F-022

方策ベースの複利型強化学習

Policy-Based Compound Reinforcement Learning

伊藤 徳晃 *1　　松井 藤五郎 *2 *3
Noriaki Ito　　Tohgoroh Matsui

*1 中部大学 大学院工学研究科 情報工学専攻
Department of Computer Science, Graduate School of Engineering, Chubu University

*2中部大学 生命健康科学部 臨床工学科
Department of Clinical Engineering, College of Life and Health Sciences, Chubu University

*3中部大学 工学部 情報工学科
Department of Computer Science, College of Engineering, Chubu University

1. はじめに

強化学習が適用できる問題としては、金融の他に、AlphaGo、AlphaGo Zero [1] といったゲーム AI や金融問題 [2] への利用が挙げられる。これらの問題は、一般に確率制御の問題として見ることができる。例えば金融問題であればどのタイミングで売買を成立させるかということを考えると一種の制御問題となるし、ゲーム AI の場合もキャラクターの位置などを制御する制御問題として見ることができる。制御問題において強化学習を利用する場合、高速で省メモリな方策ベースの手法が用いられることが多い。

複利型強化学習 [2] は、報酬の替わりに利益率を獲得する試行錯誤を通じて将来に獲得する利益率（リターン）の複利効果（複利リターン）を最大化する行動を学習する。複利型強化学習はこれまで株取引や為替取引などで用いられており、既存研究では価値ベースの手法は利用されてきたが、方策ベースの手法は利用されていなかった。

本論文では、方策勾配法に基づいた複利型強化学習の手法を提案する。提案手法を制御問題に使用し、通常の強化学習と比較した場合に優れているものはどちらかを考察する。また、これまで用いられてきた価値ベースの複利型強化学習アルゴリズムと、方策ベースの複利型強化学習アルゴリズムを比較し、どちらが優れているかを考察する。

2. 提案手法

本論文では、方策ベースの複利型強化学習として、複利型 PPO を提案する。PPO を複利型にするには、PPO における報酬 $r(s,a,s')$ を対数グロス・リターン $\log(1+R(s,a,s')f)$ に置き換える。(ここで、s はある時点での状態、a は状態 s における行動、s' は状態 s の次の状態、f は投資比率を表す。）そのアルゴリズムを Algorithm 1 に示す。ここで、以下に示す clip 関数は方策の急激な変化を防ぐために用いられている。

$$\mathrm{clip}(r(\theta),1-\epsilon,1+\epsilon)=\begin{cases}1+\epsilon & \text{if } r(\theta)>1+\epsilon\\ 1-\epsilon & \text{if } r(\theta)<1-\epsilon\\ r(\theta) & \text{otherwise}\end{cases}$$

Algorithm 1 複利型 PPO

最初の方策 π_{θ_0} を初期化
初期状態 s_0 を初期化
最初の行動 a_0 をランダムに初期化
$Q(s_0,a_0)$ を計算
for *episode* $=1,2,\ldots$ **do**
　状態を初期状態 s_0にする
　for $t=0,1,\ldots$ **do**
　　方策 $\pi_{\theta_{epsode-1}}$ によって状態 s_t における行動 a_{t+1} を決定
　　行動 a_{t+1}によって遷移した状態 s_{t+1}を観測
　　リターン $R(s_t,a_t,s_{t+1})$ を獲得
　　$Q^{\pi_\theta}(s_t,a_t)\leftarrow Q^{\pi_\theta}(s_t,a_t)+\alpha(\log(1+R(s_t,a_t,s_{t+1})f$
　　$+\gamma Q^{\pi_\theta}(s_{t+1},a_{t+1})-Q^{\pi_\theta}(s_t,a_t))$
　　Critic 側のニューラルネットで $V^{\pi_\theta}(s_t)$ を生成
　　$s_t\leftarrow s_{t+1}$
　end for
　アドバンテージ $A^{\pi_\theta}(s,a)=Q^{\pi_\theta}(s,a)-V^{\pi_\theta}(s)$
　Actor 側のニューラルネットで
　　$\theta_{episode-1}\leftarrow\underset{\theta}{\mathrm{argmax}}\,\mathbb{E}[\min(r(\theta)A^{\pi_\theta}(s,a),\mathrm{clip}$
　　$(r(\theta),1-\epsilon,1+\epsilon)A^{\pi_\theta}(s,a))]$
　$\theta_{epsode}\leftarrow\theta_{episode-1}$
end for

制御問題に複利型 PPO を適用するには、まず報酬 r に対応するリターン R を設計する必要がある。報酬は制御対象によっても変わってくるが、リターンは -1 以上の値しか取れないため、最悪な結果に対するリターンを $R=-1$ とし、ニュートラルな結果に対するリターンが $R=0$ となるように報酬をリターンに変換する。また、リターンが $R=-1$ のときに $\log(1+Rf)$ が負の無限大に発散しないように、投資比率 f を $0\leq f<1$ としなければならない。

