

UČENÍ N-VRSTVÉ NEURONOVÉ SÍTĚ

V. Konečný, A. Matiašová, I. Rábová

Došlo: 8. června 2005

Abstract

KONEČNÝ, V., MATIÁŠOVÁ, A., RÁBOVÁ, I.: *Learning of N-layers neural network*. Acta univ. agric. et silvic. Mendel. Brun., 2005, LIII, No. 6, pp. 75–84

In the last decade we can observe increasing number of applications based on the Artificial Intelligence that are designed to solve problems from different areas of human activity. The reason why there is so much interest in these technologies is that the classical way of solutions does not exist or these technologies are not suitable because of their robustness. They are often used in applications like Business Intelligence that enable to obtain useful information for high-quality decision-making and to increase competitive advantage.

One of the most widespread tools for the Artificial Intelligence are the artificial neural networks. Their high advantage is relative simplicity and the possibility of self-learning based on set of pattern situations.

For the learning phase is the most commonly used algorithm back-propagation error (BPE). The base of BPE is the method minima of error function representing the sum of squared errors on outputs of neural net, for all patterns of the learning set.

However, while performing BPE and in the first usage, we can find out that it is necessary to complete the handling of the learning factor by suitable method. The stability of the learning process and the rate of convergence depend on the selected method.

In the article there are derived two functions: one function for the learning process management by the relative great error function value and the second function when the value of error function approximates to global minimum.

The aim of the article is to introduce the BPE algorithm in compact matrix form for multilayer neural networks, the derivation of the learning factor handling method and the presentation of the results.

neural networks, learning algorithm, back-propagation error, matrix form, learning rate, handling of learning rate

V posledních desetiletích lze zaznamenat značný nárůst aplikací pro řešení úloh z různých oblastí lidské činnosti využívající technologie umělé inteligence. Zájem o tyto technologie lze připsat skutečnosti, že klasické způsoby řešení buď neexistují nebo pro svoji robustnost nejsou vhodné. Často jsou využívány v aplikacích „Business Intelligence“ umožňujících získávat potřebné informace pro kvalitní rozhodování a zvyšování konkurenceschopnosti.

Jedním z nejrozšířenějších nástrojů umělé inteligence jsou vícevrstvé neuronové sítě. Jejich velkou

výhodou je relativní jednoduchost a možnost samoučení na základě souboru vzorových situací.

Pro etapu učení se nejčastěji používá algoritmus zpětného šíření chyby. Avšak při jeho realizaci a počátečním využívání zjistíme, že je nutné jej doplnit vhodným způsobem řízení velikosti koeficientu učení, na jehož volbě je závislá konvergence procesu učení.

Cílem tohoto příspěvku je kompaktní vyjádření algoritmu učení v maticovém tvaru, odvození metody řízení koeficientu učení n-úrovňové neuronové sítě

a prezentace dosažených výsledků. Pro řízení procesu učení jsou odvozeny dvě funkce: jedna pro řízení procesu učení při relativně velkých hodnotách chybové funkce a druhá pro řízení procesu učení při hodnotách chybové funkce blízkých se globálnímu minimu.

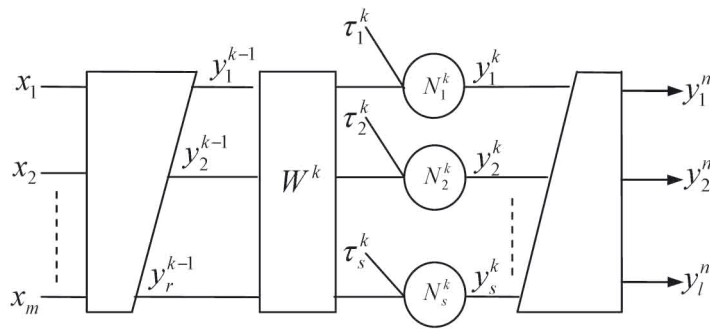
MATERIÁL A METODY

Víceúrovňová neuronová síť je schematicky znázorněná na obrázku 1. Počet neuronů vstupní vrstvy N_{in} je dán počtem vstupů a výstupní vrstvy počtem výstupů neuronové sítě. Počty neuronů N_k ve skrytých

vrstvách nelze stanovit přesně, pro praktické aplikace však stačí většinou dodržet podmínku

$$N_k \geq \min(N_{out}, N_{in}), \quad (1)$$

kde N_{out} – počet neuronů výstupní vrstvy. Taktéž neexistuje ani exaktní pravidlo pro stanovení počtu úrovní neuronové sítě. Z praxe je však známo, že pro většinu praktických úloh jsou postačující tři úrovně: *vstupní*, *jedna skrytá* a *výstupní*, přičemž vstupní úroveň pouze opakuje vstupní signály ($y_i^1 = x_i$). V případně potřeby ji lze využít pro transformaci vstupních hodnot.



