Zadanie 05

Uratujmy lemingi!

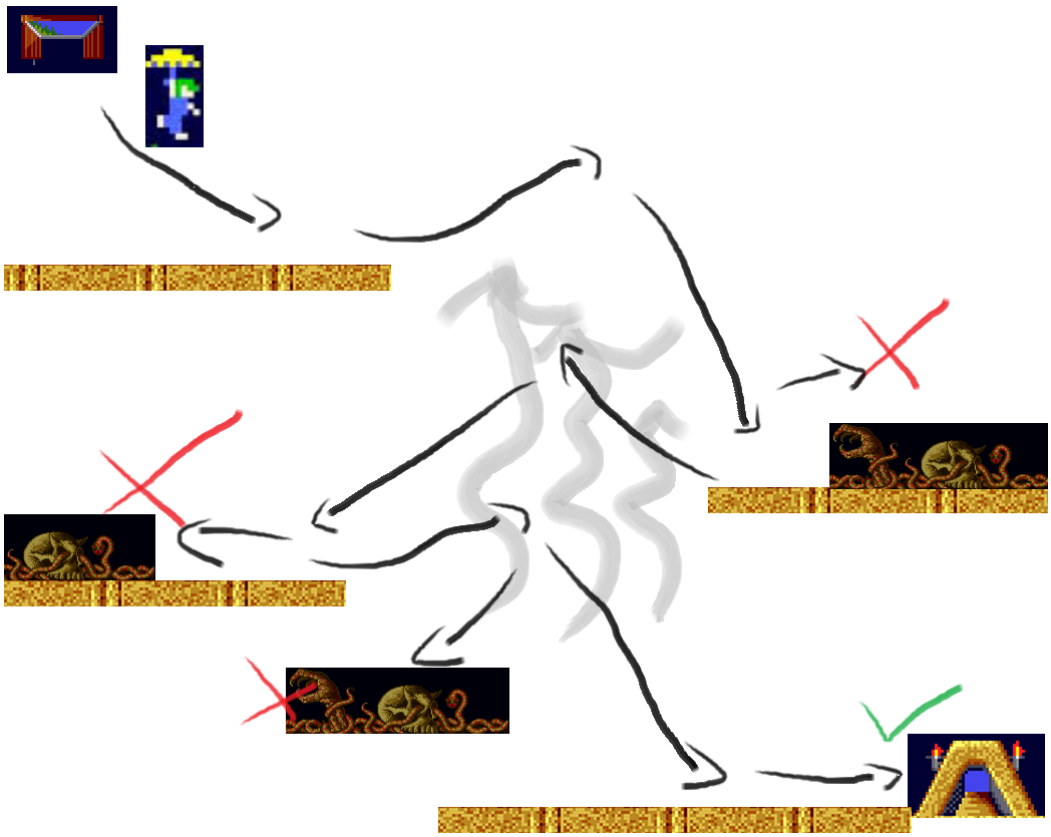
Zadanie jest mocno inspirowane podobnymi ćwiczeniami zawartymi w podręczniku spisanym przez parę Sutton&Barto oraz klasyczną grą *Lemmings* z 1991 roku (i niespecjalnie to ukrywa ;]). Jego celem jest zapoznanie się z podstawowymi koncepcjami pojawiającymi się w kontekście uczenia ze wzmocnieniem (na przykładzie algorytmu SARSA).

Lot na parasolce

Zacząć musimy od definicji samego problemu. Oryginalne "lemingi" mają nieco zbyt dużą przestrzeń stanów (i akcji) jak na skalę tego zadania - wykorzystamy więc znacznie mniej skomplikowany wariant. Środowisko gry trzeba niestety zaimplementować samodzielnie - ale nie powinno zjeść to zbyt dużo czasu, jego założenia są maksymalnie uproszczone.

