

## Teme la alegere Invatare Automata (T4)

Pentru tema T4, va puteti alege oricare din temele de mai jos. Tema se poate rezolva in LISP, SCHEME, PROLOG, C/C++ sau Java, la alegerea dumneavoastra. Pentru cerinte suplimentare asupra redactarii/implementarii/predarii se va consulta cadrul didactic de la laborator.

Fiecare student se va inscrie cu optiunea temei alese la cadrul didactic de la laboratorul lui/ei, pana pe data de 16 martie (2009) cel mai tarziu. Studentii care nu se inscriu pana pe 16 martie NU mai pot prezenta tema de casa.

### TEMA Nr. 1

Problema calugarilor (the MONK's problem) este una dintre problemele clasice de testare a performantelor unui algorithm de invatare. Problema, de fapt un set de 3 probleme inrudite (M1, M2, M3), se refera la o lume de roboti in care acestia sunt descrisi de 6 atribute diferite [Wnek, Sarma, Wahab and Michalski, 1991]:

x1: head\_shape  $\in$  round, square, octagon  
x2: body\_shape  $\in$  round, square, octagon  
x3: is\_smiling  $\in$  yes, no  
x4: holding  $\in$  sword, balloon, flag  
x5: jacket\_color  $\in$  red, yellow, green, blue  
x6: has\_tie  $\in$  yes, no

In toate cele 3 cazuri este vorba de o invatare din exemple (inductiva) care trebuie sa conduca la o clasificare. Fiecare din cele 3 probleme are o descriere logica a clasei; un robot particular (instanta) apartine sau nu clasei respective.

Din cei 432 de roboti posibili, numai o submultime de roboti este selectata ca multime de exemple de invatare (training examples), pentru fiecare robot din aceasta multime avand asociata clasificarea. Scopul programului este de a invata din aceste exemple prin generalizare si de a obtine o descriere simbolica a generalizarii invatate (a clasei obtinute prin invatare). Se va utiliza un algorithm de clasificare bazat pe arbori de decizie (ID3) cu optimizari (arbore minim), se va termina clasificarea pentru fiecare din cele 3 probleme, se va genera din program o descriere simbolica a clasei (pentru fiecare din cele 3 probleme) sub forma de reguli si se vor interpreta rezultatele.

Problemele sunt:

- Problema M1:

**(head shape = body shape) or (jacket color = red)**

Din 432 de roboti se selecteaza, aleator, din setul de invatare 124 de roboti, toate clasificarile din setul de invatare fiind corecte.

- Problema M2:

**Exact 2 din cele 6 atribute au ca valoare prima valoare din lista de valori**

(de exemplu: body shape = head shape = round inseamna is\_smiling  $\neq$  yes, holding  $\neq$  sword, jacket\_color  $\neq$  red si has\_tie  $\neq$  yes)

Din 432 de roboti se selecteaza, aleator, din setul de invatare 169 de roboti, toate clasificarile din setul de invatare fiind corecte.

- Problema M3:

**(jacket color is green and holding a sword) or (jacket color is not blue and body shape is not octagon)**

Din 432 de roboti se selecteaza, aleator, din setul de invatare 122 de roboti, din care 5% au clasificarile din setul de invatare incorecte.

### TEMA Nr. 2

Sa se implementeze o retea neurala de tip BackPropagation (sau variatii) care sa recunoasca conturi de dreptunghiuri sau cercuri intr-o imagine binara (alb/negru). Functia de transfer este functia sigmoida.

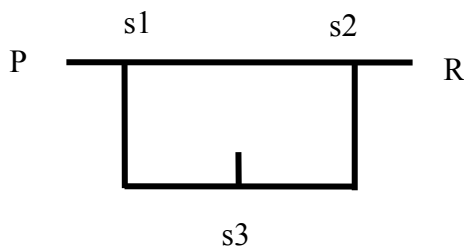
Scalati valorile elementelor sabloanelor a.i. sa se incadreze intr-o dimensiune fixa (nr fix de pixeli) care va reprezenta intrarea retelei. Antrenati reteaua pe baza a 2/3 din nr. de sabloane, afisand curba de erori. Evaluati performantele retelei, mai intai pe sabloanele de antrenament, apoi pe cele ramase ("sabloane de test").

