Summary of Policy-Based Methods

Monday, December 28, 2020 2:02 AM

Policy-Based Methods

- w/ **value-based methods**, the agent uses its experience w/ the environment to maintain an estimate of optimal action-value function estimate.

- **Policy-based methods** directly learn the optimal policy, without having to maintain a seperate value estimate.

Policy Function Approximation

- In Deep RL: policy is represented by a neural network.

  ↳ Input: State (Environment)

  ↳ Output: If the environment has discrete

actions, the output layer has a node for each possible action. (Contains probability of action)

- The weights is this neural network are initially set to random values. Then the weights adapt to learn the environment.

More on Policy

Policy Based Methods either:

1. Stochastic policies: Randomness
2. Deterministic policies: Based on values

Both can solve finite or continious action spaces.

Beyond Hill Climbing

- Hill Climbing Algorithm: Iterative Algorithm to find weights θ for optimal solution.

- At each iteration:
  - → slightly perturb values of the current best estimate for weights θ_{best} to yield new sets of weights
  - → Test these weights for an episode, if the return is higher $\theta_{best} \leftarrow \theta_{new}$.

Beyond Hill Climbing

- **Steepest Ascent Hill Climbing**: Variation of Hill Climbing that chooses a small number of neighbouring policies at each iteration & chooses the best among them.

- **Simulated Annealing** uses pre-defined schedule to control how the policy space is explored. (gradually reduce radius as we get closer to optimal solution).

- **Adaptive Noise Scaling:** decreases the search radius w/ each iteration when the new best policy is found, otherwise increase search radius.

More Black Box Optimization:

- **Cross-Entropy Method:**
  → iteratively suggest a small amount of neighbouring policies, & use a small percentage of the best performing policies to find new estimated.

- **Evolution Strategies:** technique considers the return corresponding to each candidate policy. The policy at next iteration is a weighted sum of all the candidates.

(Higher returned are weighted more)

Why Policy-Based Methods?

1. Simplicity: Policy-Based Methods directly get to the problem at hand (estimating policy, w/o action value estimate)

2. Stochastic Policy: Policy-based methods can learn true stochastic policies (e.g Rock-Paper Scissors)

3. Continious Action Spaces: Policy-based

Methods are well-suited for continious action spaces.