1: n -úrovňová neuronová síť

Hodnoty výstupů neuronů jsou napojeny na vstupy neuronů následující úrovně váhami, které jsou specifikovány maticí ve tvaru

$$W^k = \begin{bmatrix} w_{11}^k & w_{12}^k & \dots & w_{1m}^k \\ w_{21}^k & w_{22}^k & \dots & w_{2m}^k \\ \dots & \dots & \dots & \dots \\ w_{r1}^k & w_{r2}^k & \dots & w_{rm}^k \end{bmatrix}. \quad (2)$$

Počet řádků r matice W^k je dán počtem neuronů vrstvy $(k-1)$ a počet sloupců – počtem neuronů vrstvy k . Obecně neurony N_i^k vrstvy provádí transformaci

$$y_s^k = f_s^k(\sum_r y_r^{k-1} \cdot w_{rs}^k + \tau_s^k). \quad (3)$$

Hodnota výstupu $y_s^1 = x_s$, případně $y_s^1 = f_s^1(x_s)$, provádí-li vstupní úroveň sítě nějakou transformaci vstupů. V obecném případě může každý neuron sítě realizovat vlastní funkci, i když nejčastěji všechny neurony (kromě vstupní vrstvy) realizují stejnou funkci. Hodnoty τ_s^k jsou prahové hodnoty neuronů.

Vlastní proces korekce parametrů neuronové sítě vychází z algoritmu zpětného šíření chyb (BPE –

Back Propagation Errors), pro který jsou v další části odvozeny jednoduché maticové vztahy.

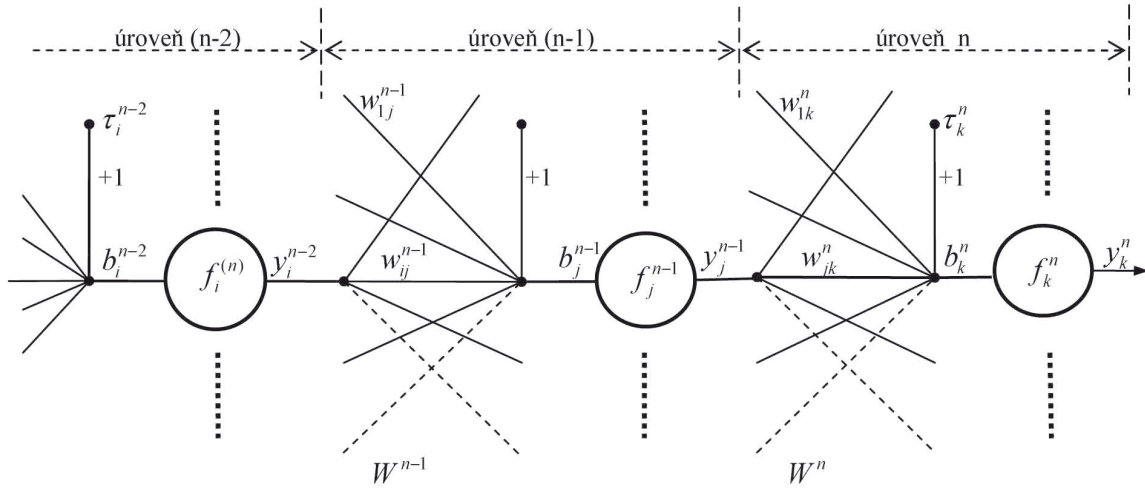
Problém konvergence algoritmu je řešen kontinuálním řízením velikosti koeficientu učení funkcemi: $\lambda = \lambda_1(Ch, G)$ – pro počáteční fázi učení, $\lambda = \lambda_2(Ch, G)$ – pro závěrečnou fázi učení.

UČENÍ NEURONOVÉ SÍTĚ

Znalosti naučené neuronové sítě jsou obsaženy ve všech parametrech sítě, tj. váhách w_{ij} a prahových hodnotách τ_j neuronů všech úrovní sítě. Jejich počáteční hodnoty jsou generovány zpravidla náhodně. V procesu učení se parametry neuronové sítě upravují tak, aby se minimalizovala výpočetní chyba neuronové sítě hodnocená chybovou funkcí

$$Ch = \sum_v E(X_v) = \sum_v \frac{1}{2} \sum_k [y_k^n(X_v) - y_k^{on}(X_v)]^2, \quad (4)$$

kde X_v je v -tý vstupní vektor vzoru učícího souboru. Tj. provádí se minimalizace součtu čtverců rozdílů požadovaných – y_k^{on} a skutečných – y_k^n hodnot výstupů, za všechny výstupy sítě a všechny vzory učícího souboru.



2: Vrstvy n-úrovňové neuronové sítě

Aby změna parametrů sítě minimalizovala skalární chybovou funkci Ch , je nutno ji provádět v opačném směru gradientu, protože gradient udává směr nejrychlejšího růstu, a tedy nové hodnoty parametrů w_{pq}^u , τ_{*q}^u úrovně u ($n \geq u > 1$) se stanoví podle vztahů:

$$w_{pq}^u = w_{pq}^u - \lambda \cdot \frac{\partial Ch}{\partial w_{pq}^u}, \quad (5)$$

$$\tau_{*q}^u = \tau_q^u - \lambda \cdot \frac{\partial Ch}{\partial \tau_q^u}, \quad (6)$$

kde λ – je koeficient učení sítě. V maticovém tvaru lze korekci parametrů neuronové sítě zapsat vztahy

$$W_*^u = W^u - \lambda \cdot \sum_v G_w^u(X_v) \quad (7)$$

a

$$T_*^u = T^u - \lambda \cdot \sum_v G_\tau^u(X_v), \quad (8)$$

s maticemi:

W_*^u , T_*^u – po provedení korekce,
 W^u – vah w_{pq}^u definovaná ve tvaru (2),

T^u – prahových hodnot τ_q^u ,
 G_w^u – gradientů s prvky $g_{w_{pq}}^u = \partial E(X_v) / \partial w_{pq}^u$,
 G_τ^u – gradientů s prvky $g_{\tau_q}^u = \partial E(X_v) / \partial \tau_q^u$,
a koeficientem učení neuronové sítě λ .

