Stock Market Trading with LSTM Analysis

Franklin Doane
dept. of Computer Science
Tennessee Technological University
Cookeville, United States
fgdoane42@tntech.edu

I. Dataset Creation

A. Get list of stocks by market cap

B. Generate stock list

C. Download data

D. Store data in csv files

II. Training

1) Load and unpack config: Config loaded as a json object .

A. Load dataset

1) Init dataset object: The dataset object is initialized and given config. Then the load dataset method is called which does the following.

2) Verify files: Loads all csvs in dataset and verifies a minimum length. A legacy feature at this point.

3) Split files into task groups: List of valid files is split into groups to have each group loaded with threads.

4) Open dataset thread pool: Open thread pool to load files.

5) Load csv files: Open every file from directory listed in config. Load as pandas dataframe with date index column and put them all in a list.

6) Calculate indicators: Calculate all additional features listed in configs. Cut out any data at the front of the dataframes that is missing these features (e.g. for rolling avgs).

7) Create date feature: 1hot encode day of the week and add the new weekday features to all dataframes.

8) Slice data into training examples: Use rolling window to cut each dataframe into training example dataframes with n day history. All examples get added to a list and labels are made into a dataframe of their own.

9) Normalize examples: Each dataframe training example is normalized using norm function from config.

10) Create tensor input: Each dataframe is converted to numpy then a tensor with the datatype and device listed in the config. All labels are converted a single tensor as well.

11) :

12) Dynamically load model class: The Python file containing the model class is loaded by name using importlib.

13) Init model: The pytorch model is initialized. The custom model class is given a name, starting LR, and input feature count from dataset.

14) Set optimizer: