ПЛ-2 .: Лаб. вежба #2 (***Matlab*** апликативни софтвер и робот ***LEGO Mindstorms NXT***)

Увод у МАТЛАБ

1. Основне информације

Матлаб представља интерактивно окружење за напредна израчунавања и рад са матрицама за потребе моделирања инжењерских проблема. Само име је акроним енглеског термина **Mat**rix **Lab**oratory - MATLAB. Матлаб поседује велики број додатних софтверских модула који су наменски развијени за одговарајуће потребе корисника (тзв. *Toolboxes*). У оквиру ових додатних софтверских модула могуће је моделирати инжењерске проблеме применом вештачке интелигенције (*Neural Network Toolbox*, *Fuzzy Logic Toolbox*), вршити аквизицију података са одговарајућих давача (*Data Acquisition Toolbox*), моделирати сигнале (*Signal Processing Toolbox*), вршити симболичка израчунавања (*Symbolic Toolbox*) итд.

У Матлабу је могуће користити готове функције и на тај начин моделирати и одредити решење изучаваног проблема. С друге стране, основни разлог због кога је Матлаб „занимљив" научницима, студентима и инжењерима у пракси је могућност коју Матлаб пружа у погледу директног програмирања и накнадне имплементације кода у домену рада реалног система без потребе за „спуштањем" на ниво машинског језика. Наиме, с обзиром да је Матлаб писан у *C* језику, који представља језик вишег нивоа и који може директно да се инсталира на одговарајућу управљачку јединицу. Из тог разлога Матлаб представља моћан апарат у рукама обучених и искусних корисника.

2. Дефинисање матрица и вектора

Рад у Матлаб окружењу је једноставан. Наиме, само (програмско) окружење је развијено с основним циљем да се кориснику што више олакша и поједностави коришћење Матлаба. У наставку ћемо приказати како се у овом окружењу дефинишу матрице и вектори, које су основне величине са којима софтвер манипулише.

Матрица се дефинише на следећи начин:

```
>> A=[1 2 3 4; 5 6 7 8]

A =

     1     2     3     4
     5     6     7     8
```

Дакле, обратити пажњу да се помоћу оператора „;" (тачка-запета) одређује број врста матрице. Другим речима, када се заврши унос свих елемента одговарајуће врсте са ; се та информација и проследи Матлабу.

Транспонована матрица се одређује на следећи начин:

```
>> A'

ans =

     1     5
     2     6
     3     7
     4     8
```

Или еквивалантно,

```
>> A=[1 2 3 4; 5 6 7 8]'

A =

     1     5
     2     6
     3     7
     4     8
```

Инверзна матрица је:

```
A=[1 2;3 4]

A =

     1     2
     3     4

>> A^-1

ans =

   -2.0000    1.0000
    1.5000   -0.5000

>> inv(A)

ans =

   -2.0000    1.0000
    1.5000   -0.5000
```

Дакле, инверзна матрица матрице *A* се може одредити помоћу команде `A^-1` или `inv(A)`. Препорука је да се користи команда `inv(A)`, с обзиром да поменута команда за веће матрице одређује прецизније решење.

У претходном примеру смо видели примену команде `^`, која служи за дефинисање степена (експонента).

Вектор у Матлабу може бити дефинисан као вектор врста или вектор колона. Вектор врста је:

```
>> a=[1 2 3 4 5 6]

a =

     1     2     3     4     5     6
```

Док се вектор колона може дефинисати на два начина. Као што је из одговарајућих курсева математике познато, оператор транспоновања је могуће применити и на векторе. У том смислу, вектор колона је:

```
>> a=[1 2 3 4 5 6]'

a =

     1
     2
```

```
     3
     4
     5
     6
```

С друге стране, применом команде „;” могуће је дефинисати вектор колону као:

```
>> a=[1; 2; 3; 4; 5; 6]

a =

     1
     2
     3
     4
     5
     6
```

Као што се из примера може видети, једноставније и ефикасније је дефинисати вектор колону помоћу операције транспоновања.

Уколико имамо задану матрицу *A* и хоћемо да спроведемо неку одређену трансформацију над једном или више њених врста или колона, то можемо урадити применом следећих команди.

```
>> A=[1 2 3;4 5 6;7 8 9]

A =

     1     2     3
     4     5     6
     7     8     9
```

Ако желимо да „извучемо” једну колону (нпр. прву) то можемо урадити на следећи начин:

```
>> A(:,1)

ans =

     1
     4
     7
```

За трећу врсту и све чланове те колоне важи:

```
>> A(3,:)

ans =

     7     8     9
```

Аналогно претходним примерима:

```
>> A(1:2,3)

ans =

     3
     6
```

Матлаб има заиста велики број функција које се „свакодневно" користе у инжењерској пракси (тригонометријске, степене итд.) и било би потребно одржати *бар један курс* који је посвећен искључиво основним могућностима Матлаба. С тим на уму, за све недоумице потребно је консултовати *Help* или одговарајућу референтну литературу.