Pro jeden vzor X_v a parametr w_{jk}^n (výstupní úrovně) bude gradient funkce E roven parciální derivaci

$$\frac{\partial E}{\partial w_{jk}^n} = \sum_k (y_k^n - y_k^{on}) \frac{\partial y_k^n}{\partial w_{jk}^n} = d_k^n \cdot f_k'^n \cdot (b_k^n) \frac{\partial b_k^n}{\partial w_{jk}^n} = d_k^n \cdot f_k'^n \cdot (b_k^n) \cdot y_j^{n-1}, \quad (9)$$

kde

$$\frac{\partial b_k^n}{\partial w_{jk}^n} = \frac{\partial}{\partial w_{jk}^n} [\sum_s y_s^{n-1} \cdot w_{sk}^n + \tau_k^n] = y_j^{n-1} \quad \text{pro } s=j \quad \text{a} \quad \frac{\partial b_k^n}{\partial w_{jk}^n} = 0 \quad \text{pro } s \neq j.$$

Podobně lze odvodit i parciální derivaci $\partial E / \partial \tau_k^n$ avšak s tím rozdílem, že

$$\frac{\partial E}{\partial \tau_k^n} = d_k^n \cdot \frac{\partial y_k^n}{\partial \tau_k^n} = d_k^n \cdot f_k^n(b_k^n). \quad (10)$$

$$\frac{\partial b_k^n}{\partial \tau_k^n} = \frac{\partial}{\partial \tau_k^n} [\sum_s y_s^{n-1} \cdot w_{sk}^n + \tau_k^n] = 1,$$

a tak

Výraz (9) reprezentuje hodnotu prvku g_{jk}^n matice G_w^n (stejných rozměrů jako matice vah n -té vrstvy). Všechny hodnoty matice G_w^n , pro daný vzor učícího souboru, lze vypočítat součinem sloupcové matice

Y^{n-1} – výstupů neuronů úrovně $n-1$ a řádkové matice s prvky $d_k^n \cdot f_k^n$. Pro získání matice s prvky $d_k^n \cdot f_k^n$ je definována zvláštní maticová operace $\otimes$.

Definice. Necht' A s prvky a_{ij} , B s prvky b_{ij} jsou matice stejného typu (m,n) , pak $C = A \otimes B$ je matice s prvky $c_{ij} = a_{ij} \cdot b_{ij}$.

Využitím této operace lze parciální derivace chybové funkce E podle parametrů w a τ zapsat výrazy

$$G_w^n = Y^{n-1} \cdot [D^n \otimes F'^n]^T, \quad (11)$$

$$G_\tau^n = D^n \otimes F'^n, \quad (12)$$

kde:

$$D^n = Y^n - Y^{on},$$

Y^{on} – sloupcová matice požadovaných výstupů neuronové sítě,

Y^n – sloupcová matice výstupů neuronové sítě (obecně výstupů neuronů n -té úrovně),

F'^n – sloupcová matice hodnot derivací funkcí $f'_k(b_k^n)$ neuronů n -té vrstvy.

Podobně lze odvodit parciální derivace a gradienty chybové funkce E podle parametrů sítě v úrovni $n-1$.

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^{n-1}} &= \sum_k (y_k^n - y_k^{on}) \frac{\partial y_k^n}{\partial w_{ij}^{n-1}} = -\sum_k d_k^n \cdot f'_k(b_k^n) \frac{\partial b_k^n}{\partial w_{ij}^{n-1}} = \\ &= \sum_k d_k^n \cdot f'_k(b_k^n) \cdot \frac{\partial}{\partial w_{ij}^{n-1}} [\sum_s y_s^{n-1} \cdot w_{sk}^n + \tau_k^n] = \\ &= \sum_k d_k^n \cdot f'_k(b_k^n) \cdot w_{jk}^n \cdot \frac{\partial}{\partial w_{ij}^{n-1}} y_j^{n-1} = \\ &= \sum_k d_k^n \cdot f'_k(b_k^n) \cdot w_{jk}^n \cdot f'_j(b_j^{n-1}) \cdot \frac{\partial}{\partial w_{ij}^{n-1}} b_j^{n-1} = \\ &= \sum_k d_k^n \cdot f'_k(b_k^n) \cdot w_{jk}^n \cdot f'_j(b_j^{n-1}) \cdot \frac{\partial}{\partial w_{ij}^{n-1}} [\sum_s y_s^{n-2} \cdot w_{sj}^{n-1} + \tau_j^{n-1}] = \\ &= \sum_k d_k^n \cdot f'_k(b_k^n) \cdot w_{jk}^n \cdot f'_j(b_j^{n-1}) \cdot y_i^{n-2} = d_j^{n-1} \cdot f'_j(b_j^{n-1}) \cdot y_i^{n-2}, \end{aligned}$$

tj.

$$\frac{\partial E}{\partial w_{ij}^{n-1}} = d_j^{n-1} \cdot f'_j(b_j^{n-1}) \cdot y_i^{n-2} \quad (13)$$

