

## NEURONOVÉ SÍTĚ PRO CHEMII I. TEORIE, SOFTWARE, HARDWARE

VLADIMÍR KVASNIČKA, ŠTĚPÁN SKLENÁK a JIŘÍ POSPÍCHAL

Katedra matematiky, CHTF STU, 812 37 Bratislava

Věnováno prof. RNDr. Ing. Alexandrovi Tkáčovi, DrSc.,  
při příležitosti jeho 70. narozenin

Došlo dne 24.II.1992

## Obsah

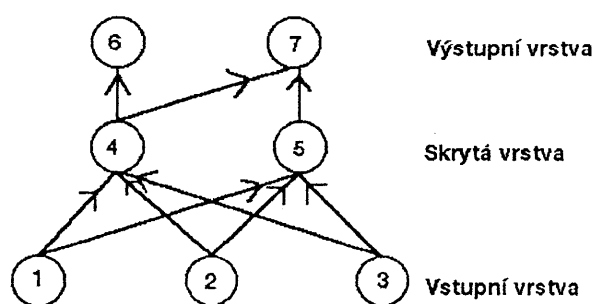
1. Úvod
2. Všeobecný klasifikační problém
3. Definice neuronové sítě
4. Adaptační proces neuronové sítě
5. Design neuronové sítě
6. Programová a jiná implementace neuronových sítí
7. Závěr

## 1. Úvod

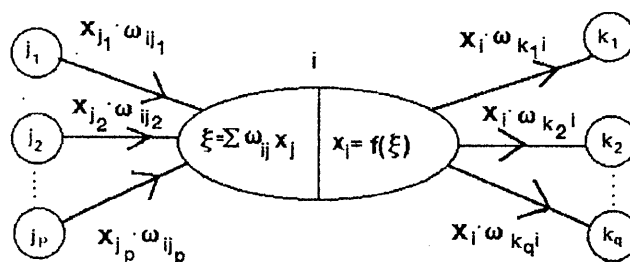
Výzkum neuronových sítí<sup>1</sup> patří mezi hlavní zdroje rozvoje moderní počítačové vědy (informatiky) ve snaze pochopit a algoritmizovat procesy související s fenoménem nazývaným "lidský intelekt" nebo "rozum". Zdá se, že sehraje podobnou úlohu, jakou sehrála před 10-20 lety nerovnovážná termodynamika a synergetika při pochopení evolučních procesů na rozhraní neživých a živých systémů. V této práci se omezíme jen na jeden typ neuronových sítí (tzv. vrstevnaté sítě s adaptací zpětného šíření<sup>1</sup>), nejčastěji využívané pro klasifikaci a předpovědi v chemii.

Zájem o neuronové sítě přináší výsledky, mimo jiné, v návrzích nových typů počítačů a/nebo algoritmů, které jsou schopné řešit rozpoznávací úlohy. Tento přístup je založený na zjednodušených představách o mozku. Skupina neuronů je vzájemně propojena spoji. Neuronové sítě jsou rozděleny do tří nebo více vrstev: vstupní neurony, skryté neurony a výstupní neurony, viz obr. 1. Každý neuron

přijímá signály jednosměrnými (orientovanými) spoji od předcházejících neuronů. Vstupní signál je "vážený" proměnným váhovým koeficientem. Jestliže suma "vážených" vstupů do neuronu přesáhne hodnotu určitého prahového koeficientu, potom neuron začne vysílat signály neuronům z následující vrstvy, viz obr. 2.



Obr. 1. Neuronová síť se třemi vstupními, dvěma skrytými a dvěma výstupními neurony.

Obr. 2. Každý neuron přijímá signály jednosměrnými spoji od předcházejících neuronů. Vstupní signál je "vážený" proměnným váhovým koeficientem  $\omega$ . Jestliže suma "vážených" vstupů do neuronu přesáhne hodnotu určitého prahového koeficientu, potom neuron začne vysílat signály neuronům z následující vrstvy.

Struktura neuronové sítě, t.j. počet neuronů a způsob rozdělení spoji mezi neurony, je navržena tak, aby byla vhodná pro daný studovaný problém. Potom se neuronová síť podrobí tzv. adaptační fázi, ve které jsou váhové a prahové

koeficienty postupně měněny - přizpůsobovány tak, aby odezva neuronové sítě na vstupní signály, brané z tréninkové množiny, byla co nejbližší požadovaným hodnotám. Takto adaptovaná neuronová síť je potom použita k řešení problémů mimo tréninkovou množinu v tzv. aktivní fázi.

