TML Assignment 2

Model Stealing in Self-Supervised Learning

This assignment implements a Model Stealing attack using the access to a victim encoder through API. The victim encoder is said to be protected by B4B. Based on this fact, we note that the representations returned on querying the victim encoder are first applied to different transformations and then returned to the attacker. We make use of this information for our approach. The stolen encoder, model in onnx format submitted for evaluation, and all representations can be found [here](link)

Data accessible to the adversary
- Partial training data of the victim model.
- Information on the structure of the dataset.
- API access to the victim encoder.
- The victim encoder is protected using B4B defense.

Approach used
The implemented approach can be accessed in the [**stealing.py**](stealing.py) file. Bucks for Buckets (B4B) is an active defense strategy used to protect the victim encoder. As per B4B defense, when we query images to the victim encoder, transformations like Affine, Pad + shuffle, Affine + Pad + Shuffle, and Binary are applied to the actual image representations. We leverage this fact and also consider that augmented versions of an image when queried to the victim encoder return similar representations. (This is a further description of the solution that we remove so that this cannot be copied ...)

Why only X queries per image?
The query limit per API is 100. The length of the provided dataset is 13,000. ....

Why we used X images provided for stealing the encoder?
The number of stealing queries can be less than ...

Results
The above approach results in an L2 distance of X for Y% of the private data.
On using InfoNCE loss contrastive loss function, we obtain an L2 distance of around X, which was further reduced after hyperparameter tuning to ...
Our solution was Xth on the leaderboard.

Other ideas on implementation that our approach has leveraged
- Using other stolen model architectures ...
- Using contrastive loss functions ...
- Using no augmentations for the provided dataset ...

Files and there descriptions

stealing.py - ...

defense.py - ...

augment.py - ...

...