ISyE 6767 Homework 4 Report

Vidit Pokharna

November 5, 2025

Overview

This report summarizes the implementation of two homework tasks:

- **Problem 1: Trading Strategy** — implemented in NumPy to generate trading signals, track positions and account values, save results, and produce a cumulative P&L chart
- **Problem 2: Energy Maze** — computes the minimum initial energy needed to traverse a maze without the total energy dropping below 1

Problem 1: Trading Strategy

Implementation Summary

- Core logic resides in `strategy_core.py` and is reused for CSV input, the textbook example, and custom unit tests
- Input file: `aapl.csv`; output files: `output/trading_results.csv` and `output/cumulative_pnl.png`
- The final cumulative P&L (account value – $10,000) is printed at the end of the script

Replication of Assignment Example

```
Example prices:  [ 98. 100. 102. 104. 103. 101.  99. 100. 102. 104. 106. 107. 105.]
Example signals: [ 0  0  0  1  0 -1  0  0  0  1  1  0 -1]
Example positions: [ 0  0  0 10 10  0  0  0  0 10 20 20  0]
Example account values: [10000. 10000. 10000. 10000.  9990.  9970.  9970.  9970.  9970.  9970.
  9990. 10010.  9970.]
```

CSV Evaluation and Results

Running `python3.11 hw4.py` on `aapl.csv` produced:

- **Cumulative P&L: $1,377.38**
- Output files: `output/trading_results.csv` and `output/cumulative_pnl.png`

Table 1: Sample rows from `output/trading_results.csv`

Date	Signal	Position	AccountValue
2010-01-04	0	0	10000.0
2010-01-05	0	0	10000.0
2010-01-06	0	0	10000.0
2010-01-07	0	0	10000.0
2010-01-08	0	0	10000.0
2010-01-27	1	10	10000.0

Custom Validation Test

```
Test prices:    [50. 52. 54. 56. 55. 53. 51. 49. 47. 48. 50. 52. 54. 53.]
Test signals:   [ 0  0  0  1  0 -1  0  0  0  0  0  1  1 -1]
Test positions: [ 0  0  0 10 10  0  0  0  0  0  0 10 20  0]
Test account values:
[10000. 10000. 10000. 10000.  9990.  9970.  9970.  9970.  9970.  9970.
  9970.  9970.  9990.  9970.]
```

This case stresses consecutive buy signals, forced liquidation, and stable holding periods. Outputs confirm correct signal handling and value tracking.

Problem 2: Energy Maze

Implementation Summary

- Function `min_initial_energy` uses reverse dynamic programming with NumPy arrays
- Works for arbitrary rectangular mazes with entries in $[-100, 100]$

Replication of Assignment Example

```
Example maze:
[[-2  -3   3]
 [-5 -10   1]
 [10  30  -5]]
Minimum initial energy: 7
```

Custom Validation Test

```
Test maze:
[[ 1 -4  3]
 [-2 -2 -2]
 [ 5 -3 -1]]
Minimum initial energy: 2
```

This maze includes both positive and negative values, confirming the algorithm's ability to balance losses and gains dynamically.

Cumulative P&L Plot

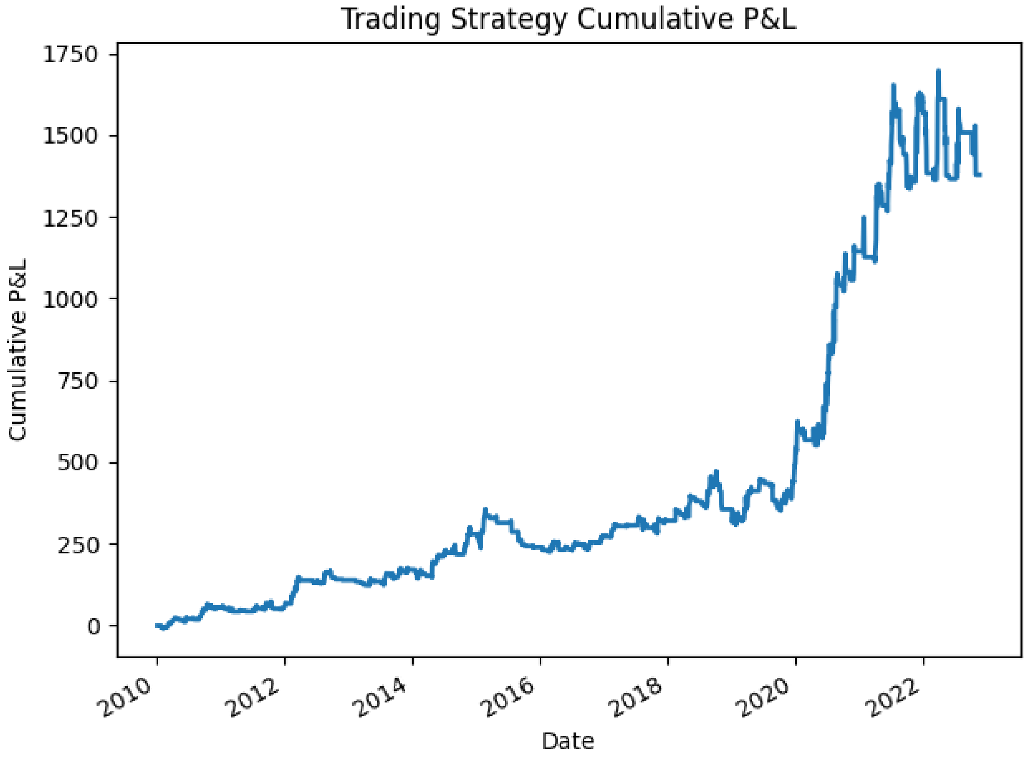

Figure 1: Cumulative P&L for `aapl.csv` starting from $10,000