```
### DNN(FCN) for GBAR experiment Monte Carlo simulation data
# 100,000 raw datas of signals by annihilation of upward, downward gravitation and
  background signals each.


import numpy as np
import matplotlib as plt
import random
import uproot
import h5py

import tensorflow as tf
from keras import models
from keras import layers
from keras.utils import to_categorical
from keras.utils.io_utils import HDF5Matrix
import keras.backend.tensorflow_backend as K


filename = "SignalMCwaveform_flatten.hdf5"

X_train = HDF5Matrix(filename, 'X_train')
X_test = HDF5Matrix(filename, 'X_test')
Y_train = HDF5Matrix(filename, 'Y_train')
Y_test = HDF5Matrix(filename, 'Y_test')


# Use gpu as processor
with K.tf.device("/device:GPU:0"):

    # Sequential model
    model = models.Sequential()


    ## Change the number of 'hidden layers' and 'neurons'.

    # The first hidden layer
    model.add(layers.Dense(2048, activation='relu', input_shape=(112*88,)))
    model.add(layers.Dropout(0.2))

    # The second hidden layer
    model.add(layers.Dense(2048))
    model.add(layers.Activation('relu'))
    model.add(layers.Dropout(0.2))

    # The third hidden layer
    model.add(layers.Dense(2048))
    model.add(layers.Activation('relu'))
    model.add(layers.Dropout(0.2))

    # The fourth hidden layer
    model.add(layers.Dense(2048))
    model.add(layers.Activation('relu'))
    model.add(layers.Dropout(0.2))
```

```
    # The fifth hidden layer
    model.add(layers.Dense(2048))
    model.add(layers.Activation('relu'))
    model.add(layers.Dropout(0.2))

    # The final layer
    model.add(layers.Dense(3, activation='softmax'))


    # Compile
    # Change 'epochs' and 'batch_size'
    model.compile(optimizer = 'adam',
                  loss = 'categorical_crossentropy',
                  metrics=['accuracy'])

    model.fit(X_train, Y_train, epochs = 7, shuffle = 'batch', batch_size = 2048,
      verbose = 1)
    test_loss, test_acc = model.evaluate(X_test, Y_test)
    print('test_acc : ', test_acc)


    '''
    # Save the model
    # model.save("FCN_1.h5")
    '''
```