Model Stealing Attack for Trustworthy Machine Learning (TML25_A2_21)

Nima Dindarsafa (7072844), Samira Abedini (7072848)

Abstract

This report presents our Model Stealing Attack against a black-box ResNet-based encoder protected by the Batch4Batch (B4B) defense, which adds noise to API output representations. Our goal was to minimize the L2 distance between our student model's 1024-dimensional outputs and the target encoder's on a private test set. Using ModelStealingPub.pt and API-queried representations, we tested four methods, achieving a best L2 score of 5.46 with a modified ResNet18 model. This report details our approach, implementation, results, and insights, demonstrating robustness against B4B noise through knowledge distillation, Siamese losses, and tailored augmentations.

Introduction

Model stealing attacks aim to replicate a black-box model's functionality by querying its API and training a student model on the outputs [3]. In this assignment, we targeted a ResNet-based encoder protected by the B4B defense, which introduces noise to 1024-dimensional output representations [1]. Our objective was to minimize the L2 distance between our student model's outputs and the target's on a private dataset, using ModelStealingPub.pt and API queries. We tested four methods, leveraging PyTorch for training and ONNX for submission, achieving a best L2 score of 5.46.

Methods

We explored four methods to steal the B4B-protected encoder, addressing its noise through model architecture, normalization, and loss functions:

1. Pretrained ResNet20: Used a pretrained ResNet20 (CIFAR-10) to leverage existing features, fine-tuned with knowledge distillation loss (DISTILLATION_WEIGHT=1.0) on. The L2 score was 6.47, limited by pretrained weight misalignment with B4B noise.

2. ResNet18 with Modified Inputs: Adopted a higher-capacity ResNet18, modified for 3×32×32 inputs (conv1: 3×3 kernel, stride 1, padding 1; maxpool: Identity). Trained on 1000 images (750/250 split) with (DISTILLATION_WEIGHT=1.0) and (INVARIANCE_WEIGHT=0.5).

3. Changed Mean and Std: Adjusted normalization to CIFAR-10 parameters (MEAN=[0.4914, 0.4822, 0.4465], STD=[0.2470, 0.2435, 0.2616]), retraining ResNet18. Yielded L2 score of 5.55, slightly worse than the baseline ResNet18.

4. L2 Normalization in Outputs: Applied L2 normalization to ResNet18 outputs (F.normalize(out, dim=1)) to align scales, but resulted in a poor L2 score of 25.46, likely disrupting representation alignment.

Implementation

Our codebase, stored in TML25_A2_21, is modular:

- **main.py**: Orchestrates API queries, training, and ONNX submission with a 750/250 train-validation split.

-**config.py**: Defines BACKBONE_TYPE="resnet18", LEARNING_RATE=3e-4, DISTILLATION_WEIGHT=1.0, INVARIANCE_WEIGHT=0.5, and augmentations (RandomRotation, RandomErasing).

- **train.py**: Trains with KD and Siamese losses to counter B4B noise.

- **cnn_encoder.py**: Implements ResNet18 with modified conv1 and 1024-dimensional output.

- **query_api.py**: Manages 1000-image API queries and submission.

- **dataset**/: Loads data/ModelStealingPub.pt, applies augmentations.

Results

Results are summarized below:

The best score (5.46) was achieved with ResNet18, modified inputs, and Siamese loss, stored in stolen_model_1.pth. The score reflects the immediate scoreboard (30%); the final 70% is revealed post-deadline.

Method	L2 Score
Pretrained ResNet20	6.47
ResNet18 Modified	5.46
Changed Mean/Std	5.55
L2 Normalized Outputs	25.46

Table 1: L2 scores on the immediate scoreboard (30% of test set).

Conclusion

Our best model (L2 5.46) used ResNet18 with modified inputs, CIFAR-10 normalization, and Siamese loss to counter B4B noise, approaching the top score of 4.88. The modular codebase facilitated experimentation.

References

[1] Dubiński, J., Pawlak, S., Boenisch, F., Trzcinski, T., & Dziedzic, A. (2023). Bucks for Buckets (B4B): Active Defenses Against Stealing Encoders. In Advances in Neural Information Processing Systems, 36 (NeurIPS 2023). https://proceedings.neurips.cc/paper_files/paper/2023/hash/ad1efab57a04d93f097e7fbb2d4fc054-Abstract-Conference.html

[2] Liu, Y., Jia, J., Liu, H., & Gong, N. Z. (2023). StolenEncoder: Stealing Pre-trained Encoders in Self-supervised Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://arxiv.org/abs/2201.05889

[3] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531. https://arxiv.org/abs/1503.02531