

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NEURONOVÉ SIETE A HRUBÉ MNOŽINY

DIPLOMOVÁ PRÁCE

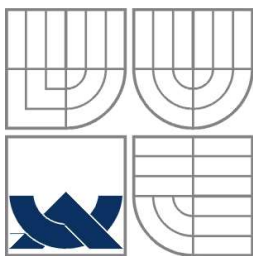
MASTER'S THESIS

AUTOR PRÁCE

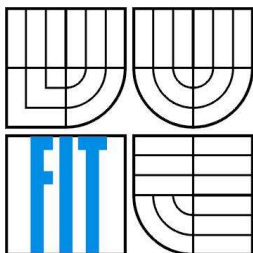
AUTHOR

Bc. Matej Čurilla

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NEURONOVÉ SÍTĚ A HRUBÉ MNOŽINY

NEURAL NETWORKS AND ROUGH SETS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Matej Čurilla

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. Ing. František Vítězslav Zbořil, CSc.

BRNO 2015

Abstrakt

Hrubé množiny a neuronové sítě poskytují dobré prostředky pro zpracování a analýzu dat. Obě metody však mají určité problémy. Tato práce se zabývá výzkumem možností spojení těchto dvou metod, z kterých budou některé implementované a porovnané s klasickým algoritmem pro analýzu výhod a nevýhod tohoto přístupu.

Abstract

Rough sets and neural networks both offer good theoretical background for data processing and analysis. However, both of them suffer from few issues. This thesis will investigate methods by which these two concepts are merged, and few such solutions will be implemented and compared with conventional algorithm to study the benefits of this approach.

Klíčová slova

hrubé množiny, neuronové sítě, hrubé neuronové sítě, hrubý neuron, hrubá logika, předzpracování dat, hrubé granulórní neuronové sítě

Keywords

rough sets, neural networks, rough neural networks, rough neuron, rough logic, data preprocessing, rough-granular neural networks

Citace

Čurilla Matej: Hrubé množiny a neuronové siete. Diplomová práce, Brno, FIT VUT v Brně, 2015

Neurónové siete a hrubé množiny

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Doc. Ing. Františka Vítězslava Zbořila, CSc.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Matej Čurilla, Bc.
27.5.2015

Poděkování

Týmto by som chcel poďakovať svojmu vedúcemu, pánovi docentovi Františkovi Zbořilovi.

© Matej Čurilla, 2015

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod.....	2
1.1 Hrubé množiny.....	2
1.2 Neurónové siete	4
2 Hrubé neurónové siete	6
2.1 Neurónové siete s predspracovaním údajov hrubými množinami	6
2.1.1 Redukcia atribútov na základe matice rozlíšiteľnosti	6
2.1.2 Redukcia na základe informačného pohľadu	7
2.1.3 Ďalšie typy predspracovania	9
2.2 Neurónové siete s hrubým neurónom	9
2.3 Neurónové siete založené na hrubej logike.....	11
2.4 Hrubé granulárne neurónové siete	13
2.4.1 Aproximačná neurónová sieť	14
3 Návrh a implementácia	16
3.1 Kontrolná neurónová sieť	16
3.2 Redukcia atribútov maticou rozlíšiteľnosti	18
3.3 Neurónová sieť s hrubými neurónmi	19
3.4 Neurónová sieť s hrubou logikou.....	21
3.5 Aproximačná neurónová sieť	22
4 Vyhodnotenie algoritmov	25
4.1 Redukcia atribútov maticou rozlíšiteľnosti	25
4.2 Neurónová sieť s hrubými neurónmi	28
4.3 Neurónová sieť s hrubou logikou.....	31
4.4 Aproximačná neurónová sieť	33
5 Záver	36

1 Úvod

Od počiatku vývoja počítačov a oboru informatiky sa objavovali problémy, ktoré neboli rozumne riešiteľné konvenčnými nástrojmi, či už z dôvodu neefektívnosti konvenčných algoritmov pre daný problém alebo konfliktov vyplývajúcich z charakteru výpočetných zariadení. Problémy konvenčných algoritmov sú často spojené s neefektívnou spracovania nepresných, konfliktných, či neúplných dát. Ako odozva na túto prekážku vznikli nové rodiny algoritmov, ktoré dnes spoločne tvoria oblasť nazývanú *soft computing*. Hoci toto označenie sa začalo používať až v deväťdesiatych rokoch, koncepty ktoré sú dnes s ním spojené boli vyvíjané už takmer od začiatku vzniku počítača. V súčasnosti sa do tejto oblasti zaraďujú teórie a na nich vzniknuté algoritmy z oblastí fuzzy množín, umelých neurónových sietí, genetických algoritmov, či teórie hrubých množín. Táto práca sa bude zaoberať najmä umelými neurónovými sieťami a teóriou hrubých množín, a ich možných spojeniach pre efektívnejšie spracovávanie dát.

Teóriu hrubých množín predstavil Zdzisław Pawlak [8] ako matematický nástroj pre analýzu nepresných dát. Tá sa pri analýze dát ukázala ako vhodným prostriedkom na vyhľadávanie závislostí medzi vlastnosťami, alebo extrakciou pravidiel a tvorbe rozhodovacích tabuliek efektívne a to aj z veľmi rozmerných dát. Ich slabinou sa však ukazuje ich slabá schopnosť vyrovnat' sa s chybnými dátami. Umelé neurónové siete sú výsledkom pokusu skopírovať proces spracovávania informácií biologickými neurónovými sieťami (ľudským mozgom). Ich potenciál spočíva v silnej schopnosti spracovávať dáta rôznych druhov, možnosti paralelných výpočtov vyplývajúcej zo štruktúry siete, schopnosti zapamätat' si údaje, učiť sa, či vysporiadať sa aj s dátami obsahujúcimi chyby. Slabina neurónových sietí sa začala objavovať s nárastom komplexnosti dát, kedy čas potrebný na ich učenie alebo spracovávanie dát začal narastať. Komplementárny charakter neurónových sietí a hrubých množín viedol ku vzniku hrubých neurónových sietí. [2]

1.1 Hrubé množiny

Teória hrubých množín [15] je založená na predpoklade, že každý objekt univerza nesie určité informácie. Objekty charakterizované rovnakými informáciami sú nerozlíšiteľné (podobné) pri pohľade na dostupné informácie. Táto relácia nerozlíšiteľnosti tvorí základ teórie hrubých množín.

Množina všetkých objektov, ktoré tvoria univerzum U , a množina všetkých atribútov A dohromady tvoria informačný systém $S = (U, A)$. Každému atribútu a môžeme tiež priradiť množinu hodnôt V_a . Ak vyberieme ľubovoľnú podmnožinu atribútov $B \subseteq A$, tá vytvára *reláciu nerozlíšiteľnosti* $I(B)$ nad univerzom U , definovanú ako $I(B) = \{(x, y) \mid x, y \in U, a(x) = a(y), \forall a \in B\}$, kde $a(x)$ je hodnota atribútu a prvku x . Pomocou $I(B)$ následne môžeme určiť *ekvivalenčné triedy*

$U / I(B) = U / I$. Ekvivalenčná trieda, nazývaná tiež *elementárna množina*, je teda množina prvkov univerza, ktoré sú navzájom nerozlišiteľné (s ohľadom na dané atribúty). Pokiaľ sa množina $X \subseteq U$ dá definovať pomocou zjednotenia elementárnych množín, voláme ju *množina definovateľná* (v S), alebo *precízna množina*. V opačnom prípade je *hrubou množinou*. Pre hrubú množinu X však môžeme získať dve definovateľné množiny, ktoré ju charakterizujú, nazývané dolná a horná aproximácia:

$$B_*(X) = \bigcup_{x \in U} \{B(x) \subseteq X\},$$

$$B^*(X) = \bigcup_{x \in U} \{B(x) \cap X \neq \emptyset\}.$$

Dolnou aproximáciou B_* teda rozumieme zjednotenie všetkých elementárnych množín, ktoré sú podmnožinami X , a hornou aproximáciou zjednotenie všetkých elementárnych množín, ktorých prvky sa nachádzajú v X . Horná a dolná aproximácia potom určujú aj hraničnú oblasť $BN_B(X) = B^* - B_*$. Pre definovateľnú množinu platí $BN_B(X) = \emptyset$. [15]

Ak rozdelíme pôvodnú množinu atribútov A na dve množiny C, D , kde $C \cap D = \emptyset$, dostaneme informačný systém $S = (U, C, D)$, kde C značí množinu *podmienkových atribútov* a D značí množinu *rozhodovacích atribútov*. Tento informačný systém tvorí rozhodovaciu tabuľku. Každý objekt univerza $x \in U$ potom určuje sekvenciu $c_1(x), c_2(x), \dots, c_n(x), d_1(x), d_2(x), \dots, d_m(x)$, kde $c_i \in C, d_i \in D$. Táto sekvencia sa nazýva *rozhodovacie pravidlo* $C \rightarrow_x D$. Tento vzťah tvorí riadok rozhodovacej tabuľky. *Podporou* budeme nazývať počet ekvivalentných prvkov vzhľadom na dané atribúty, teda $supp_x(C, D) = |A(x)| = |C(x) \cap D(x)|$. *Silou*, rozhodovacieho pravidla $C \rightarrow_x D$ potom bude

$$\sigma_x(C, D) = \frac{supp_x(C, D)}{|U|}.$$

Veľmi dôležitou informáciou o pravidle je tiež *faktor dôvery*. Ten určuje pravdepodobnosť, že prvok y patrí do $D(x)$, ak vieme že patrí do $C(x)$ a je definovaný ako:

$$cer_x(C, D) = \frac{supp_x(C, D)}{|C(x)|} = \frac{\sigma_x(C, D)}{\pi(C(x))}, \pi(C(x)) = \frac{|C(x)|}{|U|}.$$

Ak pre dané pravidlo je faktor dôvery rovný 1, pravidlo je *isté*. Pravidlo, pre ktorého faktor dôvery platí $0 < cer_x(C, D) < 1$, voláme *neisté*. Pre isté pravidlo teda platí, že v rozhodovacej tabuľke existuje pre danú sadu hodnôt podmienkových atribútov jediná sada rozhodovacích atribútov.

Pre predstavu, akú veľkú časť pravidiel s rovnakými hodnotami rozhodovacích atribútov pokrýva dané pravidlo slúži *koeficient pokrytia*:

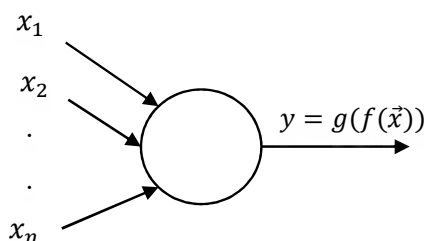
$$cov_x(C, D) = \frac{supp_x(C, D)}{|D(x)|}.$$

Z ľubovoľnej rozhodovacej tabuľky je možné získať rozhodovacie pravidlá v tvare *ak ...potom ...*, pre teóriu hrubých množín teda pravidlá *ak* $C(x)$ *potom* $D(x)$. Ak tieto pravidlá zachovávajú vzťahy nerozlišiteľnosti a pokrývajú všetky skutočnosti určené informačným systémom, jedná sa o rozhodovací algoritmus. [15]

1.2 Neurónové siete

Umelé neurónové siete [2] sú systémy vychádzajúce z princípov, akými biologické neurónové systémy (mozog) spracúvajú informácie. Podobne ako biologické neurónové siete, aj umelé siete sa skladajú z veľkého počtu jednoduchých elementov, neurónov, ktoré sú navzájom husto poprepájané. Umelé neurónové siete sa snažia vyriešiť problém s klasickými výpočtovými paradigmami, ktoré nie sú schopné učenia v dostatočnej miere a zvyčajne majú len veľmi malú mieru tolerancie na chyby. Súčasné použitie umelých neurónových sietí zahŕňajú úlohy ako klasifikácia, zhukovanie, rozpoznávanie vzorov, asociácia, alebo predikcia.

Chovanie neurónu je veľmi jednoduché a dá sa rozdeliť na niekoľko etáp. Najskôr neurón prijme vstupné dáta, spracuje ich pomocou bázevej funkcie, z ktorej následne pomocou aktivačnej funkcie vyhodnotí výstup. Výber bázevej a aktivačnej funkcie závisí na type riešenej úlohy.



Obrázok 1: Neurón

Neurón získa hodnoty vstupov x_i . Každý z nich má priradenú hodnotu váhy w_i . Pomocou vstupov a ich váh je vypočítaná hodnota *bázevej funkcie* $f(\vec{x})$. Výber bázevej funkcie závisí od typu úlohy, no typickým príkladom je tzv. lineárna bázevá funkcia $u = \sum_{i=1}^n w_i x_i + \Theta$, kde Θ značí *prah*¹. Ďalšou známou bázevou funkciou je tiež radiálna bázevá funkcia, ktorá sa počíta ako $u = \sqrt{\sum_{i=1}^n (x_i - w_i)^2}$. Hodnota bázevej funkcie je potom použitá a výpočet výstupu neurónu, ktorý je určený *aktivačnou funkciou* $g(f(\vec{x}))$. Aktivačná funkcia je ohraničená a typicky jej výstup spadá do intervalov $< -1, 1 >$ alebo $< 0, 1 >$. Rovnako ako pri bázevej funkcii, aj výber aktivačnej funkcie závisí na type problému. Medzi často používané patria

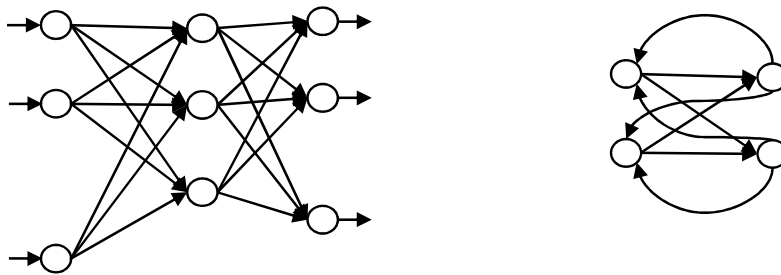
$$\begin{aligned} \text{skoková funkcia } f(u) &= \begin{cases} 1; & u \geq 0 \\ 0; & u < 0 \end{cases}, \\ \text{po častiach lineárna funkcia } f(u) &= \begin{cases} 0; & u < 0 \\ u; & 0 \leq u \leq 1 \\ 1; & u \geq 1 \end{cases}, \end{aligned}$$

¹ Častou úpravou je $u = \sum_{i=0}^n w_i x_i$, kde w_0 a x_0 nahrádzujú prah

$$\text{sigmolda } f(u) = \frac{1}{1+e^{-x}},$$

a iné.

Hoci už samotný neurón dokáže riešiť niektoré jednoduché problémy, sila neurónových sietí je práve v použití veľkého počtu neurónov na riešenie zložitých problémov. Jednotlivé neuróny sú medzi sebou pospájané väzbami, kde výstup jedného neurónu je vstupom ďalších. Usporiadanie a zapojenie neurónov sa nazýva *topológia siete*. Topológia je ďalší dôležitý faktor pri výbere neurónovej siete vzhľadom na použitie. Siete sa podľa topológie delia na plne prepojené siete, kde každý neurón je spojený s každým, alebo dopredné siete, kde neuróny sú usporiadané do vrstiev a dáta môžu prúdiť smerom od nižšej vrstvy ku vyššej (prípade medzi neurónmi vrstvy). Medzi známejšie patria siete ako *Madaline*, *Kohonenove neurónové siete/mapy*, *Hopfieldova sieť* a iné.



