

VYUŽITÍ NEURONOVÝCH SÍTÍ PROSTŘEDÍ MATLAB K PREDIKCI HODNOT NÁKLADŮ PRO ELEKTRICKÉ OBLOUKOVÉ PECE

V. Hon

VŠB – TU Ostrava, FEI, K455, 17. Listopadu 15, Ostrava – Poruba, 70833

Abstrakt

Neuronová síť (dále jen NS) je výkonná metoda, která se používá k modelování vztahu mezi vícerozměrnou vstupní proměnnou x a vícerozměrnou výstupní proměnnou y . Předložený článek se zabývá využitím NS pro predikci hodnot nákladů v podmínkách elektrické obloukové pece. V tomto předloženém článku pak bude dále prezentováno několik stávajících modelů NS vytvořených v prostředí MATLAB a budou zde navzájem porovnány.

1. Úvod

Neuronové sítě (dále jen NS) mají pro predikci výhodu v tom, že jsou schopny učit se na příkladech a po naučení jsou schopny vystihnout skryté i silně nelineární závislosti a to i za přítomnosti šumu v trénovací množině. Nevýhoda spočívá v tom, že se mohou naučit závislost platnou pouze v určitém období a chybu předpovědi zpravidla nelze odhadnout. [1]

V podniku, pro který byl model NS vytvořen, se snaží minimalizovat hodnoty nákladů pro elektrické obloukové pece v údobí tavení a hledají optimální počet taveb na vyzdívku stěn a vyzdívku víka. K tomu potřebují vytvořit model, který by byl schopen napodobit chování elektrické obloukové pece (Obr.1.) a umožňoval by tak další možné úpravy a práce pro následnou minimalizaci nákladů.

V tomto předloženém článku pak bude dále prezentováno několik stávajících modelů NS vytvořených v prostředí MATLAB a budou zde navzájem porovnány.



Obr.1. Průřez elektrickou obloukovou pecí

2. Trénovací množina

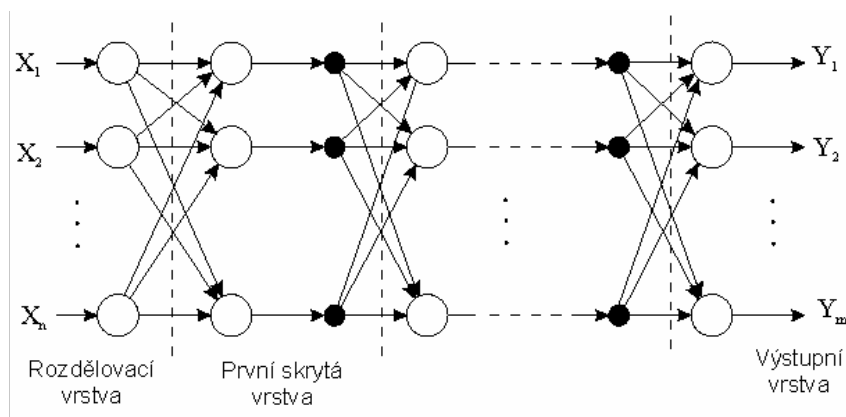
Výběr trénovací množiny je velice důležitý pro sestavení modelu NS a pro jeho chování. V tomto případě prováděl podnik, pro který je model realizován, několik měsíců důkladné analýzy a selekce měřených dat z elektrické obloukové pece, než jsme dostali do rukou finální verzi trénovací množiny.

Přesto se nám podařilo během tvorby modelu a učení NS odhalit v datech chyby, které bylo potřeba opravit. Přítomnost těchto chyb měla zásadní vliv na chování modelu NS. Nakonec se nám přece jen podařilo vytvořit takovou trénovací množinu, na kterou se nám NS podařilo naučit tak, že predikuje hodnoty nákladů s potřebnou přesností.

Trénovací množina má v současnosti 7 vstupních a jednu výstupní proměnnou. Původní plán počítal s počtem 13 vstupních a jednou výstupní proměnnou. Vstupní proměnné byly později redukovány na 7, s ohledem na jejich relevantnost. Model s touto trénovací množinou byl pak označen jako *VariantaIVa*, respektive *VariantaIVnormovana*.

