Отчет о проделанной работе за последнее время

21 февраля 2017 г.

1 Хакатон

Хакатон http://rl.deephack.me прошел, в целом, достаточно результативно. В финальном очном туре нужно было играть в игру Pacman. Вот этот энвайромент на aigym:

https://gym.openai.com/envs/MsPacman-v0

наш алгоритм - называется "sw1sh's algorithm". sw1sh - это аккаунт моего коллеги с работы, с которым мы были в команде.

Наша команда заняла 4е место, однако мы использовали стандартный алгортим для обучения DQN на нескольких GPU. Он разработан DeepMind и называется The General Reinforcement Learning Architecture (Gorila). Этот алгоритм описан вот в этой статье:

https://arxiv.org/pdf/1602.01783v2.pdf

Там описывается как нужно играть несколькими агентами параллельно, чтобы обучать одну сеть. С точки зрения машинного обучения самая интересная вещь там -это то, что такой тип алгоритмов не нуждается в experience replay

http://rll.berkeley.edu/deeprlworkshop/papers/database_composition.pdf

Это очень простой метод, который закулючается в том, чтобы обучать Q-сеть не на последних нескольких наблюдениях, а записывать в память много разных налюдений и обучаться на них. Этот способ обучения DQN всегда необходим при использовании методов с одним агентом, потому что иначе , если приобучении использовать только последний опыт, получается что налюдения слишком скорреливарованы и сеть обучается очень плохо.

Так вот, при асинхронном обучении по опыту нескольких агентов одновременно, в этом методе нет необходимости.

Несмотря на то, что мы заняли четвертое место, мы по сути использвали самый стандартный алоритм и благодаря большим вычислительным мощщностям смогли получить результат.

2 Обучение агента в задаче, описанной в пердыдущем отчете

В предыдущем отчете я описывал, что я решал задачу, когда агент должен ловить пиксель, падающий сверху вниз по черно-белой картинке. Я успешно

решил задачу для небольших картинок с помощью feedforward сети, однако, обучить агента для больших задач не получилось.

Я решил попробовать использование сверточных сетей, чтобы решить задачу побольше, но несмортря на то, что я перебрал множество конфигураций сверточных сетей, в том числе такие, котрые были описаны в различных статьях по обучению с подкреплением, мне не удалось обучить сверточную сеть для решения этой задачи.

Я планирую продолжить попытки.

3 Safe reinforcement learning

Также я заинтресовался хорошей и нужной, на мой взгляд задачей - безопасное обучение с подкреплением. Я прочитал эту статью с обзором основных техник бехопасного обучения с подкреплением:

http://www.jmlr.org/papers/volume16/garcia15a/garcia15a.pdf

Статья делится части по основным направлениям безопасного обучения с подкреплением:

- Введение модифицириванного критерия оптимальности для учета риска
  - Критерии худшего случая в разных модификациях. Такие критерии оптимизаруют не среднюю награду, а награду худшего случая.
  - Риск-критерии - это критерии, которые учитывают при оптимизации одновременно среднее значение наград и его дисперсию, отдавая предпочтения стратегиям, у которых меньше дисперсия, а значит, которые более надежные.
  - Критерии, которые оптимизируются при помощи алгоритмов условной оптимизации. Идея тут тоже на поверхности: мы учитываем наш риса, просто не позволяя выбирать небезопасные стратегии на уровне оптимизации.
  - Прочие критерии, учитывающие риск: среди них можно выделить такие, которые, например используют VAR вместе дисперсии в качестве меры разброса. А также другие меры разброса.
- Учет риска в ходе исследовательского процесса. В этой части рассматриваются алгоритмы, которые не позволяют агенту быть поврежденным(как это может быть в случае падения дорогостоящего робота) в ходе исследовательского процесса.
  - Внешнее знание. Тут в систему агент-среда добавляется некоторый эксперт, который отвечает агенту на вопросы о безопасности того или иного действия. В разных модификациях эксперт может оценивать любое действие или только по завпросу агента.
  - Подходы, в которых мера риска используется для определения вероятностей выбора различных событий в ходе исследовательного процесса

4 Выступление на конференции Ломоносов

Изучать тему Safe reinforcement learning я начал, потому что меня заинтересовала одна прикладная задача. Допустим, у нас есть манипулятор на заводе, скажем, роботизированная рука. Нам нужно научить ее дотягиваться до отдаленных частей сложных деталей, при этом ничего лишнего не задеть.

Мы можем создать виртуальную среду, которая симулирует реальность и физику движения этой роботизированной руки. Сделать систему, в которой агент получает награду за достижение искомых частей детали. Обучить там агента и встроить уже обученную модель в реального робота.

Я понял, что ни один алгоритм из описанных выше не позволяет быть на 100 процентов уверенным в том, что робот не сделает ошибки. Дело в том, что когда речь идет о любой модификации критерия оптимальности, мы все равно делаем некоторую модель для описании функции Q (если используем, например, Q-learning). И мы не можем быть уверенным, что во всех точках она обучена достаточно хорошо и не произойдет критической ошибки.

Однако, немного пожертвовав оптимальностью, мы можем получить полностью надежную систему для таких задач. Для этого нужно обучить некоторую количество надежных траекторий заранее в виртуальной среде, а потом для точек пространства, в которых может оказаться агент вначале, жадно расчитать стратегию достижения этих траекторий.

Это врядли можно сделять для каждой точки, но для реальной задачи, можно наложить условия гладкости на пространство, в котором мы работаем и тогда можно проходить лишь часть точек, с каким-то шагом. Это очень дорого, с точки зрения вычислений, но, кажется, это может позволить нам получит абсолютно надежную систему.

Я хотел бы сделать какой-нибудь работающий прототип и показать его на конференции Ломоносов.