前言

在本文中，我們將創建強化學習 agents，學習如何通過股票交易賺錢。本文不介紹強化學習的具體原理，僅說明強化學習在量化交易中的應用。

強化學習

Algorithm 5 PPO with Clipped Objective

Input: initial policy parameters θ_0, clipping threshold ϵ
for $k = 0, 1, 2, \ldots$ **do**
  Collect set of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$
  Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
  Compute policy update

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\theta_k}^{CLIP}(\theta)$$

  by taking K steps of minibatch SGD (via Adam), where

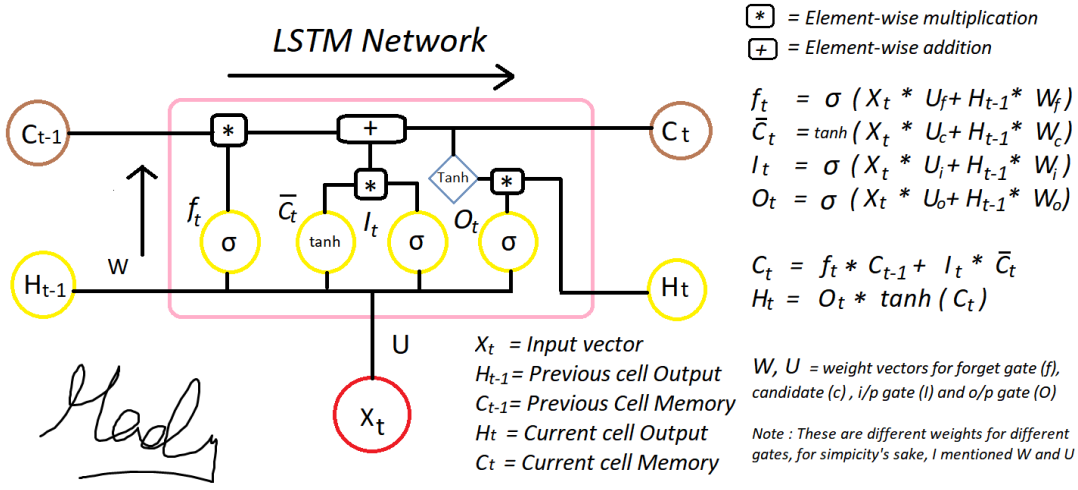
$$\mathcal{L}_{\theta_k}^{CLIP}(\theta) = \mathop{\mathrm{E}}_{\tau \sim \pi_k} \left[\sum_{t=0}^{T} \left[\min(r_t(\theta)\hat{A}_t^{\pi_k}, \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t^{\pi_k}) \right] \right]$$

end for

我們使用 PPO[1]作為基础演算法，訓練一個目标是最大化盈利的決策網路。該演算法的特點是通過 Clip 的方式近似 TRPO。

神經網路

我們使用 LSTM 處理歷史資料，由於 LSTM 能夠隨著時間的推移保持內部狀態，我們不再需要一個滑動的“look-back”視窗來捕捉價格的運動。相反，它本質上是由網路的遞迴特性捕獲的。在每個時間步長，資料集的輸入和最後一個時間步長的輸出都被傳遞到演算法中。



PyTorch 神經網路庫

```
x = F.relu(self.fc1(x))
x = x.view(-1, 1, hidden_size)
x, lstm_hidden = self.lstm(x, hidden)
x = F.relu(self.fc2(x))

pi = self.fc_pi(x)        胡健, 2 days
pi = F.softmax(pi, dim=2)
v = self.fc_v(x)
```

特征處理

我們的資料非平穩的，因此，任何機器學習模型都很難預測未來的值。最重要的是，我們的時間序列包含明顯的趨勢和季節性，這兩個因素都會影響我們演算法準確預測時間序列的能力。我們可以通過使用差分和變換方法從現有的時間序列中得到一個正態分佈來解決這個問題[2]。

差分可以消除趨勢，但是資料仍然具有明顯的季節性。我們可以試著通過在差分前的每個時間步上取對數來去除它，這樣操作，我們可以得到平穩的時間序列。

pandas 處理特征

```
# log diff
windowsSize = 200
pastData = dailyOhlcvFile.tail(windowsSize + 1)[feature_list]
pastData = (np.log(pastData) - np.log(pastData.shift(1))
            ).values[1:].astype(np.float32)
pastData = torch.from_numpy(pastData).to(
    dtype=torch.float32, device=model.device).unsqueeze(1)
```

獎勵函數

考慮使用盈利率最為強化學習的獎勵函數，比如當前買進新增盈利 2.1%，則獲得獎勵值 2.1。

模型保存

因為 JudgeGirl 無法上傳模型文件，所以比較合適的方法是導出神經網路的權重值到程式碼中。缺點是文件長度增加到一萬行，總共幾百千字節。

對比基線

使用 RSI 黃金交叉作為對比基線。並在數據上測試，發現神經網路已經學習到了優秀的交易策略，並且效果遠超 RSI 黃金交叉。（注：RSI 黃金交叉回報率約為 400%，本演算法達到了 3000%）

優劣分析

1. 基於值得強化學習演算法不適用於盈利預測，因為股票市場的噪音非常大，更適合使用基於策略的強化學習演算法[3]。

2. 用盈利作為回報沒有考慮風險因素，股票交易數據噪音大，神經網絡很容易過擬合。所以需要學習比較“保守”的策略，比如用夏普比例作為獎勵函數。

参考文献

1. Proximal Policy Optimization Algorithms

2. 在统计学中为什么要对变量取对数？

3. Agent Inspired Trading Using Recurrent Reinforcement Learning and LSTM Neural Networks