

Veštačka inteligencija, praktični ispit SEPTEMBAR1

Matematički fakultet

Školska godina 2019/2020

Napomena: Na Desktop-u se nalazi direktorijum `vi.sep1`. Preimenujte ga u oblik `Ime_Prezime_BrojIndeksa_GodinaUpisa`. Na primer za indeks 283/2016 student Pera Perić treba direktorijum `vi.sep1` preimenovati u `Pera_Peric_283_2016`. U ovom direktorijumu ostavite Vaše rešenje. Na Desktop-u možete pronaći dokumentaciju za potrebne Python biblioteke.

1. Data je kvadratna mapa dimenzije 10×10 po kojoj se kreće figura. Figura se kreće kao konj po šahovskoj tabli. Svako polje ima neku vrednost (cenu). Potrebno je naći najjeftiniji put za figuru koja kreće od polja $[0, 0]$ i treba da stigne do polja $[9, 9]$. Korišćenjem A* naći najjeftiniji put. Sve vrednosti na mapi su pozitivni nenula brojevi. Definisati (smisliti) dopustivu heuristiku (i kratko je obrazložiti u komentaru iznad funkcije). Deo implementacije je već dat u datoteci `zadatak1.ipynb`. Dopuniti implementaciju.
2. Korišćenjem MINISAT rešavača naći sve moguće valuacije koje zadovoljavaju formulu:

$$[(\neg a \vee \neg b) \wedge (b \vee \neg a)] \Rightarrow a$$

Ulazne datoteke korišćene prilikom pokretanja minisat rešavača nazvati redom: `minisat1.txt`, `minisat2.txt`, `minisat3.txt`, itd. Ispisati i obrazložiti dobijene izlaze u datotekama `minisat_izlaz1.txt`, `minisat_izlaz2.txt`, `minisat_izlaz3.txt`, itd.

3. Napraviti agenta koji algoritmom tabelarnog Q učenja pokušava da maksimizuje svoju nagradu u okruženju NChain-v0. Okruženje NChain-v0 predstavlja simulaciju kretanja po lancu, pri čemu su dozvoljene dve akcije:
 - Akcija 0: Kretanje unapred po lancu, bez nagrade
 - Akcija 1: Restart u stanje 0, dobija se nagrada vrednosti 2

Ako se dostigne kraj lanca, dobija se nagrada u vrednosti 10, a agentu je tu dozvoljeno da nastavi kretanje unapred (biće vraćen u to isto stanje), i dobiće ponovo nagradu 10.

Prilikom preduzimanja akcije, postoji mala verovatnoća da se agent oklizne, i da bude učinjena suprotna akcija od zahtevane.

- (a) Napraviti okruženje NChain-v0 kroz biblioteku `gym`
- (b) U funkciji `get_action` implementirati odabir akcije na epsilon pohlepan način
- (c) U funkciji `update_q_table` implementirati ažuriranje q tabele
- (d) U funkciji `update_epsilon` implementirati ažuriranje epsilon vrednosti po formuli date u nastavku
- (e) Implementirati neophodne korake u algoritmu Q učenja u okviru funkcije `q_learning`
- (f) Nacrtati kako se kroz epizode menjala nagrada koju je ostvarivao agent (označiti šta X i Y osa prikazuju)
- (g) Nacrtati kako se kroz epizode menjala vrednost za epsilon (označiti šta X i Y osa prikazuju)

Epsilon ažurirati po formuli:

$$\epsilon = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min} \cdot e^{-\alpha \cdot i})$$

pri čemu je:

- ϵ_{min} - minimalno dozvoljeno epsilon
- ϵ_{max} - maksimalno dozvoljeno epsilon
- α - faktor umanjenja za epsilon
- i - indeks epizode

Koristiti postavku `zadatak3.py`.