Actor-Critic Learning for Optimal Control Problems

Gabriele Stulzer

January 2025

1 Introduction

This report presents the implementation and results of an actor-critic learning framework applied to solving optimal control problems (OCPs). The goal was to learn a value function (critic) and a policy (actor) for a simple integrator system, and evaluate the performance using various hyperparameters.

2 Problem Formulation

The OCP is defined as:

$$\min_{u_t} \quad \sum_{t=0}^{T-1} \left[\frac{1}{2} u_t^2 + (x_t - 1.9)(x_t - 1.0)(x_t - 0.6)(x_t + 0.5)(x_t + 1.2)(x_t + 2.1) \right], \tag{1}$$

$$\text{subject to} \quad x_{t+1} = x_t + \Delta t \cdot u_t, \tag{2}$$

where x_t is the state, u_t is the control input, and $\Delta t = 0.1$ is the time step.

3 Methodology

3.1 Neural Network Architectures

The critic and actor networks are both feedforward neural networks with two hidden layers of 64 neurons each and ReLU activations. The output layer of the critic is a single linear neuron predicting the value function. The actor network's output is a single neuron with a hyperbolic tangent activation, representing the control signal.

3.2 Training Procedure

Critic Training: The critic network was trained using mean squared error loss to approximate the cost-to-go from given states. Training data consisted of randomly sampled states and control inputs.

Actor Training: The actor network was trained by minimizing the action-value function, computed as the sum of the running cost and the value of the next state predicted by the critic.

4 Results and Analysis

4.1 Hyperparameter Configurations

Table 1 summarizes the key hyperparameters used.

4.2 Value and Policy Functions

Figure **??** shows the learned value function and policy for the system. The critic successfully approximates the value function, while the actor learns a policy that minimizes the action-value function.

Parameter	Value
Learning Rate	0.001
Batch Size	32
Critic Epochs	50
Actor Epochs	50

Table 1: Hyperparameter settings for training.

5 Discussion

The results demonstrate the effectiveness of the actor-critic framework in solving simple OCPs. The learned value function aligns well with the theoretical cost, and the policy shows a smooth control signal minimizing the given cost.

5.1 Suggested Improvements

To improve performance, several tests and enhancements can be performed:

- **Network Architecture:**
  - Use deeper networks with more hidden layers (e.g., 3-5 layers) to capture complex patterns.
  - Experiment with alternative activation functions such as Leaky ReLU, ELU, or Swish to enhance gradient flow.
  - Test convolutional layers if spatial correlations exist in state-action mappings.
- **Hyperparameters:**
  - Adjust the learning rate (e.g., test 0.0001, 0.0005) for better convergence.
  - Try larger batch sizes (e.g., 64, 128) to stabilize gradient updates.
  - Increase training epochs to allow better fitting of the critic and actor networks.
  - Use learning rate schedulers to adapt the rate during training.
- **Optimization Techniques:**
  - Incorporate regularization techniques such as L2 penalties or dropout to prevent overfitting.
  - Use advanced optimizers like AdamW or RMSProp for better weight updates.
  - Implement target networks for the critic to stabilize training.
- **Expanded Dynamics:**
  - Apply the framework to more complex dynamics, such as a double integrator or pendulum systems.
  - Include stochastic elements in dynamics for robustness testing.

6 Conclusion

This project implemented an actor-critic learning approach for OCPs. The framework achieved good performance in approximating the value function and policy, demonstrating its potential for more complex control problems. Future work can focus on testing advanced architectures, optimizing hyperparameters, and applying the method to higher-dimensional systems.