

Mašinsko učenje

Drugi domaći zadatak

19.5.2024.

Drugi domaći zadatak sastoji se od tri nezavisna problema, opisana na narednim stranama. Uz svaki problem tj. njegov deo stoji odgovarajući broj poena, a ukupan broj poena je 100 (što se kasnije skalira na 25 predispitnih poena).

Pri izradi domaćeg zadatka moguća je saradnja studenata u grupama od najviše troje. Pritom, svaki student će biti nezavisno ocenjivan na osnovu odbrane pri kojoj se proverava razumevanje predatog rešenja i relevantnog gradiva. Iako je podela posla dozvoljena, ukoliko jedan član tima ne razume neke delove predatog rešenja, za te delove će mu biti dodeljeno 0 poena. Ukoliko postoji deo predatog rešenja koji niko iz tima ne razume, svim članovima tima će biti dodeljeno 0 poena na celom domaćem zadatku.

Domaći zadatak se izrađuje i predaje isključivo na sledeći način:

1. Downloadovanje arhive **ml_d2_x_y_z.zip** koja se nalazi uz ovaj dokument. U ovoj arhivi se nalaze svi potrebni podaci.
2. Rešenja sačuvati u odgovarajućim direktorijumima u skladu sa zahtevima datim u problemima. Dozvoljena je upotreba Google Colab platforme. U tom slučaju se umesto .py fileova šalje download-ovan colab notebook. Obavezno sačuvati izlaz izvršavanja programa (ili kao tekstualni file ili u colabu) i po potrebi treniran model (samo jedan koji predstavlja rešenje). Nemojte slati data setove.
3. Zapakivanje foldera **ml_d2_x_y_z** u arhivu, pri čemu treba zameniti slova x/y/z brojevima indeksa (u formatu RN-br-god) članova tima.
4. Slanje fajla **ml_d2_indeks1_indeks2_indeks3.zip** na mejl adrese ymatovic@raf.rs ili iciganovic@raf.rs asistentu kod kojeg po rasporedu slušate vežbe, pre isteka roka. Subject mejla mora biti u obliku "[ML D2] prezime1 prezime2 prezime3". U tekstu mejla obavezno navesti članove tima sa brojevima indeksa.

Rok za slanje rešenja je nedelja 2. jun u 23:59.

Odbrana domaćeg zadatka će biti u terminu poslednjih vežbi pre kolokvijumske nedelje (4.6.2024.).

Na narednim stranama nalazi se opis problema sa jasnim smernicama koje fajlove treba popuniti i šta njihovo pokretanje treba da da kao izlaz. Svi problemi su urađivi korišćenjem znanja sa časova i uz malo samostalnog istraživanja. Naravno, dozvoljeno je koristiti kod sa vežbi (dokle god shvatate šta on zapravo radi), ali nije dozvoljeno koristiti kompletna rešenja direktno kopirana sa interneta. U slučaju da ima pitanja/nedoumica pošaljite mejl na ymatovic@raf.rs ili iciganovic@raf.rs a možemo organizovati i konsultacije po dogovoru.

Problem 1: Stabla odlučivanja [30p]

U arhivi se nalazi skup podataka **crop.csv**. Detaljan opis podataka i cilj istraživanja možete naći [ovde](#).

Potrebno je istrenirati model stabla odlučivanja da preporuči seme za sejanje na osnovu podataka o zemljištu. Dozvoljeno je i očekivano po potrebi srediti podatke. Dozvoljeno je koristiti ugrađene klasifikatore koje smo koristili na času.

Problem 2: Neuralna Mreža [30p]

Na istom data setu iz prethodnog zadatka potrebno je istrenirati model neuralne mreže da preporuči seme isto kao u prethodnom zadatku. Neuralnu mrežu implementirati manuelno, u formi matrica, a ne koristeći Keras biblioteku. Dozvoljeno je i očekivano po potrebi srediti podatke.

Problem 3: Behavioral Cloning [40p]

Istrenirati neuralnu mrežu upotrebom Keras biblioteke da rešava **MiniGrid-Empty-Random-6x6-v0** okruženje. Detalje o okruženju možete naći [ovde](#). Sve potrebne biblioteke se nalaze u **requirements.txt** fajlu. Zadatak se rešava u tri koraka.

Na početku je potrebno pokrenuti **record.py** koji služi da se sačuvaju trening podaci. Biće prikazan prozor u kojem treba ručno rešiti okruženje. Komande možete naći (i promeniti) u samom fajlu. Nije potrebno menjati kod ovog fajla. Svaka komanda koju zadate će biti sačuvana zajedno sa stanjem okruženja u tom trenutku. Kada rešite okruženje ono će se resetovati i možete nastaviti sa snimanjem. Snimanje se završava pritiskom na taster **Esc** i u tom trenutku će biti sačuvani fajlovi sa stanjima i akcijama. Prilikom ponovnog pokretanja snimanja stari podaci će biti prepisani.

Zatim treba pokrenuti fajl **training.py** koji učitava podatke iz prethodnog koraka, izvršava trening i na kraju čuva treniran model. Učitavanje podataka je obezbeđeno. Sve ostalo treba sami da implementirate. Dimenzije izlaza modela možete naći u dokumentaciji okruženja.

Na kraju, u fajlu **test.py** se nalazi kod za testiranje treniranog modela. Prvo je potrebno implementirati učitavanje modela sačuvanog u prethodnom koraku. Zatim izvršiti skriptu. Prilikom izvršavanja otvoriće se prozor gde će biti prikazano kako vaš model rešava okruženje. Na kraju će biti prikazana prosečna vrednost rewarda u 10 uzastopnih testova. Očekivno je da reward bude veći od 0.6.

Snimanje podataka se mora izvršiti na lokalnom računaru zbog prikazivanja prozora. Trening je moguće izvršiti na Colabu. Testiranje se mora izvršiti lokalno ukoliko želite da prikazete prozor i vidite šta agent radi. Ukoliko ne možete pokrenuti TF lokalno testiranje se može takođe pokrenuti na Colabu ali je neophodno postaviti flag **LOCAL** na False.