

MPUM project - flappy bird reinforcement learning

Bartosz Bromblik & Jacek Markiewicz

1 Wstęp

"*Flappy bird*" to prosta gra z długą historią irytowania wszystkich graczy. Mimo swojej prostoty, wcale nie jest prosta. A przynajmniej dla człowieka. A czy tak samo jest dla maszyny? Celem projektu jest sprawdzenie skuteczności domowej roboty implementacji kilku algorytmów uczenia przez wzmocnienie (ang. RL - Reinforcement Learning), tu:

- Q-learning,
- TD(0),
- przez sieć neuronową,
- Algorytm genetyczny.

Do raportu załączone jest (chaotyczne) repozytorium z zaklepanymi algorytmami, ich wynikami oraz samą grą.

2 Opis gry

Jesteśmy ptakiem (żółtą kulką) i chcemy dolecieć jak najdalej. Ale na naszej drodze jest pełno rur (to zielone). Na szczęście są w nich dziury, przez które możemy spróbować przelecieć. I to tyle co można powiedzieć o celu gry.

Całość kontroli to klikanie *spacji*, które powoduje skok ptaka. Poza tym *Esc* wyłącza grę, *R* (niekoniecznie duże) restartuje rozgrywkę. Każdy inny przycisk działa jako pauza (lub tę pauzę cofa).

Gra jest mniej więcej w pełni konfigurowalna. Parametry można znaleźć w pliku *game_config.py*. Ptak zawsze zaczyna na wysokości połowy mapy, z prędkością poziomą równą 0. Grawitacja jest stała, prędkość pozioma też. Skok ustawia prędkość pionową na tę samą wartość. Grubość (pozioma) rury jest stała. Odległości między rurami są brane z rozkładu normalnego. Środek dziury jest brany z rozkładu jednostajnego. Promień dziury jest brany z rozkładu normalnego i z każdą kolejną rurą maleje (tempo zależy od poziomu trudności).

Hiperparametr *HARD* określa poziom trudności gry (*False* to łatwy, *True* to trudny).

Silnik gry nie ma wbudowanego zegara. Dzięki temu można trenować modele szybciej niż w

czasie rzeczywistym. A w trybach do grania jest oddzielny zegar, który pilnuje odpowiedniego framerate'u.

Moduły do wyświetlania gry jest napisany w pygame'ie.

3 Granie

Pliki *play_*.py* to ta grywalna część projektu. Każdy odpowiada któremuś z algorytmów (sposób implementacji może się różnić między plikami) poza plikiem *play-yourself.py*, gdzie nie ma żadnego wspomagania. Tak, sterowanie działa wszędzie i algorytmom można przeszkadzać.

4 Ogólnie o uczeniu

Podczas gry, na górze okna wyświetlają się na czerwono 3 tuple liczb. Oznaczają one kolejno:

- obecną (uogólnioną) stratę, czyli odległość pozioma minus różnica poziomów środka ptaka i środka następnej dziury, używana do oceny modelu. Często zwana również **wynikiem**.
- stan, czyli (pozycję pionową, prędkość pionową, pozycję dolnego końca następnej dziury, pozycję górnego końca następnej dziury, odległość poziomą do następnej rury). Przy czym to ostatnie jest liczone od prawego końca ptaka do lewego końca rury. Dopiero gdy ptak w całości minie linię końca rury, zmieniana jest "następna rura".
- (ilość miniętych rur, status), gdzie to drugie to 1 gdy dalej żyjemy i 0 gdy już nie.

5 Uczenie nie maszynowe

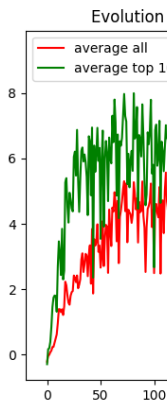
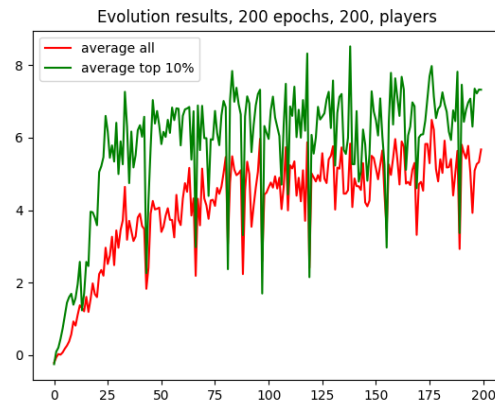
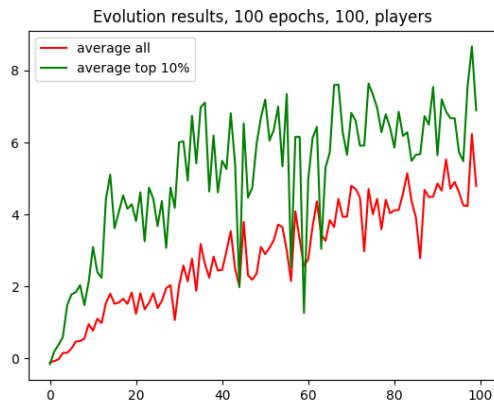
Jako kontekst powiem, że pomimo wielu prób, nie udało mi się zdobyć wyniku powyżej 7 (samej siódemki też zresztą nie).

6 Q-learning

7 TD(0)

8 Sieć neuronowa

9 Algorytm genetyczny



10 Podsumowanie