Optimal control course project report

Han-Dong Lim & Kihong Park

December 29, 2022

1 Introduction

Recent advance of deep learning methods have lead to exceptional achievements in control applications (Hansen et al., 2022; Janner et al., 2021; Jiang et al., 2022; Karl et al., 2016; Srinivas et al., 2018; Watter et al., 2015). Hansen et al., 2022 combines cross entropy method and model-free reinforcement learning (RL) to plan in task-orientated latent space. Janner et al., 2021; Jiang et al., 2022 consider a trajectory as one sequence and use Transformer (Vaswani et al., 2017) to plan walking motion of humanoid. Srinivas et al., 2018 unifies planning and learning representation and shows good performance in terms of imitation learning.

Learning representation has been a major topic in the artificial intelligence literature. Regarding the research on the so-called latent space that learns a representation in low-dimensional space conserving important aspects of original space, has a long history. Variational auto-encoder (VAE) (Kingma, Welling, et al., 2019) has been widely used to model a probability distribution of the underlying latent space. In particular, it has shown remarkable success in generative tasks such as image generation (Van Den Oord, Vinyals, et al., 2017). In terms of image generation, to generate high quality image, giving particular structure to latent space has been studied recently. Mathieu et al., 2019; Nagano et al., 2019; Ovinnikov, 2019 models latent space in hyperbolic space and shows good performance in learning hierarchical representation. Chen et al., 2020 gives Euclidean structure to latent space using the concept of Riemannian geometry. Meanwhile, the extension of variational auto-encoders to learning latent state-space representation for control problems have not been studied extensively.

On the other hand, trajectory optimization of nonlinear control system has been applied to various fields (Todorov & Li, 2005; H.-j. Zhang et al., 2012). In particular, iterative-Linear-Quadratic-Regulator (iLQR) has been a popular trajectory optimization method to deal with nonlinear control system. Its combination with RL has been studied recently (Zong et al., 2021). In a similar line there is technique called differential dynamic programming (DDP) which leverages second-order derivative of the dynamics.

In this project, we aim to solve an optimization problem in high dimensional space $x \in \mathbb{R}^N$, where the transition dynamics f is unknown and N is large. Since solving a an optimization problem in high-dimensional state-space has a huge computational cost, we try to find a low-dimensional latent space which well represents the high-dimensional space. We focus on what the structure of the latent space and the dependency on the parameters of optimal control method. In particular, we study the model that gives locally linear dynamics to the latent space, which is so-called embed to control (E2C) model (Watter et al., 2015). We implement the E2C model in Watter et al., 2015 and test it in the inverted pendulum environment. The aim of this project can be summarized as follows:

1) We implement the algorithm called E2C that gives locally linear structure to latent-space.

2) We analyze the limitation of locally linear structure model and discuss future research direction.

2 Background

2.1 Embed to Control

Embed to control (E2C) (Watter et al., 2015) uses locally linear dynamics to represent the high-dimensional space in the latent space. Suppose we have observations $\{x_k\}_{k=0}^{\infty}$ from high-dimensional space which evolves along the following dynamics:

$$x_{k+1} = f(x_k, u_k). \tag{1}$$

$x_k \in \mathbb{R}^N$ is the state variable and $u_k \in \mathbb{R}^p$ is control input. $f : \mathbb{R}^N \times \mathbb{R}^p \to \mathbb{R}^N$ is an unknown transition dynamics. The E2C model aims to learn a low-dimensional representation of x_k, which is z_k whose dynamics can be expressed as following locally linear dynamics:

$$z_{k+1} = A_k z_k + B_k u_k.$$

$A_k \in \mathbb{R}^{d\times d}$ is the state-transition matrix and $B_k \in \mathbb{R}^{d\times p}$ is the control-input matrix. Given a tuple (z_k, u_k, z_{k+1}), the transition matrices A_k, B_k are learned by the transition network parameterized by a learnable parameter ψ. The latent variable z_k can be learned using the encoder-decoder structure of standard VAE. The component of E2C model consists of three parts:

1. Encoding network : The encoding network $h_\phi : \mathbb{R}^N \to \mathbb{R}^n$ takes the observation x_k as input and outputs μ_k, Σ_k which are mean and covariance of Gaussian normal distribution. The latent variable z_k is sampled from $\mathcal{N}(\mu_k, \Sigma_k)$.

