





```
siet = newff(P,T,5,'logsig','purelin','trainlm'),
```

kde  $P$  je signál vstupov,  $T$  je signál výstupov, 5 je počet skrytých neurónov, *logsig* je nelineárna aktivačná funkcia v skrytej vrstve, *purelin* je lineárna aktivačná funkcia vo výstupnej vrstve, *trainlm* je metóda tréningu UNS (Levenberg-Marquardt) a *siet* je objekt v Matlabe, ktorý reprezentuje štruktúru UNS.

V aktuálnej verzii Matlabu (2008a) už netreba zadávať rozsahy vstupných údajov a počet neurónov vo výstupnej vrstve, ako to bolo potrebné v predchádzajúcich verziách, čím sa práca s NNT zjednodušuje.

Po vytvorení štruktúry UNS môžeme nastaviť (prístupujeme k objektu *siet* takzvanou bodkovou konštrukciou) ďalšie parametre:

```
siet.performFcn='sse'; % Suma štvorcov odchýlok
siet.trainParam.goal=0.000001; % Ukončovacia podmienka na chybu SSE
siet.trainParam.show=20; % Frekvencia zobrazovania priebehu chyby
siet.trainParam.epochs=2000; % Max. počet tréningových cyklov (epoch)
siet.trainParam.mc=0.95; % Momentum konstanta
```

### Učenie siete

Pomocou tréningového algoritmu (s učiteľom – náš prípad, bez učiteľa) sa modifikujú parametre neurónovej siete tak, aby sa dosiahla čo najlepšia aproximácia medzi skutočným výstupom siete a žiadaným výstupom siete. Toto učenie siete sa nazýva učením off-line, pretože sa vykonáva bez toho, aby bola neurónová sieť v reálnom prostredí.

```
[nsiet,tr,Y,E,Pf,Af] = train(siet,P,T),
```

kde *siet* je UNS, ktorá sa bude tréňovať,  $P$  sú vstupy,  $T$  sú výstupy, *nsiet* je natrénovaná UNS, *tr* sú parametre tréningu,  $Y$  je výstup z natrénovanej siete pri vstupe  $P$  a  $E$  je chyba siete.

Ak chceme ošetriť jav pretrénovania (UNS sa kvalitne naučí na tréningových údajoch, ale na testovacích vykazuje zlé výsledky), možno do tréningu pridať validačné údaje:

```
[nsiet,tr,Y,E,Pf,Af] = train(siet,P1,T1,P2,T2)
```

V tomto prípade prebieha učenie na  $P1$ ,  $T1$  a počas učenia sa testuje kvalita UNS na  $P2$ ,  $T2$ . Ak chyba na  $P2$  a  $T2$  narastie, znamená to ukončenie tréningového procesu.

### Testovanie siete

V tomto kroku sa otestuje schopnosť siete adaptovať sa na nové údaje z tzv. testovacej množiny vstupných údajov.

```
[Y2] = sim(nsiet, P2)
```

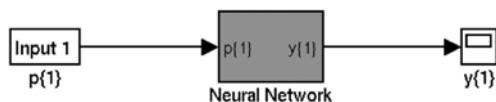
Tento krok je dôležitý, lebo sa testuje zovšeobecňovacia schopnosť UNS na nových „neznámych“ údajoch.

### Implementácia (nasadenie) siete

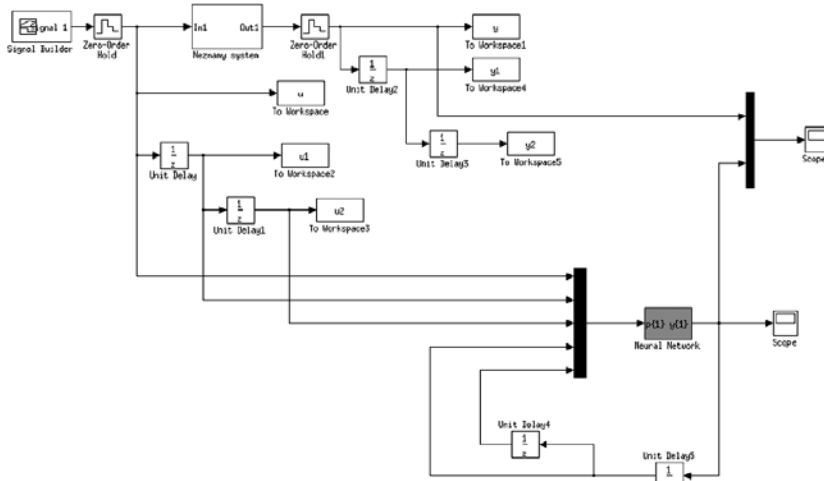
Implementácia siete sa môže realizovať už ako nemenná, keď sa parametre siete nemenia, alebo adaptabilná, keď sa nastavenie siete mení v reálnom čase – a teda sieť sa učí aj v reálnom prostredí. Tento druh učenia sa nazýva on-line. Na ďalšie použitie vytvorenej UNS možno vygenerovať blok do Simulinku, ktorý reprezentuje natrénovanú neurónovú sieť.

```
gensim(nsiet);
```

Na obr. 3 je zobrazený blok po spustení príkazu *gensim*.



Obr.3 Exportovaný model UNS v Simulinku



Obr.4 Porovnanie časových priebehov modelovaného systému a UNS

Následne môžeme porovnať výstup z pôvodného systému a modelu UNS (obr. 4).

Neural Network Toolbox predstavuje nástroj na modelovanie nelineárnych systémov pomocou neurónových sietí. Ide o moderný prístup neraz využívaný v oblasti identifikácie nelineárnych dynamických systémov. NNT je plne integrovaný do prostredia Matlab. Jednak sa opiera o jeho výpočtové jadro, využíva objektový spôsob parametrizácie a tiež je previazaný so Simulinkom. Na jednoduchšiu prácu možno využiť grafické rozhranie, kde vieme len pomocou myši vytvoriť natrénovanú umelú neurónovú sieť. Pokročilé funkcie je však výhodnejšie realizovať pomocou príkazov.

Článok vznikol s podporou APVV-99-045805 a VG-3100-KOZ-SK1.

### Literatúra

- [1] <http://www.humusoft.cz/matlab/modules>
- [2] Neural Network Toolbox – Getting Started Guide, The MathWorks 2007
- [3] <http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/>
- [4] Interné materiály (prednášky a cvičenia) k predmetu Pokročilá teória automatického riadenia
- [5] Interné materiály (prednášky a cvičenia) k predmetu Inteligentné systémy
- [6] KÖRÖSI, L. (2003): Metódy identifikácie a riadenia nelineárnych procesov pomocou neurónových sietí. Bratislava: FEI STU KASR, s. 1 – 60.
- [7] KOZÁK, Š., KÖRÖSI, L. (2002): Využitie umelých neurónových sietí v praxi a v priemysle. Konf. SSKI, Trebišov.
- [8] KAJAN, S., SARVAŠ, T., SEKAJ, I. (2008): Comparison of some neural control structures for nonlinear systems. International Conference CYBERNETICS AND INFORMATICS. Ždiar

Ing. Martin Foltin, PhD.  
Ing. Ladislav Körösi

Slovenská technická univerzita v Bratislave  
Fakulta elektrotechniky a informatiky  
Ústav riadenia a priemyselnej informatiky  
Ilkovičova 3, 812 19 Bratislava  
e-mail: martin.foltin@syprin.sk  
ladislav.korosi@stuba.sk