ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В ЗАДАЧЕ УПРАВЛЕНИЯ РОБОТОМ-КУБОМ

Шевляков А.А.

Институт проблем управления им. В.А. Трапезникова РАН,
Россия, г. Москва, ул. Профсоюзная д.65
aash29@gmail.com

Аннотация: В настоящий момент большой интерес вызывает применение различных алгоритмов машинного обучения в задачах робототехники. Нами сделан обзор наиболее широко применяемых методов обучения с подкреплением, которые мы применили к задаче стабилизации тележки с маятником как к упрощенной модели робота куба.

Ключевые слова: робототехника, управление, машинное обучение

Обзор

Если задаться целью сделать обзор самых заметных методов обучения с подкреплением, предложенных в последние годы, необходимо отметить следующие:

- Deep Q Network (DQN) [5] Одна из первых статей, где авторы успешно применили подход Q-обучения в совокупности с многослойной нейросетью. Алгоритм впервые научился играть в классические аркады для Atari лучше человека.
- Deep Deterministic Policy Gradient (DDPG) [4] Вариант DQN для задач, где управление задано непрерывной функцией. Больше подходит для задач робототехники.
- Hindsight Experience Replay (HER) [1] Предлагает новый подход к формированию функции награды, которое выполняется путем предобучения нейросети либо на сессиях с экспертом, либо на другом контроллере.

Наиболее громких результатов в машинном обучении добились компании компании OpenAI (DotA 2) и DeepMind (AlphaGo).

Вместе с тем, до недавнего времени обучение применялось в основном к игровым задачам, или же в виртуальных средах, смоделированных на компьютере. Во многом это объясняется необходимостью очень долгого обучения, которое в реальном времени заняло бы сотни лет, а также представляло бы опасность для управляемого объекта.

Компанией OpenAI была решена задача управления рукой-манипуляторов с целью захвата кубика, для чего создана система Dactyl [6]. Также OpenAI была выпущена библиотека gym, которая предоставляет набор тестовых задач с общим интерфейсом для применения и отработки различных алгоритмов обучения.

1 Модель движения робота-куба

Уравнения плоского движения куба

$$\begin{aligned} & m\frac{d\bar{V}_0}{dt} = -m\bar{g} + \sum_{i=1}^{4}(\bar{\lambda}_{n,i} + \bar{\lambda}_{t,i}), \\ & I\frac{d\omega}{dt} = \sum_{i=1}^{4}(\bar{r}_i \times (\bar{\lambda}_{n,i} + \bar{\lambda}_{t,i}))|_z + u, \end{aligned} \tag{1}$$

$$0 \le y_i \perp \lambda_{n,i} \ge 0, \tag{2}$$

$$\begin{aligned} & \dot{x}_i = 0, -\mu\lambda_{n,i} \le \lambda_{t,i} \le \mu\lambda_{n,i}, \\ & \dot{x}_i > 0, \lambda_{t,i} = -\mu\lambda_{n,i}, \\ & \dot{x}_i < 0, \lambda_{t,i} = \mu\lambda_{n,i}. \end{aligned} \tag{3}$$

Здесь V_0 — скорость центра масс куба, r_i — радиус-векторы вершин, $\bar{F}_i^N$ и $\bar{F}_i^T$ — силы реакции поверхности в этих точках. Запись $0 \le y_i \perp F_i^N \ge 0$ означает, что при наличии касания пола точкой i только одна из величин y_i и N_i^y не равна 0, и обе они больше либо равны 0. ω — угловая скорость тела. Последние 3 уравнения описывают закон трения Кулона, т.е. сухое трение и трение покоя в точках контакта.

Случай с ненулевым трением может быть сведен к линейной задаче дополнительности с помощью следующих переобозначений.

Можно упростить задачу: будем рассматривать единственную точку контакта, ровная поверхность пола, положим m=1,I=1.

$$\ddot{x} = \lambda_x$$
$$\ddot{z} = -g + \lambda_z$$
$$\ddot{\theta} = \lambda_z \sin\theta + \lambda_x \cos\theta + u,$$

где $\lambda_x^+ - \lambda_x^- = \lambda_x$.

