ANN-CI User Manual

Abstract:

The research group of Professor Debashree Ghosh focuses on the interdisciplinary subject of chemical physics. One area of focus for the group is the development of novel methodologies using cutting-edge tools like machine learning in the field of selected CI. These new techniques reduce the first principal-based method's enormous computational cost without sacrificing accuracy. This manual demonstrates how to use a handful of Prof. Ghosh's methodologies.

About ANN-CI:

ANN-CI is machine learning (artificial neural network) based software for selected configuration interaction method that can be applied to strongly correlated molecular systems. There are several ways of performing the calculations on model and molecular systems. Furthermore, active learning based approaches are also there in the module that can speed up the convergence and make the application more efficient and black box.

System requirements for local machine :

A Linux-based system is required to run the codes. If execution is performed on a local machine, the following requirements are necessary for a smooth experience:

1. CPU- intel i5 (or higher) / AMD Radeon 5 (or higher)
2. GPU- NVidia GPU (4GB / 128-core or higher)
3. RAM- 8GB+
4. Storage - 512GB+ (preferably SSD)

Required Libraries The following libraries are required to compile the codes -

1. Python3.6
2. PyTorch
3. Numba
4. f2py3
5. Lapack
6. Scikit-learn
7. Pandas
8. Matplotlib
9. Numpy

10. Rdkit

Required Files

There are two files required to run this code

1. Training data set (Data set for learning of ANN model)
2. Input file. (Contains all the information to execute the calculation)
3. Test data set (Unlabelled data set, which will be predicted by trained ANN model).

Running Instructions

On a Local Machine

1. Clone the Repository:
   Ensure you have the code from the GitHub repository.
2. Activate the Conda Environment:
   Run the following command to activate the Conda environment where all dependencies and packages are installed:
   ```
   conda activate your_environment_name
   ```
3. Execute the Code:
   Use the command below to run the script, replacing input.in with the name of your input file.
   ```
   python exe.py input.in
   ```

On an NSM HPC Cluster

Interactive job:

1. Load the appropriate module for ANN-CI:
   Eg.
   ```
   module load cdac/MSCC/ann-ci
   ```
2. Execute the Code:
   Use the command below to run the code replacing input.in with the name of your input file.
   ```
   exe.py input.in
   ```

Batch Job (Using SLURM)

1. **Create a SLURM Script:** Save the following SLURM script to a file (e.g. job.sh):

```
#!/bin/bash

#SBATCH --job-name=ann
#SBATCH --nodes=1
#SBATCH --partition=cpu     ### Please go through the user manual of the respective HPC cluster and mention the partition name accordingly
#SBATCH --time=01:00:00
#SBATCH --output=%x_%j.out
#SBATCH --error=%x_%j.err

cd $SLURM_SUBMIT_DIR
module load cdac/MSCC/ann-ci
exe.py input.in
```

2. Submit the SLURM Script:

   ```
   sbatch job.sh
   ```

Expected Outputs

There is a total of 10 output files generated after successful calculations. The six main files are

1. input_file.in.out # Main output file, which contains information on subspace size, energy, and spin value with each iteration.

2. input_file.in.out.basis # Configurations of final sub-Hilbert space
3. input_file.in.out.ci # CI coeffcienet corrosponding to configurations
4. Input_file.in.out.model.pth #Final optimized ANN model
5. input_file.in.out.error.dat # Train and test error at each AL iteration
6. input_file.in.out.TrainData_subSpace.csv # Train data set generated during calculation These files are essential for system analysis.

   Apart from that, there are four more files for deeper analysis of machine learning performance.

7. input_file.in.out.predictData.csv
8. input_file.in.out.accVsPreTest.dat
9. input_file.in.out.accVsPreTrain.dat
10. Input_file.in.out.enrich.csv

Important Links:

Github