Výhody neuronových sítí před konvenčními počítači (algoritmy) jsou následující: 1. obsahují samoučící rysy, 2. schopnost zevšeobecňovat. Samozřejmě, neuronové sítě mají i určité nevýhody, a to: 1. jsou slabé v matematice, 2. někdy mohou poskytnout nesprávný výsledek, 3. neumějí vysvětlit své předpovědi.

## 2. Všeobecný klasifikační problém

Z jistého pohledu si neuronovou síť můžeme představit jako funkci více proměnných, jejíž parametry jsou optimalizované tak, aby poskytovala hodnoty blízké požadovaným. Toto pojetí neuronových sítí úzce souvisí s jejich použitím pro klasifikaci a předpovědi.

Pro natrénování neuronové sítě (tzv. adaptační fázi, při které se optimalizují parametry) je třeba mít sadu vstupů a odpovídající sadu už známých výstupů. Tyto vstupy a výstupy si rozdělíme na dvě části. První část (tzv. tréninkovou množinu) použijeme při trénování neuronové sítě. Druhou část (tzv. testovací množinu) použijeme pro hodnocení, jak úspěšně se nám podařilo neuronovou síť natrénovat. Při trénování se snažíme, aby suma čtverců odchylek výstupů z neuronové sítě od předem zadaných hodnot byla co nejmenší. Tato suma se nazývá účelová funkce. Minimalizace účelové funkce dosahujeme změnou nějakého parametru (parametrů) neuronové sítě.

Takto formulovaná úloha představuje typický problém regresní analýzy, ve kterém se optimalizují parametry zvolené funkce tak, aby její funkční hodnoty byly blízké požadovaným (např. experimentálním) hodnotám neznámé nebo jen obtížně formulovatelné funkce.

Tím se dostáváme k praktické interpretaci neuronových sítí, která úzce souvisí s jejich významem v přírodních vědách. Základní přírodovědecký problém je hledání vztahu mezi strukturou objektů a jejich vlastnostmi. Ideálem každé přírodovědné oblasti (např. chemie) je najít analytické vztahy mezi strukturou objektu jejího výzkumu a jeho vlastnostmi. V chemii tato snaha našla své naplnění v kvantové chemii, kde se použitím aparátu kvantové mechaniky počítají vlastnosti jejich základních objektů - molekul - jen na základě znalostí několika elementárních kon-

stant. Ne vždy je podobný postup realizovatelný pro složitější systém tak, aby se použitím meta-teoretického přístupu (kvantová mechanika je základní součástí meta-teorie chemie) dostaly použitelné výsledky.

Jestliže neznáme "ideální" funkci určující ze zadání (struktury objektu) požadovanou vlastnost, nebo je tato funkce příliš složitá pro výpočet, je možné tento přístup obejít. Můžeme si zvolit (odhadnout) nějakou funkci a měnit pak její parametry tak, abychom dostali požadovaný výsledek. Tuto funkci pak použijeme i pro zadání, kde výsledek neznáme, a můžeme jen doufat, že jsme se "strefili". To je ve velmi zjednodušeném pojetí i případ neuronových sítí.

## 3. Definice neuronové sítě

Neuronová síť je definovaná jako soubor neuronů (jsou označovány  $v$ ) a spojů mezi nimi ( $e=[v,v']$ ). Každý spoj je jednosměrný, říkáme, že vychází z neuronu  $v$  a vchází do neuronu  $v'$ . Žádný neuron nemůže být izolovaný. V tomto článku se budeme zabývat jen takovými neuronovými sítěmi, které neobsahují cykly, t.j. nemůžeme se po šípkách (spojích) vrátit do neuronu, ze kterého jsme vyšli.

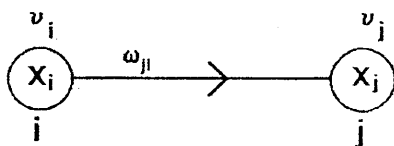
Soubor neuronů může být rozdělen na tři části:

1. vstupní neurony, které jsou sousední jen s vycházejícími spoji,
2. skryté neurony, které jsou sousední aspoň s jedním vycházejícím spojem a alespoň s jedním vcházejícím spojem,
3. výstupní neurony, které jsou sousední jen s vcházejícími spoji.

Neuronová síť z obr. 1 obsahuje 7 neuronů a 9 spojů. Neurony  $v_1, v_2, v_3$  jsou vstupní, neurony  $v_4, v_5$  jsou skryté a neurony  $v_6, v_7$  jsou výstupní.

