


**Sveučilište u Zagrebu**  
**Fakultet prometnih znanosti**  
**Zavod za inteligentne transportne sustave**  
**Katedra za primijenjeno računarstvo**

	<b>Vježba:</b>	#6
	<b>Kolegij:</b>	Umjetna inteligencija
	<b>Tema:</b>	Neuronske mreže
	<b>Vježbu pripremili:</b>	doc. dr. sc. Edouard Ivanjko Martin Gregurić, mag. ing. traff. Mario Buntić, mag. ing. traff.
<b>Upute za izradu vježbi/zadataka</b>		
Prije dolaska na vježbu potrebno je proučiti pripremu za vježbu. Vježbe je potrebno izraditi pomoću alata koji se koriste. Vrijeme za izradu zadanih zadataka iznosi 90 minuta.		
<b>Cilj vježbe</b>		
Pregled osnovnih MATLAB funkcija za stvaranje i pregled elemenata strukture umjetne neuronske mreže. Generiranje Simulink modela umjetne neuronske mreže. Upoznavanje sa razvojnim okruženjem „Neural Network Tool“ alata. Rješavanje jednostavnog problema pomoću „Neural Network Tool“ alata.		

## Opis vježbe

Neural Network Toolbox je ugrađen u MATLAB okruženje sa svrhom simulacije rada umjetnih neuronskih mreža (engl. „Artificial Neural Network“), te njenom integracijom sa ostalim funkcijama MATLAB-a. Zasniva se prvotno na nizu gotovih funkcija za stvaranje, učenje i testiranje umjetnih neuronskih mreža.

## Neural Network Toolbox funkcije

- *net* = *network* - Osnovna funkcija u Neural Network Toolbox-u je „network“ koja stvara umjetnu neuronsku mrežu bez ulaza i slojeva;
- *net* = *network*(numInputs, numLayers, biasConnect, inputConnect, layerConnect, outputConnect) - funkcija „Network“ može imati sljedeće argumente koje ćemo obraditi u objašnjavanju funkcija Neural Network Toolbox-a.

Ukoliko imenujemo praznu varijablu imenom „net“, te joj pridružimo funkciju „network“ bez argumenta u komandni prostor MATLAB-a, time stvaramo objekt „Neural Network“ u varijabli „net“ koja postaje strukturna varijabla budući je u njoj spremljen objekt. Treba napomenuti kako će sve linije koda kojima će se učiti umjetnu neuronsku mrežu započinjati sa „net.“ budući je to ime strukturne varijable u kojoj je spremljen objekt „Neural Network“, a samim time i sva svojstva umjetne neuronske mreže.

Pritiskom tipke Enter dobivamo ispis svih svojstva objekta umjetnih neuronskih mreža (objekt „Neural Network“). U ovoj vježbi analizirati će se sva svojstva objekta „Neural Network“. Za početak analizirajmo osnovno svojstvo arhitekture - objekta „Neural Network“:

```
net =  
  
Neural Network object:  
architecture:  
  numInputs: 0  
  numLayers: 0  
...
```

Može se vidjeti kako nova umjetna neuronska mreža nema ulaza, niti slojeva, koristeći iduće linije koda i funkcije svojstva možemo ih postaviti na željene vrijednosti:

- *net.numInputs* = *n*; - funkcija svojstva arhitekture objekta „Neural Network“ - „numInputs“ može primiti vrijednost „n“ kao „0“ i sve ostale pozitivne cijele brojeve (engl. „integers“), koji predstavljaju broj ulaza u umjetnu neuronsku mrežu;
- *net.numLayers* = *n*; - funkcija svojstva arhitekture objekta „Neural Network“ - „numLayers“ može primiti vrijednost „n“ kao „0“ i sve ostale pozitivne cijele brojeve, koji predstavljaju broj slojeva u umjetnoj neuronskoj mreži.

Za početak potrebno je stvoriti umjetnu neuronsku mrežu s dva ulaza i tri sloja idućim linijama koda u MATLAB radnom prozoru:

```
net.numInputs = 2; %postavljanje broja ulaza na vrijednost dva ulaza
net.numLayers = 3; %postavljanje broja slojeva na vrijednost tri sloja
```

Ostatak svojstva arhitekture umjetne neuronske mreže je prikazan idućem kodu:

```
...
biasConnect: [0; 0; 0]
inputConnect: [0 0; 0 0; 0 0]
layerConnect: [0 0 0; 0 0 0; 0 0 0]
outputConnect: [0 0 0]
...
```

Funkcija „network“ stvara umjetnu neuronsku mrežu koja nema nikakvih međusobnih veza (engl. „Connections“) između elemenata umjetne neuronske mreže. Matricama popunjenih s jedinicama opisana je prisutnost, odnosno s nulama neprisutnost veze s konstantnom vrijednosti ulaza (engl. „bias“), ulaznim težina, težinama slojeva ili veza s izlazima. Gornji ispis ukazuje da su sve vrijednosti u matricama veza postavljene na vrijednost nula, tj. ne postoje.

Ukoliko želimo postaviti da prvi i treći sloj imaju vezu s konstantnim vrijednostima ulaza koristimo iduću funkciju svojstva:

- $net.biasConnect(i) = n$ ; - funkcija svojstva arhitekture objekta „Neural Network“ - „biasConnect“ može primiti argument „i“ kao vrijednost „0“ i sve pozitivne cijele brojeve, koje predstavljaju broj sloja umjetne neuronske mreže. Pridružena vrijednost „n“ se može popuniti jedino Boole-ovim vrijednostima („1“ postoji veza ili „0“ ne postoji veza);
- $net.biasConnect = [b1; b2; \dots; bn]$ ; - funkciji svojstva „biasConnect“ moguće je pridružiti matricu „N“ x „1“, gdje „N“ predstavlja broj slojeva određen funkcijom svojstva „numLayers“, u tom slučaju funkcija svojstva „biasConnect“ nema argumenta. Matrica  $[b1; b2; \dots; bn]$ ; se može popuniti jedino Boole-ovim logičkim vrijednostima („1“ postoji veza ili „0“ ne postoji veza).

Ukoliko želimo ostvariti vezu sa biasima kako je to prikazano u tablici 1.

Tab. 1. Tablični prikaz veze slojeva sa konstantnom vrijednosti ulaza

<b>biasConnect</b>	<b>Prvi sloj</b>	<b>Drugi sloj</b>	<b>Treći sloj</b>
<b>Veza s biasima</b>	Postoji (1)	Ne postoji (0)	Postoji (1)

Za samo podešavanje veza moguće je iskoristimo iduće linije koda u MATLAB radnom prozoru:

```
net.biasConnect(1) = 1; %veza prvog sloja sa biasom postoji
net.biasConnect(2) = 0; % veza drugog sloja sa biasom ne postoji
net.biasConnect(3) = 1; % veza trećeg sloja sa biasom postoji
```

Ili kraće ukoliko prisutnost veze s biasima po pojedinim slojevima opiše matricom 3x1:

```
net.biasConnect = [1; 0; 1];
```

Ulazne težine i težine slojeva su prikazane matricom 3x2, koja predstavlja postojanje veza iz dva izvora (dva ulaza) do tri odredišta (tri sloja), linije koda za ulazne težine su sljedeće:

- *net.inputConnect(i,j) = n;* – funkcija svojstva „inputConnect“ određuje koji sloj ima težine koje dolaze iz ulaza – ulazne težine. Argument „i“ može biti bilo koji cijeli broj koji predstavlja „i“-ti sloj, u rasponu koji je određen funkcijom „numLayers“, dok argument „j“ može biti bilo koji cijeli broj koji predstavlja „j“-ti ulaza, u rasponu određen funkcijom svojstva „numInputs“. Pridružena vrijednost „n“ se može popuniti Boole-ovim vrijednostima („1“ postoji veza ili „0“ ne postoji veza između „i“ i „j“);
- *net.inputConnect = [NI x Ni];* - funkciji svojstva „biasConnect“ moguće je pridružiti matricu „NI“ x „Ni“, gdje „NI“ predstavlja broj slojeva određen funkcijom svojstva „numLayers“, a „Ni“ predstavlja broj ulaza u rasponu određen funkcijom svojstva „numInputs“. U tom slučaju funkcija svojstva „biasConnect“ nema argumenta. Pridružena matrica se može popuniti jedino Boole-ovim vrijednostima („1“ postoji ili „0“ ne postoji veza).

Za primjer ostvarimo vezu sa biasima kako je to prikazano u tablici 2.

Tab. 2. Odnos slojeva i ulaza u „inputConnection“ funkciji

<b>inputConnection</b>	<b>Prvi sloj</b>	<b>Drugi sloj</b>	<b>Treći sloj</b>
<b>Prvi ulaz</b>	1	1	0
<b>Drugi ulaz</b>	0	1	0

Koristimo iduće linije koda u MATLAB radnom prozoru:

```
net.inputConnect(1,1) = 1; %veza prvog sloja sa prvim ulazom  
net.inputConnect(2,1) = 1; %veza drugog sloja sa prvim ulazom  
net.inputConnect(2,2) = 1; %veza drugog sloja s drugim ulazom
```

ili pomoću jedne linije koda:

```
net.inputConnect = [1 0; 1 1; 0 0];
```

gdje [1 0; 1 1; 0 0], predstavlja matricu oblika  $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ . Moguće je uočiti kako su vrijednosti stupaca u sintaksi MATLAB-a odvojeni znakom „;“. Odnos slojeva i ulaza pojašnjen je tablicom 3.2. Iduća bitna funkcija svojstva je „layerConnect“, u nastavku je prikazana njezina upotreba u linijama koda:

- *net.layerConnect(i,j) = n;* – funkcija svojstva „layerConnect“ određuje koji sloj ima težine prema drugom sloju – težine sloja. Argument „i“ može biti bilo koji cijeli broj koji predstavlja „i“-ti sloj prema kojem je usmjerena težina, dok argument „j“

može biti bilo koji cijeli broj koji predstavlja „j“-ti sloj od kojeg je usmjerena težina. I „i“ i „j“ moraju biti u rasponu određen funkcijom svojstva „numLayers“. Pridružena vrijednost „n“ se može popuniti Boole-ovim vrijednostima („1“ postoji veza ili „0“ ne postoji veza između „i“ i „j“ sloja);

- *net.layerConnect* = [NI x Ni]; - funkciji svojstva „biasConnect“ moguće je pridružiti matricu „NI“ x „Ni“, gdje „NI“ predstavlja broj slojeva od kojih su usmjerene težine „Ni“ predstavlja broj slojeva, prema kojima su usmjerene težine. Raspon je određen funkcijom svojstva „numLayers“. U tom slučaju funkcija svojstva „layerConnect“ nema argumenta. Pridružena matrica se može popuniti jedino Boole-ovim vrijednostima („1“ postoji ili „0“ ne postoji veza).

Ukoliko želimo stvoriti vezu između prvog, drugog i trećeg sloja sa trećim slojem, kako je prikazano na tablici 3.

