Predicting Stock Prices using LSTM

September 1, 2024

Background

Can machine learning accurately predict stock prices? Investors often make predictions based on data analysis, including company performance, industry trends, and news. Although some theories suggest stock prices are random and unpredictable, the employment of quantitative analysts by top firms like Morgan Stanley and Citigroup indicates otherwise. Modern trading floors are increasingly filled with machine learning experts working with algorithms, and today, roughly 70% of Wall Street orders are placed by software, signifying the rise of algorithmic trading.

Investment firms, hedge funds, and individual investors have long used financial models to better comprehend market behavior and execute profitable trades. The availability of historical stock prices and company performance data provides a rich source of information for machine learning algorithms to analyze.

This project aims to apply deep learning models, specifically Long-Short Term Memory (LSTM) neural networks, to predict stock prices. While Recurrent Neural Networks (RNNs) are useful for time-series data, recent research highlights that LSTMs, a variant of RNNs, are particularly effective. The project will use the Keras library to build an LSTM model that predicts stock prices using historical data on closing prices and trading volumes, with visualization of both predicted price trends and model parameters.

Problem Statement

The project aims to accurately predict the future closing prices of stocks over a given period. While numerous academic papers have applied neural networks to stock price prediction with mixed results, recent advancements in machine learning tools have democratized access to powerful predictive models. This project will utilize LSTM networks to predict the closing price of the S&P 500 index, based on past stock prices.

Goals

1. Explore stock price data.
2. Implement a basic model using linear regression.
3. Implement LSTM using the Keras library.
4. Compare the results and compile a report.

Datasets and Inputs

The project will use daily S&P 500 stock prices from January 2000 to June 2017. This time-series dataset will allow the model to predict future closing prices based on historical data. The data will be sourced from Google Finance.

Solution Statement

The proposed solution involves utilizing an LSTM neural network to learn from time-series data. The project will be implemented in a Jupyter Notebook using Keras and TensorFlow. Pandas will be used for time-series data management, and model performance will be evaluated by comparing the predicted stock prices with actual prices, as well as with predictions from a benchmark linear regression model.

Benchmark Model

A linear regression model will serve as the benchmark, allowing comparison between machine learning and deep learning approaches. The linear regression model will be based on the examples from Udacity's "Machine Learning for Trading" course. Performance metrics will include mean squared error (MSE) and root mean squared error (RMSE), both calculated using the same dataset.

Evaluation Metrics

Model performance will be measured by the mean squared difference between the predicted and actual stock prices. The results will also be compared to those from the benchmark model, evaluating the improvement achieved by using deep learning.

Project Design

The project will follow the below workflow using the Keras/TensorFlow library and LSTM neural networks:

1. **Set Up Infrastructure:**
    - Use iPython Notebook.
    - Incorporate required libraries (Keras, TensorFlow, Pandas, Matplotlib, Scikit-learn, Numpy).
    - Organize the project using Git.
2. **Prepare Dataset:**
    - Use stock data from S&P 500 companies.
    - Process the data into a Pandas DataFrame.
    - Develop a function to normalize the data.
    - Split the dataset into 80% training data and 20% testing data.
3. **Develop Benchmark Model:**
    - Set up a basic linear regression model using Scikit-learn.
    - Calibrate model parameters.
4. **Develop Basic LSTM Model:**
    - Set up a basic LSTM model using Keras, utilizing the parameters from the benchmark model.
5. **Improve LSTM Model:**
    - Develop, document, and compare results using additional labels for the LSTM model.
6. **Document and Visualize Results:**
    - Plot actual prices, benchmark predicted prices, and LSTM predicted prices.
    - Analyze and interpret the results for the final report.