

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY

DEPARTMENT OF INTELLIGENT SYSTEMS

FUZZY NEURONOVÉ SÍTĚ

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

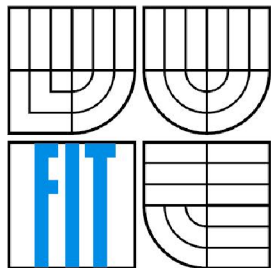
BC. MAREK GONZÁLEZ

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

FUZZY NEURONOVÉ SÍTĚ

FUZZY NEURAL NETWORKS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

BC. MAREK GONZÁLEZ

VEDOUCÍ PRÁCE

SUPERVISOR

DOC. ING. FRANTIŠEK ZBOŘIL, CSC

BRNO 2015

Zadání diplomové práce

Řešitel: **González Marek, Bc.**

Obor: Počítačové a vestavěné systémy

Téma: **Fuzzy neuronové sítě**
Fuzzy Neural Networks

Kategorie: Umělá inteligence

Pokyny:

1. Prostudujte zadanou literaturu.
2. Popište všechny zjištěné přístupy s aplikacemi fuzzy logiky v neuronových sítích.
3. Navrhněte demonstrační program pro některý z těchto přístupů (například pro Mamdani, nebo pro Takagi-Sugeno neuro fuzzy regulátory).
4. Navržený program implementujte.
5. Provedte potřebné experimenty s cílem zjištění výhod/nevýhod kombinovaných přístupů vůči klasickým přístupům (čistě neuronová síť, nebo čistě fuzzy logika - například čistě neuronový, nebo čistě fuzzy regulátor).
6. Zhodnoťte dosažené výsledky.

Literatura:

- Fuller, R.: Neuro-Fuzzy Methods, Vacation school, Neuro-Fuzzy Methods for modelling & Fault Diagnosis, 2001
- Liu, P., Li, H.: Fuzzy Neural Network Theory and Application, World Scientific, 2004
- Moraga, C.: Data Driven Fuzzy Modelling with neural networks, Springer-Verlag Berlin Heidelberg, 2005
- Chai, Y., Jia, L., Zhang, Z.: Mamdani Model based Adaptive Neural Fuzzy System and its Application, International Journal of Information and Mathematical Sciences 5:1, 2009

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Zbořil František V., doc. Ing., CSc.,** UITS FIT VUT

Datum zadání: 1. listopadu 2014

Datum odevzdání: 27. května 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Práce je věnována fuzzy neuronovým sítím. Protože se jedná o techniku kombinování fuzzy logiky s neuronovými sítěmi pro řešení nejrůznějších úloh z oblasti umělé inteligence, řízení či strojového učení, teoretická část práce se nejprve zabývá neuronovými sítěmi a fuzzy logikou odděleně. Po vysvětlení nezbytných základů je zbytek teoretické části zaměřen převážně na teorii fuzzy neuronových sítí, jejich klasifikaci a použití v praxi. Na základě popsané teorie jsou navrženy a implementovány fuzzy asociativní paměti, jejichž cílem je řešit úlohu klasifikace a úlohu z oblasti řízení. Na závěr jsou výsledné implementace a jejich výsledky zhodnoceny a porovnány s jinými přístupy k řešení zvolených úloh.

Abstract

This thesis focuses on fuzzy neural networks. The combination of the fuzzy logic and artificial neural networks leads to the development of more robust systems. These systems are used in various field of the research, such as artificial intelligence, machine learning and control theory. First, we provide a quick overview of underlying neural networks and fuzzy systems to explain fundamental ideas that form the basis of the fields, and follow with the introduction of the fuzzy neural network theory, classification and application. Then we describe a design and a realization of the fuzzy associative memory, as an example of these systems. Finally, we benchmark the realization using the pattern recognition and control tasks. The results are evaluated and compared against existing systems.

Klíčová slova

Fuzzy množiny, fuzzy logika, fuzzy řízení, umělý neuron, neuronové sítě, Hopfieldova síť, obousměrná asociativní paměť, fuzzy neuronové sítě, fuzzy asociativní paměť.

Keywords

Fuzzy sets, fuzzy logic, fuzzy control, artificial neuron, neural networks, Hopfield neural network, bidirectional associative memory, fuzzy neural networks, fuzzy associative memory.

Citace

Marek González: Fuzzy neuronové sítě, diplomová práce, Brno, FIT VUT v Brně, 2015

Fuzzy neuronové sítě

Prohlášení

Prohlašuji, že jsem svoji diplomovou práci vypracoval samostatně pod vedením Doc. Ing. Františka V. Zbořila, CSc. a že jsem uvedl všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Marek González

27.5.2015

Poděkování

Rád bych poděkoval svému vedoucímu Doc. Ing. Františku V. Zbořilovi, CSc. za všechny podněty a odbornou pomoc, které mi při psaní této práce poskytnul.

