Reworking an LSTM Model for Enhanced Stock Price Prediction Using Attention Mechanism

Introduction

After extensive research into financial time series prediction, I came across a paper from Stanford University that significantly influenced my approach to improving my LSTM model. The study emphasized the advantages of **Attention-LSTM** models over traditional stacked LSTMs, particularly for stock market data, due to the attention mechanism's ability to prioritize critical information and mitigate unnecessary noise. This insight led me to rework my LSTM model with attention, aiming to achieve better accuracy in predicting stock prices.

Transition from Stacked LSTM to Attention-LSTM

The main adjustment was shifting from a **stacked LSTM** structure to an **Attention-LSTM** model. Unlike stacked LSTMs, which focus solely on capturing temporal dependencies with additional layers, the Attention-LSTM model prioritizes relevant sequences, capturing both short-term and long-term dependencies better. For financial time series, which are notoriously noisy, this change enhances the model's ability to focus on critical price movements rather than all data equally, improving overall prediction accuracy(LSTM Model Stanford Art…).

Activation Functions: Choosing Tanh and Avoiding Sigmoid

For this model, I used the **tanh** activation function within the LSTM layers, as the research indicated its superior performance in maintaining stable gradients across time steps. Additionally, I experimented with **sigmoid activation** for the output layer but found that it did not yield satisfactory results. Sigmoid compressed the output values too narrowly, limiting the model's ability to predict the broader range of stock prices. I reverted to a linear activation for the output layer, which allowed the model to predict continuous price values more effectively.

Implementation Steps

The restructured model was built incrementally:

1. **Attention Layer Integration**: After the LSTM layers, I added a custom attention layer that assigned weights to time steps, directing the model's focus towards important moments within the sequence.
2. **Data Split and Scaling**: I revised the data split, adopting a 70-15-15 ratio for training, development, and testing. This allowed the model to generalize better by evaluating it on an entirely unseen test dataset rather than a single week's worth of data, resulting in a **Mean Absolute Percentage Error (MAPE)** of 10.08%.
3. **Inclusion of Fundamental Indicators**: Following the paper's methodology, I incorporated key fundamental indicators such as **Debt-to-Equity Ratio, Return on**

Equity, Price-to-Book Ratio, Profit Margin, Diluted Earnings Per Share (EPS), and Beta. Each of these indicators contributed unique insights:
   - **Debt-to-Equity Ratio**: Provided context on leverage, affecting risk assessment.
   - **Return on Equity (ROE)**: Offered a measure of profitability, important for growth potential.
   - **Price-to-Book Ratio**: Provided a value assessment relative to assets.
   - **Profit Margin**: Informed on operational efficiency.
   - **Diluted EPS**: Represented earnings potential, relevant for stock value.
   - **Beta**: Measured volatility, indicating sensitivity to market movements(LSTM Model Stanford Art…).

Hyperparameter Tuning and Optimization

To further refine the model, I systematically adjusted the following hyperparameters:

- **Dropout Rate**: Managed overfitting by testing values between 0.1 and 0.5.
- **Lookback Period**: Tested different lookback periods (e.g., 10, 20, 30 days) to find the most predictive range for past prices.
- **Batch Size and Epoch Count**: Adjusted batch sizes (16, 32) and epochs (up to 100) to balance training efficiency and model convergence.
- **Units in LSTM Layers**: Experimented with varying the number of LSTM units, testing values like 32, 64, and 128 to optimize complexity without overfitting.
- **Data Period**: Adjusted the date ranges to observe how longer training periods influenced model accuracy.

These experiments confirmed that a lookback period of 20 days and a dropout rate of 0.2 yielded the best performance, aligning with findings in the research paper.

Regularization Attempts and Decision Against L2

During testing, I implemented **L2 regularization** in the LSTM layers. However, it significantly underfit the data, limiting the model's ability to learn nuanced trends. Given the underfitting observed, I chose to omit L2 regularization to maintain the model's predictive power.

Results and Conclusion

The reworked Attention-LSTM model, incorporating refined hyperparameters and fundamental indicators, resulted in a **MAPE of 8.23%** on the test data—a substantial improvement. The **Data Document** records the model's progression, showing each incremental enhancement and its effect on predictive accuracy. The research paper's insights provided invaluable guidance in achieving this optimized model, demonstrating the benefit of attention mechanisms for complex financial time series data.