

Simulátor neuronové sítě ve výuce projektantů znalostních systémů

J. Teda^a, J. Schindler^b

^a VÍTKOVICE CAD/CAM Systémy spol. s r.o., Ruská 60, 706 02 Ostrava 6, Česká republika

^b Ekonomická fakulta VŠB-TU Ostrava, Sokolská 33, 701 21 Ostrava, Česká republika

Abstrakt

V současné době roste význam praktického využití metod umělé inteligence při realizaci moderních informačních systémů. Na tento trend musejí být připraveni především budoucí projektanti IS. V příspěvku jsou nastíněny hlavní záměry výuky teorie neuronových sítí ve volitelném předmětu Umělá inteligence a expertní systémy na Ekonomické fakultě VŠB - TU Ostrava a je prezentován simulátor neuronové sítě, který bude používán ve cvičeních k tomuto předmětu od školního roku 1997/98, spolu s připravenými typovými úlohami.

1. ÚVOD

Problematika neuronových sítí je progresivní, ale novou a poměrně málo známou metodou, která se jen zřídka využívá při budování informačních systémů. O jejich přednostech je teprve nutno uživatele přesvědčit. Z tohoto důvodu jsme považovali za vhodné využít ve výuce simulátor neuronové sítě, který vznikl jako vedlejší produkt disertační práce k usnadnění experimentů. Program původně napsaný v TurboPascalu byl přepracován v programovacím jazyce Visual C++ pro využití pod operačními systémy Windows 3.11, Windows NT a Windows 95. Pro možnost použití pod MS-DOS je dále udržovaná starší verze v TurboPascalu.

Významným předpokladem pro vývoj programu se postupně stala jeho názornost a snadná obsluha. Student má možnost volit různé typy sítí, vstupní data se zadávají v přehledných tabulkách a rovněž výsledky je možno si prohlédnout v tabulkách, které poskytují srovnání zadaných a vypočtených hodnot v jednotlivých krocích. V souboru výsledků je možno se pohybovat pomocí tlačítek nebo rolovacích lišt. Studenti tak mohou sledovat, jak se postupně vypočtené hodnoty přibližují skutečným.

Abychom seznámili studenty s možností praktického využití teorie neuronových sítí, jsou zpracovány pro každou síť typové úlohy včetně nastavení potřebných parametrů. Pro tyto úlohy jsou připravena trénovací data, která vycházejí z reálných potřeb praxe.

Účelem úloh je pouze ukázat základní směry aplikačního využití uvedené teorie, aby z nich studenti mohli vycházet při realizaci vlastních projektů.

2. ZÁKLADNÍ SMĚRY PŘÍPRAVY

Při přípravě plánu výuky jsme vycházeli především ze zkušenosti z využití teorie neuronových sítí ve výrobním a servisním podniku. S ohledem na dosavadní výsledky se ukazují tyto hlavní směry využití neuronových sítí:

- v expertních učicích se systémech*
- v datových pumpách.*

2.1 Využití neuronových sítí v expertních systémech

2.1.1 Prognózování

V účetním zpracování se často musí provádět více či méně přesné odhady pravděpodobných údajů v příštím roce nebo několikaletém výhledu. Pokud dosavadní vývoj těchto veličin není čistě náhodný, je možno s pomocí neuronové sítě vysledovat určité trendy a s větší pravděpodobností odhadnout hodnoty v příštím období. Typickým příkladem je růst cen vlivem inflace, snižování materiálové náročnosti výrobků apod. V servisním středisku je možno sledovat průměrné náklady na opravy v souvislosti se zastaráváním nebo naopak inovací strojního parku.

2.1.2 Hodnocení vstupů

V praktické činnosti se často setkáváme s výběrem z několika možností. Na základě určité množiny dat můžeme objektivněji využít naše dosavadní zkušenosti a doporučit pořadí výběru podle určitých kritérií. Příkladem může být hodnocení výrobků přibližně stejného sortimentu podle dosavadní kvality, hodnocení pracovníků podle výkonu apod.

