Применение методов исследования среды в алгоритмах model-based RL

Свидченко Олег Анатольевич
Научный руководитель: А. А. Шпильман

Санкт-Петербургская школа физико-математических
и компьютерных наук

НИУ ВШЭ - Санкт-Петербург

Введение

1. Современные алгоритмы обучения с подкреплением требуют большого количества опыта взаимодействия со средой

2. Одним из способов уменьшения количества опыта является использование model-based подхода

3. Также существует ряд алгоритмов, позволяющих эффективнее исследовать среду, и тем самым уменьшающих требуемое количество опыта

4. На данный момент два этих подхода существуют независимо друг от друга

Обучение с подкреплением

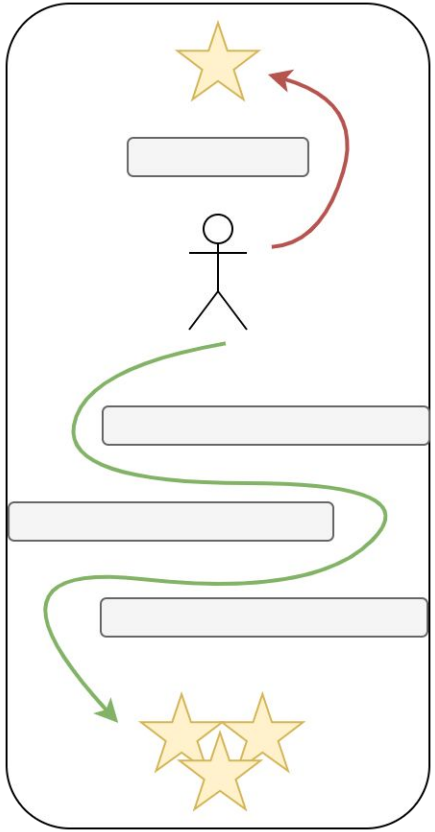
Проблема исследования среды

Как правило, найти оптимальную политику в среде сложно. Зачастую для этого требуется сложная последовательность действий.

Существуют простые способы исследования среды. Например:

1. Совершение случайных действий
2. Добавление случайного шума к действиям

В приведенном справа примере зеленая политика является оптимальной, однако найти ее с использованием простых методов исследования среды значительно сложнее, чем не оптимальную красную политику.



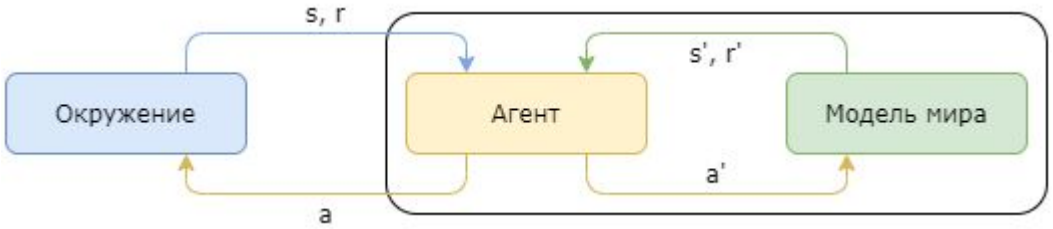
Современные методы исследования среды

1. Curiosity-driven Exploration by Self-supervised Prediction
   Pathak et. al., 2017
2. Episodic Curiosity through Reachability
   Savinov et. al, 2018
3. Count-Based Exploration with Neural Density Models
   Ostrovski et. al., 2018
4. Exploration by Random Network Distillation
   Burda et. al., 2018
5. Provably efficient maximum entropy exploration
   Hazan et. al., 2019
6. Novelty Search in Representational Space for Sample Efficient Exploration
   Tao et. al., 2020

Все работы кроме [5] используют механизм внутренней награды.

