制御システム論分野研究会資料

2020 年 8 月 28 日

M2 竹内 維吹

進捗報告 8.28

1　サンプル値系システムの強化学習

対象のシステムが

$$s_{t+1} = f(s_t) + g(s_t)a_t \tag{1}$$

と書かれていて, τ ステップ毎に状態を観測し, 状態フィードバック制御則を変えて次の観測までは同じ入力を加え続ける, サンプル値系における最適制御問題の強化学習を考える. この動機は τ ステップ毎に観測・制御則更新を行うのと, それを毎ステップ行う (以下, 毎時刻系と呼ぶ) のとで学習に必要なステップ数が変わるのかを検証するためである.

結論から記すと, サンプル値系にすると学習時間が増えることが確認できた.

1.1　倒立振子による実験

倒立時の振子の角度を $\theta = 0$ とし, 加えられる入力が $A = [-10\mathrm{N}\cdot\mathrm{m}, 10\mathrm{N}\cdot\mathrm{m}]$ と制限されるような倒立振子を考える. この倒立振子のダイナミクスは, 以下のように与えられる.

$$\theta_{t+1} = \theta_t + \dot{\theta}_t\delta_t + \frac{3g}{2l}\sin\theta_t\delta_t^2 + \frac{3}{ml^2}a\delta_t^2 \tag{2}$$

$$\dot{\theta}_{t+1} = \dot{\theta}_t + \frac{3g}{2l}\sin\theta_t\delta_t + \frac{3}{ml^2}a\delta_t \tag{3}$$

ただし, δ_t は離散化定数であり $\delta_t = 0.05$ とする.

1.2　エピソード間蓄積報酬による検証

毎時刻系とサンプル値系の強化学習に必要なステップ数を確認するために, 方策の変化とエピソード報酬の変化の関係を見る. ただしエピソードとは, 200 ステップの離散システムの動作を 1 まとまりにしたものとする. 「環境とのインタラクション」という言葉を「状態変数・即時報酬の観測と, 新たな入力信号をアクチュエータに送信すること」と定義すると, $\tau = 5$ のサンプル値系では 1 エピソードの間に 40 回のインタラクションを行う事になる. 1 回のインタラクションにつき, 1 回制御則を更新することに注意されたい.

では, 毎時刻系とサンプル値系の学習におけるエピソード報酬の変化を確認していく.

図 1 の上段が毎時刻系, 下段がサンプル値系をそれぞれ表す. これらのグラフの横軸はエピソード数を表しているので, 当然ステップ数も同じである. しかしながら毎時刻系とサンプル値系では「インタラクション」の回数が異なるため, $\tau = 5$ のサンプル値系では毎時刻系に対してインタラクション回数は, 毎時刻系の $\frac{1}{5}$ となる. これを見ると毎時刻系では 1200 エピソード付近で最適に近い制御則が学習できていることが確認できるが, サンプル値系では「まだ」改善はみられていない. これはサンプル値系にすると必要なステップ数が増えることの証左となる.

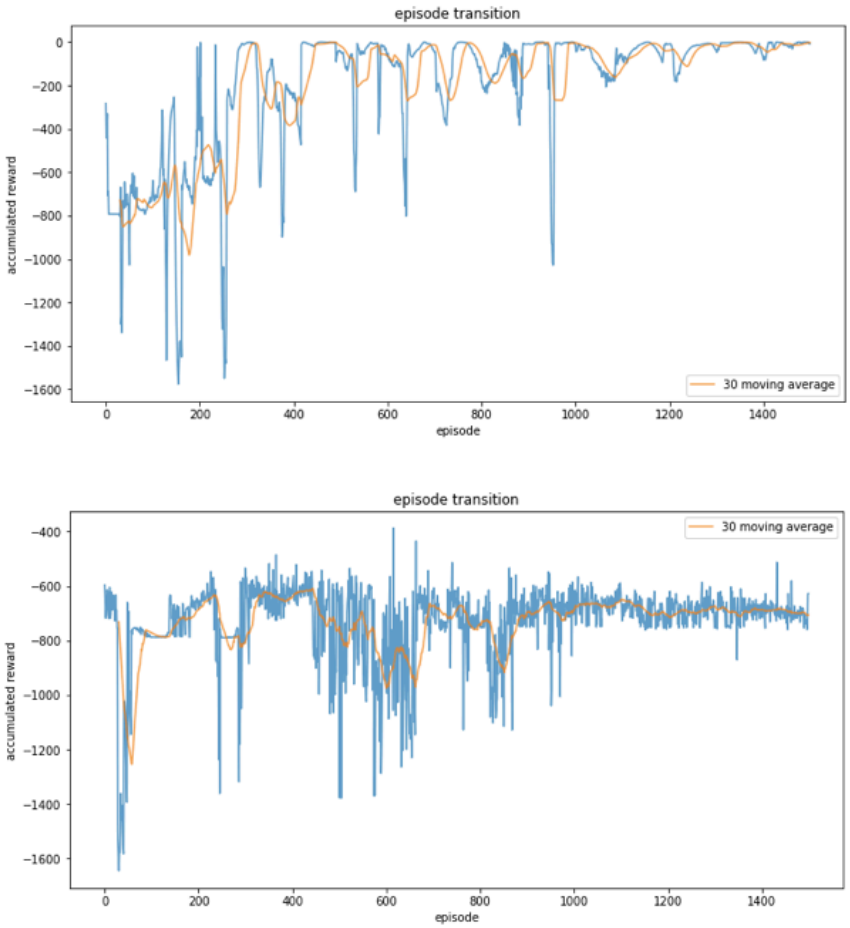

図 1　毎時刻系とサンプル値系のエピソード報酬

では毎時刻系で得られているエピソード報酬を得られる制御則を, サンプル値系でも学習するにはどのくらいのエピソードが必要なのか, この学習の続きを見ていく.