Tab. 3. Odnos slojeva i ulaza u „layerConnection“ funkciji

<b>layerConnection</b>	<b>Prvi sloj</b>	<b>Drugi sloj</b>	<b>Treći sloj</b>
<b>Prvi sloj</b>	0	0	1
<b>Drugi sloj</b>	0	0	1
<b>Treći sloj</b>	0	0	1

Koristimo primjerice iduću liniju koda za realizaciju tablice 3.:

```
net.layerConnect = [0 0 0; 0 0 0; 1 1 1];
```

Veze sa izlazom su predstavljene matricom 1x3, koja ukazuje da je spojena sa jednim odredištem („vanjskim svijetom“) iz tri izvora (tri sloja). U umjeravanju veza slojeva sa izlazom koristimo funkciju svojstva „outputConnect“:

- *net.outputConnect(i) = n;* - funkcija svojstva arhitekture objekta „Neural Network“ - „outputConnect“ može primiti argument „i“ kao vrijednost „0“ i sve pozitivne cijele brojeve, koje predstavljaju broj sloja umjetne neuronske mreže s kojim je spojen ulaz. Pridružena vrijednost „n“ se može popuniti jedino Boole-ovim vrijednostima („1“ postoji veza ili „0“ ne postoji veza);
- *net.outputConnect = [b1 b2 ... bn];* - funkciji svojstva „outputConnect“ moguće je pridružiti matricu „1“ x „N“, gdje „N“ predstavlja broj slojeva određen funkcijom svojstva „numLayers“, u tom slučaju funkcija svojstva „outputConnect“ nema argumenta. Matrica [b1 b2... bn] se može popuniti jedino Boole-ovim vrijednostima („1“ postoji veza ili „0“ ne postoji veza).

Ukoliko želimo spojiti drugi i treći sloj sa izlazom prema tablici 4.:

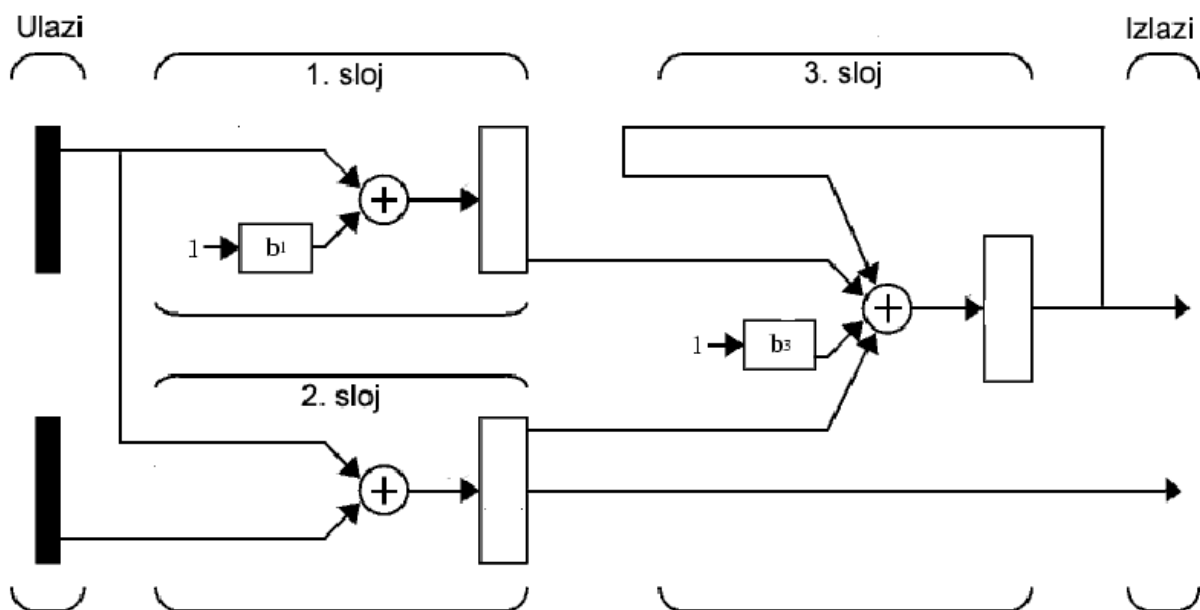
Tab. 4. Tablični prikaz veze slojeva sa ulazima

inputConnect	Veze s ulazima
Prvi sloj	0
Drugi sloj	1
Treći sloj	1

Koristimo primjerice iduću liniju koda:

```
net.outputConnect = [0 1 1];
```

Dosadašnje naredbe koje smo unijeli u MATLAB radni prozor dobit ćemo strukturu umjetne neuronske mreže koja je prikazana na slici 1.



Slika 1. Struktura umjetne neuronske mreže sa biasima i vezama između slojeva, ulaza i izlaza.

Kada smo drugi i treći sloj spojili s mrežnim izlazom, automatski smo odredili da umjetna neuronska mreža ima dva izlaza. U objektu Neural Network postoji i podobjekt struktura (engl. „Subobject structures“):

subobject structures:

- inputs: {2x1 cell} of inputs
- layers: {3x1 cell} of layers
- outputs: {1x3 cell} containing 1 output
- biases: {3x1 cell} containing 2 biases
- inputWeights: {3x2 cell} containing 3 input weights
- layerWeights: {3x3 cell} containing 3 layer weights

Kada pomoću funkcije „numInputs“ postavimo vrijednost „2“, ulazna struktura umjetne neuronske mreže postane polje ćelija sa dvije ulazne strukture. Dodatnim svojstvima

strukture „i“-te strukture ulaza pristupamo funkcijom podobjeka „inputs“ u slijedećim linijama koda:

- `net.inputs` – ukoliko pozovemo ovom naredbom funkciju podobjeka „inputs“ dobit ćemo ispis rasporeda struktura pojedinog ulaza kojeg sadrži objekt „inputs“;
- `net.inputs{i}` – postavljanjem argumenta „i“, dobiva se detaljni pregled svojstva i njihovih vrijednosti za pojedinu strukturu koja odgovara broju u argumentu „i“, tj. argument „i“ može biti bilo koji cijeli broj u rasponu ukupnog broja ulaznih struktura. Svaka ulazna „i“-ta struktura, do koje se pristupa pomoću funkcije „inputs“, sadrži dodatna svojstva povezana sa ulaznom „i“-tom strukturom.

Kako bi vidjeli kako su strukture raspoređene unutar ulaza unesimo slijedeću naredbu u MATLAB radni prozor:

```
net.inputs
```

pritisком tipke Enter dobivamo:

```
ans =  
    [1x1 struct]  
    [1x1 struct]
```

Kako bi vidjeli svojstva za strukturu prvog ulaza unesimo slijedeću naredbu u MATLAB radni prozor:

```
net.inputs{1}
```

pritisком tipke Enter dobiva se:

```
ans =  
    exampleInput: [0 1]  
    processFcns: {}  
    processParams: {}  
    processSettings: {}  
    processedRange: [0 1]  
    processedSize: 1  
    range: [0 1]  
    size: 1  
    userdata: [1x1 struct]
```

Prvo analizirajmo liniju koda sa funkcijom svojstva „exampleInput“ podobjeka „inputs“:

- `net.inputs{i}.exampleInput = [Ri x Q]`; - svojstvo „exampleInput“ podobjeka „inputs“ može se pridružiti matrica „Ri“ x „Q“, gdje je „Ri“ broj elemenata koje ima „i“-ta struktura ulaza koja će biti određena svojstvom „range“ podobjeka „inputs“. Postavljanjem „exampleInput“ svojstva, svojstva „range“, „size“ i „processedSize“ će se automatski postaviti na nove vrijednosti kako bi zadovoljili

vrijednosti unesene funkcijom „exampleInput“. „Q“ predstavlja broj podataka u elementu.

Postavimo „exampleInput“ svojstvo na slijedeće vrijednosti:

```
net.inputs{1}.exampleInput = [0 10 5; 0 3 10]
```

postavili smo ovom naredbom u sadržaj strukture prvog ulaza tri elementa sa dvije vrijednosti, nakon pritiska tipke Enter, dobit ćemo slijedeća promijenjena svojstva:

```
ans =  
  
exampleInput: [2x3 double]  
name: 'Input'  
processFcns: {}  
processParams: {}  
processSettings: {1x0 cell}  
processedRange: [2x2 double]  
processedSize: 2  
range: [2x2 double]  
size: 2  
userdata: [1x1 struct]
```

Sadržaj strukture drugog ulaza postavimo primjerice na pet elementa idućom linijom koda:

```
net.inputs{2}.exampleInput = [0 8 5; 0 5 10; 0 5 5; 0 8 8; 0 1 2];
```

Također, možemo pregledavati vrijednosti pojedinačnih svojstva, ukoliko primjerice koristimo svojstvo „size“ podobjekta „inputs“:

- *net.inputs{i}.size*; - svojstvo „size“ podobjekta „inputs“ ne mora se pridružiti niti jedna vrijednost u tom slučaju ukoliko je naredba unesena u MATLAB radni prostor bit će ispisana njena vrijednost za „i“-tu strukturu ulaza;
- *net.inputs{i}.size = n*; - ukoliko želimo postaviti vrijednost svojstva „size“ na željenu vrijednost morat ćemo najprije unesti svojstvo „size“ linijom koda gdje je „n“ željena cjelobrojna vrijednost broja elemenata „i“-tog ulaza, a tek onda u skladu s vrijednosti „n“ unesti svojstvo „exampleInput“.

Unesemo za primjer liniju koda u radni prozor MATLAB-a koja će prikazati broj elemenata u prvom ulazu:

```
net.inputs{1}.size
```

nakon pritiska tipke Enter, dobit ćemo slijedeći rezultat koji je posljedica prethodno umjerenog svojstva „exampleInput“ :

```
ans =
```



Svojstvo „processFcns“ predstavlja procesnu funkciju koja dodatno postavlja ograničenja na elemente pojedinog ulaza. Primjerice procesna funkcija „removeconstantrows“ će u ulaznim elementima maknuti redove elementa ulaza koji se po svojim vrijednostima jednaki, ili procesna funkcija 'mapminmax' koja će najvećem elementu u matrici strukture ulaza pridodati vrijednost „1“, a najmanjem -1. Iduća linija koda se primjenjuje ukoliko se želi umjeriti svojstvo „processFcns“:

- `net.inputs{i}.processFcns = {'pf1','pf2', ... , 'pfn'};` - svojstvu „processFcns“ podobjekta „inputs“ može se pridružiti više procesnih funkcija - 'pf1','pf2', ... , 'pfn'. Maksimalni broj procesnih funkcija ovisi o broju elemenata „i“-tog ulaza.

Popisu i objašnjenje svih procesnih funkcija koje podržava MATLAB pristupa se pomoću naredbe u radnom prostoru MATLAB-a: „help nnprocess“. Postavimo za primjer, na strukturu prvog ulaza dvije spomenute procesne funkcije slijedećom linijom koda:

```
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
```

Iduća važna funkcija podobjekta je „layers“. Koristi se u osnovnom obliku u slijedećoj liniji koda:

- `net.layers{i}` – funkcijom podobjekta „layers“ sa argumentom „i“, dobivamo detaljni pregled svojstva i njihovih vrijednosti za strukturu „i“-tog sloja, tj. argument „i“ može biti bilo koji cijeli broj u rasponu ukupnog broja slojeva. Svaka „i“-ta struktura sloja do koje se pristupa pomoću funkcije „layers{i}“, sadrži dodatna svojstva povezana sa „i“-tim slojem.

