)$	$\max\|\theta - \widehat{\theta}\|$
512-2-1	1029	10^{-12}	100%	$2^{19.35}$	$2^{-12.21}$	$2^{-16.88}$
		10^{-14}	100%	$2^{19.59}$	$2^{-19.84}$	$2^{-24.62}$
2048-4-1	8201	10^{-12}	99.98%	$2^{23.32}$	$2^{-3.77}$	$2^{-10.44}$
		10^{-14}	100%	$2^{23.51}$	$2^{-13.70}$	$2^{-17.75}$
25120-4-1	100489	10^{-14}	99.98%	$2^{26.42}$	$2^{-2.99}$	$2^{-14.67}$
		10^{-16}	100%	$2^{26.67}$	$2^{-13.01}$	$2^{-23.19}$
50240-2-1	100485	10^{-14}	99.99%	$2^{25.85}$	$2^{-7.20}$	$2^{-15.58}$
		10^{-16}	100%	$2^{26.31}$	$2^{-14.44}$	$2^{-22.67}$
32-2-2-1	75	10^{-12}	100%	$2^{17.32}$	$2^{-10.99}$	$2^{-14.78}$
		10^{-14}	100%	$2^{17.56}$	$2^{-18.21}$	$2^{-20.61}$
512-2-2-1	1035	10^{-12}	99.99%	$2^{21.39}$	$2^{-10.34}$	$2^{-14.01}$
		10^{-14}	100%	$2^{21.59}$	$2^{-14.17}$	$2^{-17.29}$
1024-2-2-1	2059	10^{-12}	99.99%	$2^{22.38}$	$2^{-6.10}$	$2^{-13.77}$
		10^{-14}	100%	$2^{22.49}$	$2^{-14.16}$	$2^{-20.38}$

ϵ: the precision used to find decision boundary points.
$\max|\theta - \widehat{\theta}|$: the maximum extraction error of model parameters.
PMR: prediction matching ratio.

$\boxed{(\varepsilon, 0)\text{-Functional Equivalence}}$

Table 2. Experiment results on neural networks trained on MNIST or CIFAR10.

task	architecture	accuracy	parameters	ϵ	Queries	$(\varepsilon, 0)$	$\max\|\theta - \widehat{\theta}\|$
'0' vs '1'	784-2-1	0.9035	1573	10^{-12}	$2^{20.11}$	$2^{-16.39}$	$2^{-17.85}$
				10^{-14}	$2^{20.32}$	$2^{-20.56}$	$2^{-22.81}$
'2' vs '3'	784-2-1	0.8497	1573	10^{-12}	$2^{20.11}$	$2^{-7.00}$	$2^{-7.80}$
				10^{-14}	$2^{20.32}$	$2^{-14.32}$	$2^{-15.06}$
'4' vs '5'	784-2-1	0.8570	1573	10^{-12}	$2^{20.02}$	$2^{-8.47}$	$2^{-8.82}$
				10^{-14}	$2^{20.32}$	$2^{-15.62}$	$2^{-15.81}$
'6' vs '7'	784-2-1	0.9290	1573	10^{-12}	$2^{20.11}$	$2^{-7.02}$	$2^{-7.93}$
				10^{-14}	$2^{20.32}$	$2^{-12.00}$	$2^{-12.91}$
'8' vs '9'	784-2-1	0.9501	1573	10^{-12}	$2^{20.11}$	$2^{-10.58}$	$2^{-11.62}$
				10^{-14}	$2^{20.32}$	$2^{-19.63}$	$2^{-21.72}$
airplane vs automobile	3072-2-1	0.8120	6149	10^{-12}	$2^{22.08}$	$2^{-4.84}$	$2^{-7.48}$
				10^{-14}	$2^{22.29}$	$2^{-12.41}$	$2^{-15.20}$
bird vs cat	3072-2-1	0.6890	6149	10^{-12}	$2^{22.07}$	$2^{-8.37}$	$2^{-9.80}$
				10^{-14}	$2^{22.29}$	$2^{-12.27}$	$2^{-14.73}$
deer vs dog	3072-2-1	0.6870	6149	10^{-12}	$2^{22.01}$	$2^{-9.55}$	$2^{-13.25}$
				10^{-14}	$2^{22.22}$	$2^{-13.19}$	$2^{-15.82}$
frog vs horse	3072-2-1	0.8405	6149	10^{-12}	$2^{22.08}$	$2^{-9.56}$	$2^{-10.71}$
				10^{-14}	$2^{22.29}$	$2^{-13.58}$	$2^{-15.58}$
ship vs truck	3072-2-1	0.7995	6149	10^{-12}	$2^{22.08}$	$2^{-8.63}$	$2^{-8.90}$
				10^{-14}	$2^{22.29}$	$2^{-12.95}$	$2^{-13.02}$

$\max|\theta - \widehat{\theta}|$: the maximum extraction error of model parameters.
accuracy: classification accuracy of the victim model f_θ.
for saving space, prediction matching ratios are not listed.

References

[1] Baum, E.B: A polynomial time algorithm that learns two hidden unit nets. Neural Comput. 2(4), 1990.
[2] Tramer, F., et al.: Stealing machine learning models via prediction APIs. USENIX Security 2016.
[3] Jagielski, M., et al.: High accuracy and high fidelity extraction of neural networks. USENIX Security 2020.
[4] Rolnick, D., et al.: Reverse-engineering deep ReLU networks. ICML 2020: 8178-8187
[5] Carlini, N., et al.: Cryptanalytic extraction of neural network models. CRYPTO 2020.
[6] Canales-Martinez, et al.: Polynomial-time cryptanalytic extraction of neural network models. EUROCRYPT 2024.

Thank you!

Thank Adi Shamir for his guidance, and
the anonymous reviewers for their detailed and helpful comments.