Initial Value Problem Enhanced Sampling for Closed-Loop Optimal Control Design with Deep Neural Networks

Nov 2022

- Put on arxiv on 2022/09/09.
- The authors are: Xuanxi Zhang, Jihao Long, Wei Hu, Weinan E, and Jiequn Han

Open-loop and close-loop control

There are 2 types of optimal control problems.

- Open-loop control:
  - Deals with the problem with a given initial state;
  - Relatively easy to solve.
- Closed-loop control:
  - Aims to find the optimal control policy as a function of the state;
  - More general;
  - More difficult to deal with.

Using NN to solve open-loop control

- Policy based:
  - Parameterizes the policy function by neural networks;
  - Minimizes the mean value of the total cost.
- Value based:
  - Parameterize the control function or value function using NNs;
  - Obtain the optimal trajectories for different initial points and collect them as training data, and train the NN models to fit the closed loop optimal controls (or optimal values) along the trajectories.

Adaptive sampling (for initials)

The learned controller may deteriorate badly at some difficult initial states even though the error is small in the average sense. Several adaptive sampling strategies regarding the initial points are hence proposed for improvement.

- For example, Nakamura-Zimmerer et al. (2021a) propose to sample initial points with large gradients of the approximated value function since the value function around these points tend to be steep and hard to learn.
- Landry et al. (2021) propose an adversarial method to add initial points with large gaps between learned values and optimal values.

Problems

- Existing sampling methods all focus on choosing optimal paths according to different initial points and ignore the effect of dynamics;
- The paths controlled by the NN will deviate from the optimal paths further and further over time due to the accumulation of the error;
- Applying adaptive sampling only on initial points is insufficient for solving the challenging problems

Distribution mismatch phenomenon

The discrepancy between the state distribution of the training data and the state distribution generated by the NN controller typically increases over time and the training data fails to represent the states encountered when the trained NN controller is used.

Problem formulation (open-loop)

Consider the following deterministic controlled dynamical system:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), t \in [t_0, T] \\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \end{cases}$$

Solving open-loop optimal control problem means to find a control path $u^* : [t_0, T] \to \mathcal{U}$ to minimize

$$J(\boldsymbol{u}; t_0, \boldsymbol{x}_0) = \int_{t_0}^{\mathrm{T}} L(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{d}t + M(\boldsymbol{x}(T))$$

s.t. $(x, \boldsymbol{u})$ satisfy the system above

Closed-loop

In contrast to the open-loop control being a function of time only, closed-loop control is a function of the time-state pair (t, x).
Given a closed-loop control $u : [0, T] \times \mathbb{R}^n \to \mathcal{U}$, we can induce a family of the open-loop controls with all possible initial time-state pairs (t_0, x_0) :

$$u\left(t; t_0, x_0\right) = u\left(t, x_u\left(t; t_0, x_0\right)\right),$$

where $x_u\left(t; t_0, x_0\right)$ is defined by the following initial value problem (IVP):

$$\mathrm{IVP}\left(x_0, t_0, T, u\right) : \begin{cases} \dot{x}_u\left(t; t_0, x_0\right) = f\left(t, x_u\left(t; t_0, x_0\right), u\left(t, x_u\left(t; t_0, x_0\right)\right)\right) \\ t \in [t_0, T] \\ x_u\left(t_0; t_0, x_0\right) = x_0. \end{cases}$$

Obtain closed-loop control through open-loop control

- Since IVPs can be easily solved, one can handle the open-loop control problems with all possible initial time-state pairs if a good closed-loop control solution is available;
- In this paper, our goal is to find a near-optimal closed-loop control $\hat{u}$ such that for $x_0 \in X \subset \mathbb{R}^n$ with X being the set of initial states of interest, the associated total cost is near-optimal, i.e.,

$$\left| J\left(\hat{u}\left(\cdot; 0, x_0\right); 0, x_0\right) - J\left(u^*\left(\cdot; 0, x_0\right); 0, x_0\right) \right|$$

  is small.