Obrázok 2: Dopredná sieť(vľavo) a plne prepojená sieť(vpravo)

Pred vlastným použitím neurónovej siete je najskôr nutné ju nakonfigurovať. Konfigurácia siete prebieha pomocou tzv. *učenia*. Učenie pracuje na princípe úprav váh vstupov neurónov. Existujú dve pravidlá, pomocou ktorých sa váhy upravujú: *Hebbovo pravidlo*, a *delta pravidlo*. V závislosti na tom, či sa sieť učí so znalosťou požadovaného výsledku sa delia siete na tri typy: *učenie bez učiteľa*, *posilované učenie* a *učenie s učiteľom*. Učenie bez učiteľa je asi najbližšie ku biologickému algoritmu, ale nemusí byť vhodné pre všetky problémy. Na vstup siete sú privádzané vstupné vzory, ktoré sa sieť snaží sama identifikovať a naučiť sa medzi nimi rozlišovať. Pri učení s učiteľom sa trénovacia množina skladá z trénovacích vstupov a ich správnych výsledkov. Pri každom trénovacom vstupe teda sieť vie zistiť chybu vo výstupe a na základe toho upraviť váhy. Nakoniec posilované učenie je kombináciou oboch predchádzajúcich možností, teda vyhodnotenie či výstupy siete boli správne alebo nie sa sieť dozvie až po spracovaní určitého počtu vstupov.

2 Hrubé neurónové siete

Hoci teória hrubých množín a umelé neurónové siete majú potenciál navzájom potlačiť svoje nedostatky, spôsob akým ich efektívne skombinovať je stále otvorenou otázkou. Hrubé neurónové siete sú však stále mladým oborom a objavilo sa mnoho úspešných algoritmov. Súčasný metódy je možné rozdeliť do štyroch základných kategórií: neurónové siete s predspracovaním dát založenom na hrubých množinách, neurónové siete založené na logike hrubých množín, neurónové siete s použitím hrubých neurónov a hrubé granulózne neurónové siete. [2]

2.1 Neurónové siete s predspracovaním údajov hrubými množinami

Zložitosť problémov riešených umelými neurónovými sieťami narastá, čo so sebou prináša narastajúcu časovú a výpočtovú náročnosť pri ich učení aj vlastnom použití, a taktiež sa podpisuje nad zníženou presnosťou neurónových sietí. Jednou z možností ako túto prekážku prekonať je predspracovanie vstupných dát. Hrubé množiny, ktoré sa osvedčili aj ako efektívny nástroj na analýzu vlastností atribútov dát, sa ukazujú ako vhodný nástroj na predspracovanie dát a to redukciou redundantných či inak nepodstatných atribútov a tým znížiť dimenzionalitu dát. Predspracovávanie dát hrubými množinami spočíva v analýze dát, na základe ktorej je vytvorená rozhodovacia tabuľka, pomocou ktorej sú analyzované vzťahy medzi podmienkovými a rozhodovacími atribútmi, a na základe ktorej sa z pôvodnej množiny vyberú iba nutne potrebné atribúty. Takto zjednodušené dáta sú potom spracované pomocou štandardnej neurónovej siete, ktorá však dosahuje menšej zložitosti. [2]

Cieľom predspracovania je dosiahnuť čo najmenšiu množinu atribútov, ktorá ale stále uchováva charakteristiky pôvodných dát a to čo najefektívnejším spôsobom. To viedlo k vzniku niekoľkých oblastí záujmu. Medzi tie hlavné patria: redukcia atribútov na základe matice rozlíšiteľnosti, redukcia na základe pozitívnej oblasti, redukcia na základe informačného pohľadu, redukcia na základe granulózných výpočtov a ďalšie. [3]

2.1.1 Redukcia atribútov na základe matice rozlíšiteľnosti

Matica nerozlíšiteľnosti [13] je jeden z najskúmanejších prístupov k redukcii atribútov pomocou teórie hrubých množín. Vychádza z rozhodovacej tabuľky $S = (U, C, D)$ a pracuje na princípe porovnávania jednotlivých objektov univerza a ich atribútov, kde hľadá rozdielne hodnoty.

Pozitívnu oblasťou rozumieme $POS_C(D) = \bigcup_{X \in U/D} C_*(X)$, teda zjednotenie dolných aproximácií (podľa podmienkových atribútov) ekvivalenčných tried určených rozhodovacími atribútmi D . Atribút $c \in C$ je postradateľný, ak $POS_C(D) = POS_{C-\{c\}}(D)$, inak je nepostradateľný. Teda postradateľný atribút je taký, že ak ho odstránime, tak pozitívny región ostane nezmenený. Následne,

ak sú všetky atribúty $c \in C$ nepostradateľné v danom informačnom systéme S , tento systém voláme *nezávislým*. Podmnožina podmienkových atribútov $R \subset C$ je *redukciou* C , ak platí že systém $S' = (U, R, D)$ je *nezávislý*. *Jadrom* je potom množina všetkých atribútov nepostradateľných v S : $CORE(C) = \cap RED(C)$, kde $RED(C)$ je množina všetkých redukcií. Maticou rozlíšiteľnosti M systému S potom bude matica o $n \times n$ prvkoch definovaná ako

$$m_{ij} = \begin{cases} \{c \in C: c(u_i) \neq c(u_j)\} & \text{ak } \exists d \in D: d(u_i) \neq d(u_j) \\ \emptyset & \text{inak} \end{cases}$$

Prvky matice sú teda množiny podmienkových atribútov C , ktoré rozlišujú objekty u_i a u_j univerza U do rôznych tried určených rozhodovacími atribútmi D .

Redukcia atribútov potom funguje nasledovne:

1. Najskôr sa vypočítajú všetky hodnoty matice nerozlišiteľnosti m_{ij} . Keďže matica je symetrická, stačí vypočítať hodnoty pod uhlopriečkou
2. Vytvorí sa logický výraz $f_S = \bigwedge_j \{ \bigvee m_{ij} \}, \forall j \in n, i \in \{1..j-1\}$, teda konjunkcia disjunkcií atribútov v množine na danej pozícii matice. Hodnoty i a j sú obmedzené, aby výpočet prebiehal len pod uhlopriečkou matice.
3. Vypočíta sa *normálna disjunktná forma*² získaného výrazu, kde jednotlivé disjunktné výrazy sú redukciami. Ich množina je $RED(C)$. Ich prienikom je tiež možné získať $CORE(C)$.

Jadro obsahuje najkritickejšie atribúty, bez ktorých rozhodovanie nie je možné, zatiaľ čo ľubovoľná z redukcií má ekvivalentnú rozhodovaciu silu ku pôvodnej množine podmienkových atribútov.

Tento základný algoritmus má však niekoľko problémov, medzi ktoré patri jeho vysoká náročnosť pre väčšie kvantá dát a veľkým problémom je jeho neschopnosť vyrovnat' sa s chybnými dátami. Stal sa však základom ďalšieho vývoja v tomto smere. [3]

2.1.2 Redukcia na základe informačného pohľadu

Niektoré algoritmy sa zameriavajú na bližšiu analýzu dát a heuristický výber atribútov na základe ich dôležitosti pre daný informačný systém $S = (U, C, D)$. Typický postup spočíva v nájdení jadra $CORE(C)$ a následným postupným pridávaním jednotlivých atribútov podľa dôležitosti, až kým nemá táto redukovaná množina atribútov rovnakú klasifikačnú schopnosť ako pôvodná. Získanie jadra je možné dosiahnuť niekoľkými spôsobmi, typicky z matice nerozlišiteľnosti (uvedenej vyššie) alebo priamo z definície jadra. Po získaní jadra ostáva problémom spôsob, akým sa ďalšie atribúty vyberajú a pridávajú do výslednej množiny. Cieľom je dosiahnuť najvhodnejšiu redukciu, teda redukciu, ktorá obsahuje najmenej atribútov s čo najväčšou presnosťou. Výber vhodného atribútu typicky určuje heuristická funkcia. Heuristická funkcia sa teda stáva centrom výskumu v tejto triede algoritmov.

² Disjunktná normálna forma – forma zápisu logickej formule, ktorá je tvorená disjunkciou konjunkcií $(\bigvee_j (\bigwedge_i x_{ij}))$

Objavilo sa niekoľko konceptov heuristických funkcií, založených na : vzájomných informáciách³, podmienenej informačnej entropii⁴, rozhodovacej sile, pozitívnej oblasti a iných. [3]

Príklad takéhoto algoritmu je v [4]. Nech $S = (U, C, D)$ je informačný systém a $R \subset C$ a $a \in C$, potom je *dôležitosť atribútu* definovaná ako

$$\alpha_{C,D}(a) = 1 - \frac{\gamma(C-\{a\},D)}{\gamma(C,D)},$$

kde $\gamma(C, D)$ značí faktor dôvery. Dôležitosť atribútu môžeme považovať ako určenie chyby v prípade vypustenia tohoto atribútu. Medzi $POS_R(D)/I(\{R \cup D\})$, koeficientom pokrytia a dôležitosťou atribútu existuje závislosť, z ktorej táto heuristická funkcia vychádza. Tá vyberá ten atribút $b \in C$, ktorý po pridaní do množiny vybraných atribútov R , má najlepší (najväčší) pomer kardinality množiny $POS_{R \cup \{a\}}(D)$ oproti kardinalite množiny $POS_{R \cup \{b\}}(D)/I(R \cup \{b\}, D)$.

Výpočet množiny atribútov potom prebieha nasledovne:

Nech R je množina vybraných atribútov, L je množina nevybraných atribútov, $T(L)$ je potenčná množina množiny L , $T_i(L)$ je i – tý prvok $T(L)$ a δ je hranica prenosti. Počiatočné hodnoty sú $R = CORE(C)$, $k = 0$.

1. Nech $X = POS_R(D)$,
2. ak $k \geq \delta$, $k = \frac{|POS_C(D)|}{|U|}$, alebo ak $POS_R(D) = POS_C(D)$, tak skonči,
3. nech $i = 1$, $flag = 0$, $Z = A = \emptyset$, $Y = T_i(L)$,
4. vyber náhodne $y \in Y$, $A = R \cup y$, ak platí $POS_A(D) = POS_C(D)$
 - a. ak $flag = 0$, potom $Z = A$, $flag = 1$, inak
 - b. ak platí $|U/Z| > |U/A|$, potom $Z = A$,
5. Ak $flag = 1$, potom vypočítaj $V_y = |POS_{R \cup \{y\}}(D)|$, $M_p = \max_card(POS_{R \cup \{y\}}(D)/I(\{R \cup \{y\} \cup D\}))$ (maximálna kardinalita) pre všetky $y \in Y$,
6. Vyber najlepší atribút $y \in Y$, teda atribút s najväčšou hodnotou $V_y \times M_p$, nastav $Y = Y - \{y\}$, $R = R \cup \{y\}$
7. Nastav $i = i + 1$ a pokračuj od kroku 1

Oproti väčšine algoritmov, uvedený algoritmus predstavuje lepšiu časovú a priestorovú zložitosť, vďaka cielenému výberu atribútov [4]. S vysokou úspešnosťou sa mu tiež darí získať optimálnu množinu atribútov. K podobným patria aj algoritmy CEBARKNC a CEBARKCC. Tie sa však navyše snažia aj vyrovnať s možnými chybnými dátami a nimi spôsobenými inkonzistenciami v rozhodovacej tabuľke. Objavil sa aj algoritmus, ktorý nepočíta jadro rozhodovacích atribútov, ale redukuje atribúty priamo z pôvodnej sady podmienkových atribútov. [3]

³ Angl. *Mutual information* – miera vzájomnej závislosti premenných

⁴ Angl. *Conditional information entropy* – miera informácií potrebných pre predpoklad hodnoty náhodnej premennej

2.1.3 Ďalšie typy predspracovania

Algoritmy predspracovania dát, založené na čistej teórii hrubých množín, sa potýkajú s problémom efektivity pri extrémne rozmerných dátach, keďže často počítajú s tabuľkami veľkých rozmerov. Od toho sa odvíja vývoj ďalších, efektívnejších, a mnohých hybridných algoritmov, kombinujúcich iné technológie a teórie. Podobne ako vyššie uvedené typy algoritmov, ďalšie možné riešenie vychádzajúce z oblasti hrubých množín sa zameriava na analýzu pozitívnej oblasti. Vzniknuté algoritmy ukazujú, že pomocou zoradenia množiny podmienkových atribútov C , sa výrazne znížila aj výpočetná náročnosť algoritmu. Na tomto fakte bolo preskúmaných niekoľko radiácií algoritmov pre atribúty, medzi nimi aj *quick-sort*, či *radix-sort*. [3]

Ďalšie skúmané algoritmy sa snažia vylepšiť efektivitu algoritmu zabudovaním ďalších myšlienok a overených algoritmov z iných oblastí, ako napríklad použitím granulárnych výpočtov. Na kombinácií týchto myšlienok sa podarilo vytvoriť aj algoritmus, ktorý je efektívnejší ako predchádzajúce, pričom dokáže získať minimálnu redukciu atribútov, teda najmenšiu možnú množinu ktorá stále zachováva charakter pôvodných dát. Iné typy algoritmov vznikli na základe kombinácie s genetickými algoritmami, s fuzzy množinami, či využitím paralelizácie. Okrem snáh o spojenie rôznych teórií sa objavujú aj zaujímavé spojenia s rôznymi technológiami, kam patria aj algoritmy, ktoré náročné výpočty nad dátami prenášajú do databázovej technológie. Úspešne sa podarilo navrhnúť algoritmy v jazyku SQL, ktoré priniesli zrýchlenie výpočtov vďaka optimalizácii databáz pre prácu s rozmernými dátami. [3]

2.2 Neurónové siete s hrubým neurónom

O spojenie neurónových sietí a hrubých množín vrámci neurónu sa pokúsil Pawan Lingras v štúdiu [5]. Inšpirácia pre vytvorený *hrubý neurón* pochádza z aproximácií poskytovaných hrubými množinami. Takýto neurón nepracuje s jednou hodnotou, ale s rozsahom (intervalom).

Takto definovaný hrubý neurón je možné navonok chápať ako dva obyčajné neuróny, teda má dva vstupy a výstupy. Jeden neurón predstavuje hornú aproximáciu, druhý dolnú aproximáciu. Vnútročné správanie neurónu sa od obyčajného neurónu líši. Vstupy fungujú podobne ako pri klasických neurónoch, teda pre lineárnu bázovú funkciu je hodnota:

$$u_i = \sum_j w_{ji} * x_j,$$

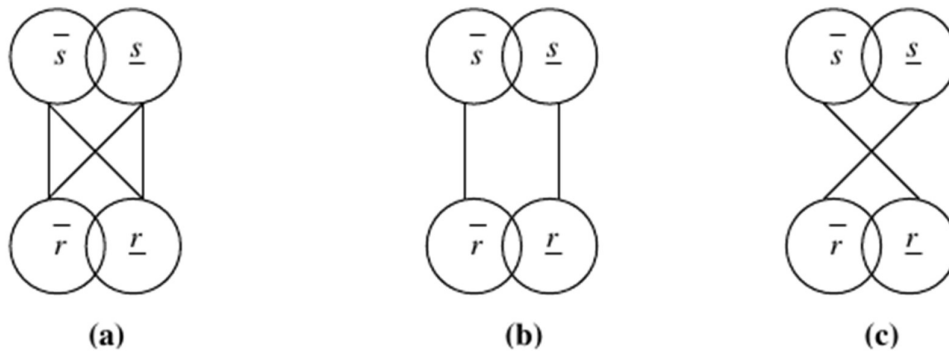
kde x_j značí hodnotu vstupu, w_{ij} značí jeho váhu. Rozdiel je však v tom, že časti neurónu majú vlastné vstupy. Výstupné hodnoty jednotlivých častí neurónu sa vypočítajú:

$$output_{\bar{r}} = \max(f(input_{\bar{r}}), f(input_{\underline{r}})),$$

$$output_{\underline{r}} = \min(f(input_{\bar{r}}), f(input_{\underline{r}})),$$

kde f je aktivačná funkcia.

Dva vstupy a výstupy hrubého neurónu prinášajú nové možnosti ako môžu byť jednotlivé hrubé neuróny navzájom poprepájané: dva hrubé neuróny r a s sú *plne prepojené*, ak výstupy hornej aproximácie neurónu $\bar{r}$ a dolnej aproximácie $\underline{r}$, sú obe spojené s oboma vstupmi neurónu s , $\bar{s}$ a $\underline{s}$. Ak je spojený výstup $\bar{r}$ s vstupom $\bar{s}$, a výstup $\underline{r}$ so vstupom $\underline{s}$, tak neurón r *budí* neurón s . Ak je zapojenie opačné, teda výstup $\bar{r}$ je spojený s vstupom $\underline{s}$, a výstup $\underline{r}$ so vstupom $\bar{s}$, tak neurón r *inhibuje* neurón s . Rôzne možné zapojenia sú znázornené na obrázku 3. Hrubé neuróny nie sú však limitované na spojenie len s ďalšími neurónmi rovnakého typu. Numerický charakter ich vstupov a výstupov umožňuje ich prepojenie aj s neurónmi štandardného typu.



