MMAI 5500 Assignment 2

The goal of this assignment is to implement the first part of deep portfolio method presented in the article *Deep learning for finance: deep portfolios* by Heaton, Polson & Witte (2016) and covered on lecture 1 of MMAI 5500.

Submission

The assignment should be submitted as Python 3 code and uploaded to Canvas as a single `PY` file (**not** a Jupyter Notebook) and the trained model. The due date is on June 15 at 8:30am.

Data

The daily closing prices of 124 stocks from the IBB biotechnology index are provided in the file `assing2_data.csv`. The prices have been normalized and span the period from 2015-05-05 to 2021-05-30.

Load the data using the following helper function.

```
def load_data(fname):

    data = pd.read_csv(fname, index_col=0)
    X = data.values[:, 2:].astype('float64')
    years = data['year']
    X_train = X[years < 2020.]
    X_valid = X[years == 2020.]
    tmp = data.index[data['year'] == 2020.]
    tickers = np.array([ticker.rstrip('_2020') for ticker in tmp])

    return X_train, X_valid, tickers
```

Use `X_train` to train and `X_valid` for model selection. The array `tickers` holds the ticker names corresponding to the rows in `X_valid`.

Task

You will train an autoencoder to encode the timeseries of price data. The network architecture is described in the article *Deep learning for finance: deep portfolios* and should be implemented using Keras.

The autoencoder have a single layer of encoder weights, an encoded dimension of 5 (i.e. 5 hidden units) and a single layer of decoder weights. Both weight layers should be regularized with an L2 penalty. The encoder layer should have ReLu activation and the decoder layer sigmoid activation. The loss should be mean squared error (`mse`). The Adam optimizer is recommended.

Use `X_valid` to find the best value for the L2 regularization parameter (lambda).

Once the network is trained and the best value for lambda is found you should select the 10 most communal and the 20 least communal stocks. The degree of communality is measured with the reconstruction loss.

Deliverable

You need to submit a single Python file (`PY` **NOT** `IPYNB`) that does the following:

1. Loads the data.

2. Sets up and trains the autoencoder.

3. Selects the 5 most communal and 20 least communal stocks. The ticker symbols for the 2 sets of stocks should be stored in lists called `most_communal` and `least_communal`.

4. Prints the ticker symbols for selected stocks as follows:

```
Most communal
-------------
STK1, STK2, STK3, STK4, ...

Least communal
-------------
STK6, STK7, STK8, STK9, ...
```

Your code should follow the PEP 8 style guide. See the original PEP 8 style guide, an easier to read version, or a short PEP 8 YouTube intro. Practically, adding a PEP 8 plugin to your text editor (e.g. Falke8) will make it easier to follow to style guide.

Grading

For full marks the submitted code needs to be bug free, execute the 5 steps described under **Deliverable**, select the same stocks (with at least 80% overlap) as a reference solution (unseen by students), and follow the PEP 8 style guide.

Help

See MMAI 5500 lecture 1 slides.

See the Keras blog about autoencoders for hints about the implementation and the Keras model API for ideas about how to train and get the losses for individual stocks.

Good luck!