

Technická univerzita v Liberci

Ekonomická fakulta

Studijní program: M 6209 Systémové inženýrství a informatika

Studijní obor: 6209T021 Manažerská informatika

Řízení skladového hospodářství pomocí metod umělé inteligence

Warehouse management using artificial intelligence methods

DP-EF-KIN-2010-24

Ondřej Švajdler

Vedoucí práce: Ing. Dana Nejedlová, Ph.D.

Konzultant : doc. Ing. Josef Sixta, CSc.

Počet stran: 78

Počet příloh: 2

7. května 2010

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci, se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že technická univerzita v Liberci (TUL) nezasahuje do mých práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užijí-li diplomovou práci, nebo licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci dne 5. 1. 2010

Podpis

Poděkování

Chtěl bych touto cestou poděkovat všem, kteří mi při psaní diplomové práce jakýmkoliv způsobem pomohli. Zvlášť bych chtěl poděkovat Ing. Daně Nejedlové, Ph.D., vedoucí diplomové práce, za její pomoc při konzultacích odborných témat z pohledu umělé inteligence a doc. Ing. Josefu Sixtovi, Csc. za konzultaci logistických témat. Dále zaměstnancům firmy StSoftware Liberec s.r.o. za pomoc při vytváření programu v praktické části práce.

Anotace

Tato práce si klade za cíl zkoumat použití vybrané metody umělé inteligence ve skladovém hospodářství. Především její schopnost klasifikovat položky umísťované na sklad a porovnat její výsledky s metodou ABC, která je v praxi nejčastěji používanou metodou ke klasifikaci zboží. Výstupem práce je vyhodnocení o tom, která z použitých metod umožní firmě minimalizovat skladovací náklady. V první kapitole teoretické části je popsána logistická část práce. V ní jsou definovány pojmy skladování, náklady na skladování, metody používané k uskladnění zásob a zjišťování minimální velikosti pojistných zásob na skladě. Ve druhé kapitole teoretické části je popsáno odvození, princip fungování a aplikace zvolené metody umělé inteligence. V praktické části jsou popsány jednotlivé bloky programu, pomocí kterého je metoda ABC a metoda umělé inteligence aplikována na vstupní data. V závěru práce je srovnání výsledku obou metod a stanovení toho, která použitá metoda se ukázala být efektivnější.

Klíčová slova

skladování, náklady na skladování, metoda ABC, pojistná zásoba, neuron, neuronová síť, synaptické váhy, metoda zpětného šíření chyby, klasifikace

Annotation

This theses examines using of selected methods of artificial intelligence in the stock warehousing. In particular, its ability to classify items affixed to the store and compare the results with the ABC method, which is in practice most commonly used method to classify goods. The output of the theses is to evaluate which of the methods used will enable the company to minimize costs. In the first chapter the theoretical part is described logistical part of the work. In it are defined the concept of storage, storage costs, the methods used for the storage of stock and calculation the minimum size of safety stocks. In the second chapter describes the theoretical derivation, the principle of operation and application of the chosen methods of artificial intelligence. The practical part is described the each of blocks of the program through which the ABC method and the method of artificial intelligence applied to the input data. At the end of the work is the comparation the results of both methods and determine which method is used has proven to be effective.

Keywords

warehousing, storage costs, ABC method, safety stock, neuron, neural network, synaptic weights, backpropagation, classification

Obsah

1 Úvod.....	12
2 Zhodnocení současného stavu.....	14
3 Teoretická část.....	16
3.1 Logistická část.....	16
3.1.1 Úvod do logistiky.....	16
3.1.2 Logistické náklady.....	16
3.1.3 Koncepce celkových nákladů.....	17
3.1.4 Skladování.....	19
3.1.4.1 Základní funkce skladování.....	20
3.1.4.2 Charakter skladování.....	21
3.1.4.3 Funkce skladů.....	21
3.1.4.4 Nejběžnější chyby při skladování.....	22
3.1.4.5 Automatizace skladu.....	22
3.1.4.6 Změna funkce skladu.....	22
3.1.5 Metody využívané v logistice.....	23
3.1.5.1 Metody sloužící k analýze logistických procesů.....	23
3.1.5.2 Statické metody.....	23
3.1.5.3 Metody operační analýzy.....	24
3.1.5.4 Metody heuristické.....	24
3.1.5.4.1 Expertní systémy.....	24
3.1.5.4.2 Systémy tvořivého myšlení.....	25
3.1.6 Analýza ABC.....	25
3.1.7 Zásoby podniku.....	27
3.1.7.1 Pojistné zásoby	27
3.1.7.2 Výpočet pojistných zásob.....	28
3.2 Část umělé inteligence.....	29
3.2.1 Úvod do umělé inteligence.....	29
3.2.2 Obecná charakteristika neuronové sítě.....	30
3.2.3 Neuronová buňka.....	30
3.2.4 Matematický model neuronové sítě.....	32
3.2.4.1 Formální neuron.....	32
3.2.4.2 Neuronová síť.....	35
3.2.4.2.1 Organizační dynamika.....	36
3.2.4.2.2 Adaptivní dynamika.....	37
3.2.4.3 Použití neuronové sítě.....	37
3.2.4.3.1 Predikce.....	38
3.2.4.3.2 Klasifikace.....	38
3.2.4.3.3 Řízení.....	38
3.2.5 Modely neuronových sítí.....	38
3.2.5.1 Síť perceptronů.....	38
3.2.5.2 Vícevrstvá neuronová síť a back-propagation	40
3.2.5.2.1 Organizační dynamika.....	40
3.2.5.2.2 Aktivní dynamika.....	41
3.2.5.2.3 Adaptivní dynamika.....	42
3.2.5.2.4 Strategie back-propagation.....	45
4 Praktická část.....	47

4.1 Úvod do praktické části.....	47
4.2 Program Neuron Network.....	47
4.2.1 Vlastní nastavení.....	49
4.2.2 Generování dat.....	50
4.2.2.1 Generované atributy.....	50
4.2.2.2 Vlastnosti generovaných dat.....	51
4.2.2.3 Datové toky v neuronové síti.....	52
4.2.3 Klasifikace metodou ABC.....	53
4.2.4 Klasifikace pomocí neuronové sítě.....	54
4.2.4.1 Normování vstupních dat neuronové sítě.....	55
4.2.4.2 Proces učení neuronové sítě.....	56
4.2.4.3 Nastavení ukončující podmínky učení.....	58
4.2.4.4 Testování a nastavování parametrů neuronové sítě.....	60
4.2.4.5 Klasifikace hodnot výstupních neuronů.....	62
4.2.5 Určení pojistné zásoby.....	63
4.2.6 Výpočet nákladů.....	63
4.3 Prezentace výsledků.....	65
4.3.1 Neuronová síť s jedním vstupním atributem.....	65
4.3.2 Neuronová síť s více vstupními atributy.....	69
4.4 Zahrnutí vedlejších nákladů.....	73
5 Závěr.....	74

Seznam zkratek a symbolů

C# – vysokoúrovňový objektově orientovaný programovací jazyk

JIT – Just in Time, vyrábění výrobků v určeném množství a určeném čase dle požadavků
zákazníka

Seznam tabulek

Tabulka 1: Generovaná data – popis atributů.....	50
--	----

Seznam obrázků

Obrázek 1: Biologický neuron.....	31
Obrázek 2: Formální neuron.....	33
Obrázek 3: Geometrická interpretace funkce neuronu.....	34
Obrázek 4: Architektura sítě perceptronů.....	39
Obrázek 5: Vícevrstvá neuronová síť typu 3-4-3-2.....	41
Obrázek 6: Standardní logistická sigmioda.....	42
Obrázek 7: Gradientní metoda.....	44
Obrázek 8: Overview programu.....	48
Obrázek 9: Advanced panel.....	48
Obrázek 10: Blokové schéma programu.....	49
Obrázek 11: Panel Vlastní nastavení.....	50
Obrázek 12: Tok dat v programu.....	53
Obrázek 13: Paretovo optimum 80:20.....	54
Obrázek 14: Algoritmus procesu učení.....	57
Obrázek 15: Graf celkové chyby učení.....	61
Obrázek 16: Schéma neuronové sítě s jedním vstupním atributem.....	66
Obrázek 17: Skladovací náklady dat z testovací množiny jednot. klas. skupin neuronovou sítí v závislosti na počtu učících cyklů.....	66
Obrázek 18: Úspěšnost klasifikace neuronové sítě dat z testovací množiny.....	67
Obrázek 19: Rozdíl celkových nákladů mezi oběma klasifikačními metodami.....	68
Obrázek 20: Schéma neuronové sítě s více vstupními atributy.....	69
Obrázek 21: Úspěšnost klasifikace položek z testovací množiny u neuronové sítě s více vstupními atributy.....	70
Obrázek 22: Skladovací náklady položek z testovací množiny jednot. klas. skupin u neuronové sítě s více vstupními atributy.....	71
Obrázek 23: Porovnání celkových nákladů metody ABC a neuronové sítě s více vstupními atributy.....	72

1 Úvod

Obor Manažerská informatika jsem si vybral především proto, že studijní plány obsahují dvousemestrový předmět Řízené praxe. Během nich si každý student najde podnik nebo společnost, ve které se může naučit novým praktickým poznatkům nebo použít získané teoretické znalosti z předchozích ročníků. Moje zvolená firma byla StSoftware Liberec s.r.o., jejíž náplní je především zabezpečení sportovních akcí po technické stránce. Tím se rozumí, že firma zajišťuje veškerá potřebná měřicí zařízení k získání dat, ze kterých jsou poté zpracovávány výsledkové listiny. Jejimi produkty jsou tedy informační systémy a programy na vyhodnocení výsledků. Důvod výběru této firmy byl ten, že se jedná o práci v oblasti informatiky a je možnost zde uplatnit znalosti z předmětů Databázové systémy, Informační systémy a Programování. Po skončení praxe mi bylo nabídnuto i nadále v této firmě pracovat.

Jako téma své diplomové práce jsem si zvolil využití metod umělé inteligence ve skladovém hospodářství. Přestože se nejedná o totožnou oblast, ve které působí firma StSoftware, mohl jsem využít znalosti a dovednosti získané v této firmě. Především se jedná znalosti z programování, které jsem použil v praktické části diplomové práce. Její stěžejní část představuje program, pomocí kterého lze vykonat všechny procesy potřebné k dosažení potřebných cílů práce.

Cílem práce bude zkoumání, zdali je možné aplikovat jednu z metod umělé inteligence v nějaké části logistického systému. Logistickou oblastí, kterou jsem si pro svůj výzkum zvolil, je skladování. A to z toho důvodu, že se jedná o jednu z nejdůležitějších a nejvíce sledovaných oblastí v logistice. Firmy k ní upínají svoji pozornost především proto, že skladovací náklady, resp. náklady na udržování zásob, představují převážnou část celkových nákladů jedné firmy.

Teoretická část diplomové práce je rozdělena na dvě části. První část je zaměřena na logistickou oblast. V ní jsou popsány a analyzovány veškeré logistické náklady podniků a jejich vliv na celkové náklady jednotlivých firem. Dále je zde definován termín skladování včetně jeho funkcí. V poslední kapitole logistické teoretické části jsou vyjmenovány metody skladování, kterými je možno klasifikovat výrobky na sklad, a vysvětlen termín pojistná zásoba. Ve druhé části teorie je popsána metoda umělé inteligence, jež byla v diplomové práci použita, včetně oblasti, ze které byla odvozena.

V úvodu praktické části jsou definována pravidla, která musela být před začátkem praktické části stanovena. V dalších kapitolách jsou detailně popsány jednotlivé bloky programu, na které je program logicky rozdělen. V závěru praktické části je vyhodnocení dvou modelových situací, které slouží k porovnání úspěšnosti obou klasifikačních metod. Výsledky jsou zde prezentovány pomocí grafů.

2 Zhodnocení současného stavu

V dnešní době se pro umístování výrobků na sklad v drtivé většině používají klasifikační metody, které rozdělí vstupní zboží do jednotlivých skupin. Každé z těchto skupin poté podnik věnuje jinou pozornost v závislosti na tom, jaký jim položky z dané skupiny přinášejí zisk.

Jak publikovala Petra Zajíčková v [19] existuje několik třídících hledisek a klasifikací, která lze seřadit podle míry vlivu, s jakou se projevují v koncepci těchto modelů. Prvním klasifikačním hlediskem je regulovatelnost procesu doplňování zásob, který se dále rozděluje do dvou skupin modelů. Do první skupiny lze zahrnout modely, v nichž jsou podniky schopny ovlivňovat průběh doplňování zásob, do druhé skupiny patří modely, kde tato možnost není. Toto hledisko je pro praktické využití výhodné především kvůli tomu, že se modely obou skupin liší nejen svoji regulovatelností, ale především tím, že je zde zahrnuta i povaha řešeného problému. Cílem těchto modelů je nalezení optimální strategie doplňování zásob. Druhým třídícím kritériem, podle kterého lze zásoby klasifikovat, je druh uspokojované spotřeby. Toto hledisko se člení na tři klasifikační stupně – modely deterministické a stochastické, modely dynamické a modely s konstantní a variabilní spotřebou. Posledním třídícím hlediskem je režim doplňování zásob, kde se kromě modelů stacionárních a nestacionárních rozeznávají ještě režimy s volnými a pevnými termíny pro objednání zásob.

Klasifikace podle předem stanoveným zásad je také uvedena v odborné publikaci [6]. Její autoři poukazují na fakt, že není možné ve velkých skladovacích jednotkách věnovat stejně velkou pozornost všem skladovým položkám, proto je nutné tyto položky sdružovat do určitých skupin podle jednoznačně stanovených pravidel. Toto umožní podnikům efektivněji řídit, sledovat a kontrolovat skladové hospodářství. V publikaci je především porovnávána v praxi nepoužívanější metoda ABC s metodou, která je schopna klasifikovat do většího počtu skupin, než je tomu u metody ABC. Metoda se ve volném překladu nazývá Operativní vytváření skupin a vychází ze statických dat, které podnik během minulosti získal. Proto je možné na jejich základě rozčleňovat položky přicházející na sklad do daných skupin. Při vzájemném porovnání s metodou ABC lze vypořádat jednoznačný přínos z hlediska provozu a statistiky výkonu. Také je možné

vzít v úvahu manažerské priority, jako je seznam služeb a skladovací náklady, které jsou prokazatelně lepší než u metody ABC.

Další publikací, která se zabývá klasifikováním výrobků na sklad, je odborný článek [18]. Jeho hlavním tématem je metoda na odhadování maximalizace zisku při použití klasifikace metodou ABC. V tomto matematickém modelu je třeba si stanovit určité úrovně peněžních obrátů, které jsou spojené s maximálním dosažitelným ziskem. Kromě stanovení zisku lze pomocí tohoto modelu zobrazit peněžní obraty jako lineární funkce u nejvíce ziskových klasifikovaných skupin. To umožňuje získat na tyto obraty zpětnou vazbu, což slouží jako měřítko k odhadu relativní úspěšnosti řízení zásob daného podniku. Na základě zjištěných rozdílů může podnik zlepšit klasifikaci, resp. řízení zásob, o významnou nákladovou položku.

3 Teoretická část

3.1 Logistická část

3.1.1 Úvod do logistiky

Logistika využívaná v hospodářské praxi patří k mladým vědním disciplínám. Výraznější prosazení logistiky jako takové nacházíme na počátku padesátých let dvacátého století, kdy koncentrace výrobních kapacit předstihla možnosti dosavadních metod distribuce hotových výrobků. Před tímto obdobím nebylo nutné věnovat mimořádnou pozornost procesům přemísťování hotového zboží finálním zákazníkům.

Dalším výrazným faktem, který urychlil zavedení logistiky do hospodářské praxe, byl postupný přechod od trhu výrobce k trhu zákazníka. Trh výrobce lze charakterizovat jako výrobu omezeného sortimentu výrobků ve velkých výrobních sériích. Postupem času se přešlo na trh zákazníka, jehož důsledkem byla potřeba rychlé inovace výrobků širokou sortimentní paletou.

Změna z trhu výrobce na trh zákazníka měla za následek nejen změny v chápání marketingu, ale také to znamenalo silný tlak na snižování nákladů v oblasti distribuce, které se změnou neúměrně zvyšovaly. To vše vedlo k vývoji nových, levnějších a účinnějších metod distribuce hotových výrobků a surovin. Následné pokusy o uplatnění komplexnějšího řešení však narážely na nedostatek technických prostředků, modernějších technologií a výpočetní techniky.

Na závěr logistického úvodu je uvedena charakteristika logistiky podle B.I. Ghosta z Univerzity v Mannheimu:

„Logistika představuje ekonomický postoj, manažerskou a tvůrčí koncepci, která v podmínkách integrovaného řetězce vytváří přidané hodnoty, v kombinaci se slučitelnou organizační realizací vede k přesné alokaci odpovědnosti za všechny pohyby a zásoby použitých materiálů.“¹

3.1.2 Logistické náklady

Logistické náklady jsou pro většinu firem velmi významné a jsou vyvolávány či tvořeny činnostmi, které podporují celý logistický proces. Jedná se např. o náklady

¹ Citováno z [2]

na dopravu, ložné operace, manipulaci, skladování, balící procesy. Náklady jsou společně se ziskem velmi důležitým ekonomickým ukazatelem každého podniku. V nedávné době se veškerá výrobní činnost obchodních společností odvíjela od základní rovnice

$$cena = náklady + zisk \quad (1.1)$$

V současnosti však tento vztah již ne zcela úplně odpovídá realitě. Cena hotového výrobku je totiž většinou určována trhem. Proto ji neurčuje pouze vlastník (prodejce) zboží, ale také konkurence. V matematické terminologii již není cena závislou proměnnou, a má-li být podnik životaschopný, musí vytvářet určitý zisk, který musí obratem investovat do inovací podnikových zařízení. Na základě předchozích úvah je nutné výše uvedenou rovnici pozměnit do následujícího tvaru

$$náklady = cena + zisk \quad (1.2)$$

Matematickou terminologií tato rovnice definuje, že náklady jsou veličinou závislou. Ekonomicky lze tuto rovnici interpretovat: „Chce-li podnik přežít, musí své náklady snížit tak, aby dosáhly maximálně hodnoty ceny zboží.“²

Každý logistický výkon se rozpadá na dvě složky – logistické služby a logistické náklady. Mezi logistické služby patří především služby zákazníkům a zákaznický servis. Logistické náklady se mohou sledovat u následujících činností – řízení stavu zásob, logistické komunikace, balení, pořizování, podpory servisu a náhradních dílů, vyřizování objednávek a v neposlední řadě i skladování.

3.1.3 Koncepce celkových nákladů

Koncepce celkových nákladů je zcela zásadním prvkem k efektivnímu řízení logistického systému. Výrobní podnik se nesmí zaměřovat jen na jednotlivé izolované činnosti, ale musí se pokoušet minimalizovat celkové logistické náklady najednou. Snížení nákladů v jedné oblasti může vyvolat zvýšení nákladů v oblasti druhé, a to vlivem změny vstupních veličin způsobených snížením nákladů v předcházející oblasti. Toto zvýšení poté může být vyšší než snížení nákladů v předcházející oblasti.

„Logistika s nejmenšími celkovými náklady je takový stav, kdy při dosažení stanovené úrovně logistického servisu se minimalizuje součet všech logistických nákladů.“³ Pro úspěšnou implementaci analýzy nákladových vazeb však musí mít

2 Citováno z [2]

3 Citováno z [2]

management k dispozici příslušná data o jednotlivých druzích nákladů. Management by měl stanovovat politiku velikosti a obratu zásob na základě kvalitních znalostí o nákladech na udržování zásob, o celkových nákladech logistického systému a o potřebné strategii zákaznického servisu. Nákladové oblasti, které jsou vzájemně propojeny, lze rozdělit do 6 základních částí:[2]

1. Úroveň zákaznického servisu

Dobrý zákaznický servis podporuje spokojenost zákazníků. Lze jej definovat jako filozofii orientace na zákazníka, která spojuje a řídí všechny složky napojení na zákazníka v rámci stanoveného poměru nákladů a poskytování služeb. Také je vstupem do logistického systému a měl by zprostředkovat dobré služby, které podporují spokojenost zákazníků.