Obs: Alegerea dimensiunii retelei se face prin "trial & error".

Indicatie: Sunt necesare cel mult 2 straturi ascunse, iar nr. de noduri/strat ascuns trebuie sa fie mai mic decat nr. sabloane de antrenament, altfel reteaua ajunge sa "memoreze" sabloanele, pierzand din capacitatea de generalizare.

### TEMA Nr. 3

Un agent se misca intr-un spatiu care este format din mai multe coridoare legate in trei puncte de intersectie in forma de T, etichetate s1, s2, si s3.



Intersectiile s1 si s2 au coridoare la est, sud si vest; intersectia s3 are coridoare la est, nord si vest. In fiecare intersectie agentul are 2 actiuni posibile: la stanga sau la dreapta. Actiunile sunt deterministe. Daca un agent ajunge la un perete (deadend) este teleportat instantaneu in intersectia s3 si pozitionat cu fata la sud. Astfel, singurele alegeri posibile ale agentului sunt atunci cand ajunge in una din cele 3 intersectii si este cu fata la peretele intersectiei in T. Din punct de vedere al unui algorithm de invatare prin recompensa, cele 3 puncte de selectie sunt cele 3 stari ale mediului. Matricea cu trazitiile de stari ale agentului este data in tabela 1.

Recompensa imediata este data in tabela 2. Recompensa imediata poate fi interpretata astfel: o cantitate negativa de recompensa pentru fiecare metru de coridor parcurs de la ultimul punct de selectie plus o recompensa data de teleportare (R si P).

Tabela 1. Matricea de tranzitie a agentului.

Actiune Stanga	s1	s2	s3		Actiune Dreapta	s1	s2	s3
s1	0	0	1		s1	0	0	1
s2	0	0	1		s2	0	0	1
s3	0	1	0		s3	1	0	0

Tabela 2. Recompensa imediata

r	Actiune Stanga	Actiune Dreapta
s1	-1.0	+0.7
s2	-1.3	+1.0
s3	-0.7	-0.5

Sa se scrie un program care:

- determina perechile de valori stare-actiune optime (Q-learning) si le afiseaza;
- extrage o politica optima din perechile stare-actiune si o afiseaza astfel:

Stare lume | Actiune selectata cf politicii optime

s1	
s2	
s3	

#### TEMA Nr. 4

Se va implementa un consultant autonom al unui fermier care trebuie sa-si conduca optim afacerea - o ferma de bovine. Fermierul are ca scop sa obtina cat mai multi bani. Este vorba de un management continuu, in care bovinele se reproduc si cireada trebuie mentinuta timp indelungat (pentru ca fermierul sa evite criza financiara). Singurele actiuni pe care le poate face fermierul este sa vanda bovine de diferite tipuri. Dimensiunea maxima a cirezii este  $H=12$ . Fiecare bovina poate sa fie de 3 tipuri: *tanara*, *matura* si *batrana*. In timpul fiecarei epoci, fiecare tip de bovina aduce o utilitate estimata  $r$  cf. tabelului 1. In plus, fiecare bovina de tip  $i$  poate deveni de tip  $j$  cu o probabilitate  $p_{ij}$  cf. tabelului 2. Viteii (bovine *tinere*) sunt produci cf. distributiei de probabilitate din tabelul 3. Numai bovinele *mature* pot produce vitei. Cireada nu poate creste la o dimensiune mai mare de  $H$ , ceea ce inseamna ca se pot produce vite numai daca cireada scade sub  $H$ .