Model-based обучение с подкреплением

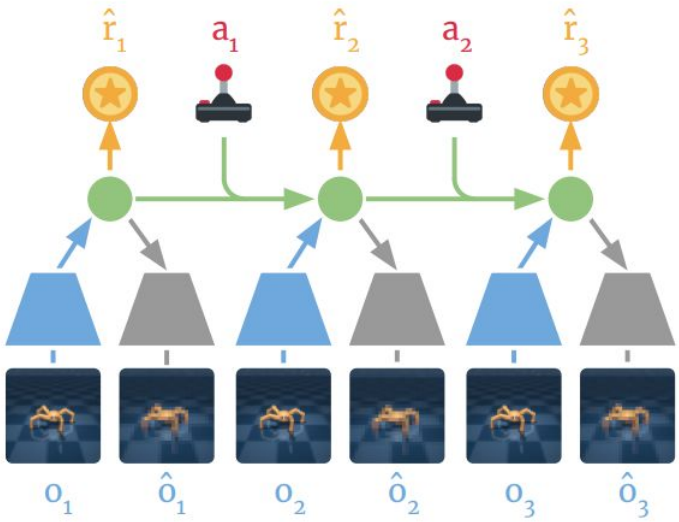
Методы model-based обучения с подкреплением предполагают, что нам известны функция перехода T и функция награды R. Это позволяет минимизировать количество необходимых взаимодействий со средой.



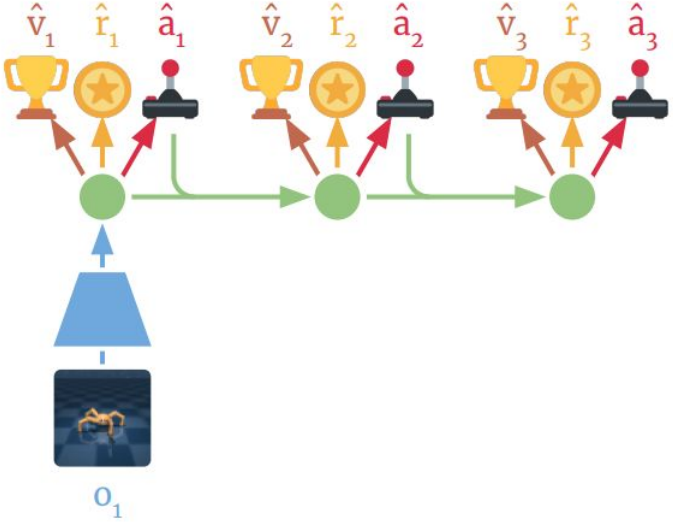
Замечание: Если T и R дифференцируемы, то суммарную награду можно максимизировать с помощью градиентного спуска.

Замечание 2: Для большинства окружений T и R неизвестны, поэтому на практике их приближают.

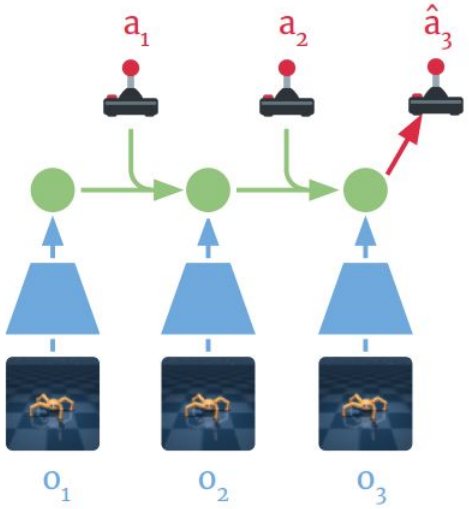
Алгоритм Dreamer [7]



