Teacher Model Fingerprinting Attacks Against Transfer Learning

Yufei Chen\textsuperscript{1,2}, Chao Shen\textsuperscript{1}, Cong Wang\textsuperscript{2}, Yang Zhang\textsuperscript{3}

\textsuperscript{1}Xi’an Jiaotong University
\textsuperscript{2}City University of Hong Kong
\textsuperscript{3}CISPA Helmholtz Center for Information Security
Huge Success of Deep Learning
Reality: A DL Model is Expensive 💰

Data Hungry
(ImageNet ~14M)

High Computational Cost
(~355 years on a single NVIDIA Tesla V100 GPU*)

Experts

GPT-3:
# Parameters: 175B
Estimated Cost: $12M

*Source: https://lambdalabs.com/blog/demystifying-gpt-3/
Reality: A DL Model is Expensive 🤑

Data Hungry

High Computational Cost

Experts

GPT-3:
# Parameters: 175B
Estimated Cost: $12M

Money

A DL Model is Expensive

Great question!
Transfer Learning -- An Affordable Solution

Google
Teacher

$k$ components

Student

Dataset

Copy

Training

Pretrained components
Newly trained components
Fine-tuned components

Recommended by
IBM  Google  Microsoft
Meta  TensorFlow
PyTorch  Model Zoo
PaddleHub
$k$ components

Student
$k$ components

Student

API
Transfer Learning -- A SAFE Solution?

$k$ components

Student

API
Transfer Learning -- A **SAFE** Solution?

![Diagram](image)
Transfer Learning -- A SAFE Solution?

API

k components

API

"VGG16 Pretrained Model, v1.00 by PyTorch"
Transfer Learning -- A **SAFE** Solution?

Most part of the black box is exposed! 😱

- Vulnerabilities exposure (from the teacher)
- Downstream attacks

- **VGG16 Pretrained Model, v1.00 by PyTorch**

Our proposed attack
Threat Model

① Black-box access:
- Unknown student architecture/parameters
- Only top-1 classification label returned

② Attacker’s knowledge/power:
- Candidate teacher models
- Public datasets (e.g., ImageNets, CIFAR10)
- Limited query budget

API
Overview: Teacher Fingerprinting Attack

Fingerprinting pairs \( T \) Model

\( x \) Probing Input
\( x' \) Synthetic Input

**Attacker Side**

\[ \text{argmax}_i T'(x_i) : \text{"Airplane"} \]
\[ \text{argmax}_i T'(x'_i) : \text{"Airplane"} \]

\[ f(x) : \text{"Bird"} \]
\[ f(x') : \text{"Bird"} \]

\[ S(x) \approx S(x') \]

**Insight:**

Fingerprinting pairs
Similar latent representation
Same API responses

**Victim Side**

API
\[ k \text{ components} \]
\[ F_T(\cdot) \]

API
\[ k \text{ components} \]
\[ F'_T(\cdot) \]

Student Model \( (S) \)
Attack Stage 1: Synthetic Input Generation

- Solving constrained optimization

  \[ \text{Original problem (Constrained)} \]

  \[ \begin{align*}
  \mathbf{x}' &= \arg\min_{\mathbf{\mathbf{x}}} \| \mathcal{F}_T(\mathbf{\bar{x}}) - \mathcal{F}_T(\mathbf{x}) \|_2 \\
  \text{s.t. } \mathbf{\bar{x}} &\in [0, 255]
  \end{align*} \]

  \[ \tanh(w) = \frac{2\mathbf{x}}{255} - 1 \]

  \[ \mathbf{w}' = \arg\min_{w} \left\| \mathcal{F}_T \left(255 \times \frac{1}{2} (\tanh(w) + 1) \right) - \mathcal{F}_T(\mathbf{x}_i) \right\|_2 \]

  \[ \text{Converted problem (Unconstrained)} \]

  Adam optimizer

  Learning rate: 0.001

  \#Iterations: 30,000
Attack Stage 2: Teacher Model Inference

- Inference Metric
  - Matching proportion:

  \[
  \frac{\#\text{Matched Responses}}{\#\text{Fingerprinting Pairs}}
  \]

  | Actual teacher model | VGG19 | AlexNet | AlexNet (PTCV) | DenseNet121 | MobileNetV2 | ResNet18 | VGG16 | VGG19 |
  |----------------------|-------|---------|----------------|-------------|-----------|---------|-------|-------|
  |                      | 0.08  | 0.08    | 0.14           | 0.07        | 0.07      | 0.24    |       | 0.91  |

Inference: VGG19

> Threshold?