Starea sistemului poate fi reprezentata ca un vector tri-dimensional

$$s = (s_{tanara}, s_{matura}, s_{batrana})$$

La fiecare pas, fermierul trebuie sa decida cate bovine din fiecare tip sa vanda. Actiunea la fiecare pas este deci reprezentata printr-un vector tri-dimensional:

$$a = (a_{tanara}, a_{matura}, a_{batrana}), \text{ unde } a_i \text{ este numarul de bovine care se vinde.}$$

Restrictia este  $0 \leq a_i \leq s_i$  pentru toate tipurile  $i$ . Daca decizia este sa se vanda 1 bovina de tip  $i$  atunci plata este  $c_i$ , cf. Tabelului 1. Vanzarea a  $n$  bovine va produce  $n \cdot c_i$ .

Se se implementeze un program care:

- determina politica optima pe baza de iterare a valorii (value iteration) si traseaza graficul utilitatilor in functie de stari. Selectati factorul de atenuare  $\delta = 0.9$
- determina politica optima pe baza de iterare a politicii (policy iteration) si o reprezinta grafic. Selectati factorul de atenuare  $\delta = 0.9$

Tabela 1

Tip bovina	tanara	matura	batrana
utilitatea asteptata $r$	0.3	0.4	0.2
plata asteptata $c$	2	6	4

Tabela 2: probabilitati de tranzitie  $p_{ij}$ 

i / j	tanara	matura	batrana
tanara	0.9	0.1	0
matura	0	0.75	0.25
batrana	0	0.15	.85

Tabela 3: Probabilitatile de producere vitei

Tip bovina / Nr vitei produci	0	1	2
matura	0.05	0.8	0.15

**TEMA Nr. 5**

Sa se implementeze o lume simulata bazata pe un algoritm genetic. Lumea se afla pe un grid in care sunt pasari si plante. Pasarea mananca planta daca a ajuns in patratul in care este planta. Numarul de plante este fix (generat la inceput). Initial pasarile sunt generate aleator. Pasarile au o stare interna reprezentata printr-un numar de la 0 la 15 (starea pasarii) si posibilitatile de miscare: deplasare in fata 1 patrat, deplasare in spate 1 patrat, rotire la stanga cu 90 grade, rotire la dreapta cu 90 de grade. Daca pasarea ajunge intr-un patrat cu planta, manca planta si castiga 1 punct (planta mancata apare in alt patrat pe grid). Fitnessul unei pasari este nr. puncte + 1. Fiecare pasare are asociate regulile de comportare (cromozomul de reprezentare al pasarii): 48 de reguli (pt fiecare combinatie a uneia din cele 16 stari cu ceea ce vede in patratul din fata: gol, planta sau zid). Fiecare regula specifica o actiune si noua stare a ratei.

La fiecare generatie pasarile se evalueaza din punct de vedere al fitnessului (cele din generatia curenta dispar si sunt inlocuite de descendentii) si se aplica un algorithm genetic care selecteaza pasarile cele mai potrivite (indicati schema de selectie) care apoi produc descendentii ce sunt mutati cu o anumita probabilitate (specificati schema de crossover si mutatie).

Implementati evolutia lumii, inclusiv grafic (plante si pasari pe grid).

**TEMA Nr. 6**

Sa se implementeze o retea neurala de tip BackPropagation (sau variatii) care sa recunoasca cifra 0 intr-o imagine binara (alb/negru). Se considera ca toate imaginile de test au aceeasi dimensiune fixa (numar de pixeli) care va reprezenta intrarea retelei. Imaginile pot fi afectate de zgomote. Antrenati retea pe baza a 2/3 din nr. de sabloane, afisand curba de erori. Evaluati performantele retelei, mai intai pe sabloanele de antrenament, apoi pe cele ramase ("sabloane de test").

**TEMA Nr. 7**

Sa se implementeze deplasarea unui robot intr-un labirint bazat pe un algoritm genetic. Robotul se poate deplasa N, S, E, V, comportamentul sau fiind determinat de lista miscarilor posibile in labirint. Labirintul are o singura intrare (de unde robotul isi incepe miscarea) si o singura iesire (unde va trebui sa ajunga robotul). Cu cat robotul va ajunge mai aproape de iesire, intr-un numar mai mic de miscari cu atat scorul sau va fi mai bun. Implementati miscarea robotului in labirint (inclusiv grafic).