ConvLOB

Stock prices forecasting through CNNs

Tommaso Battistini Edoardo De Matteis

Task

In today's financial markets most trades are performed electronically and the majority is automated. Therefore by analyzing this vast amount of transactions an opportunity has risen.

Develop a deep neural network that leverages the attention mechanism to predict future stock price movements in the F1-2010 high-frequency LOB Dataset.

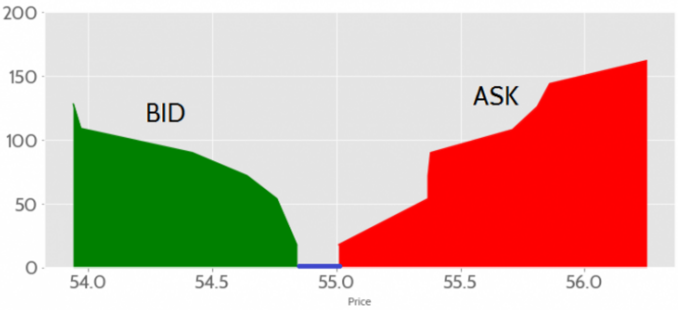
Limit Order Book (LOB)

The Limit Order Book (LOB) has three components: **buy orders**, **sell orders** and **order history**. These are organized in **price levels**. Buy/sell (ask/bid) limit orders will sit in the order book until an order of the opposite type matches their price

Both **bid** and **ask** orders are characterized by a price and volume.

$$(p_a, v_a, p_b, v_b) \tag{1}$$

The distance between the highest bid order and the lowest ask order is called **spread**, the average betweem them is also called **mid price**.


200
150
100
50
0
BID
ASK
54.0
54.5
55.0
55.5
56.0
Price

Dataset

We used the *F1-2010* dataset in order to train and test our model. Its data consist of high-frequency limit order data extracted from the *Nasdaq Nordic* stock market, for a window of 10 consecutive days.

We have a total of 40 features for each timestamp, since each state of the LOB contains 10 levels of both buy and sell orders. Data have been used as follows:

- We used 5 days for training, and 2 days for validation i.e. a 80/20 data split.
- Last 3 days for testing.

Normalization

In order to obtain the best possible performances, data are already normalized using z-score normalization:

$$z = \frac{x - \mu}{\sigma} \tag{2}$$

Labelling

In order to create labels that somehow represent the direction of changes in price, we use the mid price:

$$p_t = \frac{p_a + p_b}{2} \tag{3}$$

We may use **strict** or **smooth** labelling, in F1-2010 the latter is applied by default.
Labels are computed using the mean of k steps' mid-prices, comparing the percentage change against a threshold α:

Smooth labelling

$$m_t = \frac{1}{k} \sum_{i=0}^{k} p_{t+1} \tag{4}$$

$$l_t = \frac{m_t - p_t}{p_t} \tag{5}$$

$$l_t > \alpha \Rightarrow \quad \uparrow \tag{6}$$

$$l_t = \alpha \Rightarrow \quad \rightarrow \tag{7}$$

$$l_t < \alpha \Rightarrow \quad \downarrow \tag{8}$$

Model Architecture

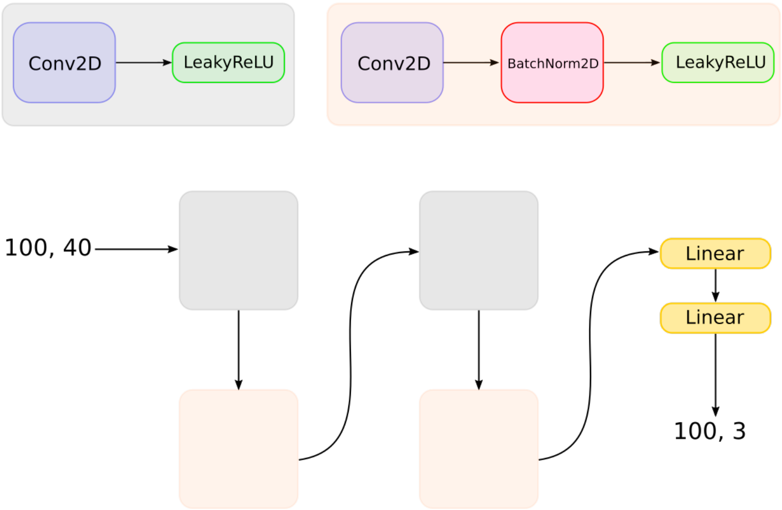
Our models' architectures revolve around 3 main layers:

- Convolutional modules.
- Multi-head attention modules.
- Classification heads.

Convolutional model

This model is inspired by reference paper's model, we exploit 2D temporal convolution, thus we need to reshape input data to a compatible shape.

$$100, 40 \longrightarrow 100, 40, 1$$

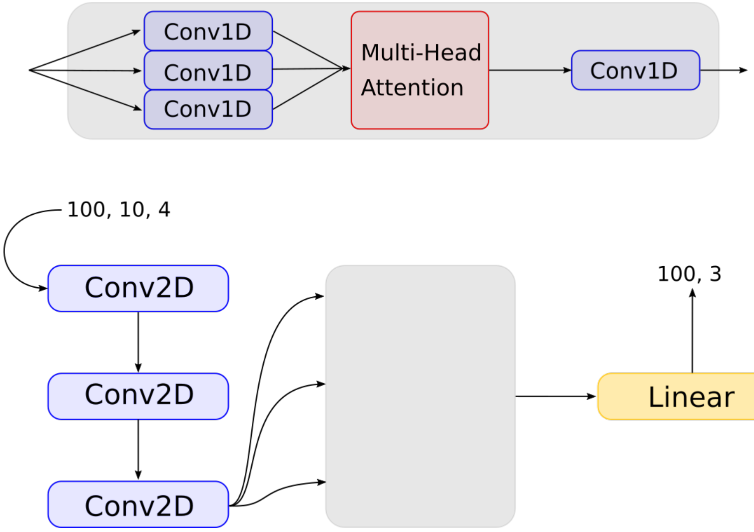

Conv2D
LeakyReLU
Conv2D
BatchNorm2D
LeakyReLU
100, 40
Linear
Linear
100, 3

Attention model

Again we use 2D temporal convolutions, yet data is reshaped in a more channel-centered model.

$$100, 40 \longrightarrow 100, 10, 4$$

Furtherly to learn sequential data we introduce a multi-head attention module.


Conv1D
Conv1D
Conv1D
Multi-Head Attention
Conv1D
100, 10, 4
Conv2D
Conv2D
Conv2D
100, 3
Linear

Convolutional model

Epochs	36
LR	1×10^{-3}
F1	60
Cohen K	0.36

Attention model

Epochs	99
LR	1×10^{-3}
F1	59.2
Cohen K	0.38

Thank you for your attention.