PHYMOL Kick-off conference Hands-on ML tutorial

Dahvyd Wing
University of Luxembourg

September 2023

1 Installation

These instructions create a python virtual environment with PyTorch and SchNetPack[1].

1. Install anaconda and open an anaconda prompt
2. Create a virtual environment:

   `conda create --name ml_tutorial python=3.11`
3. Activate the new environment:

   `conda activate ml_tutorial`
4. Test to make sure that the pip, python, etc. all point to executables in the conda environment:

   If you are using linux run `which pip`. If you are using windows, run `where pip`. Check that the first/only executable is in the anaconda virtual environment path.
5. Install the following packages

   `pip install schnetpack spyder tensorboard`

You don't need to use spyder if you are more comfortable with a different IDE (especially vim or emacs), but you have to ensure that they are using the right virtual environment/right packages in the case of pycharm or possibly vscode.

2 Example: A simple neural network to model a 1D Lennard-Jones potential

Here we show a very simple feed-forward, fully-connected neural network that is trained to model a 1D Lennard-Jones potential. These scripts are very simple so that you can easily see the components of a training a neural network.

1. `lj_1_overfit.py` trains a model with so many parameters and so little data that the model will overfit. The point is just to show that the mean squared error of the model on training data is not a good measure of the actual performance of the model.
2. `lj_2_with_validation.py` changes the previous script slightly by separating the dataset into a training and validation set. It produces a plot of the training process so that you can see the difference between training and validation loss.
3. `lj_3_hyperparameters.py` changes the previous script by giving you the ability to easily change hyperparameters of the model in the dictionary at the top of the script. Open up a tensorboard plot and play with the hyperparameters and see how this affects the model. See the comments at the top of the script for more information.

3 Kick-start for using schnetpack 2.0

The tutorials in Schnetpack 2.0 are well done. The only thing I add here is a modified dataset script that can read a numpy .npz file and turn this into a dataset that schnetpack can read.

1. Download the md17 ethanol database [2] at `http://quantum-machine.org/gdml/data/npz/md17_ethanol.npz` (this file is 219 MB) and place this in the `ml_tutorial` folder
2. Run `generate_schnet_dataset.py` to generate the ethanol database from the numpy .npz file
3. Continue on from this script directly to schnetpack's tutorial on training using energies and forces code block #5. The only change to that code that needs to be made is replacing every instance of MD17.energy with the string "energy" and MD17.forces with the string "forces".
4. Try running the molecular dynamics tutorial. This will create the folder `ase_calcs` in which you will find the MD trajectory stored in `simulation.traj`
5. In the anaconda prompt run `ase gui` and in the GUI open `simulation.traj`
6. Then go to `tools/movies` and then select play to see the trajectory. Obviously the ethanol molecule will fall apart for such a poorly trained network.

References

1. Schütt, K. T., Hessmann, S. S. P., Gebauer, N. W. A., Lederer, J. & Gastegger, M. SchNetPack 2.0: A neural network toolbox for atomistic machine learning. *The Journal of Chemical Physics* **158,** 144801. ISSN: 0021-9606 (Apr. 2023).
2. Chmiela, S. *et al.* Machine learning of accurate energy-conserving molecular force fields. *Science Advances* **3,** e1603015 (2017).