2. Přepravní náklady

Logistickou činností s velmi důležitým významem je vlastní přesun materiálů a zboží z místa výroby do místa spotřeby. Do přepravních nákladů lze také započítat pohyb materiálu až do konečného místa jeho likvidace. Zajištění přepravy zahrnuje výběr způsobu přepravy (např. letecké, silniční, vlakové, lodní nebo potrubní), výběr přepravní trasy a výběr dopravce.

3. Náklady na udržování zásob

Řízení stavu zásob má za úkol udržovat takovou úroveň zásob, aby bylo při minimálních nákladech dosaženo co nejvyšší úrovně zákaznického servisu. Do nákladů na udržování zásob se započítávají náklady na pořízení zásob a náklady na likvidaci zastaralého zboží.

4. Skladovací náklady

Skladování se významně podílí na tvorbě užitné hodnoty prostřednictvím času a místa. Současně umožňuje, aby bylo vyrobené zboží uchováno pro pozdější spotřebu. Je tedy výhodné zboží skladovat v blízkosti místa následné spotřeby nebo místa další přepravy. Náklady, které vznikají skladováním, jsou ovlivněny výběrem místa výrobních kapacit a skladů podniku. Zahrnují všechny náklady vznikající v návaznosti na změnu počtu nebo změnu umístění skladů. V porovnání s ostatními

logistickými aktivitami představuje doprava jednu z největších samostatných nákladových částek z celkových nákladů.

5. Množstevní náklady

Hlavní logistické množstevní náklady mají svůj původ v množstvích, která jsou použita v toku materiálu. Jsou to náklady spojené se změnami v nakupovaných množstvích a se změnami ve výrobě či prodeji. Na tyto náklady ovšem nelze nahlížet izolovaně, protože ovlivňují řadu dalších nákladů.

6. Náklady na informační systém

Informační systém podnik používá k přijímání objednávek od zákazníků, ke kontrole stavu objednávek, k samostatnému vyřízení objednávek a jejich dostupnosti pro zákazníky. Součástí tohoto systému je i kontrola stavu zásob, fakturace a stavu pohledávek.

3.1.4 Skladování

Skladování je jednou z nejdůležitějších částí logistického systému. Tvoří spojovací článek mezi výrobcí a zákazníky a také zabezpečuje uskladnění produktů (např. surovin, dílů, hotových výrobků) v místech jejich vzniku a také mezi místem výroby a místem spotřeby. Současně poskytuje managementu informace o stavu, podmínkách a rozmístění skladových produktů. Jednotlivé jednotky sloužící ke skladování jsou sklady, které umožňují překlenout prostor a čas. Výrobní zásoby zajišťují plynulost výroby a plynulé zásobování obyvatelstva.

Odhaduje se, že na světě existuje okolo 750 000 skladovacích zařízení – od nejmodernějších profesionálně řídicích skladů až po podnikové skladovací garáže, místnosti a drobné sklady v rámci prodejen. Skladovací zařízení lze rozdělit podle následujících kritérií:

- vybavenost skladu včetně organizace a řízení skladů
- rozsah a centralizace skladů
- vlastní nebo cizí skladování
- poloha skladu
- úroveň zásob uložených ve skladu

3.1.4.1 Základní funkce skladování

Skladování lze rozdělit na tři základní funkce. Jde o činnosti mající za úkol přesun zboží, jejich uskladnění a funkci přenosu informací.

„Přesun produktů je možné dále členit na:

- příjem zboží – vyložení, vybalení, aktualizace záznamů, kontrola stavu zboží
- ukládání zboží – přesun produktů ze skladu, uskladnění a jiné přesuny
- kompletace zboží – přeskupování zboží podle jednotlivých objednávek
- překládka zboží – z místa příjmu do místa expedice, vynechání uskladnění
- expedice zboží – zabalení a přesun zásilek do dopravního prostředku

Uskladnění produktů se dělí na:

- přechodné - uskladnění nezbytné pro doplňování základních zásob
- časově omezené – týká se zásob nadměrných, které se na skladě drží z důvodu sezonní poptávky nebo kvůli zvláštním podmínkám obchodu“⁴

Přenosem informací se rozumí stav zásob, stav zboží v pohybu, umístění zásob, vstupní a výstupní dodávky, personál a využití skladových prostor. Osobní počítače při přenosu informací zastávají při výměně dat velmi podstatnou roli. Nejrozličnější informační systémy značně urychlují, zefektivňují a zkvalitňují přenos informací potřebných k zajištění všech funkcí skladování. Neméně podstatné je v této oblasti i vzájemné propojení počítačů do lokálních sítí.

Podniky se snaží o to, aby skladování, jež má významný vliv i na zajišťování potřebné úrovně zákaznického servisu, způsobilo co nejnižších celkových náklady. Proto se v současné době podstatně zvýšil význam této složky logistického systému podniku a stala se z ní jedna z nejdůležitějších součástí zákaznického servisu.

Skladování také můžeme definovat jako tu část podnikového logistického systému, která zabezpečuje uskladnění produktů (dílů, zboží ve výrobě, hotových výrobků) v místech jejich vzniku a také mezi místem výroby a místem jejich spotřeby. Dále

4 Citováno z [2]

poskytuje managementu informace o stavu, podmínkách a o rozmístění skladových produktů.

3.1.4.2 Charakter skladování

Skladování tradičně zabezpečuje uskladnění produktů v průběhu všech fází logistického procesu. Existují dva základní typy zásob, které podnik potřebuje uskladnit:

- suroviny, součástky a díly – vstupní materiál do podniku
- hotové výrobky – výstup materiálu z podniku

Dalším typem zásob, které potřebuje podnik uskladnit, jsou zásoby zboží ve výrobě a zásoby určené k likvidaci nebo recyklaci. V současnosti většina podniků tyto typy zásob pečlivě sleduje, přestože představují jen malý podíl z celkových zásob. Všechny výše uvedené typy zásob podniky udržují ve skladech z následujících důvodů – snaha o dosažení úspor nákladů na spotřebu, úspor ve výrobě, využití množstevních slev, snaha udržet si dodavatelský zdroj, podpora programů JIT u dodavatelů nebo zákazníků a překlenutí časových a prostorových rozdílů, které existují mezi výrobcem a koncovým zákazníkem.[2]

3.1.4.3 Funkce skladů

Základním úkolem skladu je ekonomické sladění rozdílně dimenzovaných toků. Mezi hlavní funkce skladování patří:

- vyrovnávací funkce – využívá se při odchýlném materiálovém toku a materiálové potřebě z hlediska jejich kvantity
- zabezpečovací funkce – vyplývají z nepředvídatelných rizik během výrobních procesů
- kompletační funkce – pro tvorbu sortimentu v obchodě nebo pro tvorbu individuálních provozů v průmyslových podnicích
- spekuláční funkce – vyplývá z očekávaných cenových zvýšení na zásobovacích a odbytových trzích
- zušlechťovací funkce – zaměřená na jakostní změny uskladnění druhů sortimentu (např. stárnutí, kvašení, zrání, sušení)

Skladů se stále více využívá jako průtokových bodů, které nahrazují svou předchozí funkci. V některých případech se dokonce funkce úschovy produktů úplně obcházejí. To je způsobeno tím, že se podniky ve zvýšené míře snaží zásoby nahradit informacemi, nakupují zásoby v menších množstvích nebo používají sklady jako konsolidační body. Tím podniky získají výhodnější přepravní sazby a sníží si logistické náklady.

3.1.4.4 Nejběžnější chyby při skladování

Je velmi důležité, aby se management snažil odstranit všechny neefektivy, které se vyskytnou při přesunu zboží a zásob, jejich uskladnění nebo přenosu informací v rámci skladu. Tyto neefektivy se projevují různými formami jako jsou například přebytečná nebo nadměrná manipulace, nízké využití skladové plochy, nadměrné náklady na údržbu skladových zařízení nebo zastaralé způsoby příjmu a expedice zboží. Podnik je nucen, chce-li si uchovat svoje postavení na konkurenčním trhu, používat stále přesnější a preciznější systémy manipulace, uskladnění a vyhledávání zboží.

3.1.4.5 Automatizace skladu

Automatizace skladu by měla vycházet z logistických trendů. Vedle hospodárnosti zvoleného systému by měl sklad bez problémů zvládnout budoucí rozmanitost položek, vývoj zásob a především průtok zásob skladem. Automatizaci skladu lze rozdělit podle druhu skladu na výrobní sklad, sklad hotových výrobků a obchodní sklad.

V případě opatřování zásob JIT by se měl výrobní sklad přetvořit na velmi rychlý vyrovnávací zásobník s relativně malou zásobou. Naproti tomu sklad hotových výrobků by měl růst v závislosti na sezonním kolísání poptávky. Také uvnitř skladu existují rozdíly podle rychlosti obratu položek kategorií A, B a C.

3.1.4.6 Změna funkce skladu

Původní funkcí skladu byl dlouhodobě vyrovnávací zásobník. Ideálním řešením pro velké množství zásob a malý skladový průtok se nabízely výškové regálové sklady. Postupem času se však ve výrobě staly stále žádanějšími rychlé krátkodobé vyrovnávací zásobníky, které umožňovaly změnit pořadí zakázek čekajících na zpracování operace nebo přichystání montážních dílů. Také vychystávání může být účelnější, pokud se nejprve shromažďují jednotlivé položky zakázek a teprve potom se předají logistickému prvku kompletně všechny položky od jedné zakázky. V obou případech může být přídatné třídící

funkce podporována provozními pochody. Kombinace skladovací a třídící funkce se požaduje u tzv. konsolidačních center. Tato centra jsou v rostoucí míře provozována logistickými podniky v rámci opatrovací logistiky před finální montáží automobilek.

3.1.5 Metody využívané v logistice

Ve své podstatě je logistika „kvantitativní řízení“, které musí upozorňovat na úzká místa v toku materiálu a současně musí utvářet všechny příslušné procesy od počátku tvorby strategie až po okamžik dodání hotového výrobku koncovému zákazníkovi. Tím, že připustíme kvantitativní aspekt logistiky, neobejdeme se tím ale bez schopnosti použití následujících metod, které lze rozdělit na dvě skupiny – exaktní a heuristické.

Metody exaktní jsou podloženy poznáním exaktních vědních oborů, především pak matematických disciplín a využívají se zejména diagnostické a optimalizační úlohy rozhodovacích procesů.[2]

3.1.5.1 Metody sloužící k analýze logistických procesů

- systémová analýza – slouží k objasnění celkových procesů v podniku
- analýza ABC – vhodná k určení prvků, které mají v celém systému pro ekonomiku největší význam
- hodnotová analýza – je ověřeným způsobem hodnocení hospodárnosti při pohybu toku materiálu
- analýza nákladů – metoda na zjišťování nákladů v pohybových úsecích logistiky

3.1.5.2 Statické metody

Tyto metody lze rozdělit do dvou případů. První případ jsou metody matematické statistiky, které jsou využívány ve formě statistické analýzy pro diagnózu řídicích systémů. Zde se využívá v analýze časových řad, nebo v analýze příčinných vazeb mezi ekonomickými, provozními nebo kvalitativními ukazateli. Výsledkem těchto analýz je zjištění změny trendu nebo zjištění, jaké následné změny vyvolá změna trendu na kauzálních řetězcích.

Ve druhém případě se matematická statistika používá pro různé formy matematického modelování, jenž se zabývá získáváním, popisem a zpracováním údajů

s cílem nalézt zákonitosti hromadných jevů. Příkladem statických úloh je nalezení míry závislosti dvou nebo více náhodných hromadných jevů a odhady neznámého rozložení pravděpodobnosti na základě výběrového souboru empiricky zjišťovaných hodnot, odhady neznámých parametrů nebo ověřování statistických hypotéz.

3.1.5.3 Metody operační analýzy

Pod pojmem metody operační analýzy, resp. operačního výzkumu rozumíme souhrn metod, které pomocí řady matematických disciplín modelují určité stavy technologických procesů nebo procesů rozhodovacích. Z operační analýzy se nejvíce využívá metod teorie zásob, teorie obnovy a teorie front.

3.1.5.4 Metody heuristické

Využívají se obvykle pro rozhodovací procesy s vysokou mírou neurčitosti a v těch úlohách, které s ohledem na svou stabilitu a informační zabezpečení nejsou algoritimizované. Pro řešení úloh v rozhodování, kde není možné aplikovat exaktní metody, se často využívá technik, které zpracovávají intuitivní, avšak vysoce kvalifikované odhady. Tyto metody lze rozdělit do dvou skupin – expertní systémy a systémy tvořivého myšlení.

3.1.5.4.1 Expertní systémy

Expertní systémy se bezprostředně využívají v rozhodovacích procesech diagnostického nebo plánovacího typu. Systémy využívají možností paměťové kapacity výpočetní techniky, její rychlosti zpracování informací a možností určité algoritmizace procesů, které jsou schopny vytvářet určitý předstupeň umělé inteligence. Nástup expertních systémů na přelomu 50. a 60. let 20. století přispěl k formování nové vědní disciplíny a to umělé inteligence. Umělá inteligence se snaží vytvořit programové systémy, projevující v jistém smyslu inteligentní chování. Jednou z možností aplikací je tvorba a využívání expertních nebo kognitivních systémů. Expertní systémy jsou programy založené na znalostech řešících úloh. Expert se při řešení úlohy opírá o svoje znalosti a zkušenosti z řešení podobných úloh v minulosti. Expertní systémy umožňují tyto zkušenosti částečně převzít a prezentovat je tak, aby je mohl program využívat. Kvality řešení úloh tak závisí na znalostech, které systém má k dispozici a pomocí nichž poskytuje odezvu na vstupní data.

3.1.5.4.2 Systémy tvořivého myšlení

Metody tvořivého myšlení se většinou využívají na úrovni strategického rozhodování, případně při racionalizaci podnikového managementu v měnících se podmínkách, které nelze statisticky analyzovat. Základním principem tvořivého procesu je výběr, přetváření a spojování prvků předchozích zkušeností. Analýzy ukázaly, že tvořivý potenciál organizací je často důležitější než ostatní zdroje (finanční, materiálové, lidské atd.). Současné poznatky výzkumu a praxe ukazují, že tvořivost není výsledkem samovolně probíhajících procesů, ale že je možné a nutné ji posilovat vhodným řízením a usměrňováním.

3.1.6 Analýza ABC

Analýzu ABC lze uplatnit s různými modifikacemi v mnoha podnikových oblastech, např. v oblasti řízení pohledávek a závazků, řízení lidských zdrojů, řízení rozpracovanosti ve výrobě, řízení prodeje, ale především v oblasti řízení zásob. Zásoby jsou podstatným logistickým faktorem, který ovlivňuje řízení produkce. Podniky si uvědomují, že vysoké zásoby jsou limitujícím prvkem hospodaření, proto se snaží najít způsoby jejich redukce. Ovšem nevýhodou omezení zásob je narušení plynulosti výroby, neprodejnosti, zvýšení vlastních dodacích lhůt atd.

ABC analýzu lze definovat jako systém diferencovaného řízení zásob. Důvodem difference je to, že je neefektivní věnovat všem položkám daného druhu stejnou pozornost. Proto se aplikuje ABC analýza, která umožňuje zásoby diferencovat a k vymezeným skupinám přistupovat odlišným způsobem. Analýza ABC vychází z Paretova principu 80:20, který spočívá v matematickém vzorci, který odráží nerovnoměrné rozložení bohatství a vyjadřuje skutečnost, že zhruba 20% obyvatel vlastní 80% bohatství. Při aplikaci do logistiky je princip modifikován tak, že 20% výrobků podniku zajistí 80% zisku.

Analýza ABC se uskutečňuje ve čtyřech fázích:

- zjištění hodnoty roční spotřeby pro každou položku
- výpočet procentního podílu na celkové spotřebě
- zjištění procentního podílu na celkovém počtu položek
- definování mezitřídních intervalů

Stejným způsobem lze rozdělit jednotlivé položky sortimentu do tří (či více) skupin (A, B, C) s rozdílným dopadem na sledovaný ukazatel.

Skupina A obvykle představuje jen asi 20% položek s kumulativně 80% podílem na celkové hodnotě sledovaného ukazatele. Představuje položky, kterým je potřeba věnovat nejvyšší pozornost a které by měly být sledovány každodenně. Skupina B je tvořena asi 30% položek, jejichž podíl na celkové hodnotě sledované veličiny představuje asi 15%. Skupina B obsahuje středně důležité položky, které by měly být sledovány často. Skupina C potom zahrnuje zbývající položky se zhruba 5% podílem na hodnotě sledované veličiny. Položky z této skupiny vyžadují menší pozornost, objednávány jsou ve velkých množstvích, jejich stav by měl být kontrolován periodicky.

ABC analýza se aplikuje se specifickým cílem v každé oblasti podniku. U zásob je jiný cíl pro vstupní položky, pro nedokončenou výrobu a jiný pro zásoby hotových výrobků. Cílem aplikace ABC vstupních zásob je optimalizace řízení a plánování položek z hlediska plynulosti výroby, zajištění dostatečného množství vstupů pro řešení aktuálních požadavků na výrobu, snižování rizika nedostatku či hromadění nevyužitelných zásob apod. U aplikace ABC na zásoby nedokončené výroby je cíl zaměřen více na optimální využití výrobní kapacity (zajištění pokrytí volných časů ve výrobě), snižování rizika nedostatku položek pro potřeby dalšího zpracování plynoucího z nadměrné délky průběžné výrobní lhůty apod. Analýza zásob hotových výrobků sleduje a odráží vývoj daného segmentu trhu. Společným cílem všech těchto dílčích oblastí je generování přehledných výstupů, poskytujících aktuální informace o daných položkách potřebných pro řízení.

Před dílčím řešením je nutné jednotlivé skupiny zásob zařadit do komplexního systému řízení a definovat nejdůležitější parametry, které je ovlivňují. Tyto parametry jsou základem pro hledání ukazatele, dle kterého se položky daného druhu zásob rozdělují do skupin. Ukazatel pro vstupní položky může být ovlivněn spotřebou za vymezené časové období, dodací lhůtou a váhou příslušného období. Ukazatel pro nedokončenou výrobu je dán spotřebou za vymezené časové období, průběžnou výrobní dobou a váhou příslušného období apod. Tvorba ukazatele je nejobtížnější částí ABC analýzy. Parametry definované v návaznosti na komplexní systém řízení je v mnoha případech třeba podrobit detailnímu průzkumu a najít nejvhodnější způsob zakomponování do ukazatele. Například průběžná výrobní doba je ovlivněna přípravným časem, zkušebním časem, jednicovým časem, velikostí výrobní dávky atd. Všechny tyto vlivy se musí odrazit v ukazateli. Váha časového

období zvyšuje využitelnost a vypovídající schopnost výsledků ABC analýzy. Zahrnutím delšího časového období do analýzy se docílí zpřesnění výsledků i zohlednění periodických vlivů. Dle úrovně a přesnosti plánování v dané společnosti může být součástí analýzy vedle aktuálního či minulého i budoucí období. ABC analýza není tedy vázána pouze na již proběhlou skutečnost. Přiřazením váhy lze pak zdůraznit pro analýzu význam vybraného období.

3.1.7 Zásoby podniku

Zásoby a náklady s nimi spojené představují pro každý podnik nezanedbatelnou investici. Management se proto snaží efektně a efektivně stav zásob řídit, aniž by přitom došlo ke snížení úrovně zákaznického servisu. Důvodů proč podnik udržuje zásoby je několik. Jedním z nich je, že vyrovnávají poptávku s nabídkou, umožňují specializaci ve výrobě a poskytují ochranu před nepředvídatelnými výkyvy v poptávce a v době cyklu objednávky. Typy zásob lze klasifikovat podle jejich účelu, pro který jsou udržovány. Z tohoto pohledu se zásoby dělí na běžné (cyklické) zásoby, spekulativní zásoby, sezonní zásoby, zásoby na trase, pojistné zásoby nebo na neprodejné zásoby. Kvalitnější řízení zásob má za následek zvyšování rentability podniku. Tohoto lze dosáhnout přesnější predikcí dopadů metod strategií na stav zásob a minimalizací celkových nákladů logistických činností při neměnném uspokojování požadavků na zákaznických servis.