3. Topologie sítě

Propojení neuronů může být libovolné. V praxi však používáme tzv. síť vícevrstvé, v nichž jsou neurony uspořádány do několika vrstev nad sebou. První vrstva je vrstvou vstupní, poslední je vrstva výstupní. Mezilehlé vrstvy se nazývají vrstvy skryté (Obr.2.). [1] [2]



Obr.2. Příklad možné topologie vícevrstvé NS

Jakmile byla určena konečná podoba trénovací množiny, mohla být vytvořena podoba modelu NS vhodná pro řešení daného problému. Nastavení topologie sítě je velice komplikovaná záležitost a je při tom potřeba mít jisté zkušenosti. V našem případě pro naše modely byla použita topologie dvouvrstvé NS $\{12000, 1\}$ pro první model (nenormovaný, *VariantaIVa*) a $\{12, 1\}$ pro druhý model (normovaný, *VariantaIVnormovana*). První číslo označuje počet neuronů v první vrstvě a druhé počet neuronů ve výstupní vrstvě. Ostatní pokusné topologie sítí se zatím ukázaly jako nevýhodné pro daný problém. [3]

Jak již bylo napsáno výše, pro model označený jako *VariantaIVa*, byly tedy vytvořeny dvě verze modelu, jeden označený jako *VariantaIVa* a druhý jako *VariantaIVnormovana*. Druhý model se liší od prvního tím, že všechny hodnoty vstupních i výstupních proměnných jsou převedeny, normovány, na interval $\langle 0, 10 \rangle$ podle vzorce (1). Poté NS pracuje s normalizovanými proměnnými a výsledné proměnné jsou opět převedeny, denormovány, na interval původních proměnných $\langle X_{min}, X_{max} \rangle$ podle vzorce (2).

Oproti tomu první model pracuje s nenormovanými proměnnými. Z toho také plyne odlišný počet neuronů v jednotlivých vrstvách.

$$X_n = \frac{10 * (X - X_{\min})}{(X_{\max} - X_{\min})}, X_n \in \langle 0,10 \rangle, X \in \langle X_{\min}, X_{\max} \rangle \quad (1)$$

$$X = \frac{X_n * (X_{\max} - X_{\min})}{10}, X_n \in \langle 0,10 \rangle, X \in \langle X_{\min}, X_{\max} \rangle \quad (2)$$