2. Decoding network : The decoding network $h_\theta : \mathbb{R}^n \to \mathbb{R}^N$ outputs a probability value such that it can reconstruct x_k from z_k.

3. Transition network : The transition network $h_\psi : \mathbb{R}^n \to (\mathbb{R}^{n\times n}, \mathbb{R}^{n\times p})$ outputs matrices A_k and B_k given z_k.

The above networks can be learned by maximizing the following loss which is motivated from the evidence lower bound loss (ELBO) (Kingma, Welling, et al., 2019):

$$L := \mathbb{E}_{z_t \sim Q_\phi, \hat{z}_{t+1} \sim \hat{Q}_\psi} \left[-\log P_\theta(x_k \mid z_k) - \log P_\theta(x_{k+1} \mid \hat{z}_{k+1}) \right] + \mathrm{KL}(Q_\phi \mid\mid P(Z)),$$

KL stands for the Kullback-Leibler divergence and $P(Z)$ is the prior of latent variable, which is usually assumed to follow Gaussian normal distribution. $Q_\phi, \hat{Q}_\psi$ and P_θ all stands for parametric inference model, where the first two corresponds to Gaussian normal distribution and the last to Bernoulli distribution. Moreover it adds a so-called consistency loss, which leads to the following total loss function:

$$L + \mathrm{KL}\left(\hat{Q}_\psi(\hat{Z} \mid \mu_k, u_k) || | Q_\phi(Z \mid x_{k+1}) \right). \tag{2}$$

μ_k is the output of encoding network h_ϕ. The consistency loss helps to learn the transition dynamics. As noted in Karl et al., 2016, even though the above loss is not exactly the evidence lower bound of the marginal likelihood $p(\{x_k\}_{k=0}^L \mid \{u_k\}_{k=0}^L)$ where L is the size of data, it well captures the property of ELBO and succeeds in learning appropriate representation.

2.2 iLQR

For trajectory optimization of non-linear control system, there are several methods. In particular, iterative-Linear-Quadratic Regulator (iLQR) (Tassa et al., 2012) is a LQR method applied to nonlinear dynamics in (1). Given a horizon length T, iLQR tries to minimize the following cost-to-go function:

$$J_i(x_i, \{u_k\}_{k=i}^{T-1}) = \sum_{k=i}^{T-1} ||x_k - x^*||_Q^2 + ||u_k - u^*||_R^2.$$

Q and R are the weight matrices for the state cost and input cost. Moreover, we can define the value function V and action-value function Q at time step i as follows:

$$V(x, i) := \min_u J_i(x_0, \{u_k\}_{k=i}^{T-1}), \quad Q(x, u, i) = ||x - x^*||_Q^2 + ||u - u^*||_R^2 + V(f(x, u), i + 1).$$

Following the spirit of dynamic programming, we can see the following relation:

$$V(x, i) = \min_u Q(x, u, i)$$

Minimizing the perturbation $Q(x + \delta x, u + \delta u, i) - Q(x, u, i)$ with respect to δu, we have

$$\delta u^* = -Q_{uu}^{-1}(Q_u + Q_{ux}\delta x),$$

where the partial derivatives are defined as

$$
\begin{aligned}
Q_{uu} &= -R + \nabla_u f(x,u)^\top \nabla_f^2 V(f(x,u), i+1) \nabla_u f(x,u) \\
Q_u &= R(u - u^*) + \nabla_u f(x,u)^\top \nabla_f V(f(x,u), i+1) \\
Q_{ux} &= \nabla_u f(x,u)^\top \nabla_f^2 V(f(x,u), i+1) \nabla_x f(x,u).
\end{aligned}
$$

