CryptoQuant LSTM Trading Strategy - Documentation

Overview

This project implements an end-to-end cryptocurrency trading pipeline leveraging LSTM (Long Short-Term Memory) neural networks for price movement prediction using BTC on-chain and technical indicator data. The system is built using Python, TensorFlow, and Scikit-learn, and integrates the Cybotrade API to access and combine OHLCV, whale, and miner-related features.

Architecture

1. Configuration

- config.py defines essential parameters including:
  - Window size for LSTM input
  - Training-test split ratio
  - Model hyperparameters (e.g., epochs, batch size)
  - API credentials and query options for Cybotrade

2. Data Ingestion

- Uses the Cybotrade API to fetch BTC hourly OHLCV data.
- Adds on-chain features such as:
  - Whale ratio
  - Miner outflow
  - Exchange netflow
  - Open interest
  - Funding rate

3. Preprocessing

- Features are merged by timestamp.
- Missing values are forward/backward filled.
- Targets are labeled as binary classes (1: price up, 0: price down) using future closing price.
- All features are scaled using StandardScaler.

4. Feature Engineering

- Technical indicators: RSI, SMA, EMA.
- On-chain features.
- Hidden Markov Model (HMM) states for market regime detection.
- Lag features for previous time steps.

5. Modeling - LSTM

- Built with TensorFlow Keras.
- Model takes sequence input of shape (window_size, num_features).
- Architecture:
  - LSTM layer with dropout
  - Dense output layer with sigmoid activation
- Binary Crossentropy loss function and Adam optimizer.
- Early stopping based on validation loss.

6. Prediction and Signal Generation

- Outputs a binary prediction: 1 (buy signal), 0 (sell/hold signal).
- Predictions are aligned with original timestamps.

7. Backtesting Engine

- Compares model-generated signals with actual price changes.
- Calculates:
  - Strategy returns
  - Buy & Hold returns
  - Sharpe ratio
  - Maximum drawdown
- Option to include slippage/transaction fees.

8. Visualization

- Matplotlib and Plotly charts for:
  - Predicted vs actual prices
  - Cumulative returns comparison
  - Signal overlay on price chart

Requirements

```
pip install tensorflow scikit-learn pandas numpy matplotlib cybotrade hmmlearn
```

Files Structure

```
crypto-lstm-strategy/
 config.py
 data_loader.py
 feature_engineering.py
 lstm_model.py
 backtester.py
```

CryptoQuant LSTM Trading Strategy - Documentation

main.py

README.md

results/

Key Benefits

- Combines on-chain analytics with price data.

- Captures temporal patterns using LSTM.

- Uses market regime detection with HMM.

- Fully backtestable pipeline.

Future Improvements

- Integrate with live trading platforms like Binance via API.

- Extend to multi-asset support.

- Improve accuracy with attention-based models or ensemble learning.

- Use reinforcement learning for portfolio optimization.

Author

Built by [Your Name] as part of a crypto alpha strategy research project.