Postavljanjem broja slojeva pomoću funkcije „numLayers“ na „3“, svojstvo slojeva se postavi u polje ćelija organizirano u 3 sloja, tj. reda (kako je prikazano u podobjektu struktura „{3x1 cell}“). Ukoliko upišemo iduću liniju koda:

```
net.layers{1}
```

u MATLAB radni prostor i pritisnemo tipku Enter dobivamo idući prikaz svojstva prvog sloja:

```
ans =
  dimensions: 1
  distanceFcn: 'dist'
  distances: 0
  initFcn: 'initwb'
  netInputFcn: 'netsum'
  netInputParam: [1x1 struct]
  positions: 0
  size: 1
  topologyFcn: 'hextop'
  transferFcn: 'purelin'
  transferParam: [1x1 struct]
  userdata: [1x1 struct]
```

Broj neurona u strukturi sloja podobjekta „layers“ određen je svojstvom „size“. Iduće se naredbe koriste za umjeravanje svojstva „size“:

- *net.layers{i}.size*; - svojstvo „size“ (broj neurona) podobjekta „layers“ ne mora se pridružiti niti jedna vrijednost, u tom slučaju ukoliko je naredba unesena u MATLAB radni prostor bit će ispisana njena vrijednost za „i“-ti sloj;
- *net.layers{i}.size = n*; - ukoliko želimo postaviti vrijednost svojstva „size“ pridružiti ćemo varijabli „n“ željenu cijelobrojnu vrijednost broja neurona „i“-tog sloja.

Kako bi primjerice promijenili broj neurona u prvom sloju sa jednog u četiri, a u drugom sloju s jednog na tri, u radni prostor MATLAB-a unesimo slijedeće linije koda:

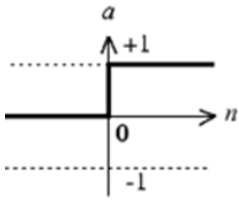
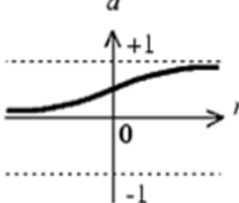
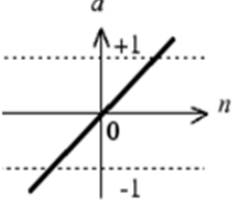
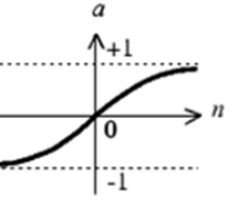




```
net.layers{1}.size = 4;
net.layers{2}.size = 3;
```

Važno svojstvo sloja umjetne neuronske mreže podobjekta „layers“ je prijenosna funkcija koju će imati neuroni dotičnog sloja. U MATLAB kodu to je svojstvo označeno svojstvom „transferFcn“, a koristi se kako je opisano u idućoj liniji koda:

- *net.layers{i}.transferFcn = 'tF'*; - ukoliko želimo izabrati prijenosnu funkciju svojstva funkcije „transferFcn“ podobjekta „layers“ pridružiti ćemo varijabli „tF“ MATLAB kod za određenu prijenosnu funkciju koja će se primijeniti na svim neuronima „i“-tog sloja.

MATLAB podržava nekoliko prijenosnih funkcija koje su opisane i prikazane tablicom 5.

Tab. 5. Prikaz uobičajenih prijenosnih funkcija i njihovih obilježja u MATLAB-u

Ime prijenosne funkcije	Funkcija skoka, praga	Log-sigmoidna funkcija	Linearna funkcija	Tan-sigmoidna funkcija
Prikaz prijenosne funkcije u koordinatn-om sustavu				
Grafički prikaz u Simulink-u				
MATLAB kod prijenosnih funkcija	$a = \text{hardlim}(n)$	$a = \text{logsig}(n)$	$a = \text{purelin}(n)$	$a = \text{tansig}(n)$

Za primjer, postavimo svojstvo prijenosne funkcije prvog sloja na Tan-sigmoidnu funkciju, što se može učiniti idućom linijom koda:

```
net.layers{1}.transferFcn = 'tansig';
```

Drugi sloj umjetne neuronske mreže imati će svojstvo prijenosne funkcije postavljeno na vrijednost Log-sigmoidne funkcije, a taj postupak će biti izveden slijedećim linijom koda:

```
net.layers{2}.transferFcn = 'logsig';
```

Neural Network Toolbox-u za stvaranje umjetne neuronske mreže potrebne su inicijalizacijske funkcije ili algoritmi koji moraju biti definirani za svaki sloj kao njegovo svojstvo. Inicijalizacijska funkcija ili algoritam prilagođava početne vrijednosti biasa i težina pojedinog sloja. Za primjer ćemo koristiti *Nguyen-Widrow* algoritama (u MATLAB kodu sadrži ga varijabla „initnw“). Iduća linija koda sadržava funkciju svojstva „initFcn“ za odabir svojstva inicijalizacijske funkcije:

- *net.layers{i}.initFcn = 'iF';* - ukoliko želimo izabrati inicijalizacijsku funkciju, odnosno algoritam svojstva funkcije „initFcn“ podobjekta „layers“ pridružiti ćemo pomoću varijable „iF“ MATLAB kod za određenu inicijalizacijsku funkciju, odnosno algoritam koji će se primijeniti u „i“-tom sloju.

Sljedeće linije koda postaviti će svojstva inicijalizacijske funkcije na *Nguyen-Widrow* algoritam u sva tri sloja:

```
net.layers{1}.initFcn = 'initnw';  
net.layers{2}.initFcn = 'initnw';  
net.layers{3}.initFcn = 'initnw';
```

Kod umjetne neuronske mreže od izrazite važnosti su funkcije za pregled svojstva biasa za svaki sloj. Naša umjetna neuronska mreža ima postavljene biase na prvom i trećem sloju. Ukoliko želimo pregledati pojedina svojstva skupine biasa po spomenutim slojevima koristimo iduću liniju koda:

- *net.biases{i}* – funkcija „biases“ prikazuje pregled svojstva i njihovih vrijednosti za biase. Argument „i“ može biti bilo koji cijeli broj koji predstavlja sloj u kojem postoji veza sa biasima. Svaka „i“-ti sloj s biasima, do kojeg se pristupa pomoću funkcije „biases“, sadrži dodatna svojstva povezana sa „i“-tim slojem s vezom s biasima.

Moguće je ispisati slijedeće linije koda sa funkcijom svojstva „biases“:

```
net.biases{1} %pregled svojstva biasa u prvom sloju  
net.biases{3} %pregled svojstva biasa u trećem sloju
```

Ukoliko unesemo samo prvu liniju koda radni prostor MATLAB-a i pritisnemo tipku Enter, dobit ćemo slijedeći rezultat:

```

ans =
    initFcn: ''
    learn: 1
    learnFcn: ''
    learnParam: ''
    size: 4
    userdata: [1x1 struct]

```

Svojstvo „size“ je postavljeno na vrijednost „4“, budući prvi sloj ima 4 neurona, zaključujemo kao svaki neuron u sloju ima jedan pripadajući bias. Za prikaz svojstva svih težina ulaza koristimo funkciju „inputWeights“ u slijedećoj liniji koda:

- *net.inputWeights{i,j}* – funkcija svojstva „inputWeights“ koja predstavlja težinu od ulaza „j“ do sloja „i“, pod uvjetom da je funkcija „inputConnect(i,j)“ postavljena na vrijednost „1“.

Prema tom naputku moguće izvesti sljedeće linije koda za postojeću umjetnu neuronsku mrežu:

```

net.inputWeights{1,1} %pregled svojstva težina ulaza iz prvog ulaza prema prvom sloju
net.inputWeights{2,1} %pregled svojstva težina ulaza iz prvog ulaza prema drugom sloju
net.inputWeights{2,2} %pregled svojstva težina ulaza iz drugog ulaza prema drugom sloju

```

Jedno od bitnih svojstva funkcije „inputWeights“ je svojstvo kašnjenja (eng. „delays“). Svojstvo kašnjenje odgađa prolazak signala, tj. vrijednosti težina sloja iz jednog sloja u drugi za uzorak vremena koji je određen u svojstvu „delays“.

Kašnjenje koristimo ukoliko želimo umjeriti umjetnu neuronsku mrežu izvođenju nekoliko obrada između ulaza i sloja s kojim je ulaz u vezi, pa tek onda proslijediti drugom sloju. Za uvođenje kašnjenja koristi se iduća naredba:

- *net.inputWeights{i,j}.delays = n;* - funkcija svojstva „delays“ predstavlja kašnjenje za težinu od ulaza „j“ do sloja „i“, pod uvjetom da je funkcija „inputConnect(i,j)“ postavljena na vrijednost „1“. Pridružena vrijednost „n“ može poprimiti vrijednost „0“ ili bilo koji cijeli broj koji odgovara jednom vremenskom intervalu kašnjenja.

Linije kodova u MATLAB radnom prostoru kao primjer umjeravanja kašnjenja u vezama između ulaza i pripadajućih slojeva na vrijednost „1“, tj. nepostojanje kašnjenja, su slijedeće:

```

net.inputWeights{2,1}.delays = 1;
net.inputWeights{2,2}.delays = 1;

```

Kod težina slojeva za umjetnu neuronsku mrežu imamo sličnu situaciju kao i kod težina ulaza. Za prikaz svojstva svih težina slojeva koristimo funkciju „layerWeights“ u idućoj liniji koda:

- *net.layerWeights{i,j}* – funkcija svojstva „layerWeights“ predstavlja svojstva težine sloja od sloja „i“ prema sloju „j“, pod uvjetom da je funkcija „layerConnect(i,j)“ postavljena na vrijednost „1“.

Prema tome naputku možemo izvesti sljedeće linije koda za postojeću umjetnu neuronsku mrežu:

```
net.layerWeights{3,1} % pregled svojstva težine sloja od prvog sloja prema
trećem sloju
net.layerWeights{3,2} % pregled svojstva težine sloja od drugog sloja prema
trećem sloju
net.layerWeights{3,3} % pregled svojstva težine sloja od trećeg sloja prema
trećem sloju
```

Dok se primjerice kašnjenje između trećeg sloja prema trećem sloju postavlja na vrijednost „1“ pomoću iduće linije koda:

```
net.layerWeights{3,3}.delays = 1;
```

U podobjektu strukture objekta „Neural Network“, postoje i svojstva vrijednosti biasa i težina slojeva i ulaza, čija je osnovana struktura prikazana izvodom objekta „Network“:

weight and bias values:

```
IW: {3x2 cell} containing 3 input weight matrices
LW: {3x3 cell} containing 3 layer weight matrices
b: {3x1 cell} containing 2 bias vectors
```

Uviđamo iz izvoda kako svojstvo vrijednosti ulaznih težina („IW“) i težina slojeva („LW“) sadrži tri matrice (engl. „containing 3 input weight matrices“, odnosno „containing 3 layer weight matrices“), tj. po jednu matricu za svaku vezu između ulaza i sloja, odnosno sloja i sloja. Svojstvo biasa („b“) ima dva vektora svaki za pojedini sloj u kojemu su prisutni biasi („engl. containing 2 bias vectors“). Vrijednosti težina ulaza, težina slojeva i vektori biasa raspoređene su matricama različite strukture. Primjerice u izvodu su vrijednosti težine ulaza prikazane u obliku matrice s 3 retka i 2 stupca „{3x2 cell}“, odnosno 3 retka i 3 stupca „{3x3 cell}“ za vrijednosti težina slojeva.