The perturbed control input $u + \delta u^*$ provides a lower cost-to-go than applying control input u. With the above set of derivatives, and defining $w := -Q_{uu}^{-1} Q_u$ and $W := -Q_{uu}^{-1} Q_{ux}$, we can compute a new nominal trajectory :

$$\hat{u}(k) = u(k) + w(k) + W(k)(\hat{x}(k) - x(k)) \tag{3}$$

$$\hat{x}(k+1) = f(\hat{x}(k), \hat{u}(k)). \tag{4}$$

We can repeat this process until δu^* becomes significantly small, which implies that it has converged to a local optima. In details, iLQR can be implemented in three steps :

1) Compute the derivative of cost function and dynamics, f.

2) Backward pass : Compute Q_{uu}, Q_u, Q_{ux} starting from time step T to 1.

3) Forward pass : Compute (3) and (4).

Repeating the above process until convergence δu^*, the over all pseudo-code is written in Algorithm 1.

Algorithm 1 iLQR

```
Require: $x_0$, input sequence $\{u_j^*\}_{j=0}^T$,
  while until converge do
    Compute the derivatives of cost function and dynamics
    Compute $Q_{uu}, Q_u, Q_{ux}$, which can be done in backward manner, i.e., from $T$ to 1
    Compute the new nominal trajectory in (3) and (4).
  end while
```

2.3 Overall scheme

The overall scheme can be described as follows. The iLQR method is implemented in receding horizon control (RHC) law manner. At time step k, computing the optimal control input using iLQR algorithm on latent space starting from concatenated input of the previous optimal control $\{u_j^*(k-1)\}_{j=k-1}^{k+T-1}$ and a random input $\tilde{u}$, . The first argument of the output of iLQR algorithm is applied as the input to the system, and we observe the next state x_{k+1}. Here, we provide a pseudo-code to apply E2C model in RHC manner.

Algorithm 2 Optimal control using E2C model

```
Require: $x_0$, random input sequence $\{u_j^*\}_{j=0}^T$,
  $k = 0$;
  while until converge do
    Compute optimal control sequence $\{u_j^*(k)\}_{j=k}^{k+T}$ using iLQR
    Apply the first input of $\{u_j^*(k)\}_{j=k}^{k+T}$ and observe $x_{k+1} = f(x_k, u_k^*(k))$
    $k \leftarrow k + 1$
  end while
```

3 Experiments and discussion

In this section, we verify the performance of E2C model with iLQR method. Even though the inverted pendulum dynamics is a relatively simple environment and its dynamics is well-known, it is important to check its performance before applying to a more difficult problem. Moreover, testing in a simple environment offers a glimpse of the behavior of the model and chances to understand the limitations.

The dynamics of inverted pendulum model can be described by two states, which are angle and angular velocity. θ and $\dot{\theta}$ stands for angle and angular velocity of the pendulum respectively. The goal is to swing up the pendulum to the upright state, which is the equilibrium point, $(\theta, \dot{\theta}) = (0, 0)$. The pendulum will keep stay at the equilibrium point unless small perturbation is applied. Since it is an unstable equilibrium point, we need to apply appropriate input to drive the pendulum to the equilibrium point. We can apply the input torque constrained between $[-2, 2]$ to the pendulum to overcome the effect of gravity and velocity to pause the pendulum at the upright state. To reformulate it as an optimal control problem, we first check that the dynamics of inverted pendulum is nonlinear which can be expressed as follows:

$$\dot{\theta}_{k+1} = \dot{\theta}_k + ((3 \cdot 10)/2 \cdot \sin(\theta_k) + 3.0 \cdot u_k) \cdot 0.05$$
$$\theta_{k+1} = \theta_k + 0.05 \cdot (\dot{\theta}_k + (3 \cdot 10/2 * \sin(\theta_k) + 3.0 \cdot u_k) \cdot 0.05).$$