3.1.7.1 Pojistné zásoby

Pojistné zásoby si každý podnik udržuje z toho důvodu, aby měl jistotu, že nenastane situace nedostatku nějakého výrobku či suroviny na skladě. Jedná se o zásoby, které jsou vyšší než by podnik za normálních okolností držel. Běžně je průměrná zásoba jednotlivých skladových položek rovna polovině objednávacího množství plus pojistná zásoba, v případě že existuje proměnlivost poptávky nebo celkové doby doplnění zásob. Existují čtyři kombinace, jak udržovat zásoby:

- konstantní poptávka s konstantní celkovou dobou doplnění zásob
- variabilní poptávka s konstantní celkovou dobou doplnění zásob
- konstantní poptávka s variabilní celkovou dobou doplnění zásob
- variabilní poptávka s variabilní celkovou dobou doplnění zásob

Je zřejmé, že variabilita objednávky má podstatný vliv na velikost pojistných zásob. Ovšem jejich udržování je velice nákladné, a proto se management podniku snaží tuto variabilitu minimalizovat, popř. v ideálním případě odstranit. Pokud by se podniku podařilo přesněji poptávku odhadovat, vedlo by to k nižším pojistným zásobám. Tato metoda se nazývá prognózování. Pro snížení variability je vhodné, aby se podnik zaměřil na spolehlivého dodavatele, který poskytuje objednávky včas. Tato koncepce je nazývaná dodávky s pevnou dodací lhůtou, jejichž cílem je především dodávat zboží co nejspolehlivěji i na úkor rychlosti dodání.[2]

3.1.7.2 Výpočet pojistných zásob

Existují dva způsoby, jak vypočítat výši pojistných zásob. Buď je lze vypočítat pomocí počítačové simulace, nebo ji lze spočítat pomocí statistických metod. V případě statistických metod se vychází z předpokladu, že na výši pojistných zásob působí dva faktory. Variabilita poptávky a variabilita cyklu doplnění zásob. Nutnou podmínkou ovšem je mít k dispozici statisticky významný vzorek dat. V případě splnění této podmínky lze výši pojistné zásoby pro konstantní úroveň zákaznického servisu spočítat podle tohoto vzorce:

$$\sigma_c = \sqrt{R(\sigma S^2) + S^2(\sigma R^2)} \quad (1.3)$$

kde

σ_c ... jednotky pojistné potřebné pro uspokojení 68% všech pravděpodobností

R^2 ... průměrný cyklus doplnění zásob

$$\sigma_R = \sqrt{\frac{\sum_{i=1}^n f_i d^2}{n-1}} \quad \text{směrodatná odchylka cyklu doplnění zásob}$$

f_i ... četnost případu

d ... odchylka případu od střední hodnoty

n ... celkový počet pozorování

S^2 ... průměrná denní spotřeba

$$\sigma_S = \sqrt{\frac{\sum_{i=1}^n f_i d^2}{n-1}} \dots \text{směrodatná odchylka denní spotřeby}$$

Je možné počítat pojistné zásoby i pro různé úrovně zákaznického servisu. Jedná se ovšem o natolik komplikované výpočty, které jsou mimo rámec této diplomové práce. [11]

3.2 Část umělé inteligence

3.2.1 Úvod do umělé inteligence

Umělá inteligence je obor informatiky zabývající se tvorbou strojů připomínajících příznaky inteligentního chování. Definice pojmu „inteligentní chování“ je stále předmětem diskuse, nejčastěji se jako vzor inteligence užívá lidský rozum. Pokusy vytvořit obecnou umělou inteligenci, která by byla srovnatelná s lidskou, se ukázaly být nesmírně obtížné, přesto se podařilo vědcům během posledních padesáti let vyvinout sadu postupů, které dosahují dílčích úspěchů v řešení jednotlivých problémů.

Umělá inteligence jako vědní disciplína se postupně formuje v posledních třiceti letech z několika vědních disciplín, jakými jsou např. psychologie, neurologie, kybernetika, matematická logika, teorie rozhodování, informatika, teorie her, lingvistika atd. Její vývoj stále není ukončen. V současnosti se již umělá inteligence považuje za samostatnou vědní disciplínu, přestože má mezi ostatními vědními disciplínami poněkud specifické postavení. Prvním důvodem je neexistence všeobecně přijímané definice. Druhý důvodem je to, že umělá inteligence neposkytuje jednotící teorie, metody a techniky, které lze úspěšně používat k počítačovému řešení některých složitých úloh rozhodování, plánování, diagnostiky apod.

Umělé neuronové sítě v umělé inteligenci vycházejí z chování odpovídajících biologických struktur, kde nejvýznamnějším funkčním blokem je neuron. Neurony lze studovat z celé řady různých hledisek, od jejich fyziologických projevů až po jejich různé modely, které jsou základem pro vytváření umělých neuronových sítí. Modely lze rozdělit do dvou základních skupin. Modely s nespojitými přenosovými funkcemi, které jsou starší a bývají označovány jako modely tzv. první generace, a modely se spojitými přenosovými

funkcemi nesoucí označení modely druhé generace. Praktické použití neuronových sítí je např. pro rozpoznávání a kompresi obrazů nebo zvuků, předvídání vývoje časových řad (např. burzovních indexů) nebo k filtrování spamu. Lze je také využít v lékařství, kde slouží k prohlubování znalostí o fungování nervových soustav živých organismů.[1]

3.2.2 Obecná charakteristika neuronové sítě

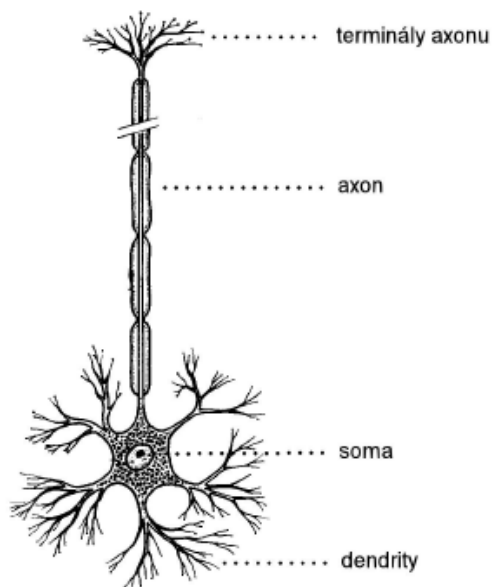
Metod, které mohly být použity pro klasifikaci dat, existuje celá řada. Vybraná metoda umělé inteligence, která byla v práci zvolena, je použití neuronové sítě s učícím se algoritmem zpětného šíření chyby učení. Její hlavní výhodou je to, že je schopna klasifikovat na základě většího počtu vstupních atributů. Aplikace neuronové sítě má ale nevýhodu v tom, že učicí proces je velmi zdoluhavý a je velice náročné neuronovou síť na začátku nastavit tak, aby vstupní data správně klasifikovala. Existuje i jisté riziko, že neuronová síť nebude schopná se v závislosti na vstupních trénovacích datech správně „naučit“ a rozpoznávat tyto závislosti. Tyto dva nedostatky lze eliminovat vhodným nastavením parametrů pro použitou neuronovou síť a dlouhodobějším testováním pro jednotlivá nastavení jejích parametrů. Dále v textu bude uváděn pouze termín neuronová síť, který nahrazuje termín metoda umělé inteligence.

Existují dva pohledy na podstatu neuronové sítě. První pohled vychází z konekcionismu, který je inspirován výzkumem a poznáváním procesů v živých organismech. Konekcionismus modeluje poznávání jako dynamický vzorec vzájemných interakcí ve složité síti složené z relativně jednoduchých prvků a jejich vztahů. Prvky této sítě lze chápat jako symboly, kterým ale již nelze přiřadit konkrétní pojmy reprezentující objekty. Druhý přístup k umělé inteligenci je tzv. symbolická umělá inteligence, která zdůrazňuje význam pravidel pro manipulaci se symboly různého druhu. Obecně lze tento přístup definovat jako problém reprezentace znalostí s inteligentním prohledáváním stavového prostoru.

3.2.3 Neuronová buňka

Základním stavebním prvkem každé neuronové sítě je nervová buňka, tzn. biologický neuron, a lidský mozek jich má přibližně 13 až 15 milionů, z nichž každý může být spojen s 5000 jinými neurony. Neurony jsou samostatné specializované buňky primárně určené k přenosu informací, zpracování a uchování informací, které jsou nutné pro realizaci životních funkcí organismu.

Grafické schéma biologického neuronu je na obrázku 1.



Obrázek 1: Biologický neuron

Zdroj: [13]

Z neurofyzilogického hlediska můžeme neuronovou buňku rozdělit do čtyř hlavních částí – synapse, soma, dentrit a axon. Dentrit a axon mají přizpůsobenou funkci pro oboustranné vedení vzruchů buňky. V technologickém pojetí tyto dvě části neuronu lze chápat jako vedení. Vlastnosti těchto částí jsou specifikovány především jako přenosové parametry, tj. vstupní a výstupní impedancí nebo propustností. Zbývající dvě části neuronové buňky z hlediska matematického modelu realizují operace konfluence⁵, agregace, prahování a nelineárního zobrazení. Synapse je něco na způsob paměťových elementů, které se učí na základě přichozích informací z svého okolí. Ve skutečnosti jsou ale velmi složitým útvarem, jejichž vlastnosti jsou velice aproximovány. V případě použití v neuronových sítích vycházíme z předpokladu, že synapse mají především paměťovou funkci a že spojitě adaptují své hodnoty vyjadřující průchodnost pro přicházející vzruchy, což způsobuje ukládání nabytých zkušeností. Současně s tím předpokládáme, že tato hodnota může být v uvažovaném modelu vyjádřena pouze reálným číslem w_{ij} . Matematicky tedy synapse v tomto modelu představují operaci konfluence aktuálních vstupů do neuronu a získaných zkušeností reprezentovaných hodnotami vah. Konfluence

⁵ V matematické terminologii se jedná o skalární součin.

znázorňuje vektor synaptických vah uložených zkušeností do neuronu, které se ovšem mohou i nadále měnit v závislosti při aplikovaném procesu učení. Operace konfluence poté kombinací vstupního vektoru $x_{(t)}$ a vektoru synaptických vah $w_{(t)}$ přiřazuje ke každé složce vstupního vektoru určitou váhu, která odpovídá uložené zkušenosti. Výstupem ze synapse jsou poté signály, které dále postupují přes dendrity jiného neuronu až do jeho těla. Soma, která prostřednictvím dendritů vstupní signály přijímá, je poté slučuje pomocí operace agregace. Je-li výsledná hodnota agregace větší než daný práh neuronu, pak se tento neuron aktivuje, tj. na jeho výstupu se objeví signál odpovídající nelineární funkci výsledné agregační hodnoty. Celkově tedy soma vykonává tři matematické operace – agregaci, prahování a nelineární zobrazení. Neuron proto můžeme považovat za matematický preprocesor, který má na vstupu n -rozměrný vektor vstupních signálů. Ten je reprezentován signály přicházející z celkového počtu n sousedních neuronů. Na výstupu má neuron pouze jeden výstupní signál. [13]

3.2.4 Matematický model neuronové sítě

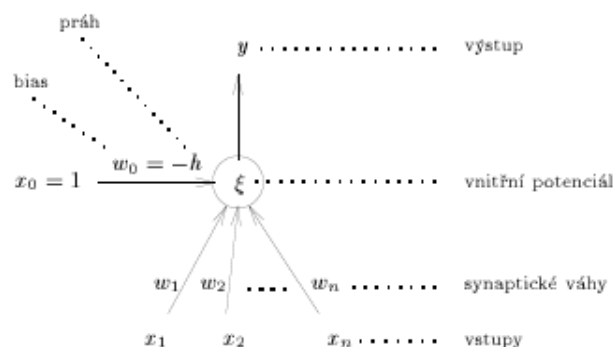
Z předchozí kapitoly, kde byla popsána stavba a funkce neurofyzilogických neuronů, je zřejmé, že popsat neuron matematickým modelem je velice komplikované. V současnosti totiž nejsme schopni vytvořit takové modely neuronů a neuronových sítí, které by nám nějakým způsobem pomohly pochopit fungování lidského mozku. Přesto lze vytvářet modely neuronových sítí, kterými můžeme nasimulovat chování některých výpočetních zařízení, která jsou schopna pomoci při řešení některých úloh jako jsou např. rozpoznávání písma nebo obrazu.

Celá tato kapitola včetně podkapitol byla převážně volně citována z [3]

3.2.4.1 Formální neuron

Základním stavebním prvkem matematického modelu neuronové sítě je formální neuron, který je transformací neurofyzilogického neuronu do matematické terminologie. Někdy bývá také z historických důvodů označován jako perceptron.

Schéma formálního neuronu je na obrázku 2.



Obrázek 2: Formální neuron

Zdroj: [3]

Formální neuron má obecně n reálných vstupů $x_1 \dots x_n$, které jsou modelovány dendrity. Vstupy jsou většinou reprezentovány buď jako podmínky z vnějšího okolí, nebo se jedná o výstupy jiných neuronů. Každý vstup má definovanou určitou synaptickou váhu $w_1 \dots w_n$, kde w je také reálné číslo. Synaptické váhy určují propustnost formálního neuronu. Každý formální neuron má svůj vnitřní potenciál, který se spočítá jako skalární součin vstupních hodnot a synaptických vah.

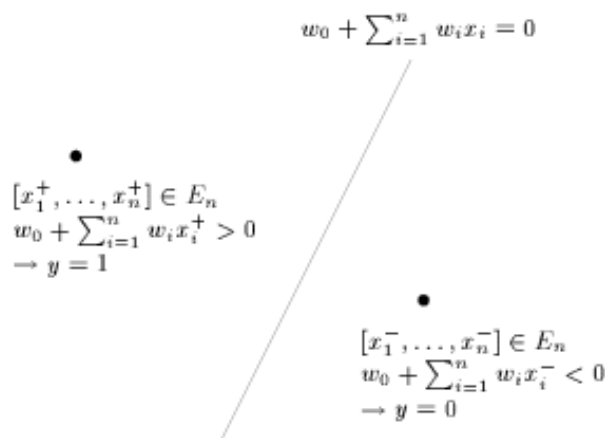
$$\xi = \sum_{i=1}^n w_i x_i \quad (2.1)$$

Hodnota vnitřního potenciálu ξ (2.1) po dosažení tzv. prahové hodnoty h indikuje výstup (stav)

$$y_t = s(x) \quad (2.2)$$

neuronu y_t (2.2), který modeluje elektrický impuls axonu. Nelineární nárůst výstupní hodnoty $y = (\xi)$ při dosažení prahové hodnoty potenciálu je dán tzv. aktivační přenosovou funkcí σ . Nejjednodušším typem aktivační funkce je tzv. ostrá nelinearita, která nabývá hodnot 0 a 1 v závislosti na polaritě vnitřního potenciálu ξ . Pro $\xi > h$ nabývá hodnoty 0 a pro $\xi < h$ nabývá hodnoty 1. Matematickými úpravami lze dosáhnout toho, aby funkce měla nulový práh. V případě, že je práh záporný, se jedná o tzv. bias, tj. synaptická váha w_0 nabývá hodnoty $-h$ dalšího formálního vstupu $x_0 = 1$ a konstantní jednotkovou hodnotou.

Graficky lze znázornit výstup jednoho neuronu na obrázku 3.



Obrázek 3: Geometrická interpretace funkce neuronu

Zdroj: [3]

Jednotlivé vstupy se berou jako souřadnice bodu v n -rozměrném euklidovském prostoru, který je brán jako vstupní. V tomto prostoru má rovnice nadroviny tvar

$w_0 + \sum_{i=1}^n w_i x_i = 0$ a současně rozděluje tento vstupní prostor na dva poloprostory.

Množina bodů, které splňují nerovnost $\sum_{i=1}^n w_i x_i > 0$ leží v jednom poloprostoru

a množina bodů splňující nerovnost $\sum_{i=1}^n w_i x_i < 0$ se nachází ve druhém poloprostoru.

Obě množiny většinou bývají disjunktní. Synaptické váhy neuronu $w_0 \dots w_n$, které také obsahují bias, lze chápat jako koeficienty této nadroviny. Současně nám neuron v tomto euklidovském prostoru klasifikuje, ve kterém ze dvou poloprostorů definovaných touto nadrovinou leží bod, který odpovídá vstupním souřadnicím. Neuron v takovém případě realizuje tzv. dichotomii vstupního prostoru, tj. je právě aktivní ($y = 1$), když souřadnice vstupních bodů leží v prvním poloprostoru. Lze to také interpretovat tak, že vstupy

neuronu splňují podmínku $\sum_{i=1}^n w_i x_i > 0$. Naopak v případě, že souřadnice vstupních

bodů leží ve druhém poloprostoru, tj. vstupy neuronu vyhovují podmínce $\sum_{i=1}^n w_i x_i < 0$, je neuron pasivní $y = 0$.

Pomocí jednoduchého příkladu na rozpoznání znaků lze osvětlit fungování neuronu. Mějme jeden formální neuron, který se použije na rozpoznání dvou znaků – X a Y. Obě písmena lze zakreslit do matice bodů, kterou lze popsat reálnými čísly. Proto oba znaky lze jednoznačně definovat n -rozměrným vektorem, který je současně vstupem pro neuron. Pro oba znaky si určíme, jakému reálnému číslu budou odpovídat. Je třeba, aby obě čísla byla od sebe relativně vzdálena. Tím zajistíme přesnou identifikaci obou znaků. Při procesu učení se na vstup přivádějí velmi podobná reálná čísla, která všechna odpovídají v našem příkladu např. písmenu X. Další fází učení se na vstup neuronu přivádějí v úzkém intervalu čísla odpovídající druhému písmenu Y. Tím dostaneme dvě vzájemně disjunktní množiny čísel odpovídající oběma znakům. Funkcí uvedeného neuronu lze obě tyto množiny oddělit příslušnou nadrovinou (obrázek 3). Po správném nastavení synaptických vah, které jednoznačně oddělí obě množiny, dokáže neuron na vstupu rozpoznat neznámé reálné číslo a určit, jakému znaku odpovídá. Na začátku učení je zřejmé, že neuron nedokáže vstupní čísla rozpoznat, protože synaptické váhy jsou nastaveny náhodně. Ale během procesu učení se váhy postupně adaptují a obě množiny se začnou oddělovat. Problém ale nastane v případě, kdy na vstup přivedeme reálné číslo, které není jednoznačně identifikovatelné a neuron ho zařadí do nesprávné množiny. Poté je nutná korekce učitele, aby upravil synaptické váhy tak, aby se nadrovina natočila a zahrнула nové číslo do správného poloprostoru. Z uvedeného příkladu je zřejmé, že jedním formálním neuronem lze řešit pouze velmi jednoduché příklady. Klasickou úlohou, kterou jím nelze řešit, je vylučovací disjunktní funkce XOR.

3.2.4.2 Neuronová síť

Pod pojmem neuronová síť si lze představit několik vzájemně propojených neuronů. Jednotlivé výstupy každého neuronu jsou současně vstupy dalšího, stejně tak jako u neurofyziologických neuronů, kde terminály axonu jsou přes synaptické váhy spojeny s dendrity jiných neuronů. Schéma propojení všech neuronů se nazývá topologie neuronové sítě. V neuronové síti rozlišujeme několik druhů neuronů. Jsou to neurony vstupní, pracovní (skryté) a výstupní. Jednotlivé dráhy mezi neurony se v matematickém modelu nazývají cestami. Změny stavů neuronů jsou zapříčiněny šířením informací

po těchto cestách. Stavby všech neuronů v neuronové síti určují tzv. stav neuronové sítě a synaptické váhy všech spojů představují tzv. konfiguraci neuronové sítě.

Topologie neuronové sítě, adaptující se váhy, počet neuronů v síti a jejich vzájemné propojení se neustále mění. Z tohoto důvodu se v matematickém modelu celková dynamika neuronových sítí rozděluje do tří dynamik nebo tzv. režimů práce sítě. První je dynamika organizační, při které dochází ke změně topologie. Další je dynamika aktivní, během níž se mění stav sítě a poslední dynamikou je dynamika adaptivní, při které se konfiguruje synaptické váhy. Toto dělení lze pouze u matematického modelu formálního neuronu, a nikoliv u neurofyzilogického neuronu, protože u něj dochází ke změnám v jednom časovém okamžiku. Jednotlivé dynamiky je možné definovat pomocí jejich počátečních stavů a příslušných matematických rovnic, které co nejpřesněji odpovídají vývojovým charakteristikám sítě v čase. Různé modely neuronových sítí, které jsou vhodné pro řešení určitých typů úloh, lze konkretizovat pomocí jednotlivých dynamik. To znamená, že pro specifikaci konkrétních modelů sítě postačí, když se definují všechny uvedené dynamiky – organizační, adaptivní a aktivní dynamika.