Soubor skrytých neuronů může být dále rozdělen na vrstvy. Jednotlivé vrstvy obsahují neurony, které mají stejnou maximální vzdálenost od vstupních neuronů, t.j. postupujeme-li od vstupních neuronů ve směru šipek po spojích mezi jednotlivými neurony, do neuronu  $n$ -té vrstvy se dostaneme v nejhorším případě po průchodu  $n$  spoji.

Neurony a spoje sítě jsou ohodnoceny reálnými čísly. Každému skrytému nebo výstupnímu neuronu  $v_i$  je přiřazen prahový faktor  $v_i$ , každému spoji  $e = [v_i, v_j]$  je přiřazen váhový faktor  $\omega_{ji}$  a každému neuronu  $v_i$  je přiřazena aktivita  $x_i$ , viz obr. 3.



Obr. 3. Neurony a spoje sítě jsou ohodnoceny reálnými čísly. Každému skrytému nebo výstupnímu neuronu  $v_i$  je přiřazen prahový faktor  $v_i$ , každému spoji  $e=[v_i, v_j]$  je přiřazen váhový faktor  $\omega_{ji}$  a každému neuronu  $v_i$  je přiřazena aktivita  $x_i$ .

Postulujeme, že aktivity vstupních neuronů jsou konstantní, zatímco aktivity ostatních neuronů jsou určeny vztahem

$$x_i = f(\xi_i), \quad (1a)$$

$$\xi_i = \sum_j \omega_{ij} x_j + v_i, \quad (1b)$$

kde sumace je uskutečněna pro všechny neurony  $v_j$  se spojením vstupujícím do daného neuronu  $v_i$ . Přechodová funkce  $f(\xi)$  je kladná a monotónně rostoucí a vyhovuje asymptotickým podmínkám  $f(\xi) \rightarrow 1$  pro  $x \rightarrow \infty$  a  $f(\xi) \rightarrow 0$  pro  $x \rightarrow -\infty$ . Tyto podmínky jsou splněny např. pro funkci

$$f(\xi) = \frac{1}{1 + \exp(-\xi)}, \quad (2)$$

s první derivací určenou  $f'(\xi) = f(\xi)[1 - f(\xi)]$ . Tato funkce zobrazuje množinu reálných čísel  $\mathbb{R}$  na otevřený interval  $(0,1)$ ,  $f: \mathbb{R} \rightarrow (0,1)$ .

Výraz (1) je formální realizací našich úvah o neuronové síti, kde neurony přijímají "vážené" signály - aktivity ( $\omega_{ij}x_j$ ) ze sousedních neuronů a jakmile suma těchto "vážených" aktivit přesáhne určitou prahovou hodnotu ( $v_i$ ), neuron začne vysílat signál do svého okolí, viz obr. 2.

Obraťme naši pozornost na výpočet aktivit  $x_i$  neuronové sítě pro dané váhové a prahové faktory. Jak jsme už řekli, aktivity vstupních neuronů jsou konstantní, představují vstupy do neuronové sítě. Pod vstupy rozumíme proměnné popisující zadání problému. Použitím vztahu (1) určíme aktivity skrytých neuronů z první vrstvy, které bezprostředně sousedí se vstupními neurony, tento rekurentní

postup opakujeme pro následující vrstvy neuronů tak dlouho, až vypočítáme aktivity výstupních neuronů z poslední vrstvy.

V případě, že neuronová síť má jen dvě vrstvy, a to vrstvu vstupních neuronů a vrstvu výstupních neuronů, a nemá žádné skryté neurony, redukuje se na perceptron<sup>2</sup>, který sehrál významnou úlohu při klasifikaci chemických objektů pomocí metod rozpoznávání vzorů (pattern recognition)<sup>3</sup>. Popis chemického problému (molekuly, objektu) pomocí sady proměnných budeme nazývat deskriptorem. Jednoduchými úvahami lze ukázat, že perceptron je schopen klasifikovat jen objekty, jejichž deskriptory jsou lineárně separovatelné, t.j. v prostoru deskriptorů existuje nadrovina, která rozděluje body - objekty - do dvou poloprostorů. Objekty s deskriptory z daného poloprostoru buď danou vlastnost (klasifikaci) mají, nebo ji nemají. Tato skutečnost, že perceptron je schopen správně klasifikovat jen objekty s lineárně separovatelnými deskriptory, vedla Minského a Paperta<sup>2</sup> ke kritice paradigmatu perceptronů, která na určitý čas zpomalila rozvoj neuronového modelování.

