Attention based DeepLOB for Prediction in Limit Order Books

Group :- 49
Guide : Anup Aprem

Department of Electronics and Communication Engineering
National Institute of Technology, Calicut

Nov 16, 2023

Overview

Introduction

1. Predicting price movements from Limit Order Books (LOB) in the financial market has historically been a challenging task due to the high-frequency (order of milliseconds) and complexity of the data.
2. **Problem Statement :** Prediction of movement of the mid price of a stock in limit order books using deep learning models.
3. DeepLOB model combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to capture temporal dependencies and patterns in financial time series.
4. **Novel Contribution:** Introducing Attention Mechanism to the Model after LSTM in DeepLOB [1] , has led to an **increase in accuracy by 9 percent (from 66.42 percent to 74.41 percent).**

Limit Order Books

1. Limit Order Book (LOB) data is a crucial component of financial markets. It contains a real-time record of buy and sell orders for a particular security or financial instrument. Each entry in the LOB includes information such as price, volume, and time of the order.

	SHARES	PRICE
ASKS ↑	11,000	180.07
	12,500	180.06
	12,900	180.05
	9,700	180.04
	1,100	**180.03**
BIDS ↓	6,400	**180.02**
	9,700	180.01
	9,600	180.00
	14,700	179.99
	11,500	179.98

Figure: 1 Limit Order Book[1]

Dataset Description

1. Utilized the FI-2010 dataset with normalized z-score values for features which includes ask/bid prices and volumes, with 10 orders on each side of the mid price of the Limit Order Book[1].
2. The dataset contains 149 rows with, 1 to 144 contain the feature data, where the initial 40 rows that represent LOB points are taken as input features[1].
3. Last 5 rows are labels for various k(Prediction Horizon) values.i.e k=(10,20,30,50,100)[1].

Input

1. We Used 203800 time steps data as training set,50950 as validation set and 139587 as testing set [1].
2. We used the most recent 100 states of the LOB features as input to our model and trained it on label corresponding to prediction horizon k=10 .
3. So the input tensor to models is [1] :

   Training - Featrures :(203701,100,40,1) ; Labels :(203701,3)

   Validation - Features :(50851,100,40,1) ; Labels : (50851,3)

   Testing - Features :(139488,100,40,1) ; Labels : (139488,3)

Labeling

1. Mid-price (Pt) is used to find the price moment.

$$p_t = \frac{p_a^1(t) + p_b^1(t)}{2} \quad (1)[1]$$

$$m_+(t) = \frac{1}{k}\sum_{i=1}^{k} p_{t+i} \quad (2)[1]$$

2. Here m+ denotes the mean of the next k mid-prices and pt is the mid-price defined in Equation (1) and k is the prediction horizon.
3. The labels are then decided based on a threshold(α)

Labeling

$$l_t = \frac{m_+(t) - p_t}{p_t} \quad (3)[1]$$

$l_t > \alpha : +1$ (up – 1)

else: 0 (stationary – 2)

$l_t < -\alpha : -1$ (down – 3)

DeepLOB ADLOB (Proposed)

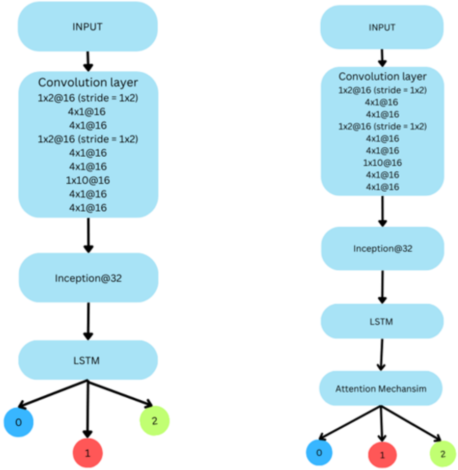

Figure: 2 Model Diagram of (a)DeepLOB[1] (b)ADLOB

Convolution Layers

1. Convolutional Layer processes the input sequence, which represents the 100 most recent updates of an order book, with each update having 40 features per timestamp.
2. Layer has filter size (1×2) and stride of (1×2) for analyzing pairs of adjacent features (alternating price and size).
3. CNN is used to capture short term dependencies as it has look back time step $T = 100$.

