RL competition: Super Mario

Gregor Majcen (63070199)

19. januar 2013

1 Uvod

Vsi poznamo igrico *Super Mario* in tudi večina jo zna igrati. Vendar kaj pa računalnik sam? S pomočjo spodbujevalnega učenja je vse mogoče.

2 Algoritem

Kot glavni algoritem sem izbral ne-deterministično **Q-učenje**. S poskušanjem sem prišel do končne enačbe:

$$alpha_n = \frac{1}{\text{število obiskov } (s,a) \text{ do } n\text{-te iteracije}} \tag{1}$$

$$\begin{aligned} Q_n(s,a) &= (1 - alpha_n) * Q_{n-1}(s,a) \\ &+ alpha_n * (R(s,a) + R(s',a') + max_{a'} Q_{n-1}(s',a') - Q_{n-1}(s,a)) \end{aligned} \tag{2}$$

$$\begin{aligned} Q_n(s[end],a) &= (1 - alpha_n) * Q_{n-1}(s,a) \\ &+ alpha_n * (R(s,a) + R(s,a) - Q_{n-1}(s,a)) \end{aligned} \tag{3}$$

3 Stanja

Vsako stanje je določeno v spremenljivki *observation*, vendar je zelo preveliko. Zato sem zmanjšal na manjši kvadrat in sicer relativno od pozicijo mario-ta: 4 gor, 4 dol, 5 naprej in 0 nazaj. Temu kvadratu sem dodal še lastnosti pošasti, ki so v manj kot 3 mesta naprej od mario-ta. To stanje sem poimenoval w

Zaradi močno zmanjšanega kvadrata je možno, da se stanja ponovijo v zelo različnih okoliščinah sem dodal še eno tabelo stanj wof, in sicer: (*int(mario.x), int(mario.y), dy, isMonsterNear)*), kjer $dy = [DEC, STD, INC]$ in $isMonsterNear = [True, False]$ glede na to, ali obstaja 3 mesta naprej od mario-ta kakšna pošast.

4 Akcije

Na voljo je 12 akcij (kombinacija [-1,1], [0-1], [0-1]). Tudi to sem zaradi hitrejšega učenja skrčil na 7 in sicer: [1, 1, 1], [1, 0, 1], [1, 1, 0], [1, 0, 0], [0, 0, 0], [-1, 0, 0], [-1, 1, 0]. Ostale akcije so

se mi zdele odveč, saj jih lahko dosežemo s temi sedmimi. Vse akcije imajo začetno Q oceno 0.

5 Izbiranje akcije

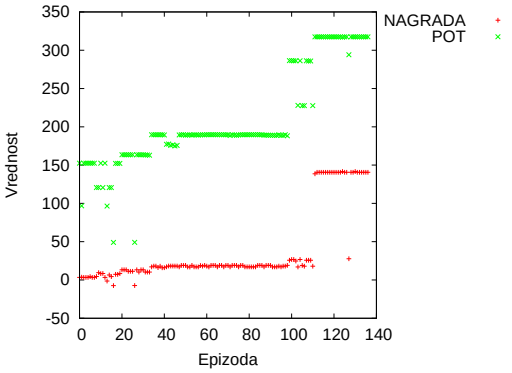
V vsakem koraku najprej preberemo oba stanja w in wof. Nato najprej pogledamo našo Q tabelo:

- wof obstaja: Poiščemo $argmax_a(Q(wof, a)$, ki je naša zmagovalna akcija. Poglej še, če obstaja w, in če ne ga ustvarimo.
- wof ne obstaja in w obstaja: Poiščemo $argmax_a(Q(w, a)$, ki je naša zmagovalna akcija. Nato ocene w prepišemo v wof.
- wof in w ne obstajata: Ustvarimo w in wof in izberemo prvo možno akcijo.

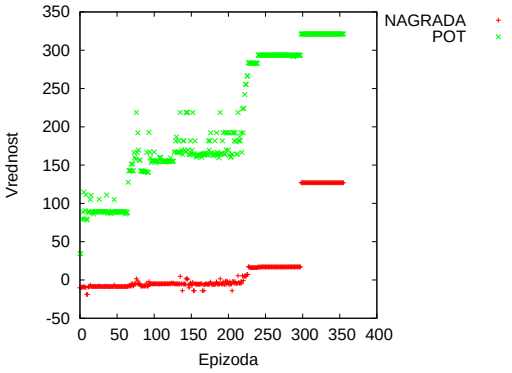
6 Dodatne zanimivosti

1. Ker želim, da mario čim prej konča, mu pri vsakem premiku naprej prištejem 0.01 (toliko kot izgubi zaradi narejenega koraka).
2. Če se mario zaletava v zid ali kaj podobnega (da sta si dve sosednji stanji enaki): Q(s,a) odštejem 5 (kar močno kaznujem in s tem preprečim, da bi po nepotreben zapravljal čas)
3. Ko mario konča, se pravi se ubije ali pa zmaga, hočem nagraditi njegovo dosedanje delo. Vsem Q(s,a), ki so bile izvajane do $mario.x.end - 10$ prištejem 10, vendar le v primeru, če je prišel dlje kot kadarkoli do sedaj.
4. Ker python funkcija max vzame prvi element, ki je največji, to tudi izkoriščam. Če ima več akcij isto maksimalno oceno, vzame tisto, ki je prva (glej poglavje Akcije).

7 Rezultati



(a) Rezultati druge težavnosti



(b) Rezultati pete težavnosti

8 Izjava o izdelavi domače naloge

Domačo nalogo in pripadajoče programe sem izdelal sam.