Výraz $d_j^{n-1} = \sum_k d_k^n \cdot f'_k(b_k^n) \cdot w_{jk}^n$ reprezentuje j -tý prvek sloupcové matice obsahující přetransformované chyby z výstupů neuronové sítě k výstupu j -tého neuronu úrovně $(n-1)$. Tuto transformaci chyb z n -té úrovně (výstupů neuronové sítě) k výstupům úrovně $(n-1)$ lze vyjádřit maticovým výrazem

$$D^{n-1} = W^n \cdot (D^n \otimes F'^n). \quad (14)$$

Při akceptování zavedených označení, výraz $d_j^{n-1} \cdot f'_j(b_j^{n-1}) \cdot y_i^{n-2}$ představuje prvek maticového součinu $Y^{n-2} \cdot [D^{n-1} \otimes F'^{n-1}]^T$, takže parciální derivace funkce E podle vah úrovně $(n-1)$ budou

$$G_w^{n-1} = Y^{n-2} \cdot [D^{n-1} \otimes F'^{n-1}]^T \quad (15)$$

Derivace funkce E podle parametru τ_j^{n-1} je

$$\begin{aligned} \frac{\partial E}{\partial \tau_j^{n-1}} &= \sum_k (y_k^n - y_k^{on}) \frac{\partial y_k^n}{\partial \tau_j^{n-1}} = -\sum_k d_k^n \cdot f'_k(b_k^n) \frac{\partial b_k^n}{\partial \tau_j^{n-1}} = \\ &= \sum_k d_k^n \cdot f'_k(b_k^n) \cdot \frac{\partial}{\partial \tau_j^{n-1}} [\sum_s y_s^{n-1} \cdot w_{sk}^n + \tau_k^n] = \\ &= \sum_k d_k^n \cdot f'_k(b_k^n) \cdot w_{jk}^n \cdot \frac{\partial}{\partial \tau_j^{n-1}} y_j^{n-1} = \\ &= \sum_k d_k^n \cdot f'_k(b_k^n) \cdot w_{jk}^n \cdot f'_j(b_j^{n-1}) \cdot \frac{\partial}{\partial w_{ij}^{n-1}} [\sum_s y_s^{n-2} \cdot w_{sj}^{n-1} + \tau_j^{n-1}] = \\ &= \sum_k d_k^n \cdot f'_k(b_k^n) \cdot w_{jk}^n \cdot f'_j(b_j^{n-1}) = d_j^{n-1} \cdot f'_j(b_j^{n-1}). \end{aligned} \quad (16)$$

Očividně parciální derivace funkce E podle všech τ_j^{n-1} lze zapsat pomocí maticového výrazu

$$G_T^{n-1} = D^{n-1} \otimes F'^{n-1}. \quad (17)$$

Vztahy (11) a (12), (15) a (17) představují stejné operace, ale v jiné vrstvě neuronové sítě, a tedy obecně lze pro korekci parametrů w a τ k -té vrstvy n -úrovňové neuronové sítě napsat obecné maticové vztahy

$$W_*^k = W^k - \lambda \cdot \sum_v Y_v^{k-1} \cdot [D_v^k \otimes F'_v{}^k]^T, \quad (18)$$

$$T_*^k = T^k - \lambda \cdot \sum_v [D_v^k \otimes F'_v{}^k], \quad (19)$$

kde

$$D_v^k = W_*^{k+1} \cdot (D_v^k \otimes F'_v{}^k). \quad (20)$$

Pro poslední úroveň sítě $D_v^n = Y_v^n - Y_v^{on}$ a za předpokladu, že vstupní úroveň neuronové sítě pouze opakuje vstupní hodnoty, pak pro korekci vah druhé úrovně $Y_v^{k-1} = Y_v^1 = X_v$.

Vztahy (18), (19) a (20) jsou základní maticové vztahy algoritmu zpětného šíření chyb. Vlastní zpětné šíření chyb, resp. transformace chyb k výstupům skrytých vrstev je reprezentováno vztahem (20).

Výpočetní režim k -té vrstvy neuronové sítě, tj. prvky matice Y^k jsou definovány vztahem (3), kde $\sum_r y_r^{k-1} \cdot w_{rs}^k + \tau_s^k = b_s^k$ je prvek sloupcové matice $B^k = W^k \cdot [Y^{k-1}]^T - T^k$ excitačních hodnot neuronů k -té vrstvy.

Pro praktické použití algoritmu BPE zbývá vyřešit ještě jeden problém, a to stanovení velikosti koeficientu učení λ .

ŘÍZENÍ KOEFICIENTU UČENÍ

Parametry neuronové sítě w a τ jsou obvykle gene-

rovány zcela náhodně z intervalu $(0,1)$ nebo $(-1,1)$. Při nevhodném koeficientu učení a nepříznivých počátečních hodnotách parametrů w a τ nemusí hodnota chybové funkce konvergovat k hodnotě globálního minima. Proces učení se může zacyklit, zastavit nalezáním lokálního extrému chybové funkce nebo může probíhat velmi pomalu.

S ohledem na funkci koeficientu učení (viz vztahy (5) a (6)) lze říci, že jeho hodnota bude menší než 1. Pro řešení řady konkrétních úloh může vyhovovat konstantní hodnota stanovená experimentálně pro určitý učicí soubor hodnot. Nicméně při opakovaném učení (přeučení) sítě a jiných počátečních hodnotách w a τ nemusí dříve použitý koeficient učení vyhovovat.