Inception Module

1. This enhances the model's adaptability by combining different convolutions.
2. In our method [1], we use 1×1 convolution layer to break down the input into simpler forms.These forms go through transformations with filters (3×1 and 5×1), and the results are combined.
3. We apply a max-pooling layer, with 1×1 convolutional layer ,this approach is used to capture non-linear data properties improving prediction and accuracy[1]

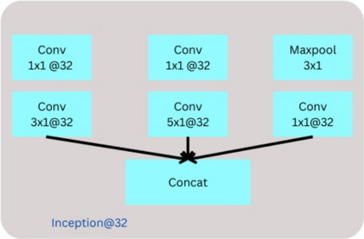

Figure: 3 Inception Module[1]

LSTM

1. LSTM units with a feedback mechanism to preserve memory of past activations enables observing of temporal dynamics in the features.
2. LSTM is used to discover additional time dependencies (long term dependencies)[1].
3. LSTM is followed by Attention and the final output layer which uses softmax activation, with output elements representing the probability of each price movement class (0,1,2) at each time step.

Attention Mechanism

1. The attention layer acts like a spotlight, allowing the model to focus on specific parts of the input sequence when making predictions.
2. This enables the model to pay more attention to relevant information, improving its ability to capture complex patterns and relationships in the data.

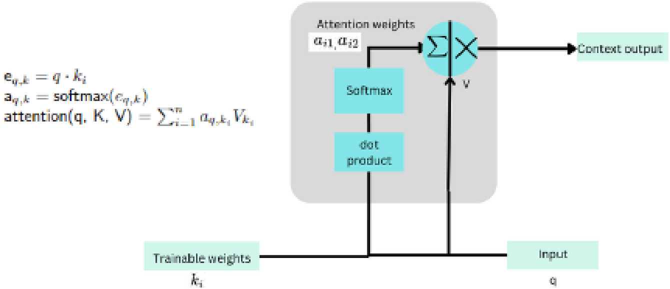

Figure: 4 Weighted Attention Mechanism [2]

Attention Mechanism

1. The input sequence to attention is the output from the previous layer (LSTM).
2. We initialize learnable weights(ki) defined within the Attention Layer. The weights(ki) determine the importance or relevance of different elements in the input sequence when computing the attention scores.
3. Scores (e) are the Dot product of input(q) and W.
4. Softmax operation normalizes scores to produce a probability distribution representing the attention weights (a).
5. Context vector is computed by the weighted sum of input sequence elements(V) based on attention weights(a), allowing the model to determine the importance or weight of each element in the input sequence.

Simulation Results with shrunken dataset for 50 epochs

1. Intially, simulation of the model was done in Google Colab platform with GPU backend on a shrunk down dataset for 50 epochs
2. We experimented with 2 different models the base model DeepLOB and our proposed ADLOB.
3. Fig.5 depicts the Confusion Matrix without attention (DeepLOB) and with attention (ADLOB) respectively
4. From TABLE.1 we observe **increase in accuracy score from 77.01 to 79.05 (2 percent increase).**

Testing Results

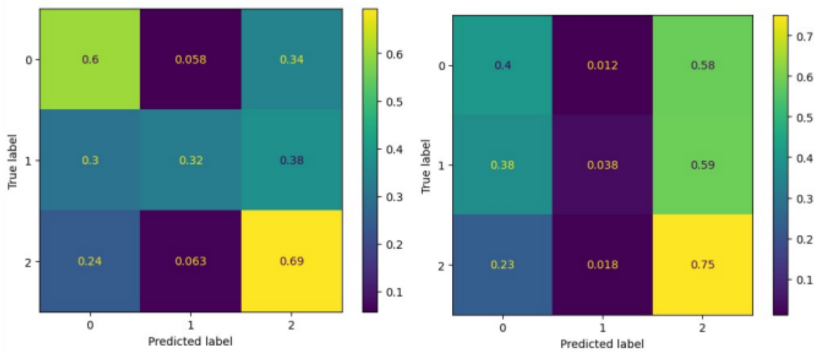

Figure: 5 Confusion Matrix for (a) DeepLOB and (b) ADLOB respectively

Testing Results

Label	DeepLOB			ADLOB		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
0	0.7919	0.7599	0.7755	0.7874	0.7675	0.7773
1	0.7793	0.8911	0.8315	0.8653	0.8332	0.8490
2	0.7651	0.7119	0.7376	0.7383	0.7822	0.7596
Accuracy Score			**0.7701**			**0.7905**