Obrázok 3: Hrubý neurón, (a) plne prepojený, (b) r budí s , (c) r inhibuje s

Použitie hrubých neurónov nie je limitované na žiadny konkrétny druh neurónovej siete. Takýto neurón by malo byť možné použiť v ľubovoľnej topológii, pri problémoch, ktoré si vyžadujú vysporiadanie sa s dátami s nepresným charakterom, ktoré sa dajú reprezentovať ako interval, napríklad ak nie je možné zaznamenávať presné hodnoty.

Pri testovaní neurónových sietí s hrubým neurónom Pawan Lingras navrhol niekoľko kontrolných sietí. Cieľom bola sieť schopná predpovedať premávku na diaľnici v konkrétnom čase. Ako referenčná sieť slúžila trojvrstvová dopredná sieť (vstupná vrstva, jedna skrytá vrstva, výstupná vrstva), s použitím výlučne štandardných neurónov. Prvá experimentálna sieť vychádzala z návrhu referenčnej siete, ale neuróny vstupnej vrstvy boli vymenené za hrubé neuróny. Druhá experimentálna sieť nahradila neuróny vstupnej a skrytej vrstvy hrubými neurónmi. Výsledky ukázali, že prvá experimentálna sieť mala polovičnú chybovosť oproti referenčnému návrhu, a druhá experimentálna sieť tiež vykazovala slabšie známky zlepšenia oproti prvej, avšak minimálne. V štúdií [19] bol koncept hrubého neurónu znova overený, na neurónovej sieti určenej na diagnostiku chýb. Opäť sa porovnáva

tradičná neurónová sieť s neurónovou sieťou s hrubými neurónmi, a následne so sieťou s hrubými neurónmi a zredukovanou množinou vstupov (atribútov) pomocou predspracovania dát. Oproti štandardnej sieti sa neurónová sieť s hrubými neurónmi opäť ukazuje s lepším časom učenia a presnosťou. Predspracovanie dát pre sieť s hrubými neurónmi malo za následok mierne zvýšenie času učenia, avšak presnosť siete sa opäť zvýšila. Tento typ siete však nezaznamenal žiadny ďalší významný vývoj. Za následovníka tohoto konceptu je vnímaná aproximačná neurónová sieť uvedená neskôr [10].

2.3 Neurónové siete založené na hrubej logike

Neurónové siete založené na hrubej logike predstavujú ďalší spôsob kombinácie teórie hrubých množín a umelých neurónových sietí. Predchádzajúce prístupy sa venovali zjednodušeniu dát pred použitím vlastnej neurónovej siete, alebo zapracovanie hrubých množín do samotného neurónu, obe použiteľné na ľubovoľný typ úlohy. Neurónové siete s hrubou logikou sú určené na rozpoznávanie vzorov, a predstavujú koncept neurónovej siete, s upravenou topológiou a metódou učenia, inšpirovanými teóriou hrubých množín. V článku [17] autori porovnali ekvivalentnú doprednú sieť s učením backpropagation s navrhnutou sieťou s hrubou logikou na príklade rozpoznávania zemského povrchu, kde dosiahla súčasne vyššiu rýchlosť konverencie, kratší čas učenia a vyššiu presnosť.

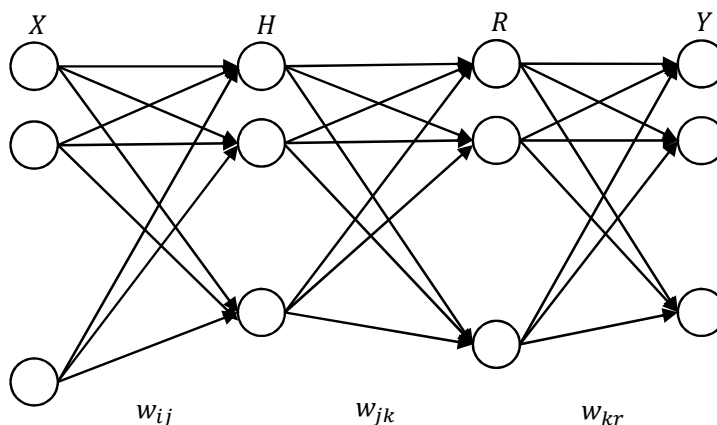
Návrh neurónovej siete s hrubou logikou je popísaný v článku [18]. Skladá sa z troch hlavných krokov:

1. Je potrebné vybrať vhodné rozhodovacie atribúty pre podmienky a rozhodnutia. Pred ich analýzou hrubými množinami je potrebné tieto hodnoty diskretizovať vhodnou metódou.⁵
2. Vytvorí sa rozhodovacia tabuľka, ktorá odráža vzťah medzi podmienkovými a rozhodovacími atribútmi. Tá sa následne redukuje a extrahujú sa rozhodovacie pravidlá.
3. Na základe informácií získaných z predchádzajúcich krokov je vytvorená neurónová sieť.

Prvé dva kroky predstavujú spracovanie dát pomocou teórie hrubých množín. Až v treťom kroku dochádza k tvorbe neurónovej siete na základe získaných informácií.

Topológiou sa jedná o štandardnú doprednú sieť so štyrmi vrstvami (obrázok 4), kde jednotlivé vrstvy reprezentujú špecifickú časť spracovania.

⁵ Možno pomocou diskretizácie na ekvivalentné intervaly, Booleovej algebry a inými



Obrázok 4: Topológia siete s hrubou logikou

Prvá vrstva (X) reprezentuje vstupný vektor, teda počet neurónov zodpovedá počtu vstupných hodnôt. Druhá vrstva (H) je typickou skrytou vrstvou, ktorej význam je v zvýšení presnosti a rýchlosti konvergencie siete. Výstup je definovaný ako $H_j = f_1(\sum_{i=1}^m w_{ij}X_i + \alpha_j)$, $j = 1 \dots q$, kde q je počet neurónov vrstvy a f_1 je aktivačná funkcia. Počet neurónov je voliteľný. Tretia vrstva (R) reprezentuje pravidlá. Počet neurónov odpovedá počtu pravidiel, a každý neurón reprezentuje práve jedno pravidlo. Výstup je definovaný ako $R_k = f_2(\sum_{j=1}^q w_{jk}H_j + \beta_k)$, $k = 1 \dots s$, kde s je počet neurónov vrstvy a f_2 je aktivačná funkcia. Posledná, výstupná vrstva reprezentuje rozhodnutie vyplývajúce z pravidiel. Počet neurónov vychádza z počtu rozhodovacích pravidiel vyplývajúcich z teórie množín. Výstup je definovaný ako $Y_r = f_3(\sum_{k=1}^s w_{kr}R_k + \theta_r)$, $r = 1 \dots n$, kde n je počet neurónov vrstvy.

Neurónová sieť je pri učení rozdelená na dve časti, ktoré sa učia postupne. Prvú časť predstavuje druhá a tretia vrstva, ktorých výstup predstavuje rozhodovacie pravidlá. Druhú časť predstavuje štvrtá vrstva. Jednotlivé časti siete je možné učiť štandardnými algoritmami, ako backpropagation. Pred učením neurónovej siete je potrebné vstupnú množinu dať najskôr vhodne pripraviť (diskretizovať hodnoty, vytvoriť rozhodovaciu tabuľku).

Učenie siete potom prebieha nasledovne:

1. Zvolí sa počet neurónov druhej vrstvy, inicializujú sa počiatočné hodnoty váh, zvolí sa maximálny počet tréningových cyklov, koeficient učenia a akceptovaná chyba neurónovej siete *errors*.
2. Vyberajú sa vstupy z tréningovej množiny a vypočíta sa odozva tretej vrstvy siete. V tomto kroku dochádza k učeniu siete. Nakoniec sa vypočíta rozdiel ϵ medzi skutočnou odozvou tretej vrstvy a očakávanou odozvou.

3. Pokiaľ je $\epsilon > error$ alebo nebol dosiahnutý maximálny počet tréningových cyklov, predchádzajúci krok sa opakuje. Inak je ukončené učenie prvej časti siete.
4. Opäť sa zvolí koeficient učenia, počet tréningových cyklov a maximálna akceptovaná chyba siete *errors*.
5. Na vstup štvrtej vrstvy sú vkladané očakávané výstupy tretej vrstvy na tréningové vstupy siete, vrstva sa učí podľa očakávaných výstupov siete a počíta sa rozdiel ϵ medzi očakávaným výstupom a skutočným.
6. Opakuje sa predchádzajúci krok, kým nie je splnené $\epsilon < errors$ alebo počet opakovaní nie je väčší ako zvolený limit. Po tomto kroku je neurónová sieť naučená. [18]

2.4 Hrubé granulárne neurónové siete

Neurónové siete sú typicky usporiadané na spracovávanie numerických dát a pracujú ako čierna skrinka. Tieto fakty majú za následok problematickú interpretáciu výsledku neurónových sietí v niektorých prípadoch, alebo už tiež spomínanú vysokú výpočtovú náročnosť. Poslednou kategóriou sietí ktoré sa snažia o spojenie s hrubými množinami aplikujú princíp *informačných (vedomostných) granúl*. [9]

Pojem informačnej granule môže byť popísaný ako kolekcia objektov, ktoré vykazujú podobné vlastnosti, teda sú nerozlíšiteľné. Granulácia informácií je proces bežný pri ľudskom uvažovaní, kedy komplexný problém je rozdelený na sadu menších a lepšie definovaných podproblémov (modulov), ktoré sú jednoduchšie spracovateľné ako pôvodný problém. Informačné granule a granulácia informácií poskytujú spôsob spracovania informácií, ktorého výsledkom sú zjednodušené koncepty pôvodných objektov. [9]

Informačné granule a granulácia informácií predstavujú iba abstraktný koncept a procesy, na základe ktorých vzniklo niekoľko odvodených formálnych modelov v rôznych matematických oblastiach, ako množiny, fuzzy množiny, pravdepodobnosť, či hrubé množiny. Na ich základe boli navrhnuté neurónové siete, označované spoločne ako granulárne neurónové siete. Doposiaľ zaznamenali najväčší rozvoj granulárne neurónové siete založené na princípe fuzzy množín. Hrubé množiny zatiaľ zaostávajú, pravdepodobne aj preto, že v porovnaní s inými riešeniami (fuzzy množiny) je hodnota patričnosti objektu do informačnej granule skoková, čo spôsobuje ostré vzťahy pre objekty na rozhraní niekoľkých granúl. Dôležitou vlastnosťou granulárnych neurónových sietí je veľkosť informačných granúl, *granularita informácií*. Tú je typicky možné vyjadriť ako počet objektov v informačnej granule. Čím väčší počet objektov je v informačnej granule, tým je granularita informácií menšia. Ak by sme si za príklad zobrali podmnožinu reálnych čísel $X \subset R$, najväčšiu granularitu by sme dosiahli, ak by sme každé číslo považovali za individuálnu informačnú granulu. Ak by sme jednotlivé čísla začali zoskupovať do intervalov, s rastúcou veľkosťou intervalu by klesala granularita informácií. V takomto prípade by sme mohli granularitu kvantifikovať ako $\frac{|A|}{|X|}$. Pre teóriu hrubých

množín môže byť granularita definovaná ako presnosť aproximácie, teda $\alpha(A) = \frac{|A_*|}{|A^*|}$, kde A_* je dolná aproximácia a A^* je horná aproximácia. [9]

Učenie granulárnej neurónovej siete pozostáva z dvoch hlavných fáz: vstupné dáta sú spracované a zhustené na informačné granule, ktoré sú potom v ďalšom kroku použité namiesto pôvodných dát ako vstup neurónovej siete. Vzhľadom na granularitu štruktúry siete, teda spojení medzi jednotlivými neurónmi, a granularitu vstupných dát siete, môžu nastať štyri rôzne kombinácie:

- Pri vysokej granularite siete a súčasne vysokej granularite dát je vytvorená sieť veľmi podobná klasickým neurónovým sieťam, ak by granularita dosahovala úroveň ako v prípade s reálnymi číslami vyššie, jednalo by sa o štandardnú neurónovú sieť.
- Ak by sme sieť trénovali pomocou pôvodných (numerických) dát, avšak chceli ju používať na dátach s vyššou granularitou, vzniknutá sieť by bola štandardnou neurónovou sieťou s granulárnymi vstupmi, kde je nutné predstaviť spôsob kódovania granulárnych informácií do pôvodného systému.
- Neurónovú sieť, navrhnutú s granulárnymi spojeniami medzi jednotlivými neurónmi, je stále možno použiť s dátami s vysokou granularitou (vrátane numerických dát), avšak výsledné dáta v tomto prípade budú vykazovať menšiu granularitu, určenú sieťou.
- Poslednou kombináciou sú siete, ktoré využívajú granulárnych spojení a pracujú s granulárnymi dátami.

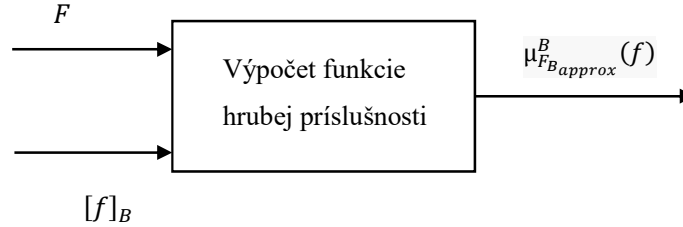
Prechod od štandardných neurónových sietí ku granulárnym neurónovým sieťam prináša so sebou určité problémy. Pri tvorbe takejto siete je treba dbať na výber vhodnej úrovne granularity ako aj na výber reprezentácie granulárnych dát. Medzi úspešné siete v tejto kategórii patrí *aproximačná neurónová sieť*. [2][9]

2.4.1 Aproximačná neurónová sieť

Aproximačná neurónová sieť, popísaná v [11] a [7], je považovaná tiež ako evolúcia neurónových sietí s hrubým neurónom, pretože aj tento druh siete predstavuje upravené neuróny. Tu sa však stretávame s dvoma novými typmi neurónov, ktorých správanie a komunikácia so zbytkom siete je veľmi odlišná od klasických sietí. Predstavuje dvojicu nových typov neurónov, *aproximačný neurón*, ktorý nahradzuje štandardné neuróny vstupnej vrstvy a *rozhodovací neurón*, ktorý nahradzuje neuróny výstupnej vrstvy.

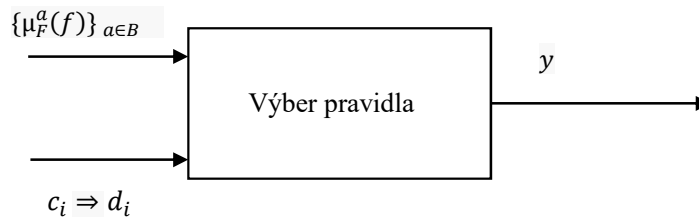
Nech B je množina atribútov, F je množina vstupov (archív predchádzajúcich vstupov neurónu, forma pamäte prístupná neurónu), $F_{B_{approx}}$ je množina aproximácie, a $[f]_B$ je ekvivalenčná trieda získaná zo známych objektov. Aproximačný neurón na základe týchto údajov počíta hodnotu *funkcie hrubej príslušnosti* množiny $F_{B_{approx}}$ do triedy do ekvivalenčnej triedy $[f]_B$. Pre informačný systém $S = (U, A)$, je funkcia hrubej príslušnosti hodnota, určujúca príslušnosť daného objektu $u \in U$ do danej

množiny X , vzhľadom na množinu atribútov $B \subseteq A$. Hodnota sa vypočíta ako $\mu_X^B(u) = \frac{|[u]_B \cap X|}{|[u]_B|}$, kde hodnota $[u]_B$ značí ekvivalenčnú triedu prvku u .



