

:: МЕТОДЕ ОДЛУЧИВАЊА ::

ПЛ-5 .: Лаб. вежба #5 (Перформансе претраживања и функције одлучивања)

У циљу представљања основних могућности вештачких неуронских мрежа као парадигме вештачке интелигенције покушаћемо да моделирамо присуство препреке на путањи током кретања. У том смислу, лабораторијска вежба је базирана на:

1. Функционалној апроксимацији,
2. Класификацији и
3. Предикцији.

као основним примерима развоја и имплементације.

1. *Функционална апроксимација* представља домен примене вештачких неуронских мрежа у циљу апроксимације неке одређене функције. Пример (само један од многих) примене функционалне апроксимације је апроксимација трајекторије мобилног робота током кретања. Трајекторија је углавном дефинисана као функција реалних аргумената, тако да се праћењем ове трајекторије обезбеђује испуњавање циља.

2. *Класификација* је једна од основних примена (система!) вештачких неуронских мрежа у коме се од система очекује да класификује (раздвоји) одговарајуће податке на основу усвојеног критеријума у одређене класе. У домену развоја мобилних робота примери класификације су многобројни, нпр. *присуство препреке* (постоји / не постоји / недефинисано), *изглед окружења* (ходник / раскрсница / зид са леве стране / зид са десне стране итд.) => пројектни задатак.

3. На крају, *предикција* је примена вештачких неуронских мрежа у циљу предвиђања вредности неке функције у следећем тренутку. Један од примера је предвиђање кретања мобилног робота на основу усвојеног механичког модела у кораку *предикције* Калмановог филтера.

Основни циљ лабораторијске вежбе

На основу спроведених мерења у експерименталном процесу потребно је применом система вештачких неуронских мрежа урадити следеће:

1. Применом *Matlab* софтверског пакета извршити класификацију улазних података у циљу одређивања присуства евентуалне препреке; користити моделе вештачких неуронских мрежа са простирањем сигнала унапред;
2. У *BPnet* и *Matlab* окружењу извршити предикцију читавања са сонара на основу претходног мерења.

Пример кода за вршење класификације:

```
%=====
% VNM koja modelira prisustvo prepreke;
% za vrednosti manje od 60 [cm] => prepreka. Ako su vrednosti vece od 60
% [cm] nema prepreke. U ovom primeru u obzir uzimamo ocitavanje jednog i samo
% jednog sonara.

P = [d601_0(:,3);d601_30(:,3);d602_minus30(:,3)]

o1 = ones(size(P,1),1)
o2 = zeros(size(P,1),1)
%
P1 = [75 79 87 70 71 75 84 86 95 87 82 73 96 120 118 93 89 88 98 94 ...
```

:: МЕТОДЕ ОДЛУЧИВАЊА ::

ПЛ-5 .: Лаб. вежба #5 (Перформансе претраживања и функције одлучивања)

```
75 79 87 70 71 75 84 86 95 87 82 73 96 120 118 93 89 88 98 94 ...
75 79 87 70 75 79 87 70 71 75 84 86 95 87 82 73 96 120 118 93 89 88 98 94
...
75 79 87 70 71 75 84 86 95 87 82 73 96 120 118 93 89 88 98 94 ...
75 79 87 70 75 79 87 70 71 75 84 86 95 87 82 73 96 120 118 93 89 88 98 94
...
75 79 87 70 71 75 84 86 95 87 82 73 96 120 118 93 89 88 98 94 ...
75 79 87]'

i = [P;P1]'
o = [o1;o2]'

net1= newff(i,o,[10],{},'trainlm','learngdm')

net1 = train(net1,i,o)
net1.trainParam.show = 50;
net1.trainParam.lr = 0.05;
net1.trainParam.epochs = 100;
net1.trainParam.goal = 1e-5;
net1.trainParam.max_fail = 5;

y1 = sim(net1,P1')
y2 = sim(net1,d603_0(:,3)')
y3 = sim(net1,d403_0(:,3)')

%=====
% Slicno prethodnom s tim sto se u sistem uvodi očitavanje drugog sonara, kao i
% očitavanje enkodera.

P = [d601_0; d603_30; d602_minus30]

o1 = ones(size(P,1),1)
o2 = zeros(size(P,1),1)

P1 = [P(:,1:2) P1(1:end-1) P1(1:end-1)]
i = [P;P1]'
o = [o1;o2]'

net2= newff(i,o,[10],{},'trainlm','learngdm')

net2 = train(net2,i,o)
net2.trainParam.show = 50;
net2.trainParam.lr = 0.05;
net2.trainParam.epochs = 100;
net2.trainParam.goal = 1e-5;
net2.trainParam.max_fail = 10;

y1 = sim(net2,d403_0')
```

Потребно је развити модел вештачке неуронске мреже која ће бити у стању да препозна постојање препреке. Развијени модел имплементирати на роботу у реалном времену.