Aktivity tvoří stavový vektor  $\mathbf{x} = (x_1, x_2, \dots, x_p)$ , kde  $p$  je celkový počet neuronů. Stavový vektor můžeme formálně rozložit na "podvektory" obsahující aktivity vstupních, skrytých a výstupních neuronů. Neuronovou síť s fixovanými váhovými a prahovými koeficienty můžeme formálně chápat jako zobrazení - funkci  $G$ , která přiřadí vstupnímu stavovému vektoru  $\mathbf{x}_1$  (deskriptoru) výstupní stavový vektor  $\mathbf{x}_0$  (klasifikátor) s komponenty z otevřeného intervalu  $(0,1)$ . Klasifikátor je zároveň vektorem obrazů (funkčních hodnot):

$$\mathbf{x}_0 = G(\mathbf{x}_1; \omega, v) : \quad (3)$$

Skryté aktivity nejsou uvedeny v tomto výrazu, hrají úlohu jen přechodných výsledků, vyskytujících se při rekurentním výpočtu aktivit výstupních neuronů.

Výraz (3) je realizací úvah z druhé kapitoly této práce, kde byl diskutovaný všeobecný klasifikační problém objektů pomocí nějaké odhadnuté funkce, která aproximuje neznámou klasifikační funkci. Přístup neuronových sítí poskytuje možnost navrhnout analyticky funkci  $G(\mathbf{x}_1; \omega, v)$  aproximující neznámou klasifikační funkci. Toto vyjádření vyžaduje určení přechodové funkce  $f(\xi)$  zadané např. vztahem (2).

Pomocí Kolmogorovy věty o zobrazení<sup>4</sup> se ukázalo<sup>5</sup>,

že třívrstvá neuronová síť je schopná realizovat libovolné spojité zobrazení definované na kompaktní množině. Bohužel, tento důkaz je jen existenční, neposkytuje návod, jak tuto síť sestrojít.

V kapitole 2 byl všeobecný klasifikační problém charakterizován jako typický problém regresní analýzy. Z pohledu neuronových sítí se jedná o speciální regresní analýzu, ve které funkce  $G$ , zadaná (3), je určena pomocí přechodových funkcí (2). Jak už bylo poznamenáno, jedná se o silně nelineární problém v důsledku vlastností přechodových funkcí. Jeden z nejrozšířenějších přístupů v regresní analýze je použití polynomiálních funkcí k simulaci funkce  $G$ . Jestliže se pokusíme tento přístup použít i pro neuronové sítě, potom v důsledku silných nelinearit funkce  $G$  musíme použít nejen přímé členy, ale též tzv. smíšené (cross) členy vysokého řádu. Z těchto důvodů počet členů a rovnic je neúměrně vysoký vzhledem k počtu objektů z tréninkové množiny a jejich deskriptorů, tedy určení polynomiálních koeficientů se stává nejednoznačným problémem. Polynomiální model je potom jen schopný nejvýš zreprodukovat klasifikaci objektů z tréninkové množiny, jeho použití i pro jiné objekty (např. z testovací množiny) už vede k fatálním chybám, t.j. zevšeobecnovací schopnost klasifikátorů založených na polynomiálních funkcích je silně ohraničená. Současné numerické zkušenosti<sup>6</sup> s použitím neuronových sítí jako klasifikátorů ukazují, že jejich schopnost zevšeobecnění je poměrně velká a jejich použití k interpretaci objektů z testovací množiny je úspěšné. Zdá se, že tato úspěšnost neuronových sítí jako klasifikátorů je důsledkem jejich struktury jako mnohvrstvého systému a analytického tvaru přechodové funkce (2).

