Assignment 1

Due March 23

1 Theoretical assignment: TD(λ)

Данное теоретическое задание посвящено алгоритму TD(λ).

Задание TD.1 Доказать, что если вычислять обновления весов на каждом шаге (без их применения), то суммарное обновление онлайнового и оффлайного варианта одинаково.

Под оффлайновым обновлением подразумевается оффлайновый алгоритм λ-отдачи. Под онлайновым обновлением подразумевается алгоритм TD(λ). Подсказки можно найти в заданиях в конце главы 12.1 книги Саттона и Барто.

Задание TD.2 Доказать равнозначность оффлайного алгоритма λ-отдачи и истинно онлайнового алгоритма TD(λ).

См. соответствующую главу 12.5 книги Саттона и Барто и статью van Seijen et al., 2016.

2 Practical assignment: Imitation learning

Цель задания — реализовать и поэкспериментировать с алгоритмами имитационного обучения. По имеющимся демонстрациям предлагается обучить два варианта стратегии: с помощью алгоритма клонирования поведения (лекция по имитации поведения) и алгоритма DAgger (см. оригинальная статья), а затем сравнить их качество в средах на базе Mujoco. Данное задание основано на первом задании курса по Deep RL Университета Беркли.

Код задания расположен по адресу

https://github.com/pkuderov/mipt-rl-hw-2023

Есть возможность выполнять задание как локально, так и в Google Colab. Подробности, в том числе по установке зависимостей, вы найдете в README в репозитории.

2.1 Behavioral Cloning

Требуется заполнить пропуски в реализации алгоритма клонирования поведения. По адресу `expert_data` содержатся *pickled* данные экспертных стратегий. Рекомендуется начать ознакомление с точки запуска `scripts/run_hw1.py` и затем продвигаться вглубь последовательно:

- `infrastructure/rl_trainer.py`,
- `agents/bc_agent.py`,

- `policies/MLP_policy.py`
- и далее, заполняя пропуски, отмеченные `TODO`.

Задание BC.1 Приведите результаты запуска вашей реализации в двух средах: Ant и любой другой. Требуется, чтобы в среде Ant качество было не хуже 25% от качества эксперта. Пример запуска приведен в README. В качестве результатов приведите следующие данные: среднее значение (`Eval_AverageReturn`) и стандартное отклонение отдачи (`Eval_StdReturn`) по набору тестовых эпизодов.

Обратите внимание, чтобы собрать данные по нескольким эпизодам, потребуется при запуске указать параметр суммарного количества шагов ***`eval_batch_size`*** *в несколько раз больше параметра максимальной длины эпизода* ***`ep_len`****. Собственно, их отношение и задает минимальное число эпизодов, которые войдут в выборку.*

Чтобы включить генерацию видео при логировании, воспользуйтесь аргументом запуска `--video_log_freq -1`*. Удалите его при необходимости, но помните о возможности существенного замедления выполнения.*

Задание BC.2 Поэкспериментируйте с разными наборами гиперпараметров (например, число шагов обучения, объем предоставленных экспертных данных и т.п.). Для одного фиксированного гиперпараметра постройте график изменения качества работы агента [на одной из сред из первого задания] и поясните ваш выбор этого параметра.

2.2 DAgger

Заполнив все пропуски `TODO`, вы также сможете запустить DAgger (см. аргументы запуска в README).

Задание DA.1 Проведите запуски алгоритма на выбранных в задании 1 средах. Постройте график обучения DAgger'a — число итераций алгоритма vs. средняя отдача за эпизод (с указанием стандартного отклонения). Добавьте в этот график горизонтальными линиями результаты эксперта и клонирования поведения. Укажи использованные гиперпараметры.

3 Формат сдачи

Сдача предполагается в виде коммитов и комментариев к ним в соответствующем предложении изменения кода (pull request) в GitHub classroom (ссылка в тг канале курса).

Ожидается, что сдача будет содержать непосредственно код заполненных вами недостающих частей выданной заготовки решения и логи финальных запусков (для каждого задания и каждой из использованных сред).

Оригинально все логи лежат в папке `data`. Логи финальных запусков скопируйте из `data` в отдельную папку `run_logs` и отправьте вместе с вашим решением. Отличной альтернативой будет вместо этого снабдить решение ссылкой на отчет (report) в wandb, что потребует самостоятельно интегрировать использование данного сервиса в ваше решение.

Также в сообщении к предложению необходимо добавить результаты, описание и решение по каждому из пунктов задания (в соответствии с тем, что оно требует). Разметка markdown позволяет и вставку картинок, и оформление табличек. Опционально, вы можете оформить результаты в виде отдельного файла .doc или .pdf и добавить их в

посылку (commit), а в сообщении сослаться на этот файл. Не забудьте для каждого пункта задания код запуска, чтобы можно было воспроизвести ваши результаты.