Real time AI-driven stock price prediction

Problem statement: To make a toy AI model which can make financial decisions with minimal human intervention

Dataset used: BTC Minute Historical Data - BTC2020min.csv

Tech Stack:
1. Numpy, pandas, scikit-learn -> for data loading, processing and cleaning
2. Tensorflow, Keras -> For building and training the neural network

Proposed solution:
I propose a TCN(Temporal convolution network) with a multihead attention layer for the prediction purposes. We take a window of 100 mins and our horizon of prediction is just the next minute's closing price. We are actually predicting log of closing values to bring numerical stability. Then we apply inverse_transform to get predicted closing values.

Feature engineering:

- *Close*: simply the closing price of the bitcoin after every window (100 in our case)
- *rv_5* : The realized volatility (standard deviation of log returns) of the data over the past 5 periods. Helps the model learn the volatility in the small time duration
- *rv_120*: The realized volatility (standard deviation of log returns) of the data over the past 120 periods. Helps the model learn the volatility in the long time duration
- *rel_vol_60*: The measure for abrupt spike up or down of the market volume
- *ma_5close:* moving average of 5 data points to filter out the noise

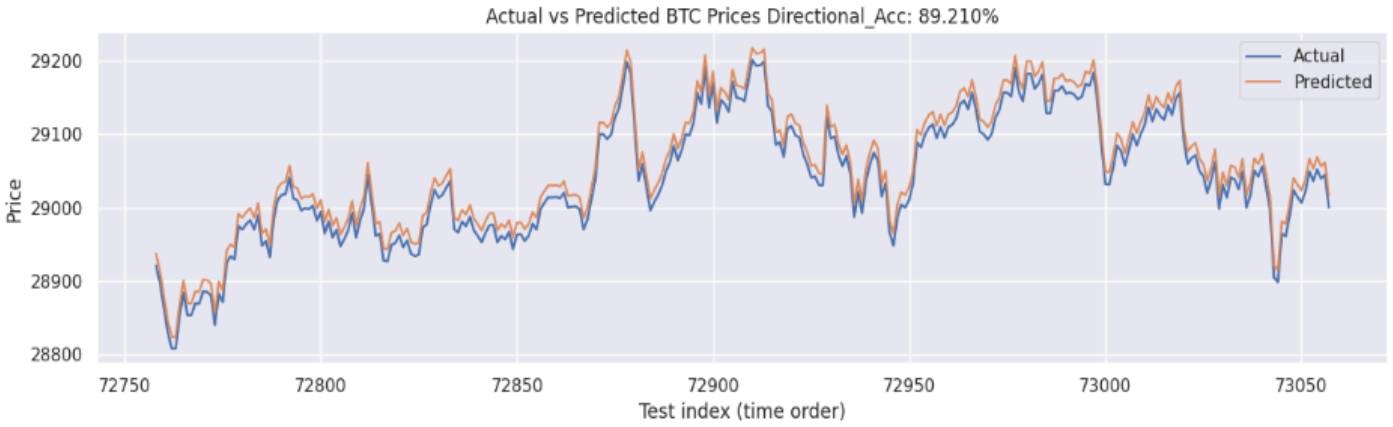
After removing NaN values and scaling the data, it is fed into the neural net.

1. It goes through 2 blocks of temporal convolution block
    a. One branch has smaller kernel size for learning patterns on shorter scales
    b. Other branch has larger kernel size for learning on longer scales
2. The output is concatenated and fed to a single convolution to learn further complex patterns
3. In order to understand the pattern distributed over time we use the MultiHeadAttention layer.
4. Then it passes through GlobalAvgPooling1D which flattens the data while reducing the parameters preventing overfitting and making the model faster

5. It is then passed to 2 dense layers which finally learn the global relations of the input from the previous layers and send it to the last 1 neuron layer for regression.

During model training, a learning rate sweeper is employed which is used to train for a few epochs to see what learning rate works the best for the model.
The mode uses a custom loss which is a modified MSE loss. Essentially along with the MSE loss a component of penalty from the incorrect direction of closing value. This new component is controlled by a parameter alpha which is kept small to ensure smooth training but at the same time telling the model to predict the correct market direction.

PLOT: Accuracy v/s Predicted BTC prices



PLOT: Accuracy v/s Predicted BTC prices; Zoomed