

# Отчет о проделанной работе за последнее время

7 февраля 2017 г.

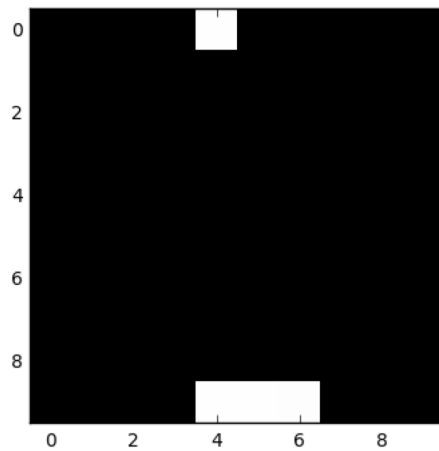
В последние две недели, как я и говорил, я начал заниматься подготовкой к хакатону deerHack.RL <http://rl.deerhack.me>

Там нужно было сначала (для попадания на очную часть хакатона) решить задачу типа RL решения для одной игры Atari. Особенно никто этого делать не умеет, но несколько команд, в том числе наша, смогли предоставить хоть какое-то решение. Игра предлагалась вот такая:



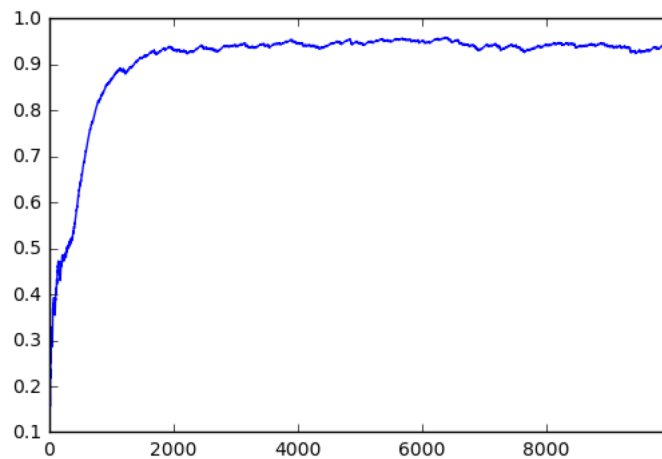
Задача в том, чтобы попадать в ворота и спускаться как можно быстрее. На вход, в качестве состояния, агенту подается только картинка вроде этой. Нужно было научиться им управлять, но, повторюсь, никто особенно этого так и не научился.

Я решил начать с более простой задачи обучения сетей в RL, но по сути что-то похожее:

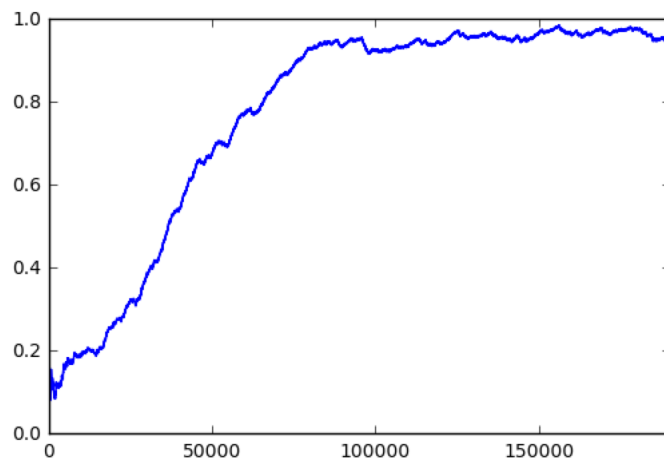


Тут в качестве состояния агент получает черно-белую картинку (никаких оттенков серого, только черный и белый цвет) и должен научиться ловить падающий вертикально пиксель при помощи "корзинки" снизу. Эта задача намного проще и поэтому ее удалось решить сразу для размеров 10x10, 20x20, 30x30, вот история обучения алгоритма:

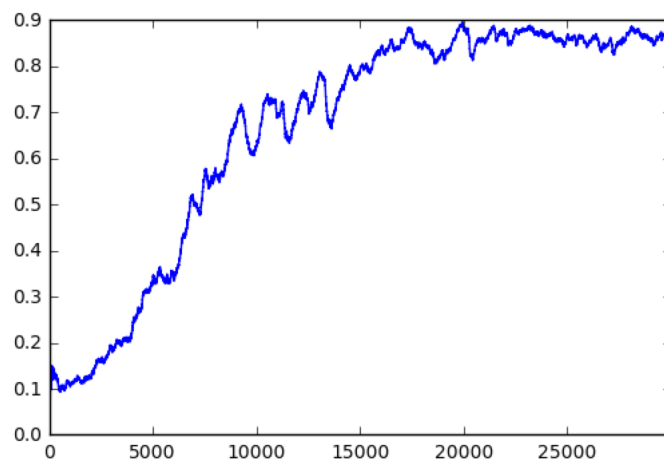
Случай 10x10, уже меньше, чем за 2000 эпизодов сеть обучилась нормально. По оси y - процент удачных попыток поймать пиксель (0.9 - это лучшее, но что мы надеемся, потому что остальные 10 процентов мы оставили для случайных действий для исследования среды), по x - количество эпизодов обучения.



Случай 20x20, тут число эпизодов нужно делить на 20, это связано с моим багом, но я уже не мог регенерить картинку, потому что процесс обучения достаточно долгий. В общем, чуть меньше, чем за 5000 эпизодов сеть удалось обучить.



Случай 30x30, тут потребовалось уже больше 15 000 эпизодов для обучения сети.



Игру размером 40x40 и больше мне не удалось обучить вовсе, хотя я долго пробовал менять разные конфигурации сети.

Стоит заметить, что в предыдущих случаях использовалась обычная feed forward сеть, которая запоминает практически все случаи отдельно (мне так кажется). Я пробовал обучать сверточную сеть, так как она может выделять признаки все более и более общие от слоя к слою. Кажется, что это как раз то, что нам нужно, потому что на первых этапах как раз достаточно знать, что писель где-то в левых 10ти пикселях. А потом уже можно было бы как-то это дальше уточнять. Это очень важно в больших задачах, чтобы получить хорошее обобщение и не выучивать все отдельные случаи.

Но, повторюсь, обучить сверточную сеть мне пока не удалось даже для сравнительно небольшой задачи (20x20).

В ближайшее время мне хотелось бы научиться обучать сверточные сети хотя бы для этой простой задачи и решить ее для размеров 40x40 и 50x50, заодно поработаю над обучением нейронных сетей в целом, благо на работе нам выделили сервер с GPU картами и дают ими пользоваться достаточно

свободно.

$$Loss(X_{pred}, X_{real}) = -\log P(\xi = X_{real}),$$

$$\xi \sim Poiss(X_{pred})$$