Změna parametrů neuronové sítě prováděná na základě vztahů (5) a (6) bude nulová, je-li dosažen lokální extrém chybové funkce (derivace podle všech parametrů jsou rovny nule), a tedy bez ohledu na hodnotu λ i hodnotu chybové funkce Ch se proces učení zastaví. Bylo by však žádoucí lokální extrém eliminovat a to již při přibližování parametrů k hodnotám, kdy se proces učení zpomaluje. Pro jeho zrychlení, resp. rychlé opuštění je nezbytné zvýšit hodnotu koeficientu učení.

Nutnost eliminace lokálních extrémů v procesu učení vede k potřebě řízení hodnoty koeficientu učení λ a tedy stanovení vhodné závislosti hodnoty koeficientu učení na stavu optimalizace parametrů neuronové sítě.

Konvergenční proces učení ke globálnímu nebo lokálnímu extrémům chybové funkce lze sledovat a řídit prostřednictvím hodnoty chybové funkce Ch a absolutní hodnoty gradientu této funkce $|grad Ch|$. Nechť Ch_{min} je hodnota globálního minima chybové funkce, pak lze v rámci procesu učení charakterizovat následující základní situace:

1. $|Ch - Ch_{min}| \gg 0$ a $|grad Ch| \gg 0$ – proces je daleko od lokálního i globálního extrému.
2. $|Ch - Ch_{min}| \gg 0$ a $|grad Ch| \rightarrow 0$ – proces je blízko lokálního extrému.
3. $|Ch - Ch_{min}| \rightarrow 0$ a $|grad Ch| > 0$ – proces se blíží ke globálnímu extrému.
4. $|Ch - Ch_{min}| \rightarrow 0$ a $|grad Ch| \rightarrow 0$ – dosažen globální extrém, pokud hodnota chybové funkce je menší nebo rovna požadované hodnotě.

Změnu chybové funkce při malých změnách parametrů w a τ neuronové sítě lze vyjádřit jejím totálním diferenciálem jako

$$dCh = \sum_{i,j,k} \frac{\partial Ch}{\partial w_{ij}^k} dw_{ij}^k + \sum_{l,k} \frac{\partial Ch}{\partial \tau_l^k} d\tau_l^k. \quad (21)$$

Vzhledem k tomu, že

$$dw_{ij}^k = \lambda \frac{\partial Ch}{\partial w_{ij}^k}, \quad d\tau_l^k = \lambda \frac{\partial Ch}{\partial \tau_l^k}, \quad \text{pak}$$

$$dCh = \lambda \left[\sum_{i,j,k} \left(\frac{\partial Ch}{\partial w_{ij}^k} \right)^2 + \sum_{l,k} \left(\frac{\partial Ch}{\partial \tau_l^k} \right)^2 \right] = \lambda \cdot |grad Ch|^2. \quad (22)$$

Odtud

$$\lambda = \lambda_1 = \frac{dCh}{|grad Ch|^2} = \frac{dCh}{Ch} \cdot \frac{Ch}{|grad Ch|^2} = \varepsilon_1 \cdot \frac{Ch}{|grad Ch|^2}. \quad (23)$$

Koeficient ε_1 ve vztahu (23) představuje relativní změnu chybové funkce pro aktuální epochu. Např. volbou hodnoty $\varepsilon_1 = 0,05$ je požadována změna (snížení) chybové funkce o 5 %.

Akceptování hodnoty koeficientu učení λ_1 podle vztahu (23) umožňuje eliminovat zpomalování procesu učení, případně jeho zastavení, při přiblížení parametrů hodnotám lokálního extrému. Při velké chybě a malé absolutní hodnotě gradientu chybové funkce je λ_1 dostatečně velké pro zabezpečení úniku z oblasti lokálního extrému směrem ke globálnímu extrému.

Vzhledem k členitosti povrchu chybové funkce mohou některé úlohy vyžadovat zadání menší hodnoty relativní změny. Při požadavku větší relativní změny chybové funkce (tj. rychlejší konvergence) může však dojít k nevhodně velké změně parametrů sítě a místo konvergence chybové funkce k minimální hodnotě se proces změny na divergentní, případně se zacyklí.

Při řešení regresní úlohy, kdy při stejných vektorech vstupních hodnot mohou být požadovány různé výstupní hodnoty sítě, nebude limitní hodnota chybové funkce (resp. globálního extrému) nulová. V těchto případech i požadavek snížení hodnoty chybové funkce o 5 % může být příliš vysoký.

Plynulejší průběh procesu učení (jak bude patrné z řešených příkladů) lze docílit úpravou funkce řízení koeficientu učení požadavkem konstantní hodnoty poměru

$$\frac{dCh}{Ch \cdot |grad Ch|} = \frac{\varepsilon_1}{|grad Ch|} = konst = \varepsilon_2. \quad (24)$$

Pak nová funkce řízení koeficientu učení bude

$$\lambda_2 = \varepsilon_2 \cdot \frac{dCh}{|grad Ch|}. \quad (25)$$

Nevýhodou využití tohoto způsobu řízení koeficientu učení neuronové sítě bude očividně pomalejší konvergence v závěrečné fázi, tj. při malých absolutních hodnotách gradientu chybové funkce.

DOSAŽENÉ VÝSLEDKY

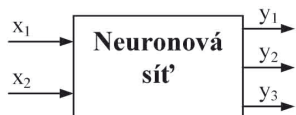
Na základě odvozených vztahů byl realizován experimentální model n-vrstvé neuronové sítě v programovacím jazyce LPA Win-Prolog doplněný o potřebné maticové operace. Z hlediska časového řešení (hlavně etapy učení) to není nejvhodnější volba, nicméně pro ověření teoretických poznatků naprosto vyhovující. Realizace potvrdila správnost i jednoduchost aplikace odvozených vztahů pro učení n-vrstvé neuronové sítě včetně funkcí pro řízení hodnoty koeficientu učení.

