MMAI 5500 Assignment 3

This assignment continues where assignment 2 ended. The goal is to implement the second 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.

During the first part you used an autoencoder to select the subset of socks making up a portfolio. Now, it is time to find the relative weights of the stocks in the portfolio.

Supporting code for the parts from assignment 2 is provided in the file `MMAI5500_Assignment3_supporting_code.py` .

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 July 13 at 8:30am.

Data

The daily closing prices of 119 stocks from the IBB biotechnology index are provided in the file `assing3_data.csv` , and the closing prices for the IBB biotechnology index itself are provided in the file `assign3_benchmark.csv` . The prices have not been normalized (in contrast to assignment 2) and span the period from 2016 to end of 2020.

Use the arrays `*_train*` (e.g. `X_train` , `Y_train` , and `Y_train_mod`) to train and `*_valid*` for model selection. The array `tickers` holds the ticker names corresponding to the rows in `X_valid` . The arrays `Y_train` and `Y_valid` hold to IBB index for the period corresponding to `X_train` and `X_valid` . The arrays `Y_train_mod` and `Y_valid_mod` hold the modified IBB index. The shapes of the arrays are described in the function `load_modify_normalize()` in the supporting code.

Task

In the paper they compute a dynamic portfolio weights (called a portfolio map) using a neural network with one hidden layer. Here, we will simplify the task a little and instead compute static weights using regression.

1. Load the data using the function `load_modify_normalize()` (see supporting code).
2. Use the method developed in assignment 2 (see supporting code) to select stocks for the portfolio (`train_autoencoder()` and `select_portfolio()`).
3. Use `X_train_port` , `Y_train_mod` , `X_valid_port` and `Y_valid_mod` to train a linear regression model using Keras. The weights to this model will be weights for the stocks in the portfolio.
4. Use the weights from step 3 to predict the portfolio performance from the `X_valid` .
5. Plot the predicted performance and compare it to both the modified and unmodified IBB biotechnology index (all three in the same plot).

Deliverable

You need to submit a single Python file (`PY` **NOT** `IPYNB`) that executes the 5 steps described above under **Task**.

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 **Task**, 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!