MLDS 490 Lab 4

Fall 2023 Shuyang Wang

Agenda

- Assignment 1 Part 1 is due today at 10pm:

  Checklist:

  1. Submit: code, report (plots & instruction to run the code).
  2. Upload to your private GitHub repo, add me as collaborator, due at 10pm.
- Assignment 1 Part 2 is due on Oct 24 (Tuesday) at 10pm:
  - Grading is based on the performance
- Baselines

Baselines

Recall that REINFORCE computes: $\nabla_\theta J(\theta) = \mathbb{E}\Big[\sum_{t=1}^{T} [\nabla \log \pi(a_t|s_t) \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}]\Big]$

At each step, a discounted future return is used: $G_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$

Episode rewards can be of high variance due to the sampling noise.

Use **a baseline** to reduce the variance:

$$\nabla_\theta J(\theta) = \mathbb{E}\Big[\sum_{t} [\nabla \log \pi(a_t|s_t) G_t]\Big] \qquad \text{REINFORCE}$$

$$\nabla_\theta J(\theta) = \mathbb{E}\Big[\sum_{t} [\nabla \log \pi(a_t|s_t)(G_t - b(s_t))]\Big] \qquad \text{REINFORCE with baselines}$$

Baselines

Candidates

- Constant baseline
  - Average of episode reward
  - Shown to be unbiased
- **Non-constant baseline** - should be used for assignment 1 part 2
  - Should be based on the current state
  - (Policy) value function is a good candidate

$$b(s_t) = V^{\pi_\theta}(s_t) = \mathbb{E}_{\pi_\theta}\left[\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}(a_{t'}, s_{t'}) \middle| s_t\right]$$

Policy Gradient with Baselines

$$\nabla_\theta J(\theta) = \mathbb{E}\Big[\sum_t [\nabla \log \pi(a_t|s_t) G_t]\Big] \qquad \text{REINFORCE}$$

$$\nabla_\theta J(\theta) = \mathbb{E}\Big[\sum_t [\nabla \log \pi(a_t|s_t)(G_t - V(s_t))]\Big] \qquad \text{REINFORCE with value function baseline}$$

$$= \mathbb{E}\Big[\sum_{t=1}^{T} \nabla \log \pi(a_t|s_t) A_t\Big]$$

$$A_t = G_t - V(s_t) = (\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}) - (\mathbb{E}_{\pi_\theta} \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'})$$

approximates the **advantage**

Policy network: $NN_\theta(s_t) = [p_0, p_1]$ or logits

Update the policy network: $$\nabla_\theta J(\theta) = \frac{1}{N} \sum_{NEpisodes} \Big[\sum_{t=1}^{T} \nabla_\theta \log \pi(a_t|s_t)(G_t - V_w^{\pi_\theta}(s_t)) \Big]$$

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\theta_k)$$

Value network, outputs 1 value: $NN_w(s_t) = v \in \mathbb{R}$

Update the value network by Mean Squared Error:

NN outputs: $[NN_w(s_1), ..., NN_w(s_T)]$

Targets: the discounted future returns $[G_1, ..., G_T]$

$$w_{k+1} = w_k - \eta \nabla_w MSE\big(NN_{w_k}(s_t), G_t\big)$$

Policy Gradient with Baseline

Algorithm 1 Policy Gradient with Baseline

Require: Policy network π_θ, value network V_w, learning rates $\alpha_\theta > 0, \alpha_w > 0$
  Initialize policy and value networks
  while not converged **do**
    Sample a batch of episodes, collect states, actions, rewards histories.
    for each episode and t = 1, .., T **do** (use vectorization in implementation instead of for-loop)
      Compute discounted future returns $G_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$
      Forward pass of policy network to compute log probabilities $\log \pi_\theta(a_t|s_t)$ of selected actions
      Forward pass of value network to compute state values $V_t = V_w(s_t)$
      Compute advantage vector $A_t = G_t - V_t$ *Comment: normalization or rewards or advantages may still apply*
    end for
    Update θ using loss $-\frac{1}{N} \sum_{NEpisodes} \left[\sum_{t=1}^{T} \log \pi_\theta(a_t|s_t)\ A_t \right]$ with lr α_θ
    Update w using mean squared error loss $MSE(V_t, G_t)$ with lr α_w
  end while