Stock Market Prediction Using Deep Learning(CNN-LSTM Model)

Agamjot Singh
agam.9427@gmail.com
Mentored By:
Agam

Jan,2023

Authors Of The Original Article- (?, ?).

Contents

1 Introduction

The Stock Market is a crucial platform for humans to trade and get equity in firms they have confidence in. For some, it serves as an additional source of income while for others it's the only source of income. The change in the trend of the stock market price has always been identified as a very important problem in the economic field. The traditional analysis method is based on economics and finance and involves *fundamental and technical analysis.* This is a tedious method and for a living, it can involve a lot of risks.

In recent years Machine Learning has blown up and one of its subsets- *Deep Learning* has a notable use in this field. In this project, we use *Convolutional Neural Network(CNN) - Long short-term memory(LSTM)* Model to predict stock prices.The accuracy of traditional methods is difficult to be convincing. On the contrary, this model is a *high accuracy-high performance* model.

2 Dataset

2.1 Data Collection

Stock Market data was provided by the mentors of the project.

2.2 Data Preprocessing

3 Understanding CNN-LSTM Model

3.1 CNN-LSTM Model

CNN(*Convolutional Neural Network*) helps in understanding and *finding patterns and features* using convolution layer and pooling layer.
LSTM(*Long Short-Term Memory*) networks are well-suited to classifying , processing and making predictions based on time series data.

According to the characteristics of CNN and LSTM, a stock forecasting model based on CNN-LSTM is established. The model structure (refer Figure 2) includes input layer, one-dimensional convolution layer, pooling layer, LSTM hidden layer, and full connection layer.

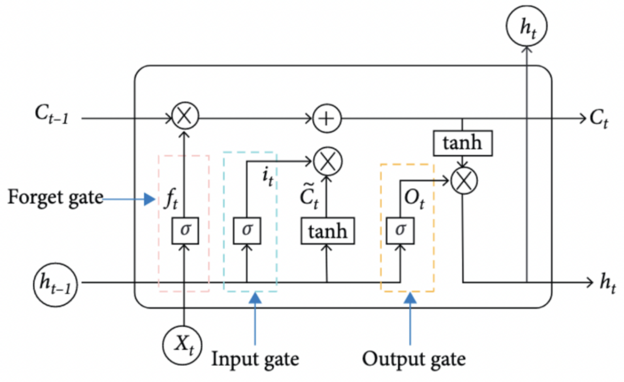


Figure 1: CNN-LSTM structure diagram.

3.2 CNN

CNN is mainly composed of two parts: **convolution layer** and **pooling layer**. Each convolution layer contains a plurality of convolution kernels, and its calculation formula is shown in formula (1). After the convolution operation of the convolution layer, the features of the data are extracted, but the extracted feature dimensions are very high, so in order to solve this problem and reduce the cost of training the network, a pooling layer is added after the convolution layer to reduce the feature dimension:

$$l_t = tanh(x_t * k_t + b_t) \tag{1}$$

where l_t represents the output value after convolution, $tanh$ is the activation function, x_t is the input vector, k_t is the weight of the convolution kernel, and b_t is the bias of the convolution kernel.

3.3 LSTM

LSTM is a network model designed to solve the longstanding problems of gradient explosion and gradient disappearance in RNN. The LSTM memory cell consists of three parts: **the forget gate, the input gate, and the output gate**, as shown in Figure **??**.

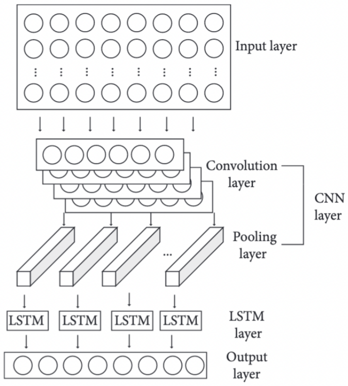


Figure 2: Architecture of LSTM memory cell.

4 CNN-LSTM Training and Prediction Process

The CNN-LSTM process of training and prediction is shown in Figure 3.

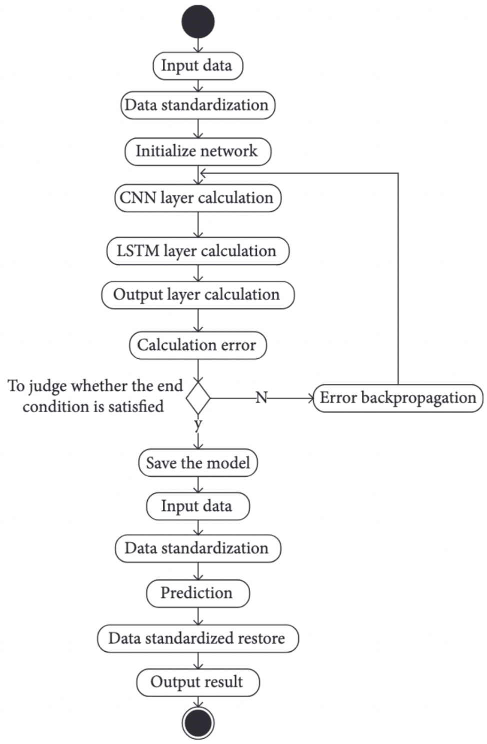


Figure 3: Activity diagram of CNN-LSTM training and prediction process.