As one can notice, the dynamics of inverted pendulum is nonlinear due to sin function. Hence, we can not directly apply the LQR methods as in linear state space model. When the system is nonlinear, iLQR introduced in Section 2.2 is an effective solution for the optimal control problem because it requires only first-order derivative of the dynamics while DDP requires second order information of the dynamics. Even though, we can calculate second order information of the above dynamics easily, we stick to first order information because computational cost of second order derivative is expensive for more complex environments.

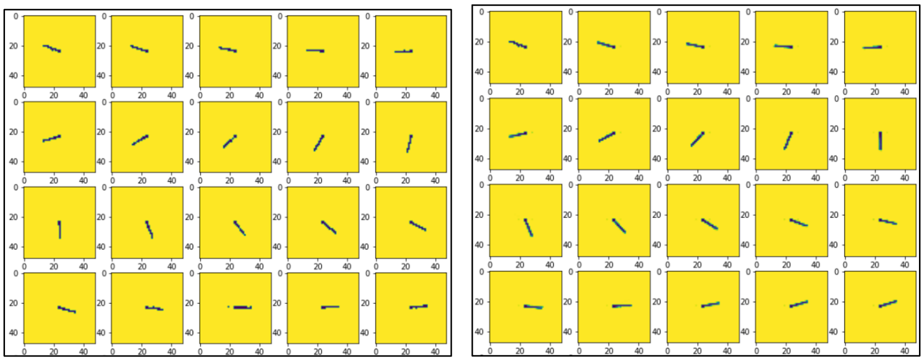
3.1 Results and discussion about E2C

This section illustrates the training process of E2C and how we evaluated E2C model. To train the E2C model, we used two stacked image of inverted pendulum converted to black and white image. Stacking images reflects the Markovian property of the inverted pendulum model. Hence, the input vector x_k has dimension of $\mathbb{R}^{48 \cdot 48 \cdot 2}$ where single image is of size 48 by 48. Using the encoder network h_ϕ in the E2C model, the high-dimensional observation x_k is embedded into low-dimensional space $z_k \in \mathbb{R}^3$. We generated 15,000 images randomly initializing the initial state, θ and $\dot{\theta}$ and applying random input using OpenAI gym environment.

Compared to the original work, we added batch normalization (Ioffe & Szegedy, 2015), which slightly improved the quality of the predicted images. The training process, which we continued until the loss term in (2) converged, took about 3000 episodes with batch size of 128.

Even though we can numerically check whether the model has been trained in optimization perspective, it is difficult to check whether the model learned an appropriate latent space. To evaluate such qualitative performance, we checked the decoded images of the predicted sequences in latent space and compared it with the ground-truth image. Starting from a initial point, and applying the same random input sequence $\{u_k\}_{k=0}^{20}$ in the original space and latent space, the predicted images in Figure 2 shows similar results. E2C well predicts for the horizon $T = 10$ but it does not predict well for a longer time steps, since locally linear dynamics have innate limitation to reflect to accurate transition model.

However, for optimal control problem, images being similar is not enough compared to image generation task. Furthermore, considering the ground-truth image and predicted images in Figure 1 look similar, the actual trajectory in latent space and predicted trajectory in latent space differs as can be seen in Figure 2. The inaccurate prediction of the trajectory leads to degraded performance of iLQR method.



(a) Ground-truth images of $\{x_k\}_{k=0}^{20}, \{u_k\}_{k=0}^{20}$

(b) Decoded images of predicted trajectory for 20 time steps

Figure 1: Result of applying same input to original space and latent space for 20 time steps. The time step starts from the top-left and ends at right-end.