Supervised-learning-based approach

- The first step is to generate training data by solving the open-loop optimal control problems with zero initial time and initial states randomly sampled in X
- Then, the training data is collected by evenly choosing points in every optimal path:

$$\mathcal{D} = \left\{ \left(t^{i,j}, \boldsymbol{x}^{i,j} \right), \boldsymbol{u}^{i,j} \right\}_{1 \le i \le M, 1 \le j \le N},$$

  where M and N are the number of sampled training trajectories and the number of points chosen in each path, respectively.
- Finally, a function approximator (mostly neural network, as considered in this work) with parameters θ is trained by solving the following regression problem:

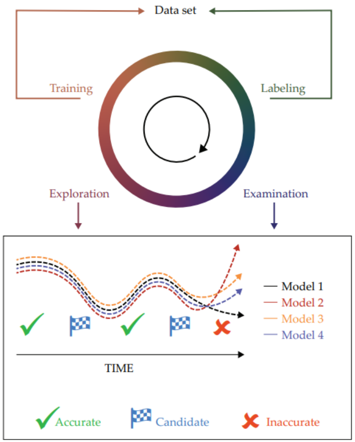
$$\min_{\theta} \sum_{i=1}^{M} \sum_{j=1}^{N} \left\| u^{i,j} - u^{\mathrm{NN}} \left(t^{i,j}, x^{i,j}; \theta \right) \right\|^2,$$

  and gives the NN controller u^{NN}.

Improve the efficiency

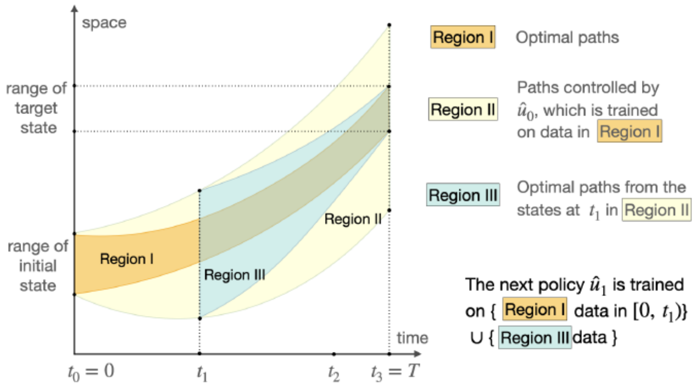
- Use the adaptive sampling method. Adaptive sampling methods aim to sequentially choose the time-state pairs based on previous results to improve the performance of the NN controller.
- Improve the efficiency of data generation, i.e., solving the open-loop optimal control problems.
- Improve the learning process of the neural networks.

ELT



Data set
Training
Labeling
Exploration
Examination
Model 1
Model 2
Model 3
Model 4
TIME
Accurate
Candidate
Inaccurate

Distribution mismatch phenomenon


space
range of target state
range of initial state
Region I
Region II
Region III
time
$t_0 = 0$
t_1
t_2
$t_3 = T$
Region I Optimal paths
Region II Paths controlled by $\hat{u}_0$, which is trained on data in Region I
Region III Optimal paths from the states at t_1 in Region II
The next policy $\hat{u}_1$ is trained on { Region I data in $[0, t_1)$} ∪ { Region III data }

(a) first iteration

Distribution mismatch phenomenon

- The error between state $\boldsymbol{x}$ driven by $\boldsymbol{u}^*$ and $\boldsymbol{u}^{\mathrm{NN}}$ accumulates and makes the discrepancy between $\mu_{\boldsymbol{u}^*}(t)$ and $\mu_{\boldsymbol{u}^{\mathrm{NN}}}(t)$ increases over time.
- Hence, the training data fails to represent the states encountered in the exploiting process, and the error between $\boldsymbol{u}^*$ and $\boldsymbol{u}^{\mathrm{NN}}$ dramatically increases when t is large.