3.2.4.2.1 Organizační dynamika

Pomocí organizační dynamiky se charakterizuje topologie sítě a její případné změny. Organizační dynamika předpokládá pevnou strukturu neuronové sítě v adaptivním režimu, ale přesto dochází k její změně a to tím způsobem, že se změní počet neuronů a počet spojů mezi jednotlivými neurony. Existují dva základní typy architektury – cyklická a acyklická. U cyklické topologie existuje určitý počet neuronů, které svým spojením vytváří kruh popř. cyklus (od toho odvozený název cyklická), kde výstup prvního neuronu je současně vstupem druhého atd., až výstup posledního neuronu je vstupem prvního. Nejtriviálnějším případem cyklické architektury je zpětná vazba, kdy výstup prvního neuronu je zároveň vstup pro neuron druhý a naopak. Existuje i tzv. úplná topologie cyklické neuronové sítě, kde vstup jednotlivých neuronů je výstup dalšího neuronu. Taková topologie obsahuje nejvíce možných cyklů.

Druhým případem organizační dynamiky je topologie acyklická. V této struktuře neexistují žádné cykly. Neurony jsou disjunktne rozděleny do jednotlivých vrstev, které jsou hierarchicky uspořádány od nejnižší vrstvy po nejvyšší. Spoje mezi neurony nemusejí být pouze mezi dvěma sousedícími vrstvami, ale lze spojovat neurony přes více vrstev.

Speciálním typem acyklické sítě je tzv. vícevrstvá neuronová síť. Tato síť obsahuje několik vrstev, z nichž nultá vrstva je současně i vstupní vrstvou a poslední (nejvyšší) vrstva je vrstvou výstupní. Ostatní vrstvy se skládají ze skrytých neuronů, proto se tyto vrstvy označují jako skryté. Ve vícevrstvých sítích jsou neurony jedné vrstvy spojeny se všemi neurony vrstvy, která následuje bezprostředně po ní. Proto topologii této sítě lze definovat pouze počtem neuronů v jednotlivých vrstvách. Obvykle se taková neuronová síť zapisuje ve formátu $3 - 3 - 2$, kde uvedená čísla reprezentují počty neuronů v jednotlivých vrstvách vzestupně od nulté vrstvy.

3.2.4.2.2 Adaptivní dynamika

Adaptivní dynamika se používá pro vlastní učení neuronové sítě pomocí výpočtu funkce sítě pro konkrétní vstup. Cílem učícího procesu je to, aby konfigurace sítě ve váhovém prostoru realizovala předepsanou funkci. Stejně jako u aktivní dynamiky se v adaptivní dynamice uvažuje spojitý matematický model se spojitým vývojem konfigurace neuronové sítě v čase. Na začátku se také definuje počáteční konfigurace sítě a způsob, jakým se budou v čase měnit synaptické váhy, které jsou nejprve nastaveny na náhodné hodnoty. Tyto hodnoty se pak během učícího procesu mění, a tím vlastně dochází k vlastní adaptaci, tj. učení neuronové sítě. Učení představuje netriviální optimalizační problém, jehož řešení je velmi složité a zdlouhavé i pro relativně jednoduché úlohy. Samotné učení lze rozdělit do dvou hlavních skupin. Na učení s učitelem a na učení bez učitele. V prvním případě je funkce sítě definována tzv. tréninkovou množinou. Tu si lze představit tak, že každý vstup má přesně odpovídající výstup (tréninkový vzor). Na vstup se postupně přivádějí vstupní hodnoty a podle jejich odpovídajících výstupů se nastavují synaptické váhy. Učitel dané neuronové sítě potom hodnotí správnost skutečného výstupu, která je zadána místo požadované hodnoty. Tento proces se nazývá klasifikované učení a typickým zástupcem je algoritmus back-propagation. Druhým typem adaptace je tzv. samoorganizace neboli učení bez učitele, kde tréninková data obsahují pouze vstupní hodnoty. Neuronová síť si během této adaptace sama organizuje tréninkové vzory a poznává jejich společné vlastnosti.

3.2.4.3 Použití neuronové sítě

Neuronové síť lze použít v širokém spektru činností. Většinou se používají v oblastech, kde je potřeba predikovat, klasifikovat nebo řídit.

3.2.4.3.1 Predikce

Neuronová síť je schopna předpovídat hodnoty nějaké posloupnosti a to na základě dostatečného počtu dat, která byla nasbírána někdy v minulosti. Například se může jednat o předpověď teploty, zátěže energických zdrojů, poptávce nebo nabídky na trhu nebo předpověď vývoje na burze.

3.2.4.3.2 Klasifikace

Pomocí neuronových sítí je možné klasifikovat vstupní data do dvou a více tříd. V některých případech dokonce takto klasifikovaná data jsou přesněji rozdělena do požadovaných skupin než použitím jiných metod. „Základem klasifikace pomocí neuronových sítí je, že dolní vrstvy provedou takovou kombinaci lineárních a nelineárních transformací vstupů, že ve výstupní vrstvě se mohou uplatnit separující nadroviny i u množin, které samy o sobě nejsou lineárně separabilní. Chceme-li klasifikovat do více tříd, použijeme více výstupních neuronů nebo rozložení intervalu výstupních hodnot do dostatečně vzdálených tříd. Nutnou podmínkou úspěšnosti je, aby typické vstupy navrhoval a třídění odpovědí na ně prováděl znalec v daném oboru.“⁶

3.2.4.3.3 Řízení

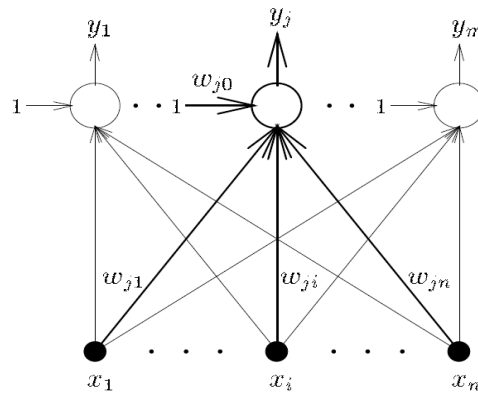
Použití neuronových sítí je také v nahrazení člověka, který na základě svých dlouholetých zkušeností a neuvědomované intuice vykonává nějakou činnost. Většinou se jedná o nějaký druh řízení pomocí řídicích povelů. Stejně tak jsou neuronové sítě používány k modelování senzomotorických reakcí robota nebo ke kontrole jakosti výrobků.[1]

3.2.5 Modely neuronových sítí

3.2.5.1 Síť perceptronů

Síť perceptronů je první úspěšně aplikovaný model neuronové sítě. Organizační dynamika definuje pevnou topologii, která je znázorněna na obrázku 4.

6 Citováno z [1]



Obrázek 4: Architektura sítě perceptronů

Zdroj: [3]

Z obrázku 4 je patrné, že se jedná o jednovrstvou síť typu $n - m$, kde n je počet vstupních neuronů $x_1 \dots x_n$ a kde m je počet výstupních neuronů $y_1 \dots y_m$. Čáry mezi vstupními a výstupními neurony znázorňují spoje neboli synaptické váhy $w_1 \dots w_n$ a w_{j0} je bias, tj. práh s opačným znaménkem j -tého výstupu, který odpovídá jednotkovému vstupu $x = 0$. Adaptivní dynamika zajišťuje v modelu vypočtení funkce sítě, která probíhá následujícím způsobem. Reálné stavy neuronů na vstupní vrstvě se nastaví na vstup sítě, tím je pak možné spočítat binární výstup sítě, který je určen stejně jako u formálního neuronu. Nejprve tedy dojde k vypočtení vnitřního potenciálu sítě, který se spočítá jako skalární součin vstupních hodnot a synaptických vah (2.3)

$$\xi_j = \sum_{i=1}^n w_{ji} x_i \quad (2.3)$$

Vzorec obsahuje koeficienty w_{ji} , které tvoří konfiguraci sítě. Vypočítaný vnitřní potenciál spolu s aktivační funkcí (nyní jsme použili ostrou nelinearitu) poté určuje výstupní funkci y .

$$y_j = \sigma(\xi_j) \quad \text{kde } \sigma(\xi) = 1 \text{ pro } \xi \geq 0 \text{ a } \sigma(\xi) = 0 \text{ pro } \xi < 0 \quad (2.4)$$

„V adaptivním režimu je požadovaná funkce sítě perceptronů zadána tréninkovou množinou

$$\tau = (x_k, d_k), \quad (2.5)$$

kde x_k je reálný vstup k -tého tréninkového vzoru a d_k je odpovídající požadovaný binární výstup (daný učitelem). Cílem adaptace je, aby síť pro každý vstup x_k ($k = 1, \dots, p$)

z tréninkové množiny odpovídala v aktivním režimu požadovaným výstupem d_k tj, aby platilo

$$y(w, wk) = dk \quad k = 1, \dots, p \quad (2.6)$$

Je zřejmé, že tuto podmínku není možné splnit vždycky, protože zdaleka ne všechny příklady lze řešit pomocí sítě perceptronů. V takovýchto případech je vhodné, aby se síť naučila co nejvíce tréninkových vzorů, ovšem s omezující podmínkou, že síť se naučí všechny tréninkové vzory na sto procent, protože ne vždycky musí být správně určené.⁷

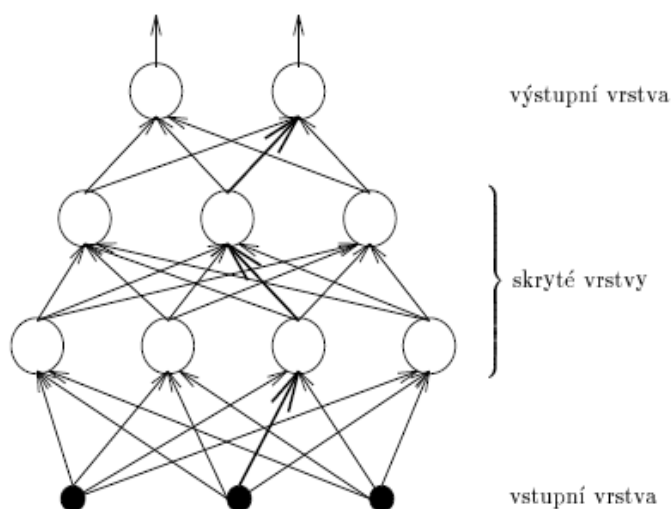
3.2.5.2 Vícevrstvá neuronová síť a back-propagation

V převážné části řešených úloh je nutno zpracovávat i neúplná nebo nepřesně zadaná data, proto je potřeba chybějící hodnoty predikovat. Tento postup ovšem není tak úplně jednoduchý, obzvláště chybí-li dostupné podklady pro rekonstrukci neexistujících dat. V těchto příkladech se hojně používá vícevrstvá neuronová síť. Existuje mnoho modelů těchto neuronových sítí, ale nejznámější je vícevrstvá neuronová síť s učícím se algoritmem zpětného šíření. Tento model se v praxi používá k řešení v přibližně 80% úloh, které se řeší pomocí neuronových sítí a vychází ze sítě perceptronů, která byla popsána v předchozí kapitole. Model má také mnoho modifikací, které se přizpůsobují podle jednotlivých řešených úloh.

3.2.5.2.1 Organizační dynamika

Model vícevrstvé neuronové sítě má obvykle pevnou topologii a standartně se používá dvouvrstvá nebo třívrstvá síť.

7 Citováno z [3]



Obrázek 5: Vícevrstvá neuronová síť typu 3-4-3-2

Zdroj: [3]

Ve spodní části jsou vstupní neurony, v prostřední části se nacházejí skryté neurony, jejichž význam není přesně definován, a v horní části jsou výstupní neurony. Z obrázku je také patrné, že se jedná o acyklický model. Množina vstupních neuronů se označuje jako $x_1 \dots x_n$ a množina výstupních neuronů jako $y_1 \dots y_n$. Neurony jsou značeny indexy i, j atd. se svým vnitřním potenciálem ξ . Stejně jako u sítě perceptronů jsou spojení mezi neurony, které jsou znázorněny jako čáry, přiřazeny synaptické váhy w_{ji} , které jsou reprezentovány reálným číslem. Synaptická váha w_{j0} by představovala bias.

3.2.5.2.2 Aktivní dynamika

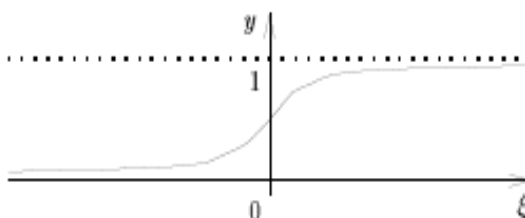
Vlastní výpočet vícevrstvé neuronové sítě, tak jako u všech typů neuronových sítí, probíhá v aktivní dynamice. Konfigurací synaptických vah w se nastavuje vstupní funkce $y(w)$ z definičního oboru reálných čísel, které na výstupu odpovídá uzavřený interval od 0 do 1. Matematicky lze toto popsat $y(w) = R^n \rightarrow (0,1)^m$. Výpočet je prováděn od času $t = 0$. V něm se nastaví vstup sítě na odpovídající stavy vstupních neuronů y_i , kde i se nachází v množině X . U zbylých neuronů se ponechá neurčitý stav. V čase $t > 0$ jsou vypočteny hodnoty vnitřních potenciálů u jednotlivých neuronů, u kterých byl již nastaven určitý stav.

$$\xi_j = \sum_{i \in j} w_{ji} y_i \quad (2.7)$$

Obecně lze říci, že v čase t se aktualizují stavy všech neuronů na t -té vrstvě. Z vypočtených vnitřních potenciálů se určí pomocí sigmoidální funkce reálný stav výstupních neuronů.

$$\sigma(\xi) = \frac{1}{1 + e^{-\lambda_j \xi}} \quad (2.8)$$

Z této přenosové funkce plyne diferencovatelnost funkce sítě, která je podstatná pro učící algoritmus back-propagation. Parametr λ , který je prezentován reálným číslem, znázorňuje nelineární nárůst standardní sigmoidy v okolí nuly pro kladná λ a její nelineární pokles pro záporná λ . Standardní sigmoida také definuje tzv. míru rozhodnosti neuronu a její graf je vyobrazen na obrázku 6.



Obrázek 6: Standardní logistická sigmoida

Zdroj: [3]

Obecně se v aplikovaných matematických modelech většinou nastavuje parametr strmosti na hodnotu 1, ale obecně lze tento parametr nastavit pro každý nevstupní neuron na libovolnou hodnotu. Poté je stav neuronu vypočten aktivační funkcí (2.9).

$$y_j = \sigma_j(\xi_j), \text{ kde } \sigma_j(\xi) = \frac{1}{1 + e^{-\lambda_j \xi}} \quad (2.9)$$

Z důvodu toho, že se jedná o acyklickou neuronovou síť, jsou tímto postupem nastaveny hodnoty všech neuronů v takové síti. Aktivní režim je ukončen, jakmile jsou vypočteny všechny stavy neuronů v jednotlivých vrstvách neuronové sítě. Především je důležité, aby byly nastaveny hodnoty výstupních neuronů, které určují výstup neuronové sítě pro daný vstup.

3.2.5.2.3 Adaptivní dynamika

Adaptivní režim vícevrstvé neuronové sítě je totožný s adaptivním režimem u sítě perceptronů. Tréninková množina definuje požadovanou funkci a chyba sítě $E(w)$ je

vzhledem k této funkci definována jako součet parciálních derivací chyb sítě $E_k(w)$ v závislosti na tréninkových vzorech a synaptických vahách w sloužící ke konfiguraci sítě.

$$E(w) = \sum_{k=1}^p E_k(w) \quad (2.10)$$

Parciální chyba $E_k(w)$ sítě vzhledem ke k -tému tréninkovému vzoru je úměrná součtu mocnin odchylek skutečných hodnot výstupů sítě pro vstup k -tého tréninkového vzoru od odpovídajících požadovaných hodnot výstupů u tohoto vzoru:

$$E_k(w) = \frac{1}{2} \sum_{j \in Y} (y_j(w, x_k) - d_{kj})^2 \quad (2.11)$$

Smyslem adaptační dynamiky je snaha minimalizovat chyby sítě ve váhovém prostoru. Toto ovšem bývá obtížné z důvodu toho, že chyba sítě závisí na komplikované nelineární složené funkci vícevrstvé neuronové sítě, což představuje netriviální optimalizační problém, který lze jednoduše řešit pomocí gradientní metody vyžadující diferencovatelnost chybové funkce. Stejně jako u aktivní dynamiky, tak i v této dynamice začíná celý adaptační postup v čase $t = 0$. V něm jsou nastaveny synaptické váhy na hodnoty z intervalu $< -1, 1 >$. Tento interval není pevně stanovený a lze jej proto měnit. Samotná adaptace probíhá v jednotlivých krocích, které odpovídají tréninkovým cyklům. Nová konfigurace v čase $t > 0$ je vypočtena ze vzorce (2.12)

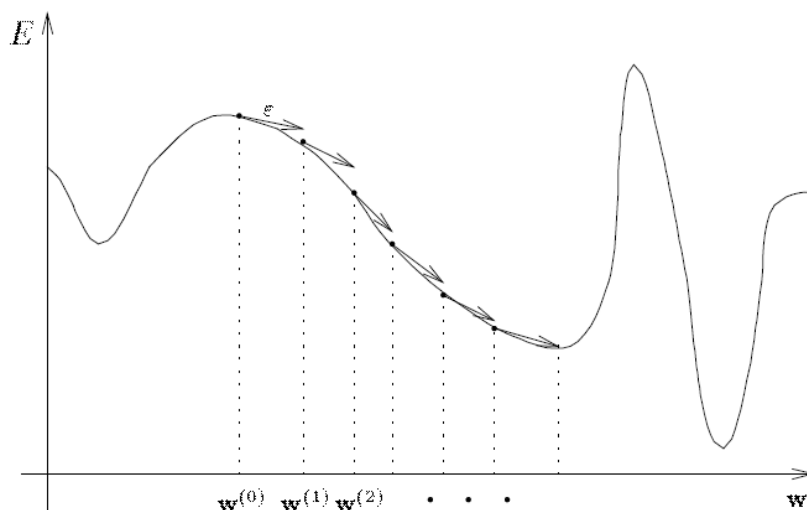
$$w_{ji}^{(t)} = w_{ji}^{(t-1)} + \Delta w_{ji}^{(t)}, \quad (2.12)$$

kde Δw představuje změnu synaptických vah v čase $t > 0$ a tato změna odpovídá hodnotě záporného gradientu chybové funkce $E(w)$ v bodě $w_{(t-1)}$

$$\Delta w_{ji}^{(t)} = -\varepsilon \frac{\delta E}{\delta w_{ji}}(w^{(t-1)}), \quad (2.13)$$

kde proměnná ε představuje rychlost učení. Tento parametr může ovlivňovat vývoj chybové funkce, proto je důležité jej nastavit na vhodnou hodnotou. Při velmi nízkých hodnotách metoda zpětného šíření chyby konverguje ke svému minimu velice pomalu a při velmi vysokých hodnotách chyba diverguje. „Ladění parametru ε připomíná ladění správné frekvence televizního signálu. Pro větší topologie je lepší začínat s menším ε (např. řádově tisíciný nebo deseti tisíciný). Při úspěšné konvergenci lze ε nepatrně zvýšit, při následné diverzifikaci nebo oscilaci hodnotu ε zase snížíme. I při velkém vzrůstu chyby po předešlé

konvergenci bez dlouhodobější změny ε pokračujeme v adaptaci se stejným ε , protože metoda v některých případech přechází ve váhovém prostoru do oblasti lepší konvergence, což se potvrdí, nebo vyvrátí až v dalších tréninkových cyklech.“⁸



Obrázek 7: Gradientní metoda

Zdroj: [3]

Princip fungování gradientní metody vychází z obrázku 7, na němž je vyobrazena chybová funkce $E(w)$, která určuje chybu sítě vzhledem k pevné tréninkové množině v závislosti na její konfiguraci. Konfigurace je současně odrazem mnohorozměrného vektoru vah w a odpovídá hodnotám na ose x . V adaptivním režimu hledáme takovou konfiguraci, pro kterou nabývá chybová funkce svého minima. V ideálním případě se jedná o minimum globální, ale někdy gradientní metoda konverguje pouze do minima lokálního. Do jakého minima gradientní metoda konverguje, závisí na dané síti, resp. na její „inteligenci“. Čím vyšší inteligenci daná síť má, tím je pravděpodobnější, že bude konvergovat do hlubšího minima a naopak. Na začátku celého procesu se nastaví synaptické váhy na náhodné hodnoty $w^{(0)}$. To má za následek, že chyba sítě požadované funkce bude s velkou pravděpodobností nabývat vysokých hodnot. Poté sestrojíme

v bodě $w^{(0)}$ k chybové funkci tečnu $\frac{\delta E}{\delta w}(w^{(0)})$ a posunem o ε po této tečně směrem dolů dostaneme novou konfiguraci neuronové sítě $w^{(1)}$, tak že $w^{(1)} = w^{(0)} + \Delta w^{(1)}$. Hodnota chybové funkce odpovídající synaptickým vahám $w^{(0)}$ je větší než hodnota v bodě odpovídajícím vahám $w^{(1)}$. Celý postup se analogicky opakuje až do té doby, než se

8 Citováno z [3]

dostaneme do jednoho z minim chybové funkce. Zcela zásadním je pro gradientní metodu nastavení hodnoty rychlosti učení ε . V případě, že se tento parametr nastaví na příliš vysokou hodnotu, je možné, že se do lokálního nebo globálního minima chybové funkce vůbec nedostaneme. Naopak pokud se parametr zvolí na příliš nízkou hodnotu, poté samotný proces učení trvá neúměrně dlouhou dobu. Ovšem ani vhodným nastavením rychlosti učení není zaručeno, že gradientní metoda bude v reálném čase konvergovat k nějakému lokálnímu nebo globálnímu minimu.