The main steps are:

- Input data: input the data required for model training.
- Data standardisation: to train the model better, the *z-score* standardisation method is applied to standardise the input data according to the following formula:

$$y_i = \frac{x_i - \bar{x}}{s},$$

$$x_i = y_i * s + \bar{x}$$

where y_i is the standardised value, x_i is the input data , $\bar{x}$ is the average of the input data and s is the standard deviation of the input data.

- Initialise network: initialise the weights and biases of each layer of the CNN-LSTM.
- CNN layer calculation: the input data are successively passed through the convolution layer and pooling layer in the CNN layer, the feature extraction of the input data is carried out, and the output value is obtained.
- LSTM layer calculation: the output data of the CNN layer are calculated through the LSTM layer, and the output value is obtained.
- Output layer calculation: the output value of the LSTM layer is input into the full connection layer to get the output value.
- Calculation error: the output value calculated by the output layer is compared with the real value of this group of data, and the corresponding error is obtained.
- To judge whether the end condition is satisfied
- Error backpropagation: propagate the calculated error in the opposite direction, update the weight and bias of each layer, and go to step 4 to continue to train the network.
- Save the model
- Input Data
- Data standardisation
- Forecasting
- Data standardized restore: the output value obtained through the model of CNN-LSTM is restored to the original value.
- Output result: output the restored results to complete the forecasting process.

5 Model Implementation

In order to evaluate the forecasting effect of CNN-LSTM, *the mean absolute error (MAE)*, *root mean square error (RMSE)*, and *R-square (R2)* are used as the evaluation criteria of the methods.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i|$$

$$RMSE = \sqrt[2]{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

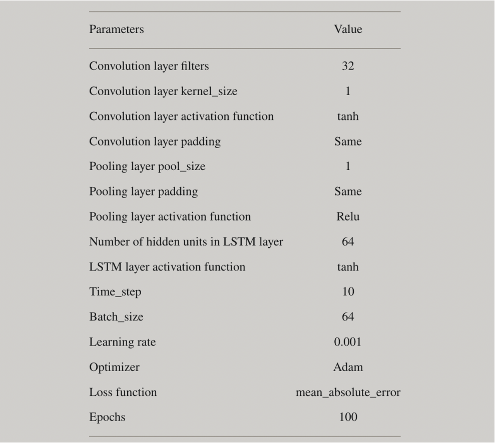
$$R^2 = 1 - \frac{(\sum_{i=1}^{n} (y_i - \hat{y}_i)^2)/n}{(\sum_{i=1}^{n} (\bar{y}_i - t\hat{y}_i)^2)/n}$$

The closer the value of MAE and $RMSE$ to 0, of R^2 to 1, the smaller the error between the predicted value and the real value, the higher the forecasting accuracy.

6 Implementation of CNN-LSTM

6.1 Parameter setting

The parameter setting of the CNN-LSTM for this experiment is shown in Figure 6.1.



Parameters	Value
Convolution layer filters	32
Convolution layer kernel_size	1
Convolution layer activation function	tanh
Convolution layer padding	Same
Pooling layer pool_size	1
Pooling layer padding	Same
Pooling layer activation function	Relu
Number of hidden units in LSTM layer	64
LSTM layer activation function	tanh
Time_step	10
Batch_size	64
Learning rate	0.001
Optimizer	Adam
Loss function	mean_absolute_error
Epochs	100

6.2 Model Structure and Implementation

According to the parameter setting of CNN-LSTM network, we can know that the specific model is constructed as follows: the input training set data is a three-dimensional data vector (None, 10, 6), in which 10 is the size of the time-step and 8 is the 8 features of the input dimension. First, the data enter the one-dimensional convolution layer to further extract features and obtain a three-dimensional output vector (None, 10, 32), in which 32 is the size of the convolution layer filters. Next, the vector enters the pooling layer, and a three-dimensional output vector (None, 10, 32) is also obtained. And then, the output vector enters the LSTM layer for training, and the output data (None, 64) after training enter another layer of full connection layer to get the output value; 64 is the number of hidden units in the LSTM layer.

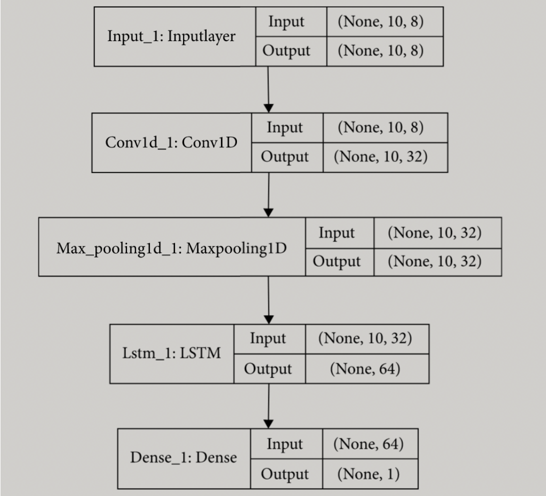


Figure 4: The model structure of CNN-LSTM.