Obrázok 5: Aproximačný neurón

Rozhodovací neurón reprezentuje rozhodovaciu tabuľku pre pravidlá tvaru $c_i \Rightarrow d_i$, kde c_i je vektor hodnôt funkcie hrubej príslušnosti a hodnota $d_i \in \{0,1\}$. Na základe výstupov aproximačných neurónov sa vyberie to pravidlo, ktorého vektor c_i je najbližšie vstupnému vektoru c_{exp} . Na základe výťažného vektoru sa vypočíta hodnota relatívnej chyby $e_i = \|c_{exp} - c_i\| / \|c_i\|$, kde operácia $\| \cdot \|$ značí veľkosť vektoru. Pokiaľ je hodnota relatívnej chyby $e_i = 0$, potom výstupom rozhodovacieho neurónu je hodnota $AND(1 - e_i, d_i) = d_i$ a klasifikácia je úspešná. Ak však platí $e_i = 1$, potom je výstupom neurónu $AND(1 - e_i, d_i) = 0$, čo značí že klasifikácia nebola úspešná.



Obrázok 6: Rozhodovací neurón

Pomocou týchto neurónov je možné zložiť dva typy sietí. Prvou je čistá aproximačná sieť, ktorá sa skladá z jednej vrstvy aproximačných neurónov. Táto sieť sa ukázala ako vhodným nástroj na riešenie klasifikačných problémov, kde zmeny hodnôt atribútov v čase môžu byť rozpoznané pomocou hrubých aproximácií. Každý neurón predstavuje klasifikačnú triedu a výstupom je teda úroveň s akou vstup patrí do danej triedy. S použitím rozhodovacích neurónov je možné vytvoriť doprednú vrstvenú sieť, kde vstupná vrstva sa skladá z klasifikačných neurónov a výstupná vrstva je zložená z rozhodovacích neurónov. Pri oboch spomínaných sieťach je možné použiť na učenie štandardné metódy učenia. Oba typy sa taktiež osvedčili v aplikáciách ako klasifikácia porúch elektrickej siete. [11] [7]

3 Návrh a implementácia

Všetky predstavené algoritmy sú implementované v programovacom jazyku Java a jedná sa o konzolové aplikácie. Jazyk Java bol zvolený hlavne vďaka jednoduchšej prenositeľnosti zdrojového kódu medzi platformami.

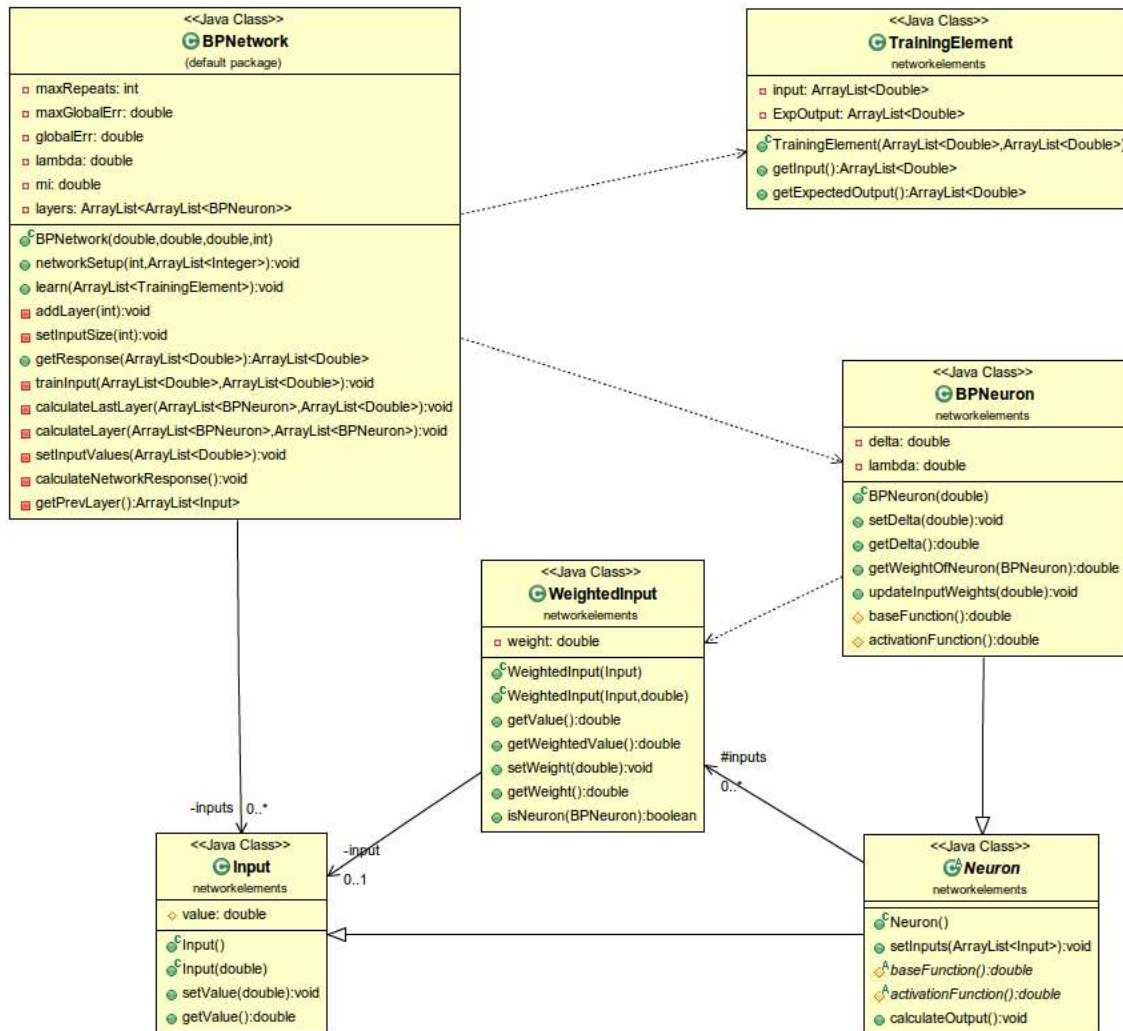
Navrhnutých a implementovaných bolo päť algoritmov. Prvým algoritmom je štandardná dopredná neurónová sieť, s neurónmi s lineárnou bázovou funkciou a sigmoidálnou aktivačnou funkciou, a s učením typu backpropagation. Tento algoritmus tvorí základ pri porovnaní s algoritmami založenými na kombinácii hrubých množín a neurónových sietí, ktoré boli predstavené v prechádzajúcej kapitole. Z nich boli implementované štyri riešenia, každé reprezentujúce jeden prístup. Druhým implementovaným algoritmom bol algoritmus na redukciu atribútov na základe matice rozlíšiteľnosti. Jeho vstupom je množina atribútov a ich hodnôt pre sadu vzorkov a výstupom je množina redukcii atribútov s rovnakou schopnosťou rozlíšenia daných vzorkov. Tretím implementovaným algoritmom je neurónová sieť s hrubými neurónmi. Návrh tejto siete je rozšírením návrhu štandardnej neurónovej siete o vstupy hrubého charakteru a hrubé neuróny. Štvrtým implementovaným algoritmom je algoritmus neurónovej siete s hrubou logikou. Rovnako ako predchádzajúci algoritmus, jej návrh opäť vychádza zo štandardnej neurónovej siete, ale dochádza ku úprave učenia, ktoré pri tejto sieti prebieha v dvoch krokoch. Posledným algoritmom je aproximačná neurónová sieť, zložená z aproximačných a rozhodovacích neurónov, ktorých správanie je výrazne odlišné od štandardne používaných neurónov a je založené na teórii hrubých množín.

3.1 Kontrolná neurónová sieť

Prvým implementovaným algoritmom bola štandardná dopredná neurónová sieť, slúžiaca ako základ pre porovnanie s ďalšími algoritmami, prípadne ako ich súčasť. Jedná sa o štandardnú neurónovú sieť. Ako bázová funkcia bola zvolená lineárna bázová funkcia. Za aktivačnú funkciu bola zvolená sigmoidálna funkcia. Pre učenie neurónovej siete bol vybraný algoritmus backpropagation.

Keďže táto sieť bola využitá v rôznych konfiguráciách a bola základom pre implementácie ďalších algoritmov, bola navrhnutá a implementovaná takým spôsobom, ktorý umožňuje jej jednoduchú úpravu alebo rozšírenie. Návrh tried priamo kopíruje myšlienku funkcie neurónovej siete. Celá sieť je reprezentovaná triedou `BPNetwork`, ktorá spravuje vrstvy siete, ich neuróny, počítanie odozvy a učenie. Každý neurón je reprezentovaný objektom triedy `BPNeuron`. Diagram tried je možné vidieť na obrázku 7.

Prvky tréningovej množiny sú reprezentované ako objekty typu `TrainingElement`. Objekty tohto typu obsahujú dva zoznamy hodnôt. Prvým sú vstupné hodnoty neurónovej siete a druhým očakávané výstupné hodnoty pre daný vstup. Vlastné vstupy neurónovej siete modeluje trieda `Input`.



Obrázok 7: Diagram tried kontrolnej neurónovej siete

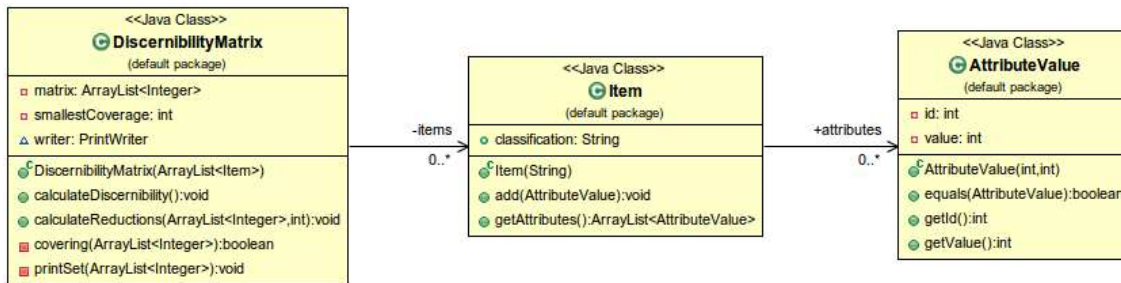
Objekty tejto triedy udržiavajú hodnotu vstupu a umožňujú ju získať alebo nastaviť novú hodnotu. Podtriedou triedy `Input` je abstraktná trieda `Neuron`. `Neuron` je abstraktnou triedou, ktorá obsahuje dodatočné metódy všeobecného neurónu, ako nastavenie vstupov neurónu, metódu na výpočet výstupu neurónu a abstraktné metódy pre báзовú a aktivačnú funkciu. Vstupy neurónu sa nastavujú metódou `setInputs`, ktorá prijíma zoznam prvkov triedy `Input`. Pre všetky prvky tohto zoznamu si vytvorí objekty triedy `WeightedInput`. `WeightedInput` okrem referencie na daný vstup ukladá aj hodnotu váhy vstupu a poskytuje metódy pre manipuláciu hodnoty váhy vstupu a získanie priamo váženej hodnoty vstupu $w_i * x_i$. Keďže `Neuron` je podtriedou triedy `Input`, nie je nutné rozlišovať, či vstupy neurónu sú vstupy siete triedy `Input` alebo neurón predchádzajúcej vrstvy. Okrem dodaných vstupov si `Neuron` do tohto zoznamu pridáva jeden dodatočný, ktorý obsahuje staticky hodnotu 1, ktorý reprezentuje prah báзовej funkcie neurónu. Ďalším rozšírením triedy `Neuron` je trieda `BPNeuron`.

Táto trieda obsahuje konkrétnu implementáciu bábovej funkcie (lineárna bábová funkcia), a aktivačnej funkcie (sigmoidálna aktivačná funkcia), a pre učiaci algoritmus backpropagation pridáva hodnotu δ , metódy pre jej získanie a zmenu a ďalšie metódy na aktualizáciu váh pre jednotlivé vstupy neurónu.

Hlavnou triedou algoritmu je trieda `BPNetwork`. Táto trieda spravuje celú inicializáciu, učenie a počítanie výstupov neurónovej siete. Inicializácia siete spočíva v nastavení parametrov siete a nastavení vstupov a vrstiev siete. Parametre siete sú predávané pomocou konštruktoru triedy a pozostávajú z hodnôt koeficientu učenia siete μ , koeficientu aktivačnej funkcie neurónu λ , maximálneho počtu opakovaní učenia *maxRepeats* a maximálnej prípustnej hodnoty globálnej chyby učenia *maxGlobalErr*. Pomocou metódy `networkSetup` neurónová sieť vytvorí požadovaný počet vstupov, vrstvy siete o požadovaných veľkostiach a nastaví vstupy neurónov. Vstupy sú tvorené zoznamov objektov triedy `Input`. Vrstvy neurónov sú uložené ako zoznam zoznamov neurónov, kde posledná vrstva je súčasne výstupnou vrstvou. Neurónová sieť sa učí metódou `learn`, ktorá ako parameter prijíma zoznam objektov triedy `TrainingElement`. Trénovanie siete prebieha pomocou algoritmu backpropagation: pre všetky prvky trénovacej množiny sietí nastaví dané hodnoty vstupov, následne vypočíta odozvu siete. Z očakávanej odozvy siete a skutočnej odozvy vypočíta lokálnu chybu. Z nej sa postupne vypočíta hodnota δ pre všetky neuróny poslednej vrstvy, a táto chyba sa postupne šíri smerom k prvej vrstve. Po vypočítaní hodnoty δ pre všetky neuróny všetkých vrstiev sa s jej pomocou aktualizujú hodnoty váh vstupov jednotlivých neurónov. Tento proces sa opakuje cyklicky, pokiaľ sieť buď dosiahne maximálny počet opakovaní učenia *maxRepeats*, alebo keď hodnota globálnej chyby klesne pod zadanú hodnotu *maxGlobalErr*.

3.2 Redukcia atribútov maticou rozlíšiteľnosti

Algoritmus na redukciu atribútov pomocou matice rozlíšiteľnosti neslúži priamo na vyhodnocovanie dát, ale len na ich predspracovanie, na optimalizáciu počtu vstupov pre neurónovú sieť. Jeho vstupom je množina vzorov a ich známych tried a jeho úlohou je vypočítať maticu rozlíšiteľnosti, na základe ktorej získa redukované množiny atribútov, ktoré stále dokážu rozlíšiť prvky množiny vstupov do daných tried. Návrh tohto algoritmu používa tri triedy: `Item`, `AttributeValue` a `DiscernibilityMatrix`. Diagram tried je na obrázku 8.