4. Průběh učení

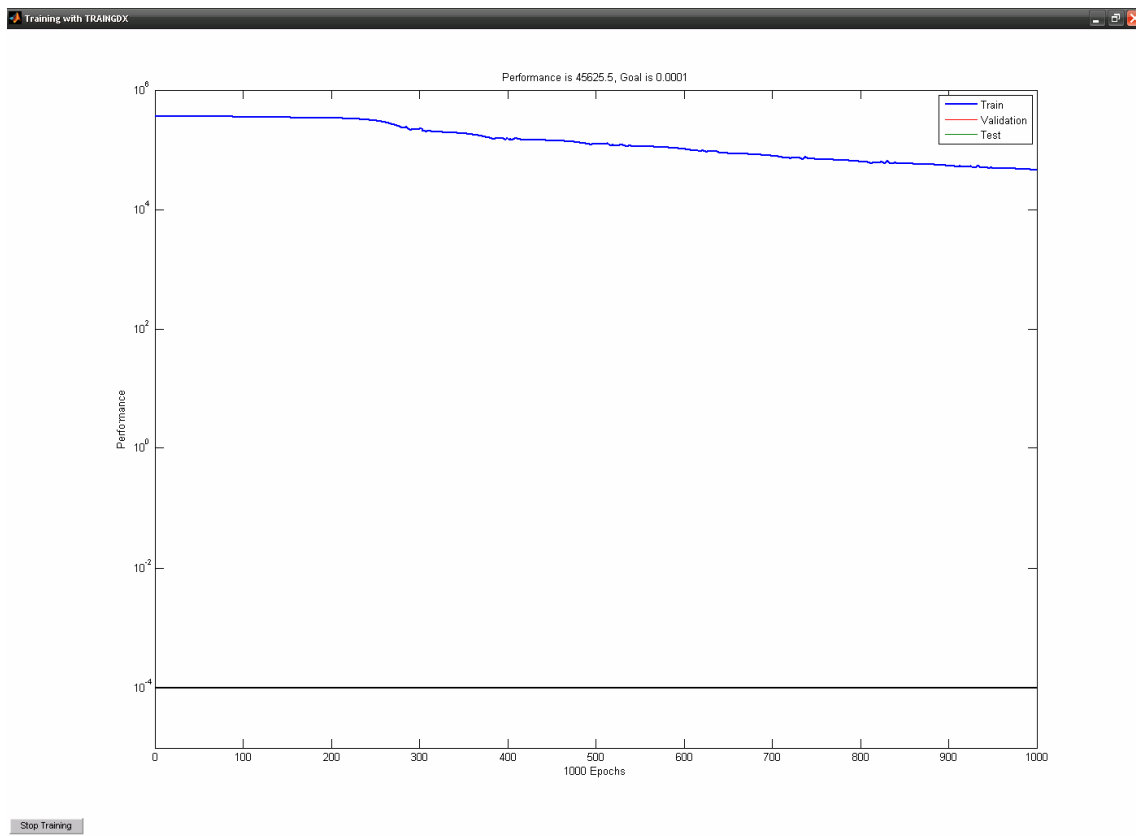
Způsob transformace vstupních dat určuje právě fáze učení (adaptace NS), založená na využití vzorů (příkladů) požadovaného chování, tedy trénovací množiny. Proces hledání optimálního nastavení struktury i parametrů NS k dosažení požadované transformace vstupní množiny do množiny výstupní (požadované chování NS) se nazývá adaptace sítě. Pro adaptaci vícevrstvé NS se spojitými *perceptrony* byla vyvinuta a často je používána adaptační metoda nazvaná *back-propagation* (BP), metoda zpětného šíření, která byla využita i v našich modelech NS (Obr.3.). [1] [2] [3]

<pre>%VariantaIVa %Volba struktury neuronove site Layers=[12000,1]; %Nastaveni aktivacnich funkci TransFcns={'logsig','purelin'}; %Vyber trenovaci funkce BTF='traingdx'; %Vyber ucici funkce BLF='learngdm'; %Vyber vyhodnocovaci funkce PF='mse'; %Zmena parametru trenovani net.trainParam.epochs=1000; net.trainParam.lr=0.000000001; net.trainParam.show=50; net.trainParam.goal=0.0001; %Trenovani site [net,tr]=train(net,IV,TV);</pre>	<pre>%VariantaIVnormovana %Volba struktury neuronove site Layers=[12,1]; %Nastaveni aktivacnich funkci TransFcns={'logsig','purelin'}; %Vyber trenovaci funkce BTF='traingdx'; %Vyber ucici funkce BLF='learngdm'; %Vyber vyhodnocovaci funkce PF='mse'; %Zmena parametru trenovani net.trainParam.epochs=100000; net.trainParam.lr=0.000000001; net.trainParam.show=50; net.trainParam.goal=0.0001; %Trenovani site [net,tr]=train(net,IV,TV);</pre>
--	---