Funkcije biasa, težina slojeva i ulaza vraćaju odgovarajuće numeričke vrijednosti, te se pokreću idućim linijama koda:

- *net.IW{i,j}* – funkcija „IW“ predstavlja vrijednost težina ulaza od ulaza „j“ do sloja „i“, pod uvjetom da je funkcija „inputConnect(i,j)“ postavljena na vrijednost „1“;
- *net.LW{i,j}* - funkcija svojstva „LW“ predstavlja vrijednosti težina sloja od sloja „i“ prema sloju „j“, pod uvjetom da je funkcija „layerConnect(i,j)“ postavljena na vrijednost „1“;
- *net.b{i}* – funkcija svojstva biasa „b“ vraća vrijednost biasa „i“-tog sloja u kojem je funkcija „biasConnect(i)“ postavljena na vrijednost „1“.

Svaka matrica vrijednosti koju vraća težina ulaza „ $IW\{i,j\}$ “, težine sloja „ $LW\{i,j\}$ “ i vektora biasa „ $b\{i\}$ “ ima onoliko redaka koliko odgovarajući sloj „ $i$ “ na koji se odnosi ima neurona ( naredba „ $net.layers\{i\}.size$ “). Matrica vrijednosti težina ulaza „ $IW\{i,j\}$ “, ima onoliko stupaca koliko ima elemenata u odgovarajućem ulazu „ $j$ “ ( naredba „ $net.inputs\{j\}.size$ “) pomnoženo s odgovarajućom vrijednosti kašnjenja ( naredba „ $length(net.inputWeights\{i,j\}.delays)$ “). Prema tome, svaka matrica vrijednosti težine sloja „ $LW\{i,j\}$ “ ima onoliko stupaca koliko sloj „ $j$ “ ima neurona ( naredba „ $net.layers\{j\}.size$ “) pomnoženo s vrijednosti kašnjenja pripadnog sloja ( naredba „ $length(net.layerWeights\{i,j\}.delays)$ “).

Prema navedenom moguće je ispisati slijedeće linije koda u MATLAB radni prostor:

```
net.IW{1,1}, net.IW{2,1}, net.IW{2,2}
net.LW{3,1}, net.LW{3,2}, net.LW{3,3}
net.b{1}, net.b{3}
```

Pokretajući pojedinačno svaku od tih linija koda vidjet ćemo da su one sve postavljene na vrijednost nula, tj. da su prazne (neinicijalizirane). Inicijalizacija se vrši pomoću iduće linije koda s funkcijom „init“:

- $i = init(i)$ ; - funkcijom „init“ se izvršava inicijalizacija umjetne neuronske mreže spremljene pod nazivom „ $i$ “ u Workspace-u.

Pa se inicijalizacija umjetne neuronske mreže spremljene pod nazivom „net“ izvršava slijedećom linijom koda:

```
net = init(net);
```

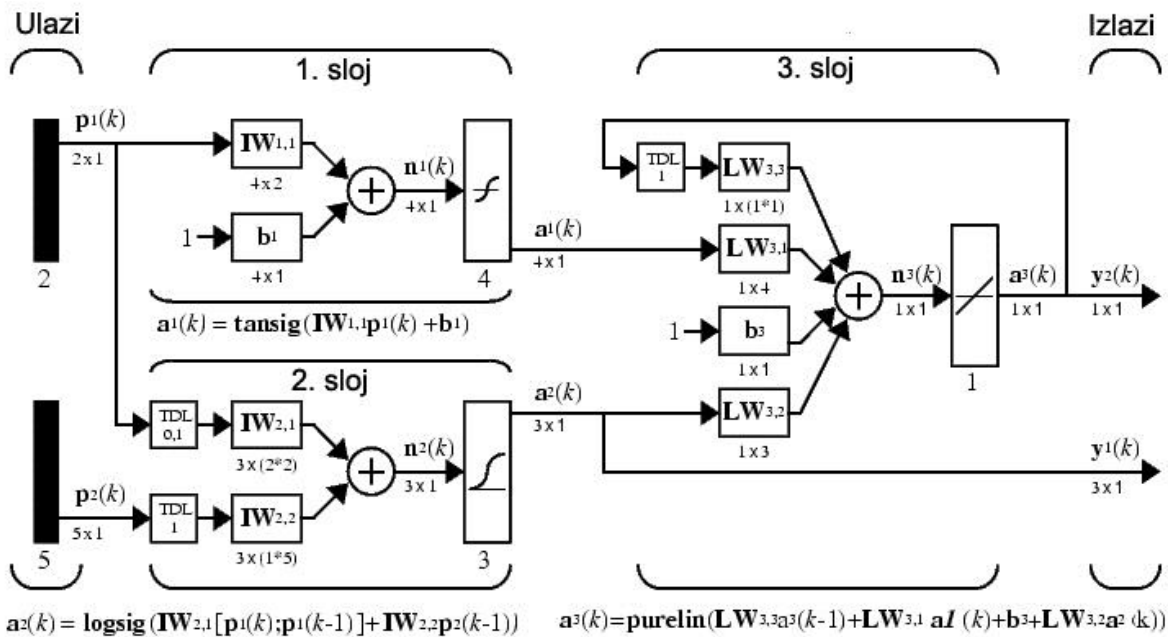
Ukoliko nakon inicijalizacije unesemo iduću liniju koda za prikaz vrijednosti težina ulaza iz prvog ulaza prema prvom sloju u MATLAB radni prostor:

```
net.IW{1,1}
```

dobivamo idući matricu kao rezultat:

```
ans =
-0.3040  0.4703
-0.5423 -0.1395
 0.5567  0.0604
 0.2667  0.4924
```

Opisane su do sada temeljne funkcije za stvaranje neuronske mreže uz pomoć „Network“ funkcije, te je stvorena struktura konačne umjetne neuronske mreže kao na slici 2.



Slika 2. Konačna struktura umjetne neuronske mreže

## Neural Network Toolbox za Simulink

Neural Network Toolbox podržan je i u Simulink knjižnici. Dijeli se na pet skupova blokova, koje možemo dodatno podijeliti na blokove sa implementiranom cjelokupnom umjetnom neuronskom mrežom:

- Upravljački sustavi/regulatori (engl. „Control Systems“) – blokovi koji koriste umjetne neuronske mreže kao kompletni upravljački sustav.

Blokove za izradu umjetne neuronske mreže od razine sastavnih dijelova pojedinog neurona:

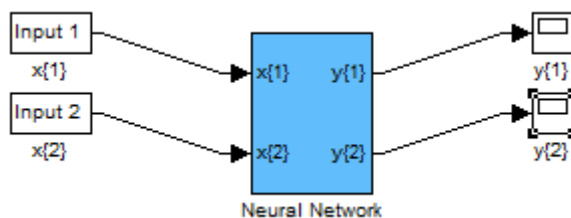
- Funkcije za sumiranje/umnožak ulaza u neuron umjetne neuronske mreže (engl. „Net Input Systems“),
- Procesne funkcije (engl. „Processing Functions“),
- Prijenosne funkcije (engl. „Transfer functions“),
- Funkcije koje služe za postavljanje/mijenjanje težinskih vrijednosti ulaza u pojedini neuron umjetne neuronske mreže (engl. „Weight Functions“).

Ukoliko se želi dobiti prikaz modela umjetne neuronske mreže u Simulink-u može se koristiti funkcija „gensim“. Ulazni argument je ime strukturne varijable u kojoj je spremljena umjetna neuronska mreža.

Naredba za izradu Simulink modela naše umjetne neuronske mreže je:

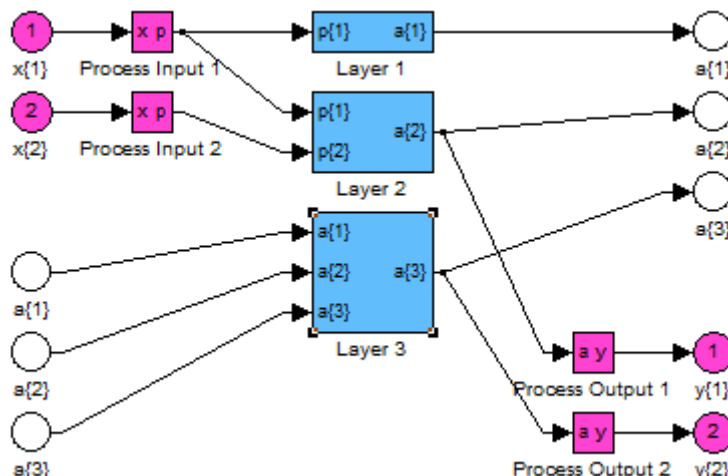
```
gensim(net);
```

Rezultat je Simulink model na kojem uočavamo kako postoje dva ulaza i dva izlaza, te podsustav „Neural Network“ u kojem se nalazi model same umjetne neuronske mreže kao što je prikazano na slici 3.:



Slika 3. Prikaz osnovnog modela umjetne neuronske mreže

Dvostrukim klikom lijeve tipke miša na blok podsustava „Neural Network“ dobivamo model na kojem se jasno razaznaju tri sloja kao podsustavi koji su povezani na prethodno definiran način pomoću naredba za konstrukciju umjetne neuronske mreže. Prikaz modela same umjetne neuronske mreže u Simulink-u može se vidjeti na slici 4.:



Slika 4. Prikaz modela umjetne neuronske mreže

### Neural Network Toolbox u grafičkom korisničkom sučelju

Na temelju prethodno objašnjenih funkcija i na njima baziranim linijama koda stvoreni su gotovi alati koji se zasnivaju na grafičkom korisničkom sučelju (engl. „Graphical User Interface – GUI“). Svaki od njih stvoren je kako bi izvršio specifičnu zadaću koju je u mogućnosti izvesti umjetna neuronska mreža ukoliko se u tom smjeru umjeri njeni tip i parametri. Alati Neural Network Toolbox-a izvedeni u GUI-u, ne zahtijevaju detaljnije poznavanje rada umjetnih neuronskih mreža, već općenito teoretsko znanje. Zajednička im je i osobina što zahtijevaju posjedovanje minimalno dva skupa podataka (s iznimkom za alat za uskupljivanje kojem su potrebni samo ulazni podatci). Prvi skup podataka je ulazni (engl. „Input“), dok drugi skup podataka predstavlja očekivane vrijednosti s obzirom na ulazne podatke – ciljevi (engl. „Targets“). Skupovi se uglavnom koristi za učenje same umjetne neuronske mreže. Postoje tri specijalizirana alata sa već ugrađenim određenim tipom umjetne neuronske mreže koja se koristi za rješavanje određenog tipa problema:

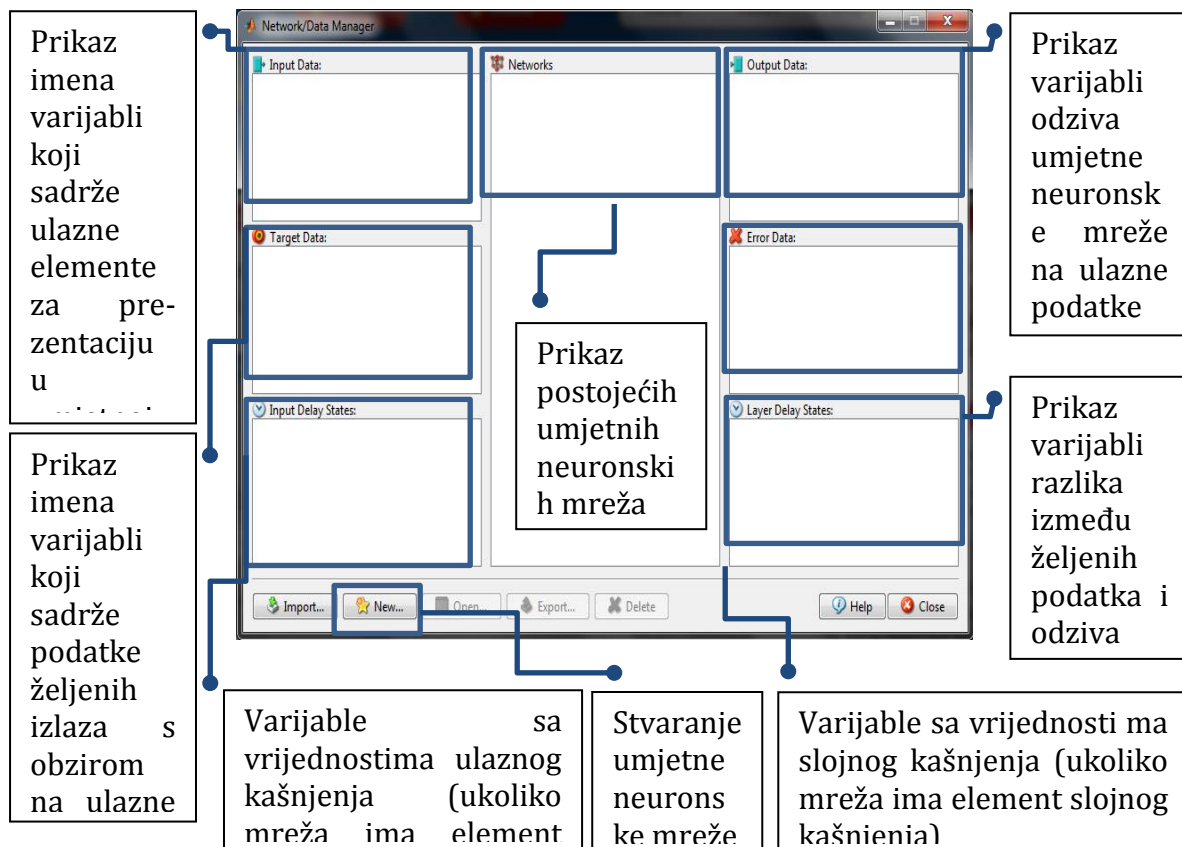


- **Neural Fitting Tool („nftool“)** – alat služi za pronalaženje odnosa između ulaznih skupova podataka i izlaznih (ciljeva), koristeći unaprijednu ili acikličku (engl. „feed-forward“) dvoslojnu umjetnu neuronsku mrežu
- **Neural Clustering Tool („nctool“)** – alat za rješavanje problema u kojem trebamo grupirati skup podataka po sličnostima, koristeći samo-organizirajuću neuronsku mrežu (engl. „self-organizing map – SOM network“)
- **Neural Pattern Recognition Tool („nprtool“)** – alat za probleme u kojima je potrebno izvršiti prepoznavanje uzoraka. U takvim problemima želimo da umjetna neuronska mreža klasificira ulaze u kategorije koje predstavljaju skupovi ciljnih podataka. Ovaj alat koristi unaprijednu dvoslojnu umjetnu neuronsku mrežu.

Od alata je također potrebno spomenuti i alat „Neural Network Training Display“ koji se pokreće za vrijeme trajanja učenja umjetne neuronske mreže. Uz zaustavljanje učenja u bilo kojem trenutku i mijenjanje parametara učenje, po završetku učenja omogućuje razne grafičke prikaze rezultata učenja.

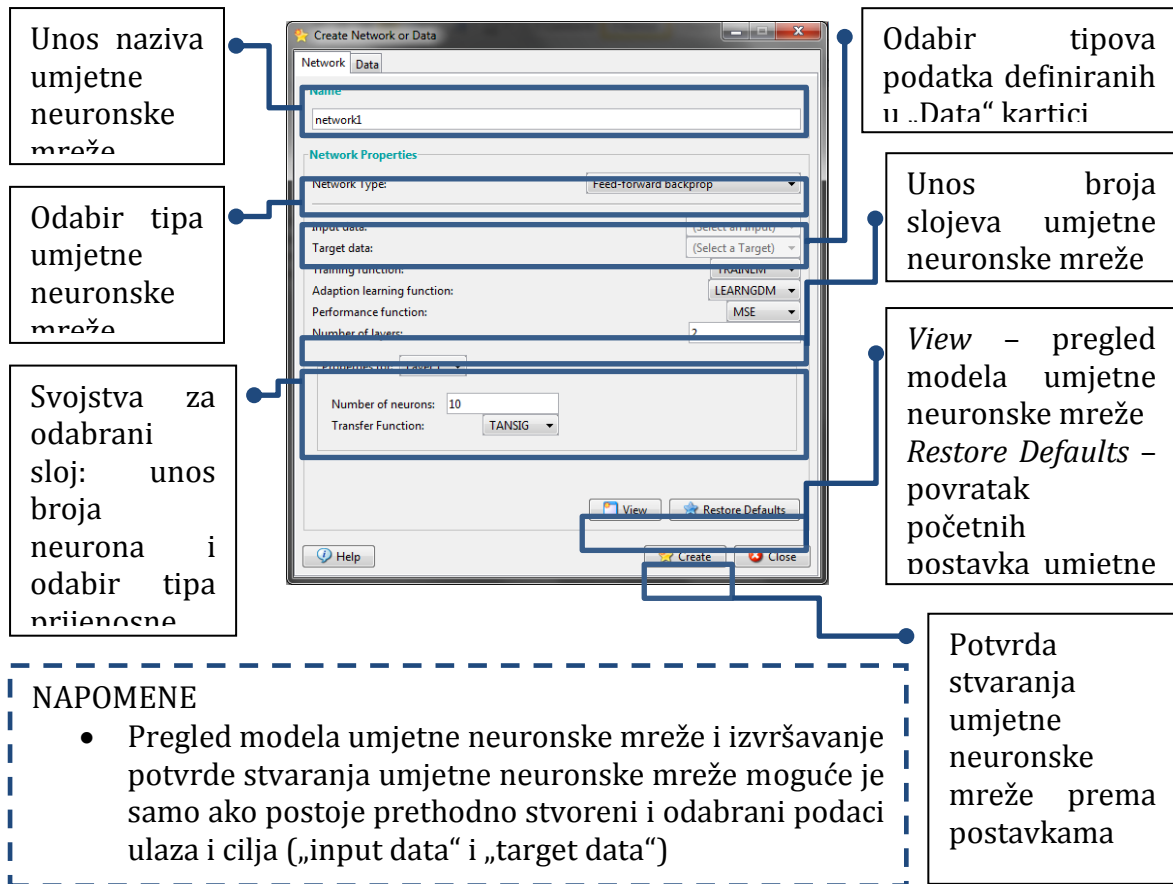
## Neural Network Tool

U vježbama će se koristiti **Neural Network Tool („nntool“)**. Budući je to alat koji nije usmjeren za rješavanje specifičnih tipova problema, već u svojim opcijama omogućava slobodnije umjeravanje umjetnih neuronskih mreža. Spomenuta mogućnost mu omogućava rješavanje šire skupine problema uključujući i one koje rješavaju specijalizirani alati. Početni prozor i osnovne funkcije alata „Neural Network Tool“ koje će se koristiti u vježbama prikazane su na slici 5.



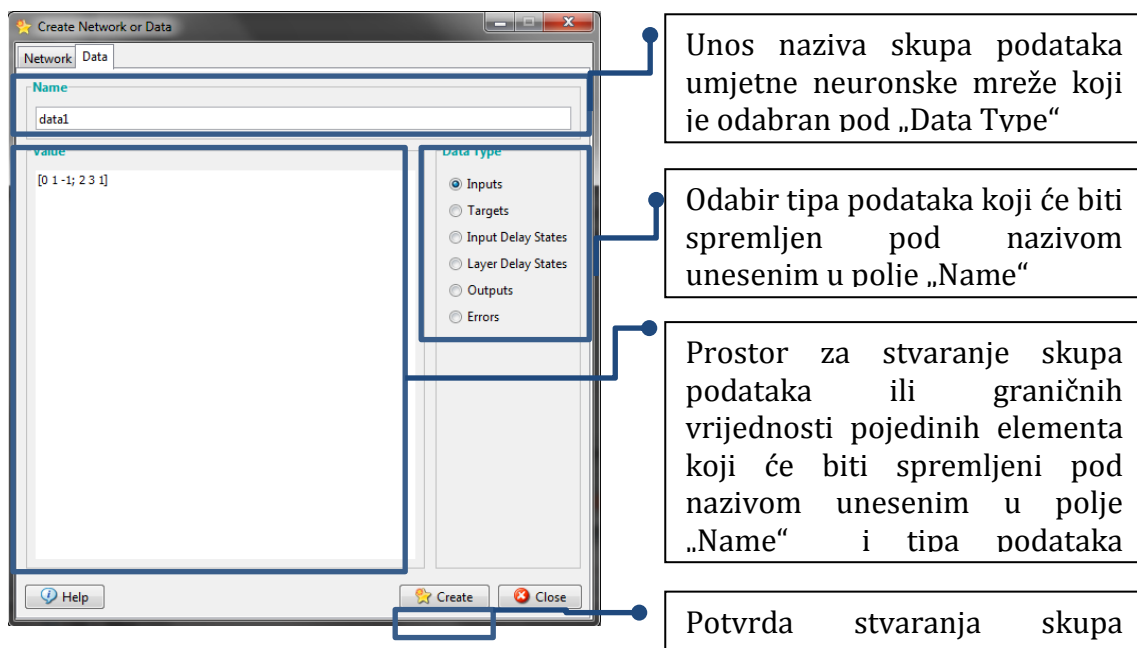
Slika 5. Početni prozor i osnovne funkcije alata „Neural Network Tool“

Ukoliko se želi stvoriti nova umjetna neuronska mreža ili potrebne podatke za nju, odabrat ćemo opciju „New...“ na početnom prozoru alata „Neural Network Tool“. Otvorit će se prozor „Create Network or Data“ sa početnom aktivnom karticom „Network“. Izgled i osnovne funkcije „Create Network or Data“ sa početnom aktivnom karticom „Network“ prikazane su na slici 6.



Slika 6. Pregled alata „Create Network or Data“ sa početnom aktivnom karticom „Network“

Izborom kartice „Data“ u prozoru alata „Neural Network Tool“ otvaraju se u istom prozoru nove opcije za stvaranje svih potrebnih podataka za stvaranje, učenje i evaluaciju umjetne neuronske mreže čije su početne karakteristike određene u opcijama kartice „Network“. Na slici 7. imamo osnovne funkcije i izgled kartice „Data“.



Slika 7. Pregled alata „Create Network or Data“ sa aktivnom karticom „Data“

## Rad na vježbi

Rad na vježbi sastoji se od kreiranja umjetne neuronske mreže pomoću „Network“ funkcije sa i bez argumenata, te generiranja Simulink modela iste. Riješit će se jednostavan problem množenja pomoću „Neural Network Tool“ alata kojim će se provesti učenje, evaluacija i simulacija rada umjetne neuronske mreže, te komentirati izlazne rezultate.

