**# gas-dataset-regression**

This is a project on chemical sensor calibration using ML methods.

**1. Requirements:**

scikit-learn >= 0.18

numpy >= 0.12.0

keras >= 2.0.4 (a pre-requirement is either the latest theano or tensorflow)

pandas >= 0.20

mathplotlib >= 2.0.2

h5py >= 2.7.0 (pre-requirement is an installed hdf5 file system)

**2. Usage**

For different ML model, simply run the corresponding python file and it will show the calibration results.

If you want to use your own dataset, put the file in '/data'. The file must be an excel or csv file with parameters' name in the first line.

**3. About source code**

main.py: generate a plot to compare the performance of different models

read\_data.py: reading data from the file

NN\_class.py: neural network classes

dataprocess.py: some functions used for processing data in this work

histogram.py: used for generate a histogram for observing data distribution

Linear.py: training and test using linear model

DNN.py: training and test using BPNN

TDNN.py: traning and test using TDNN (main objective)

test2.py: some pre-experiments, not useful at all

**4. About the trained model file**

The trained NN model is stored by a hdf5 file called "model.h5". Inside the file, the model's structure is organized as follows:

Top level: Root group with 3 sub-groups, indexes are ‘model\_weights’, ‘optimizer\_weights’, ‘iterations’. These 3 sub-groups contain the weights of neurons, optimizer parameters and iteration information (not useful exactly).

The model\_weights group contains k sub-groups with index of ‘dense\_1’ to ‘dense\_k’. k depends on the number of layers in this model. And there exists a sub-group with the same name in each ‘dense\_i’ group, which contain two dataset ‘bias’ and ‘kernel’, recording the true values of neuron and bias weights.

The optimizer\_weights group contains k sub-groups with index of ‘param\_1’ to ‘param\_k’. k depends on the number of layers in this model.

Example: to achieve the weights in layer 1, using

f = h5py.File(“file path”)

w1 = f[‘model\_weights’][‘dense\_1’][‘dense\_1’][‘kernel’] (result is an ndarray)