Bridging Physics-Informed Neural Networks with Reinforcement Learning: Hamilton-Jacobi-Bellman Proximal Policy Optimization (HJBPPO)

Amartya Mukherjee, Jun Liu

Department of Applied Mathematics, University of Waterloo

New Frontiers in Learning, Control, and Dynamical Systems

Abstract

This paper introduces the Hamilton-Jacobi-Bellman Proximal Policy Optimization (HJBPPO) algorithm into reinforcement learning. The Hamilton-Jacobi-Bellman (HJB) equation is used in control theory to evaluate the optimality of the value function.

Our work combines the HJB equation with reinforcement learning in continuous state and action spaces to improve the training of the value network. We treat the value network as a Physics-Informed Neural Network (PINN) to solve for the HJB equation by computing its derivatives with respect to its inputs exactly. The Proximal Policy Optimization (PPO)-Clipped algorithm is improvised with this implementation as it uses a value network to compute the objective function for its policy network.

The HJBPPO algorithm shows an improved performance compared to PPO on the MuJoCo environments.

The HJB equation

Consider a controlled dynamical system modeled by the following equation:

$$\dot{x} = f(x,u), \qquad x(t_0) = x_0$$

In control theory, the optimal value function $V^*(x)$ is useful towards finding a solution to control problems:

$$V^*(t) = \sup_u \int_{t_0}^{\infty} \gamma^t R(x(\tau; t_0, x_0, u(\cdot)), u(\tau)) d\tau$$

Where $R(x,\sigma)$ is the reward function and γ is the discount factor.

Theorem 2.1. A function $V(x)$ is the optimal value function if and only if:

1. $V \in C^1(\mathbb{R}^n)$ and V satisfies the Hamilton-Jacobi-Bellman (HJB) Equation
$$V(x) \ln\gamma + sup_{u\in U} \{R(x,u) + \nabla_x V^T(x) f(x,\sigma)\} = 0$$

2. For all $x \in \mathbb{R}^n$, there exists a controller $u^*(\cdot)$ such that:
$$R(x,u^*(x)) + \nabla_x V^T(x) f(x,u^*(x)) = \sup_{\hat{u}} \{L(x,\hat{u}(x)) + \nabla_x V^T(x) f(x,\hat{u}(x))\}$$

New loss functions for the value network

Derived from the HJB equation:

$$\widehat{MSE}_f = \frac{1}{T}\sum_{t=0}^{T-1}\left(V(x_t)\ln\gamma + R(x_t,a_t) + \nabla_x V(x_t)^T f(x_t,a_t)\right)^2$$

We compute $\nabla_x V(x_t)$using auto-differentiation. Approximate $f(x_t,a_t)$ using finite differences.

$$MSE_f = \frac{1}{T}\sum_{t=0}^{T-1}\left(V(x_t)\ln\gamma + R(x_t,a_t) + \nabla_x V(x_t)^T\left(\frac{x_{t+1}-x_t}{\Delta t}\right)\right)^2$$

Loss function:

$$J(\phi) = 0.5 MSE_u + \lambda_{HJB} MSE_f$$

Where MSE_u is derived from the discrete-time Bellman equation

$$MSE_u = \frac{1}{T}\sum_{t=0}^{T-1}\left(V(x_t) - \left(R(x_t,a_t) + \gamma V(x_{t+1})\right)\right)^2$$

Hamilton Jacobi Bellman Proximal Policy Optimization (HJBPPO)

1. Initiate policy network parameter θ and value network parameter ϕ
2. Run action selection as given earlier for T timesteps and observe samples $\{(s_t, a_t, R_t, s_{\{t+1\}})\}_{t=1}^T$
3. Compute the advantage $A_t = \delta t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t-1}\delta_{T-1}$ where $\delta = R_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$, γ: Discount factor (≈ 0.99) and λ: Smoothing factor (≈ 0.95)
4. Compute $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$
5. Compute the objective function of the policy network: $L(\theta) = \frac{1}{T}\sum_{t=0}^{T-1} \min[r_t(\theta)A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t]$ where ε: clipping parameter (≈ 0.2)
6. Update $\theta \leftarrow \theta - \alpha_1 \nabla_\phi L(\theta)$
7. Compute the value network loss as: $J(\phi) = 0.5MSE_u + \lambda_{HJB} MSE_f$
8. Update $\phi \leftarrow \phi - \alpha_2 \nabla_\phi J(\phi)$
9. Run steps 2-5 for multiple iterations

Results