## Priprema za vježbu

- Proučiti predavanja vezana za umjetne neuronske mreže.
- Proučiti vježbe iz osnova MATLAB-a i Simulink-a.

## Stvaranje umjetne neuronske mreže pomoću funkcije „Network“

Stvorite unaprijednu neuronsku mrežu (engl. „*feed-forward neural network*“) pomoću „Network“ funkcijom bez argumenta pod imenom „mreza“. Strukturu unaprijedne neuronske mreže kreirajte na način da ima jedan ulaz, dva sloja (oba sloja postavite da imaju veze s biasima). Vezu ulaznih težina ostvarite sa prvim slojem iz prvog ulaza, a vezu težina slojeva ostvarite prema sloju dva od sloja jedan. Prvi sloj postavite da ima pet neurona, kod acikličnih mreža taj sloj se naziva skriveni sloj (engl. „*hidden layer*“). Drugi sloj postavite na samo jedan neuron iz razloga što unaprijedna neuronska mreža koristi zadnji sloj kao izlazni sloj (engl. „*output layer*“).

Za procesne funkcije ulaza koristite jednu koja će najvećem elementu u matrici strukture ulaza pridodati vrijednost „1“ i drugu koja će izbrisati kolonu sa istim vrijednostima. Inicijalizacijske funkcije u oba sloja postaviti na *Nguyen-Widrow* algoritam. Prijenosna funkcija prvog sloja neka bude log-sigmoidna funkcija, a drugog sloja linearna funkcija. Unesite četiri elementa iz tablice 6. kao elemente ulaza:

Tab. 6. Elementi ulaza

Prvi element	Drugi element	Treći element	Četvrti element
3	0	3	0
3	3	0	0

Prijedlog rješavanja problema u MATLAB komandom prozoru:

```
mreza = network; %stavaranje objekta „Neural Network“

mreza.numInputs = 1; %postavljanje broja ulaza na vrijednost jednog
ulaza
mreza.numLayers = 2; %postavljanje broja slojeva na vrijednost dva
sloja

mreza.biasConnect(1) = 1; %veza prvog sloja sa biasom postoji
mreza.biasConnect(2) = 1; %veza drugog sloja biasom postoji

mreza.inputConnect(1,1) = 1; %veza prvog sloja sa prvim ulazom
```

```

mreza.layerConnect(2,1) = 1; %veza drugog sloja prema prvom sloju

mreza.outputConnect = [0 1]; %veza drugog sloja sa izlazom

mreza.inputs{1}.exampleInput = [3 3; 0 3; 3 0; 0 0];%ulazni elementi
mreza.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
%procesne funkcije ulaza

mreza.layers{1}.size = 5; %broj neurona prvog sloja
mreza.layers{1}.transferFcn = 'tansig'; %prijenosna funkcija za prvi
sloj
mreza.layers{1}.initFcn = 'initnw'; % inicijalizacijska funkcija
prvog sloja

mreza.layers{2}.size = 1; %broj neurona drugog sloja
mreza.layers{2}.transferFcn = 'logsig'; %prijenosna funkcija za
drugi sloj
mreza.layers{2}.initFcn = 'initnw'; % inicijalizacijska funkcija
drugog sloja

mreza.initFcn = 'initlay'; %inicijalizacija svaki sloj „i“ prema
inicijalizacijskoj funkciji -net.layers{i}.initFcn.

```

Linije koda kojima se određuju veze sa težinama slojeva i veze sa težinama ulaza, te veze s biasima napišite u jednoj liniji koda koristeći polja.

Iskoristite funkciju „Network“ sa argumentima kako bi odredili strukturu umjetne neuronske mreže.

## Analiza svojstva umjetne neuronske mreže stvorene funkcijom „Network“

Inicijalizirajte početne težine i biase funkcijom “init”. Nakon inicijalizacije uz pomoć odgovarajućih naredbi u MATLAB radnom prozoru odredite vrijednosti svih postojećih biasa i težina slojeva i ulaza (u skladu sa ostvarenim vezama) te ih unesite u tablicu 7.

Tab. 7. Vrijednosti svih postojećih biasa i težina slojeva i ulaza

Vrsta vrijednosti umjetne neuronske mreže nakon inicijalizacije	Inicijalne vrijednosti
---	------------------------

„IW“ (Ulazne težine)

„LW“ (Težine sloja)

Prvi sloj	Drugi sloj
-----------	------------

„b“ (vektor biasa)

Objasnite razlog postojanja broja redaka i broja stupaca za svaku skupinu vrijednosti u tablici 7.

Odgovor:

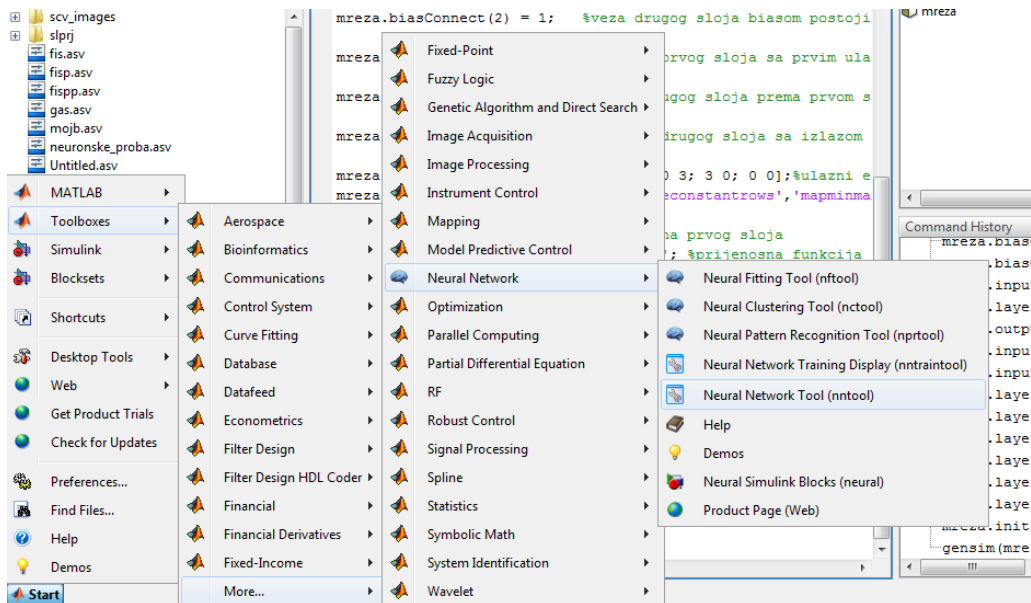
Generirajte Simulink model umjetne neuronske mreže objekta „mreže“, te ga komentirajte.

Komentar:

## Stvaranje umjetne neuronske mreže pomoću GUI alata „Neural Network Tool“

Stvorite unaprijednu umjetnu neuronsku mrežu koja će računati umnožak dva broja do 10. Ulazi su cijeli brojevi u intervalu [1, 5]. Stvorena aciklična umjetna neuronska mreža ima tri sloja, prvi sloj ima 12 neurona, a drugi 5. Prijenosna funkcija prva dva sloja je Log-sigmoidna funkcija, a trećeg linearna.

Alatu „Neural Network Tool“ pristupite odabirom sljedećih opcija, Start -> More... -> Neural Network -> Neural Network Tool (nntool), kako je prikazano na slici 8.

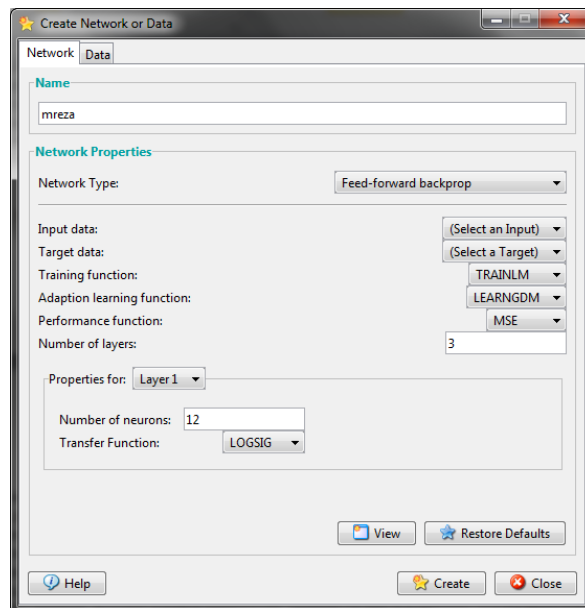


Slika 8. Način pristupa alatu „Neural Network Tool“

Najprije stvorite novu umjetnu neuronsku mrežu, odabiranjem opcije „New...“ u početnom prozoru „Neural Network Tool“, Nakon odabiranje spomenute opcije otvorit će se prozor „Create Network or Data“. U polje „Name“ upišite ime „mreza“. Alat „Neural Network Tool“ nije u mogućnosti stvoriti precizne veze između slojeva i ulaza, te dodavati veze s biasima u slojevima, stoga ima gotove tipove umjetnih neuronskih mreža koje se mogu izabrati pomoću padajućeg izbornika „Network Type“. Za „Network Type“ odaberite iz padajućeg izbornika acikličnu umjetnu neuronsku mrežu sa povratnim rasprostiranjem („Feed-Forward backprop“). Postavite broj slojeva („Number of Layers“) na tri u prostor svojstva „Properties for:“ za sloj 1, te potvrdite tipkom Enter. Također, postavite broj neurona „Number of neurons:“ na „12“, a prijenosnu funkciju „Transfer Function“ iz padajućeg izbornika na Log-sigmoidnu funkciju „LOGSIG“. Sloj 2

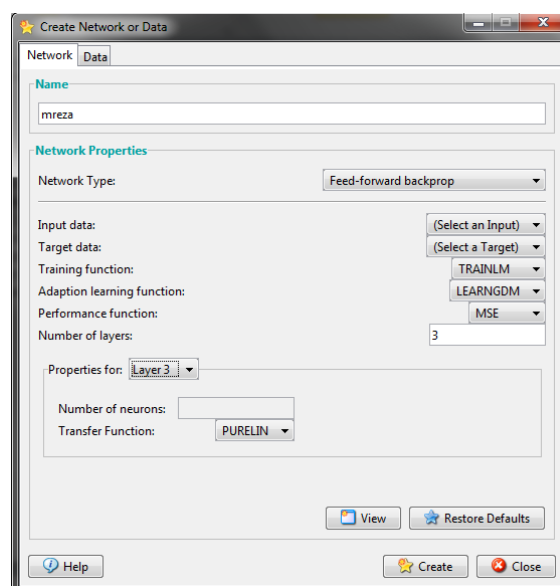


neka ima istu prijenosnu funkciju kao i sloj 1, ali po zadatku neka ime „5“ neurona. Konačni pregled spomenutih postavki sloja 1 možete vidjeti na slici 9.



Slika 9. Pregled postavki prozora „Create Network or Data“ sa prikazom svojstva sloja 1

Napomenuli smo ranije kako alat „Neural Network Tool“ ima već gotove tipove umjetnih neuronskih mreža, prilikom promjene svojstva u sloj 3 učit ćete da je nemoguće promijeniti broj neurona. Broj neurona u drugom sloju nemoguće je promijeniti iz razloga što je tako određen tip aciklične umjetne neuronske mreže sa povratnim rasprostiranjem u „Neural Network Tool“ alatu (postavljen je na vrijednost jedan u ovom slučaju). U svojstvima trećeg sloja moguće je promijeniti prijenosnu funkciju, pa ju promijenite u linearnu funkciju koju u padajućem izborniku možete pronaći pod „PURELINE“ opcijom. Konačni pregled spomenutih postavki za sloj 3 u prozoru „Create Network or Data“ prikazan je na slici 10.