2.1.3 Sledování závislosti

Při rozhodování máme často neúplné nebo nepřesné údaje. Je užitečné vědět, jak v minulosti určité veličiny závisely na ostatních, abychom mohli lépe posoudit spolehlivost našich poznatků. Například nároky na materiál, pracovníky jednotlivých profesí ap. v servisním středisku závisí na typu a rozsahu opravy. Je možno sledovat dobu bezporuchového chodu zařízení v závislosti na jeho složkách.

2.1.4 Klasifikace

V praxi je často nutné zařadit objekty reálného světa do tříd podle podobných vlastností. Vedle toho mohou být vstupní informace zatíženy šumem, který je nutno eliminovat. Možnost takové klasifikace poskytuje například Kohonenova kompetiční metoda.

2.2 Datové pumpy

Jednotlivé organizace disponují v současné době rozsáhlými databázemi a do popředí se dostává otázka, jak co nejlépe shromážděná data zpracovat a vyhodnotit pro účely rozhodování.

Databázové systémy poskytují rozsáhlý aparát k tomu, aby uživatel mohl vybírat efektivně potřebné informace. Podmínkou ovšem je, aby údaj byl v databázi k dispozici. S využitím neuronové sítě je možno vybíraná data obohatit také o údaje, které sice v databázi přítomny jsou, ale pouze ve formě vztahů a nikoliv fyzických dat. Typickou situací je odhad údaje v následujícím roce z dosavadních trendů.

Do modelu neuronové sítě budou importována data především v **číselné podobě**. Je však možný převod dalších typů dat.

Znakové položky je možno převést na vstupní data neuronové sítě tak, že

- každému novému symbolu přiřadíme další celé číslo v pořadí od 1,
- každému novému symbolu přiřadíme novou vstupní proměnnou, která nabývá hodnotu 1 v případě, že symbol v dané větě existuje a 0 pokud neexistuje.

Příkladem může být prognózování materiálové nebo profesní náročnosti oprav, jestliže je druh opravy zadán textem. Textový údaj nemůžeme jednoduchým způsobem do datové pumpy realizované neuronovou sítí vložit, program však může každému novému identifikátoru přiřadit číslo opravy. Ani takový vstup by však pro neuronovou síť typu ADALINE nebyl vhodný. Vhodnější se ukazuje vytvořit n proměnných s názvem rovným typu opravy a hodnotou 1 = oprava provedena, 0 = oprava neprovedena.

Datové položky je možno převádět na číselné tak, že se stanoví počet časových jednotek od určitého počátečního data.

Součtované položky je nutno před vstupem do neuronové sítě rozložit na jednotlivé složky (například hodnoty za první čtvrtletí, pololetí, tři čtvrtletí a rok).

Problematika datových skladů a jejich využití, tj. datových pump realizovaných neuronovými sítěmi v servisní organizaci je podrobněji zkoumána v disertační práci.

3. STRUKTURA A FUNKCE PROGRAMU

3.1 Struktura programu

Základem datového modelu simulačního systému je stromová struktura seznamů, realizujících jednotlivé prvky neuronové sítě. Kořenem stromu je uzel síť (net). Další úrovně jsou reprezentovány uzly vrstva (layout) a neuron. Nejnižší úroveň tvoří jednotlivé synaptické váhy.

Pro realizaci stromové struktury byl zaveden datový typ seznam (list) jako množina ukazatelů. To umožnilo využívat stejných základních operací - CreateList pro

vytvoření seznamu, GetListItem pro čtení a SetListItem pro zápis do seznamu - na uzlech libovolného typu a tím výrazně zjednoduší strukturu programu.

Popsané seznamy jsou vytvářeny dynamicky v čase běhu programu a tím je zajištěna univerzálnost sítě. Popis struktury je uložen v textovém souboru a obsahuje počet vrstev, počet vstupů a počet neuronů v každé vrstvě a další parametry sítě jako koeficient učení a počet iterací.

3.2 Funkce programu

Simulátor neuronové sítě realizuje tyto funkce: Definování charakteristik sítě, vytvoření a změna trénovací množiny, testování a učení, zobrazení výsledků a odpověď na dotaz. Pořadí použití funkcí je kontrolováno, nelze například vypisovat výsledky pokud neproběhlo učení.

3.2.1 Definování charakteristik neuronové sítě.

V této funkci se definuje název sítě, zvolí se typ sítě a její topologie a určí se parametry, které ovlivňují činnost sítě.