3.2.5.2.4 Strategie back-propagation

„K tomu, aby se mohla realizovat adaptivní dynamika, je potřeba vypočítat gradient chybové funkce ve vzorci (2.12). Nejprve užitím pravidla o derivaci součtu převedeme tento gradient na součet gradientů parciálních chybových funkcí:

$$\frac{\delta E}{\delta w_{ji}} = \sum_{k=1}^p \frac{\delta E_k}{\delta w_{ji}}. \quad (2.14)$$

Protože funkce sítě je složena z funkcí jednotlivých neuronů, pro výpočet gradientu parciální chybové funkce je vhodné použít pravidlo o derivaci složené funkce:

$$\frac{\delta E_k}{\delta w_{ji}} = \frac{\delta E_k}{\delta y_j} \cdot \frac{\delta y_j}{\xi_j} \cdot \frac{\delta \xi_j}{\delta w_{ji}} \quad (2.15)$$

Parciální derivaci $\frac{\delta \xi_j}{\delta w_{ji}}$ v rovnici (2.15) získáme derivováním vnitřního potenciálu (2.7)

$$\frac{\delta \xi_j}{\delta w_{ji}} = y_i \quad (2.16)$$

a parciální derivaci $\frac{\delta y_j}{\delta \xi_j}$ získáme derivováním aktivační funkce (2.9), jejíž derivaci lze vyjádřit pomocí funkční hodnoty následujícím způsobem

$$\frac{\delta y_j}{\delta \xi_j} = \lambda_j y_j (1 - y_j). \quad (2.17)$$

Dosazením do rovnic (2.16), (2.17) do (2.15) dostáváme:

$$\frac{\delta E_k}{\delta w_{ji}} = \frac{\delta E_k}{\delta y_j} \lambda_j y_j (1 - y_j) y_i. \quad (2.18)$$

Pro výpočet parciální derivace $\frac{\delta E_k}{\delta y_j}$ v předcházejícím vzorci se používá strategie zpětného šíření. Pokud y náleží Y , lze pak uvedenou derivaci vypočítat přímým derivováním parciální chybové funkce (2.11)

$$\frac{\delta E_k}{\delta y_j} = y_j - d_{kj}, \quad j \in Y \quad (2.19)$$

což odpovídá chybě výstupního neuronu j pro k -tý tréninkový vzor. Pro skrytý neuron j , který neleží ve sjednocené množině X a Y , se při výpočtu $\frac{\delta E_k}{\delta y_j}$ znovu uplatní pravidlo o derivaci složené funkce, kde vypočteme derivace, které lze získat přímým derivováním:

$$\frac{\delta E_k}{\delta y_j} = \sum_{r \in J} \frac{\delta E_k}{\delta y_r} \lambda_r (1 - y_r) w_{rj}, \quad j \notin X \cup Y \quad (2.20)$$

Ve přecházejícím vzorci jsme výpočet parciální derivace $\frac{\delta E_k}{\delta y_j}$ pro skrytý neuron j převedli na výpočet parciálních derivací $\frac{\delta E_k}{\delta y_r}$ u vstupních neuronů, do nichž vede spoj od neuronu j , a proto mají neuron j jako svůj vstup. Při výpočtu $\frac{\delta E_k}{\delta y_j}$ tedy lze postupovat od výstupních neuronů, pro které lze tuto parciální derivaci počítat z (2.20) postupně směrem ke vstupním tak, že u skrytých neuronů vypočteme $\frac{\delta E_k}{\delta y_j}$, pokud jsou již vypočteny parciální derivace $\frac{\delta E_k}{\delta y_r}$ pro všechny r z množiny výstupních neuronů. Korektnost tohoto postupu opět vyplývá z acykličnosti architektury sítě.^{“9}

9 Citováno z [3]

4 Praktická část

4.1 Úvod do praktické části

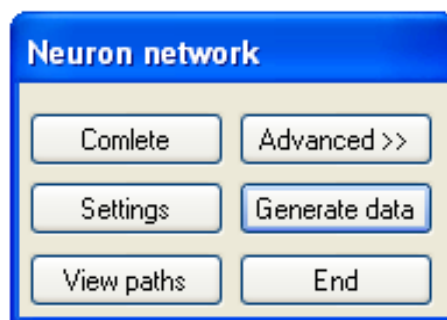
Předtím, než bylo možné začít tvořit praktickou část diplomové práce, bylo jednak nutné definovat, na jaký typ skladu budeme analýzu ABC a neuronovou síť aplikovat, a jednak také získat potřebná data. Byl vybrán sklad vstupních zásob, u kterého je potřeba hledat optimální řízení a plánování položek z hlediska plynulosti výroby a zajištění dostatečného množství vstupů pro aktuální výrobní potřebu, samozřejmě s co nejnižšími náklady na skladování těchto zásob. Další nutnou podmínkou pro úspěšnou aplikaci neuronové sítě byla potřeba získat řádově tisíce skladových položek. V úvahu připadaly dvě možnosti získání těchto dat. První je získání dat od nějakého podniku nebo společnosti. Tady se ale naskytl problém už v pouhém výběru podniků. Zdaleka ne každý podnik disponuje tak velkým sortimentem výrobků či účtovacích položek. Další překážkou pro získávání dat z praxe je ten, že firmy nechtějí zveřejňovat své interní údaje. Proto byla zvolena druhá možnost, a to potřebná data si generovat pomocí generátoru dat.

Důvodem, proč byla vybrána klasifikace pomocí neuronové sítě, je ten, že neuronová síť umožňuje klasifikovat data na základě více vstupních atributů než je tomu u metody ABC, která klasifikuje data pouze na základě jednoho vstupního atributu.

Celá praktická část diplomové práce je tvořena programem Neuron Network, který umožňuje vykonat všechny potřebné procesy. Současně také slouží k nastavování parametrů neuronové sítě a k jejímu testování.

4.2 Program Neuron Network

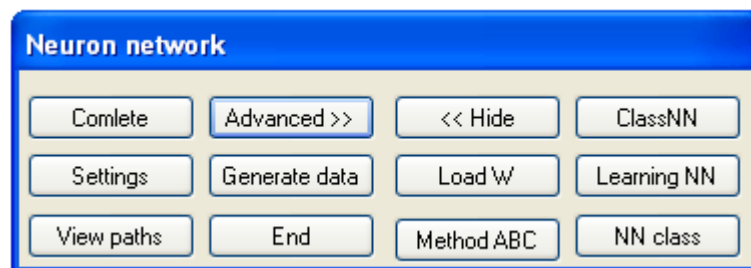
Program Neuron Network byl vytvořen v programovacím jazyce C# v prostředí Visual Studio 2008. Jedná se o aplikaci Windows (Windows Forms). Po jejím spuštění se objeví panel, který obsahuje všechna tlačítka, která umožňují ovládání programu.



Obrázek 8: Overview programu

Zdroj: Neuron Network

Program lze rozdělit na několik separátních částí, každá vykonává jednu z potřebných funkcí k úspěšnému splnění zadání. Z uživatelského pohledu lze požadovaného výstupu docílit dvěma způsoby. Buď se celý program, tj. aplikace metody ABC, procesu učení neuronové sítě a výpočet nákladů, provede najednou stisknutím tlačítka Complete, nebo také lze výstupu dosáhnout pomocí jednotlivých částí programu zvlášť za použití tlačítek na panelu, který se zobrazí po stisku tlačítka Advanced. Tento panel je zobrazen na obrázku 9.

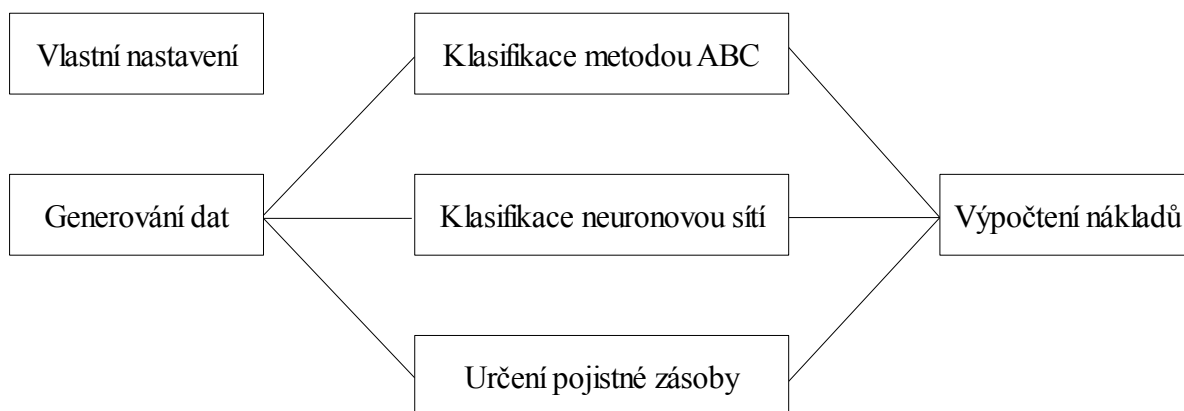


Obrázek 9: Advanced panel

Zdroj: Neuron Network

Po stisku tlačítka Settings se zobrazí panel se všemi nastavitelnými parametry, které jsou nastaveny na předdefinované hodnoty, které lze podle potřeb upravovat. Po stisknutí tlačítka View paths se zobrazí panel s cestami souborů, které jsou v programu použity, a na zobrazeném panelu je také možno tyto cesty nastavit. Tlačítkem End se ukončí spuštěná instance programu.

Rozčlenění programu je zachyceno na obr. 10 a v dalších částech práce bude každý blok popsán v jednotlivých kapitolách.



Obrázek 10: Blokové schéma programu

Zdroj: vlastní zpracování

Zdrojový kód programu je přiložen k práci a je uveden v seznamu příloh (Příloha A).

4.2.1 Vlastní nastavení

Tato část programu není závislá na ostatních blocích. Pomocí ní je možné nastavovat všechny parametry, které nějakým způsobem ovlivňují chod programu. V procesu generování dat jsou nastavovány počty generovaných položek. V procesu klasifikace neuronovou sítí lze nastavovat počty neuronů v jednotlivých vrstvách a parametry ovlivňující učení neuronové sítě, mezi které patří citlivost, přesnost, parametr učení a maximální počet učících cyklů. Ve dalším bloku, pomocí kterého se klasifikují generovaná data metodou ABC, je možné nastavovat, podle kterého atributu se budou vstupní data klasifikovat metodou ABC a jaké jsou hranice pro jednotlivé skupiny této metody. Ve zbylých dvou blocích programu není třeba žádné parametry nastavovat. Všechny potřebné hodnoty jsou jasně definované a hodnoty používané v těchto částech programu jsou předávány z předchozích bloků.

Po spuštění programu jsou všechny parametry nastaveny na přednastavené hodnoty (obrázek 11), které jsou uloženy v textovém souboru Settings.txt. Parametry lze přednastavit buď přepsáním textového souboru, což může být nebezpečné z důvodu vložené špatné hodnoty, která by způsobila nefunkčnost programu, nebo uložením nových hodnot. Stisknutím tlačítka Save to file se ve spuštěné instanci programu takto nastavené

hodnoty uloží do textového souboru. Lze také změnit nastavení programu pouze pro danou instanci bez uložení do textového souboru. Tuto funkci zastává tlačítko Save.

Settings	
Count items	100
Count training items	2500
Parameter method ABC	Cena
Group A [%]	80
Group B [%]	95
Count neurons in 1st level	6
Count neurons in 2nd level	6
Count neurons in 3rd level	3
Precision	0,010
Sensitivity	1
Learning parameter	0,010
Count learning cycle [ths]	1000
Cancel Save Save to file	

Obrázek 11: Panel Vlastní nastavení

Zdroj: Neuron Network

4.2.2 Generování dat

Jak již bylo nastíněno v úvodu praktické části, získání potřebných dat z praxe by bylo velice komplikované, a proto byla vybrána možnost si data generovat pomocí generátoru dat. Existuje mnoho předdefinovaných generátorů dat. Ve své diplomové práci jsem ale zvolil vlastní vytvořený generátor dat. Proto je současností vytvořeného programu i blok, který umožňuje potřebná data vygenerovat.

4.2.2.1 Generované atributy

Generovaná vstupní data obsahují 5000 položek s logistickými atributy, které jsou v praxi běžně používány, a jsou rozdělena do dvou množin dat. Více o datových tocích v kapitole 4.2.2.3, ve které jsou detailněji popsány generované množiny. Obě generované množiny dat obsahují totožné atributy, které popisují každý jednotlivý prvek.

Tabulka 1 znázorňuje tyto atributy:

Tabulka 1: Generovaná data – popis atributů

Název položky	Zkratka atributu	Měrné jednotky	Interval hodnot	Typ atributu
Číslo položky	Položka	-	-	-
Cena položky	Cena	Kč	100 – 1000	Generovaný
Maximální doba dodání	MDD	Dny	1 – 60	Generovaný
Počet kusů v jedné objednávce	Množství	Ks	1 – 1000	Generovaný
Důležitost ve výrobě	Důležitost	10 – vysoká důležitost 1 – nízká důležitost	1 – 10	Generovaný
Riziko nedodání objednávky	Míra rizika	1 – 100%	1 – 100	Generovaný
Spotřeba ve výrobě	Spotřeba	ks/měsíc	1 – 100	Generovaný
Pojistná zásoba	Zásoba	Ks	-	Vypočtený
Náklady na jednu položku	Náklady	Kč/den	-	Vypočtený

Zdroj: vlastní zpracování

V prvním sloupci je název atributu, ve druhém je jeho zkrácený název, který je použit v textovém souboru s vygenerovanými daty, ve třetím jsou jeho měrné jednotky, ve čtvrtém sloupci jsou uvedeny intervaly, kterých může daný atribut nabývat a v posledním, zdali se tento atribut generuje, nebo je počítán z jiných předcházejících atributů.

Atribut Číslo položky je primárním klíčem, který jednoznačně identifikuje danou položku. Všechny generované hodnoty jsou celočíselné.

4.2.2.2 Vlastnosti generovaných dat

Každý jednotlivý atribut je generován v určitém číselném intervalu. Tento interval je pevně nastaven v kódu a jeho rozmezí není možné změnit na uživatelské úrovni. Další vlastností generovaných dat je jejich vzájemná závislost, která je nutná pro aplikovanou neuronovou síť, která tyto vzájemné závislosti mezi daty potřebuje ke svému „naučení“. V případě, že by data byla generována bez jakýchkoliv pravidel, neuronová síť by nebyla

schopná nastavit svoji konfiguraci tak, aby mohla správně klasifikovat data z testovací množiny. Z toho důvodu jsou data generována podle jasně definovaných pravidel. Také bylo potřeba zajistit, aby pravidla nebyla na sobě závislá nahodile. Proto byl určen jeden atribut, který není závislý na žádném z dalších atributů (atribut Důležitost). V závislosti na hodnotách tohoto atributu jsou generovány hodnoty dalších atributů, ovšem ne pro všechny hodnoty, kterých může nezávislý atribut nabývat. Toto nám zajistí, že generované hodnoty nebudou z omezeného intervalu, ale mohou nabývat různých hodnot z vymezeného intervalu.

Hodnoty závislých atributů jsou generovány podle následujících podmínek:

- Cena

200 – 500	důležitost ve výrobě < 5
-----------	--------------------------

600 – 1 000	důležitost ve výrobě ≥ 7
-------------	--------------------------

- Maximální doba dodání

1 – 10	počet kusů v jedné objednávce < 300
--------	-------------------------------------

8 – 60	počet kusů v jedné objednávce > 500
--------	-------------------------------------

- Počet kusů v jedné objednávce

1 – 100	cena > 500
---------	------------

500 – 1 000	cena < 300
-------------	------------

- Spotřeba ve výrobě

1 – 50	počet kusů v jedné objednávce < 300
--------	-------------------------------------

200 – 300	počet kusů v jedné objednávce > 800
-----------	-------------------------------------

- Riziko nedodání objednávky

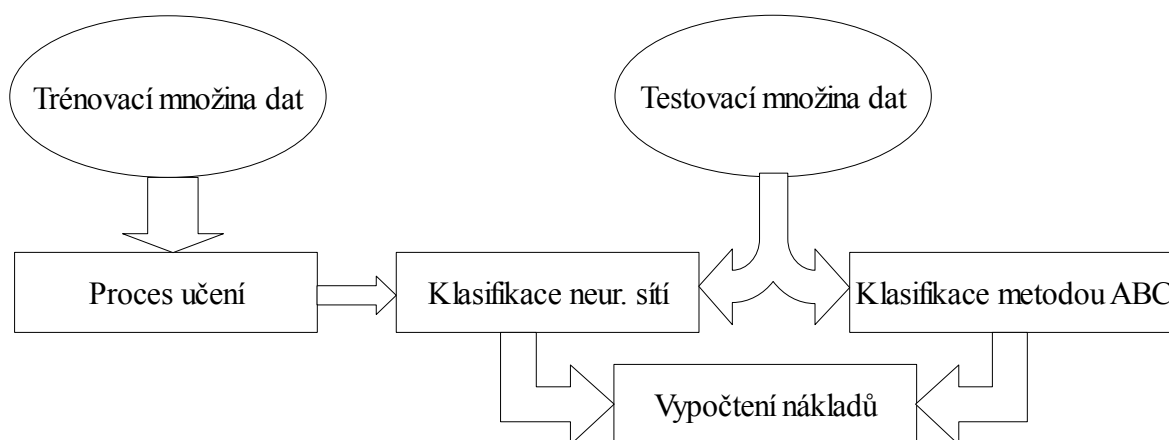
1 – 70	důležitost ve výrobě > 7
--------	--------------------------

40 – 100	důležitost ve výrobě < 6
----------	--------------------------

4.2.2.3 Datové toky v neuronové síti

Data jsou členěna do dvou množin o stejném počtu prvků. Jednotlivé skupiny dat jsou uloženy ve dvou textových souborech. Tréninková množina je uložena v souboru

trainingData.txt a testovací data jsou uložena v souboru data.txt. Na obrázku 12 jsou zachyceny datové toky v programu Neuron Network.