#### 4. Adaptační proces neuronové sítě

Obraťme nyní naši pozornost na problém adaptace funkce  $G(x_1; \omega, v)$ . Nechť pro vstupní stavový vektor  $x_1$  požadujeme výstupní stavový vektor  $\hat{x}_0$ . Naší úlohou je najít takové váhové a prahové faktory, aby odezva neuronové sítě na vstupní vektor  $x_1$  byla co nejbližší požadovanému výstupnímu vektoru  $\hat{x}_0$ . Sestrojíme účelovou funkci

$$E(\omega, v) = 1/2 (x_0 - \hat{x}_0)^2 = 1/2 \sum_k (x_k - \hat{x}_k)^2, \quad (4)$$

kde  $x_k$  a  $\hat{x}_k$  jsou komponenty vektorů  $x_0$  a  $\hat{x}_0$  a index  $k$  jde

přes všechny výstupní neurony. Cílem adaptačního procesu je najít takové váhové  $\bar{\omega}$  a prahové  $\bar{v}$  koeficienty, které minimalizují účelovou funkci  $E$ . K minimalizaci použijeme některou z gradientových metod pro funkce  $n$ -proměnných<sup>7</sup>. Proto je potřebné znát parciální derivace  $\partial E / \partial \omega_{ji}$  a  $\partial E / \partial v_j$ . Uvedeme výsledné vzorce bez důkazů (který není obtížný)<sup>8</sup>. Vztah mezi parciálními derivacemi  $\partial E / \partial v_i$  a  $\partial E / \partial \omega_{ji}$  je následný

$$\frac{\partial E}{\partial \omega_{ji}} = \frac{\partial E}{\partial v_j} x_i, \quad (5a)$$

a platí pro každý neuron  $v_i$ , ze kterého vede spoj do neuronu  $v_j$ . Tento vztah podstatně zjednodušuje výpočet potřebných parciálních derivací, stačí počítat jen parciální derivace  $\partial E / \partial v_i$ , 2-indexové parciální derivace  $\partial E / \partial \omega_{ji}$  jsou jednoduše určené vztahem (5a). Parciální derivace  $\partial E / \partial v_i$  pro výstupní a skryté neurony mají tvar

$$\frac{\partial E}{\partial v_j} = x_j (1 - x_j) (g_j + \sum_l \frac{\partial E}{\partial v_l} \omega_{lj}), \quad (5b)$$

$$g_j = \begin{cases} (x_j - \hat{x}_j) & (\text{pro } j \in V_0), \\ 0 & (\text{pro } j \notin V_0), \end{cases} \quad (5c)$$

kde sumační index  $l$  v (5b) jde přes všechny neurony, z nichž vycházejí spoje do neuronu  $v_j$ . Pomocí vztahů (5a-c) jsme schopni postupně vypočítat parciální derivace účelové funkce (4). V prvním kroku spočítáme pomocí vzorce (5a) parciální derivace podle prahových faktorů pro výstupní neurony. V druhém kroku už můžeme spočítat parciální derivace pro skryté neurony, které bezprostředně sousedí s výstupními neurony. Tento postup opakujeme tak dlouho, až spočítáme všechny potřebné parciální derivace. Protože postup výpočtů derivací účelové funkce postupuje od výstupních neuronů k vstupním neuronům, nazýváme učicí algoritmus strategií zpětného šíření.

Výše uvedený postup výpočtu parciálních derivací je použitelný i pro více než jeden předepsaný pár vstupně-výstupních vektorů  $\hat{x}_0$  a  $x_1$ , t.j. pro  $r$  párů

$$x_1^{(1)} / \hat{x}_0^{(1)}, x_1^{(2)} / \hat{x}_0^{(2)}, \dots, x_1^{(r)} / \hat{x}_0^{(r)} \quad (6)$$

Účelová funkce má potom tvar

$$E = \sum_{i=1}^r E^{(i)} \quad (7a)$$

$$E^{(i)} = 1/2 (x_0^{(i)} - \hat{x}_0^{(i)})^2 \quad (7b)$$

kde  $x_0^{(i)}$  je vektor výstupních aktivit neuronové sítě, které jsou odezvou na vstupní aktivity  $x_1^{(i)}$ , a  $\hat{x}_0^{(i)}$  jsou požadované výstupní aktivity přiřazené vstupnímu vektoru aktivit  $x_1^{(i)}$ . Parciální derivace takto zevšeobecněné účelové funkce  $E$  jsou potom jednoduše vyjádřeny jako suma parciálních derivací  $E^{(i)}$ , jejichž výpočet byl popsán. Poznamenejme, že pro účelovou funkci (7a-b) jednoduchý vztah (5a) mezi parciálními derivacemi  $\partial E / \partial \omega_{ji}$  a  $\partial E / \partial v_j$  již neplatí.