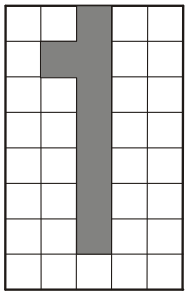
3. Модул за вештачке неуронске мреже (*Neural Network Toolbox*)

Сваком од софтверских модула Матлаба могуће је приступити на два начина. Први начин је директно преко *едитора* позивањем одговарајућих команди. Други начин приступа односи се на иницијализацију модула путем следећих команди – start\Toolboxes\Neural Network.

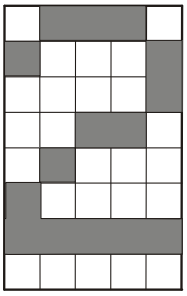
Овом приликом ћемо применити први приступ.

Као што знамо, најбоље се учи на примерима. С тим на уму, урадићемо следећи пример.

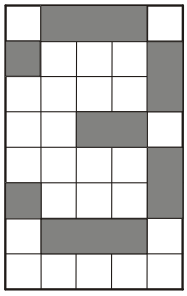
Нека су дате четири цифре (1, 2, 3, 4) и нека је потребно развити одговарајући модел вештачке неуронске мреже са простирањем сигнала у напред, који ће са задатом тачношћу „научити" да препознајје улазне векторе.



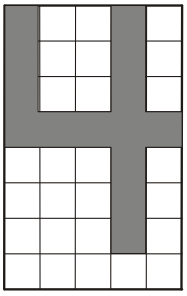
Прва цифра



Друга цифра



Трећа цифра



Четврта цифра

Слика 1

```
clc,clear, close all
% Initialise input vectors
i1 = [0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0
0 0 0]';
i2 = [0 1 1 1 0 1 0 0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 1 1 0 0
0 0 0]';
i3 = [0 1 1 1 0 1 0 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 0
0 0 0]';
```

```
i4 = [1 0 0 0 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0
0 0 0]';
i = [i1 i2 i3 i4]
% Initialise output vectors
o1 = [1 0 0 0]';
o2 = [0 1 0 0]';
o3 = [0 0 1 0]';
o4 = [0 0 0 1]';
o = [o1 o2 o3 o4]
%
% Define the range of input vector
kk = [zeros(size(i,1),1) ones(size(i,1),1)];
% Define the structure of feedforward neural network
net = newff([kk],[10,8],{'logsig','logsig'},'trainlm');
```

Први параметар `[kk]` дефинише опсег улазних вектора (минималну и максималну вредност).

Другим параметром `[10,8]` одређена је архитектура вештачке неуронске мреже која се у овом случају састоји од три слоја (један улазни, један скривени и један излазни слој) са следећим бројем процесирајућих јединица *40x10x8*. Важно је нагласити да се у Матлабу не дефинише експлицитно број неурона у улазном слоју, с обзиром да је та величина увек одређена типом улазног вектора.

Команда `{'logsig','logsig'}` одређује примену сигмоидне активационе функције у скривеном и излазном слоју. Као што се може видети, Матлаб омогућује кориснику да дефинише различите активационе функције за скривени и излазни слој вештачке неуронске мреже.

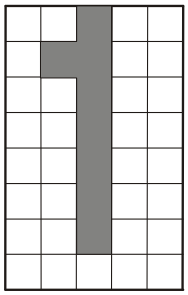
На крају, последњи параметар (`'trainlm'`) одређује примену Левенберг-Маркеоовог поступка приликом обучавања вештачке неуронске мреже. У случају да се на поменутом месту не упише алгоритам учења, Матлаб ће увек као први избор (тзв. *default*) изабрати градијентни поступак.

Поступак формирања вештачке неуронске мреже и њеног обучавања је:

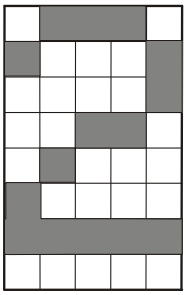
```
%
y = sim(net,i);
% Training parameters
net.trainParam.show = 100;
net.trainParam.lr = 0.5;
net.trainParam.epochs = 1000;
net.trainParam.goal = 1e-5;
%Initialise training
net = train(net,i,o);
%
% View training results
y = sim(net,i(:,:))

TRAINLM-calcjx, Epoch 0/1000, MSE 0.632019/1e-005, Gradient 0.148979/1e-010
TRAINLM-calcjx, Epoch 11/1000, MSE 8.78452e-006/1e-005, Gradient 3.04823e-
005/1e-010
TRAINLM, Performance goal met.

y =[0.9999    0.0000    0.0000    0.0000
    0.0001    0.9967    0.0021    0.0006
    0.0001    0.0026    1.0000    0.0005
    0.0066    0.0013    0.0001    1.0000]
```

Дакле, након 11 итерација (епоха) одређено је решење са задатом тачношћу. Међутим, то не значи да смо завршили задатак, напротив. Приликом обучавања вештачке неуронске мреже, односно приликом одређивања оптималне структуре вештачке неуронске мреже, неопходно је имплементирати више могућих решења да би се одредило оно решење које ће са довољном тачношћу извршити пресликавање из простора улазног вектора у простор излазног вектора.