В случае проскальзывания можно выразить λ_z через переменные состояния

$$\lambda_z = \frac{g - u\sin\theta - \cos\theta\dot{\theta}^2}{1 + \sin^2\theta - sign(V_1)\mu\sin\theta\cos\theta}$$

Как, управляя моментом, приложенным к кубу, переместить его в нужную точку?

У такой системы будет 3 режима:

1. Свободное движение в отсутствие контакта (3 степени свободы)
2. Контакт с проскальзыванием (2 степени свободы)
3. Контакт без проскальзывания (1 степень свободы)

2 Обучение с подкреплением

Движение в режиме 2 напоминает задачу о тележке с перевернутым маятником. Задача о тележке является популярным "тестовым стендом" для различных алгоритмов управления. В их число входят и алгоритмы машинного обучения и обучения с подкреплением, начиная от классических статей [2] и кончая окружением cartpole из среды OpenAI gym.

При этом, однако, постановка задачи как правило отличается от принятой в теории управления.

A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The system is controlled by applying a force of +1 or -1 to the cart. The pendulum starts upright, and the goal is to prevent it from falling over. A reward of +1 is provided for every timestep that the pole remains upright. [8]

Мы видим, что управление дискретно, а максимизация описанной функции награды обеспечивает лишь то, что маятник не упадет, но не ведет, вообще говоря, даже к стабилизации его верхнего положения равновесия, и тем более к стабилизации положения всей системы.

Для решения задачи стабилизации необходимо рассмотреть другую функцию награды. По аналогии с другими робототехническими исследованиями (напр. [7]), используем расстояние до заданной точки.

Поскольку функция награды должна возрастать при приближении к целевому состоянию, расстояние было учтено следующим образом:

$$r = 100e^{-10((x_1 - x^*)^2 + (y_1 - y^*)^2)} - 20,$$

где x^*, y^* – желаемое положение конца маятника. Данная функция выбрана так, чтобы иметь ярко выраженный максимум окрестности целевой точки, "штрафовать" агента за слишком сильное удаление от нее и иметь всюду ненулевой градиент.

Помимо награды, в стандартном окружении cartpole-v1 gym было модифицировано управление. От дискретного управления, равного ± 1, мы перешли к непрерывной силе $F \in [-100,100]$.

3 Результаты

Для тестирования были выбраны методы, реализованные в пакете Stable Baselines [3].

Контроллер обучался по алгоритмам DDPG, SAC и GAIL в течение 10^6 итераций. В случае GAIL использовалось предобучение с помощью линейного регулятора, удовлетворительно решавшего исходную задачу.

Начальное и целевое положения системы задавались случайным образом в заданных пределах. При выходе за допустимые рамки (± 30 угол отклонения,± 2.4м перемещение), либо по истечении 5000 шагов по времени эпизод останавливался, и обучение перезапускалось. Параметры методов оптимизации брались из пакета Stable Baselines.

К сожалению, ни один из представленных алгоритмов не справился с обучением. Обученный контроллер как правило немедленно нарушал установленные ограничения.

Причины такого провала не вполне ясны. Рассмотренная система является, по меркам теории управления, довольно простой и хорошо изученной. Для нее предложены десятки различных законов управления, даже самые простые из которых удовлетворительно решают задачу в рассматриваемой области. Один из них использовался для предобучения нейронной сети, что однако не дало ожидаемого эффекта.

Литература

1.*Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba.* Hindsight experience replay. In Advances in Neural Information Processing Systems, pages 5048–5058, 2017.

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

3. Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. https://github.com/hill-a/stable-baselines, 2018.

4. Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015.

5. Volodymyr Mnih, Kavukcuoglu Koray, David Silver, Alex Graves, Ioannis Antonoglou, and Riedmiller Martin. Playing atari with deep reinforcement learning. In Workshop on Deep Learning, NIPS, 2013.

6. OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, RafaE, JГizefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. CoRR, 2018.

7. *Axel Rottmann, Christian Plagemann, Peter Hilgers, and Wolfram Burgard*. Autonomous blimp control using model-free reinforcement learning in a continuous state and action space. In *2007 IEEE/RSJ International* Conference on Intelligent Robots and Systems, pages 1895–1900. IEEE, 2007.

8. Cartpole-v1 gym. https://gym.openai.com/envs/CartPole-v1/.