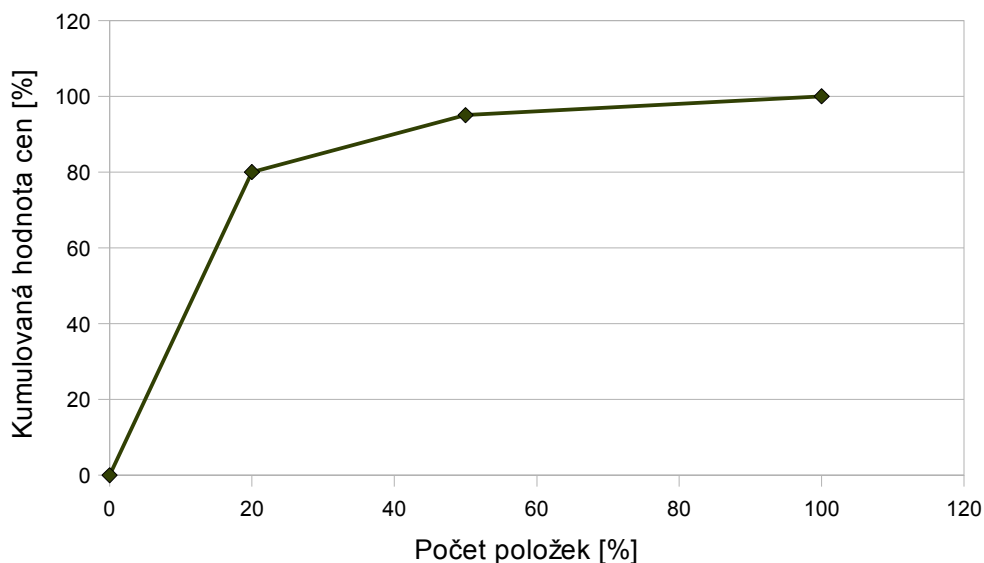
Obrázek 12: Tok dat v programu

Zdroj: vlastní zpracování

Data z trénovací množiny jsou použita ke konfiguraci neuronové sítě. Z takto nakonfigurované neuronové sítě jsou poté vzaty váhové vektory, které slouží ke klasifikaci dat z testovací množiny, která je také použita ke klasifikaci metodou ABC. Obě množiny klasifikovaných dat jsou poté předány bloku Vypočtení nákladů, kde jsou vypočteny celkové náklady sloužící k porovnání úspěšnosti obou metod klasifikace.

4.2.3 Klasifikace metodou ABC

Tento blok programu má za úkol klasifikovat generovaná data metodou ABC. Jak již bylo zmíněno v teoretické části diplomové práce v kapitole 3.1.6, analýza ABC vychází z Paretova optima 80:20. Graficky toto optimum zachycuje obrázek 13. Z grafu je patrné, že pouze 20% z celkového počtu položek představuje 80% kumulované hodnoty, která může odpovídat např. zisku, tržbám nebo ročnímu obratu. Především na tyto položky by měla firma nejvíce zaměřit svoji pozornost. Dalších 30% položek představuje pro podnik 15% kumulované hodnoty a zbylých 50% z celkového počtu všech položek odpovídá pouhým 5% kumulované hodnoty. Těmto položkám podnik nemusí věnovat v podstatě žádnou pozornost, protože náklady na jejich sledování vysoko přesahují potenciální zisk.



Obrázek 13: Paretovo optimum 80:20

Zdroj: vlastní zpracování

Na začátku aplikace metody ABC je nutné stanovit, podle jakého parametru se budou generovaná testovací data rozčleňovat do skupin a procentní hranice pro každou ze tří předdefinovaných skupin – A, B a C. V programu je členící parametr přednastaven na atribut Cena. Lze jej ale podle potřeb změnit ve vlastních nastaveních na jiný libovolný atribut. Poté jsou data pomocí programu sestupně seřazena podle členícího atributu. Nakonec jsou takto seřazená data klasifikována do jednotlivých skupin v závislosti na nastavených procentních hranicích. V práci jsou přednastaveny hodnoty 80% pro skupinu A a 95% pro skupinu B. Zbýlých 5% jsou data klasifikována do skupiny C. Pro takto klasifikovaná data jsou poté vypočteny celkové náklady, jak je uvedeno v kapitole 4.2.6. Všechna klasifikovaná data jsou uložena v textovém souboru ABC.txt. V souboru jsou také uloženy hodnoty vypočtených celkových nákladů a nákladů pro jednotlivé skupiny.

4.2.4 Klasifikace pomocí neuronové sítě

Mezi parametry, které lze u neuronové sítě nastavit, patří počty neuronů v jednotlivých vrstvách. Počet neuronů ve vstupní vrstvě je nastaven v závislosti na počtu vstupních atributů. Na začátku procesu učení byl nastaven počet vstupních neuronů na hodnotu 1 tak, aby bylo možné vzájemně porovnat klasifikaci metodou ABC

a klasifikaci neuronové sítě. Poté byl počet neuronů postupně zvyšován a bylo testováno chování a učení neuronové sítě, zdali je schopná klasifikovat i při vyšším počtu vstupních parametrů. Počet neuronů ve skryté vrstvě většinou odpovídá počtu neuronů ve vstupní vrstvě. Pouze v případě, že by neuronová síť nesprávně klasifikovala vstupní data, lze tento počet neuronů snížit. Pouze však do minimálního počtu, který je roven počtu neuronů ve výstupní vrstvě. Počet neuronů ve výstupní vrstvě je odvozen od počtu skupin, do kterých mají být vstupní data klasifikována. Tato hodnota je nastavena na hodnotu 3, aby bylo možné ekvivalentně porovnat výsledky klasifikace metodou ABC a klasifikace pomocí neuronové sítě. Mezi další parametry, které lze u neuronové sítě nastavit, je možné zahrnout přesnost učení, citlivost, parametr učení ε a minimální, resp. maximální počet cyklů učení. Přesnost učení lze použít k ukončení procesu učení tak, jak je popsáno v kapitole 4.2.4.3. Další parametrem je citlivost, který by měl nabývat nenulové hodnoty. V programu je nastaven na hodnotu -1 a je přidán ke vstupnímu vektoru. V modelu neuronové sítě je tento parametr nazýván práh. Pomocí parametru učení, který je popsán v kapitole 3.2.5.2.3 a je obsažen ve vzorci (2.13), se nastavuje délka kroku v procesu učení. Tento parametr rozhoduje o celkové chybě učení (obrázek 7) a v případě jeho špatného nastavení se nemusí neuronová síť správně nakonfigurovat. Na začátku testování byl nastaven na hodnotu 0,01, ale během testování byl v závislosti na generovaných datech upravován tak, aby se podařilo neuronovou síť správně nakonfigurovat. Testovaná hodnota tohoto parametru se pohybovala v intervalu $<0,05; 0,1>$. Posledním nastavitelným parametrem u neuronové sítě je počet cyklů učení. Maximální délka a minimální délka jsou použity v ukončovací podmínce procesu učení, která je obsáhleji popsána v kapitole 4.2.4.3. Na začátku testování jsou oba parametry přednastaveny na hodnoty v řádech tisíců.

Všechny výše uvedené parametry, popř. jejich změna, mají na učení neuronové sítě velký vliv, a proto jsou na začátku testování nastaveny na doporučené hodnoty a během testování se poté v závislosti na úspěšnosti učení neuronové sítě upravují.

4.2.4.1 Normování vstupních dat neuronové sítě

Generovaná vstupní data do neuronové sítě nenabývají hodnot ze stejných intervalů, což v důsledku znamenalo, že proces učení neuronové sítě nemohl být vykonán správně. Například atribut Cena může nabývat hodnot v řádech tisíců, oproti tomu atribut Míra rizika pouze hodnot v řádově desítkách. Tato vzájemná disproporcionalita způsobila

nesprávné nastavení synaptických vah během procesu učení. Tuto rozdílnost vstupních dat bylo nutné odstranit a převést tato data na vzájemně měřitelnou jednotku.

Toho bylo docíleno pomocí vzorce (3.1)

$$i_{norm.} = \frac{i_{vst.}}{i_{max}}, \quad (3.1)$$

kde

$i_{norm.}$... normovaná hodnota

$i_{vstp.}$... určitá vstupní generovaná hodnota

i_{max} ... maximální možná hodnota, kterou může daný atribut nabývat

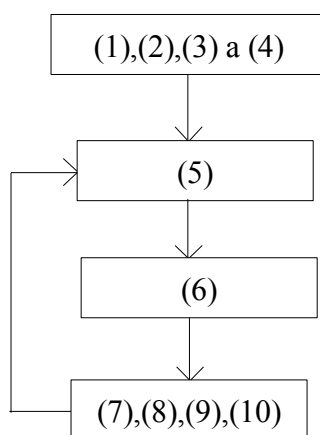
Vzorec (3.1) převede všechny hodnoty do intervalu $<0, 1>$, které jsou poté použity jako vstup do neuronové sítě. Proces normování je prováděn pomocí metody¹⁰ v programu Neuron Network a je použit na obě generované množiny dat – tréninkovou i testovací.

4.2.4.2 Proces učení neuronové sítě

Proces samotného učení neuronové sítě se skládá z několika kroků. Na začátek je nutné nejdříve získat požadovaný výstup k daným trénovacím vzorům a rozdílem skutečného a požadovaného výstupu je poté počítána celková chyba učení. Požadovaný výstup je získán klasifikací dat z trénovací množiny podle daných klasifikačních kritérií. Dalším krokem je přidání parametru citlivosti (prahu) ke vstupním datům, a tím dojde k rozšíření vstupního vektoru o jeden rozměr. Poté jsou vstupní data znormována. Tato znormovaná trénovací množina je určena k naučení neuronové sítě, resp. k nastavení synaptických vah skrytých a výstupních neuronů.

Po splnění ukončovací podmínky (viz. kapitola 4.2.4.3) je celý proces učení ukončen a konfigurace neuronové sítě je uložena do textových souborů w1.txt a w2.txt.

¹⁰ Metoda je část zdrojového kódu programu a obsahuje algoritmus se vzorcem (3.1), pomocí kterého jsou vstupní data znormována.



Obrázek 14: Algoritmus procesu učení

Zdroj: vlastní zpracování

- (1) Položení diskrétního času rovno 1
- (2) Vynulování všech pomocných proměnných
- (3) Nastavení parametrů neuronové sítě
- (4) Nastavení synaptických vah na hodnoty z intervalu $< -1, 1 >$ (hodnoty se nastavují pro každou instanci zvlášť)
- (5) Inkrementace diskrétního času o jednotku
- (6) Pro každý tréninkový vzor vykonej
 - (a) Vypočtení stavů a vnitřních potenciálů všech neuronů
 - (b) Pomocí zpětného šíření vypočtení parciální derivace v bodě pro každý nevstupní neuron
 - (c) Vypočtení gradientu parciální chybové funkce v bodě
 - (d) Vypočtení nové hodnoty parciální chyby přičtením její změny k původní hodnotě
- (7) Nastavení změny synaptických vah, která se rovná celkové hodnotě parciální derivace v bodě
- (8) Vypočtení nové hodnoty synaptických vah přičtením její změny k původní hodnotě
- (9) Vypočtení celkové chyby sítě pro současnou hodnotu synaptických vah

(10) Je-li chyba menší než požadovaná odchylka nebo učící cyklus proběhl v požadovaném opakování, proces učení ukonči, jinak (5)

S učícím se vstupem a vypočtením parciálních derivací se poté signál šíří zpět směrem od výstupu ke vstupu. Z tohoto výpočtu při zpětném algoritmu je patrné, že pro každý tréninkový vzor probíhá nejdříve aktivní režim pro jeho vstup a to tím způsobem, že se tyto informace šíří od vstupu směrem k výstupu. Poté je tento výstup porovnáván s požadovanou hodnotou. Celý proces probíhá v daných sekvencích po jednotlivých vrstvách, avšak v rámci jedné vrstvy probíhá paralelně.

4.2.4.3 Nastavení ukončující podmínky učení

Nastavení, za jakých okolností se má neuronová síť přestat učit, je velice důležité. Možností, jak tento proces ukončit, je několik. Délku procesu učení lze ovlivnit především parametry přesnosti učení a počtem učících cyklů. První možností, jak definovat ukončovací podmínku, je nastavení vhodné hodnoty parametru přesnosti učení. Podstata této ukončovací podmínky spočívá v tom, že se porovnává skutečný výstup neuronové sítě, který odpovídá hodnotám výstupních neuronů s požadovaným výstupem. Pokud je celková odchylka skutečných a požadovaných hodnot pro všechny testovací vzory nižší než nastavená přesnost učení, proces učení je poté ukončen. V opačném případě se celý proces opakuje. Na jeho konci se opět vyhodnotí, zdali je celková chyba menší než přesnost učení. Tento proces se opakuje až do té doby, než je celková chyba minimálně rovna parametru přesnosti učení. Nevýhodou této ukončovací podmínky je fakt, že se nemusí podařit neuronovou síť na trénovací množině dostatečně naučit takovým způsobem, aby celková chyba učení někdy dosáhla nižších hodnot než je požadovaná přesnost. To má za následek, že proces učení by nebyl nikdy ukončen. Tato podmínka nebyla v práci použita.

Proces učení lze také ukončit po vykonání definovaného počtu učících cyklů. Výhodou této ukončovací podmínky je záruka toho, že proces učení bude vždycky ukončen. Naopak jeho nevýhodou je to, že není zaručeno, že po ukončení procesu učení nemusí být neuronová síť nastavena na neoptimálnější konfiguraci.

Poslední možností, jak cyklus učení ukončit, je sledování celkové chyby učení, která daná součtem jednotlivých chyb učení pro každý vzor z trénovací množiny (vzorec 3.2).

$$E_i = \frac{\sum_{i=1}^n (O_{ri} - O_{si})^2}{t} \quad (3.2)$$

kde

E_i ... chyba učení na jeden trénovací vzor

O_{ri} ... požadovaný výstup

O_{si} ... skutečný výstup

t ... počet použitých cyklů učení

Na konci každého učicího cyklu se porovnají současná s předchozí celkovou chybou učení. V případě, že je přecházející celková chyba vyšší než aktuální, pokračuje se i nadále v učení. V opačném případě se proces učení ukončí. Tuto podmínku si lze graficky představit z obrázku 7, kde je celková chyba znázorněna. V případě, že je předchozí chyba větší než aktuální hodnota, pohybujeme se po klesající části křivky celkové chyby. Poté proces učení bude pokračovat až do okamžiku, kdy se dostaneme do lokálního nebo globálního minima chybové funkce. Poté již bude aktuální hodnota celkové chyby vyšší než hodnota předchozí a proces učení je poté ukončen.

Ukončovací podmínka, která byla v práci zvolena, je kombinací počtu cyklů učení a sledování chyby učení a má tvar:

opakuj, dokud platí $((t < t_{\min}) \text{ nebo } ((t > t_{\max}) \wedge (sumaErrSq < prewSumaErrSq)))$,

kde

t ... počet vykonaných cyklů

$t_{\min}$... minimální počet vykonaných cyklů učení

$t_{\max}$... maximální počet vykonaných cyklů učení

$sumaErrSq$... chyba učení při t

$prewSumaErrSq$... chyba učení při $t - 1$

Testováním bylo zjištěno, že chyba učení nemá pouze klesající nebo rostoucí charakter, ale během procesu učení se tzv. zavlní. Proto bylo nutné přidat i podmínku na minimální počet učicích cyklů. Tím je zajištěno, že proces učení neskončí při nalezení prvního lokálního minima. Hodnota minimálního počtu cyklů ovšem musí být nastavena

úměrně vzhledem k počtu vstupních dat a výkonu testovacího počítače. V případě, že je nastavena na příliš vysokou hodnotu, není testovací počítač schopen proces učení dokončit.

4.2.4.4 Testování a nastavování parametrů neuronové sítě

Po procesu normování vstupních dat a procesu učení neuronové sítě byla provedena poslední a nejdůležitější fáze praktické části. Jedná se o testování, během kterého docházelo k přenastavování parametrů v závislosti na generovaných vstupních datech. Mezi parametry, které byly během testování měněny, jsou strmost sigmoidální funkce, parametr rychlosti učení a minimální počet cyklů učení.

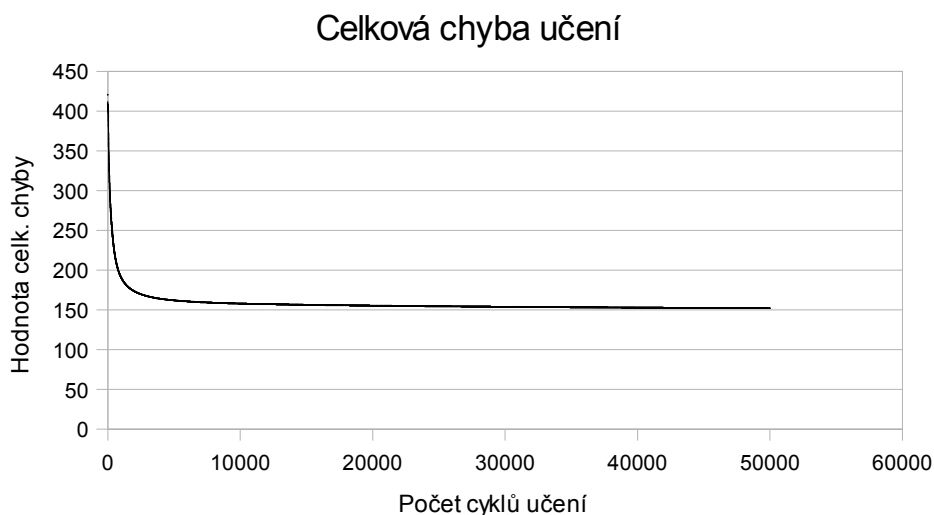
Prvním jmenovaným parametrem je strmost sigmoidální funkce (vzorec 2.8), která ovlivňuje velikost výstupu skryté a výstupní vrstvy. Přestože jsou vstupní data normována, docházelo k velkým diferencím mezi vstupními hodnotami do daných vrstev neuronových sítí u jednotlivých testovacích vzorů, což způsobovalo, že výstupem dané vrstvy neuronové sítě byly konstantní hodnoty. To mělo dále za následek, že dané vzory byly nesprávně klasifikovány neustále do stejné skupiny. V případě, že vstupní hodnoty nabývaly velmi vysokých hodnot, bylo nutné strmost zvýšit, a tím došlo ke oblejšímu schodovitému průběhu sigmoidální funkce. V opačném případě, kdy vstupní hodnoty nabývaly příliš nízkých hodnot, se strmost nastavila na nižší hodnoty, což způsobilo, že sigmoidální funkce se stala ostře schodovitou. U takto modifikované funkce již nebyl výstup konstantní, ale nabýval rozdílných hodnot. Při dosažení extrémních hodnot by tvar sigmoidální funkce odpovídal přímce v případě, kdy by strmost byla nekonečno. Tvar ostré nelinearity by nastal, když by se strmost rovnala nulové hodnotě.

Dalším parametrem, který byl během testování nastavován, je rychlost učení. Tento parametr je obsažen ve vzorci (2.13) a jeho velikost určuje změnu synaptických vah. Jak již bylo zmíněno v předcházející kapitole, na začátku testování byl parametr nastaven na hodnotu 0,01. Tato hodnota byla ale příliš nízká a celková chyba učení klesala v závislosti na počtu učících cyklů velmi pomalu. Proto byla tato hodnota nepatrně zvýšena. Pokud by byla zvýšena na několikanásobně vyšší hodnotu než původní, celková chyba učení by neklesala, ale její hodnoty by oscillovaly v určitém intervalu. To by způsobilo nedosažení minima celkové chyby učení, resp. nesprávnou konfiguraci

neuronové sítě. Optimální nalezený interval používaný v práci je od hodnoty 0,05 do hodnoty 0,1.

Posledním parametrem, který byl předmětem testování, je minimální počet cyklů učení. Nalezení jeho správného nastavení bylo velmi složité. Pokud byl počet učících cyklů příliš malý, neuronová síť se nedokázala během tak krátké doby správně nakonfigurovat, a pokud nabývala hodnoty příliš vysoké, došlo k tzv. přetrénování neuronové sítě. To se projevuje tak, že takto nakonfigurovaná neuronová síť si data z trénovací množiny už příliš pamatuje a poté již není schopná správně klasifikovat neznámá data z množiny testovací. Optimální hodnoty, které byly během testování nalezeny, spadají do intervalu pět až dvacet tisíc cyklů učení.

Sledované veličiny, které sloužily jako nástroj ke zjištění správnosti nastavení parametrů, byly celková chyba učení a procentuální úspěšnost klasifikace neuronové sítě. U celkové chyby učení se sledoval její vývoj, který je závislý na počtu trénovacích vzorů nebo na počátečním nastavení synaptických vah. Jeden z grafů, který odpovídá dosaženým výsledkům (kapitola 4.3.1) uvedený na obrázku 15.



