REINFORCEMENT LEARNING CONTROL OF A CART-POLE

Darya Amin-Shahidi (daryaa@stanford.edu)

Overview

Demo URL: *https://tinyurl.com/skojteg*

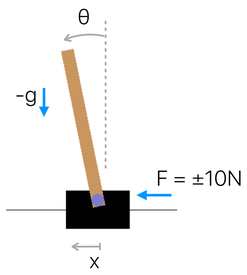
Problem

Balance an inherently unstable, non-linear, and dynamic cart-pole system using RL with no prior knowledge of the model for 200 time steps

RL Methods

Three variations of the Q-learning are tried

- **Vanilla Q-Learning** model-free, simple, and generic method for estimating optimum policy
- **Modified Reward** favoring desired states expedites learning and improves convergence
- **Eligibility Traces** crediting states leading up to current state did not show an improvement



cart-pole diagram

Problem Details & Oracle

Environment

OpenAI Gym's cart-pole-v0 library [1] used to simulate physics:

- Physics include dynamics, angle non-linearities, and friction [2]
- Time step of 0.02s with Euler1
- Actions are limited to ±10N exerted on the cart
- Observations are [x, dx/dt, θ, dθ/dt]
- Basic reward of 1 is granted per time step

Problem is solved if for time steps i ≤ 200, | θ | < 0.2 rad, | x | < 2.4

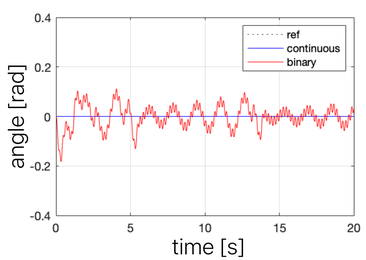
Oracle

Model-based controller tuned to the physics

- Works out of the box
- Can balance indefinitely
- Model based! (breaks if inaccurate or changing)

Goal

Use model-free RL with no knowledge of the complex physics to solve the problem



Oracle balancing time-plot for continuous and binary force actions

$$\ddot{\theta} = \frac{g\sin\theta + \cos\theta\left[\frac{-F - mL\dot{\theta}^2\sin\theta + \mu_c\, sgn(\dot{x})}{m_c+m}\right] - \frac{\mu_p\dot{\theta}}{mL}}{L\left[\frac{4}{3} - \frac{m\cos^2\theta}{m_c+m}\right]} \qquad \ddot{x} = \frac{F + mL[\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta] - \mu_c sgn(\dot{x})}{m_c+m}$$

Cart-pole equations of motion

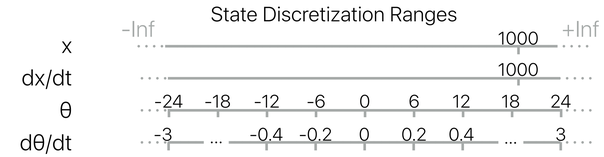
Vanilla Q-Learning

Algorithm Details

- Discretize state space based on ranges/buckets
- Update: $Q(s,a) \leftarrow Q(s,a) - \alpha\,(\,Q(s,a) - (R + \gamma Q(s',a'_{opt})\,)$
- Use ε-Greedy policy

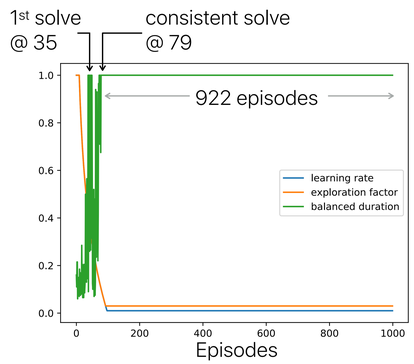
Observations

- Use minimum required state space to expedite learning (shown below)
- Skip x and dx/dt from state definition (assign to a single bucket)
- Use discount factor $\gamma = 0.9$ to stabilize learning and convergence
- Start with $\varepsilon_0 = \alpha_0 = 1$ and reduce with episodes to ε_{min} and α_{min} such that $\varepsilon_{min} > \alpha_{min}$ (avoid learning without exploring)
- Adjust decay period for ε and α to match anticipated learning period



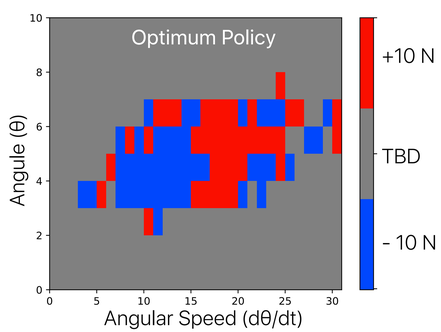
Learning Results

- 1st solve @ 35th episode
- Consistent solves at 79th episode
- 922 consecutive balancing episodes



Optimum Policy

- Optimum policy map based on the learned Q
- Policy not a function of x or dx/dt
- Unexplored (TBD) bins do not impact learning



Q-Learning Variants

QL with Rewards Favoring Desired States

- The reward function is modified to favor states with small |θ| and |x|:

$$R(x,\theta) = 1 + 10 * max(\,0,\, 1 - (\theta/\theta_{max})^2\,) + 0.1 * max(\,0,\, 1 - (x/x_{max})^2\,)$$

- Observed minor improvement in learning speed: 1st solve at 28th
- Learning convergence likelihood improved to ~100%

QL with Eligibility Traces

- Eligibility traces (ET) is tried using an update similar to Watkins's. At reach time step,

$$\delta = (\,R + \gamma Q(s', a'_{opt})\,) - Q(s,a)$$

$$Q(s,a) = Q(s,a) + \alpha\,\delta\,e(s,a)$$

where **e** is the eligibility vector and is updated for all (s'', a'') as

$$e(s'',a'') = \begin{cases} 0 & \text{if } a'' \neq a_{opt} \\ \gamma\,\lambda\,e(s,a) & \text{if } a'' = a_{opt} \end{cases}$$

$$e(s,a) = e(s,a) + 1 \qquad \text{for } s'' = s,\ a'' = a \text{ only}$$

- For λ = 0, the implementation is equivalent to Vanilla QL
- Performance improves as λ→0 indicating ET is not helpful here

Conclusion & Next Steps

Conclusion

- QL can balance the cart-pole with no prior knowledge of the physics in presence of complex dynamics and non-linearities
- State discretization ranges and hyper parameters significantly impact the learning performance
- Learning performance of QL variants for this problem rank as

  QL + Modified Rewards > QL > Eligibility Traces

Next Steps

- **Two step QL** where model 1st learns, next learns to control position
- **Position tracking** where a position reference is followed by the cart while balancing the pole
- **QL with Deep Neural Net** modeling Q in a more expressive way

References

[1] https://gym.openai.com/envs/CartPole-v1/

[2] Andrew G Barto, Richard S Sutton, and Charles W Anderson. "Neuronlike adaptive elements that can solve difficult learning control problems". In: IEEE transactions on systems, man, and cybernetics 5 (1983), pp. 834–846.