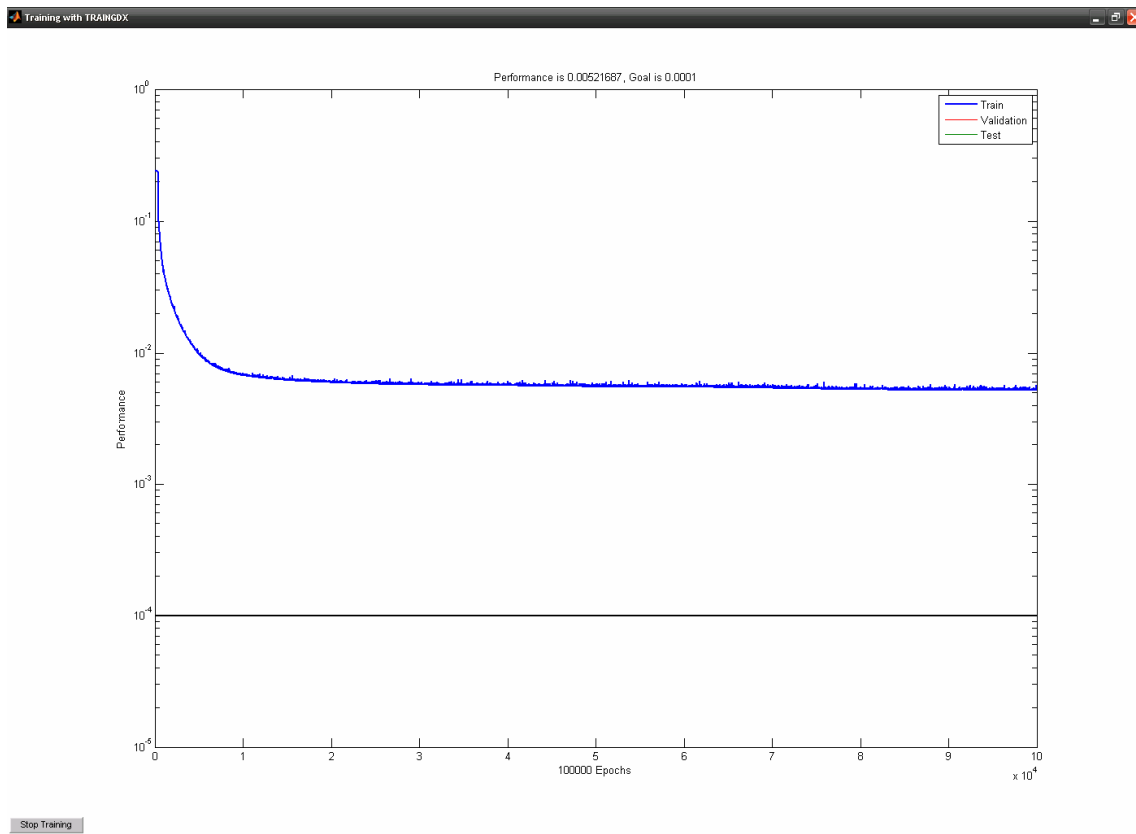
Obr.3. Porovnání parametrů NS v prostředí MATLAB

Pro oba modely byly použity nejprve standartní přednastavené parametry učení, potom byly tyto parametry optimalizovány. Samotné nastavení parametrů je nutno provést experimentálně a jedná se o experimentálně náročnou část práce, samotný výpočet je již celkem rychlý, ovšem záleží na počtu neuronů v jednotlivých vrstvách a na počtu běhů samotného učení.

Výsledky učení můžete vidět na následujících obrázcích (Obr.4. a Obr.5.). Jako vyhodnocovací funkce v obou případech byla použita střední kvadratická chyba (mean square error).



Obr.4. Průběh střední kvadratické chyby pro model NS nenormovaný



Obr.5. Průběh střední kvadratické chyby pro model NS normovaný

5. Simulace

Simulace funkčnosti modelu NS byla provedena následujícím způsobem. Pro oba modely, *VariantaIVa* a *VariantaIVnormovana*, byla použita stejná trénovací množina. Soubor dat byl rozdělen na dvě části, dostali jsme 650 hodnot pro každou proměnnou v obou částech, tedy dohromady 1300 hodnot. Na první polovinu byla NS naučena a na druhé polovině byla testována schopnost modelu NS predikovat.

Byl vytvořen soubor v prostředí MATLAB (m-file), respektive dva soubory (*VariantaIVa* a *VariantaIVnormovana*), ve kterých pak pomocí definovaných funkcí byla vytvořena neuronová síť s požadovanými parametry a byla provedena simulace chování modelu.