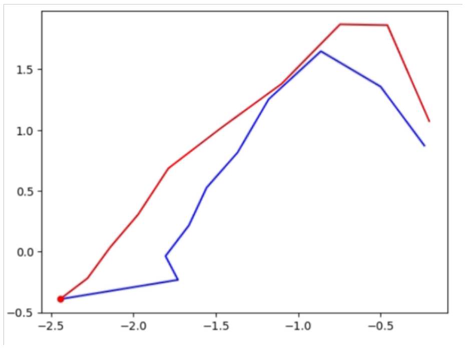


Figure 2: Trajectory of $\{x_k\}_{k=0}^{20}(red)$, $\{z_k\}_{k=0}^{20}$ (blue) in $\mathbb{R}^2$, which is the projected space.

3.2 Results and discussion about iLQR

In this section, the details on the implementation of iLQR are described. Instead of repeating the backward and forward process in Algorithm 1 until convergence, we iterated the process for 20 times considering the computational cost. Moreover, since there is a input constraint between $[-2, 2]$ we used a tanh function to convert it into an unconstrained problem.

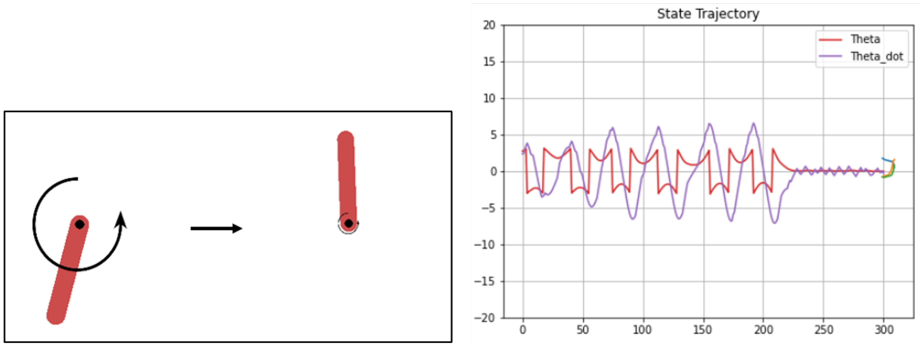
As in Tables 1 and 2, the performance of iLQR is highly sensitive to the horizon length and weight matrices. From Table 1, we can see that if the horizon length is too short ($T = 5$) or if the horizon length is too long ($T = 20$) , then the iLQR fails to position the pendulum at the upright position. We need to select an appropriate horizon length that is not too short to achieve the goal or not too long such that predicted trajectory is not accurate. Starting from $(5/6\pi, 2)$ the iLQR succeeds in uprighting pendulum as can be seen in Figure 3. After swinging several times, the pendulum achieves enough velocity to go to upright position and appropriate torque is applied to force the pendulum to stay at the equilibrium state.

Horizon length (T)	5	10	20
Success/Failure	Fail	Success	Fail

Table 1: Result of optimal control depending on the horizon length T

Q	I	$10I$	$100I$
Success/Failure	Fail	Success	Fail

Table 2: Result of optimal control depending on the cost weight matrix Q



(a) Image of starting point and goal image

(b) $\theta, \dot{\theta}$ corresponds to angle and angular velocity respectively. (0,0) is the goal state.

Figure 3: Applying iLQR in latent space

4 Further improvements

In this section, we briefly discuss how to improve the E2C model and suggest future research directions.

4.1 Switched system modeling

A switched linear system is a dynamical system that evolves with a finite set of matrices $\{A_i\}_{k=1}^{M}$. Switched system can learn the dynamic more accurately than simple locally linear dynamics. However, optimal control for switched system is known to be NP-hard (W. Zhang et al., 2009). Using switched linear system for two time steps, the latent-space can be represented as:

$$z_{t+j} = A_{\sigma(t+j)} z_{t+j} + B_{\sigma(t+j)} u_{t+j}, \quad j = 0, 1.$$

σ is the switching mode. Even though learning switched system dynamics is non-trivial, it well represents the dynamics than locally linear structure. Compromising between the difficulty of solving optimal control problem and expressiveness of the transition dynamics would be interesting future research direction.

