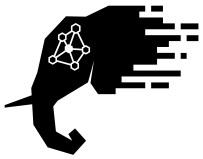

Task 2: Sybil Attack

New day, new task. In your mailbox you see an email from Mike.

> *Hi XYZ,*
>
> *Very good job on stealing our model yesterday. We now can adjust the defense's parameters to prevent this attack vector. Maybe hiring you was not as bad of an idea after all? :)*
>
> *The competition has one more trick up their sleeve. As you've learned yesterday, by using specific queries to our API they can utilize the model's outputs to train a copy of our model multiple times cheaper. Our defense deals with that, however, they can perform an attack known in literature as the Sybil Attack to avoid the adaptive noising. That is why we added a protective transformation to original representations - unique for each system user.*
>
> *We additionally constrained each user to the N=2000 query limit. We want to test if this measure is enough to secure our model even when an adversary uses a Sybil Attack and tries to somehow remap the representations to omit the query limit.*
>
> *You will have access to the system endpoints that return representations after protective transformations. I have attached the information about the data end endpoints you should use to test the defense robustness.*
>
> *Additionally, we want to test two different types of defensive transformations to assess which one prevents this kind of threat the best. Your task will be to try to break both of them. They are fully independent from each other, I'd recommend focusing on one of them at the time.*
>
> *GL,*
>
> *Mike*

From your diligent studies, you remember that the Sybil attack is a type of security threat in which a single adversary obtains access to multiple accounts to bypass system security limitations for a single user (not true, in reality, you just googled/ GPT-ed it already). With that precious piece of information, you started to think deeply about the task while scanning the attached information...

Endpoints

For each type of transformation we provide 2 types of endpoints: home (**A**) and defense (**B**). These endpoints receive as an input query the image id (or a list of ids), and your team's api key. Every endpoint returns the representation of the image from an SSL vision encoder after some transformation. You have the limit of **2000** queries (image ids) per endpoint.

An example query is provided here.

Resetting

You will use up the query limit regularly. At any point you can reset both endpoints.

An example resetting request is provided here.

Transformations

The transformations used in this task are secret. The input to the transformation is the clean representation from the hidden underlying encoder. The output of the transformation is a new representation vector. After you reset an endpoint a new transformation is drawn for the endpoint.

Dataset

You have access to the SybilAttack dataset: image_id (int), image (PIL.Image.Image), image_label (int).

See boilerplate code for loading here.

Submission

The transformed representations of images returned from **A** are your target representations. The goal is to provide target representations for all images in the dataset, i.e., **20k** images.

You achieve this goal by querying the endpoints **A** and **B** and constructing an inverse transformation between them.

The submission should be an **.npz* file with two fields: *ids* – an array of 20k indices from the SybilAttack dataset **in the same order** (your submission will be rejected if there is discrepancy), and *representations* – a 20k x [representation size] matrix (the representation size has to match the one returned by an endpoint, your submission will be rejected if there is mismatch in the shape of submitted matrix).

Here's a code with a sample submission: here.

Evaluation

Your submission's score is the **mean L2 distance** between the representations transformed using home (endpoint **A**) transformation and the submitted representation.

Once you submit, **A** and **B** endpoints are reset. You can always reset **A** endpoint without submitting, but keep in mind that all your progress for the current submission will be discarded, since the transformation to invert changes once you reset an **A** endpoint.

Tips

Here are some things to take into consideration when designing your attack:

- Querying strategy: maybe some images can provide more information than the others?
- Inverse transformation: what can you use here?