Obrázok 8: Diagram tried algoritmu matice rozlíšiteľnosti

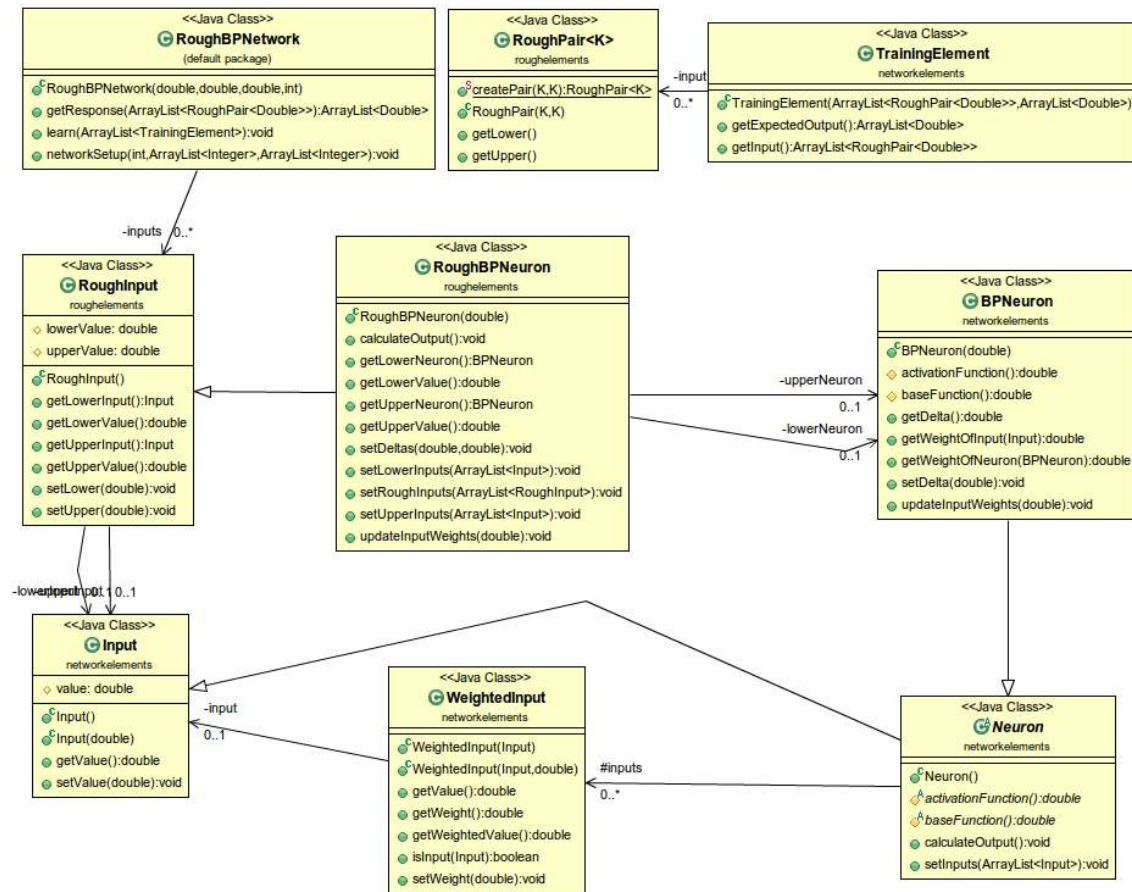
Množina vstupov pre algoritmus výpočtu matice rozlíšiteľnosti je uložená ako zoznam objektov triedy `Item`. Táto trieda obsahuje dva parametre: cieľovú triedu vzorku a hodnoty jeho atribútov. Cieľová trieda je uložená vo forme reťazca a hodnoty atribútov ako zoznam objektov triedy `AttributeValue`. `AttributeValue` je jednoduchá trieda, ktorá drží dve hodnoty: identifikátor atribútu a jeho hodnotu. Trieda `DiscernibilityMatrix` na prijíma v konštruktore ako parameter zoznam objektov `Item`. Na základe veľkosti tohto zoznamu vytvorí dvojdimenzionálnu maticu o $n \times n$ prvkoch, kde n je počet prvkov. Volaním metódy `calculateDiscernibility` algoritmus porovnáva hodnoty jednotlivých atribútov pre každé dva prvky. Ak sú dva prvky $i, j, i \neq j$ rovnakej triedy, uloží sa na dané miesto matice hodnota `null`, inak sa tam uloží zoznam identifikátorov atribútov, pre ktoré majú tieto prvky rozdielne hodnoty. Keďže matica je symetrická, počítajú sa iba prvky pod diagonálou. Po vypočítaní matice rozlíšiteľnosti je možné zavolať metódu `calculateReductions`, ktorá počíta všetky možné redukcie. Jej parametrom je zoznam identifikátorov atribútov. Keďže algoritmus pre výpočet disjunktnéj normálnej formy pre veľký počet atribútov je netriviálny, bol navrhnutý alternatívny algoritmus. Ten overí, či dodaná množina atribútov je schopná rozlíšiť všetky vzorky. Ak áno, tak túto množinu atribútov vypíše a postupne pre každý atribút vytvára nový zoznam atribútov, z ktorého odstráni práve jeden a volá túto metódu rekurzívne na nový zoznam. Ak nie, dané atribúty nie sú schopné pokryť vzorky a končí. Overenie, či daná množina atribútov je schopná rozlíšiť všetky dodané vzorky, prebieha metódou `covering`. Tá prechádza všetky miesta matice rozlíšiteľnosti pod diagonálou a testuje ich. Pokiaľ je miesto matice hodnoty `null`, teda prvky sú rovnakej triedy, tak miesto preskakuje, inak skúša či toto miesto obsahuje aspoň jeden z dodaných atribútov.

3.3 Neurónová sieť s hrubými neurónmi

Tretím implementovaným algoritmom je algoritmus neurónovej siete s hrubým neurónom. Návrh tohto algoritmu vychádza z predchádzajúceho návrhu neurónovej siete a teda použitá bazová a aktivačná funkcia je lineárna bazová funkcia a sigmoidálna aktivačná funkcia. Navrhnutá neurónová sieť pridáva nové prvky siete, ktorými sú vstupy siete hrubého charakteru a hrubé neuróny. Učenie prebieha opäť

pomocou algoritmu backpropagation, ktorý však bolo nutné upraviť tak, aby bol schopný pracovať s hrubými neurónmi. Diagram tried výsledného návrhu je možné vidieť na obrázku 9.

Keďže návrh rozširuje štandardnú pôvodnú neurónovú sieť, obsahuje niekoľko rovnakých komponent, ako `Input`, `Neuron`, `WeightedInput`. Pribudli však triedy `RoughPair`, ktorá modeluje hodnotu s hrubým charakterom, teda obsahuje hornú a dolnú hodnotu, a trieda `RoughInput`, ktorá reprezentuje hrubé vstupy siete. `RoughInput` obsahuje dve inštancie triedy `Input`, reprezentujúce hornú a dolnú hodnotu hrubého vstupu. Trieda `TrainingElement` bola mierne upravená a množina hodnôt vstupov siete je reprezentovaná zoznamom objektov triedy `RoughPair`. Ďalšou novou triedou je hrubý neurón, ktorý je modelovaný triedou `RoughNeuron`. `RoughNeuron` je podtriedou `RoughInput`, od ktorého dedí dve výstupné hodnoty. Funkciu hrubého neurónu tak ako je popísané v predchádzajúcej kapitole zabezpečuje dvojica štandardných neurónov `BPNeuron`, ktoré reprezentujú hornú a dolnú časť hrubého neurónu. Výstupné hodnoty hrubého neurónu sú určené výstupom týchto vnútorných neurónov, kde horný výstup hrubého neurónu je určený ako maximum výstupných hodnôt vnútorných neurónov a dolný výstup ako ich minimum.



Obrázok 9: Diagram tried neurónovej siete s hrubými neurónmi

Inicializácia, správa vrstiev, učenie a výpočet odozvy neurónovej siete sú opäť realizované pomocou vlastnej triedy `RoughBPNetwork`. Inicializácia neurónovej siete je podobná sieti `BPNetwork`. Konštruktor opäť prijíma parametre koeficientu učenia siete μ , koeficientu aktivačnej funkcie neurónu λ , maximálneho počtu opakovaní učenia *maxRepeats* a maximálnej prípustnej hodnoty globálnej chyby učenia *maxGlobalErr*. Inicializácia vstupov siete a vrstiev siete je však odlišná. Použité vstupy sú objekty triedy `RoughInput`. Následne sú vytvorené vrstvy s hrubými neurónmi o zadanych veľkostiach a posledné sú vrstvy s štandardnými neurónmi. Sieť musí obsahovať aspoň jednu vrstvu hrubých neurónov a aspoň jednu vrstvu štandardných neurónov. Vstupmi neurónov danej vrstvy sú neuróny predchádzajúcej vrstvy, prípadne pre prvú vrstvu sú to vstupy siete. Rovnako ako v článku [5], aj v tejto sieti bolo pre hrubé neuróny zvolené plné prepojenie a pre učenie bol zvolený algoritmus backpropagation. Ten však bolo nutné upraviť, keďže výstupná horná a dolná hodnota hrubého neurónu priamo nenaväzuje na horný a dolný vnútorný neurón, ale sú funkciou ich výstupov (minimum, resp. maximum). Algoritmus šíri hodnotu δ cez horný a dolný výstup hrubého neurónu tomu vnútornému neurónu, ktorého hodnota bola do daného výstupu priradená. Učenie znova prebieha pomocou metódy `learn`, ktorá iteráciou cez objekty typu `TrainingElement` počíta odozvu a upravuje váhy siete, pokiaľ nedosiahne maximálny počet opakovaní alebo globálna chyba neklesne pod požadovanú hodnotu.

3.4 Neurónová sieť s hrubou logikou

Rovnako ako neurónová sieť s hrubými neurónmi, návrh neurónovej siete s hrubou logikou vychádza zo štandardnej neurónovej siete. Teda opäť je to dopredná neurónová sieť s lineárnou bázovou funkciou, sigmoidálnou aktivačnou funkciou a algoritmom učenia backpropagation, ale učenie je tentokrát rozdelené na dve časti.

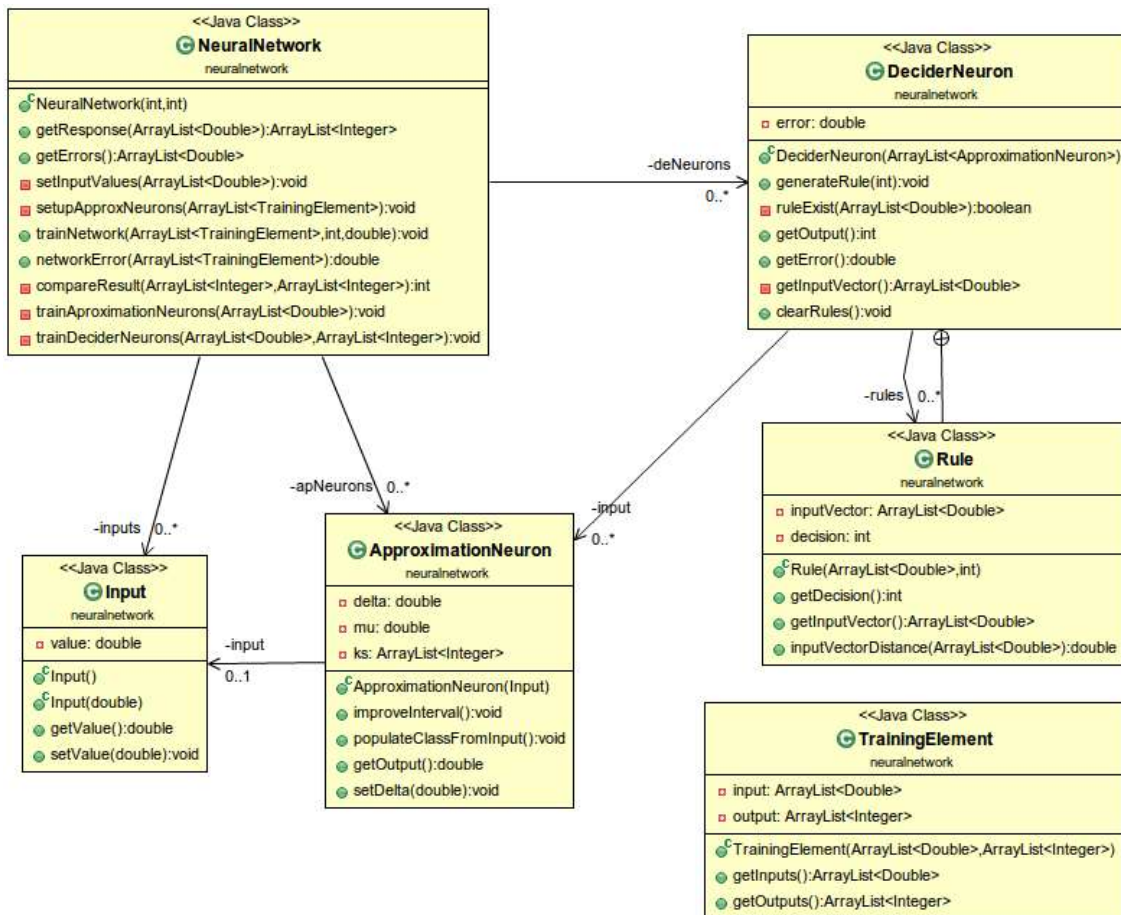
Sieť je implementovaná v triede `RoughLogicBPNetwork`, ktorej konštruktor opäť prijíma parametre koeficientu učenia siete μ , koeficientu aktivačnej funkcie neurónu λ , maximálneho počtu opakovaní učenia *maxRepeats* a maximálnej prípustnej hodnoty globálnej chyby učenia *maxGlobalErr*. Keďže neurónová sieť s hrubou logikou má pevný počet vrstiev, metóda `networkSetup` prijíma parametre počtu vstupov siete, počtu neurónov prvej vrstvy, ktorý je ľubovoľný, počtu neurónov druhej vrstvy, ktorých počet je daný počtom rozhodovacích pravidiel a počtu neurónov tretej (výstupnej) vrstvy. Sieť sa učí v dvoch krokoch, pomocou dvoch metód. Metóda `learnRules` prijíma zoznam tréningových vstupov triedy `TrainingElement`, ktoré obsahujú vstup neurónovej siete a kódovanie pravidla, teda očakávaný výstup druhej vrstvy. Na tejto množine vstupov učí sieť prvé dve vrstvy. Po dokončení učenia dosiahnutím požadovanej maximálnej hodnoty globálnej chyby alebo maximálneho opakovania je druhá vrstva naučená. Metódou `learnDecisions` je potom učená posledná vrstva siete. Parametrom tejto metódy je taktiež zoznam objektov triedy `TrainingElement`. Vstupné hodnoty

reprezentujú výstup druhej vrstvy, teda kódujú aktívne pravidlo, a výstupné hodnoty sú výstupné hodnoty tretej vrstvy a zároveň celej siete.

3.5 Aproximačná neurónová sieť

Posledným implementovaným algoritmom bola aproximačná neurónová sieť. Táto sieť využíva dvoch typov neurónov: aproximačný a rozhodovací neurón. Topológiou sa jedná o dvojvrstvú doprednú sieť, ktorej prvá vrstva je zložená z aproximačných neurónov a druhá z rozhodovacích neurónov. Žiadny z článkov [5] a [19], pojednávajúcich o aproximačných sieťach nepredstavuje algoritmus pre vlastné učenie siete, preto bol v tejto implementácii použitý algoritmus vlastného návrhu. Diagram tried je na obrázku 10.

Základnými stavebnými blokmi siete sú jej vstupy a neuróny. Vstupy sú opäť modelované triedou `Input`, ktorá udržiava nastavenú hodnotu a umožňuje jej zmenu. Chovanie aproximačného neurónu je obsiahnuté v triede `ApproximationNeuron`. Navrhnutý neurón pracuje len s jedným



Obrázok 10: Diagram tried algoritmu pre aproximačnú neurónovú sieť

vstupom. Konkrétna implementácia je založená na návrhu aproximačného neurónu na relácii nerozlišiteľnosti predstaveného v článku [12]. Takýto aproximačný neurón je potom schopný pracovať s reálnymi hodnotami, pomocou diskretizácie do intervalov o dĺžke δ . Tento parameter taktiež nahrádza váhy vstupov typické pre klasické neuróny. Horná aproximácia $\bar{BF}$, potrebná pre funkciu hrubej príslušnosti, je reprezentovaná zoznamom hodnôt ks . Ten obsahuje zoznam celých čísel k takých, pre ktoré interval $< k\delta, (k + 1)\delta >$ patrí do $\bar{BF}$. Učenie neurónu prebieha pri volaní metódy `improveInterval`, zmenšovaním hodnoty δ na hodnotu $\delta * (1 - \mu)$, kde μ je koeficient učenia neurónu. Po zmene intervalu je potrebné znova vytvoriť zoznam ks , pomocou metódy `populateClassFromInput`, ktorá pre danú vstupnú hodnotu získa interval a uloží ho do pamäte, pokiaľ sa tam už nenachádza. Metóda `getOutput` vracia výstup aproximačného neurónu a nadobúda hodnotu 0 alebo 1, podľa toho či zadaný vstup neurónu patrí do jedného z intervalov určených ks . Trieda `DeciderNeuron` zapuzdruje správanie rozhodovacieho neurónu. Jeho úlohou je z výstupov aproximačných neurónov určiť výstupnú hodnotu siete. Pravidlá rozhodovacieho neurónu sú v tvare $(a_1, a_2, a_3, \dots) \rightarrow d$, kde a_i je výstupná hodnota aproximačného neurónu a d je výstup rozhodovacieho neurónu pre dané pravidlo. Tieto pravidlá sú uložené ako objekty triedy `Rule`, ktorá uchováva hodnoty pravidiel a metódou `calculateVectorDistance` umožňuje výpočet vzdialenosti dodaného vektoru od vektoru vstupov, ktorý tvorí podmienku pravidla. Neurón umožňuje vytváranie nových pravidiel pomocou metódy `generateRule`, ktorá uloží nové pravidlo s aktuálnym vstupným vektorom a dodanou výstupnou hodnotou, alebo ich všetky zmazanie metódou `clearRules`. Výstupnú hodnotu vracia metóda `getOutput`, ktorá vráti hodnotu $y = d$ pre pravidlo s najmenšou vzdialenosťou podmienkového vektoru ku aktuálnemu vstupnému vektoru rozhodovacieho neurónu. Hodnotu vzdialenosti e_i vstupného vektoru od podmienkového vektoru pre posledný počítaný výstup neurónu je možné získať metódou `getError`. Navrhnutý výstup je teda rozdielny od neurónu predstaveného v [5]. Tam bol počítaný ako $y = \min(e_i, d)$. Ak by však pre toto pravidlo platilo $d = 1$, a hodnota e_i je nulová, výstup neurónu by bol 0, teda klasifikácia by bola označená za neúspešnú, aj keď opak je pravdou a pravidlo je s vysokou pravdepodobnosťou platné.

