This documentation is prepared to accompany an ongoing study in collaboration between MIT PV Lab and NIST. This document will be updated with the status imminently.

Installation

To run the code, please install the following:

1. Anaconda (https://www.anaconda.com/distribution/), which already consists of NumPy, matplotlib, scikit-learn.

There are several packages and/or libraries that need to be installed to run the notebook:

1. NumPy (https://docs.scipy.org/doc/numpy/user/install.html)
2. Matplotlib (https://matplotlib.org/users/installing.html)
3. Scikit-learn (https://scikit-learn.org/stable/install.html)
4. TensorFlow (https://www.tensorflow.org/install)

OR clone the following repository: pip install -r requirements.txt

Datasets

There are two datasets we need in the parent folder:

1. A combiview text file with theta and XRD diffraction values for all samples.
2. A text file with combiview mapping of composition and temperature with XRD in the first file.

Workflow

There are two python executables that need to be run. First, *vaecluster_loop.py* is used for data extraction, data selection, data encoding/decoding, clustering and saving the results. Second, *class_averaging.py* is used to read the resulting data and summarize those results. A general overview of the workflow is mentioned below:

vaecluster_loop (runtime – 6 hours in my personal computer for 150 loops):

1. We read the text files limiting XRD data and composition/temperature to the region of interest (using numpy, text reading and list operations)
2. For data input in both variational autoencoder and clustering, we perform normalization (using scikit-learn).
3. We iteratively encode and decode data to minimize reconstruction loss (using tensorflow with dense layers and ReLU activation)

4. We identify and group outputs from VAE using clustering techniques – k-mean, gmm and spectral clustering (using scikit-learn).
5. Before saving the clustering labels for each composition-temperature for each run, we ensure the labels are in ascending order in the order of occurrence and aligned with sample numbering (using list operations and numpy).

class_averaging (runtime – short, few minutes):

1. We read the data from *vaecluster_loop* and count occurrences of each cluster labels for each composition-temperature combination. A probability profile is built for each label using these numbers out of total runs of 150 (using numpy and list operations).
2. For each method, we repeat the step one and save the files in a text format. These files are provided to the leading scientist of the full study.

Authors

Janak Thapa (MIT), with the help from Shijing Sun (MIT), Felipe Oviedo (MIT/Microsoft) and Zekun Danny Ren (SMART)