Slika 10. Pregled postavki prozora „Create Network or Data“ sa prikazom svojstva sloja 3

Kako bi mogli izvršiti spremanje umjetne neuronske mreže potrebno je odrediti ulazne skup podatka i odgovarajući skup podataka ciljeva. Spomenute podatke dobit ćemo iz tablice 8. Ulazni podatci će biti množenik i množitelj, a ciljevi njihov produkt.

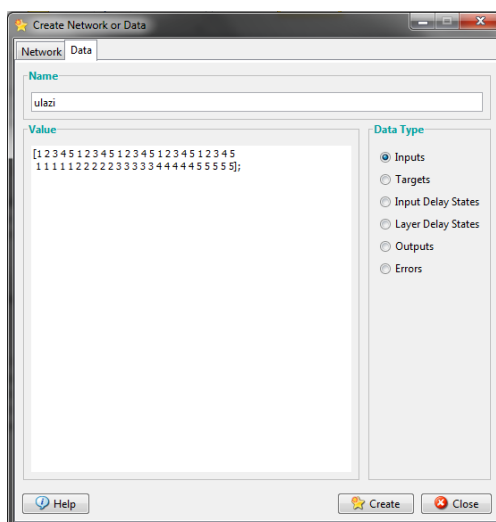
Tab. 8. Ulazne i ciljne vrijednosti

Množenik																								
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Množitelj																								
1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5
Produkt																								
1	2	3	4	5	2	4	6	8	10	3	6	3	12	15	4	8	12	16	20	5	19	15	20	25

Određivanje ulaznih podataka vrši se tako da se u prozoru „Create Network or Data“ izabere kartica „Data“, te aktivira vrijednost tipa podataka „Inputs“ pod opcijama „Data Type“. U prostor za unos vrijednosti odabranog tipa podataka „Value“ unesite vrijednosti množenika i množitelja iz tablice 4.8. u obliku polja:

```
ulazi = [1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5];
```

Kao zadnji potrebnii podatak unesite pod ime ulaznih vrijednosti „Name“ – „ulazi“. Konačni izgled prozora „Create Network or Data“ sa aktivnom karticom „Data“ nakon unosa svih podataka za izradu vrijednosti ulaza prikazan je na slici 11.

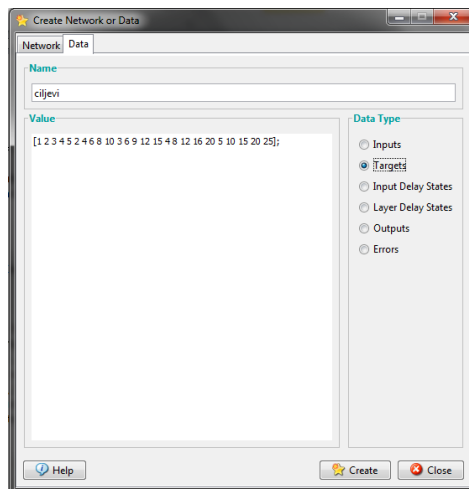


Slika 11. Konačni izgled prozora „Create Network or Data“ kartice „Data“ nakon unosa svih podataka za izradu vrijednosti ulaza

Nakon što ste unijeli sve potrebne podatke odaberite opciju „Create“, tom radnjom potvrđujete stvaranje skupa ulaznih podataka „ulazi“. Stvorite na isti način i skup podataka tipa ciljeva (engl. „Targets“) pod imenom „ciljevi“. Vrijednosti ove varijable odredit će se umnošcima podataka tablice 4.8. u obliku polja:

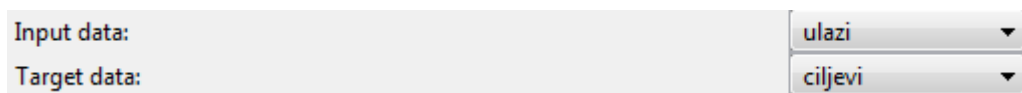
```
ciljevi = [1 2 3 4 5 2 4 6 8 10 3 6 9 12 15 4 8 12 16 20 5 10 15 20 25];
```

Konačni izgled prozora „Create Network or Data“ kartice „Data“ nakon unosa svih podataka za izradu vrijednosti ciljeva prikazan je na slici 12.



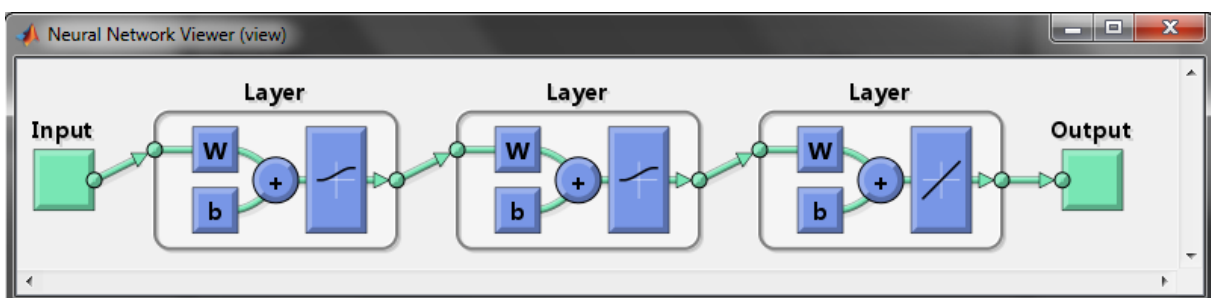
Slika 12. Konačni izgled prozora „Create Network or Data“ kartice „Data“ nakon unosa svih podataka za izradu vrijednosti ciljeva

Nakon što su unijeti svi potrebne podaci, potrebno je odabirom opcije „Create“ potvrditi stvaranje skupa podataka „ciljevi“. Nakon izrade skupa podataka „ciljevi“ i „ulazi“ vratite se na kraticu „Network“ prozora „Create Network or Data“. Postavite za ulazne podatke (engl. „Input data“) umjetne neuronske mreže prethodno stvoreni skup podataka „ulazi“, a za očekivane izlaze (engl. „Target data“) skup podataka „ciljevi“. Učiniti ćete to tako da pod karticom „Network“ u padajućem izborniku „Input data:“ izaberete opciju „ulazi“, a u padajućem izborniku „Target data:“ izaberite opciju „ciljevi“ kao što je prikazano na slici 4.13.



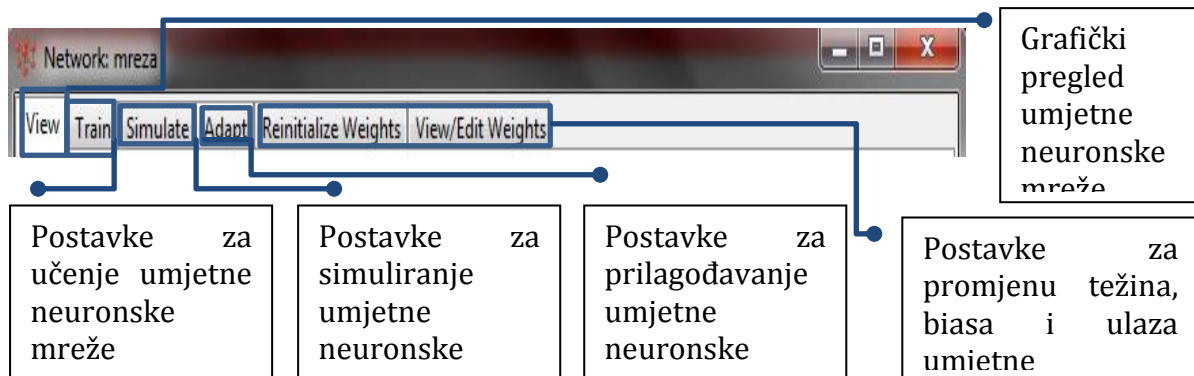
Slika 13. Prikaz odabira „Input data“ i „Target data“ iz kartice „Network“

Postavljanjem prethodno stvorenih skupova podataka za ulaze i ciljeve, omogućit ćete grafički pregled strukture umjetne neuronske mreže. Pregledajte strukturu stvorene umjetne neuronske mreže tako da odaberete opciju „View“. Otvorit će se prozor „Neural Network Viewer“ kako je prikazano na slici 14.



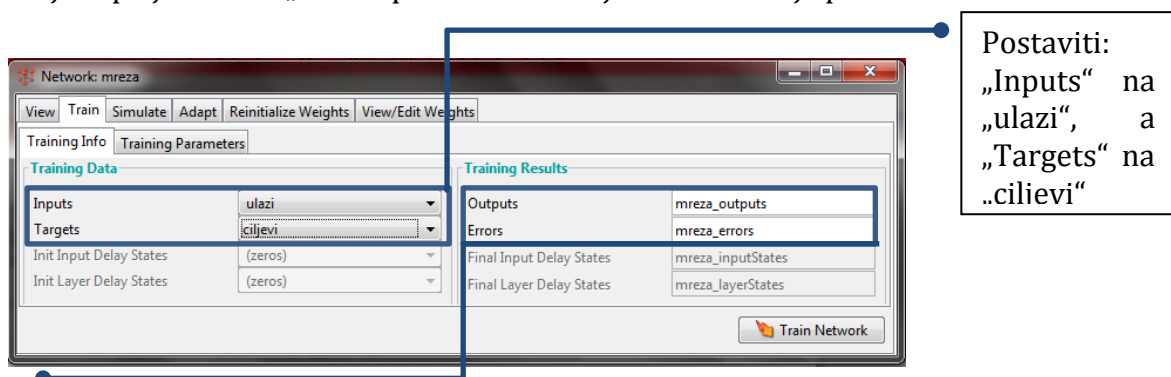
Slika 14. Prozor „Neural Network Viewer“

Spremite stvorenu umjetnu neuronsku mrežu. To ćete učiniti odabirom opcije „Create“ u prozoru „Network or Data“ kartice „Network“. Nakon odabira opcije „Create“ nemojte zatvoriti prozor „Network or Data“. U početnom prozoru alata „Neural Network Tool“ pod prostorom „Networks“ prikazat će se stvorena umjetne neuronska mreža „mreza“. Dvostrukim klikom lijeve tipke miša na ime „mreza“ otvorit će se prozor za detaljniji pregled stvorene umjetne neuronske mreže. Na slici 15. su objašnjenje glavne karakteristike umjetne neuronske mreže „mreza“ prozora „Network: mreza“.



Slika 15. Glavne karakteristike alatne trake prozora „Network: mreza“

Najprije provedite učenje umjetne neuronske mreže. To ćete učiniti odabirom kartice „Train“. U ovim vježbama nećemo detaljnije ulaziti u samu problematiku učenja, već za sada je potrebno znati da je primarni cilj učenja prilagoditi težinske vrijednosti umjetne neuronske mreže, te s obzirom na promijenjene težinske vrijednosti usporediti izlaz umjetne neuronske mreže s ciljevima, tj. odrediti stupanj greške rada umjetne neuronske mreže (engl. „Errors“). Promjena težinskih vrijednosti provodi se dok se ne postigne prihvatljiva razlika između ciljeva i izlaza, kako bi umjetna neuronska mreža odgovorila ubuduće i na ulaze koji nisu bili u skupu ulaza koji je bio predstavljen u učenju. Opcije kartice „Train“ postavite na vrijednosti kako je prikazano na slici 16.