Prácu so sieťou, zahrňujúcu jej počiatočné nastavenie či učenie, spravuje trieda `NeuralNetwork`. Sieť je inicializovaná v konštruktore podľa parametrov počtu vstupov siete a počtu výstupov. Počet vstupov siete a aproximačných neurónov je rovnaký, kde každý aproximačný neurón dostane práve jeden vstup. Počet výstupov odpovedá počtu rozhodovacích neurónov, ktorých vstupmi sú aproximačné neuróny. Sieť sa učí metódou `trainNetwork`, ktorá pracuje so zoznamom objektov typu `TrainingElement`. Tie obsahujú hodnoty vstupov a očakávaných výstupov siete. Pri učení je najskôr určený interval δ aproximačných neurónov ako maximum vstupnej hodnoty daného atribútu. Potom sieť cyklicky pre všetky prvky tréningovej množiny najskôr vytvorí vo všetkých aproximačných neurónov hornú aproximáciu $\bar{BF}$ pre všetky pozitívne klasifikované prvky. Následne z tréningových

vstupov generuje pravidlá rozhodovacieho neurónu. Cyklus končí pokiaľ chyba siete klesla pod požadovanú úroveň alebo pokiaľ bol dosiahnutý maximálny počet opakovaní. Inak aproximačné neuróny znížia veľkosť intervalu δ , zabudnú hornú aproximáciu $\bar{B}F$, a rozhodovacie neuróny vymažú všetky pravidlá a cyklus sa opakuje.

4 Vyhodnotenie algoritmov

Algoritmy navrhnuté v prechádzajúcej kapitole boli implementované a následne aplikované na praktické úlohy, na ktorých boli porovnávané s kontrolným riešením. Keďže všetky prístupy ku kombináciám neurónových sietí s hrubými množinami spočívali v snahe zvýšiť efektivitu neurónových sietí ich úpravou na základe teórie hrubých množín, ako kontrolné riešenie vo všetkých prípadoch tvorí štandardná neurónová sieť. Algoritmus na redukciu atribútov pomocou matice rozlíšiteľnosti bol testovaný porovnávaním neurónovej siete s plným počtom atribútov a sietí s niekoľkými sieťami, z ktorých každá používala rôzne veľkú redukciu počtu vstupov získanú z tohto algoritmu. Úlohou bola klasifikácia ekonomickej činnosti spoločností na základe frekvencií výskytu slov v dokumentoch. Neurónová sieť s hrubými neurónmi bola testovaná proti ekvivalentnej štandardnej neurónovej sieti na úlohe predikcie presnej hodnoty, kde cieľom bolo určiť hodnotu parametra cestnej komunikácie na základe charakteristiky premávky. Neurónová sieť s hrubou logikou bola opäť testovaná na úlohe klasifikácie, kde úlohou bolo určiť podtyp rastliny kosatec, opäť porovnávaná ekvivalentnou štandardnou neurónovou sieťou. Posledným testovaným algoritmom je aproximačná neurónová sieť, ktorá bola porovnávaná príklade klasifikácie, kde cieľom bolo určiť či pacient trpí Parkinsonovou chorobou na základe informácií o jeho hlase.

Algoritmy boli hodnotené z dvoch hľadísk. Pri testoch boli sledované výpočtové nároky jednotlivých algoritmov v podobne času potrebného na skončenie programu a maximálna operačná pamäť zabraná programom pri výpočte. Hodnoty boli merané pomocou linuxového nástroja `time` volaním `/usr/bin/time -v meno-programu`. Druhé hľadisko hodnotenia algoritmov spočíva v hodnotení učenia algoritmov. Bola meraná úspešnosť algoritmu na testovacej sade vstupov, rozdielných od tréningovej množiny, a počas učenia neurónových sietí bola priebežne zaznamenávaná hodnota globálnej chyby (odchýlky) algoritmu a analyzovaná konvergencia siete, teda rýchlosť klesania odchýlky.

4.1 Redukcia atribútov maticou rozlíšiteľnosti

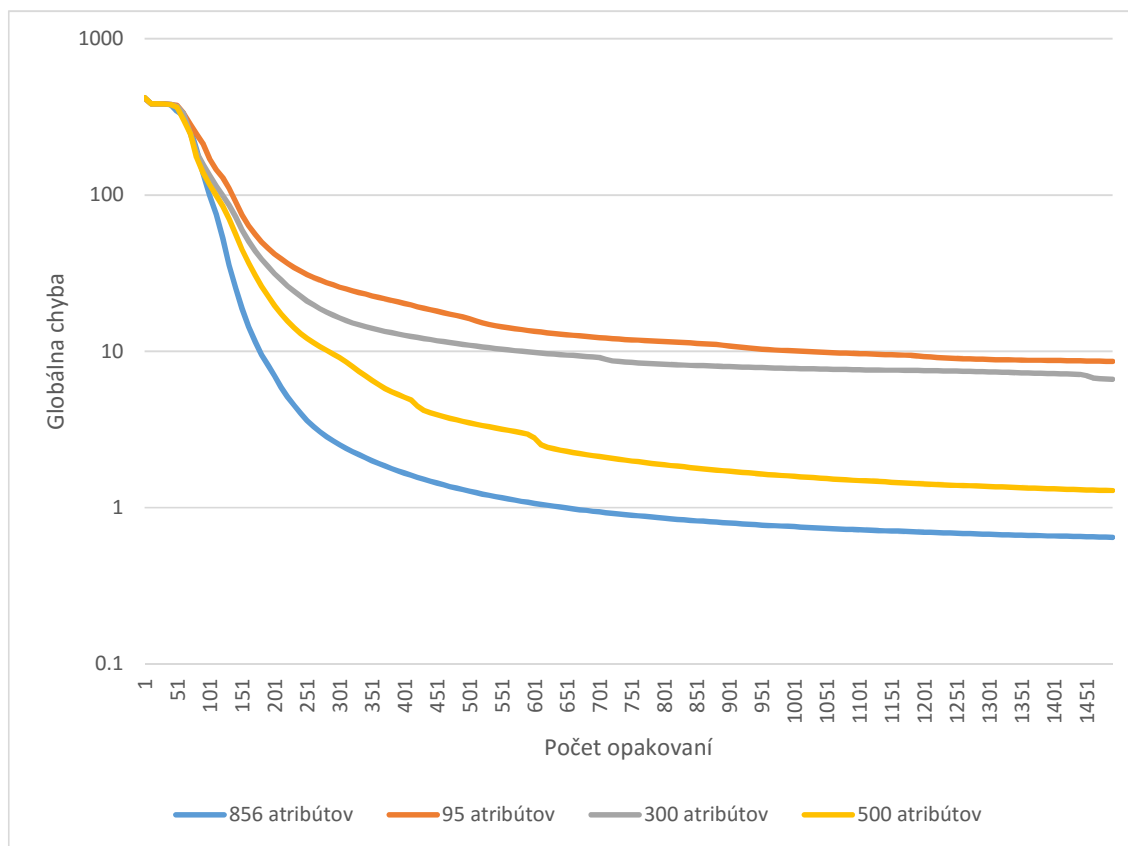
Úlohou redukcie atribútov s maticou rozlíšiteľnosti je urýchlenie spracovania dát, v tomto prípade neurónov sieťou. To je dosiahnuté použitím vhodnej podmnožiny atribútov. Dá sa predpokladať že dôsledky takéhoto predspracovania dát budú najvýraznejšie v prípadoch s vysokým počtom atribútov. Preto hlavným kritériom pri výbere dát pre testy bol počet atribútov. Riešená bude úloha klasifikácie dát. Samotná klasifikácia bude prevedená pomocou štandardnej neurónovej siete, avšak s použitím rôzne veľkých množín atribútov, ktoré budú získané pomocou algoritmu na redukciu atribútov.

Vybrané dáta pre túto úlohu pochádzajú z [14]. Jedná sa o množinu záznamov popisujúcich ekonomickú činnosť brazílskych spoločností. Celkový počet záznamov je 1080. Dáta sú rozdelené do 9 tried. Každý zo záznamov obsahuje klasifikačnú triedu a 856 atribútov, ktoré udávajú frekvenciu

výskytu slova v danom dokumente, zaokrúhlenú na celé čísla. Jedná sa o riedke dáta, teda väčšina hodnôt má nulovú hodnotu. Tieto dáta boli už vhodne spracované a nie je nutná ich ďalšia úprava, nachádzajú sa vo vhodnom formáte pre výpočet matice rozlíšiteľnosti aj pre prácu s neurónovými sieťami.

Prvým krokom bolo získane vhodných redukcii pomocou algoritmu na redukcii atribútov. Výpočet matice rozlíšiteľnosti prebehol počas niekoľkých sekúnd. Výpočet vlastných redukcii z matice rozlíšiteľnosti však trval niekoľko dní, po ktorých algoritmus stále neskončil a bol ukončený. Hoci algoritmus nikdy neskončil, nebol úplne neúspešný, keďže stále vypočítal veľké počet vhodných redukcii. Z nich boli vybrané tri redukcie o rôznych veľkostiach, prvá obsahujúca 500 atribútov, druhá s 300 atribútmi a poslednou je redukcia s najmenším počtom atribútov 95. Porovnanie redukcii prebiehalo na štyroch rovnakých štandardných dopredných neurónových sieťach, navrhnutých v predchádzajúcej kapitole. Jediná zmena spočíva vo vstupoch siete, ktoré boli dané použitou redukciiou. Tieto neurónové siete boli zložené z troch vrstiev. V prvej vrstve bolo použitých 40 neurónov, v druhej vrstve bolo 20 neurónov a posledná vrstva obsahuje 9 neurónov, kde každý reprezentuje jednu klasifikačnú triedu. Parametre sietí boli nastavené nasledovne: koeficient učenia mal hodnotu $\mu = 0.8$, koeficient sigmoidálnej aktivačnej funkcie neurónov mal hodnotu $\lambda = 0.2$, maximálny počet opakovaní bol nastavený na 1500 a maximálna prípustná globálna chyba bola nastavená na $\text{maxErr} = 0.5$.

Úspech neurónovej siete pri klasifikácii daného vstupu môže nadobúdať iba dvoch hodnôt:



Obrázok 11: Graf konvergencií sietí s redukovanými atribútmi

úspešná alebo neúspešná klasifikácia. Úspešnou je vtedy, ak neurón výstupnej vrstvy s najvyššou hodnotou odpovedá očakávanej triede vstupu, inak je neúspešná. Neurónové siete boli trénované na 768 záznamoch a po ukončení učenia boli merané ich úspešnosť klasifikácie na 256 ostávajúcich záznamoch. Porovnanie konvergencie jednotlivých sietí je zobrazené na obrázku 10. Z grafu je zrejmé, že redukcia počtu vstupov siete, aj keď redundantných, výrazne spomaľuje konvergenciu siete. Najvyššiu úspešnosť klasifikácie na testovacej sade dát dosiahli dve neurónové siete, sieť s plným počtom vstupov a sieť s počtom vstupov 500, oboje dosahujúce presnosť až 94.41%. Ďalšou bola sieť s použitím 300 vstupov, ktorá dosiahla 92.5%, pričom najhorší výsledok dosiahla neurónová sieť s 95 vstupmi s úspešnosťou 91.16%. Pri pohľade na časovú a pamäťovú náročnosť jednotlivých sietí je ich poradie obrátené. Redukcia počtu vstupov neurónovej siete má pozitívny dopad na čas spracovávania a aj použitú pamäť. Výpočtové nároky neurónovej siete s 95 vstupmi klesli na tretinu z hľadiska času a na polovicu z hľadiska pamäte oproti kontrolnej sieti. Všetky namerané hodnoty pre testované siete sú v tabuľke 1.

Použitie redukcie atribútov pomocou matice rozlíšiteľnosti na predspracovanie môže výrazným spôsobom zmenšiť veľkosť zaznamenávaných dát, pri použitých dátach až na 11.1%, či zredukovať

časové a pamäťové nároky pre neurónovú sieť. Merania však tiež ukazujú, že aj keď má redukcia rovnakú schopnosť rozlíšiť prvky, na konvergenciu a presnosť neurónovej siete má negatívny dopad. Pri ďalších pokusoch, kedy bolo neurónovej sieti s 95 vstupmi predĺžený čas učenia na rovnaký s kontrolnou sieťou, sa jej podarilo dosiahnuť úspešnosť maximálne 93%. Lepšie výsledky preukázala neurónová sieť s 500 atribútmi, ktorá aj keď konvergovala pomalšie, dosiahla rovnakú úspešnosť pri klasifikácii ako kontrolná sieť, za menší čas a s menšími nárokmi na pamäť. Redukcia atribútov vo všeobecnosti má pozitívny dopad na výkon neurónovej siete, avšak nie je vhodné použiť najmenšiu možnú redukcie, ale nájsť vhodný kompromis.

Neurónová sieť	Čas výpočtu [m:s]	Maximálna použitá pamäť [KB]
856 atribútov (kontrolná)	4:00	114 632
500 atribútov	2:47	96 776
300 atribútov	2:06	71 716
95 atribútov	1:19	60264

Tabuľka 1: Výpočtové nároky neurónových sietí s redukciami atribútov

4.2 Neurónová sieť s hrubými neurónmi

Testy s neurónovou sieťou s hrubými neurónmi prebiehali na príklade podobnom ako v článku [5], kde bola táto sieť navrhnutá. Vstupom neurónovej siete sú hodnoty hrubého charakteru, z ktorej každá predstavuje maximum a minimum počtu áut, ktoré v určitý deň v týždni prešlo daným meracím bodom cestnej komunikácie. Úlohou neurónovej siete je naučiť sa odhadovať čo najpresnejšie hodnotu DHV⁶. Táto hodnota je štandardne získaná ako 30. najvyššia hodnota premávky za hodinu v oboch smeroch, na danom úseku komunikácie, v priebehu celého roku.

Použitá data pochádzajú z [1], odkiaľ boli vybrané záznamy za roky 2011 a 2012. Sú to data zaznamenané zariadeniami na počítanie premávky. Každý zo záznamov v týchto dátach obsahuje nasledujúce informácie:

- identifikátor miesta merania,
- dátum a čas konca merania,
- smer premávky,
- označenie pruhu cestnej komunikácie,
- a počet zaznamenaných vozidiel.