Při pozorování průběhů výstupní veličiny se jeví první model, *VariantaIVa*, jako lepší, což platí pro průběhy jak natrénované, tak predikované. Statisticky, podle počtu hodnot absolutní chyby (Obr.6. a Obr.7.), je ovšem druhý model, *VariantaIVnormovana*, lepší a počet hodnot predikované proměnné v zadané toleranci je mnohem vyšší. Také rychlost výpočtu byla mnohem vyšší u druhého modelu, což je dáno především počtem neuronů v jednotlivých vrstvách.

6. Závěr

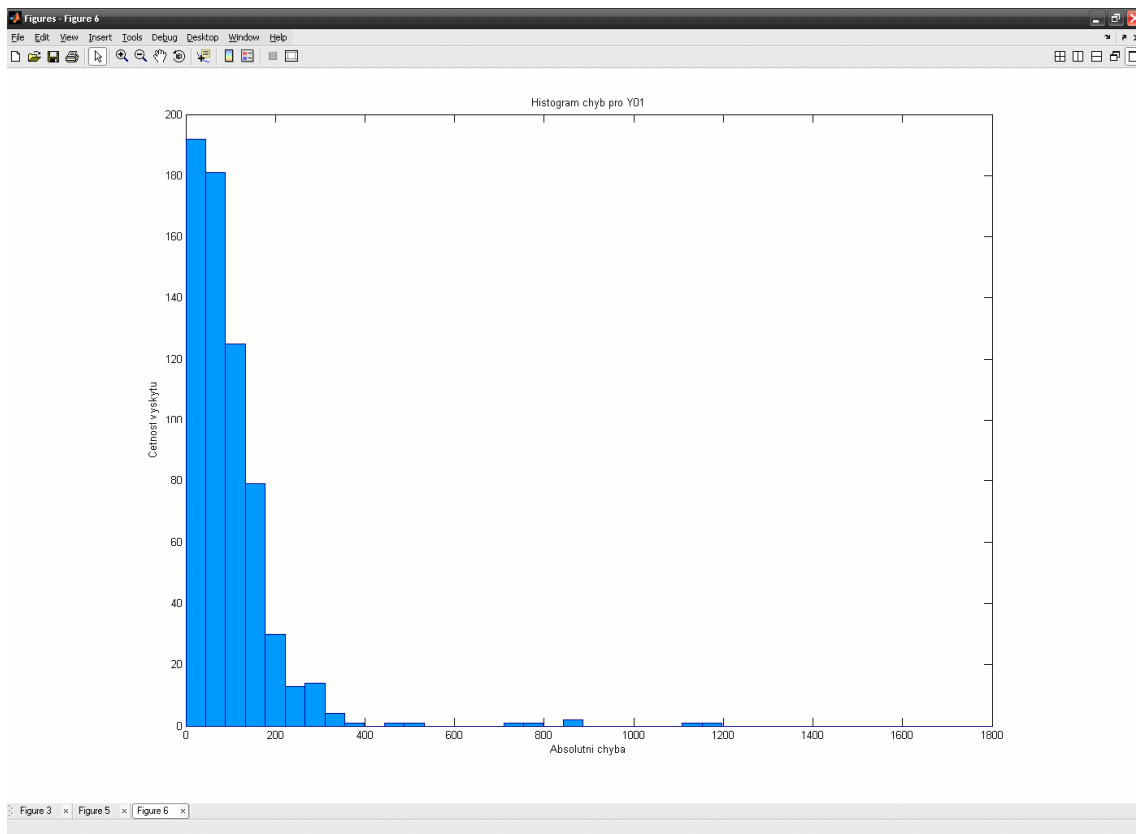
V tomto předloženém článku bylo prezentováno několik stávajících modelů NS vytvořených v prostředí MATLAB a byly zde navzájem porovnány. Dosavadní dosažené výsledky byly také podnikem hodnoceny jako velice zajímavé a na další spolupráci a zdokonalení nejprogressivnějšího modelu se bude i v budoucnu nadále pokračovat. Modelování a simulace chování složité technologické soustavy prokázaly schopnost programového systému MATLAB řešit praktické problémy.

