Stock Price Direction Prediction Based on Deep Learning

DATS 6303 Deep Learning Final Group Project Report

Group 2

Luhuan Wang, Jianjun Gao

Introduction

In the financial field, stock price trend prediction has always been an important and challenging problem. In this project, our group proposed a deep learning-based model to predict the direction of stock prices, that is, to predict whether the closing price of the next year will rise through the company's annual report data and adjusted closing price. This study uses deep learning models such as LSTM, RNN, CNN-LSTM and LSTM-Attention for comparison and analyzes their performance.

Data Cleaning and Preprocessing

Data Source:

Our dataset includes financial statements and adjusted closing price information obtained from NASDAQ and NYSE listed companies. The main data file is final_data_with_adjClose.csv, which contains the following:

Financial statements: revenue, grossProfit, netIncome, cashFlow, etc.

Stock price: adjusted closing price at the end of the year. That is, set to adjClose in the code.

Data Cleaning

Missing value processing: delete rows containing missing values.

Data label generation: Generate the label price_direction based on the adjusted closing price:

If the adjusted closing price of the next year is higher than the current year, the label is 1, otherwise it is 0.

Feature selection and normalization

Feature selection: Select key financial features and price information as input to the model.

Data normalization: Use MinMaxScaler to scale the features to the [0, 1] interval.

Time series creation

Convert each company's financial data into a 5-time step sequence for training LSTM and other models.

Data statistics after cleaning:

Number of samples: X.shape[0]

Number of features: X.shape[2]

Model Design

Model selection:

In this project, we selected the following four models for comparison:

LSTM: Suitable for capturing long-term dependencies of time series data.

RNN: Basic recurrent neural network, suitable for short-term dependency modeling.

CNN-LSTM: Combines convolutional layers with LSTM for local feature extraction and global sequence dependency modeling.

LSTM with Attention: Introducing the attention mechanism to focus more on the key time points in the time series.

Model Architecture

The detailed architecture of each model is as follows:

LSTM model:

Bidirectional LSTM layer

Fully connected layer (Dense)

Dropout to prevent overfitting

RNN model:

Two-layer SimpleRNN

Fully connected layer

Dropout to prevent overfitting

CNN-LSTM model:

1D convolution layer

Maximum pooling layer

Bidirectional LSTM

LSTM-Attention model:

Bidirectional LSTM

Attention layer

Model summary

The structure diagram and parameter statistics of each model are as follows:

LSTM model: Total X parameters

RNN model: Total X parameters

CNN-LSTM model: Total X parameters

LSTM with Attention model: Total X parameters

Experimental Setup

Data partitioning:

Training set: 80%

Test set: 20%

Validation set: 10% from the training set

Training configuration:

Optimizer: AdamW

Learning rate: 0.001, dynamically adjusted using ReduceLROnPlateau.

Loss function: binary_crossentropy.

Evaluation indicator: accuracy.

Results and Analysis

Comparison of model performance

The performance of different models is compared by verification accuracy:

LSTM verification accuracy: 65% Test accuracy: 64%

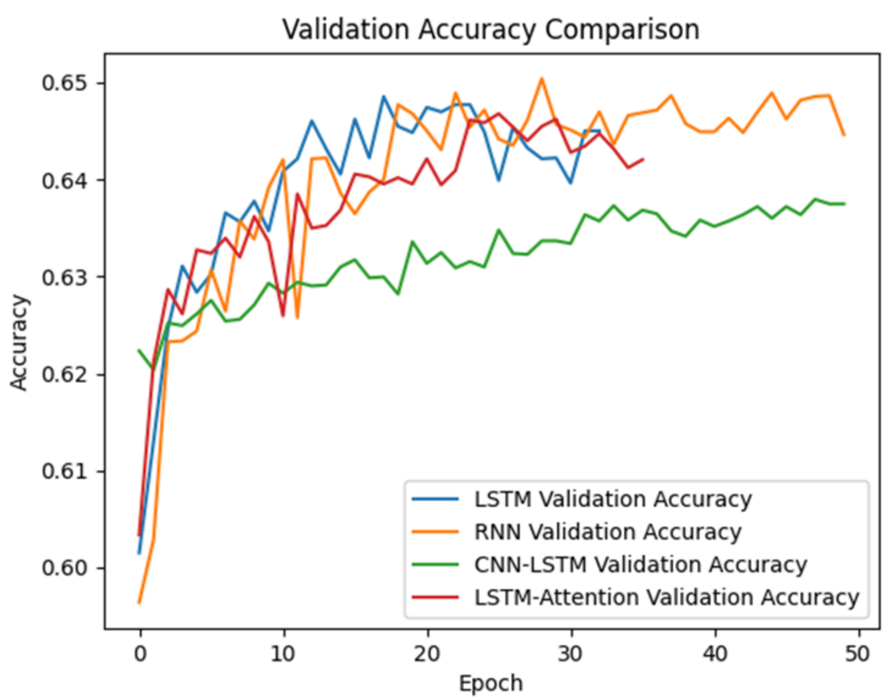
RNN verification accuracy: 63% Test accuracy: 62%

CNN-LSTM verification accuracy: 60% Test accuracy: 58%

LSTM-Attention verification accuracy: 65% Test accuracy: 65%

Verification accuracy curve

The verification accuracy curves of each model are compared as follows:



Analysis

LSTM and LSTM with Attention perform best in capturing time series patterns.

The performance of CNN-LSTM is slightly worse, probably because the local patterns in the data are not significant enough.

The performance of RNN model is lower than that of LSTM due to its insufficient ability to capture long-term dependencies.

Future work

We can increase the amount of data to cover more time spans. We can also introduce more macroeconomic indicators. In terms of code, we can try to optimize the model architecture, such as combining Transformer for time series prediction.

Conclusion

In this presentation, we explored the application of deep learning models for predicting stock price directions and conducted a comparative analysis of various models. The results demonstrated the potential of advanced techniques in financial forecasting, while also highlighting areas for improvement and future research opportunities.

LSTM and LSTM with Attention are the best models in this study.

The attention mechanism can further improve the performance and interpretability of the model.

Data quality and feature selection have a significant impact on model performance.

Code link

The complete code can be found in: *https://github.com/Donny0604/Final-Project-Group2.git*