Big Data in Finance II Group Assignment

Instructions: Please submit both your code, output such as graphs and tables, and written answers to the conceptual questions. We strongly recommend that you submit everything as one Jupyter notebook. Use markdown cells to include written answers. At the begining of the answer to each separate question, place a markdown cell that states the question.

You are welcome to use any software for data analysis, although Python (in particular, the Keras/tensorflow and sklearn packages we have used in class) is recommended.

Grading criteria: Many of the tasks in this assignment are open-ended and there is no single right answer. This is deliberate – we will not grade you exclusively on the results of your algorithm. Rather, we will award high marks to groups who set up the problem carefully, follow a rigorous process, document their reasoning, and give good economic intuition in their answers.

Questions:

1. Load the dataset CPZ_data_mini.csv included in the week 3 course pack. This contains a subset of data from the paper by Chen, Pelger and Zhu (CPZ) on your syllabus.

The data are monthly, as recorded in the “Date” column. The “permno” column is a stock identifier. Returns are in the “ret” column and are contemporaneous – for example, the return recorded at date “1970-01-01” is the return between December 1969 and January 1970. The remaining columns are the 10 most important stock (micro) characteristics from CPZ, as well as 8 macro indicators from the paper by Welch and Goyal that you studied in Big Data in Finance 1.

2. Choose 5 of the included micro variables to investigate in detail. For each of them, give a definition using the documentation in CPZ, and provide some economic intuition for why they may be predictive of stock returns.

3. Set the whole dataset up for machine learning by

a) defining a vector **y** of returns that we want to predict

b) defining a matrix **X** of features

b) splitting the sample into a training, validation and test set

c) applying an appropriate normalisation to the features (e.g., making sure each variable has mean zero and variance one in the training data)

4. Pick a baseline neural network architecture and its hyperparameters using your knowledge from our class, and train this network on the data for 20 epochs. Use

mean-square error as your loss function. Try a range of different learning rates alpha and repeat the training process. Which values of alpha yield a learning curve (i.e., a plot of the average mean-square error loss in the training data) that looks satisfactory?

5. Examine the loss on the validation data and compare it to the training loss. Would you suggest any changes to the architecture based on this comparison? Give an intuitive explanation for your answer.

6. Document the performance of your preferred architecture on the test data. Carefully select, justify and calculate a good metric of out of sample performance in return prediction.

7. Fit a penalised linear model (LASSO) to the same data. Compare its test data performance to the neural network.

8. Suppose somebody tells you to collect 20 more micro or macro variables that can predict returns and are not in this dataset. How would you choose those variables?