Obrázek 15: Graf celkové chyby učení

Zdroj: vlastní zpracování

Z grafu na uvedeném obrázku 15 je patrné, že se jedná o funkci, která klesá v celém svém definičním oboru. Což je ve svém smyslu pozitivní, protože v závislosti na počtu učících cyklů a parametru rychlosti učení celková chyba neustále klesá. Ovšem na druhou stranu také s rostoucím počtem cyklů roste i náročnost a délka konfigurace neuronové sítě.

Současně může nastat i stav přetrénovanosti dat. Proto musí být zvolena vhodná kombinace počtu cyklů učení a celkové chyby učení. Jedním z cílů testování tedy bylo nalezení této vhodné kombinace. Tato kombinace byla nalezená graficky z průběhu funkce celkové chyby učení. Z výše znázorněného grafu lze vyčíst, že do počtu 15.000 cyklů učení chyba klesá relativně rychle a od hodnoty 20.000 cyklů učení zůstává prakticky konstantní. Pro tento případ by proto zvolený počet učicích cyklů neměl přesáhnout tuto hodnotu.

Druhým sledovaným ukazatelem úspěšnosti naučení neuronové sítě je její schopnost úspěšně klasifikovat vzory z testovací množiny. Tyto klasifikované vzory se porovnávaly s hodnotou na výstupu, na kterou měl být daný vzor klasifikován. Hodnota, při které byla považována neuronová síť za správně nakonfigurovanou, byla hodnota 85%. V případě, že tato hodnota byla po procesu naučení a procesu klasifikace menší, byly změněny testovací parametry a poté se testovalo, zdali se úspěšnost zvýšila. Tento postup se opakoval až do té doby, než bylo dosaženo požadované hodnoty úspěšnosti klasifikace neuronové sítě.

4.2.4.5 Klasifikace hodnot výstupních neuronů

Výstup z neuronové sítě je ekvivalentní s hodnotami výstupní vrstvy, které jsou dány sigmoidální funkcí (2.8) a nabývají hodnot z intervalu $<0, 1>$. Aby bylo možné tato reálná čísla určitým způsobem přetransformovat na určitou klasifikační skupinu, bylo nutné zavést přesně definovaná pravidla. Podle těchto pravidel lze poté jasně stanovit, do jaké skupiny daný vzor patří. Obecně platí, že každé skupině, do které chceme vstupní data klasifikovat, odpovídá jeden výstupní neuron.

Jednou z možností, jak výstupy klasifikovat pro dané vzory z testovací množiny, je nalezení jejich nejvyšších výstupních hodnot. V případě, že nejvyšší nalezená hodnota odpovídá hodnotě prvního výstupního neuronu, je tento vzor klasifikován do první skupiny, resp. do skupiny A. Analogicky jsou klasifikovány poté i vzory do skupin B a C. Úspěšnost této klasifikace však dosahovala maximálně 50%.

Další možností je stanovení specifických limitů pro každý výstupní neuron. V případě, že aktuální hodnota pro daný vzor bude větší, případně menší než definovaný limit, je tento vzor klasifikován do dané klasifikační množiny, která odpovídá výstupnímu neuronu. V práci byla použita druhá popsaná možnost klasifikace hodnot výstupních

neuronů, protože hodnoty z těchto neuronů, u dat z testovací množiny, nabývaly takových hodnot, že nebylo možné, je jasně klasifikovat první uvedenou metodou.

4.2.5 Určení pojistné zásoby

Důvod, proč je v práci počítaná pojistná zásoba, je ten, že tato veličina slouží k výpočtu nákladů na jednotlivé položky a pomocí dalších operací lze aplikované metody klasifikace vzájemně porovnat a určit, která z nich je efektivnější. V teoretické části v kapitole 3.1.7 je nastíněno, jakým způsobem by se měla pojistná zásoba spočítat. Tímto způsobem ale není možné zajistit vypovídající hodnotu vypočtených pojistných zásob. To je způsobeno tím, že data jsou generována a tím pádem nedošlo ke splnění všech podmínek.

Z tohoto důvodu byl použit zjednodušující vzorec (3.2), který je aplikován v modelových příkladech, kdy je potřeba počítat s pojistnou zásobou. Hodnoty pojistné zásoby pro každou generovanou položku jsou počítány pomocí metody programu po vygenerování hodnot potřebných k jejímu správnému výpočtu.

Vzorec má tvar:

$$\sigma_c = \frac{1}{2} S_{max} R_{max}, \quad (3.3)$$

kde

S_{max} ... maximální spotřeba za R_{max}

R_{max} ... maximální doba dodání zboží od podání objednávky

V tomto bloku programu se nejvíce projevilo nezískání dat z praxe, a proto zde musela být použita největší abstrakce. V případě, že by se data podařilo získat, by bylo možné po malých změnách v programovacím kódu použít pro výpočet postup, který je uveden v teoretické části.

4.2.6 Výpočet nákladů

Posledním blokem, který je obsažen v programu, je blok pro vypočtení nákladů, kteý vychází z teorie uvedené v kapitole 3.1.2. V něm jsou počítány nákladové veličiny – reálné a skutečné náklady pro jednotlivé položky, skladovací náklady pro každou klasifikovanou skupinu a celkové náklady dané klasifikační metody.

Postup, jak je celkových nákladů dosaženo, je následující. Program si načte všechny klasifikované vzory z testovací množiny a pro každý tento vzor se vypočtou reálné skladovací náklady na jeden den. Vzorec, který byl pro výpočet nákladů použit, vychází z výše pojistné zásoby a má tento tvar:

$$C_r = \frac{0.3 \sigma_c P}{365} \quad (3.4)$$

σ_c ... pojistná zásoba

P ... cena zboží

Poté podle toho, do jaké skupiny daný vzor patří, se vypočítají jeho skutečné skladovací náklady na jeden den. Tyto náklady jsou počítány jako procentuální podíly z reálných nákladů pro jednotlivé vzory a celé to vychází z předpokladu, že položky z vyšší skupiny jsou podnikem více sledovány a klade se na jejich uskladnění vyšší důraz. To způsobuje, že tyto položky jsou skladovány s nižšími skutečnými náklady. V praxi si to lze představit, že položky z vyšších skupin jsou umísťovány na sklad do přístupnějších míst, což se odráží v nižších manipulačních nákladech. Případně jsou objednávány s vyšší frekvencí a tím je zajištěno, že je minimalizováno riziko jejich nedostatku, jenž by mohl způsobit ekonomickou ztrátu. Vzorec pro výpočet skutečných denních nákladů má tvar

$$C_s = C_r \cdot \phi, \quad (3.5)$$

kde

C_r ... reálné náklady pro jednotlivé skladové položky

ϕ ... nákladová konstanta pro danou klasifikační skupinu

Nákladová konstanta je definována pro každou klasifikační skupinu a může nabývat hodnot z intervalu $<0, 1>$. V práci bylo skupině A přiřazena hodnota 0.5, skupině B hodnota 0,8 a skupině C je konstanta rovna 1.

Po vypočtení skutečných nákladů u jednotlivých položek jsou počítány celkové náklady pro každou klasifikační skupinu a jejich součtem jsou zjištěny celkové náklady aplikované klasifikační metody. Tyto náklady poté slouží k formulaci závěru, která z použitých metod byla efektivnější a pomocí které lze dosáhnout nižších celkových nákladů.

4.3 Presentace výsledků

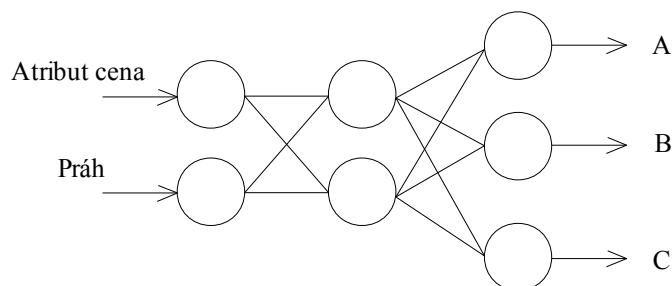
V této kapitole jsou obě klasifikační metody vzájemně porovnány a je určeno, která z nich je při minimalizování celkových skladovacích nákladů efektivnější. Hodnocení se dále rozděluje na dva případy. V prvním je porovnávána klasifikace pomocí metody ABC a klasifikace neuronovou sítí, jejíž vstup představuje pouze jeden vstupní atribut. Toto srovnání lze považovat za rovnocenné, protože na vstup každé klasifikační metody přivádíme pouze jeden atribut, na základě kterého se rozhoduje o zařazení dané položky do klasifikační skupiny. Naopak ve druhém případě se opět porovnává klasifikace prvků metodou ABC a neuronové sítě, ale jejím vstupem je již více atributů. Cílem výzkumu je dokázat, zdali klasifikací na základě více vstupních atributů je možno dosáhnout nižších celkových skladovacích nákladů.

Během hodnotícího procesu byly graficky zaznamenány následující veličiny – procentuální úspěšnost klasifikace neuronové sítě, náklady prvků z jednotlivých klasifikačních skupin a celkové skladovací náklady dané metody. Všechny tyto veličiny jsou závislé na počtu učících cyklů. Byl zvolen testovací interval od 2.000 do 20.000 učících cyklů, kde vzorkovací krok má velikost 2.000 nebo 2.500 cyklů. Průběh těchto veličin je znázorněn v jednotlivých grafech. Mezi sledované veličiny lze také zahrnout celkovou chybu učení, která ovšem sloužila jen jako pomocný ukazatel.

4.3.1 Neuronová síť s jedním vstupním atributem

Cílem tohoto výzkumu bylo rovnocenné porovnání klasifikace metodou ABC s klasifikací pomocí neuronové sítě. Cílem výzkumu tohoto modelu byla snaha zjistit, zdali může neuronová síť nahradit metodu ABC. Jako klasifikační atribut u metody ABC byla zvolena Cena, pomocí které se klasifikují položky z trénovací i testovací množiny do skupin. V praxi se ovšem neuronová síť s jedním vstupem prakticky nepoužívá, protože tím se eliminuje její hlavní přednost. V práci je použita s účelem, aby bylo možné obě klasifikace vzájemně porovnat.

Schéma neuronové sítě, která byla použita v této modelové situaci, je na obrázku 16.

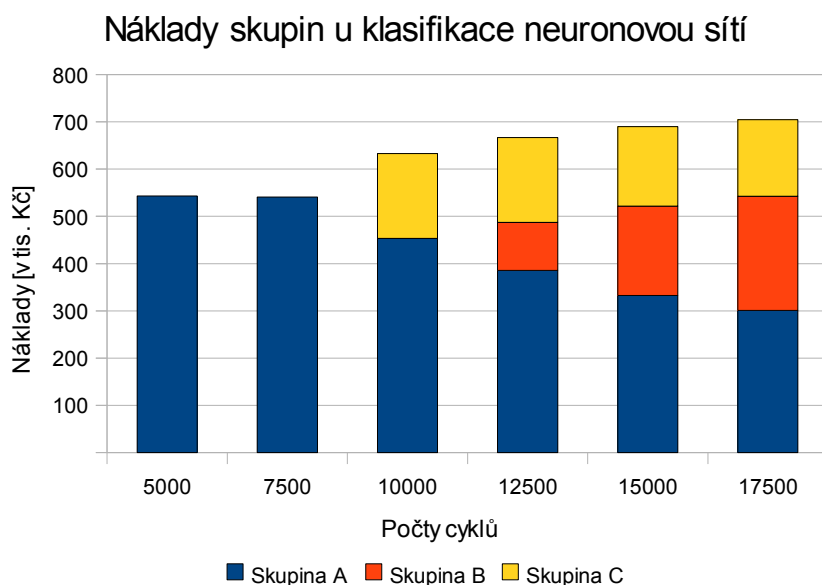


Obrázek 16: Schéma neuronové sítě s jedním vstupním atributem

Zdroj: vlastní zpracování

Data z trénovací množiny jsou poté určena ke konfiguraci neuronové sítě. Takto nakonfigurovaná síť pak klasifikuje data z testovací množiny, stejně jako metoda ABC. Klasifikační pravidla, podle kterých neuronová síť zařazovala do skupin, byla převzata od metody ABC. K oběma klasifikovaným datům se poté vypočítají náklady.

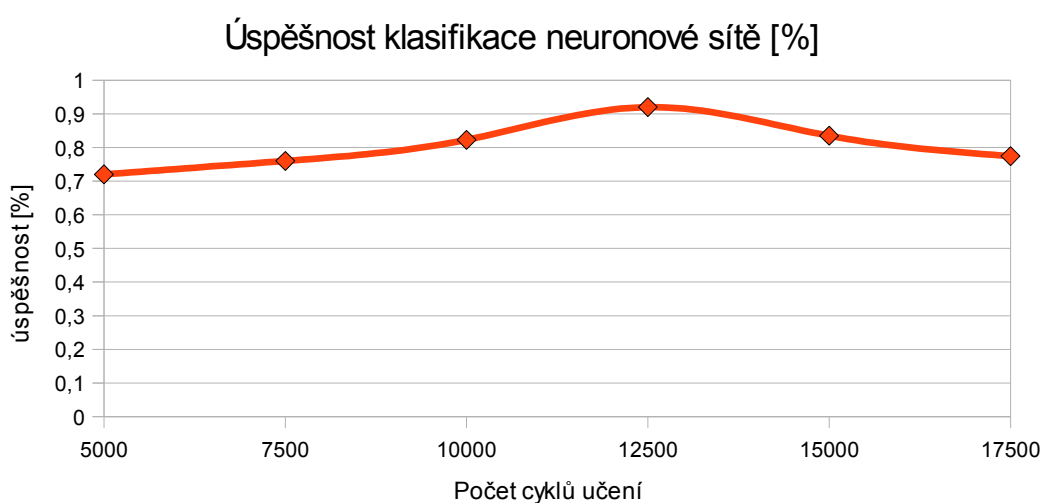
Náklady jednotlivých klasifikačních skupin jsou u metody ABC konstantní. Naproti tomu náklady klasifikací pomocí neuronové sítě jsou variabilní. To je způsobeno tím, že se síť konfiguruje v závislosti na počtu učicích cyklů. Jejich vývoj je zachycen v grafu na obrázku 17.



Obrázek 17: Skladovací náklady dat z testovací množiny jednot. klas. skupin neuronovou sítí v závislosti na počtu učicích cyklů

Zdroj: vlastní zpracování

Z grafu na obrázku 17 je patrné, že ve stavu, kdy neuronová síť není správně nakonfigurovaná, jsou všechna data zařazena do skupiny A. Se zvyšujícím se počtem cyklů a přesnější konfigurací se zvyšuje počet správně klasifikovaných položek z testovací množiny. Současně tím ale rostou i celkové náklady. To je způsobeno tím, že čím dál větší počet položek je klasifikováno do skupin B a C, které mají obecně vyšší náklady. Jednou z možných příčin může být přetrénování neuronové sítě nebo nesprávné nastavení parametrů. Z tohoto důvodu bylo nutné sledovat i úspěšnost klasifikace neuronové sítě. Její graf je znázorněn na obrázku 18.



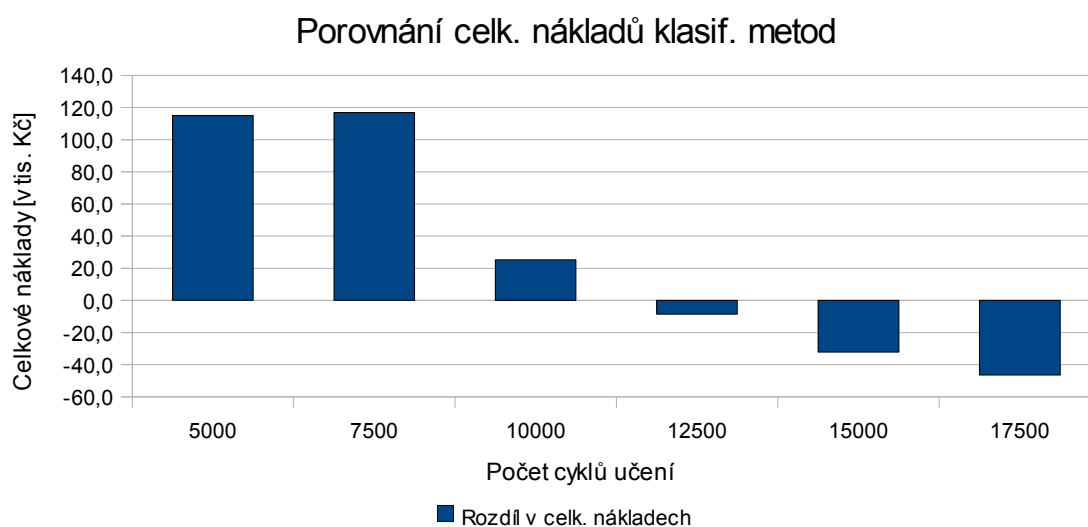
Obrázek 18: Úspěšnost klasifikace neuronové sítě dat z testovací množiny

Zdroj: vlastní zpracování

Jak z grafu na obrázku 18 vyplývá, úspěšnost klasifikace neuronové sítě z počátku pozvolna roste a po nabytí svého maxima začne pomalu klesat. Nárůst je způsoben postupným konfigurováním sítě během procesu učení, po jehož ukončení dokáže síť data klasifikovat správněji. V grafu je zřejmé, že se podařilo nejlépe nakonfigurovat síť po 12.500 cyklech učení. Tato hodnota ovšem nemusí být absolutně nejvyšší. Pokud by se použil menší vzorkovací krok, je pravděpodobné, že by se podařilo najít hodnotu ještě vyšší. Hodnoty přesahující 85% hranici úspěšnosti lze považovat za případ, kdy se neuronová síť dokázala správně nakonfigurovat a dané výstupní hodnoty lze považovat za vypovídající.

Hodnotícím kritériem o úspěšnosti obou klasifikačních metod je vzájemné porovnání skladovacích nákladů, které podniku vzniknou. Jak již bylo zmíněno,

po aplikaci metody ABC zůstávají skladovací náklady konstantní. Jejich změnu lze docílit pouze změnou klasifikačního atributu nebo změnou vstupních dat s následným klasifikačním procesem. Naopak náklady klasifikací pomocí neuronové sítě jsou variabilní. Jejich vzájemný rozdíl je znázorněn v grafu na obrázku 19, kde kladné hodnoty představují náklady, o které by podnik snížil své skladovací náklady v případě, že by použil místo metody ABC klasifikaci neuronovou sítí, která klasifikovala data z testovací množiny. Naopak záporné hodnoty představují velikost nákladů, které by podniku vznikly navíc.



Obrázek 19: Rozdíl celkových nákladů mezi oběma klasifikačními metodami

Zdroj: vlastní zpracování

Z grafu lze vyčíst, že při nízkém počtu učících cyklů, resp. při nedokonalé konfiguraci, může firma své náklady minimalizovat, ovšem to nastává v případě, kdy neuronová síť nebyla schopná správně klasifikovat, a správně tak zařadit drtivou většinu testovacích vzorů. Proto výsledné skladovací náklady nelze považovat za směrodatné. Směrodatné lze považovat náklady, u kterých přesáhla úspěšnost klasifikace alespoň 85% všech klasifikovaných vzorů. Z grafu na obrázku 18 lze vypožorovat, že toto nastává v intervalu mezi 12.000 až 13.000 učícími cykly. Pro tento interval se poté vezmou hodnoty rozdílu celkových nákladů a vyhodnotí se, zdali je přínosnější pro podnik klasifikovat metodou ABC nebo pomocí neuronové sítě.

Z modelové situace, která je vyhodnocena na obrázcích 18 a 19, je zřejmé, že je pro podnik spíše výhodné použití metody ABC, protože není jisté, že by použitím klasifikace neuronové sítě výrazněji snížil své celkové náklady.