V následujících experimentech je neuronová síť použita pro modelování funkcí a procesu rozhodování o poskytnutí úvěru. V obou případech neurony realizují funkci

$$y = \frac{1}{1 + \exp(-x)}. \quad (26)$$

Průběh změny chybové funkce Ch a koeficientu učení λ_1 v jednotlivých etapách učení neuronové sítě modelující logické funkce *součtu*, *součinu* a *implikace* (viz obr. 3) je uveden na obr. 4.

3: Model pro simulaci logických funkcí součtu, součinu a implikace

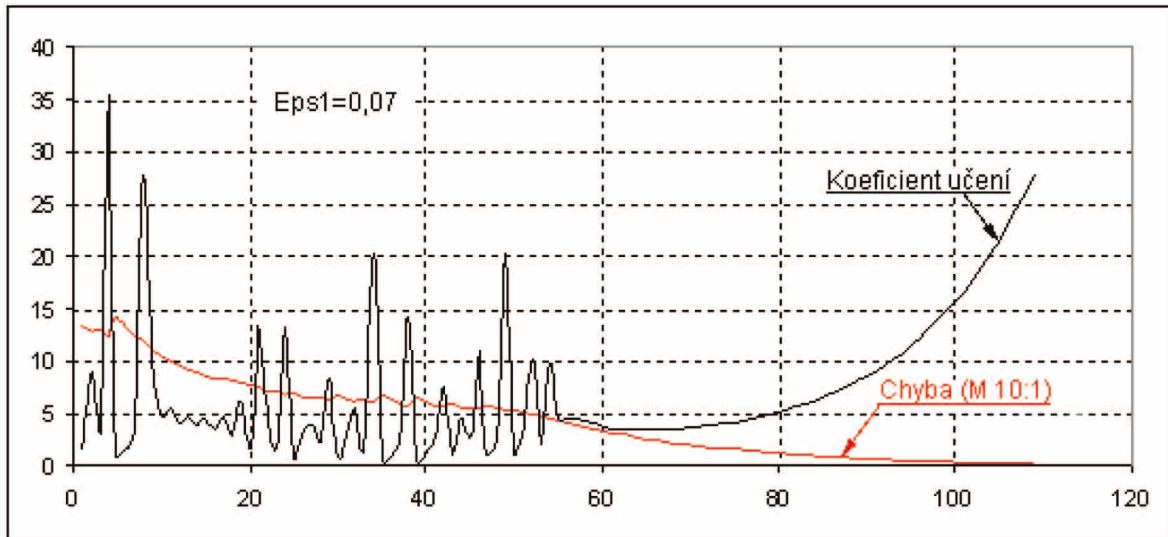


x_1	x_2	$y_1 = x_1 \vee x_2$	$y_2 = x_1 \wedge x_2$	$y_3 = x_1 \Rightarrow x_2$
0	0	0	0	1
0	1	1	0	0
1	0	1	0	0
1	1	1	1	1

Je-li $|\text{grad } Ch| < 1$, jsou změny koeficientu učení razantnější, což umožňuje rychlé opuštění okolí lokálního extrému chybové funkce a rychlejší konvergenci učení v závěrečné fázi, kdy se parametry neuronové sítě přibližují parametrům globálního extrému chybové funkce. Zvyšuje se však nebezpečí prove-

dení příliš velké změny parametrů neuronové sítě a následná možnost divergence procesu učení, případné zacyklení, obzvláště při velké hodnotě chybové funkce (tj. v počátečních etapách učení).

Proces učení pro model na obr. 3 byl ukončen po dosažení hodnoty chybové funkce $Ch \leq 0,015$.

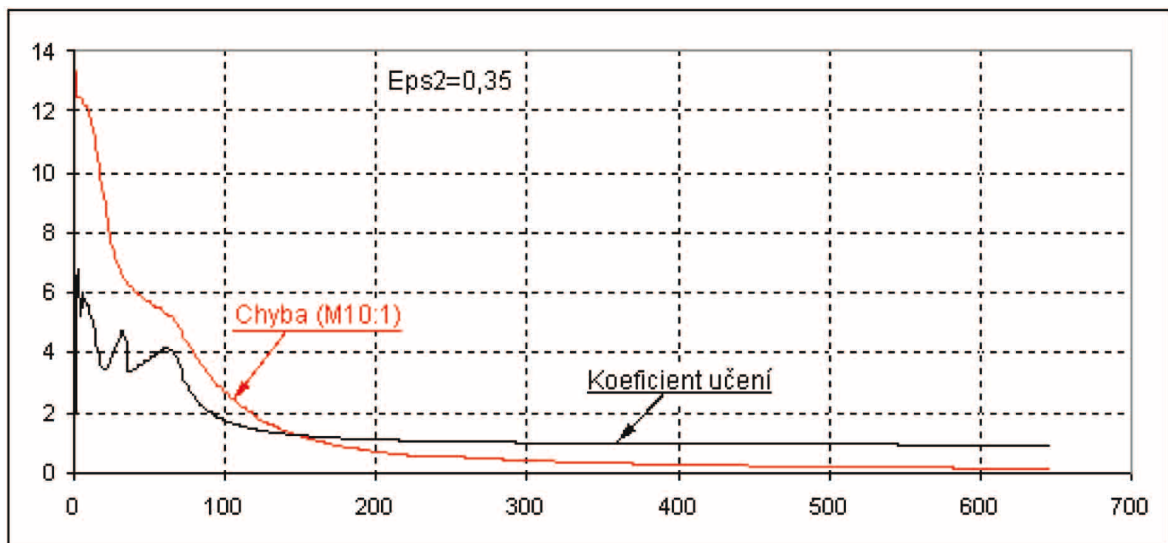


4: Koeficient učení $\lambda = \lambda_1$ (viz 23)