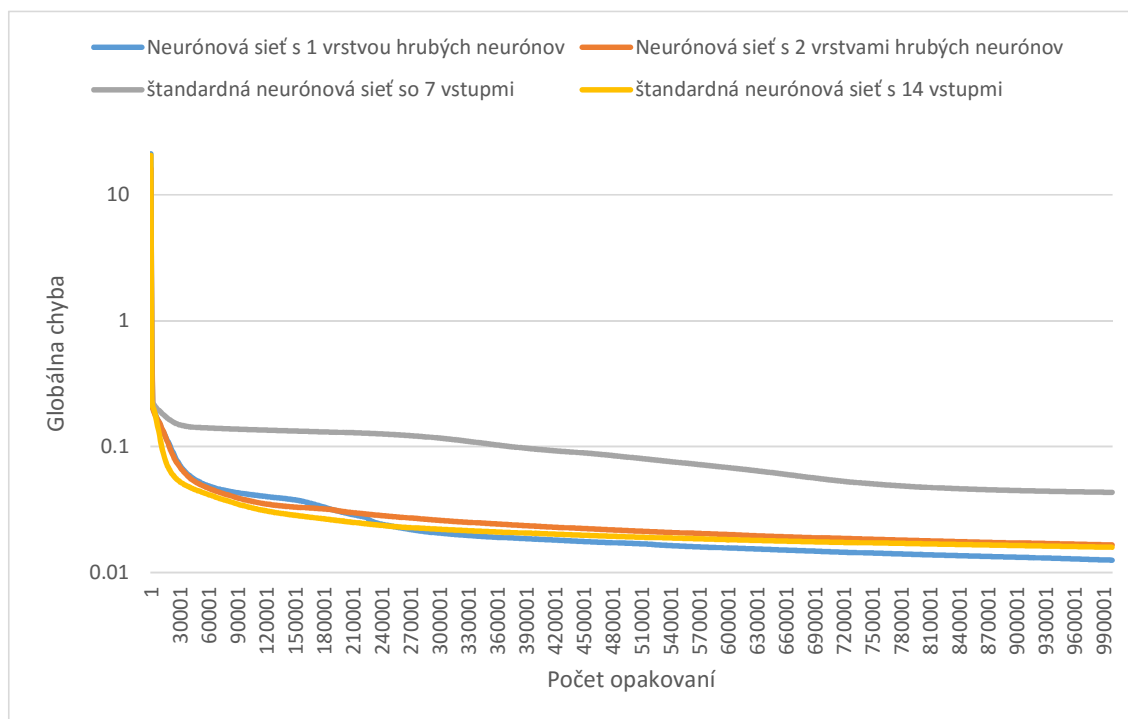
Tieto data však nie sú v požadovanom formáte pre použitie s neurónovou sieťou. Je ich potrebné agregovať a vypočítať hodnoty parametra DHV. Pri predspracovaní boli odstránené záznamy meracích

⁶ DHV – angl. *Design hourly volume* – metrika použitá pri analýze premávky na komunikáciách

bodov, ktoré mali veľký počet záznamov s nulovou hodnotou. Ďalej boli odstránené záznamy tých meracích bodov, ktoré mali málo záznamov. Z takto očistených dát boli sčítané namerané hodnoty pre jednotlivé pruhy a smery premávky, čím vznikli záznamy s počtom vozidiel prechádzajúcim daným úsekom počas hodinových intervalov. Na základe týchto hodnôt bola získaná hodnota DHV pre jednotlivé merané body, ako 30. najvyššia hodnota hodinovej prevádzky. Následne boli dáta ďalej agregované na celé dni. Z týchto dát boli potom získané maximá a minimá premávky pre dni v týždni pre každý meraný bod. Výsledné dáta sa skladajú z 364 záznamov. Z nich bolo 300 záznamov použitých na tréning neurónových sietí, a 64 záznamov slúžilo ako testovacia množina vstupov.

Pri testoch boli porovnávané štyri neurónové siete. Dve neurónové siete s hrubými neurónmi a dve štandardné neurónové siete. Neurónové siete s hrubými množinami majú obidve 7 vstupov hrubého charakteru, kde každý značí minimum a maximum dennej premávky pre daný deň v týždni. Sú zložené z troch vrstiev. Prvá z nich má prvé dve vrstvy tvorené hrubými neurónmi, obsahujúce 7 neurónov v prvej vrstve a 4 neuróny v druhej vrstve. Druhá má iba jednu vrstvu tvorenú 7 hrubými neurónmi a druhú vrstvu s 8 štandardnými neurónmi. Obidve majú poslednú vrstvu siete zloženú z jedného štandardného neurónu. Kontrolné neurónové siete sa obe taktiež skladajú z troch vrstiev. Ich prvé vrstvy obsahujú 14 neurónov, druhá vrstva obsahuje 8 neurónov a výstup je opäť jeden neurón. Rozdiel týchto sietí je v počte vstupov. Prvá kontrolná sieť má 14 vstupov, ktoré po dvojiciach reprezentujú deň v týždni hodnotami jeho minima a maxima. Druhá kontrolná sieť má len 7 vstupov, ktorých hodnoty sú vypočítané ako priemer minima a maxima pre daný deň v týždni. Keďže cieľom je zistiť prínos hrubých neurónov, počty neurónov v jednotlivých sieťach sú zvolené tak, aby každá sieť bola zložená z rovnakého počtu neurónov, s prihliadnutím na to, že hrubý neurón je tvorený dvoma štandardnými. Keďže výstupná hodnota neurónov je v intervale $< 0,1 >$, boli očakávané hodnoty normalizované. Pre všetky tri siete boli zvolené rovnaké parametre. Hodnota koeficientu učenia bola určená $\mu = 0.2$, hodnota koeficientu sigmoidálnej aktivačnej funkcie neurónov bola určená $\lambda = 0.2$, maximálny počet opakovaní učenia siete bol určený $\text{repeats} = 1\,000\,000$, a hodnota maximálnej prípustnej globálnej chyby siete bola určená ako $\text{maxErr} = 0.01$.

Výstupom neurónových sietí je predikcia presnej hodnoty DHV. Presnosť predikcie neurónovej siete je v tomto prípade reprezentovaná pomocou odchýlky medzi výstupnou hodnotou siete a očakávanou hodnotou ako $e = \frac{|d-o|}{d}$, kde d značí očakávanú hodnotu odozvy siete a o značí skutočnú odozvu siete. Opäť bola počas učenia sietí sledovaná hodnota globálnej odchýlky. Graf konverencie sietí je na obrázku 11. Po naučení sietí boli na testovacej sade vstupov merané dva parametre: priemerná



Obrázok 12: Graf konverencie sietí s hrubými neurónmi

odchýlka siete a maximálna odchýlka siete. Pri prvej kontrolnej sieti bola zistená hodnota priemernej odchýlky $e_{avg} = 6.9\%$ a maximálna odchýlka dosahovala hodnoty $e_{max} = 32.44\%$. Kontrolná sieť s redukovaným počtom vstupov dosiahla priemernej odchýlky $e_{avg} = 11.66\%$, teda použitie hodnôt maxima a minima namiesto jednej hodnoty má pozitívny vplyv na presnosť siete. Maximálna chyba tejto siete bola $e_{max} = 34.43\%$. Pre neurónovú sieť s dvoma vrstvami obsahujúcimi hrubé neuróny bola priemerná odchýlka $e_{avg} = 6.27\%$ a maximálna odchýlka $e_{max} = 24.57\%$. Priemerná odchýlka siete nebola oproti kontrolnej sieti výrazne lepšia, maximálna však ukazuje výrazný pokles. Posledná testovaná sieť, neurónová sieť s jednou vrstvou s hrubými neurónmi, sa ukázala ako najlepšia s priemernou odchýlkou $e_{avg} = 5.99\%$ a maximálnou odchýlkou $e_{max} = 22.88\%$, čo je mierne zlepšenie v oboch meraných hodnotách oproti sieti s dvoma vrstvami hrubých neurónov. Z hľadiska výpočtovej náročnosti medzi dvoma kontrolnými sieťami bol len minimálny rozdiel, rovnako medzi sieťami s hrubými neurónmi. Siete s hrubými neurónmi však v porovnaní s kontrolnými sieťami vyžadovali výrazne viac času na výpočet a aj viac pamäte.

Sieť	Čas výpočtu [m:s]	Maximálna použitá pamäť[KB]
kontrolná s 14 vstupmi	37:36	43100
kontrolná s 7 vstupmi	36:12	43000
S 2 vrstvami hrubých neurónov	54:22	48300
S 1 vrstvou hrubých neurónov	53:08	65728

Tabuľka 2: Výpočtové nároky sietí s hrubými neurónmi

Z meraní sietí vyplýva že najlepšie výsledky dosahuje neurónová sieť s jednou vrstvou hrubých neurónov, má najlepšiu konvergenciu aj najlepšiu presnosť výstupu siete. Je zaujímavé že sieť s dvoma vrstvami hrubých neurónov dosahuje mierne horšiu konvergenciu ako štandardná neurónová sieť a napriek tomu dosahuje lepších výstupov. Cenou za toto zlepšenie sa však ukazuje byť výpočtová náročnosť.

4.3 Neurónová sieť s hrubou logikou

Neurónová sieť s hrubou logikou je štruktúrou a funkciou takmer identická so štandardnou neurónovou sieťou. Rozdiel spočíva v učení, ktoré je rozdelené do dvoch častí ktoré neurónovú sieť učia po častiach. Tento prístup vyžaduje pre učenie dodatočné predspracovanie použitých dát, z ktorých je potrebné vydolovať rozhodovacie pravidlá. Neurónová sieť s hrubou logikou bude porovnávaná so štandardnou neurónovou sieťou na príklade klasifikácie vzorkov rastliny kosatca do jej poddruhov.

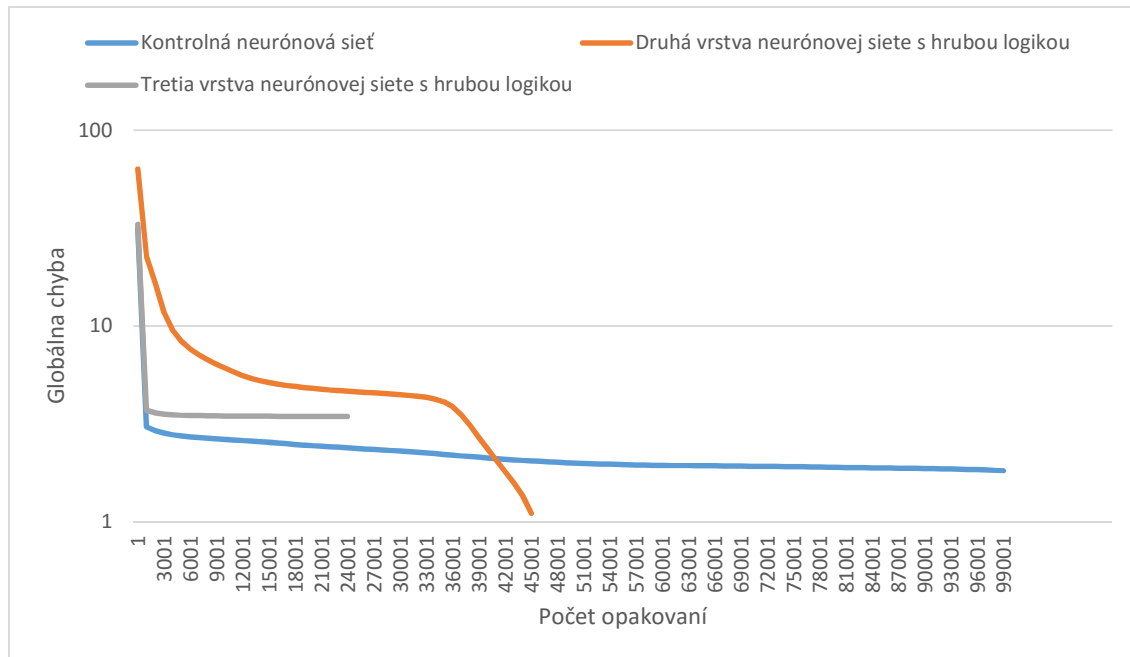
Dáta použité v tejto úlohe pochádzajú z [14]. Obsahujú 150 záznamov meraní rôznych exemplárov kvetov kosatca, ktoré sú následne rozdelené do troch poddruhov. Dáta obsahujú práve 50 záznamov pre každý z týchto poddruhov. Každý záznam sa skladá z nasledujúcich informácií:

- dĺžka okvetného lístka,
- šírka okvetného lístka,
- dĺžka kališného lístka,
- šírka kališného lístka,
- zaradenie do triedy, teda poddruhu rastliny kosatec.

Dáta sú už vo vhodnom tvare pre použitie pre štandardnú neurónovú sieť. Pre neurónovú sieť s hrubou logikou je však potrebné najskôr získať rozhodovacie pravidlá pre klasifikáciu. Tie boli pre tento príklad získané použitím nástroja RapidMiner. Pomocou neho boli hodnoty jednotlivých vlastností diskretizované a z týchto diskretizovaných hodnôt ďalej vygenerované klasifikačné pravidlá. Experimentáciou s parametrami diskretizácie a generovaním pravidiel, boli získané pravidlá s úspešnosťou klasifikácie 98% (147 zo 150 záznamov). Ďalšou výhodou získaných pravidiel je aj redukcia počtu atribútov na dva. Jednotlivé pravidlá majú nasledujúci tvar:

- if p1 = range1 [-∞ - 1.550] then Iris-setosa,
- if pw = range3 [0.800 - 1.350] then Iris-versicolor,
- if p1 = range6 [5.550 - ∞] then Iris-virginica,
- if p1 = range5 [4.950 - 5.550] then Iris-virginica,
- if pw = range4 [1.350 - 1.650] then Iris-versicolor,
- if p1 = range2 [1.550 - 2.450] then Iris-setosa,
- if p1 = range4 [4.350 - 4.950] then Iris-virginica,

kde p_l je dĺžka okvetného lístku a p_w je jeho šírka.



Obrázok 13:Graf konverencie kontrolnej siete a neurónovej siete s hrubou logikou

Neurónová sieť s hrubou logikou má daný pevný počet vrstiev a s výnimkou vstupnej vrstvy závisí počet neurónov v jednotlivých vrstvách od dát a použitých pravidiel. Použitá neurónová sieť s hrubou logikou bude porovnaná s ekvivalentnou štandardnou doprednou neurónovou sieťou. Keďže použité pravidlá pracujú iba s dvoma atribútmi, bol obmedzený počet vstupov siete iba na tieto dva. Prvá vrstva oboch sietí má 8 neurónov, druhá vrstva 7 neurónov. Počet neurónov druhej vrstvy je odvodený od počtu pravidiel získaných pre dáta, kde každý neurón reprezentuje jedno pravidlo. Výstupná vrstva má potom 3 neuróny, kde každý opäť odpovedá klasifikačnej triede. Odpoveď siete bude hodnotená ako správna, pokiaľ neurón s najvyšším výstupom odpovedá známej triede testovacieho vstupu, inak je klasifikácia neúspešná.

Obidve siete sú použité s rovnakými parametrami, ktoré boli získané empiricky. Hodnota koeficientu učenia bola určená na $\mu = 0.8$, hodnota koeficientu sigmoidálnej aktivačnej funkcie neurónov bola určená $\lambda = 0.2$, maximálny počet opakovaní učenia siete bol určený $\text{repeats} = 100\,000$, a hodnota maximálnej prípustnej globálnej chyby siete bola určená ako $\text{maxErr} = 1$. Neurónová sieť s hrubou logikou má navyše určený parameter počtu opakovaní učenia poslednej (rozhodovacej) vrstvy stanovený na štvrtinu opakovaní učenia druhej (pravidlovej) vrstvy. Siete boli učené na množine 90 vzorkov, pre každú klasifikačnú triedu práve 30. Opäť bola počas učenia sledovaná zmena globálnej chyby. Graf konverencie je na obrázku 12. Kvôli rozdeleniu učenia neurónovej siete s hrubou logikou na dve časti nie je možné siete priamo porovnať, ale je ukazuje že konvergencia neurónovej siete s hrubou logikou je výrazne rýchlejšia, najmä konvergencia poslednej

vrstvy. Pre naučené siete bola potom meraná úspešnosť pomocou testovacej množiny vstupov s 60 vzorkami. Obe siete dosahovali rovnakej úspešnosti až 98.33%. Pri meraní výpočtových nárokov jednotlivých sietí dostáva neurónová sieť s hrubou logikou výrazne lepších výsledkov. Nielen že výpočet skončil po necelých 60% času, ktorý potrebovala kontrolná sieť, ale aj použitá pamäť je menej ako polovičná. Avšak keďže algoritmy sú takmer identické, dáta a štruktúra sietí tiež, je pravdepodobné že rozdiel nameraný v použitej pamäti je možné prisúdiť chybe spôsobenej použitým jazykom Java a jeho virtuálnym strojom.