Модель обучается по данным из реальной среды



Агент обучается по предсказаниям модели мира



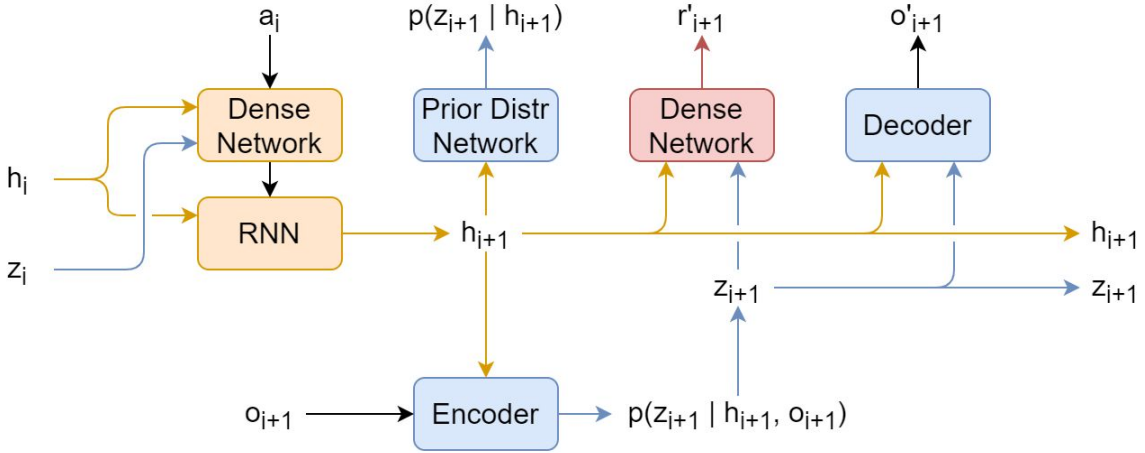
Агент применяется в реальной среде

7. Dream to Control: Learning Behaviors by Latent Imagination, Hafner et. al., 2020

Недостатки алгоритма Dreamer

Существенные недостатки алгоритма Dreamer:

1. Модель не учитывает вероятность завершения эпизода
2. Наблюдение в модель передается только после совершения действия
3. При подсчете апостериорного распределения стохастической части состояния градиенты проходят через детерминированную часть состояния
4. Обучаемое априорное распределение
5. Для исследования среды агент использует случайный шум



Цель и задачи

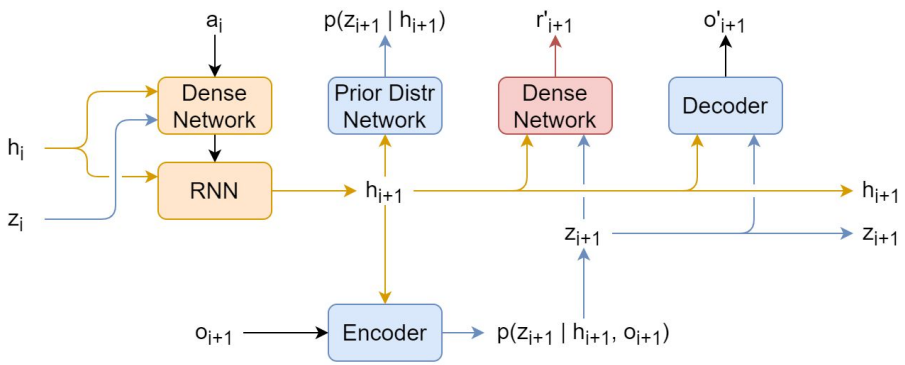
Цель: Повысить эффективность алгоритма Dreamer путем устранения его недостатков и улучшения исследования среды обучаемым агентом.

Задачи:

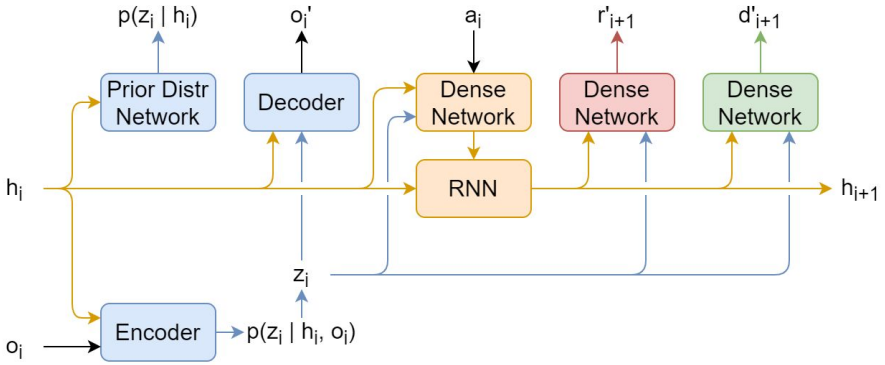
1. Разработать модификацию алгоритма Dreamer, позволяющую данному алгоритму учитывать вероятность завершения эпизода
2. Пересмотреть архитектуру модели мира алгоритма Dreamer с учетом вероятного возникновения проблем из-за неиспользования ей начального наблюдения среды
3. Разработать модификации алгоритма, повышающие стабильность и эффективность
4. Разработать способ исследования среды, учитывающий особенности model-based подхода
5. Провести сравнение оригинального и модифицированного алгоритмов