Jestliže známe gradient účelové funkce, můžeme přistoupit k adaptačnímu procesu neuronové sítě, spočívajícímu v minimalizaci účelové funkce vzhledem k váhovým a prahovým parametrům neuronové sítě tak, aby její výstupní aktivity  $x_0^{(1)}, x_0^{(2)}, \dots, x_0^{(r)}$  byly co nejbližší požadovaným výstupním aktivitám  $\hat{x}_0^{(1)}, \hat{x}_0^{(2)}, \dots, \hat{x}_0^{(r)}$  pro vstupní aktivity  $x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(r)}$ . K minimalizaci je vhodné použít libovolnou gradientovou metodu. Např. metoda nejprudšího spádu<sup>9</sup> spočívá v opravě váhových a prahových koeficientů pomocí vztahů,

$$\omega_{ji}^{(k+1)} = \omega_{ji}^{(k)} - \lambda_k \frac{\partial E}{\partial \omega_{ji}}, \quad (8a)$$

$$v_j^{(k+1)} = v_j^{(k)} - \lambda_k \frac{\partial E}{\partial v_j}, \quad (8b)$$

kde  $\lambda_k > 0$  je optimalizované tak, aby byl dosažen maximální pokles účelové funkce. Dobré zkušenosti jsou s metodou sdružených gradientů<sup>7</sup> nebo, pro menší neuronové sítě, s metodou proměnné metriky<sup>7</sup>. Protože účelová funkce  $E$  je silně nelineární, minimalizační proces není standardní úlohou. Obvykle se setkáváme s tím, že dostaneme minimum účelové funkce, které je relativně velké vzhledem k požadované hodnotě. Z těchto důvodů je nutné

realizovat minimalizační proces pro různé množiny náhodně generovaných váhových a prahových faktorů. Potom se vyberou ty optimální hodnoty parametrů, které poskytují nejnižší hodnotu účelové funkce, a ty se použijí pro aktivní proces, ve kterém se adaptované neuronové sítě předkládají vstupní aktivity z testovací množiny. Tato extrapolace (či interpolace) mimo tréninkovou množinu je dalším kritickým bodem aplikací neuronových sítí. Může nastat situace, že objekty z testovací množiny jsou klasifikovány s malou přesností. Potom musíme znovu opakovat adaptační fázi neuronové sítě pro nové množiny náhodně generovaných váhových a prahových koeficientů. Takto získané nové koeficienty jsou znovu testovány klasifikací objektů z testovací množiny. Jestliže stále dostáváme málo přesnou klasifikaci, musíme zaměřit naši pozornost buď na topologii neuronové sítě nebo na tvorbu vstupních aktivit - deskriptorů, které popisují strukturu objektů. Pravděpodobně je topologie neuronové sítě neadekvátní ke studovanému problému, nebo jsou deskriptory špatně navrženy.

## 5. Design neuronové sítě

Před odvozením<sup>1</sup> strategie zpětného šíření se používala neuronová síť typu perceptronu, který je ekvivalentní lineárnímu učicímu stroji. Obsahoval pouze dvě vrstvy neuronů. První vrstva se skládala ze vstupních neuronů (nejméně dva), druhá vrstva z jednoho či více výstupních neuronů. Každý vstupní neuron byl propojen na každý výstupní neuron.

Po odvození strategie zpětného šíření<sup>1</sup> se začala používat mnohvrstevná neuronová síť. V aplikacích neuronových sítí se velmi často používá síť se třemi vrstvami, řidčeji se čtyřmi vrstvami (t.j. dvě skryté vrstvy), a velmi zřídka s ještě více vrstvami.

Pro každou konkrétní aplikaci je nutno navrhnout vhodnou neuronovou síť, což není jednoduchá úloha. Počet vstupních neuronů je roven počtu složek deskriptoru, počet výstupních neuronů je roven počtu složek klasifikátoru. Obtížnější je to s množstvím skrytých neuronů. Většinou používáme pouze jednu skrytou vrstvu neuronů. Množství neuronů v ní pak záleží na tom, jaký deskriptor je použit pro popis daných vzorů, jak velká je tréninková množina, jak složité jsou vztahy mezi deskriptorem a klasifikátorem. Deskriptor má někdy pouze několik složek, jindy tisíce (např. použití neuronových sítí v spektroskopii). Čím více ke vstupním neuronům, tím více je třeba skrytých neuronů.

Konkrétní hodnoty lze zjistit pouze tak, že se zkusmo provedou výpočty s neuronovými sítěmi s různým počtem skrytých neuronů a použije se ta neuronová síť, která dává nejlepší výsledky (t.j. má nejnižší hodnotu účelové funkce pro tréninkovou, ale hlavně pro testovací množinu).

