REINFORCE Algorithm Tutorial

September 21, 2025

1 Introduction

REINFORCE, also known as Monte Carlo policy gradient, updates policy parameters using complete trajectory returns. It provides an unbiased estimator of the policy gradient by weighting log-probability gradients with observed returns, making it the foundational algorithm for policy gradient methods.

2 Theory and Formulas

2.1 Monte Carlo Policy Gradient

Given trajectories τ sampled from policy π_θ, the REINFORCE gradient estimator is

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\Big[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, G_t\Big], \tag{1}$$

where $G_t = \sum_{k=t}^{T-1} \gamma^{k-t} r_{k+1}$ is the return from time t.

2.2 Baselines and Variance Reduction

Adding a baseline $b(s_t)$ retains unbiasedness while reducing variance:

$$\nabla_\theta J(\theta) = \mathbb{E}\Big[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\,(G_t - b(s_t))\Big]. \tag{2}$$

Typical choices include constant baselines, value-function estimates, or EMAs of returns.

2.3 Algorithm Steps

1. Collect trajectories by rolling out π_θ.
2. Compute returns G_t for each time step.
3. Update parameters with gradient ascent: $\theta \leftarrow \theta + \alpha \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)(G_t - b(s_t))$.
4. Optionally update the baseline estimate.

Because REINFORCE relies on full trajectory returns, it exhibits higher variance than actor-critic methods but is simple and unbiased.

3 Applications and Tips

- **Episodic tasks**: environments with moderate episode length and sparse rewards.
- **Curriculum learning**: warm-start more advanced actor-critic algorithms.
- **Discrete policies**: categorical action spaces or parameterized bandits.
- **Best practices**: normalize returns within batches, use baselines, tune learning rate carefully, and consider reward-to-go to reduce variance.

4 Python Practice

The script `gen_reinforce_figures.py` trains a softmax policy in a grid-world with terminal rewards using REINFORCE and a moving-average baseline. It records episodic returns and state visitation frequencies under the learned policy.

Listing 1: Excerpt from gen$_{r}einforce_{f}igures.py$

```
returns = compute_returns(rewards, gamma)
for (state, action), G_t in zip(trajectory, returns):
    probs = softmax(theta[state])
    grad = -probs
    grad[action] += 1.0
    baseline[state] += baseline_lr * (G_t - baseline[state])
    theta[state] += alpha * grad * (G_t - baseline[state])
```

5 Result

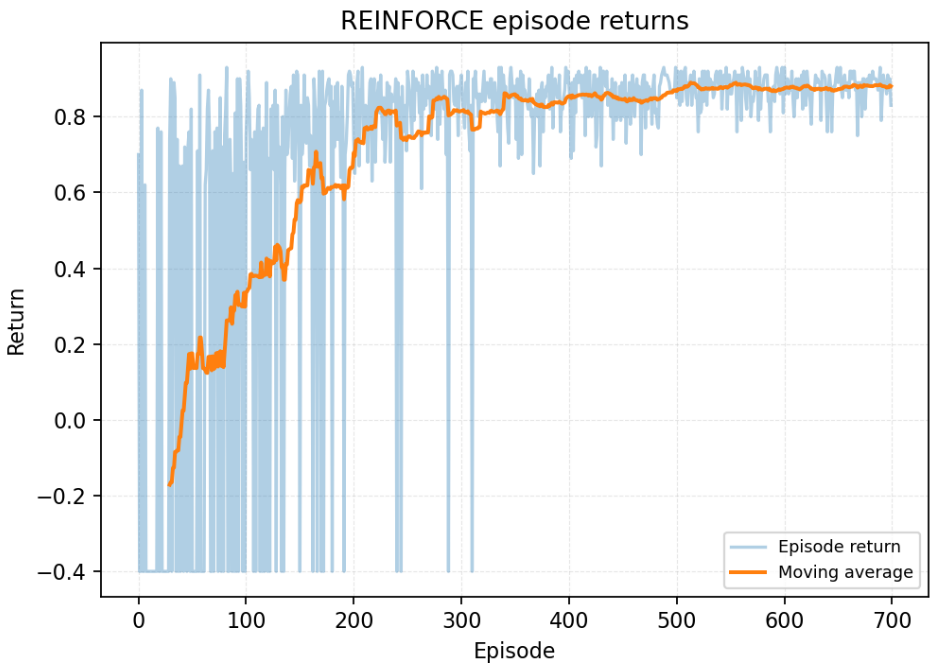

Figure 1: REINFORCE episode returns with moving average smoothing

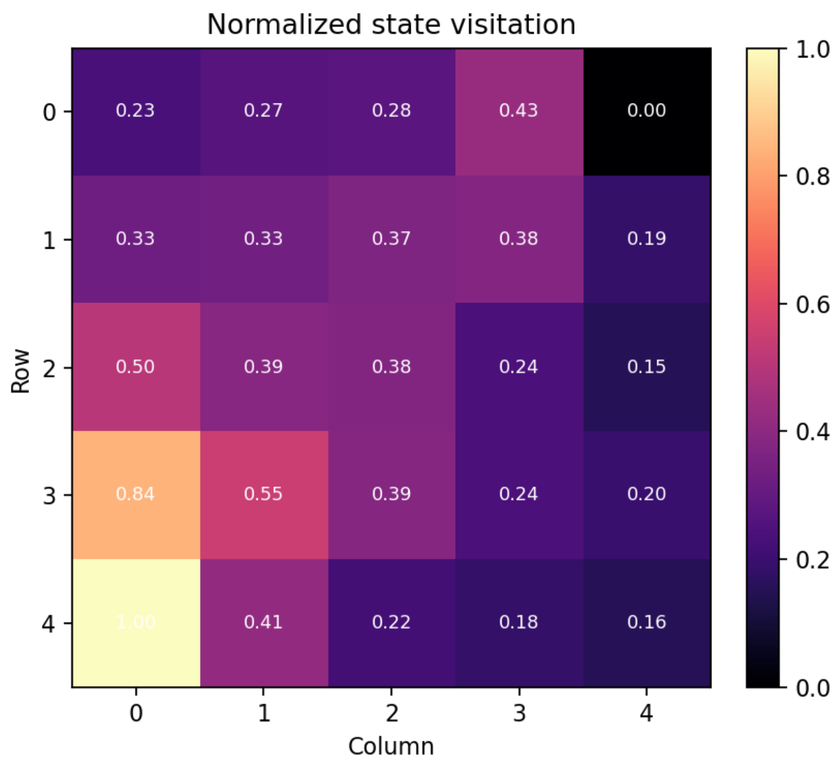

Figure 2: State visitation heatmap after training, highlighting preferred trajectories

6 Summary

REINFORCE offers a simple, unbiased policy gradient estimator but requires careful variance reduction and learning-rate tuning. Baselines, reward normalization, and batch averaging help stabilize learning. The grid-world example shows returns improving as the policy concentrates on efficient paths.