Poděkování

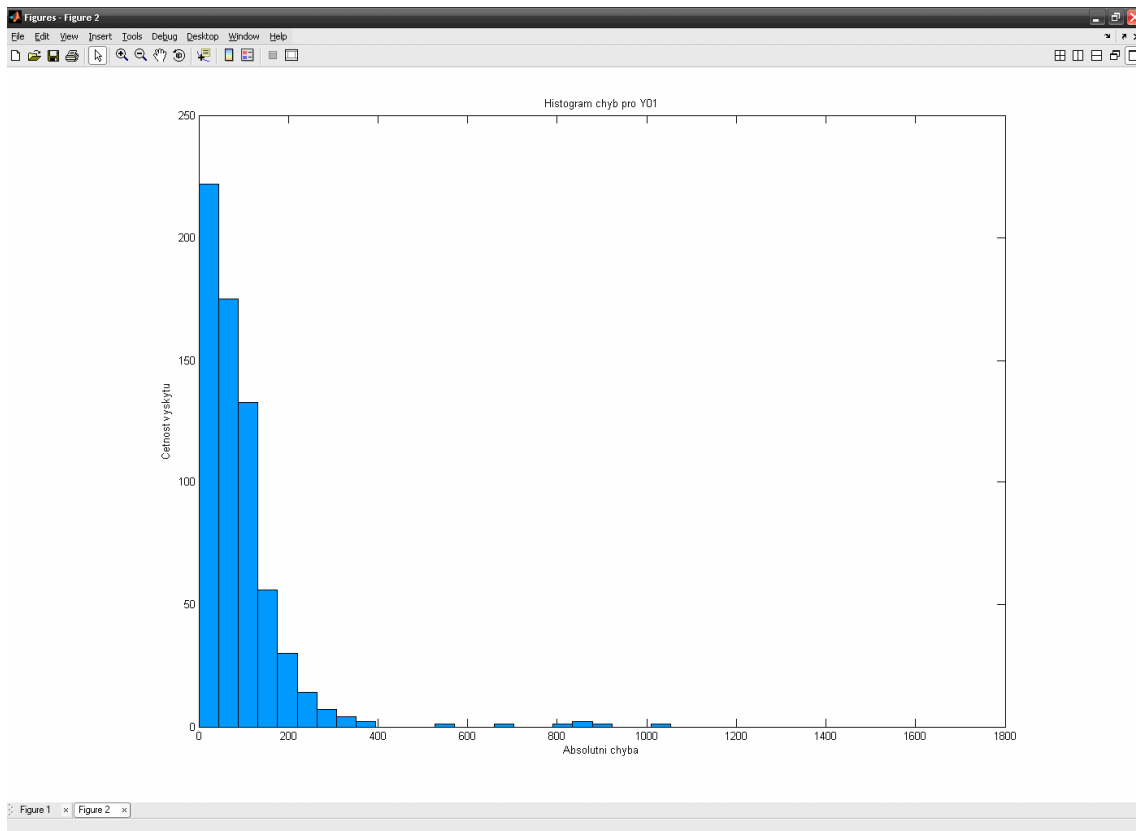
Tato práce vznikla za finanční podpory grantových projektů GAČR 102/05/H525: Racionalizace studia doktorského studijního programu na FEI VŠB-TU Ostrava a GAČR 102/06/1332: Výpočtová inteligence v řízení metalurgických procesů.

Literatura

- [1] M. Pokorný, *Umělá inteligence v modelování a řízení*, BEN, Praha 1996
- [2] I. Vondrák, *Umělá inteligence a neuronové sítě*, Skripta VŠB-TUO, Ostrava 2002
- [3] Referenční on-line příručka programového prostředí MATLAB.



Obr.6. Histogram rozložení absolutní chyby pro model NS nenormovaný



Obr.7. Histogram rozložení absolutní chyby pro model NS normovaný