Report

Implementation and Evaluation of Transformer-Based Model for Trade Recommendations

1. Introduction

The objective of this task was to implement a transformer-based model using Python and PyTorch to generate trade recommendations (Buy, Sell, Hold) based on market data. The model was evaluated based on its performance in generating meaningful and actionable trade signals.

2. Implementation Details

The transformer model was designed to process sequential market data and generate trade recommendations. The implementation process can be divided into several key steps:

1. **Data Preprocessing**:
    - Market data was loaded and preprocessed, including scaling price columns and generating technical indicators such as RSI, MACD, Stochastic Oscillator, OBV, Bollinger Bands, ATR, and ADX.
    - Data was normalized using MinMaxScaler to ensure the features were on a similar scale.
2. **Dataset and DataLoader**:
    - A custom TradeDataset class was created to manage the sequential data required for the transformer model.
    - A DataLoader was utilized to handle batching and shuffling during training.
3. **Model Architecture**:
    - The transformer model was designed with 4 transformer encoder layers, each with 8 attention heads, and a final linear layer to predict future prices.
    - The input features included technical indicators calculated from the market data.
4. **Training**:
    - The model was trained for one epoch with a batch size of 256, using MSE loss as the criterion and Adam optimizer for parameter updates.

5. **Signal Generation**:
    - After training, the model was used to generate trade recommendations based on predicted price movements. Thresholds were set for Buy (0.5%) and Sell (-0.5%) signals.

3. Fine-Tuning Process

The model was fine-tuned by adjusting various hyperparameters such as learning rate, batch size, and the number of encoder layers and attention heads. The following techniques were considered to optimize the model's performance:

- **Early Stopping**: Prevents overfitting by stopping training when the validation loss stops improving.
- **Learning Rate Scheduling**: Adjusts the learning rate during training to allow for more refined learning.

4. Evaluation Results

The model was evaluated on the basis of the generated trade recommendations. The signals produced were compared to a simple trading blotter for performance assessment.

- **Sample Output**:
    - Close Price: 0.253968, Prediction: [0.30072302], Signal: BUY
    - Close Price: 0.257937, Prediction: [0.18434688], Signal: SELL

The generated signals show that the model was able to predict some Buy and Sell signals based on the thresholds set.

- **Cumulative Reward**: The model's cumulative reward and final balance were calculated and compared to a baseline strategy (e.g., PPO-based trading strategy). However, it was noted that some predictions led to divide-by-zero errors, indicating the need for further model refinement.

5. Comparison with PPO-Based Strategy

- **Transformer Model**:
    - The transformer model was relatively quick to train and provided a diverse set of predictions.
    - However, the model encountered issues with extreme price values leading to invalid operations (divide by zero), which impacted the signal reliability.

- **PPO-Based Model**:
    - The PPO model, with a more reinforcement learning-based approach, may offer more stability in environments where the action space is more dynamic.
    - The cumulative reward of the PPO strategy was also a strong indicator of its overall performance.

6. Conclusion

The transformer-based model demonstrated potential in generating trade recommendations but required further fine-tuning to address certain issues such as divide-by-zero errors. The model's performance, while promising, did not yet surpass that of the PPO-based strategy, which was more stable and yielded a higher cumulative reward.

Future work would involve more sophisticated error handling, further hyperparameter tuning, and potentially hybridizing the transformer model with reinforcement learning approaches like PPO to enhance decision-making capabilities in a trading environment.