Název neuronové sítě je zároveň názvem všech řídicích a datových souborů, které simulátor neuronové sítě používá. Jednotlivé soubory dané sítě jsou odlišeny svou příponou. Vedle názvu sítě je také možno uvést slovní popis vysvětlující účel a funkci sítě.

Typ sítě definuje model, který je vzorem pro vytvoření sítě. V současné době může být:

- adaptivní lineární síť
- síť s aktivační funkcí neuronů tvaru sigmoidy
- síť využívající Kohonenovy metody soutěže

Další modely jsou postupně doplňovány.

Topologie sítě je určena počtem vrstev a počtem neuronů v každé vrstvě. V současné verzi programu je každý neuron nižší vrstvy propojen s každým neuronem bezprostředně vyšší vrstvy. Uvedenými údaji je zároveň určena struktura trénovací množiny. Počet vstupních proměnných je roven počtu neuronů v nejnižší vrstvě, počet výstupních proměnných je počet neuronů v nejvyšší vrstvě.

Vstupním a výstupním proměnným je možno zadat pro lepší orientaci jména, které se pak u těchto proměnných objevují v dalších funkcích.

Parametry sítě určují chování sítě v průběhu učení. Definuje se

- koeficient učení vah a způsob jeho modifikace:
 - hodnota, kterou se tento koeficient násobí, pokud úprava synaptické váhy nemění znaménko
 - hodnota, kterou se tento koeficient dělí, jestliže změna váhy má opačné znaménko než v předchozím časovém kroku
- koeficient vlivu předchozí změny vah
- nejvyšší povolenou chybu

- nejvyšší počet iterací
přitom jsou-li zadány chyba i počet iterací, učení končí při splnění alespoň jedné z obou podmínek.

U sítě s aktivační funkcí sigmoida se navíc zadávají

- horní a dolní meze vstupních a výstupních proměnných
- strmost sigmoidy

Všechny parametry jsou při definování nové sítě automaticky nastaveny tak, aby dávaly dobré výsledky ve většině případů. V průběhu práce se simulátorem je možno tyto údaje měnit a také je možné modifikovat topologii sítě. Pouze změna počtu vstupních a výstupních proměnných je blokována v případě, že již existuje trénovací množina.

3.2.2 Funkce pro modifikaci trénovací množiny

Funkce **Přenos** umožňuje vytvořit trénovací množinu většího rozsahu z textového souboru. Jednotlivé položky jsou definovány svým pořadím v rámci věty. Takto je možno převzít např. data, která byla exportována z některého databázového nebo tabulkového systému.

Změna umožňuje vytvořit trénovací množinu menšího rozsahu nebo měnit existující trénovací množinu. Edituje se vždy jeden vzor trénovací množiny. Data se zobrazí ve dvou tabulkách, obsahujících hodnoty vstupních a výstupních proměnných. Položka pro vstup nebo opravu dat se volí stisknutím myši. Při větším počtu proměnných rolují jejich hodnoty zároveň s jejich názvy.

3.2.3 Učení a testování

Ve funkci **Učení** se v souladu se zadanými charakteristikami neuronové sítě upravují synaptické váhy podle všech vzorů trénovací množiny s cílem minimalizovat celkovou chybu sítě. Průběh adaptace se zapisuje do souboru výsledků.

Naopak při **Testování** se z trénovací množiny vyloučí poslední vzor a zbývající vzory jsou opět předkládány neuronové síti. Po adaptaci vah se vypočtený výstup sítě porovná s požadovanou hodnotou posledního vzoru. Výsledky srovnání v jednotlivých iteracích se opět protokolují.

V obou případech se po ukončení adaptace vypíše počet iterací a chyba v poslední iteraci. Protože učení trvá v případě rozsáhlé trénovací množiny nebo složité topologie zpravidla dlouho, je možno si adaptované váhy uložit do souboru a později je opět obnovit.

3.2.4 Zobrazování výsledků a dotazy

Funkce **Výsledky** slouží k přehlednému zobrazení protokolu o průběhu adaptace. V tabulkách jsou srovnány pro každý krok požadované a vypočtené výstupy. Je možno sledovat, jak se vypočtené hodnoty postupně přibližují správným. Pokud

proběhlo učení, zobrazují se všechny vzory a vypočtené hodnoty, po provedení funkce test se srovnává pouze poslední vzor s jeho prognózou.

Funkce **Dotaz** umožňuje využívat již naučenou neuronovou síť. Zadají se hodnoty vstupních proměnných a systém pro ně vypočte hodnotu výstupu.

4. TYPOVÉ ÚLOHY

Aby se mohli studenti případně další uživatelé lépe orientovat ve zkoumané problematice, bylo vytvořeno několik typových úloh, které se budou postupně doplňovat. Zároveň tento systém ověřujeme na skutečných datech, abychom měli pro srovnání údaje o jeho funkci v reálných podmínkách.

Úloha č. 1 - násobení

Tato úloha je nejjednodušší, výuka s ní začne. Je připraven soubor o jedné vstupní a jedné výstupní veličině, přičemž výstupní veličina je násobkem vstupní veličiny. Program má za úkol tuto závislost "uhádnout".

Po prověření funkce na připravených datech si studenti mohou volit sami jiné závislosti vstupní a výstupní veličiny a sledovat, jak systém řeší jejich úlohu.

Další modifikace spočívá v tom, že vstupní data budou zadávána nepřesně, se šumem, a studenti sledují, jak se tyto odchylky projeví na přesnosti odhadu.

Úloha č. 2 - lineární funkce

Modifikace předchozí úlohy - jedna výstupní veličina závislejší na několika vstupních. Pro samostatnou práci studentů platí zásady uvedené v úloze č. 1. Dále je možno rozšířit úlohu o možnost sledování závislosti několika výstupních veličin na několika vstupních veličinách.

Úloha č. 3 - odhad vývoje cen

Praktická aplikace předchozí úlohy - sledujeme vývoj cen určitého výrobku za předpokladu jeho možnosti aproximace polynomem. Studenti mohou použít data, se kterými se sami setkali a ověřit si chování systému v reálné situaci.

Úloha č. 4 - prognóza časové náročnosti oprav

Úloha vychází z reálného požadavku při řešení disertační práce - na základě dosavadních informací o průběhu oprav stanovit přibližnou časovou náročnost jednotlivých typů oprav z hlediska zabezpečení pracovníky potřebných profesí. V této úloze jsou definovány 3 druhy oprav a jejich realizace pracovníky 5 profesí. Studenti si mohou vyzkoušet vlastní úlohy tohoto typu. Je možné například sledovat zabezpečení projektu materiálem, financemi apod.

Úloha č. 5 - hodnocení výkonu pracovníků

Úloha patří do personálního plánování. Předpokladem pro to, aby pracovník splnil zadaný úkol je jeho způsobilost jako kombinace kvalifikace, což je činitel přesně definovaný (SŠ, VŠ, vyučen v oboru) a schopnosti, což je činitel plauzibilní a proměnný v čase. V dané úloze je uvažováno pět pracovníků servisní organizace a pět úkolů, které při opravě automobilu musejí splnit. Přitom některý z pracovníků zvládne svůj úkol lépe a jiný hůře. Na základě hodnocení výsledků po dokončení práce dokáže program pro každou zakázku doporučit vhodné pracovníky.

Studenti mohou opět definovat své vlastní úlohy, např. hodnocení spolehlivosti dodavatelů, prognózování uplatnění firem na trhu a jiné.

Na tomto příkladě můžeme také vysvětlit, jak je důležité zahrnout do zkoumání všechny atributy. Jestliže do našich úvah nezahrneme délku, po jakou pracovníci na úkolu pracují, neuronová síť také konverguje, ale až po několikanásobně větším počtu iterací.

Úloha č. 6 - příklad diagnostiky v medicíně

Tento příklad ukazuje souvislost neuronové sítě s expertními systémy. Jsou definovány soubory příznaků nemocí a možných diagnóz. Trénovací množinou jsou dvojice vzorů skládající se ze skupiny příznaků a odpovídající diagnózy. Po naučení je možno zadávat příznaky a dotazovat se na pravděpodobnou diagnózu.

Zde můžeme znázornit souvislost s expertním systémem využívajícím produkčních pravidel. Při realizaci pomocí produkčních pravidel nám lékař - odborník musí sdělit přesný postup, jak určuje u pacientů diagnózu. Pro neuronovou síť nám poskytne jako trénovací množinu kartotéku. V prvním případě jsou výsledky exaktně definované, ve druhém získané učením. Neuronová síť pak může být zdrojem informací pro následné definování pravidel a vytvoření kvalitního expertního systému.

Úloha č. 7 - rozpoznávání obrazců

V této úloze seznámíme studenty s principy rozpoznávání obrazců. Pro přípravu použijeme tabulku $n \times m$ polí, sloupce označíme písmeny, řádky číslicemi. Zakreslením obrazce do tabulky a jeho převedením na binární tvar získáme hodnotu vstupních proměnných $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n \dots$, které tvoří trénovací množinu neuronové sítě. Po naučení dokáže neuronová síť rozpoznat zadaný objekt. Ukážeme, jak neuronová síť identifikuje i neúplné vzory, pokud obsahují údaje dostatečné k odlišení od jiných vzorů. Srovnáme výsledky perceptronové sítě s aktivační funkcí sigmoida s výsledky využitím Kohonenovy metody kompetice.

Úloha č. 8 - datová pumpa

Úloha znázorňuje možnost propojení neuronové sítě a databáze. Databáze poskytuje údaje neuronové síti, která je vyhodnotí a nalezne mezi nimi příslušné závislosti. Nalezené souvislosti pak mohou zpět doplnit databázi o nová fakta.

Vstupní data, která byla pořízena v databázovém systému FoxPro a exportována do textového souboru (příkazem `copy to jméno_souboru type delimited`), jsou přečtena do našeho programu, neuronová síť se naučí a vyhodnotí závislosti mezi daty.

5. VÝSLEDKY PRAKTICKÉHO OVĚŘENÍ MODELU

Pro vyhodnocení činnosti programu v reálném prostředí byly jako trénovací data zvoleny položky rozpočtu režijních nákladů za roky 1996 a 1997 jednoho vybraného střediska konkrétní servisní organizace.

K dispozici jsme měli údaje za jednotlivá čtvrtletí, tj. pro každou položku 8 hodnot. Z toho 7 hodnot jsme použili jako trénovací množinu, po adaptaci neuronové sítě jsme vypočetli odhad hodnoty za 4. čtvrtletí 1997 a porovnáním se skutečnou hodnotou stanovili relativní chybu.

Pro velmi přibližné porovnání jsme měli plánované a skutečné hodnoty všech veličin za rok 1997, z nichž jsme opět vypočetli relativní chybu a obě chyby porovnali.

Z 20 zkoumaných položek byly odhady pomocí neuronové sítě ve srovnání s intuitivním odhadem bez pomoci expertního systému:

ve 4 případech výrazně lepší (více než 3 krát menší relativní chyba)

v dalších 12 případech lepší

pouze ve 4 případech horší

v 7 případech byla relativní chyba odhadu menší než 10%, v dalších 8 případech 10% - 30%.

Horších výsledků dosahovala neuronová síť tam, kde neexistoval žádný trend a data byla náhodilá. V takovém případě se nejvíce projevila výhoda zkušenosti pracovníků před odhadem za pomoci exaktního výpočtu.

6. ZÁVĚR

Popsaný program včetně typových úloh je ověřován v současné době při výuce neuronových sítí ve volitelném předmětu Umělá inteligence a expertní systémy na Ekonomické fakultě VŠB - TU Ostrava. Rovněž byl distribuován do dvou organizací, kde se předpokládá provozní využití neuronové sítě, k předvedení možností, které tato metoda poskytuje a k vyhodnocení vztahů mezi položkami databáze. Na jednom z uvedených pracovišť již předběžné zkoušky proběhly a výsledky dávají dobrý předpoklad pro praktickou realizaci tohoto modelu.

Literatura

- 1 Ivo Vondrák: Umělá inteligence a neuronové sítě, VŠB - TU Ostrava 1994
- 2 Jiří Šima, Roman Neruda: Teoretické otázky neuronových sítí. MATFYZPRESS Praha 1996