Neurónová sieť	Čas výpočtu[sekundy]	Maximálna použitá pamäť [KB]
s hrubou logikou	20.28	27864
kontrolná	34.81	56168

Tabuľka 3: Výpočtové nároky kontrolnej neurónovej siete a siete s hrubou logikou

Pre použité dáta sa neurónová sieť s hrubou logikou neukázala ako lepšie riešenie z hľadiska presnosti, no ukázalo sa že algoritmus môže byť výrazne rýchlejší ako štandardná neurónová sieť. Prináša so sebou však aj komplikáciu v zmysle nutnosti dolovania pravidiel z dát. Rozhodovacie pravidlá môžu ale značne komplikovať tento prístup, keďže pri hľadaní vhodných dát mali niektoré sady úspešnosť iba okolo 60%.

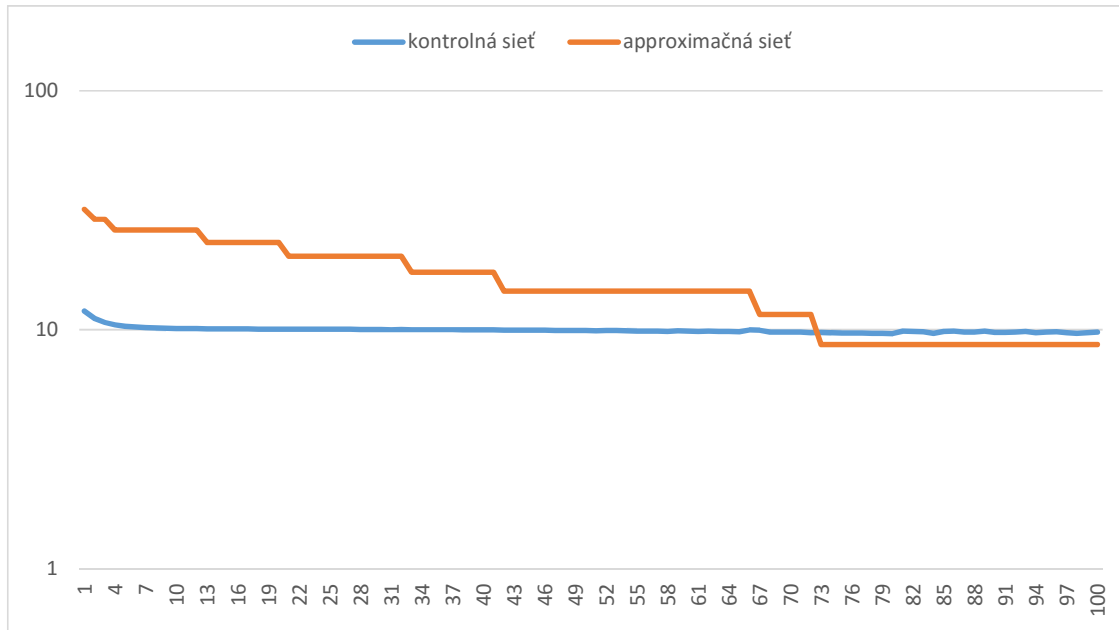
4.4 Aproximačná neurónová sieť

Aproximačná neurónová sieť bola znovu porovnávaná voči štandardnej doprednej neurónovej sieti, opäť na úlohe klasifikácie. Tentokrát bolo úlohou z dát obsahujúcich vlastnosti hlasu človeka určiť, či trpí Parkinsonovou chorobou. Dáta pochádzajú z [6] a obsahujú 195 záznamov, pochádzajúcich od 31 rôznych osôb, z ktorých 23 trpí Parkinsonovou chorobou. Jednotlivé záznamy obsahujú nasledujúce informácie:

- priemerná, maximálna a minimálna fundamentálna frekvencia,
- variácia fundamentálnej frekvencie,
- amplitúda,
- pomer tónu a šumu v hlase,
- fraktálny exponent signálu
- stav pacienta,
- a ďalšie nelineárne vlastnosti frekvencie hlasu.

Celkový počet atribútov, bez triedy klasifikácie danej stavom pacienta, je 22. Ich hodnoty sú spojité. Dáta boli rozdelené na dve množiny, trénovaciu sadu s 50 záznamami a testovaciu sadu so 145 záznamami. Kontrolná neurónová sieť je opäť trojvrstvá dopredná sieť s učením pomocou algoritmu

backpropagation. Skladá sa z troch vrstiev, ktoré obsahujú 22, 10 a 2 neurónov. Výstupná vrstva obsahuje dva neuróny kvôli sigmoidálnej aktivačnej funkcii použitých neurónov. Jeden výstupný neurón značí že pacient má Parkinsonovu chorobu, druhý že je zdravý. Vyhráva ten s vyššou výstupnou hodnotou. Aproximačná neurónová sieť má pevne dve vrstvy. Počet aproximačných neurónov prvej vrstvy je rovnaký ako počet vstupov, teda 22. Každý neurón spracúva jeden atribút. Výstupná vrstva je tvorená jedným rozhodovacím neurónom. Parametre oboch sietí boli určené empiricky. Pre kontrolnú sieť boli zvolené hodnoty koeficientu učenia ako $\mu = 0.8$, koeficientu sigmoidálnej funkcie $\lambda = 0.2$, počet opakovaní *repeats* = 10, a maximálna prípustná globálna chyba *maxErr* = 10. Pre aproximačnú sieť boli parametre nasledujúce: počtu opakovaní *repeats* = 100, maximálna chyba siete *maxErr* = 0.03 a koeficient učenia $\mu = 0.001$.



Obrázok 14: Konvergencie kontrolnej a aproximačnej neurónovej siete

Po naučení oboch sietí pomocou trénovacej množiny bola meraná úspešnosť klasifikácie oboch algoritmov. Kontrolnej sieti sa podarilo dosiahnuť úspešnosti 76.55%. Aproximačná neurónová sieť bola schopná dosiahnuť úspešnosti až 82.07%. Z grafu konvergencií sietí na obrázku 14 je vidieť, že pre kontrolnú sieť ďalšie učenie nemá význam a ani pri ďalších pokusoch s vyšším počtom opakovaní či inými hodnotami parametrov nedokázala dosiahnuť lepších výsledkov. Súčasne boli však merané aj výpočtové nároky oboch sietí, ktoré sú v tabuľke 4. Keďže časy výpočtov sietí sú v obidvoch prípadoch zlomky sekúnd, časová náročnosť nehrá veľkú úlohu. Kontrolná sieť však spotrebovala výrazne menej pamäte.

Neurónová sieť	Čas výpočtu[sekundy]	Maximálna použitá pamäť[KB]
kontrolná	0.15	28548
aproximačná	0.21	37692

Tabuľka 4: Výpočtové nároky kontrolnej a aproximačnej neurónovej siete

Aproximačná neurónová sieť sa pri tejto úlohe osvedčila dosiahnutím lepších výsledkov, než kontrolná sieť. Pri práci so touto sieťou sa však ukázalo, že je veľmi citlivá na preučenie. To s najväčšou pravdepodobnosťou vychádza z použitého algoritmu učenia. Učiace algoritmy v dostupnej literatúre pre túto sieť chýbajú a aj popis funkcie neurónu je len povrchný. Vzhľadom na získané výsledky sa dá povedať, že tento typ siete má potenciál, ale vyžaduje ďalší výskum.

5 Záver

Snaha o kombináciu neurónových sietí s hrubými množinami vychádza z prísľubu odstránením nedostatkov jednotlivých prístupov. Preto tiež došlo v tejto oblasti k rozsiahlemu výskumu. Podľa spôsobu kombinácie týchto teórií boli vzniknuté algoritmy rozdelené do štyroch kategórií: predspracovanie dát, neurónové siete s hrubými neurónmi, neurónové siete s hrubou logikou a hrubé granulórne neurónové siete. Táto diverzita však naznačuje aj určitú nezrelosť doposiaľ vyvinutých algoritmov.

Z každej kategórie bol vybraný jeden algoritmus, pre ktorý bol následne navrhnuté a implementované jeho riešenie a nakoniec porovnané s obyčajnou neurónovou sieťou. Predspracovanie dát pomocou redukcie atribútov pomocou matice rozlíšiteľnosti sa ukázalo ako efektívny spôsob ako urýchliť neurónovú sieť, jednoduchým odstránením nadbytočných atribútov, aj keď to môže viesť ku zníženiu presnosti siete. Tento spôsob má však problém pri práci so spojitými hodnotami, ktoré je nutné najskôr diskretizovať, a taktiež veľkú časovú náročnosť. Neurónová sieť s hrubými neurónmi zapracováva koncept hrubých množín do neurónu jeho rozdelením na dve časti, reprezentujúcu hornú a dolnú aproximáciu. Algoritmus je vhodný pre dáta s nepresnými hodnotami určenými ako interval. Za cenu vyššej výpočtovej náročnosti sa toto riešenie ukázalo byť presnejším než kontrolná neurónová sieť. Tretí prístup predstavuje neurónová sieť s hrubou logikou, ktorá využíva koncept rozhodovacej tabuľky ako predlohu pri učení. Pri testoch dosiahla rovnakú úspešnosť ako kontrolná sieť, ale pri výrazne rýchlejšej konvergencii. Tento prístup však vyžaduje získanie rozhodovacej tabuľky, čo je opäť problematické najmä pri dátach so spojitými hodnotami, a dôležitá je aj kvalita týchto pravidiel. Z oblasti granulórných neurónových sietí bola implementovaná aproximačná neurónová sieť, ktorá sa taktiež osvedčila vyššou presnosťou kvalifikácie. Ukázalo sa však, že je veľmi citlivá na preučenie, čo pravdepodobne je dôsledok použitého algoritmu učenia.

Všetky siete vykázali zlepšenia oproti referenčnej sieti, potvrdzujúc tak, že kombinácia hrubých množín a neurónových sietí skutočne prináša výhody. Ich cenou je nutnosť dodatočného spracovávania dát, alebo vyšších nárokov na výpočet. Všetky z implementovaných riešení sa tiež špecializujú na konkrétnejší typ úlohy, čo je najviac zreteľné z možnosti použiť jednu implementáciu kontrolnej siete pri porovnávaní, avšak testované algoritmy nebolo možné porovnať navzájom. Ďalší vývoj v tejto oblasti je ešte potrebný, ale doposiaľ získané algoritmy naznačujú ešte veľký potenciál.

Zdroje

- [1] DEPARTMENT OF TRANSPORTATION. *HDSB Raw Traffic Data: Downloadable Traffic Volume Data* [online]. [cit. 2015-05-26].
- [2] DING, Shifei, Jinrong CHEN, Xinzheng XU a Jianying LI. *Rough Neural Network: A review*. 2011.
- [3] DING, Shifei a Hao DING. *Research and Development of Attribute Reduction Algorithm Based on Rough Set*. 2010.
- [4] GENG, Zhiqiang a Qunxiong ZHU. *A New Rough set-based Heuristic Algorithm for Attribute Reduct*. 2006.
- [5] LINGRAS, Pawan. *Rough neural Network*.
- [6] LITTLE, Max, Patric MCSHARRY, ROBERTS, Declan COSTELLO a Irene MOROZ. *Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection*. 2007.
- [7] PAL, Sankar K., Lech POLKOWSKI a SKOWRON. *Rough-Neural Computing: Techniques for Computing with Words*. Germany: Springer-Verlag Berlin Heidelberg, 2004.
- [8] PAWLAK, Zdzisław. *Rough set theory and its applications*. 2002.
- [9] PEDRYCZ, Witold a George VUKOVICH. *Granular neural networks*. 2000.
- [10] PETERS, James F., Marcin S. SZCZUKA. *Rough neurocomputing: A Survey of Basic Models of Neurocomputation*.
- [11] PETERS, James, Andrzej SKOWRON, Liangxiu HAN a S. RAMANNA. *Towards Rough Neural Computing Based on Rough Membership: Theory and Application*. 2001.
- [12] PETERS, James, Andrzej SKOWRON, Jaroslaw STEPANIUK a Sheela RAMANNA. *Towards an Ontology of Approximate Reason*.
- [13] SKOWRON, Andrzej a Ning ZHONG. *Rough Sets in KDD - Tutorial Notes* [online]. 2000 [cit. 13.12.2014]. Dostupné z: <<http://roughsets.home.pl/IRSS/rs-kdd/>>
- [14] *UCI Machine Learning Repository* [online]. LICHMAN, Moshe. [cit. 2015-05-26]. Dostupné z: <http://archive.ics.uci.edu/ml>
- [15] VOLNÁ, Eva. *Introduction to Soft Computing*. 1. vyd. 2013. ISBN 978-87-403-0391-9.
- [16] YAO, Yiyu a Qunxiong ZHU. *Granular Computing: basic issues and possible solutions*. 2000. Dostupné z: <<http://www2.cs.uregina.ca/~yyao/PAPERS/basic.pdf>>
- [17] YU, Chub-yan, Ming-hui WU a Ming WU. *Combining Rough Set Theory with Neural Network Theory for Pattern Recognition*. 2003.
- [18] ZHANG, Dongbo a Yaonan WANG. *Rough Neural Network Based on Bottom-Up Fuzzy Rough Data Analysis*. 2009.
- [19] ZHAO, Yueling, Hui JIN, Lihong WANG a Shuang WANG. *Rough Neuron network for Fault Diagnosis*. 2011. Dostupné z: <<http://www.mecs-press.org/ijigsp/ijigsp-v3-n2/IJIGSP-V3-N2-8.pdf>>

Zoznam príloh

Príloha 1. Návod na spustenie vytvorených aplikácií

Príloha 2. CD, obsahujúce spustiteľné súbory, súbory s dátami a zdrojové texty

Príloha 1.

Požiadavkou pre spustenie vytvorených aplikácií je počítač s nainštalovaným JRE (Java Runtime Environment). Spustiteľné súbory sú Java balíky s príponou `.jar`. Pre spustenie tohto súboru je potrebné otvoriť terminál v danej zložke obsahujúcej tento súbor a príkazom

```
java -jar „subor.jar“
```

ho spustiť. Jednotlivé spustiteľné súbory sú pripravené tak, aby nebolo nutné zadávať žiadne parametre. Aplikácia načíta dáta priložené v rovnakej zložke a aplikuje na ne daný algoritmus, ktorý testuje.

Príloha 2.

Priložené CD obsahuje spustiteľné súbory pre každý prevedený test, kde je na dáta aplikovaný konkrétny algoritmus. Tieto súbory sú v zložkách:

- `/runnables/approximation-nn` – obsahuje súbory pre testy aproximačnej neurónovej siete
 - `control.jar` – kontrolná neurónová sieť
 - `approximation.jar` – aproximačná neurónová sieť
- `/runnables/attribute-reduction` – obsahuje súbory pre testy redukcie atribútov
 - `discernibility-matrix.jar` – výpočet matice rozlíšiteľnosti a redukcii
 - `all-attributes.jar` – neurónová sieť so všetkými atribútmi
 - `reduced-500.jar` – neurónová sieť s 500 atribútmi
 - `reduced-300.jar` – neurónová sieť s 300 atribútmi
 - `reduced-95.jar` – neurónová sieť s 95 atribútmi
- `/runnables/rough-login-nn` – obsahuje súbory pre testy neurónovej siete s hrubou logikou
 - `control.jar` – kontrolná neurónová sieť
 - `rough-logic.jar` – neurónová sieť s hrubou logikou
- `/runnables/rough-neuron-nn` – obsahuje súbory pre testy neurónovej siete s hrubými neurónmi
 - `control-average.jar` – kontrolná neurónová sieť s 7 vstupmi (priemer hodnôt minima a maxima)
 - `control-minmax.jar` – kontrolná sieť so 14 vstupmi, hodnotami minima a maxima pre každý deň
 - `rough-neuron-1-layer.jar` – neurónová sieť s 1 vrstvou hrubých neurónov
 - `rough-neuron-2-layers.jar` – neurónová sieť s 2 vrstvami hrubých neurónov

Ďalej sa na CD nachádzajú zdrojové súbory pre jednotlivé implementované algoritmy. Zdrojové súbory pre jednotlivé testy sa nachádzajú v zložke `sources`, ktorá má rovnakú štruktúru ako vyššie popísaná zložka `runnables`.