- Gra odbywa się na prostokątnej planszy podzielonej na kwadratowe pola.
- W lewym górnym rogu znajduje się brama wejściowa, przez którą wchodzą poszczególne stworki - w prawym dolnym jej wyjściowy odpowiednik.
- Głównym celem gry jest bezpieczne dotransportowanie jak największej liczby stworków od wejścia do wyjścia.
    - Celem pobocznym jest zrobienie tego jak najsprawniej - wśród dwóch wyników o tym samym odsetku dostarczonych lemingów lepszy jest ten, który wykonał operację w mniejszej liczbie kroków.
- Poziomy składają się z następujących elementów:
    - wyróżniony kwadrat wejściowy - tu będą pojawiać się stworki;
    - wyróżniony kwadrat wyjściowy - tu będą chciały dotrzeć;
    - skał - na te kwadraty stworek nigdy nie może wejść (ani świadomie, ani na skutek grawitacji czy prądów powietrza);
    - cierni - wpadnięcie na te kwadraty skutkuje zgonem stworka;

    - grawitacji - powoduje ona, że po wykonaniu swojego ruchu stworek opada (na parasolce) o jedno pole w dół (jeżeli pole to nie zawiera skał);
    - podmuchów gorącego powietrza - ich siła jest definiowana osobno dla każdej kolumny danej planszy;
        - po wykonaniu ruchu i opadnięciu na skutek grawitacji stworek unoszony jest do góry o tyle pól, ile wynosi siła wiatru w tej kolumnie (oczywiście nie może przeniknąć przez skały).
- W danym poziomie mamy do dyspozycji stałą, skończoną liczbę stworków (np. 500 - konkretną wartość dobrać rozsądnie w zależności od rozmiaru i poziomu skomplikowania poziomu).
    - Lemingi (inaczej niż w oryginale) wchodzą na planszę pojedynczo.
    - Po pojawieniu się na planszy zachowaniem leminga steruje aktualna polityka uzyskana przez algorytm SARSA.
        - Stworek ma w każdej jednostce czasu do dyspozycji dwie opcje - poruszyć się w lewo lub w prawo.
    - Cykl życia stworka wygląda więc następująco:
        - podejmij decyzję o ruchu w lewo lub w prawo;
        - wykonaj ten ruch (jeżeli skutkuje to wejściem w skały - nie wykonuj, jeżeli skutkuje to wejściem w ciernie - stworek kończy swój żywot);
        - opadnij w dół o jedno pole (jeżeli skutkuje to wejściem w skały - nie wykonuj, jeżeli skutkuje wejściem w ciernie - stworek kończy swój żywot);
        - podnieś sie do góry o tyle pól, ile wynosi siła wiatru w tej kolumnie (jeżeli skutkuje to wejściem w skały - nie wykonuj, jeżeli skutkuje wejściem w ciernie - stworek kończy swój żywot);
        - powtarzaj, aż:
            - stworek dojdzie do wyjścia - wtedy jest uznany za uratowanego i kolejny zaczyna swoją podróż;
            - stworek rozpaćka się na cierniach - wtedy jest niestety stracony;
            - upłynie więcej niż ustalona maksymalna liczba ruchów (np. 50 - znów, warto to dobrać do rozmiarów planszy) - wtedy stworek wybucha z nudów (i też traktujemy go jako straconego).
    - Mamy tylko jedno podejście do danego poziomu! Algorytm będzie musiał decydować ile lemingów może poświęcić na rzecz eksploracji poziomu dla dobra pozostałych stworków.
- W ramach zadania trzeba zaimplementować powyższe reguły i przygotować przynajmniej 2 ciekawe poziomy. Jest okazja obudzić swojego wewnętrznego projektanta gier. ;]
- Poniżej (bardzo) poglądowy rysunek obrazująco-wyjaśniajacy.



Leming swój rozum ma

Ok, pora na właściwą część zadania, czyli uczenie ze wzmocnieniem. Co mamy do zrobienia?

- Musimy się przygotować do diagnostyki rozwiązań (lepszej niż "wydaje się, że jest lepiej").
    - Przygotujmy następujące wykresy:
        - wykres zależności uratowanych lemingów od czasu (mierzonego w akcjach, nie w epizodach - epizody mają różne czasy trwania!);
        - wykres pokazujący losy ostatnich 10-20 lemingów w zależności od epizodu (jaki % dotarł do wyjścia, jaki % zginął na cierniach, jaki % eksplodował z nudów);
        - wykres pokazujący średni czas trwania ostatnich 10-20 epizodów w zależności od epizodu;
        - wykres pokazujący średnią nagrodę z ostatnich 10-20 epizodów w zależności od epizodu (ten będzie wymagał przygotowania systemu nagradzania - o tym dalej).
    - Pamiętajmy, że algorytm jest mocno stochastyczny - żeby wykresy miały sens muszą być średnią z kilku niezależnych podejść do gry, wraz z ustalonym odchyleniem standardowym (jak w pierwszym z zadań na przedmiocie). To oznacza, że dla niektórych punktów będzie to średnia (po kilku uruchomieniach gierki) ze średnich (po ostatnich kilkudziesięciu lemingach).
    - Dla wygodnej pracy warto zacząć od najprostszej implementacji funkcji decyzyjnej (losowo 50% lewo, 50% prawo), a następnie przygotować logowanie przebiegu gry i generowanie wykresów na podstawie logów. Dopiero potem warto zabierać się za implementację właściwego uczenia.
- Kiedy diagnostyka jest gotowa, to zabieramy się za właściwą implementację uczenia maszynowego - w następujących wariantach.
    - Wariant bazowy - zachowanie w pełni losowe (wspomniane wyżej).
    - Wariant właściwy - uczenie algorytmem SARSA .
        - Musimy w tym celu zaprojektować system kar i nagród (warto się chwilkę zastanowić, jaki będzie właściwy!).

- Konieczne jest też ustalenie, co będzie tu stanem (i jak go reprezentować), a co akcją.
- Co do samego algorytmu - ściąga poniżej (przyjmijmy na razie bezpieczne parametry - *learning rate* = 0.5, *discount factor* = 0.95, *experiment rate* = 0.05).

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

- Uruchamiamy oba warianty i obserwujemy ich zachowanie z użyciem wcześniejszych wykresów (dla obu przygotowanych plansz).
  - Czy działa - czy stworki zarządzane metodą SARSA zachowują się lepiej niż losowe?
  - Jaką drogą poruszają się stworki pod sam koniec gry? Zwizualizuj przykładowe trasy.

Pora trochę zamieszać

- Na koniec przygotujmy kilka modyfikacji i zobaczmy, jak wpłyną na obserwowane trendy.
  - Zmieńmy pulę dostępnych ruchów z (lewo, prawo) - 2 ruchy, na (lewo, prawo, lewo-dół, dół, prawo-dół, zostań w miejscu) - 6 ruchów.
    - Jak zmieniły się trendy? Większa swoboda ruchów powinna pozwolić na uzyskanie optymalniejszych tras. Z drugiej strony - jest o wiele więcej opcji do rozważenia.
  - Zmieńmy zachowanie podmuchów powietrza z deterministycznego na stochastyczne. Siła przesunięcia do góry za każdym razem może się (losowo) różnić o +- 1. Przykładowo: powiewy o sile dwa mogą podnieść leminga o 1, 2 lub 3 pola - z równą szansą na każdą opcję.
    - Jakie zmiany w trendach teraz obserwujemy?
  - Zmieńmy algorytm z SARSA na Q-Learning (to nic strasznego, algorytmy w wariancie jednokrokowym różnią się dosłownie jedną linijką - patrz poniżej).

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

tu główna różnica

    - Jak teraz zmieniły się trendy (w porównaniu z SARSA)? Jak zmieniły się wybierane przez stworki trasy do celu?

- Wykonajmy dwa zaprojektowane przez siebie eksperymenty. Mogą dotyczyć zmiany jednego z parametrów, zmiany stałego parametru na zanikający w czasie, drobniej zmiany reguł rozgrywki - czegokolwiek.
    - Jak założenia eksperymentu wpłynęły na wyniki? Czy były to zgodne z oczekiwaniem?