:: МЕТОДЕ ОДЛУЧИВАЊА ::

ПЛ-5 .: Лаб. вежба #5 (Перформансе претраживања и функције одлучивања)

Предикција мерења сонара на основу претходног мерења. Матлаб код је приказан у наставку.

```
% predikcija
Pi = P(:, [3 4])
P2 = [P([2:end], [3 4])]
P2 = [P2;P(end, [3 4])]
i = Pi'
o = [P2]'
net3 = newff(i,o,[5],{},'trainlm','learngdm')
net3 = train(net3,i,o)

y3 = sim(net3,i)

[y3' Pi]

% predikcija
Pi = P(:, [3 4])
P2 = [P([2:end], [3 4])]
P2 = [P2;P(end, [3 4])]
i = Pi'
o = [P2]'

net4 = newff(i,o,[5],{},'trainlm','learngdm')
net4 = train(net4,i,o)

y3 = sim(net4,i)

[y3' Pi]

% predikcija broj dva
% na osnovu merenja za SetPower = 60 i ugao od nula stepeni:
Pi1 = d601_0(:, [3 4])
P3 = [Pi1(2:end,:);d601_0(end, [3 4])]
i = Pi1'
o = [P3]'
net5 = newff(i,o,[5],{},'trainlm','learngdm')
net5 = train(net5,i,o)

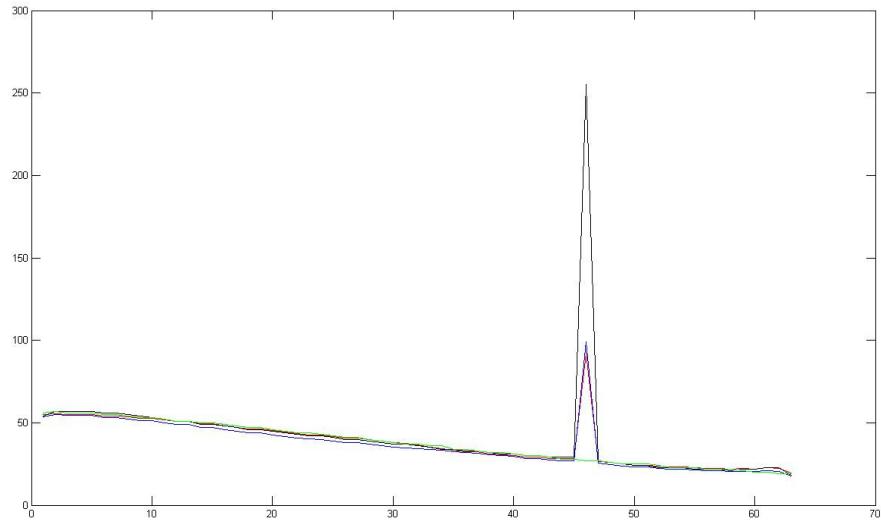
y3 = sim(net5,i)

[y3' Pi1]
```

Моделирање и симулацију урадити у Матлаб окружењу. Резултати предикције мреже бр. 5, као једне од тестираних и имплементираних мрежа приказани су на слици број један.

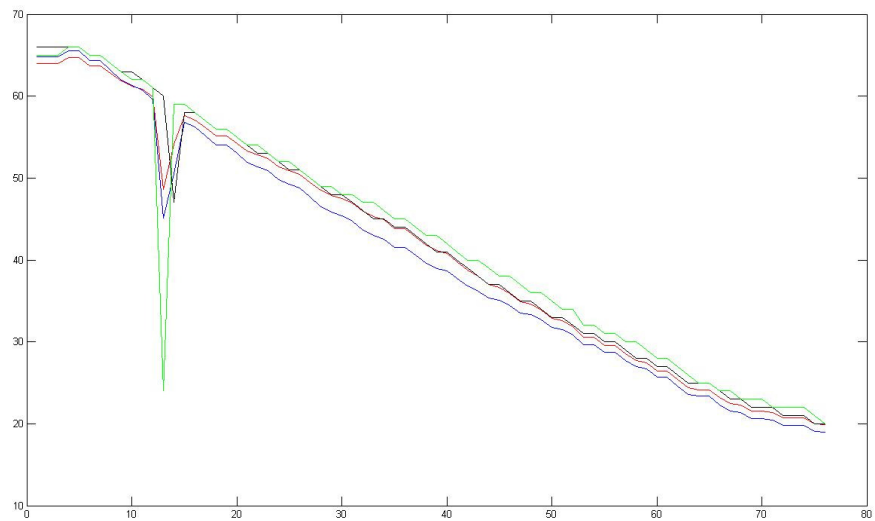
:: МЕТОДЕ ОДЛУЧИВАЊА ::

ПЛ-5 .: Лаб. вежба #5 (Перформансе претраживања и функције одлучивања)



Слика 1: Верификација обучености мреже у on-line режиму.

Дакле, на основу претходног очитавања сонара, мрежа је генерисала вредности које су са (за сада) задовољавајућом тачношћу предвиделе „скок“ (грешку).



Слика 2: Верификација обучености мреже у on-line режиму.