Další faktor, který určuje počet skrytých neuronů, je velikost tréninkové množiny. Když použijeme příliš mnoho skrytých neuronů, vzhledem k velikosti tréninkové množiny, může dojít k "přetrénování" neuronové sítě. Síť má příliš mnoho parametrů k optimalizaci (t.j. váhových a prahových koeficientů). "Přetrénování" neuronové sítě znamená, že váhy mezi neurony a prahy se nastaví tak, že neuronová síť přesně aproximuje data z tréninkové množiny. Nedojde pak ale k velmi žádoucímu "zobecnění", a když pak takto natrénovanou neuronovou síť použijeme pro výpočet klasifikátorů pro jiné vzory, t.j. pro testovací množinu, klasifikace dopadne velmi špatně.

Často lze sestavit závislost chyby klasifikace či velikosti účelové funkce na počtu skrytých neuronů<sup>10,11</sup>. Pro tréninkovou množinu většinou závislost monotónně klesá se vzrůstajícím počtem skrytých neuronů (t.j. neuronová síť se začíná postupně "přetrénovávat"). Pro testovací množinu většinou vypadá závislost jinak. Nejprve funkce klesá, dosáhne minima při určitém počtu skrytých neuronů, a pak opět stoupá, neboť neuronová síť je už "přeučena". Pro danou aplikaci se pak musí vybrat kompromisní hodnota počtu skrytých neuronů, aby chyba pro tréninkovou i testovací množinu byla co nejnižší.

Formálně to připomíná problém polynomické regrese. Jestliže použijeme vysoký stupeň polynomu, body jsou proloženy přesně, ale výsledek nelze použít ani pro interpolaci ani pro extrapolaci. Nedojde k "zobecnění" závislosti.

## 6. Programová a jiná implementace neuronových sítí

Existuje několik možností implementace neuronových sítí, nejčastější je programová. Softwarově se zde modelují paralelní procesy neuronových sítí na konvenčních (t.j. sekvenčních) počítačích. Existuje několik komerčních programů jako Braincel, Autonet, Genesis, NeuralWork Professional II/Plus. Programy jsou psány pro počítače třídy IBM PC 80286 či vyšší, často jsou i další hardwarové či softwarové požadavky<sup>12</sup>.

Program NeuralWork Professional II<sup>12,13</sup> běží na počítačích třídy IBM PC AT, či vyšších, a na počítačích

kompatibilních. Dále též jsou verze pro Macintosh, Sun Microsystems. Program nabízí 25 různých neuronových sítí, 12 transformačních funkcí, 11 sumačních funkcí a 14 učebních pravidel. Zahrnuté jsou adaptační metody Hopfieldova, Perceptron, Back-propagation, ART I, Boltzmannova, Functional Link nets a další. Nevýhodou je pouze cena činící 1795 liber.

Někteří autoři aplikací si napíší vlastní program, jako je program neuronové sítě (se strategií zpětného šíření) s optimalizací účelové funkce pomocí vysoce účinné metody proměnné metriky<sup>8,14</sup>.

Problémem praktických výpočtů je výkon počítače, na kterém se realizuje program neuronové sítě. Lze sice použít počítače tříd IBM PC 8086/8088, 80286, 80386, 80486, ale snad s výjimkou poslední třídy je výkon počítače bohužel nedostatečný. Pro větší aplikace je nutno používat výkonné počítače. Nutný výpočetní výkon jsou desítky MFLOPS. Lze též samozřejmě použít i superpočítače (výkon řádově stovky MFLOPS). Autoři tohoto článku používají počítač firmy Control Data 4680 se současným výkonem 60 MIPS/10 MFLOPS.

Další z možností implementace neuronových sítí<sup>15</sup> je použití paralelních počítačů, např. Connection Machine, obsahující 64 000 procesorů, každý procesor má 64 KB paměti, procesory jsou propojeny do hyperkrychle, cena 2.10<sup>6</sup> US dolarů.

Multiprocesorové systémy<sup>16</sup> sestavené z von Neumannových procesorů budou v budoucnu asi velmi populární. U procesorů zůstane na základě jejich vysokého výpočetního výkonu při omezeném stupni paralelismu velký centrální výkon. Takto postupuje např. firma Fujitsu u svého neuropočítače Prototyp. Systém má matici s 256 procesory. Firma Hema v lednu 1991 představila 13 transputerů tvořících neuronovou síť.

