Data Preparation

In this section, Bitcoin price data is loaded from the file `btcusdt.pkl`. Then, three new features are created:

- Percentage change in price (compared to the previous price)
- Ratio of high to close
- Ratio of low to close

After that, any rows with missing values (NaN) are removed.

Environment Definition

The code uses a trading simulation environment called `TradingEnv`. The prepared dataset is passed into this environment. Three trading positions are allowed:
Buy (1), Hold (0), Sell (-1)

Observation Normalization

Using `StandardScaler` from `sklearn`, the observation data is normalized. To do this, 1500 random samples are taken from the environment, flattened, and then used to fit the scaler. This helps the neural network learn more effectively.

Actor and Critic Definition

Two neural networks are defined:

- **Actor**: Decides which action to take (buy, sell, hold).

- **Critic**: Estimates the value of the current state (helps guide the Actor's learning).

Both networks use 2 hidden layers with ReLU activation functions.

PG_agent (Learning Agent)

This class:

- Initializes the Actor and Critic networks.
- Sets up optimizers (Adam or SGD).
- Contains a method to sample an action based on the Actor's output.
- Has a method to compute the total discounted reward for an episode.
- Has an `update` method to train both the Actor and the Critic using collected data.

Reward Plotting

There's a function that plots the total reward of each episode over time. The result is saved as an image for later review.

Training Loop

In this loop, the agent is trained:

- The environment is reset for each episode.
- The agent interacts with the environment (chooses actions and receives rewards).
- All states, actions, and rewards are stored during the episode.

- At the end of the episode, the Actor and Critic are updated.
- The total reward is stored for plotting.

Testing Different Parameters

To evaluate performance, different combinations of:

- Learning rates
- Discount factors
- Hidden layer sizes
- Optimizers (for both Actor and Critic)

are tested. A separate reward plot is saved for each combination to compare their results visually.