**Requirements**

Stored in requirements.txt, Python dependencies are:

h5py

matplotlib

numpy

pandas

git+https://github.com/szagoruyko/pyinn.git@master

torch

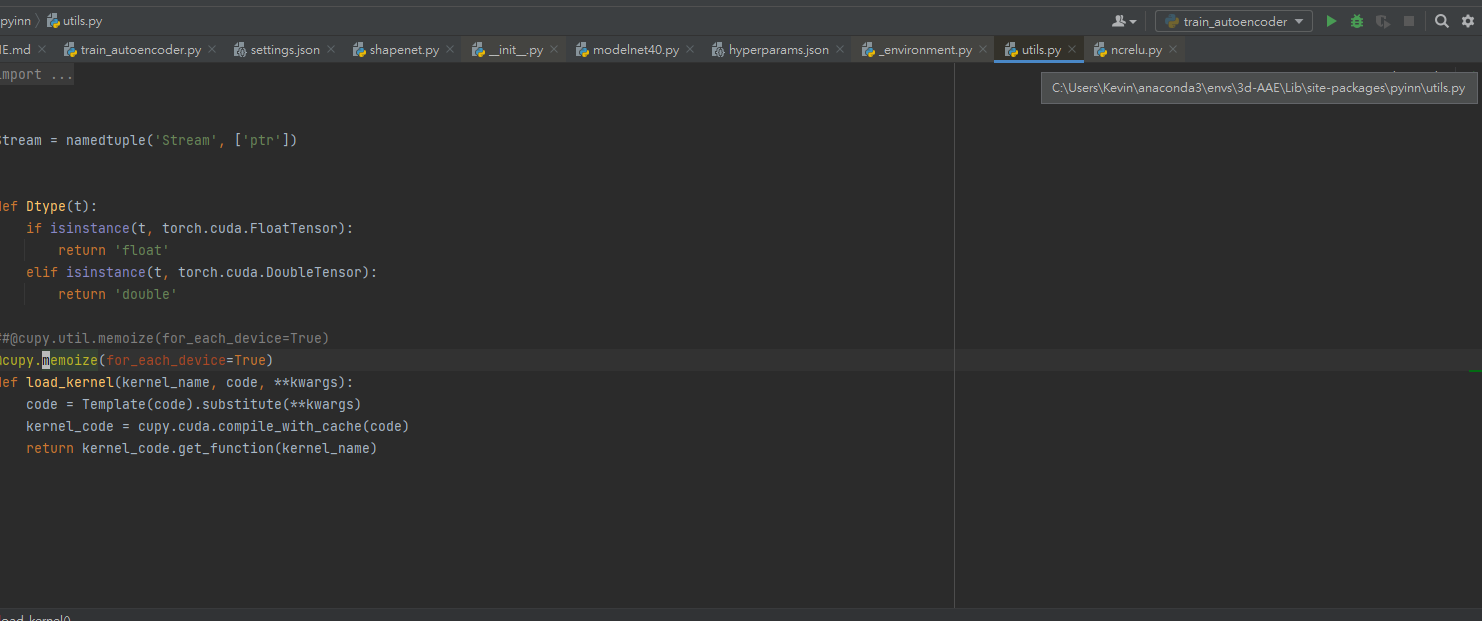
and for visualization further libraries should be included:

open3d

Download cuda and Cudnn and if you use latest torch version

Modify code in pyinn\utils.py

Change ”cupy.util.memoize” to ”cupy.memoize”



## Usage

### Training

Run an experiment with:

python3.6 experiments/[(scaphoid)train\_aae.py](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/experiments/(scaphoid)train_aae.py) --config [settings\_bone.json](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/experiments/settings_bone.json)

where

[(scaphoid)train\_aae.py](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/experiments/(scaphoid)train_aae.py) -one of the training scripts from the experiments directory

[settings\_bone.json](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/experiments/settings_bone.json) - JSON file with training settings and hyperparameter values

* The input method of datasets should be modified according to the datasets interface, see the #Datasets part in [(scaphoid)train\_aae.py](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/experiments/(scaphoid)train_aae.py)
* Be aware that the interface should also be modified for the code in evaluation part

### Evaluation

1. python3.6 evaluation/[(scaphoid)find\_best\_epoch\_on\_validation.py](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/evaluation/(scaphoid)find_best_epoch_on_validation.py) --config settings.json

Calculates JSD distance between sampled point clouds and the validation set and presents the best epoch.

1. python3.6 evaluation/[(scaphoid)generate\_data\_for\_metrics.py](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/evaluation/(scaphoid)generate_data_for_metrics.py) --config settings\_g.json(only use cpu)

Produce reconstructed and generated point clouds in a form of NumPy array to be used with validation methods from ["Learning Representations and Generative Models For 3D Point Clouds" repository](https://github.com/optas/latent_3d_points/blob/master/notebooks/compute_evaluation_metrics.ipynb)

1. after running [(scaphoid)generate\_data\_for\_metrics.py](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/evaluation/(scaphoid)generate_data_for_metrics.py), you will find the results in results\_root(you specified in the json file), you can use [print\_result.py](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/myutils/print_result.py) in myutil directory to visualize the point clouds

### Other function

1. [(sacphoid)generate\_interpolated\_pointclouds.py](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/evaluation/(sacphoid)generate_interpolated_pointclouds.py) is used to generate interpolated point clouds.
2. [make\_synthetical\_anomalies.py](https://github.com/kevinlai58/3d-AAE-master/blob/master/src/myutils/make_synthetical_anomalies.py) is used to generate synthetical anomalies point clouds.