Na obrázku 5 je zobrazen průběh procesu učení, při kterém je řízení koeficientu učení prováděno pouze podle funkce λ_2 (25). Počáteční parametry neuronové sítě jsou stejné jako v předchozím případě.

Z obrázku 5 je patrné, že změna koeficientu učení je plynulejší. Po dosažení chyby menší než $\approx 0,5$ se

učení výrazně zpomalilo. Zatímco v předchozím případě bylo od chyby 0,2 k dosažení chyby 0,015 zapotřebí přibližně ≈ 60 etap, v tomto případě je to již více než ≈ 400 etap učení. Rozdíl v rychlosti učení není v počáteční fázi učení tak výrazný ve prospěch řídicí funkce α_1 .

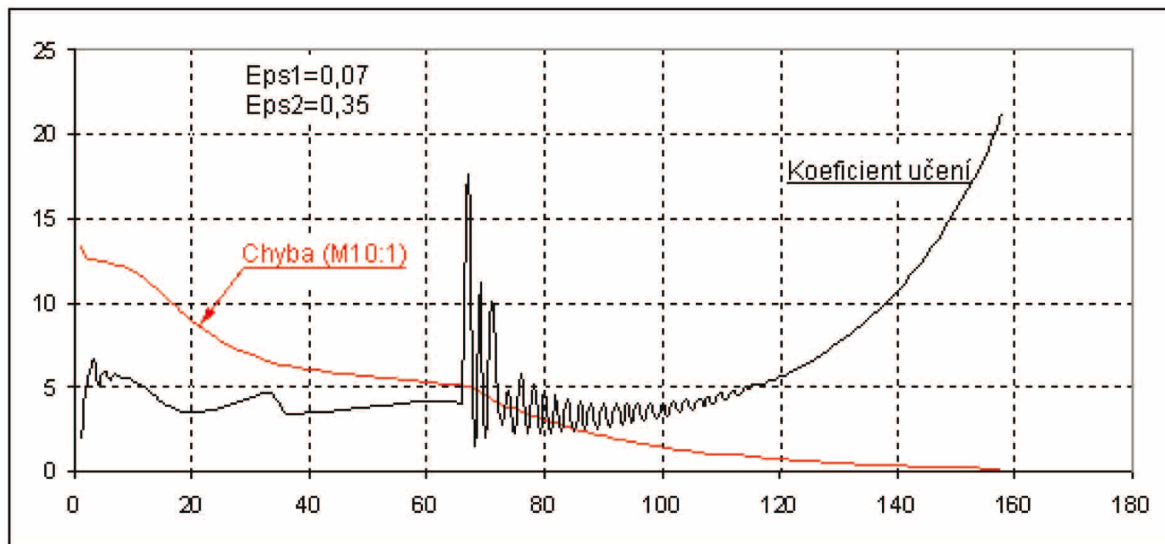


5: Koeficient učení $\lambda = \lambda_2$ (viz 25)

Prováděné experimenty s využitím obou funkcí (λ_1 , λ_2) řízení koeficientu učení vedou k myšlence využití λ_2 v první fázi učení pro větší stabilitu procesu a λ_1 v závěrečné fázi pro zrychlení procesu učení.

Na obr. 6 je znázorněn průběh procesu učení stejné neuronové sítě a při stejných počátečních podmín-

kách jako v obou předchozích případech. V první fázi učení je použito řízení koeficientu učení funkcí λ_2 a v závěrečné fázi funkce λ_1 . Její aktivace je provedena po dosažení hodnoty chybové funkce $<0,5$. Z obrázku je patrné, že pro doučení sítě postačovalo pouze ≈ 80 etap.



6: Koeficient učení $\lambda = \lambda_2$ pro $Ch \leq 0,5$ a $\lambda = \lambda_1$ $Ch > 0,5$

I: Data učícího souboru úlohy pro rozhodování o poskytnutí úvěru

№	Úvěr	Závazky	Rating	Rozhodnutí
1.	1	1	1	1
2.	1	1	2	1
3.	1	1	3	0
4.	1	2	1	1
5.	1	2	2	1
6.	1	2	3	0
7.	1	3	1	1
8.	1	3	2	1
9.	1	3	3	0
10.	2	1	1	1
11.	2	1	2	1
12.	2	1	3	0
13.	2	2	1	1
14.	2	2	2	1
15.	2	2	3	0