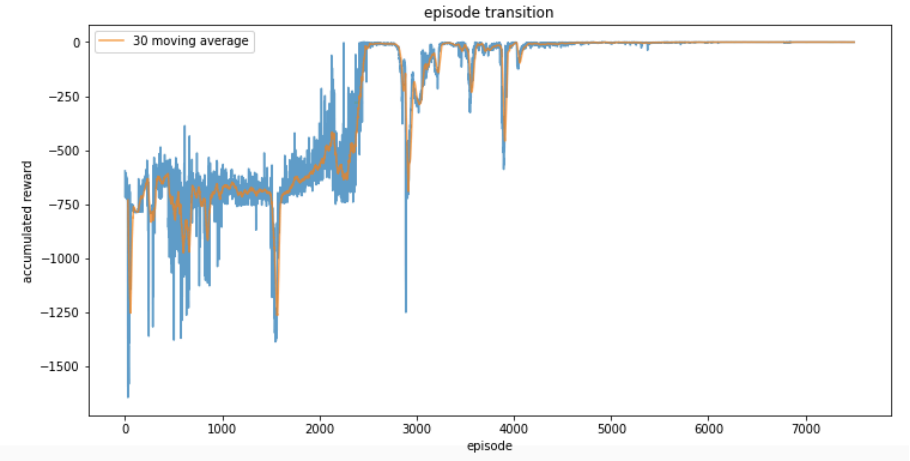

図 2　サンプル値系のさらなる学習

図 2 を見ると, 4000 エピソード付近で最適制御則を学習できていることがわかる. これは毎時刻系の約 3 倍であり, $\tau = 5$ よりも小さな値であるため, 1 回の制御則更新の効率は毎時刻系よりも $\frac{5}{3}$ 倍程度良くなっていることがわかる. したがって, システムの 1 ステップの時間発展の計算にかかる時間が, ニューラルネットワークのパラメータ更新にかかる計算時間よりも十分に小さい計算機上では, ステップ数が増えてしまうことは大きな

脅威とはならないかもしれない.

(しかしこれは 1 回の学習から得られた考察なので, より強く断定するには少し学習タスクのサンプル数を増やす必要がある.)

2　セルフトリガー制御にむけて

強化学習エージェントは制御則 $\pi(s)$ を $\pi_\theta(s)$ のようにパラメトライズし, その最適パラメータを模索する. ここでは, 毎時刻系の最適エージェントのパラメータを初期値として, サンプル間隔 τ を変えた時に制御性能を満たすことを学習できるのかどうか検証したい.

2.1　毎時刻系の最適エージェント

方策 $\pi_\theta(s)$ は以下のようなニューラルネットワークモデルを用いる.

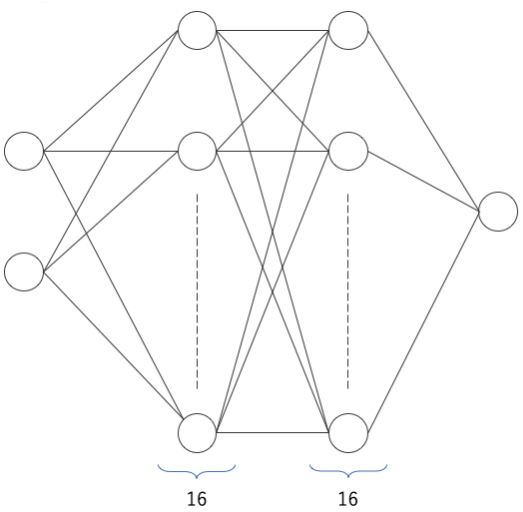

図 3　方策 $\pi(s)$ の表現モデル

この時各層の一部のニューロンの活性化関数を線形関数 $y = x$ にして, 重みを適切に設定してやることで線形フィードバック制御則を実現することが可能である. 毎時刻系で原点付近を安定化させる方策を初期値として, τ を変えても原点付近の安定化方策の学習ができるか試みた.

いま, 図 3 の出力層の活性化関数も線形関数としているが, こうしてしまうと値域が $(-\infty, \infty)$ となってしまう. これは入力信号に制限がある場合には不都合がある. 例えば入力制限を $[-10, 10]$ とし, これ以外の入力を加えようとした場合, 制限集合の最も近い入力がシステムに加わるとする. 現在用いている DDPG アルゴリズムは actor-critic 構造となっており, critic 側の損失が非常に重要である. critic の損失関数には TD 誤差を用いたものが使用されている. ここでいう TD 誤差とは

$$\delta_{TD} = Q(s_t, a_t) - \{r_t + \gamma Q(s_{t+1}, \pi(s_{t+1}))\} \tag{4}$$

である.

今, critic の損失関数の値が発散してしまうと言う問題点が発生しています.(おそらくはこれは, $\pi(s)$ の出力と実際にシステムに加えられる入力が異なることによって, TD 誤差が発散していることだろうと考えていますが, なぜそうなるのかわかっていません.)

出力層の活性化関数を tanh のスカラー倍にすることで, 予想している原因を回避できるがこれでは毎時刻系の最適エージェントをニューラルネットワークモデルとして設計することができない. よって模倣学習を用いるか, 毎時刻系エージェントを強化学習によって得るかする必要がある.