© Marek González, 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

1	Úvod.....	7
2	Neuronové sítě.....	9
2.1	Model neuronu.....	9
2.2	Typy neuronových sítí a jejich učení.....	10
2.3	Neuronové sítě jako asociativní paměti.....	11
2.3.1	Hopfieldova neuronová síť.....	13
2.3.2	Obousměrná asociativní paměť.....	14
2.3.3	Stabilita, atraktory, kapacita.....	15
3	Fuzzy systémy.....	16
3.1	Fuzzy množiny a logika.....	16
3.1.1	Operace s fuzzy množinami.....	16
3.1.2	Fuzzy relace.....	19
3.2	Fuzzy řízení.....	20
3.3	Geometrické pojetí fuzzy množin.....	23
4	Fuzzy neuronové sítě.....	25
4.1	Klasifikace fuzzy neuronových sítí.....	25
4.2	Fuzzy neuron.....	26
4.3	Fuzzy asociativní paměť.....	27
4.3.1	Učení.....	29
4.3.2	Odezva.....	32
4.3.3	Koskova složená FAM.....	33
4.3.4	BIOFAM inference.....	34
5	Úlohy.....	37
5.1	Obnova poškozeného obrazu.....	37
5.2	Parkování kamiónu s přívěsem.....	37
6	Realizace.....	39

6.1	Fuzzy asociativní paměť pro restauraci obrazu	39
6.2	Parkování vozu	40
6.2.1	Simulátor	40
6.2.2	Fuzzy regulátor	41
6.2.3	FAM systém jako regulátor.....	43
6.2.4	Adaptivní fuzzy asociativní regulátor.....	45
7	Experimenty	47
7.1	Restaurace obrazu.....	47
7.1.1	Obrazová data	47
7.1.2	Maximální kapacita	48
7.1.3	Schopnost opravy.....	50
7.1.4	Zhodnocení	53
7.2	Řízení.....	54
7.2.1	Porovnání regulátorů	54
7.2.2	Učení pravidel.....	57
7.2.3	Zhodnocení	57
8	Závěr	59
	Literatura	60

1 Úvod

Schopnost učit se a řešit úlohy, ve kterých převládá nepřesnost a nejistota, je doménou člověka. Právě lidská mysl se stala modelem pro celou třídu netradičních technologií a přístupů k řešení obtížných (často praktických) problémů počítačem. Tématem této práce je integrace dvou takovýchto technik. Konkrétně se práce zabývá fuzzy neuronovými sítěmi, tedy spojením fuzzy logiky s neuronovými sítěmi.

Výstupem práce bude návrh a implementace dvou fuzzy systémů založených na asociativních pamětech. Jeden bude sloužit k opravě poškozeného obrazu, druhý pak jako kontrolér. Kvalita těchto systémů bude posléze porovnána s odpovídajícím fuzzy regulátorem a neuronovou sítí.

Neuronové sítě i fuzzy inferenční systémy mohou být využity k řešení úloh, u kterých neznáme matematický model. Každý z obou konceptů má své výhody a nevýhody. Cílem vzájemné integrace je tyto nevýhody potlačit. Použití neuronové sítě přichází v úvahu, je-li problém vyjádřen dostatečným množstvím pozorovaných dat, která umíme vhodně vybrat a předzpracovat. Tato data se pak použijí k naučení sítě, která funguje jako černá skříňka, což je považováno za nevýhodu, protože z naučených vah se těžko extrahují srozumitelné znalosti. Na druhou stranu nepotřebujeme znát řešený problém příliš do hloubky. Naproti tomu fuzzy systémy se neučí. Pravidla a parametry funkcí příslušnosti musí být navrženy ručně, což může být (a obvykle bývá) velmi časově náročné a náchylné k chybám [1].

Způsobů kombinace obou přístupů existuje vícero a hranice mezi tím, co lze zařadit pod pojem fuzzy neuronové sítě, je sama o sobě velmi neurčitá. Asi nejznámější a v literatuře nejběžnější jsou tzv. hybridní neuro-fuzzy systémy, v klasifikaci podle Liua [2] také označovány jako fuzzy inferenční sítě. Do této kategorie spadá například systém Adaptive Neuro-Fuzzy Inference System - ANFIS. Jak uvidíme v kapitole *Klasifikace fuzzy neuronových sítí*, hybridní neuro-fuzzy systémy nejsou jediným přístupem ke spojení fuzzy logiky a neuronových sítí.

