pulse2D_MLP_preprocess

March 22, 2019

In [1]:
```
#IMPORT PACKAGES

# Scikit-learn
from sklearn import preprocessing

# data analysis packages
import numpy as np
import pandas as pd

# misc. packages
from random import shuffle  # shuffle elements in a list
import pickle  # pythonic file compression
import h5py
```

In [2]:
```
# read in and preprocess data

X = pd.read_csv('pulse2d_iterinput.txt', sep='\t')  # read in static simulation inputs (
out_dict = pickle.load(open("pulse2d_iteroutput.pkl", "rb"))  # read in simulation outpu
data = out_dict[0]  # read in the first output to get the shape
Nsamples = 1000  # number of samples for all I/O is equal to the number of simulations
Ndomain_nodes = data.shape[0]  # number of 'timesteps', in this case is equal to the num
XNfeatures = X.shape[1]  # static features
YNfeatures = Ndomain_nodes  # only one prediction of interest....(Cmax)
Ydim = (X.shape[0], YNfeatures)  # dimension for 2d output
Y = np.zeros(Ydim)  # initialize 2d dynamic output...

# build X and Y - read in dynamic I/O...i.e. output from the pulse2d model.

start_idx = 0
for i in out_dict:
    data = out_dict[i]
    Y[i,:] = data.Cmax.values

# normalize 2d inputs across features
X = preprocessing.normalize(X,norm='l2', axis=0)

# scale output datasets...this isn't strictly necessary but makes the output more interp
Y = Y/100000
```

In [3]:
```
# shuffle and partition data into train/test sets

# static data
ismpls = list(i for i in range(0,X.shape[0]))  # create list of integers 1:Nsamples (rou
shuffle(ismpls)  # randomize sample/row indices
ismpls = np.argsort(ismpls)
# reorder each dataset using the randomized row indices set above. then, partition each
X = X[ismpls,:]
split_idx = int(np.round(X.shape[0] * 0.9))
X_train = X[:split_idx,:]
X_test = X[split_idx:,:]

# 2d data
ismpls = list(i for i in range(0,X.shape[0]))
shuffle(ismpls)
ismpls = np.argsort(ismpls)
staticd = X[ismpls,:]
Y = Y[ismpls,:]
split_idx = int(np.round(X.shape[0] * 0.9))
X_train = X[:split_idx,:]
X_test = X[split_idx:,:]
Y_train = Y[:split_idx,:]
Y_test = Y[split_idx:,:]
```

In [4]:
```
# write preprocessed data to *.hdf5 file

with h5py.File('pulse2D_MLP_X.hdf5','w') as f:
    x_train = f.create_dataset('X_train', shape=X_train.shape, data=X_train)
    x_test = f.create_dataset('X_test', shape=X_test.shape, data=X_test)

with h5py.File('pulse2D_MLP_Y.hdf5','w') as f:
    y_train = f.create_dataset('Y_train', shape=Y_train.shape, data=Y_train)
    y_test = f.create_dataset('Y_test', shape=Y_test.shape, data=Y_test)
```

In [ ]: