J-PARC

加速器・ビーム物理の機械学習ワークショップ2025

機器制御に向けた強化学習アルゴリズムの検討

Investigation of Reinforcement Learning Algorithms for Machine Control

野村 昌弘（J-PARC/JAEA）

motivation

最近加速器の制御に強化学習を導入したいという雰囲気を感じるし、非常に興味深い。

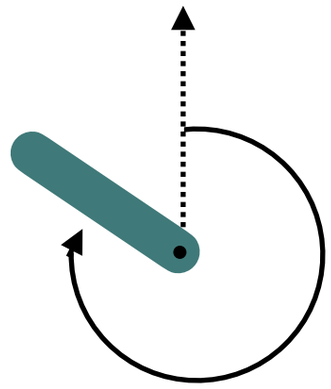
冷却水の温度制御にも使ってみたい。

強化学習では、色々行動をしながら学習する

直接実際の機器で行うにはリスクが伴う

実際の機器を使用しない強化学習のアルゴリズム — Offline強化学習 —

Offline強化学習がどんなものか、実際に使えるかどうかを調べるために、
強化学習で使われる”倒立振子”に適用してみた。



倒立振子

強化学習と倒立振子

強化学習：与えられた状態から最適な行動を選択できるようにする。 Ref.[1-3]

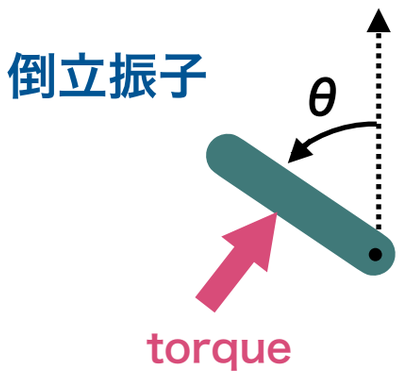
倒立振子：OpenAI Gymに含まれるゲームの一種 Ref.[4,5]

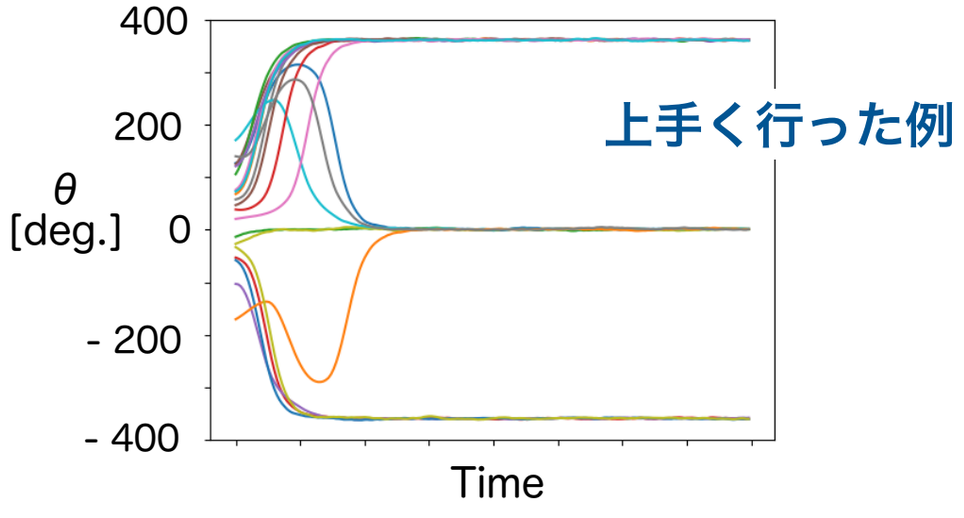
目標：与えられた状態で最適な行動を行い振子を立たせる

状態：state = [角度θ、角速度θ]　行動：action = torque

目標を達成する為に報酬を設定

報酬：$r = -(\theta^2 + 0.1 * \dot{\theta} + 0.001 * \mathrm{action})$ 報酬が最大になる様に行動を決定





Offline 強化学習

CQL (Conservative Q-Learning)を採用

事前に得たデータを使用するアルゴリズム

Ref.[6-8]

事前データ

倒立振子の場合

$\theta, \dot{\theta}$ + torque —> next_θ, next_$\dot{\theta}$

加速器制御：

透過率Xが機器Zの値をYに変えると透過率X'になった。

温度制御：

水温がTまで上がったので、冷却ファンの回転数がCになり水温がT'となった。

この種のデータは既にあるか、あるいは取得可能

どの様にして最適な行動を決めるか

Q_learning (Conservative Q-learning)

Q(s, a)値：ある状態sで行動aを取った時、
将来どのくらいの累積報酬が得られるかを表す値。
-> Q値が高い行動が最適な行動

Q値をNeural Network(NN)で記述し、

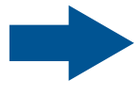
Bellman Loss

$$L_{Bellman} = 1/2 \cdot (r + \gamma \cdot \max_{a'} Q(s', a') - Q(s,a))^2$$

Bellman Loss に事前データを用いてNNを求める。

通常の強化学習でも用いられる手法

事前データの無い領域でもQ値は計算できてしまう。
そのQ値が大きいとその行動をとってしまう。



今までに無い行動
適切か？ 大丈夫か？

事前データの無い領域でQ値をどう評価するか？

Conservative (Conservative Q-learning)

正則化項

$Loss = L_{Bellman} + \alpha * cql\_reg$　　α：正則化強度(パラメータ)

cql_reg = Q_pred_Max - Q_data　　<- 考え方、正確では無い

Q_pred_Max：NNで予測したQの中での最大値 (実データでは無い)。

Q_data：事前データ(実データ)

cql_reg：最大と予測されるQと実データのQとの差

cql_regが小さくなと、

最大と予測されるQと実データのQが近づき、実データに近い行動を選択する様になる。

保守的、conservative

どれだけ実データに近い行動をとるか、

最適と予測されるデータ外の行動をとるか

これを決めるのが正則化強度α

CQLを倒立振子に適用し、
以下の項目について調べてみる

どの様なデータがどれだけ必要か？

事前データに成功(倒立)したデータが必要か？

正則化強度αの影響は？

$$loss = loss\_0 + \alpha * cql\_reg$$

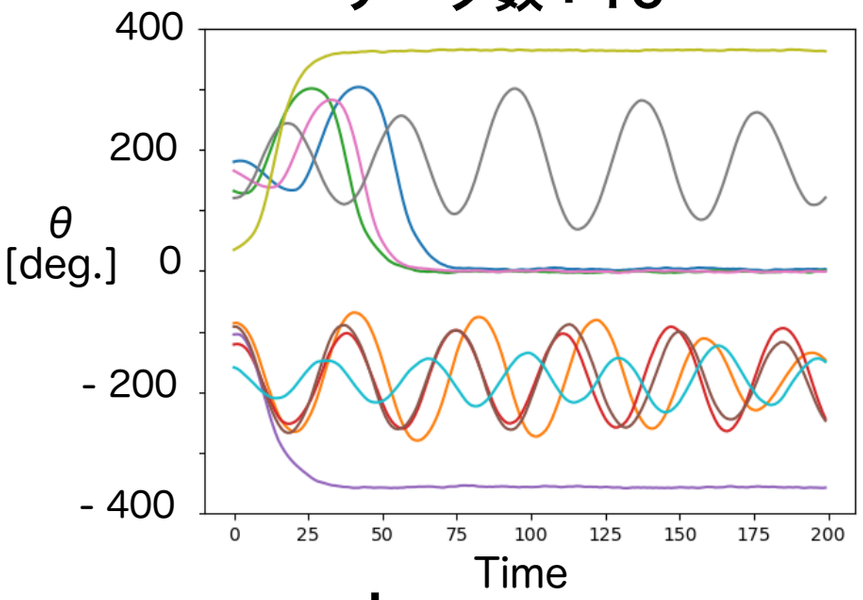
どの様な事前データがどのくらい必要か

α = 1.0

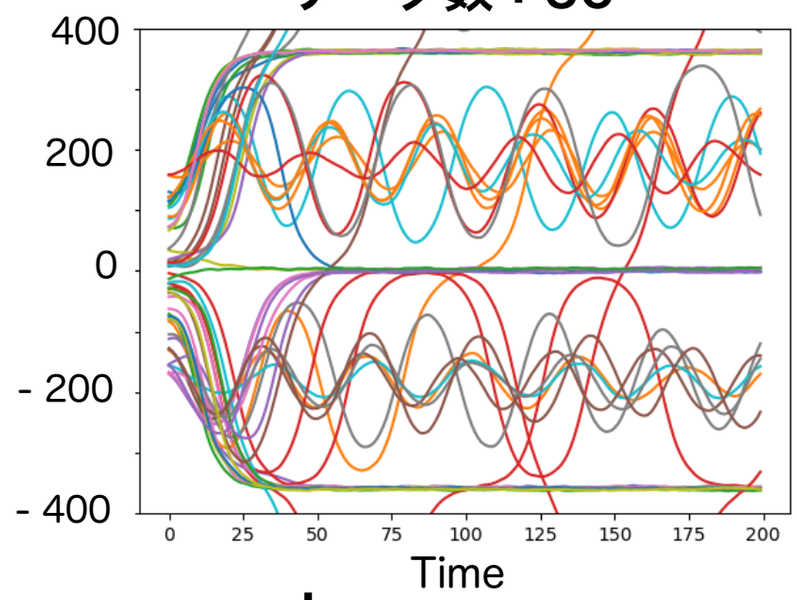
事前データ(成功50%, 失敗50%)

事前データ

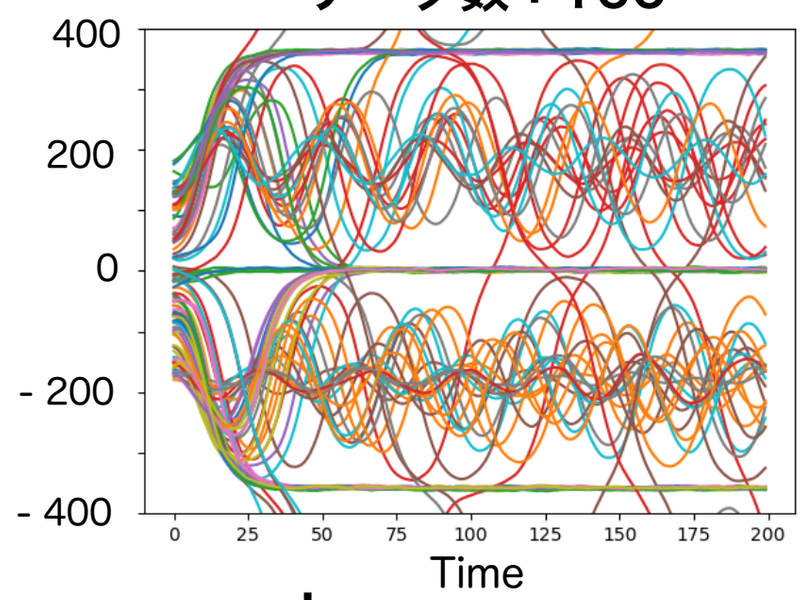
データ数：10



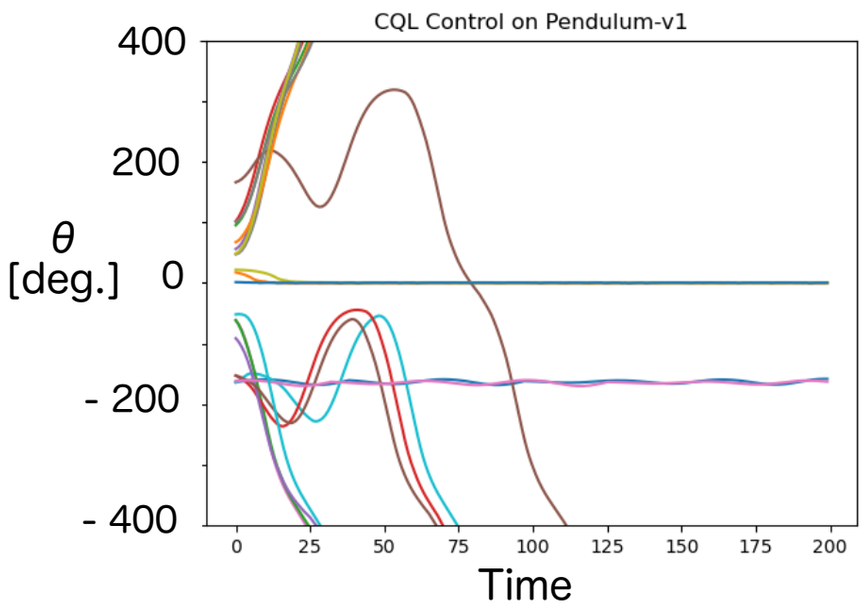
データ数：50

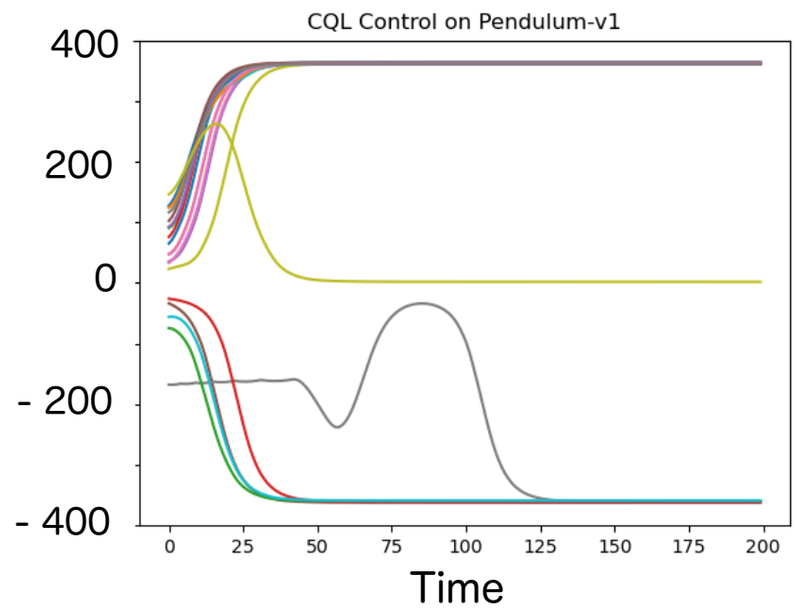


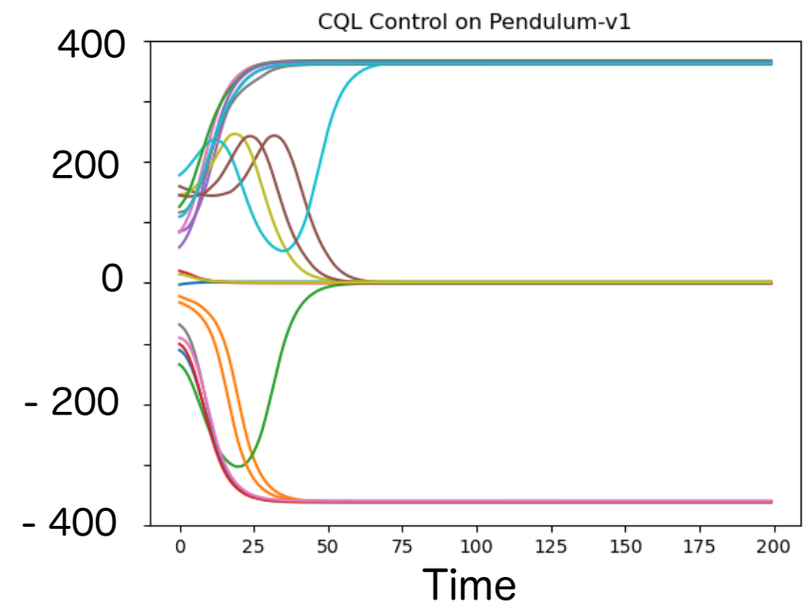
データ数：100



強化学習後 環境に適用







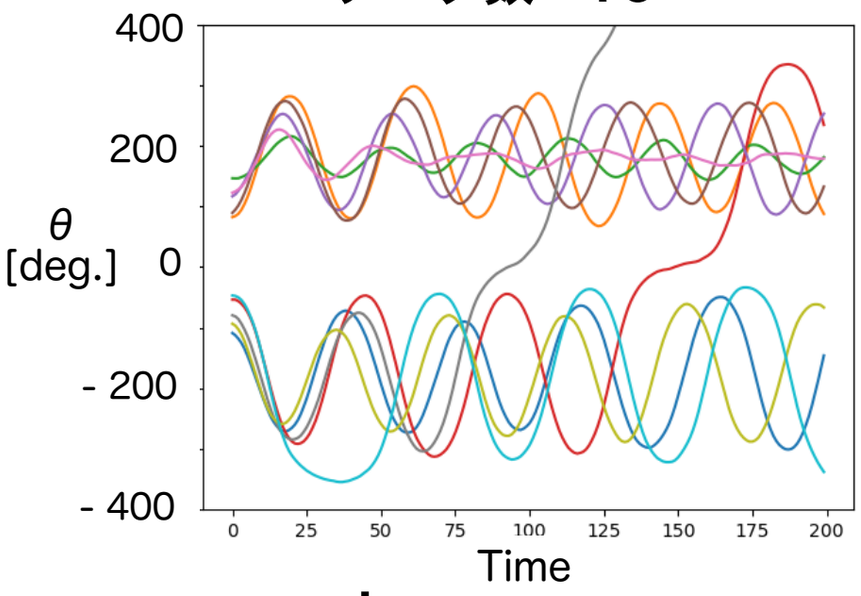
どの様な事前データがどのくらい必要か

$\alpha = 1.0$

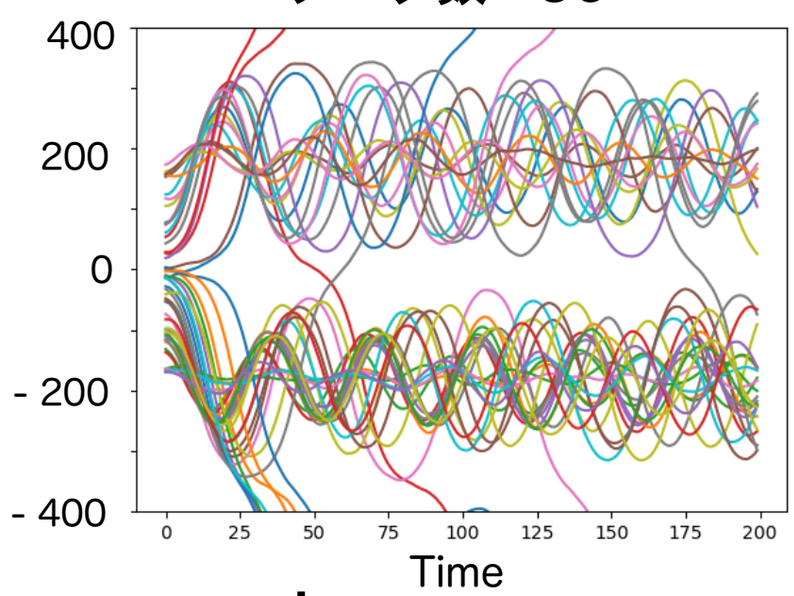
事前データ(成功0%, 失敗100%)

事前データ

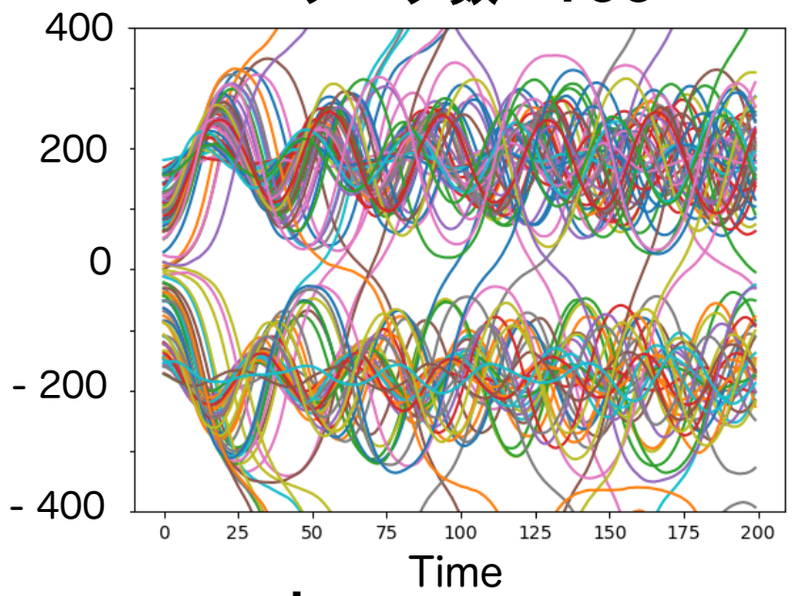
データ数：10



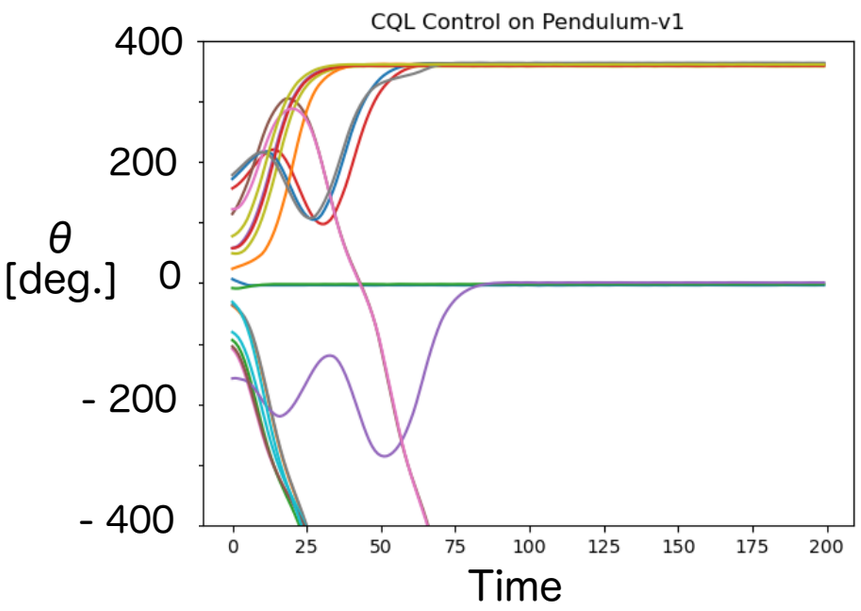
データ数：50

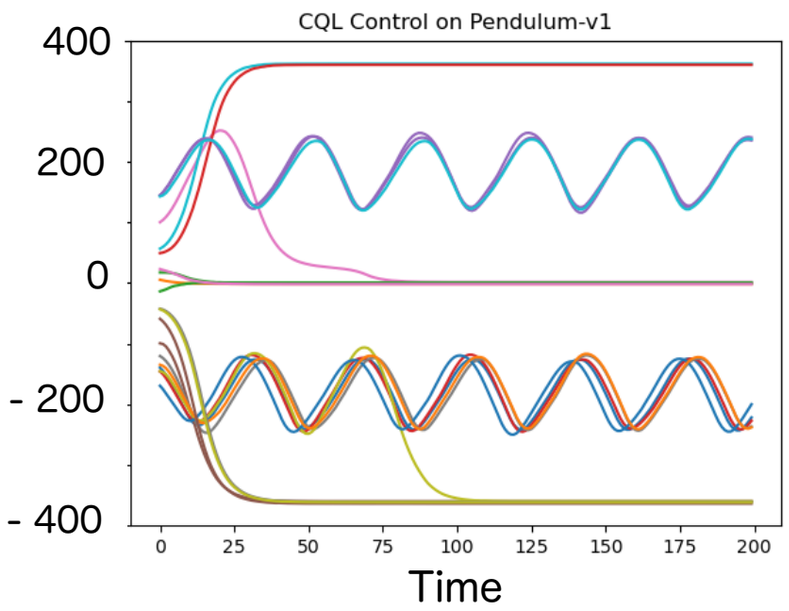


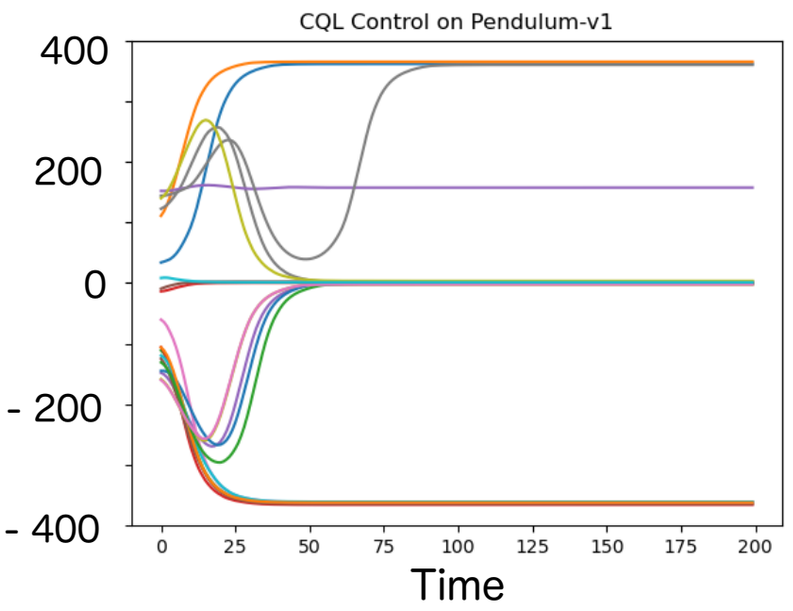
データ数：100



強化学習後
環境に適用







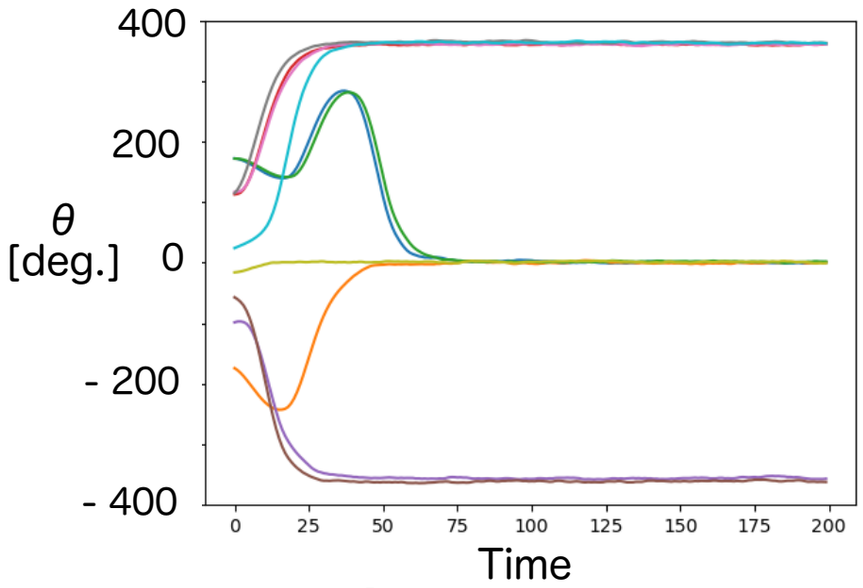
どの様な事前データがどのくらい必要か

$\alpha = 1.0$

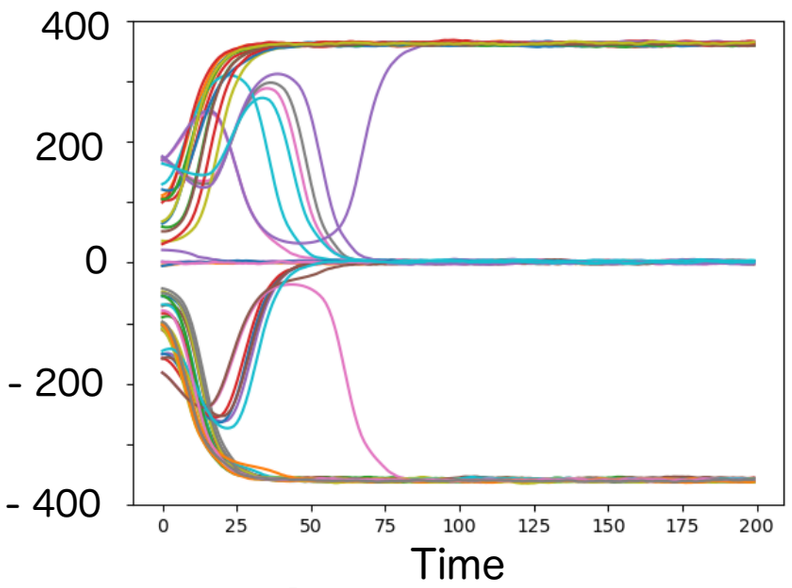
事前データ(成功100%, 失敗0%)

事前データ

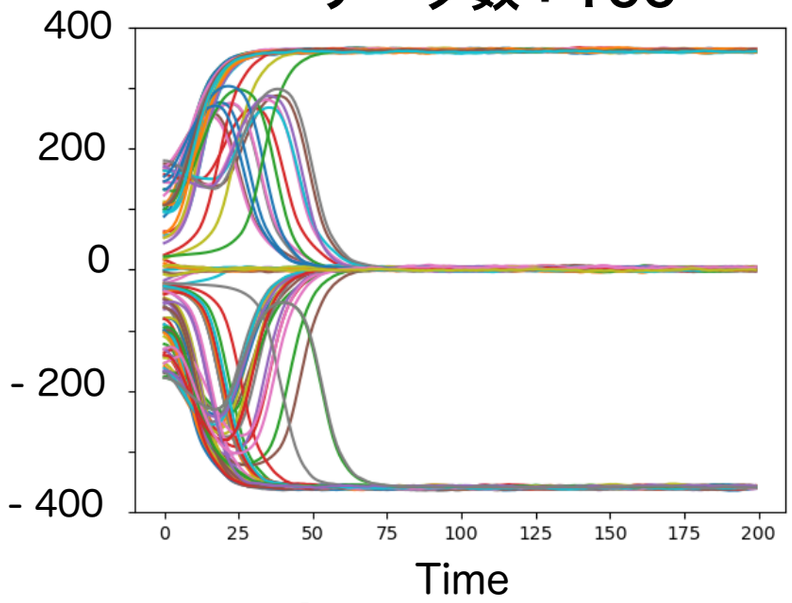
データ数：10



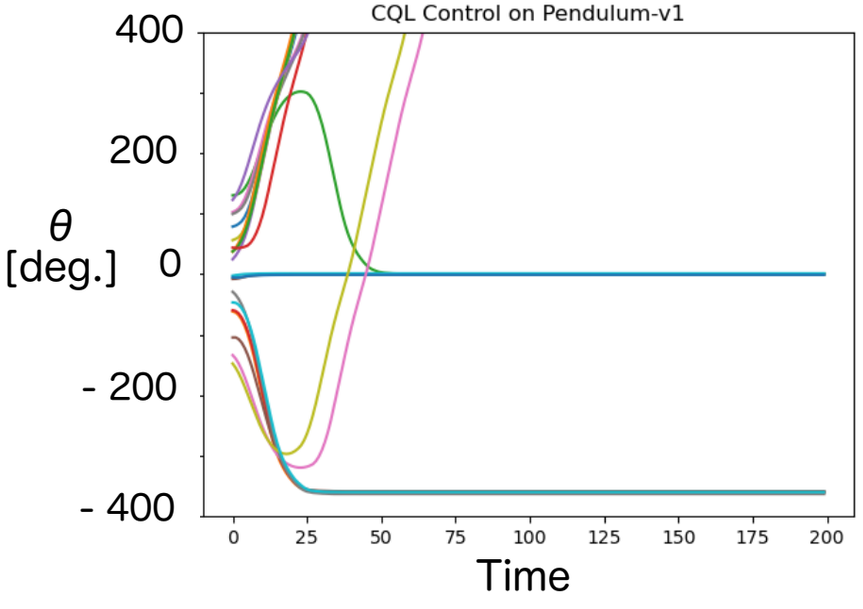
データ数：50

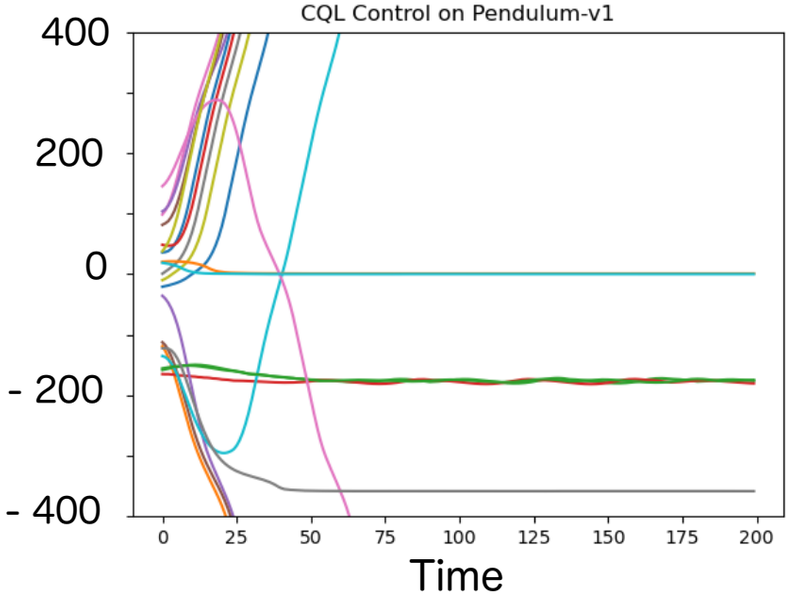


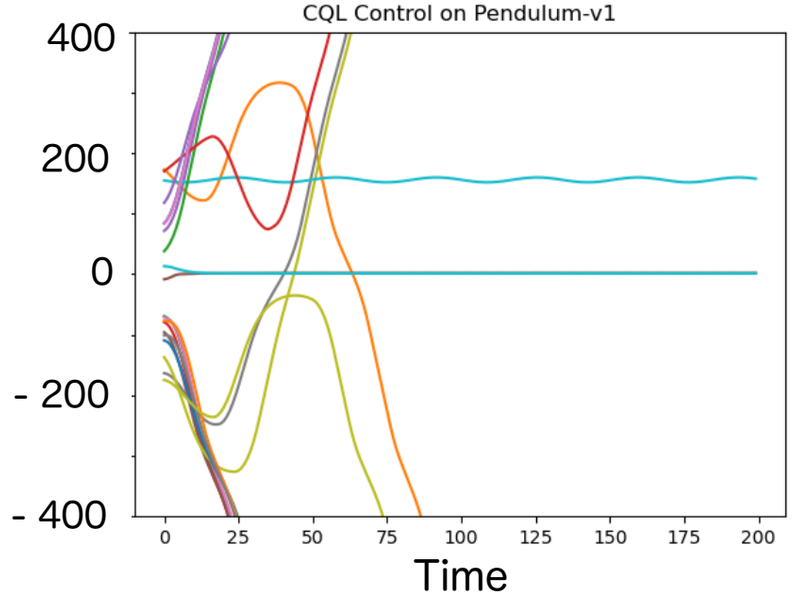
データ数：100



強化学習後
環境に適用







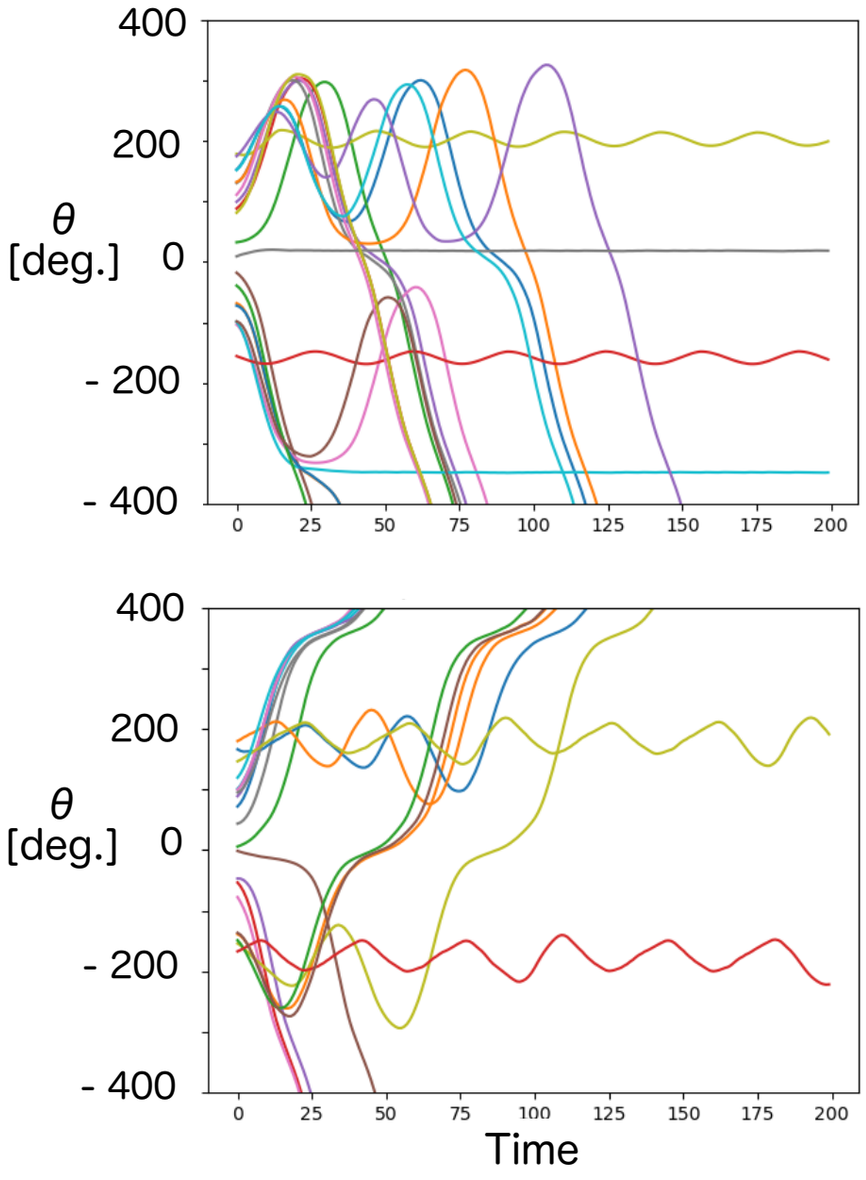
正則化強度αの影響

失敗100%のデータ100個を使用　　loss = loss_0 + α * cql_reg

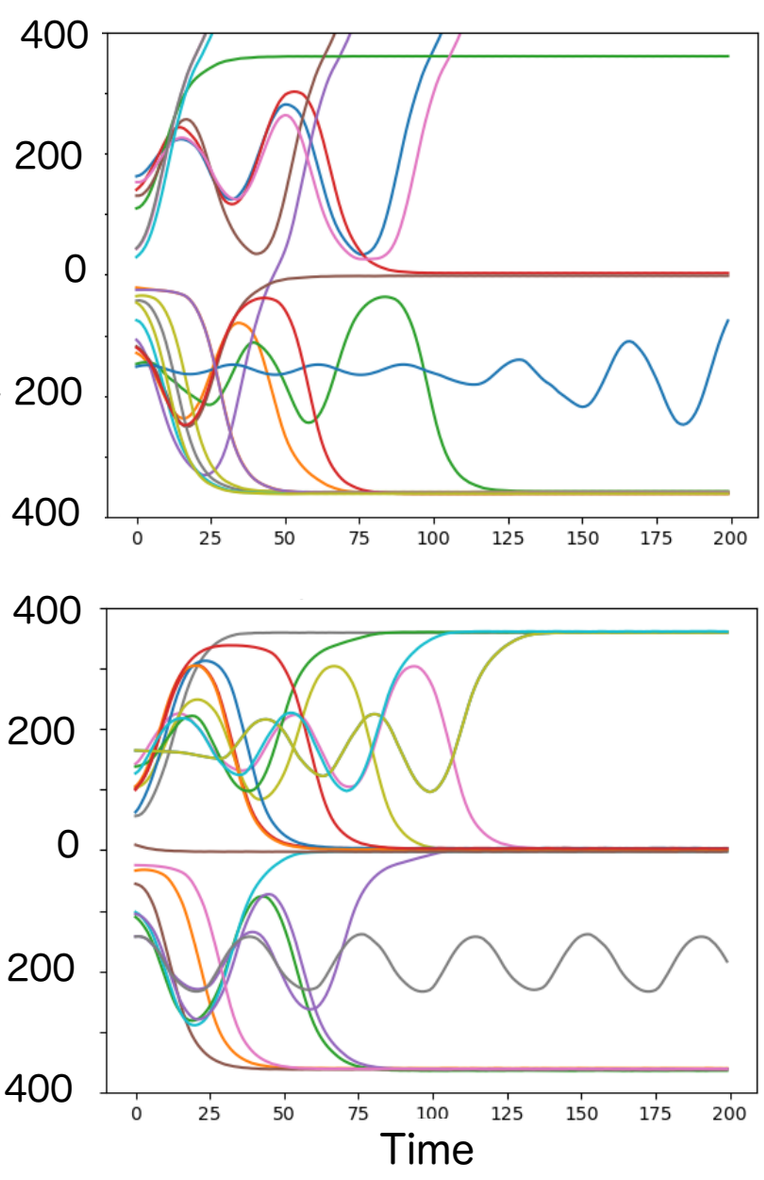
α大：行動分布がデータ分布に近くなる ⟷ α小：データ外の行動も選択しやすくなる

環境に適用

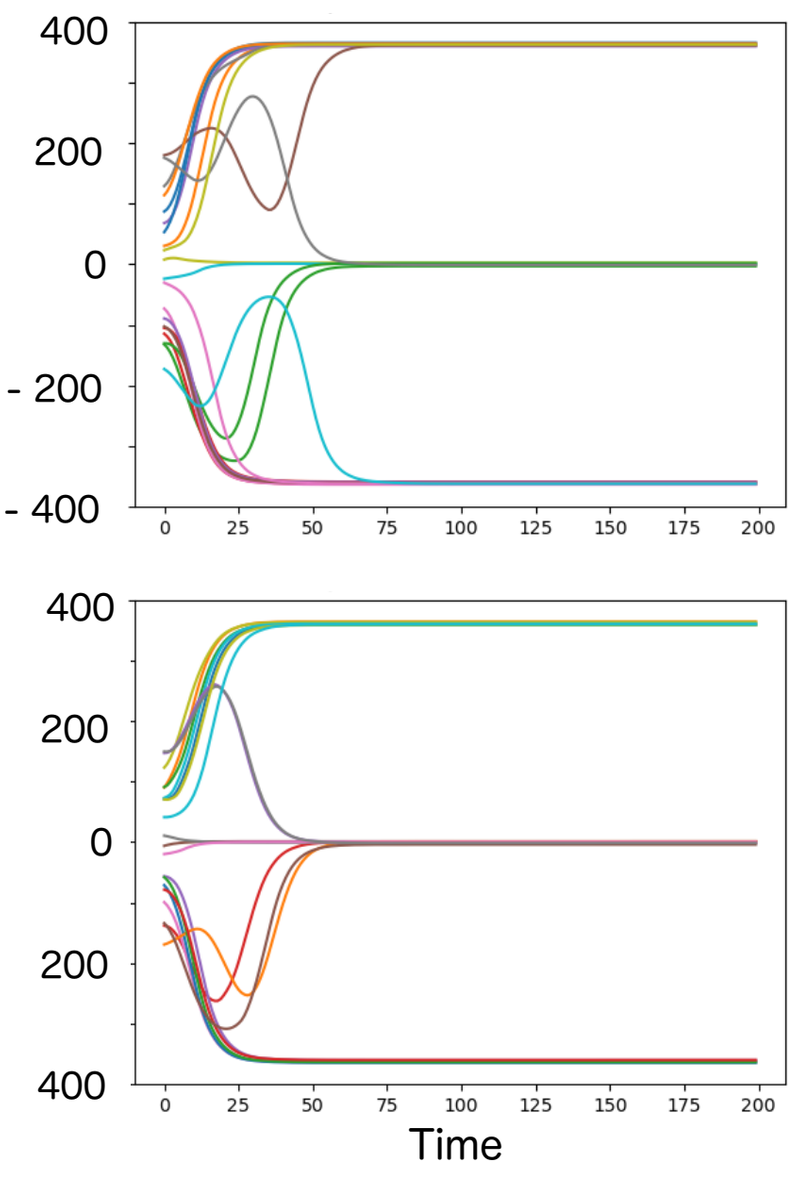
α = 50



α = 20



α = 1.0



倒立振子に適用して得た知見

正解データが必ずしも必要では無い、多様性が重要。

正則化強度α

α	行動の傾向	リスク
大きい	行動分布がデータ分布に近くなる	探索性の欠如
小さい	データ外アクションも選択しやすくなる	不適切な行動を選ぶ可能性

今回は実際の環境で挙動を調べたが、CQLでは環境を使った評価はできない
　　実際の環境で使用する場合は、大きいαから徐々に小さいαへ

実際の環境での動きを事前に知りたい、事前データから推定できないか？

　環境での動作：state, action -> next_state

PILCO (Probabilistic Inference for Learning COntrol)

Ref.[9-11]

PILCOは、事前データ等からガウス過程に基づき環境での動きを推定。
ガウス過程で求めた平均と分散を使って報酬を値ではなく、期待値として計算。
その結果、分散が大きい行動は一般に報酬の期待値を下げるため避けられる。
分散が大きい行動：その行動を取ったとき、次を正しく予測できない行動。

CQLの動作の検証

X = [state, action]
Y = [next_state - state]

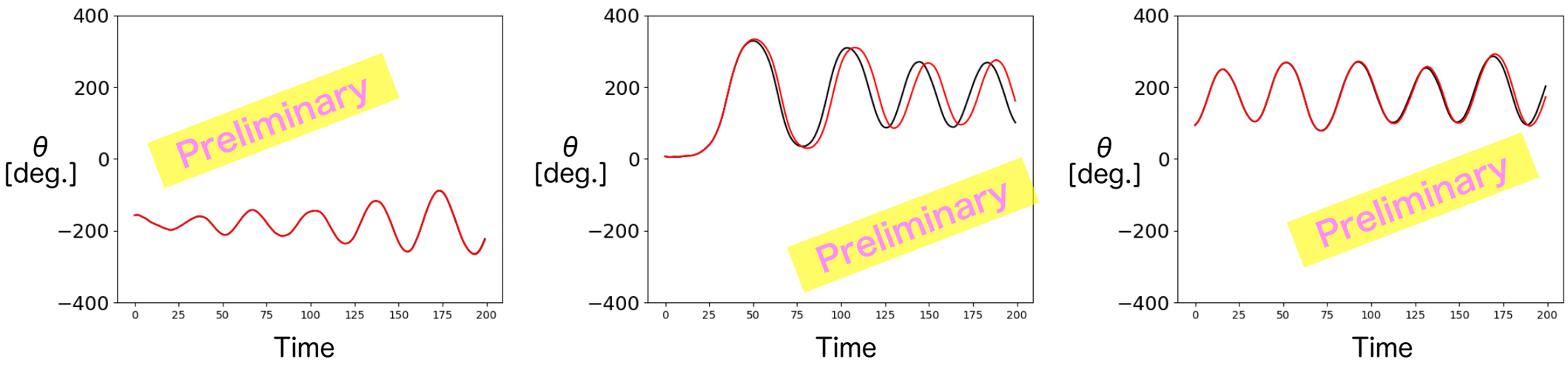
```
kernel = GPy.kern.RBF(input_dim=input_X.shape[1])
gp_model = GPy.models.GPRegression(input_X, output_Y, kernel)
```

gp_modelは、RBF kernelを使ってinput_Xからoutput_Yを推定する回帰モデル。

データ数：50

ガウス過程で推定した環境での動き

黒：環境(正解)での動き
赤：GP環境での動き



COMBO (Conservative Offline Model-Based Policy Optimization)

Ref.[12,13]

COMBOでは、Neural Network により環境での動きを予測。

その環境での動きから未知のデータを作成し、

事前のデータと未知のデータ(リスクを考慮)から行動を決定。

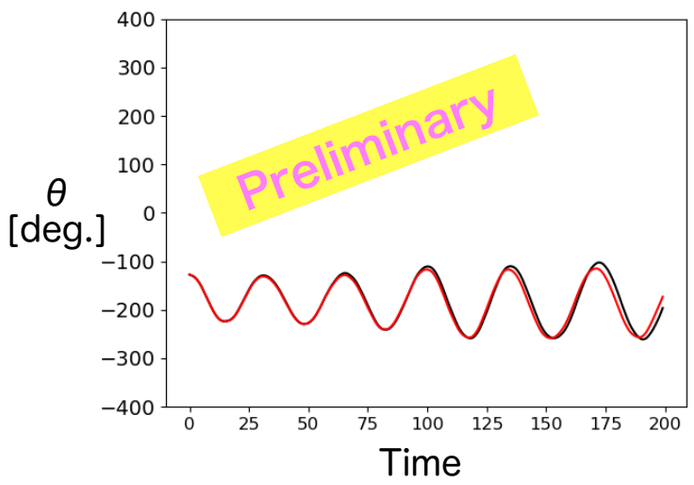
CQLから派生。

CQLの動作の検証

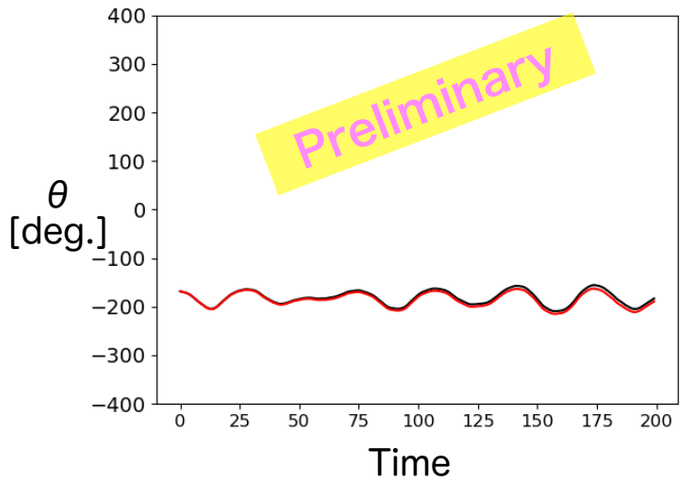
```
x = layers.Concatenate()([state, act])
.....
next_state = layers.Dense(state_dim)(x)
NN_model = Model(inputs=[state, act], outputs= next_state)
```

NN_modelは、[state, act] から[next_state] を出力するNNモデル。

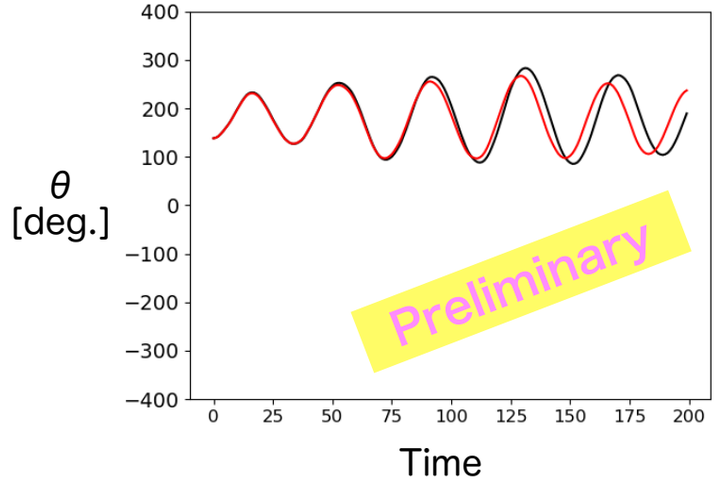
data_num = 100



NNで予測した環境での動き



黒：環境(正解)での動き
赤：NN環境での動き



まとめ

Offline強化学習アルゴリズムの一つCQLを倒立振子に適用してみた

CQLは、取得したデータ以外の行動を抑制しながら方策を得るアルゴリズム

見知らぬ一歩は、慎重に

ガウス過程やNNによって事前データから環境での動きを推定し、
正則化強度αの調整や行動の確認をするのが有効かと思える

PILCOやCOMBOのアルゴリズム自体も非常に興味深い

加速器全体の制御に使えるかは、
今回の倒立振子の例とは大きく環境や動作等が違いすぎて判断はできない

倒立振子と同じ様な環境や動作の冷却水温度の制御等には使えるのではないか

発表した内容は理解してもらう事を優先した為、正確さに欠ける記述があります。
間違って私が理解しているかもしれません。
references等を参考にして下さい。

references

強化学習全般

[1] 強化学習から信頼できる意思決定へ　梶野洸、宮口航平、恐神貴行、岩城諒、和地瞭良　サイエンス社
[2] Pythonで学ぶ強化学習　久保隆宏　講談社
[3] 強化学習　森村哲郎　講談社

gym -> gymnasium

[4] https://github.com/openai/gym
[5] https://github.com/Farama-Foundation/Gymnasium

CQL (Conservative Q-Learning)

[6] https://arxiv.org/abs/2006.04779
[7] https://qiita.com/aiueola/items/90f635200d808f904daf
[8] https://horomary.hatenablog.com/entry/2022/10/30/111031

PILCO (Probabilistic Inference for Learning COntrol)

[9] M. Deisenroth, D. Fox, and C. Rasmussen.
Gaus- sian processes for data-efficient learning in robotics and control.
IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 408–423, 2015.
[10] https://speakerdeck.com/shunichi09/pilco
[11] https://github.com/aidanscannell/pilco-tensorflow

COMBO (Conservative Offline Model-Based Policy Optimization)

[12] https://arxiv.org/abs/2102.08363
[13] https://github.com/Shylock-H/COMBO_Offline_RL?utm_source=chatgpt.com