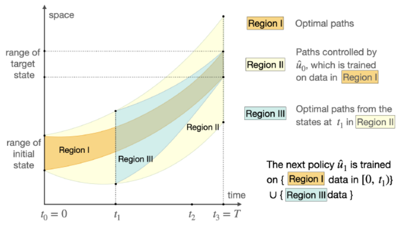
IVP

- Given predesigned (not necessarily even-spaced) temporal grid points $0 = t_0 < t_1 < \cdots < t_K = T$, we first generate a training dataset S_0 by solving open-loop optimal control problems on the time interval $[0, T]$ starting from points in X_0, a set of initial points sampled from μ_0 and train the initial model $\hat{u}_0$.
- Under the control of $\hat{u}_0$, the generated trajectory deviates more and more from the optimal trajectory. So we stop at time t_1, i.e., compute the IVPs using $\hat{u}_0$ as the closed-loop control and points in X_0 as the initial points on the time interval $[0, t_1]$, and then on the interval $[t_1, T]$ solve new optimal paths that start at the end-points of the previous IVPs

IVP

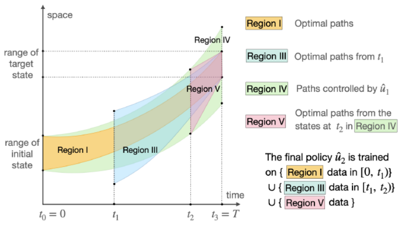
- Next, the new training dataset S_1 is composed of new data (between t_1 and T) and the data before time t_1 in the dataset S_0, and we train a new model $\hat{u}_1$ using S_1.
- We repeat this process to the predesigned temporal grid points $t_2, t_3, \cdots$ until end up with T.

IVP



space
range of target state
range of initial state
Region I
Region II
Region III
time
$t_0 = 0$
t_1
t_2
$t_3 = T$
Region I Optimal paths
Region II Paths controlled by $\hat{u}_0$, which is trained on data in Region I
Region III Optimal paths from the states at t_1 in Region II
The next policy $\hat{u}_1$ is trained on { Region I data in $[0, t_1)$} ∪ { Region III data }

(a) first iteration



space
range of target state
range of initial state
Region I
Region III
Region IV
Region V
time
$t_0 = 0$
t_1
t_2
$t_3 = T$
Region I Optimal paths
Region III Optimal paths from t_1
Region IV Paths controlled by $\hat{u}_1$
Region V Optimal paths from the states at t_2 in Region IV
The final policy $\hat{u}_2$ is trained on { Region I data in $[0, t_1)$} ∪ { Region III data in $[t_1, t_2)$} ∪ { Region V data }

(b) second iteration

IVP

Algorithm 1 IVP enhanced sampling method for closed-loop optimal control design

```
Input: Initial distribution $\mu_0$, number of time points $K$, temporal grid points $0 = t_0 < t_1 < \cdots < t_K = T$, time step $\delta$, number of initial points $N$.
Initialize: $S_{-1} = \emptyset$, $\hat{\boldsymbol{u}}_{-1}(t, \boldsymbol{x}) = 0$.
Independently sample $N$ initial points from $\mu_0$ to get an initial point set $X_0$.
for $i = 0, 1, \cdots, K-1$ do
    For any $\boldsymbol{x}_0 \in X_0$, compute IVP $(\boldsymbol{x}_0, 0, t_i, \hat{\boldsymbol{u}}_{i-1})$ according to (2).    ▷Exploration
    Set $X_i = \{\boldsymbol{x}_{\hat{\boldsymbol{u}}_{i-1}}(t_i; 0, x_0) : x_0 \in X_0\}$.
    For any $\boldsymbol{x}_i \in X_i$, call the open-loop optimal control solver to obtain $\boldsymbol{x}^*(t; t_i, \boldsymbol{x}_i)$ and $\boldsymbol{u}^*(t; t_i, \boldsymbol{x}_i)$ for $t \in [t_i, T]$.    ▷labeling
    Set $\hat{S}_i = \{(t, \boldsymbol{x}^*(t; t_i, \boldsymbol{x}_i), \boldsymbol{u}^*(t; t_i, \boldsymbol{x}_i)) : \boldsymbol{x}_i \in X_i, t \in [t_i, T], (t - t_0)/\delta \in \mathbb{N}\}$.
    Set $S_i = \hat{S}_i \bigcup \{(t, \boldsymbol{x}, \boldsymbol{u}) : t < t_i, (t, \boldsymbol{x}, \boldsymbol{u}) \in S_{i-1}\}$.
    Train $\hat{\boldsymbol{u}}_i$ with dataset $S_i$.    ▷Training
end for
Output: $\hat{\boldsymbol{u}}^{K-1}$.
```

Experimental results

We see the original paper for experimental results.