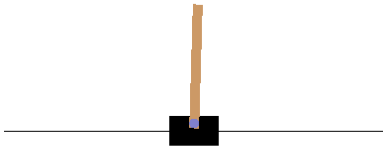

図 1　CartPole 問題の図

表 1　PPO の Actor のニューラルネット

学習率	4×10^{-4}
中間層の数	2
中間層の素子数	32
中間層の活性化関数	tanh
出力層の活性化関数	softmax

表 2　PPO の Critic のニューラルネット

学習率	4×10^{-4}
中間層の数	1
中間層の素子数	1024
中間層の活性化関数	tanh
出力層の活性化関数	恒等関数

3.　評価実験

3.1　実験方法

OpenAI が開発した強化学習ライブラリ gym に含まれている CartPole 問題の version-1 を用いて提案手法の有効性を確認するための実験を行った。CartPole 問題は図 1 のように台車の上に棒が立っており、台車を左右に動かして、その棒を倒さないように制御する問題である。

CartPole には、version-0 と version-1 があり、version-0 は最大ステップ数が 200 回であることに対し、version-1 は最大ステップ数が 500 回であるので、version1 の方がより困難な問題となっている。

また、ニューラルネットワークの構築には Google の開発した Python のライブラリである TensorFlow を用いた。表 1 に PPO の Actor のニューラルネットの構成、表 2 に PPO の Critic のニューラルネットの構成を示す。また、PPO の ϵ は 0.1 とした。エピソード数はどちらも 1000 回で、20 回実験を行った上で全体の平均を取った。

ここで、softmax 関数の入力 θ はベクトルであり、softmax の出力は離散的な確率分布として扱えるので、方策 $\pi_\theta(a|s)$ に用いることができる。本実験でも softmax(θ) を方策 $\pi_\theta(a|s)$ として用いた。また、softmax の温度パラメータは $\tau = 0.5$ とした。

報酬（リターン）$R(s, a, s')$ は、490 step までに倒れた場合は -0.5、それ以外の場合は 0.05 を与えるように設定した。ステップサイズ $\alpha = 4.0 \times 10^{-4}$、割引率 $\gamma = 0.99$、PPO における clip の閾値 $\epsilon = 0.1$、複利型強化学習における投資比率 $f = 0.1 - 10^{-4}$ とした。また、アドバンテージ関数は平均 0、分散 1 になるように標準化した。

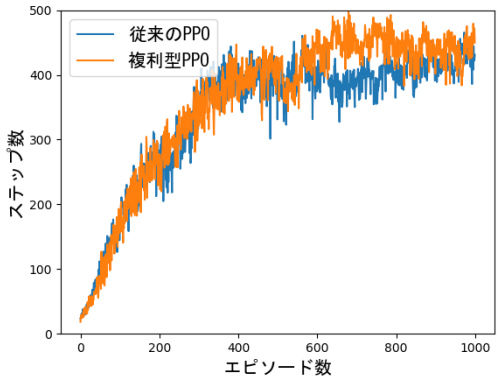

図 2　実験結果

3.2　実験結果

図 2 に実験結果の図を示した。図 2 の横軸はエピソード数、縦軸はステップ数であり、青色の線が従来の手法、オレンジ色の線が複利型の手法である。

4.　考察

図 2 を見ると、従来の PPO は 600〜800 step 付近でステップ数が下がっているのに対し、複利型 PPO は安定したステップ数を取っている。また、表**??**を見ると、従来の PPO よりも複利型 PPO の方が高い。複利型 PPO は、報酬に対数を取るため、負の報酬に対して厳しい。そして、行動価値と状態価値を最大化する上で、学習器は大きな負の報酬を避けながら学習していく。通常の PPO の step 数が途中で下がったのは、負の報酬に対して適切な評価を与えるように学習器が学習しなかったためと考えられる。これらのことから、負の報酬をより厳しく罰することのできる複利型 PPO の方が、より安定した結果を得ることができたと考察できる。また、平均実行時間は、複利型 PPO が 239.96 s で、従来の PPO の 218.25 s よりも高かったが、これは複利型 PPO が従来の PPO よりも平均的に高いステップ数を取ったことに由来するものと考えられる。

5.　まとめ

本研究では、制御問題で成果を挙げている方策ベースの強化学習を複利型に拡張する方法について検討し、方策ベースの勾配法である PPO を複利型にした複利型 PPO のアルゴリズムを示した上で、CartPole 問題を用いて評価を行った。また、図 2 から従来 PPO のよりも複利型 PPO の方がより安定した性能が出ることを示すことができた。今回は CartPole 問題を制御対象にしたが、ロボットやゲーム AI、ファイナンスなどはまだ対象にしていないので、これらを制御対象とすることが今後の課題である。

参考文献

[1] David Silver, Julian Schrittwieser, et al. Mastering the game of Go without human knowledge https://www.nature.com/articles/nature24270, 2017

[2] 松井 藤五郎. 複利型強化学習—強化学習のファイナンスへの応用—. 計測と制御, Vol. 52, No. 11, pp. 1022–1027, 2013.