Table: 1 Results for T=10

Simulation Results with Complete Dataset

1. We trained the model on the complete dataset offline in Jupyter Notebook (GPU: NVIDIA-RTX6000) for 200 epochs.
2. TABLE.2 shows increase in accuracy **from 66.42 percent to 74.41 percent (9 percent increase).**
3. Fig.2 depicts the Confusion Matrix without attention (DeepLOB) and with attention (ADLOB) respectively.

Testing Results (on complete Dataset)

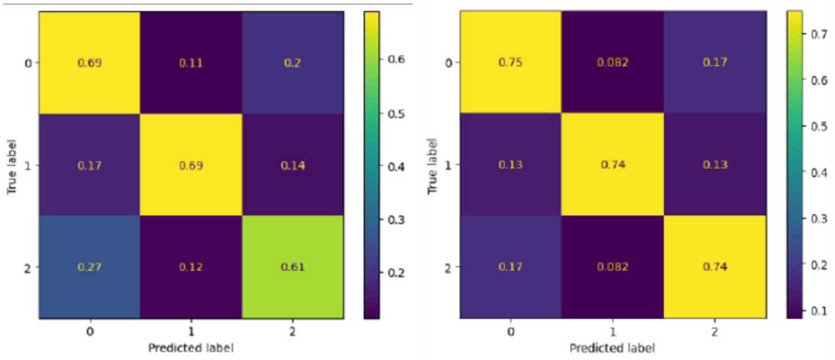

Figure: 6 Confusion Matrix of (a) DeepLOB and (b) ADLOB respectively

Testing Results (on complete Dataset)

Label	DeepLOB			ADLOB		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
0	0.6234	0.6884	0.6543	0.7198	0.7478	0.7335
1	0.7549	0.6862	0.7189	0.8253	0.7401	0.7804
2	0.6222	0.6134	0.6178	0.6951	0.7447	0.7191
Accuracy Score			**0.6642**			**0.7441**

Table: 2 Results for T=100

Future Research Scope

- Future research may focus on exploring additional data sources, refining model architecture like investigating more advanced attention mechanisms (such as Scalar-Dot Product Attention) or adding Self Attention layer and converting it to Transformer based model (TransLOB)
- We can also change the LSTM to a Bi-Directional LSTM to produce a more meaningful output, by combining LSTM layers from both directions.

Conclusion

- In conclusion, the DeepLOB with Attention model presented is and promising approach for predicting financial market movements based on limit order book (LOB) data.
- The combination of convolutional neural networks (CNNs), Long Short-Term Memory (LSTM) networks, and an Attention mechanism demonstrates the model's capability to capture both spatial and temporal dependencies inherent in financial time series.
- Additionally, we showed that Adding Attention Mechanism to DeepLOB architecture has lead to an increase in accuracy from 66 to 74 percent (9 percent increase).

Appendix

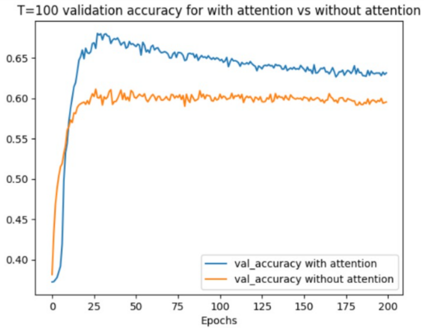

Figure: 7 Comparison of validation accuracy of DeepLOB and (ADLOB)

References

[1] Z. Zhang, S. Zohren and S. Roberts, "DeepLOB: Deep Convolutional Neural Networks for Limit Order Books," in IEEE Transactions on Signal Processing, vol. 67, no. 11, pp. 3001-3012, 1 June1, 2019, doi: 10.1109/TSP.2019.2907260.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, "Attention Is All You Need" in Advances in Neural Information Processing Systems 12 Jun 201710.1109/COMPSAC51774.2021.00173.

[3] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, Igor Mordatch "Decision Transformer: Reinforcement Learning via Sequence Modeling "