ACTOR-CRITIC METHODS

Jonathan Campbell
COMP-767
April 7, 2017

OVERVIEW

- Actor-critic method introduction.
  - Advantages of these methods
  - Example actor & critic.
- Results on cart-pole experiment.

ACTOR-CRITIC METHODS

- Actor
  - Determines policy π, e.g., P(a | s).
  - E.g., softmax policy
- Critic
  - Evaluates the current policy, e.g., through V(s).
  - Then update the actor's weights using the state's TD-error.
  - E.g., SARSA

ADVANTAGES OF ACTOR-CRITIC

- Smoother updates to policy than traditional algorithms.
  - E.g., in Q-learning:
    - small change in q-values could have large change in policy
  - But here policy has its own parameters.
- Good for continuous action spaces.
  - No need to take max q-val, e.g. (use policy parameters instead.)

CRITIC: SARSA(λ)

- Input: **α**: learning rate, **γ**: discount rate; **λ**: lambda value
- Input: **φ**(s, a): function for state-action features.
- Set of weights ***u***, size: length of state-action features.
- Eligibility trace vector ***e***

- Upon observation of (s, a, r, s'):
  - **$\delta = r + u^T[\max_{a'} \varphi(s, a')] - u^T\varphi(s, a)$**
    - **(Set next action to a'.)**
  - ***e* += φ(*s*, *a*)**
  - ***u* += α * δ * *e***
  - ***e* = γ * λ * *e***

ACTOR: SOFTMAX

- Input: **φ**(s, a): function for state-action features.
- Set of weights ***u***, size: length of state-action features.
    - Or could have ***u*** be matrix with dims [num. actions, num. feats].
- Choose action a w.r.t.:

$$\pi\left(a|s\right) = \frac{e^{u^T \varphi(s,a)}}{\sum_b e^{u^T \varphi(s,b)}}$$

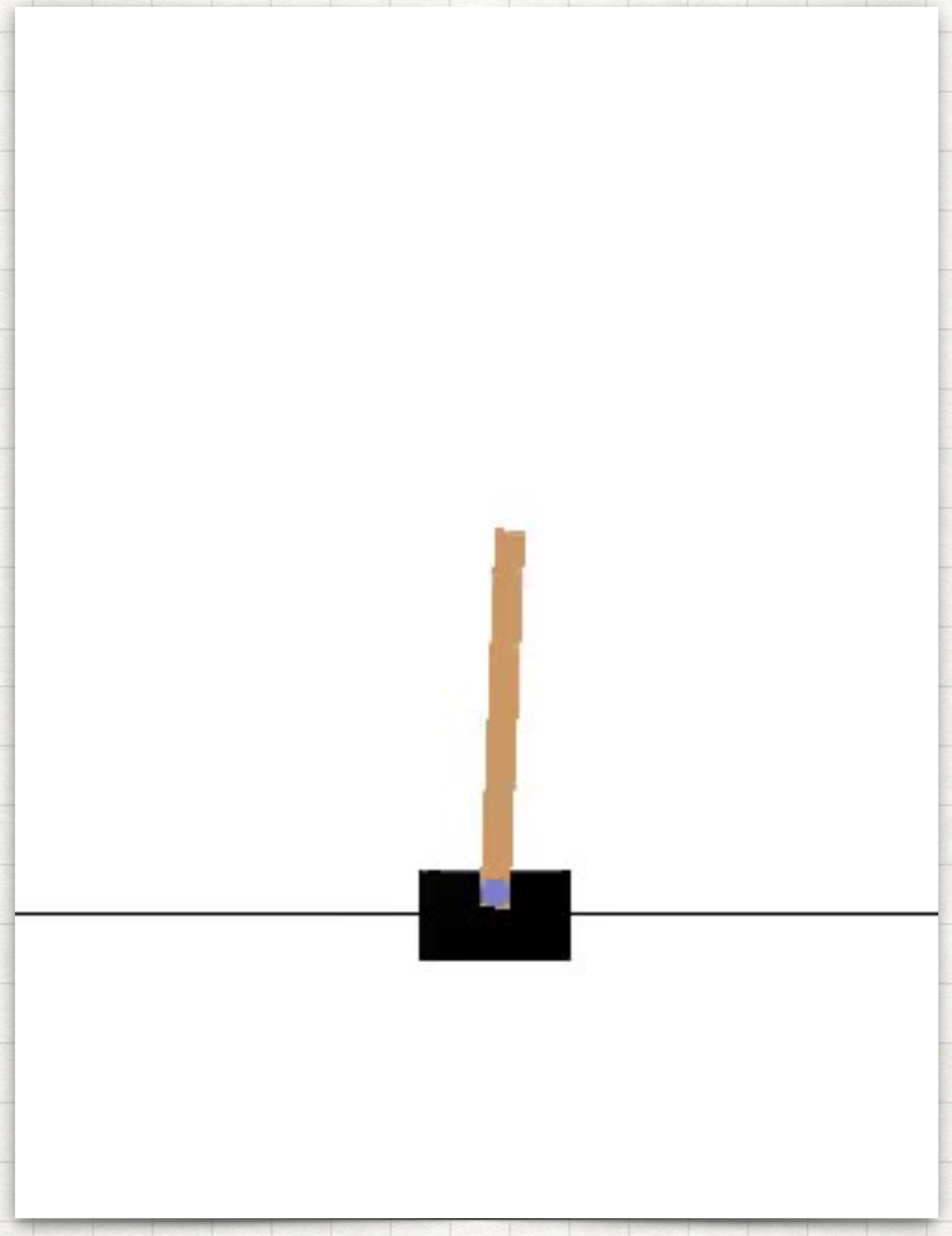
ACTOR UPDATE

- Input: **α**: learning rate, **γ**: discount rate; **λ**: lambda value
- Input: **φ**(s, a): function for state-action features.
- Input: **δ**: td-error from critic update
- Set of weights **w**
- Eligibility trace vector ***e***

- Upon observation of (s, a, r, s'):
  - ***e* += φ(s, a)**
  - ***u* += α * δ * *e***
  - ***e* = γ * λ * *e***

CART-POLE EXPERIMENT

- Goal: keep balanced a pole connected to a cart.
- Actions: apply force to move cart left/right.
- Also known as pendulum task.
- Using OpenAI gym for implementation.



METHOD

- Run grid-search over parameter space to determine best params.
  - Evaluate agent every 50 episodes.
    - Run 5 episodes to evaluate and average results.

- Best performing parameters (γ=0.995) on CartPole task:
  - $\boldsymbol{\lambda}$: 0
  - $\boldsymbol{\alpha}_{actor}$: 0.0007
  - $\boldsymbol{\alpha}_{critic}$: 0.003
  - (Better results if actor learns slower than critic.)

PARAMETER RESULTS