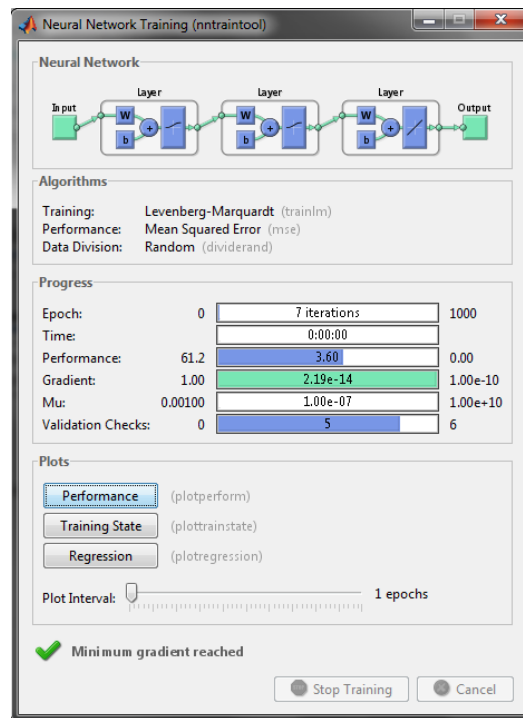


Postaviti rezultate učenja „Training Results“, da se izlazi („Outputs“) umjetne neuronske mreže spremaju pod imenom „mreza\_outputs“, a greške („Errors“) u

Slika 16. Postavke kartice „Train“

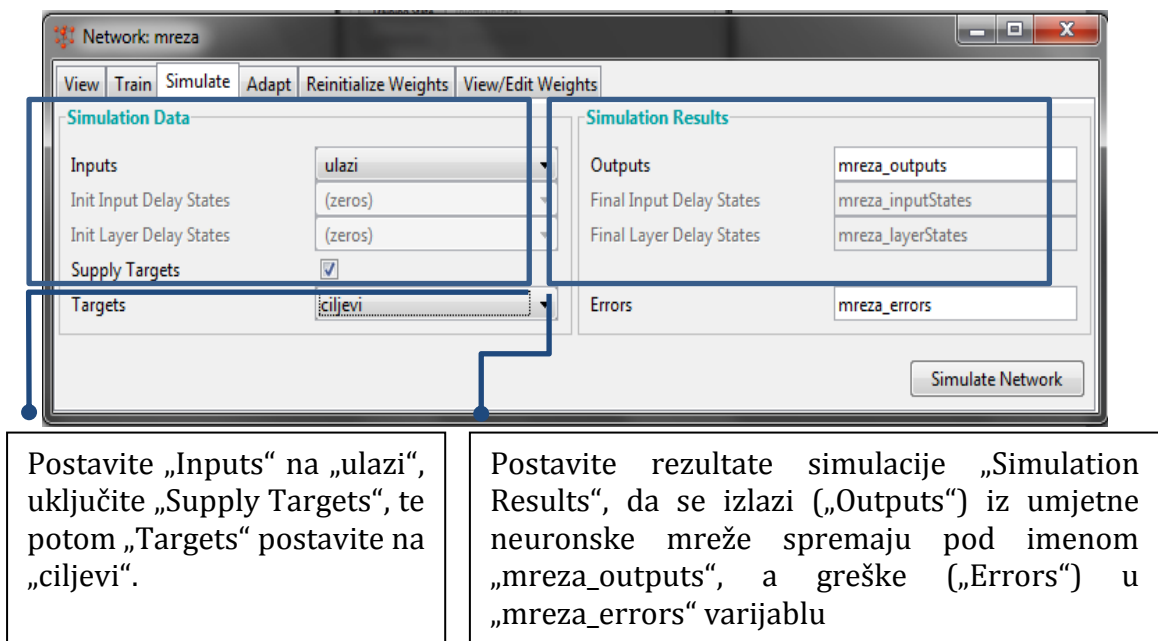
Nakon što su unijete prikazane postavke, potrebno je odaberati opciju „Train Network“. Time se pokreće alat „Neural Network Training“. Alat ćete detaljnije proučiti na

slijedećim vježbama, a zasada jednostavno pričekajte dok se u daljnjem dijelu „Neural Network Training“ prozora ne pojavi obavijest „Minimum gradient reached“ kao što je prikazano na slici 17., to znači da je proces učenja završen, te se može zatvoriti prozor.



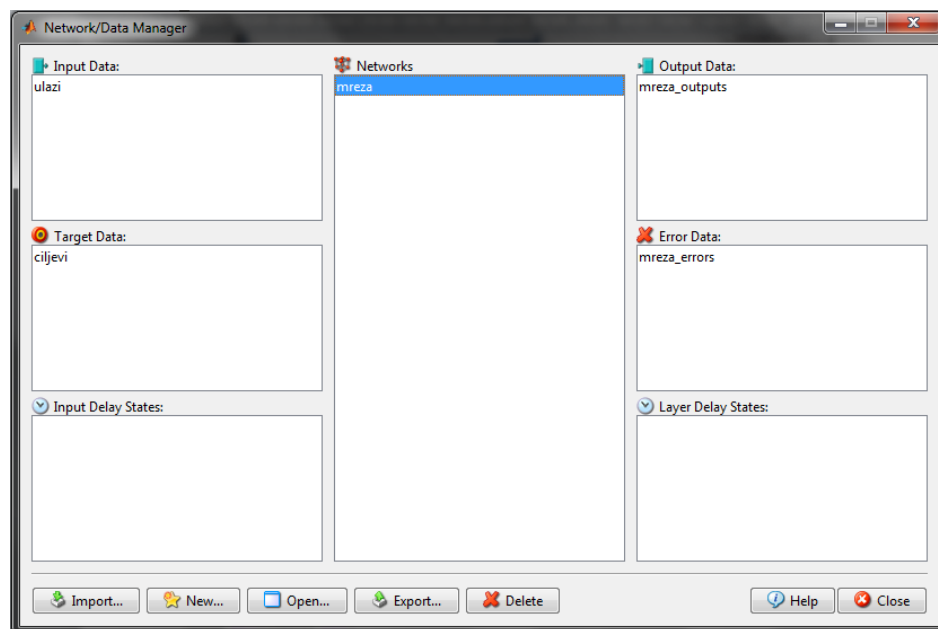
Slika 17. Izgled „Neural Network Training“ alata nakon završetka učenja

Sada će se stimulirati rad umjetne neuronske mreže „mreža“. To se može učiniti odabirom kartice „Simulate“. Parametre kartice „Simulate“ potrebno je podesiti kako je prikazano na slici 18.



Slika 18. Postavke kartice „Simulation“

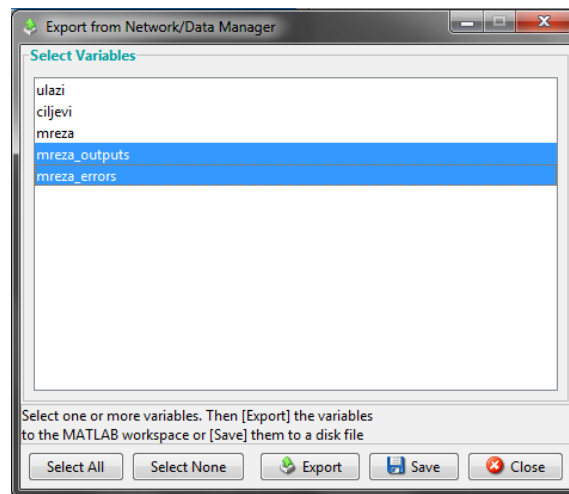
Nakon postavljanja navedenih parametara odaberite opciju „Simulate Network“, potom možemo minimizirati prozor „Network:mreze“, Pogledajte početni prozor za stvaranje umjetne neuronske mreže, njegov sadašnji izgleda prikazan je na slici 19.



Slika 19. Prikaz početnog prozora za stvaranje umjetne neuronske mreže nakon njenog stvaranja, učenja i simulacije rada

Pregledajte rezultate simulacije „Output Data“, dvostrukim klikom lijeve tipke miša na „mreza\_outputs“, a greške „Error Data“ dvostrukim klikom lijeve tipke miša na

„mreza\_errors“. Izvezite ta dva skupa podataka u MATLAB Workspace, tako da odaberete opciju „Export...“. Otvorit će se slijedeći prozor na slici 20.

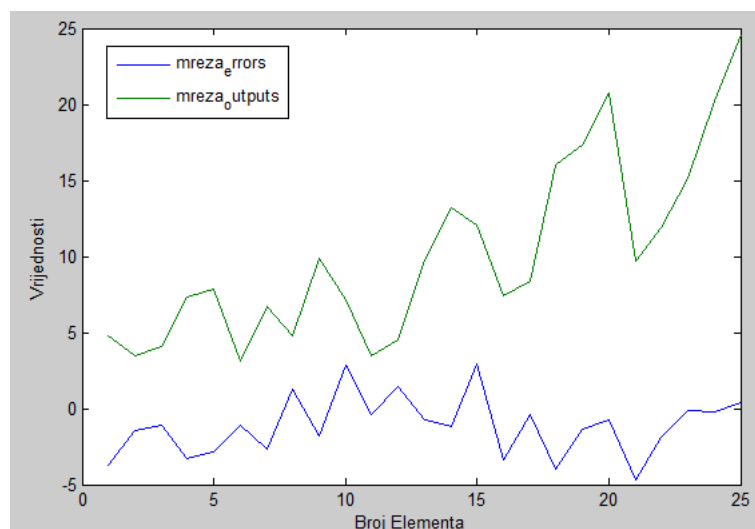


Slika 20. Prikaz prozora za izvoz elemenata umjetne neuronske mreže

Odaberite „mreza\_outputs“ i „mreza\_errors“ držeći tipku „Ctrl“ i pritiskom na spomenute skupove podataka. Zatim odaberite opciju „Export“. Ta akcija će prebaciti odabrane skupove podataka u MATLAB Workspace. Kako bi grafički vidjeli odnos ta dva skupa podataka koristite graf sa idućim linijama koda:

```
plot(mreza_errors, 'DisplayName', 'mreza_errors', 'YDataSource', 'mreza_
errors');
hold all;
plot(mreza_outputs, 'DisplayName', 'mreza_outputs', 'YDataSource', 'mrez
a_outputs');
hold all;;hold off;
figure(gcf);
xlabel('Broj Elementa');
ylabel('Vrijednosti');
```

Graf bi trebao izgledati kao na slici 21.



Slika 21. Graf odnosa skupova podataka „mreze\_errors“ i „mreza\_outputs“

## Zaključak

Za problem množenja, stvorite više umjetnih neuronskih mreža sa različitom strukturom (promijenite broj slojeva, broj neurona u pojedinim slojevima ili funkcije prijenosa) u „Neural Network Tool“ alatu. Stvorite graf grešaka svih stvorenih umjetnih neuronskih mreža, te na temelju njih komentirajte odnos strukture naspram greške rada.