Policy Gradient with Second Order Momentum

Tianyu Sun

Policy Gradient

Second Order Policy Gradient Theorem

Let the reward function be defined as

$$J(\theta) = \sum_{s \in \mathcal{S}} d^{\pi}(s) V^{\pi}(s) = \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} \pi_{\theta}(a|s) Q^{\pi}(s, a) \tag{1}$$

The the policy gradient theorem states that

$$\nabla J(\theta) \propto \mathbb{E}_{s \sim d^{\pi}, a \sim \pi(s)} \left[Q^{\pi}(s, a) \nabla_{\theta} \ln \pi_{\theta}(a|s) \right] \tag{2}$$

And the second order momentum of the policy can be calculated as

$$\nabla^2 J(\theta) \propto \mathbb{E}_{\tau \sim p(\tau; \pi_{\theta})} [\nabla_{\theta} \ln(p(\tau; \pi_{\theta})) \nabla_{\theta} \Psi(\theta) + \nabla_{\theta}^2 \Psi(\theta)] \tag{3}$$

where $\Phi(\theta; \tau) = \sum_{h=0}^{H} \ln \pi_{\theta}(a_h|s_h) \sum_{t=h}^{H} \gamma^t r(s_t, a_t)$.

Policy Gradient

Second Order Policy Gradient Theorem

Notice that

$$
\begin{aligned}
p(\tau;\pi_\theta) &= \rho(s_0)\Pi_{h=1}^{H}\mathcal{P}(s_{h+1}|s_h, a_h)\pi_\theta(a_h|s_h) \\
\nabla_\theta \ln(p(\tau;\pi_\theta)) &= \nabla_\theta \left[\ln\rho(s_0) + \sum_{h=1}^{H}\ln(\mathcal{P}(s_{h+1}|s_h, a_h)) + \sum_{h=1}^{H}\ln(\pi_\theta(a_h|s_h))\right] \\
&= \nabla_\theta \left[\sum_{h=1}^{H}\ln(\pi_\theta(a_h|s_h))\right]
\end{aligned}
\tag{4}
$$

Second Order Momentum Estimation with Runge Kutta Method (Heun's Method)

- We seek a model free method
- Heun's Method (RK2):

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ \tilde{y}_{i+1} &= y_i + \frac{1}{2} f(t_i, y_i) \\ y_{i+1} &= y_i + \frac{1}{2}(f(t_i, y_i) + f(t_{i+1}, \tilde{y}_{i+1})) \end{aligned} \tag{5}$$

Policy Gradient with Second Order Momentum

for *i* **from** *1* **to** *max_episodes* **do**

Generate a trajectory τ;

Accumulate total reward, rewards, observations, and actions;

if $i \mod (max\_episodes//20) == 0$ **then**

Evaluate π_θ

end

Compute $\nabla J(\theta^t)$ using (2) and advantage function;

Computer $\nabla^2 J(\theta^t)$ using (3) and advantage function;

$\theta^{t+1} = \theta^t + \eta \cdot (\alpha \nabla J(\theta^t) + (1 - \alpha)\nabla^2 J(\theta^t))$

end

return *episode_rewards, evaluation*

Algorithm 1: Policy Gradient with Second Order Momentum

Policy Gradient with Second Order Momentum

for *i* **from** *1* **to** *max_episodes* **do**

  Generate a trajectory τ;

  Accumulate total reward, rewards, observations, and actions;

  if $i \mod (max\_episodes//20) == 0$ **then**

    Evaluate π_θ

  end

  Compute $\nabla J(\theta^t)$ using (2) and advantage function;

  $\tilde{\theta}_t = \theta^t + \eta \cdot \nabla J(\theta^t)$;

  Computer $\nabla \tilde{J}(\tilde{\theta}_t)$ using (2) and advantage function;

  $\theta^{t+1} = \theta^t + \eta \cdot (\alpha \nabla J(\theta^t) + (1-\alpha)\nabla \tilde{J}(\tilde{\theta}_t))$

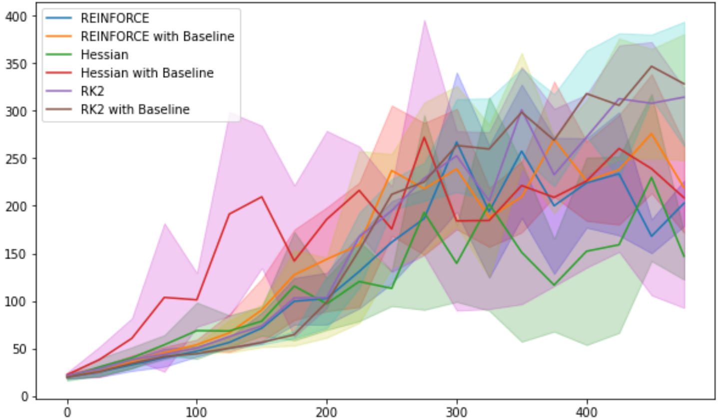
end

return *episode_rewards, evaluation*

Algorithm 2: Policy Gradient with Second Order Momentum (Heun's method)

Preliminary Result

Settings:
Parameterized policy: sigmoid
Learning rate: 0.002
Env: Cartpole



Future steps

To do:

- Try on another environment like Lunar Lander

Maybe to do:

- Use a nonlinear parameterized policy+NN?
- Use a TD style using A2C?
- Try on Mujoco

Problems:

- Not working well with large learning rate (currently $\eta = 0.002$)

Possible fix:

- Normalize gradient after each step
- Only include momentum at certain steps

References

- Shen, Zebang, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi. "Hessian Aided Policy Gradient." In *International Conference on Machine Learning*, pp. 5729-5738, 2019.
- Saber Salehkaleybar, Sadegh Khorasani, Negar Kiyavash, Niao He, and Patrick Thiran. "Momentum-Based Policy Gradient with Second-Order Information." arXiv preprint arXiv:2205.08253, 2023.
- Tran, Hoang, and Ashok Cutkosky. "Better SGD using Second-Order Momentum." In *Advances in Neural Information Processing Systems*, vol. 35, pp. 3530-3541, 2022.