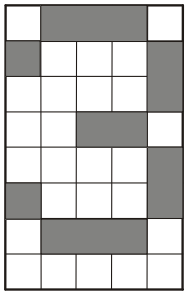
На крају, приказаћемо поступак обучавања вештачке неуронске мреже у циљу препознавања већег броја примера. Нови обучавајући скуп је приказан на слици број два.



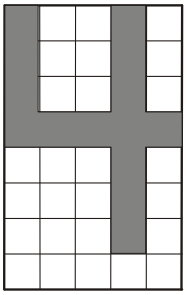
Прва цифра



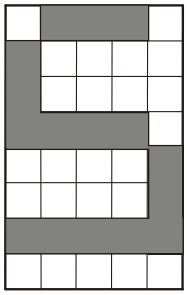
Друга цифра



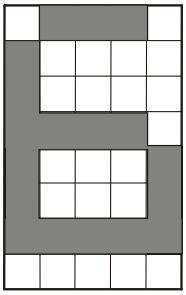
Трећа цифра



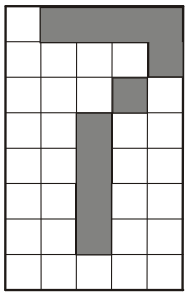
Четврта цифра



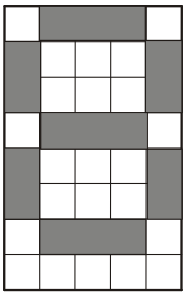
Пета цифра



Шеста цифра



Седма цифра



Осма цифра

Слика 2

Матлаб код је:

```
clc,clear, close all
% Initialise input vectors
i1 = [0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0
0 0 0]';
i2 = [0 1 1 1 0 1 0 0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 1 1 0 0
0 0 0]';
i3 = [0 1 1 1 0 1 0 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 0
0 0 0]';
i4 = [1 0 0 0 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0
0 0 0]';
i5 = [0 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 0 0
0 0 0]';
i6 = [0 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0
0 0 0]';
i7 = [0 1 1 1 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0
0 0 0]';
i8 = [0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 0
0 0 0]';
i = [i1 i2 i3 i4 i5 i6 i7 i8]
% Initialise output vectors
o1 = [1 0 0 0 0 0 0 0]';
o2 = [0 1 0 0 0 0 0 0]';
o3 = [0 0 1 0 0 0 0 0]';
o4 = [0 0 0 1 0 0 0 0]';
o5 = [0 0 0 0 1 0 0 0]';
o6 = [0 0 0 0 0 1 0 0]';
o7 = [0 0 0 0 0 0 1 0]';
o8 = [0 0 0 0 0 0 0 1]';
o = [o1 o2 o3 o4 o5 o6 o7 o8]
%
% Define the range of input
kk = [zeros(size(i,1),1) ones(size(i,1),1)];
% Define the structure of feedforward neural network
net = newff([kk],[10,8],{'logsig','logsig'},'trainlm');

y = sim(net,i);
% Training parameters
net.trainParam.show = 100;
net.trainParam.lr = 0.5;
net.trainParam.epochs = 1000;
net.trainParam.goal = 1e-5;
%Initialise training
net = train(net,i,o);
```

```
%
% View training results
y = sim(net,i(:,:))
```

Резултати обучавања вештачке неуронске мреже су:

```
TRAINLM-calcjx, Epoch 0/1000, MSE 0.292447/1e-005, Gradient 0.0577139/1e-010
TRAINLM-calcjx, Epoch 13/1000, MSE 2.3102e-006/1e-005, Gradient 1.33724e-005/1e-
010
TRAINLM, Performance goal met.


y =[0.9999    0.0000    0.0000    0.0000    0.0002    0.0002    0.0000    0.0000
    0.0000    0.9998    0.0001    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.9989    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0001    0.0000    0.0000    1.0000    0.0000    0.0000    0.0000    0.0001
    0.0001    0.0003    0.0003    0.0002    0.9968    0.0018    0.0001    0.0001
    0.0000    0.0000    0.0000    0.0000    0.0001    0.9885    0.0000    0.0000
    0.0002    0.0000    0.0000    0.0000    0.0000    0.0000    0.9999    0.0000
    0.0000      0.0000      0.0000      0.0000      0.0000      0.0000      0.0000
0.9991]
```