Další z možností implementace<sup>15</sup> neuronových sítí jsou neuropočítačové koprocesory k PC/AT, např. koprocesor ANZA AZ500, schopný modelovat síť do 30 000 neuronů s 480 000 propojeními, s rychlostí 25 000 až 45 000 elementárních kroků za sekundu.

V únoru 1990 firma Intel představila svůj neuročip ETANN-IC. Japonská firma Hitachi nabízí neuronový počítač typu notebook s 1152 neurony. Světový rekord v simulaci<sup>16</sup> neuronové sítě drží středisko Jaderného výzkumu v SRN, kde na superpočítači CRAY nasimulovali neuronovou síť s 10<sup>5</sup> neurony a 10<sup>10</sup> spoji mezi neurony. Budoucnost neuropočítačů závisí zejména na přímé hardwarové implementaci neuronové sítě.

Objevily se pokusy realizovat neuronové sítě optickou

cestou. Jako zajímavost lze uvést práci<sup>17</sup>, kde autoři zkoumali formování vodivé cesty při elektrochemické polymeraci pyrrolu v agarovém gelu pomocí pulzů napětí. Uvedený model může sloužit jako jednoduchý model neuronové sítě v živých systémech.

## 7. Závěr

Klasifikační a predikční schopnosti neuronových sítí umožňují implementaci poměrně jednoduchých systémů, které se podobají expertním systémům. Prostředky logického programování jsou nahrazeny zobrazením - funkcí (3), která přiřadí deskriptorům molekuly příslušné klasifikátory (vlastnosti molekul). Pro již jednou adaptovanou neuronovou síť numerický výpočet tohoto zobrazení je rychlou rutinní záležitostí. Bohužel adaptace neuronové sítě, spočívající v minimalizaci účelové funkce (4), je poměrně složitým problémem.

Přes všechny těžkosti se neuronové sítě používají a zkoušejí pro nové úkoly. Důvodem je často jejich relativní úspěšnost tam, kde běžné metody poskytují nedostatečné výsledky, nebo úplně selhávají. Kde je ale znám správný model, neuronové sítě nic lepšího neposkytnou. Možnost natrénovat neuronovou síť a potom z ní vyvodit dosud neznámé fyzikální či chemické vztahy je sice lákavá, ale bohužel málo nadějná. Neuronové sítě nejsou vhodné pro řešení otázek typu "proč?", poskytují jenom odpovědi bez bližšího vysvětlení.

## LITERATURA

1. Rumelhart D. E., McClelland J. L.: *Parallel Distributed Processes*, vols. 1-2 (MIT Press, Cambridge, MA, 1986.)
2. Minsky M. L., Papert S. A.: *Perceptrons* (MIT Press, Cambridge, MA, 1969.)

3. Varmuza K.: *Pattern Recognition in Chemistry*. Springer Verlag, Berlin 1980.
4. Kolmogorov A. M.: Dokl. Akad. Nauk. SSSR 114, 953 (1957).
5. Hecht-Nielsen R., v: IEEE First International Conference on Neural Networks, San Diego, June 1987.
6. Zupan J., Gasteiger J.: Anal. Chim. Acta 248, 1 (1991).
7. Lukšan L.: *Metody s proměnnou metrikou*. Academia, Praha 1990.
8. Kvasnička V.: J. Math. Chem. 6, 63 (1991).
9. Brunovská A.: *Malá optimalizácia. Metody, programy, príklady*. ALFA, Bratislava 1990.
10. Qian N., Sejnowsky T. J.: J. Mol. Biol. 202, 865 (1988).
11. Kvasnička V., Sklenák Š.: Collect. Czech. Chem. Comm., (zasláno).
12. Personal Computer World 366, (January 1992).
13. Suchanek E. G.: Tetrahedron Comput. Methodol. 3, 120 (1990).
14. Kvasnička V.: Chem. Papers 44, 775 (1990).
15. Pelikán E.: *Neuronové sítě* (výzk. zpráva), Praha, listopad 1990.
16. Jeanmaire P.: Computer World 2, No. 22 (7.6.1991).
17. Kawai T., Motobayashi H., Kuwabara T., Yoshino K.: Jpn. J. Appl. Phys., Part 2 30, L622 (1991).

V. Kvasnička, Š. Sklenák and J. Pospíchal  
(Department of Mathematics, Slovak Technical University,  
Bratislava): **Neural Networks in Chemistry I. Theory,  
Software and Hardware**

Basic principles of neural networks and their adaptation processes are presented. Hardware and software implementations of neural networks are briefly outlined.