Модифицированный алгоритм Dreamer

Оригинальная модель мира



Модифицированная модель мира



Модификации:

1. Наблюдение передается до совершения действия агентом
2. Добавлена модель, приближающая вероятность завершения эпизода и соответствующим образом модифицирована оценка суммарной награды
3. Градиенты больше не распространяются через детерминированную часть состояния при обучении стохастической части состояния
4. Априорное распределение p(z) фиксировано

Maximum entropy exploration в Model-based RL

Хотим максимизировать энтропию распределения $p_\pi(s_{t+}|s_0)$

Определяем дисконтированное распределение состояний как:

$$p_\pi^\gamma(s_{t+}|s_0) = (1-\gamma)p_\pi(s_t|s_0) + \gamma p_\pi^\gamma(s_{(t+1)+}|s_0)$$

Распределение состояний на текущем шаге

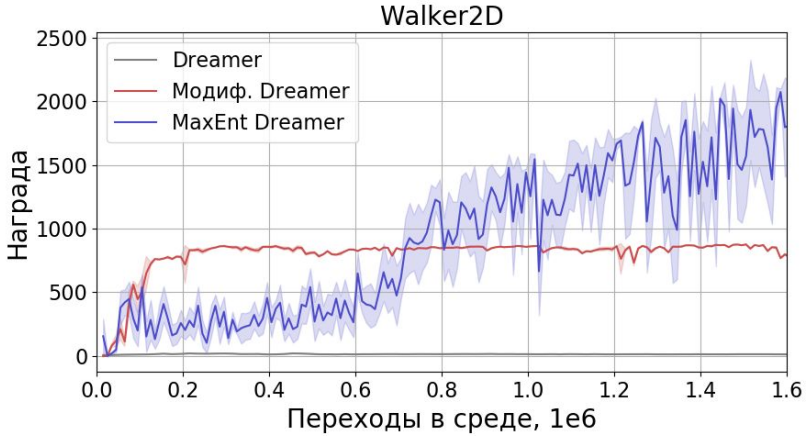
Распределение состояний на следующих шагах

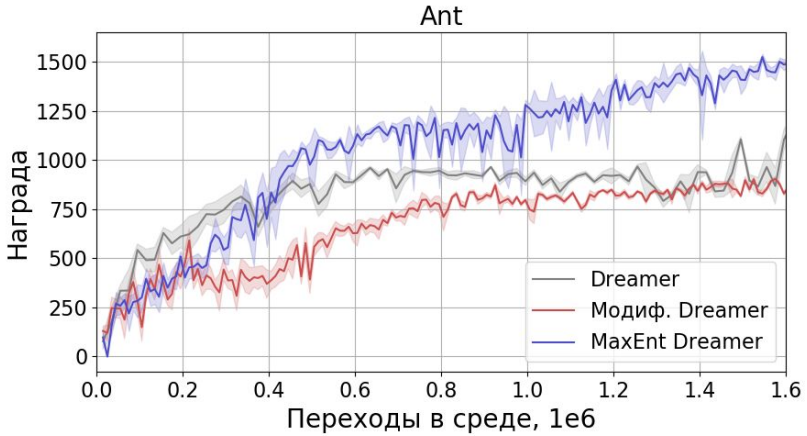
При обучении агента решается задача:

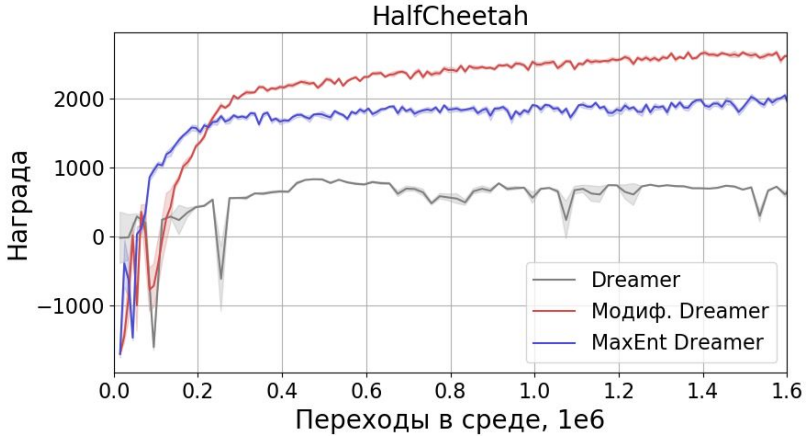
$$V_\pi(s_0) + \beta \cdot H(p_\pi^\gamma(s_{1+}|s_0)) \to_\pi \max$$

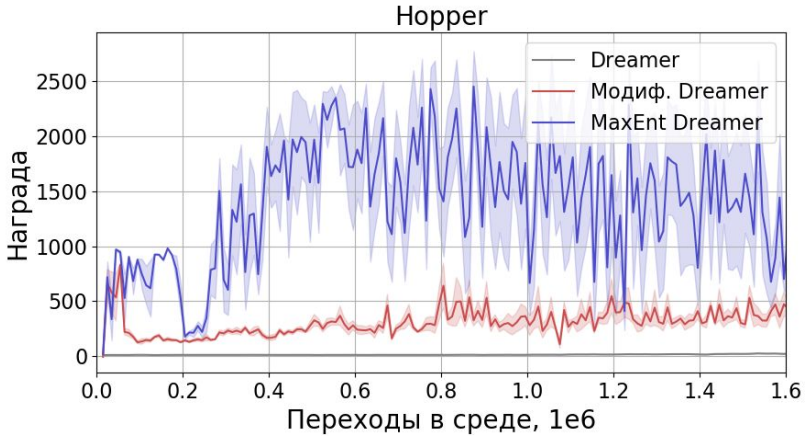
При этом дисконтированное распределение состояний приближается с помощью MDN по синтетическим данным, полученным с помощью модели мира.

Результаты


Walker2D
Награда
Переходы в среде, 1e6
Dreamer
Модиф. Dreamer
MaxEnt Dreamer


Ant
Награда
Переходы в среде, 1e6
Dreamer
Модиф. Dreamer
MaxEnt Dreamer


HalfCheetah
Награда
Переходы в среде, 1e6
Dreamer
Модиф. Dreamer
MaxEnt Dreamer


Hopper
Награда
Переходы в среде, 1e6
Dreamer
Модиф. Dreamer
MaxEnt Dreamer

Результаты

- Разработаны необходимые для повышения эффективности и стабильности модификации алгоритма Dreamer:
  - Добавлена модель, приближающая вероятность завершения эпизода
  - Стохастическая часть состояния строится до совершения действия
  - Модифицирован процесс обучения модели, отвечающий за восприятие наблюдения
- Разработан метод исследования среды, основанный на принципе максимизации энтропии распределения состояний и учитывает особенности model-based подхода
- Был проведен сравнительный анализ, который показал, что:
  - Модифицированный алгоритм с оригинальным методом исследования среды превосходит оригинальный Dreamer в 3 из 4 сред
  - Maximum Entropy Dreamer превосходит модифицированный алгоритм с оригинальным методом исследования среды в 3 из 4 сред
  - Maximum Entropy Dreamer превосходит оригинальный Dreamer во всех средах, использованных для проведения анализа