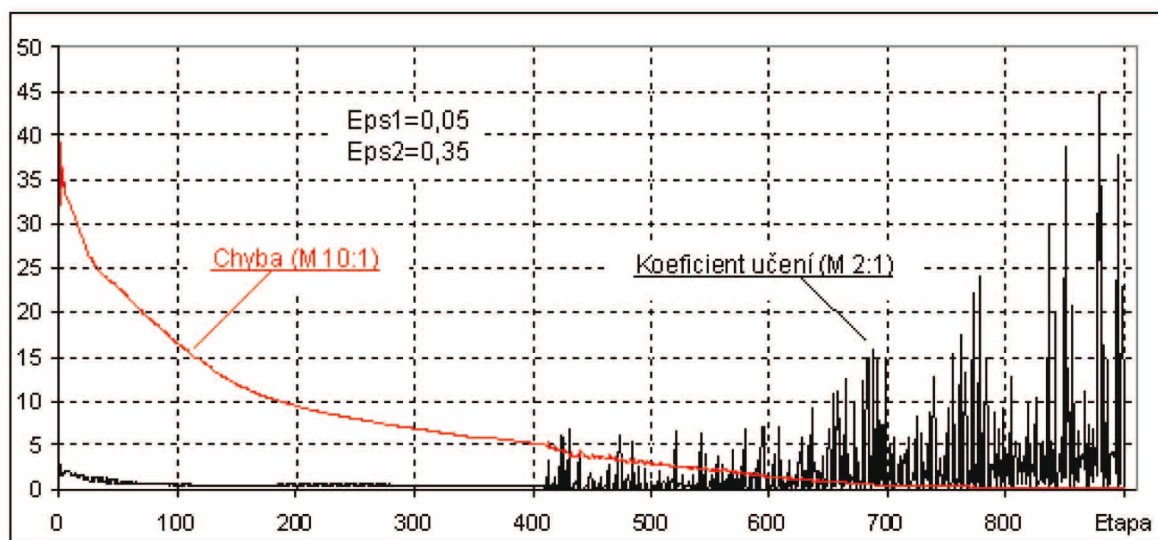
№	Úvěr	Závazky	Rating	Rozhodnutí
16.	2	3	1	1
17.	2	3	2	1
18.	2	3	3	0
19.	3	1	1	1
20.	3	1	2	1
21.	3	1	3	0
22.	3	2	1	1
23.	3	2	2	1
24.	3	2	3	0
25.	3	3	1	0
26.	3	3	2	0
27.	3	3	3	0

Průběh procesu učení modelu neuronové sítě o poskytnutí úvěru na základě tří vstupních charakteristik subjektu žádajícího o úvěr, je uveden na obr. 7. Model neuronové sítě má tři úrovně: tři neurony ve vstupní vrstvě, tři ve skryté vrstvě a jeden neuron ve výstupní. Učícím souborem (viz tab. I) je definováno 27 stavů pro vstupy *Úvěr*, *Závazky* a *Rating* a požadovaný výstup *Rozhodnutí*. Vstupy obsahují hodnoty z intervalu $<1 \div 3>$ a výstupy hodnoty z intervalu $<0 \div 1>$. Vstupní hodnota 1 označuje malý úvěr, závazek a vysoké ratingové hodnocení. Naopak vstupní hodnota 3 znamená vysoký úvěr, velké závazky klienta, ale nízké

ratingové hodnocení. Výstupní hodnota 1 představuje rozhodnutí *poskytnout* úvěr a 0 neposkytnout.

Proces učení byl ukončen při $Ch = 0,01$ po 960 epochách. Průběh změny koeficientu učení svědčí o značné členitosti chybové funkce.

Experimenty ukázaly, že počet etap potřebných k naučení neuronové sítě rozhodovat o poskytnutí úvěru podle zadaných vzorů je v případě použití samotné funkce λ_1 nebo λ_2 větší než při jejich kombinované aplikaci. I přes značné kolísání hodnoty koeficientu učení je průběh změny hodnoty chybové funkce plynulý.



7: Proces učení neuronové sítě modelující rozhodování o poskytnutí úvěru

SOUHRN

Uvedené teoretické poznatky byly aplikovány a ověřeny při realizaci experimentálního modelu n -vrstvé neuronové sítě. Rozšíření základních maticových operací o operaci $\otimes$ (součinu prvků matic) umožňuje nejen velmi kompaktní a jednoduché formální vyjádření algoritmu (zpětného šíření chyb) učení neuronové sítě, ale také jeho jednoduchou realizaci.

Navržený způsob řízení změny koeficientu učení umožňuje efektivní eliminaci lokálních extrémů a tím i plynulejší a rychlejší proces učení neuronové sítě. Například pro řešení úlohy rozpoznávání rastrového obrazu písmen velké abecedy o rozměrech 10×9 bodů byla použita neuronová síť v konfiguraci 90:10:5 (5 bitů výstupní kód znaku). Proces učení byl ukončen při $Ch = 0,5$ po 360 epochách.

Úspěšně byly řešeny i další úlohy z oblasti modelování funkcí, klasifikace do tříd a trendů vývoje ekonomických veličin.

neuronová síť, učení, algoritmus učení, zpětné šíření chyb, maticová forma, koeficient učení, řízení koeficientu učení

LITERATURA

- GUYON, I. P.: Applications of neural network to character recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 1991, 5, 353–382.
- RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J.: Learning representations by back-propagation errors. *Nature (London)*, 1986, 323, 533–536.
- NEGNEVITSKI, M.: *Artificial Intelligence*. Addison-Wesley, 2002.

Adresa

Doc. Ing. Vladimír Konečný, CSc., Ing. Anděla Matiašová, Ph.D., Ing. Ivana Rábová, Ph.D., Ústav informatiky, Mendelova zemědělská a lesnická univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika, e-mail: konecny@mendelu.cz