4.2 Imposing Euclidean structure to latent space

Even though we embedded the observation x_k in high-dimensional space into $\mathbb{R}^3$ space, there is no theoretical guarantee that the latent space has Euclidean structure. Moreover, it is unclear whether defining the objective function of iLQR using Euclidean norm is appropriate in latent space. Chen et al., 2020 proposed a way to impose such Euclidean structure but requires multiplication of Jacobian matrix which is computationally expensive. Furthermore, for typical class of control problem such as solving an optimal control problem with hierarchical structure, using the Poincare VAE (Mathieu et al., 2019; Ovinnikov, 2019) would be appropriate than locally linear structure.

4.3 Unifying planning and representation learning

Learning visual representation for optimal control problem may be a redundant process. Srinivas et al., 2018 proposed a way to learn effective representation for planning using gradient descent with respect to the action sequences. However it requires expert data and learning effective representation for control problem in an unsupervised manner still remains an open problem.

5 Conclusion

In this project, we have implemented E2C model (Watter et al., 2015) and tested in inverted pendulum environment. We mainly focused on the implementation of the algorithm that gives locally linear structure to latent space. Overall, the optimal control problem becomes highly sensitive to hyperparmeter selections. Despite the extensive study on generative models, its extension to control problems have not been fully studied and we leave it as future work to study appropriate structure for the latent representation for optimal control problem.

References

Chen, N., Klushyn, A., Ferroni, F., Bayer, J., & Van Der Smagt, P. (2020). Learning flat latent manifolds with vaes. *arXiv preprint arXiv:2002.04881*.

Hansen, N., Wang, X., & Su, H. (2022). Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, 448–456.

Janner, M., Li, Q., & Levine, S. (2021). Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, *34*, 1273–1286.

Jiang, Z., Zhang, T., Janner, M., Li, Y., Rocktäschel, T., Grefenstette, E., & Tian, Y. (2022). Efficient planning in a compact latent action space. *arXiv preprint arXiv:2208.10291*.

Karl, M., Soelch, M., Bayer, J., & Van der Smagt, P. (2016). Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*.

Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307–392.

Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R., & Teh, Y. W. (2019). Continuous hierarchical representations with poincaré variational autoencoders. *Advances in neural information processing systems*, *32*.

Nagano, Y., Yamaguchi, S., Fujita, Y., & Koyama, M. (2019). A differentiable gaussian-like distribution on hyperbolic space for gradient-based learning.

Ovinnikov, I. (2019). Poincar\'e wasserstein autoencoder. *arXiv preprint arXiv:1901.01427*.

Srinivas, A., Jabri, A., Abbeel, P., Levine, S., & Finn, C. (2018). Universal planning networks: Learning generalizable representations for visuomotor control. *International Conference on Machine Learning*, 4732–4741.

Tassa, Y., Erez, T., & Todorov, E. (2012). Synthesis and stabilization of complex behaviors through online trajectory optimization. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4906–4913.

Todorov, E., & Li, W. (2005). A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. *Proceedings of the 2005, American Control Conference, 2005.*, 300–306.

Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, *30*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Watter, M., Springenberg, J., Boedecker, J., & Riedmiller, M. (2015). Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, *28*.

Zhang, H.-j., Gong, J.-w., Jiang, Y., Xiong, G.-m., & Chen, H.-y. (2012). An iterative linear quadratic regulator based trajectory tracking controller for wheeled mobile robot. *Journal of Zhejiang University SCIENCE C*, *13*(8), 593–600.

Zhang, W., Hu, J., & Abate, A. (2009). On the value functions of the discrete-time switched lqr problem. *IEEE Transactions on Automatic Control*, *54*(11), 2669–2674.

Zong, T., Sun, L., & Liu, Y. (2021). Reinforced ilqr: A sample-efficient robot locomotion learning. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 5906–5913.