Assignment 2 (model stealing)

Implement and execute a model stealing method against the protected Self-Supervised Learning (SSL) encoder protected behind an API. Your goal is to achieve the lowest L2 distance of output representations on private images between your stolen copy and the attacked encoder.

- You are provided with a subset of the training data (MODEL STEALING PUB) of the victim model. The structure of the dataset is similar to the previous assignment, you can reuse the code from your previous work.
- You can query the model using a dedicated API, details are below. The model is protected using B4B.
- Your submission will be an ONNX file, as well as the output string you get after launching the local API. The string is necessary, and if you send an incorrect one to the evaluation API, expect your L2 error to be extremely high.
- The scoreboard works the same as for the first assignment.
- The minimum working examples are to be found HERE. You can also use the provided jupyter notebook that might be easier to use for this assignment (as you need to save the states between the different calls to the APIs): https://github.com/sprintml/tml_2024/blob/main/example_assignment_2.ipynb

Task artifacts

1. MODEL STEALING PUB

API service

You will get access to the encoder via API. You can request a new API every 4 hours, every time you'll get a different encoder to steal. The API expects 1000 images as an input and allows for one query every minute. You are limited to 100k queries per API, after crossing that threshold you will not get anything from the API. This is to simulate the real-world scenario, where you want to use as few queries as possible to steal the encoder, as the queries are pricey.
To request a new API follow the example HERE.
Important: note the seed you get back, you'll need it for submission. If you steal the encoder perfectly, but provide the wrong seed, your L2 will be on par with random submission.
To query the API follow the example HERE.

Evaluation

The encoder you steal takes an image of 3x32x32 as an input and outputs a representation of size 1024 (after transformations). The evaluation endpoint (for this task, http://34.71.138.79:9090/stealing) expects you to provide an ONNX version of your stolen copy, as well as the seed you see after launching your local API. A couple of things can go wrong on your side:

- Incorrect input dimensionality expected by your model (has to be 3x32x32)
- Incorrect name of the input (see example submission please)
- Incorrect output dimensionality expected by the evaluation endpoint (has to be 1024)
- Wrong seed (both the format and the value)

We provide you an example submission, under *example_assignment_2.py*, as well as the assertions on the side of the evaluation server regarding your submitted file. We encourage you to first run the assertions, and only then submit your model for evaluation.

The evaluation procedure is as follows:

1. We load your model
2. We obtain representations on images from a private dataset
3. We calculate the L2 distance between the submitted model's representations and the victim model's representations

Same as in the first Assignment, you're limited to only one submission per hour, and the immediate result you get back is evaluated on 30% of the data, and the final (after the deadline) will be on 70% of the private data.

README

Please, write the README.md in your GitHub repository for this assignment. Describe how you solved the assignment and point to the most important files and pieces of code. The final grade will depend heavily on how you describe your implementation.