Y

NULL

N
Effectiveness of Our Proposed Attack

- Basic setup

  # fingerprinting pairs: 
  100 for each candidate

  # student models: 
  6 datasets * 7 teacher models * 3 student FCN architectures
Effectiveness of Our Proposed Attack

• Basic Results

| Correctly inferred | Inferred as "NULL" |
|--------------------|--------------------|
| w/ known teacher model | w/ unknown teacher model | w/o transfer learning |
| 100% (126/126) | 72.2% (13/18) | 86.1% (31/36) |
Effectiveness of Our Proposed Attack

- **Impact of Query Budget**

  #Fingerprinting pairs for each candidate

100% inference accuracy
100% matching proportion

(False matching)
Towards More Robust Attack

• Supporting Set
  Remove the most frequently matched elements
Towards More Robust Attack

• Supporting Set
  Remove the most frequently matched elements
Towards More Robust Attack

- Supporting Set

Remove the most frequently matched elements

\[ |\text{Supporting Set}| \geq \left\lfloor \log_2 \frac{1}{\alpha} \right\rfloor + \left\lfloor \frac{\log_2 \frac{1}{\alpha}}{c - 1} \right\rfloor \]
Towards More Robust Attack

Most inference results are indeed invalid when \# query is small

| Query Budget | probing: VOCSegmentation | probing: MNIST | probing: CelebA | probing: Random Noise |
|--------------|--------------------------|----------------|-----------------|----------------------|
|              | original | robust | original | robust | original | robust | original | robust | original | robust | original | robust |
| 1            | 39.68%   | (50/126) | – (0/0) | 0 (0/126) | 42.06% | (53/126) | – (0/0) | 0 (0/126) | 45.24% | (57/126) | – (0/0) | 0 (0/126) | 19.84% | (25/126) | – (0/0) | 0 (0/126) |
| 2            | 61.11%   | (77/126) | – (0/0) | 0 (0/126) | 57.94% | (73/126) | – (0/0) | 0 (0/126) | 57.94% | (73/126) | – (0/0) | 0 (0/126) | 29.37% | (37/126) | – (0/0) | 0 (0/126) |
| 5            | 84.13%   | (106/126) | – (0/0) | 0 (0/126) | 69.84% | (88/126) | – (0/0) | 0 (0/126) | 80.95% | (102/126) | – (0/0) | 0 (0/126) | 42.06% | (53/126) | – (0/0) | 0 (0/126) |
| 10           | 95.24%   | (120/126) | 100.00% | (32/32) | 25.40% | (32/126) | 80.95% | (102/126) | 100.00% | (19/19) | 15.08% | (19/126) | 89.68% | (113/126) | 100.00% | (3/3) | 2.38% | (3/126) |
| 20           | 97.62%   | (123/126) | 100.00% | (97/97) | 76.98% | (97/126) | (84.92% | (107/126) | 100.00% | (52/52) | 41.27% | (52/126) | 96.83% | (122/126) | 100.00% | (87/87) | 69.05% | (87/126) |
| 50           | 100.00%  | (126/126) | 100.00% | (125/125) | 99.21% | (125/126) | 90.48% | (114/126) | 100.00% | (96/96) | 76.19% | (96/126) | 99.21% | (125/126) | 100.00% | (117/117) | 92.86% | (117/126) |
| 100          | 100.00%  | (126/126) | 100.00% | (126/126) | 100.00% | (126/126) | 96.03% | (114/114) | 100.00% | (114/126) | 90.48% | (122/122) | 100.00% | (122/122) | 96.83% | (122/126) | 65.08% | (82/126) | 100.00% | (41/41) | 32.54% | (41/126) |
Enhanced Model Stealing Attack

Attack Dataset + "VGG16 Pretrained Model, v1.00 by PyTorch" → Surrogate Model
Enhanced Model Stealing Attack

- Best performance if starting from a matched teacher model
Feasible Countermeasures

• Input distortion
  - Perturb the patterns in synthetic inputs

• Injecting neuron distances [Wang et al. 2018]
  - Deviate the student model’s feature map from the teacher model’s
Conclusion

- We propose a simple and efficient attack to infer the teacher model used by transfer learning.

- Our attack can efficiently identify the teacher model.

- Our attack can help perform further advanced attacks.
Thanks!

Q&A

Yufei Chen
yufeichen8-c@my.cityu.edu.hk