

## Veštačka inteligencija, Ispit, Primer ispita

Matematički fakultet

Školska godina 2019/2020

**Napomena:** Pokrenite konzolu i ukucajte `pocni_ispit`. Skript će Vam tražiti da unesete vaš broj indeksa u obliku `miGGBBB`. Na primer za indeks 283/2015 treba uneti `mi15283`. Skript će Vam napraviti direktorijum na osnovu unete niske gde možete pronaći šablone i podatke za rad.

Na **Desktop**-u možete pronaći dokumentaciju za potrebne Python biblioteke.

Vreme za rad: 2 sata

1. *Nim* je igra kinsekog porekla u kojoj najčešće učestvuju dva igrača i imaju tri hipa sa različitim brojem kuglica. Igrači igraju naizmenično i u jednom potezu mogu skinuti tačno sa jednog hipa bilo koji broj kuglica. Cilj je biti poslednji igrač koji može da skine kuglice.



Slika 1: Igra Nim

Primer igre dva igrača (prvi na potezu je Bojan):

Hip A	Hip B	Hip C	Potezi
3	4	5	Bojan skida 2 kuglice sa A
1	4	5	Ana skida 3 sa C
1	4	2	Bojan skida 1 sa B
1	3	2	Ana skida 1 sa C
1	2	2	Bojan skida sve sa A
0	2	2	Ana skida 1 sa B
0	1	2	Bojan skida 1 sa C
0	1	1	Ana skida 1 sa B
0	0	1	Bojan skida 1 sa C, to je poslednji potez i on je pobednik

Napisati minimax implementaciju igre *nim*.

2. (a) (5 poena) Korišćenjem logaritamskog kodiranja zapisati sledeće uslove u iskaznoj logici:  
 $a, b, c, d \in \{0, 1, 2, 3\}$   
 $a \neq c$   
 $b + d \leq 3$   
(b) (5 poena) Korišćenjem DPLL algoritma naći valuaciju koja zadovoljava sledeću formulu ukoliko takva valuacija postoji.

$$(p \Rightarrow \neg q) \wedge (\neg p \vee r) \wedge (q \Rightarrow r) \wedge (r \Rightarrow q) \wedge (q \vee r) \wedge (q \Rightarrow \neg r)$$

Rešenje ostaviti u direktorijumu `sablon/2_logika` u tekstuelnoj datoteci sa nazivom `resenje.txt`.

Za simbole u tekstualnoj datoteci koristiti:

- **A** - univerzalni kvantifikator
  - **E** - egzistencijalni kvantifikator
  - **&** - konjunkcija
  - **|** - disjunkcija
  - **=>** - implikacija
  - **<=>** - ekvivalencija
3. (a) Učitati `cifar10` podatke. Kao podatke za obučavanje uzeti prvih 1000 podataka iz učitanih podataka za obučavanje, a kao podatke za testiranje uzeti prvih 100 podataka iz podataka za testiranje.  
(b) Odabrati nasumično 9 različitih instanci iz skupa za obučavanje i napraviti crtež dimenzija  $3 \times 3$  na kojima ih je potrebno prikazati.  
(c) Izvršiti normalizaciju piksela i enkodiranje ciljne promenljive.

- (d) Napraviti konvolutivnu neuronsku mrežu sa sledećom arhitekturom:
- Konvolutivni sloj sa 32 filtera, ReLu aktivacijom i veličinom kernela  $3 \times 3$
  - Konvolutivni sloj sa 32 filtera, ReLu aktivacijom i veličinom kernela  $3 \times 3$
  - Agregirajući sloj sa veličinom prozora  $2 \times 2$
  - Regularizacija izostavljanja (Dropout) sa parametrom 0.25
  - Konvolutivni sloj sa 64 filtera, ReLu aktivacijom i veličinom kernela  $3 \times 3$
  - Konvolutivni sloj sa 64 filtera, ReLu aktivacijom i veličinom kernela  $3 \times 3$
  - Agregirajući sloj sa veličinom prozora  $2 \times 2$
  - Regularizacija izostavljanja (Dropout) sa parametrom 0.25
  - Sloj koji izravna dimenzije (Flatten)
  - Potpuno povezani sloj sa 512 neurona i ReLu aktivacijom
  - Potpuno povezani sloj sa brojem neurona koliko postoji klasa i aktivacijom mekog maksimuma
- (e) Izvršiti obučavanje neuronske mreže. Kao broj paketa pri obučavanju postaviti 32, za broj epoha uzeti 10. Podatke za testiranje postaviti kao validacione podatke.
- (f) Nacrtati grafik koji prikazuje kako se menja tačnost (accuracy) na podacima za obučavanje i testiranje tokom obučavanja mreže. Nacrtati grafik koji prikazuje kako se menja greška na oba skupa prilikom obučavanja mreže. Na  $x$  osi označiti epoha, a na  $y$  osi označiti odgovarajuću meru. Na oba grafika dodati legendu koja prikazuje šta koja funkcija označava (odnosno trening i test).

Koristiti šablon `sablon/zadatak04.ipynb`