4.3.2 Neuronová síť s více vstupními atributy

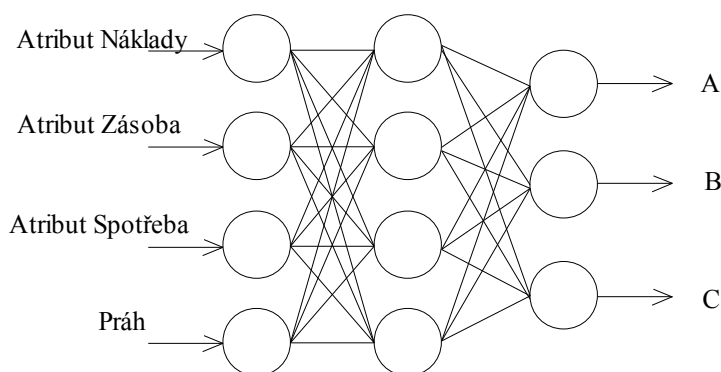
Smyslem tohoto výzkumu bylo uplatnění největší přednosti klasifikace neuronovou sítí, která spočívá v tom, že je možné klasifikovat položky umístěvané na sklad na základě většího počtu vstupních atributů. Toto umožňuje podnikům zohlednit při klasifikaci více skladových atributů, v porovnání s metodou ABC, a potenciálně si tak snížit skladovací náklady. V této modelové situaci jsou na vstup neuronové sítě přiváděny tři atributy – relativní skladovací náklady na položku, výše pojistné zásoby a spotřeba dané položky. Tento model zkoumá, zdali klasifikační neuronovou sítí lze dosáhnout nižších celkových nákladů. Klasifikační pravidla, kterými neuronová síť zařazuje do jednotlivých skupin, lze nastavit tak, aby si daný podnik výběrem a nastavením těchto pravidel minimalizoval skladovací náklady.

V práci byla nastavena taková pravidla pro vybrané atributy, aby se podařilo minimalizovat skladovací náklady pro data z testovací množiny.

- skupina A – náklady > 3000, pojistná zásoba > 2000, spotřeba > 200
- skupina B – náklady > 500, pojistná zásoba > 200
- skupina C – všechny ostatní položky

Po aplikaci na data z testovací množiny podle takto stanovených klasifikačních pravidel bylo zařazeno přibližně 17% položek do skupiny A, 30% položek do skupiny B a zbylé položky připadly skupině C. Toto procentuální rozdělení je téměř totožné s případem, jako by byla data klasifikována metodou ABC.

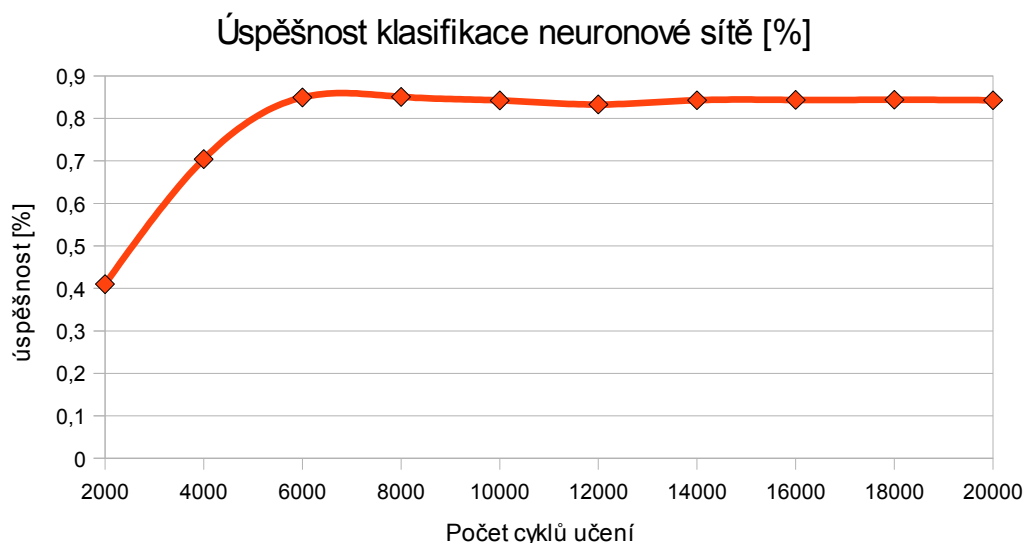
Schéma neuronové sítě, která byla použita v této modelové situaci, je na obrázku 20.



Obrázek 20: Schéma neuronové sítě s více vstupními atributy

Zdroj: vlastní zpracování

Zvoleným atributem pro klasifikaci metodou ABC jsou relativní náklady. V závislosti na něm jsou vypočteny skladovací náklady pomocí této metody. Tyto náklady jsou konstantní a během testování proto nedochází k jejich změnám. Naproti tomu celkové náklady, které získáme klasifikací neuronovou sítí, jsou proměnné. To je způsobeno postupnou konfigurací neuronové sítě, která se v závislosti na počtu učicích cyklů učí, a zpřesňuje se tak klasifikace vstupních dat. V porovnání s předchozí modelovou situací, ve které byl na vstup sítě přiveden pouze jeden atribut, je proces učení podstatně časově zdlouhavější a výpočetně náročnější. To je zapříčiněno tím, že program musí vykonat podstatně více matematických operací. Testováním bylo ale zjištěno, že počet cyklů, po kterých je neuronová síť schopná klasifikovat vstupní data s požadovanou úspěšností, je nižší, než tomu bylo v případě jednoho vstupního atributu. Jak je znázorněno v grafu na obrázku 21, úspěšnost klasifikace pomocí tohoto modelu sice nedosáhne absolutně nejvyšší hodnoty jako v předchozím zkoumaném modelu, ale hodnoty přesahující 85% hranici lze považovat za dostačující a směřovatné.



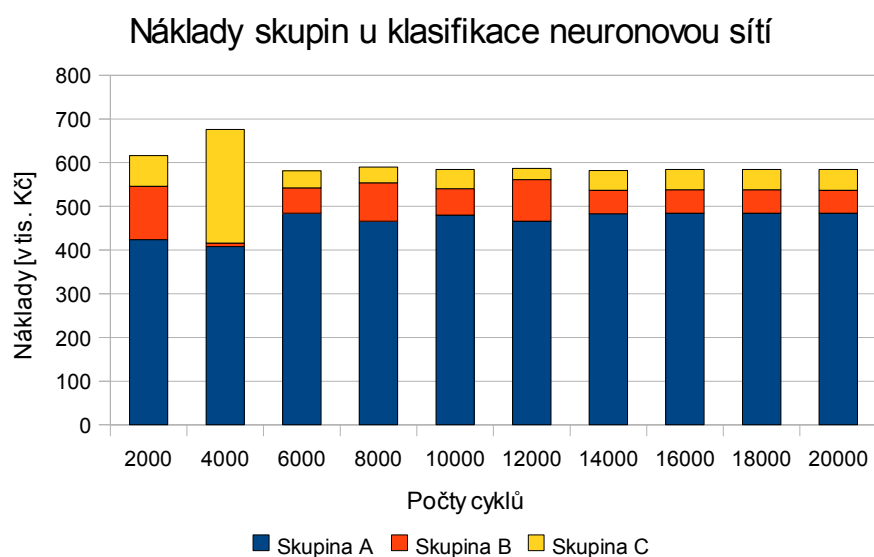
Obrázek 21: Úspěšnost klasifikace položek z testovací množiny u neuronové sítě s více vstupními atributy

Zdroj: vlastní zpracování

Rozdíl je ovšem v průběhu funkce. V tomto modelu je počáteční úspěšnost velmi nízká, avšak po dvou třech vzorkovacích krocích (jeden krok odpovídá 2.000 cyklů učení) se zvýší do požadovaného intervalu. V porovnání s předchozím modelem však v tomto intervalu zůstává i po několika dalších krocích. Toto zjištění lze charakterizovat tak, že neuronová síť s více vstupními atributy je po úspěšném naučení stálejší. Stav, kdy začne

úspěšnost klasifikace rapidně klesat, nelze přesně stanovit. Většinou toto nastane až po daleko vyšším počtu učících cyklů, než tomu bylo u modelu neuronové sítě s jedním vstupním atributem. Na základě tohoto poznatku lze říci, že efekt přetrénování nastává podstatně déle, což je způsobeno tím, že neuronová síť nerozpoznává vzájemné závislosti mezi vstupními atributy tak rychle jako v případě pouze jednoho vstupního atributu.

Úspěšnost klasifikace neuronové sítě se odráží i v další grafu na obrázku 22. Na něm jsou zachyceny náklady jednotlivých klasifikačních skupin a jejich vývoj v závislosti na počtu učících cyklů.



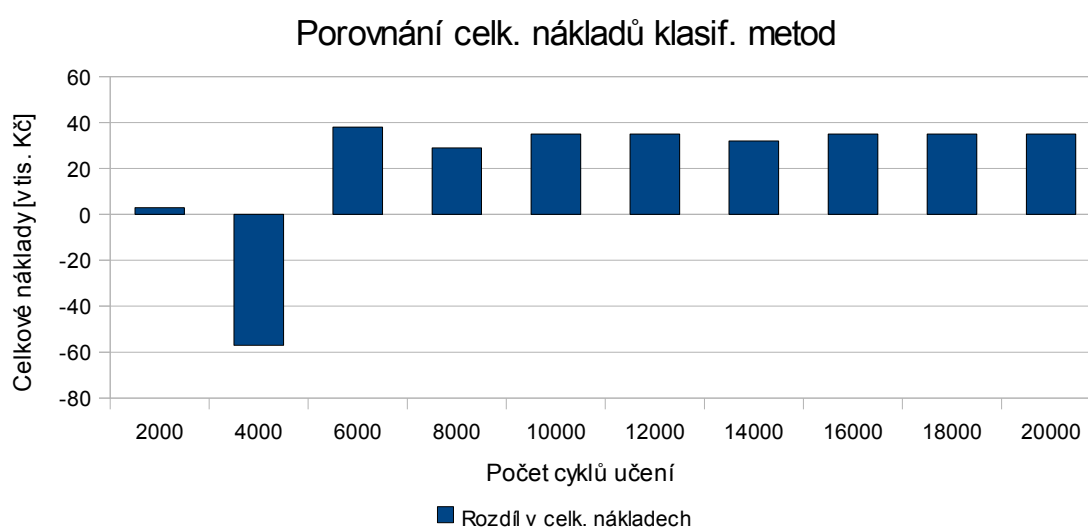
Obrázek 22: Skladovací náklady položek z testovací množiny jednot. klas. skupin u neuronové sítě s více vstupními atributy

Zdroj: vlastní zpracování

Z grafu lze vyčíst, že náklady pro skupinu A nějak výrazně nekolísají. To je způsobeno tím, že neuronová síť dokáže do této skupiny klasifikovat výrobky i ve stavu, kdy není ideálně nakonfigurovaná. Naopak zařadit položky do skupin B a C, u kterých se jejich celkové náklady s každým vzorovacím krokem mění, bylo daleko obtížnější. To je zapříčiněno tím, že hodnoty z výstupních neuronů nabývaly příliš podobných hodnot na to, aby bylo možné je jednoznačně klasifikovat do požadované skupiny.

Jako nejvíce vypovídající kritérium hodnocení obou klasifikačních metod bylo zvoleno porovnání skladovacích nákladů, které vzniknou podnikům po aplikaci jedné z použitých metod. Skladovací náklady po klasifikaci metody ABC jsou konstantní,

protože nejsou závislé na žádné další proměnné kromě klasifikačního parametru. Naproti tomu skladovací náklady, jak je znázorněno v grafu na obrázku 21, u klasifikace neuronovou sítí se mění v závislosti na správném nakonfigurování sítě, resp. na počtu učících cyklů. Jejich hodnota je dána součtem nákladů jednotlivých klasifikačních skupin. Graf, který porovnává celkové náklady obou klasifikačních metod, je na obrázku 23, kde kladné hodnoty představují náklady, o které by podnik snížil své celkové náklady v případě, že by použil místo metody ABC klasifikaci neuronovou sítí. Naopak záporné hodnoty představují velikost nákladů, které by podniku vznikly navíc.



Obrázek 23: Porovnání celkových nákladů metody ABC a neuronové sítě s více vstupními atributy

Zdroj: vlastní zpracování

U hodnot, které lze z grafu považovat za směrodatné a vypovídající, musí hodnota úspěšnosti klasifikace u neuronové sítě přesáhnout požadovanou hranici 80%. Tato podmínka je splněna při 6.000 cyklech učení, a proto lze konstatovat, že klasifikace neuronovou sítí s více vstupními atributy je v minimalizaci celkových nákladů efektivnější než klasifikace metodou ABC. To je především způsobeno nastavenými pravidly pro klasifikaci neuronovou sítí, po nichž je do nejsledovanější skupiny A zařazen vyšší počet položek než po klasifikaci metodou ABC. To v důsledku pro podnik znamená, že sice musí sledovat větší počet položek umístěvaných na sklad, ale současně mu to přinese minimalizaci celkových nákladů.

4.4 Zahrnutí vedlejších nákladů

Přestože aplikace jedné z použitých metod umožní firmě minimalizovat skladovací náklady, nezaručuje to automaticky i nižší celkové náklady. Je nutné hodnotit obě metody klasifikace z globálního pohledu, protože rozdíl v celkových nákladech u obou metod mohou být diametrální. Rozdílnost je především způsobena nástroji pro samotnou klasifikaci. Výsledky u metody ABC je možné získat použitím běžně používaných kancelářských aplikací. To podniku nezpůsobí výraznější zvýšení nákladů. Naproti tomu klasifikaci pomocí neuronové sítě není možné těmito aplikacemi jednoduše získat. Je na to zapotřebí použití speciálního programu, jehož pořizovací cena se může vyšplhat až do řádově sta tisíců korun. Mezi další vedlejší náklady, které mohou firmě navíc vzniknout, lze zahrnout například i školení svých zaměstnanců, kteří budou program používat.

5 Závěr

Cílem diplomové práce bylo zjistit, zdali je možné dosáhnout minimalizace celkových nákladů ve skladovém hospodářství použitím jiné klasifikace než v současnosti nejpoužívanější metodou ABC. Jako alternativa ke klasifikaci skladových položek byla použita metoda umělé inteligence, konkrétně neuronová síť. V práci byly testovány dva modelové případy.

Prvním případem bylo porovnání klasifikace metodou ABC a neuronové sítě s jedním vstupním atributem. Celkové náklady obou metod byly prakticky totožné, a proto lze konstatovat, že pokud podnik zohlední pouze toho hledisko, je jedno, jakou z uvedených metod zvolí. Pokud by ovšem podnik zohlednil komplexně celou klasifikační metodu, bylo by pro něj efektivnější aplikovat metodu ABC. Je to z toho důvodu, že metoda ABC není výpočetně ani časově náročná a pro zjištění jejích výsledných hodnot není třeba speciálního programu. Jinak je tomu u klasifikace pomocí neuronové sítě, u které je nutnost použití programu, který je pro klasifikaci určen. To může firmě neúměrně zvýšit náklady, než jaké by použitím tohoto programu získala. Také doba získání výsledných hodnot tímto způsobem klasifikace je podstatně delší, protože během ní je nutno provést daleko více početních operací. Také se tímto modelem nevyužije hlavní přednost neuronové sítě – klasifikace na základě více vstupních atributů.

Druhým případem bylo proto srovnání celkových nákladů získaných klasifikací metodou ABC a klasifikací neuronovou sítí s více vstupními atributy. Testováním bylo zjištěno, že rozdíl celkových nákladů jednotlivých klasifikačních metod není zanedbatelný. Podnik, který by použil klasifikaci neuronovou sítí, by si mohl snížit celkové skladovací náklady řádově o několik procent v porovnání s metodou ABC. Příčinou poklesu nákladů jsou odlišná klasifikační pravidla obou metod. U metody ABC lze nastavit pouze hranice kumulovaných součtů jednotlivých skupin a poté už není možné dále klasifikaci ovlivnit. U klasifikace neuronové sítě si lze tyto pravidla nastavit přesně podle potřeb daného podniku. Navíc je možné nastavit tato pravidla v závislosti na více vstupních attributech skladovacích položek. V konečném důsledku to znamená, že počty položek v jednotlivých skupinách jsou odlišné. Především se jedná o nejdůležitější skupinu A, na kterou podnik zaměřuje nejvíce své pozornosti a položky zařazené v této skupině minimalizují skladovací náklady.

Celkový ekonomický dopad využití klasifikace neuronovou sítí je především závislý na počtu skladovacích položek a na jejich skladovacích nákladech. Pro podniky s vyššími náklady je výhodnější spíše použít klasifikaci neuronovou sítí, protože je nutné vzít v úvahu ušetřené skladovací náklady, které by podnik získal použitým způsobem klasifikace. Nezbytné je ovšem také zohlednit i vedlejší náklady, které implementací a provozem mohou vzniknout. Především náklady spojené s pořízením nástroje pro klasifikaci dat představují nezanedbatelnou nákladovou položku. Proto podnik musí zvážit, zdali je pro něj ekonomicky výhodné použití klasifikace neuronovou sítí.

Seznam použité literatury

1. Citace

- [1] MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kolektiv: *Umělá inteligence I-IV*, Academia Praha, 1993-2003, (soubor) ISBN 80-200-0502-1.
- [2] SIXTA, J., MAČÁT, V.: *Logistika: teorie a praxe*. 1.vyd. Brno: CP Books, 2005 ISBN 80-251-0573-3
- [3] ŠÍMA, J., NERUDA, R.: *Teoretické otázky neuronových sítí*, MATFYZPRESS, 1996, ISBN 80-85863-18-9.

2. Bibliografie

- [4] BISHOP, C. M.: *Pattern Recognition and Machine Learning*, Springer, 1st edition, 2007, ISBN 0-387-31073-8.
- [5] ELLER, F.: *C# - Začínáme programovat*, 1. vyd, Grada 2002, ISBN 80-247-0324-6
- [6] ERNST, R., COHHEN, MORRIS A.: Multi-item classification and generic inventory stock control policies. *Production and Inventory Managment Journal*, Third Quarter 1988, ABI/INFORM Global
- [7] GUPTA, A., LEMUS, G.: A hybrid model based on dynamic programming, neural networks, and surrogate value for inventory optimisation applications. *Journal of the Operational Research Society*, 1999
- [8] KOCH, R.: *Pravidlo 80/20*, 2. aktualizované vyd. Management Press, 2008, ISBN 978-80-7261-175-1
- [9] KOŠTÁLOVÁ, L.: *Zlepšení podnikového procesu*. [Bakalářská práce]. Brno: Masarykova univerzita Brno – Ekonomicko-správní fakulta, 2008
- [10] KRATOCHVÍL, T.: *Skladové hospodářství konkrétního podniku*. [Bakalářská práce]. Brno: Masarykova univerzita Brno – Ekonomicko-správní fakulta, 2007
- [11] LAMBERT, D.: *Logistika*, 2. vyd. Computer Press, a.s. , 2005, ISBN 80-251-0504-0

- [12] NEJEDLOVÁ, D.: *Fonetická transkripce češtiny pomocí třívrstvé neuronové sítě*. [Výzkumná zpráva]. Liberec: Technická univerzita v Liberci – fakulta mechatroniky, 2000
- [13] NOVÁK, M. a kol. *Umělé neuronové sítě. Teorie a aplikace*. 1. vyd. C. H. BECK, Praha 1998 ISBN 80-7179-132-6.
- [14] PETZOLD, CH.: *Programování Microsoft Windows Forms v jazyce C#*, 1.vyd, Softpress s.r.o., 2003, ISBN 80-86497-54-2, Svazek 1 a Svazek 2
- [15] PETZOLD, CH.: *Programování Microsoft Windows Forms v jazyce C#*, 2. vyd, Computer Press a.s. 2006, ISBN 80-251-1058-3
- [16] RUSSELL, S. J., NORVIG, P.: *Artificial Intelligence, A Modern Approach*, Upper Saddle River: Pearson Education, 2nd edition, 2003, ISBN 0-13-080302-2.
- [17] UCHYTILOVÁ, O.: *Zásobovací logistika podniku*. [Bakalářská práce]. Brno: Masarykova univerzita Brno – Ekonomicko-správní fakulta, 2007
- [18] WARREN, M., STANFORD, ROBERT E.: A methodology for estimating the maximum profitable turns for an ABC inventory classification system. *IMA Journal of Management Mathematics*, 2006
- [19] ZAJÍČKOVÁ, P.: *Klasifikace modelů zásob* [online]. Brno: Vysoké učení technické v Brně, 2004. Dostupný z WWW: <<http://www.fce.vutbr.cz/veda/>>

Seznam příloh

Příloha A: Program Neuron Network

Příloha B: Textové soubory s výstupními hodnotami