Práce je rozdělena na část teoretickou a na část praktickou. Protože neuronové sítě a fuzzy logika jsou dvě samostatné metodologie, první dvě kapitoly, z nichž se první zabývá neuronovými sítěmi a druhá fuzzy logikou, je také probírají odděleně. Spíše než podání vyčerpávajícího teoretického výkladu mají tyto kapitoly za cíl provést čtenáře základy použitých technik a vysvětlit všechny pojmy a souvislosti, které se vyskytují ve zbytku práce.

Třetí kapitola se již zabývá samotnou integrací. Jsou zde uvedeny různé přístupy aplikace fuzzy logiky v neuronových sítích a jejich nasazení v praxi. Protože je práce zaměřena na fuzzy asociativní paměti, bude právě jim věnováno nejvíce prostoru.

Poslední kapitola teoretické části se věnuje definicím a popisu zvolených úloh pro demonstraci fuzzy neuronové sítě. Jedná se o restaurování obrazu a o úlohu z oboru řízení. Úloha

opravy obrazu je pojata zjednodušeně a spíše než o samotný obraz nám půjde o prozkoumání vlastností fuzzy asociativních pamětí. Na základě těchto vlastností je vybrán konkrétní model pro řešení řídicí úlohy, jejímž cílem je navést couvající kamión k rampě. Úloha je v angličtině známá pod názvem "*truck backer-upper problem*" nebo "*truck control problem*". Zde budeme používat volně přeložený název "parkování kamionu".

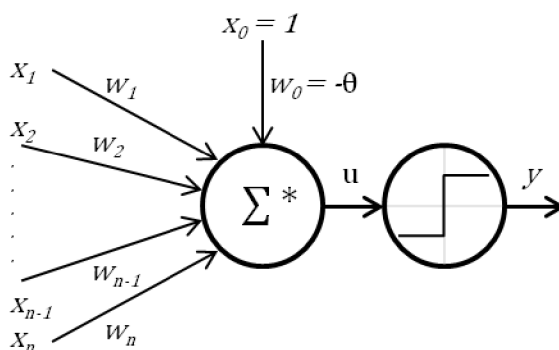
První kapitola praktické části je zaměřena na návrh a implementaci výše zmíněných fuzzy asociativních pamětí. Praktickým výstupem této diplomové práce je simulátor nákladního vozu umožňující různým kontrolérům řídit parkovací proces. V této kapitole jsou uvedeny důležité funkce aplikace, popis řídicích systémů a problémy spojené s jejich implementací. Experimenty s implementovanými algoritmy a zhodnocení jejich výsledků jsou prezentovány v kapitole č. 7.

2 Neuronové sítě

Neuronové sítě jsou zjednodušeným modelem lidského mozku. A stejně tak jako náš mozek, i neuronové sítě mají schopnost se učit. Proto se staly významnou technikou v oblasti umělé inteligence a strojového učení. Používají se pro řešení celé řady úloh, jakými jsou klasifikace, rozpoznávání, predikce časových řad, optimalizace, shlukování a mnoho dalších. V praxi byla neuronová síť například nasazena na několika amerických letištích, kde sloužila k detekci plastických trhavin [3].

2.1 Model neuronu

Umělé neuronové sítě jsou složeny z propojených výpočetních jednotek, které jsou inspirovány biologickými neurony. První matematický model neuronu vytvořili W. McCulloch a W. Pitts v roce 1943. V současnosti je však více používán jednodušší model, který je schematicky zobrazen na obrázku 2-1. Zbytek kapitoly se jím bude zabývat podrobněji.



Obrázek 2-1: Neuron s lineární bázovou funkcí a s aktivační funkcí signum

Neuron má n vstupů x , n vstupních vah w , práh θ , vnitřní potenciál u , bázovou funkci f , aktivační funkci g a jeden výstup y . Vstupy i váhy budeme zapisovat jako dva vektory \vec{x} a \vec{w} o n prvcích. Rovnice (2-1) a (2-2) popisují transformaci vstupního vektoru \vec{x} na výstup y .

$$u = f(\vec{x}) \quad (2-1)$$

$$y = g(u) \quad (2-2)$$

Nejběžnějším typem funkce f je lineární bázová funkce (2-3), která provádí skalární součin vstupního vektoru a vektoru vah. V rovnici jsou vstupní i váhový vektor rozšířeny o prvky s nulovými indexy $x_0 = 1$ a $w_0 = -\theta$, které umožňují začlenit práh do bázové funkce, čímž se aktivační funkce g posune do bodu nula a bude tak jednodušší.

$$u = f(\vec{x}) = \vec{x} \cdot \vec{w} = \sum_{i=0}^n x_i \cdot w_i \quad (2-3)$$

Jako aktivační funkci budeme pro diskrétní výstup používat skokovou funkci (2-4) nebo funkci signum (2-5) a pro spojitý výstup sigmoidu (2-6).

$$y = g_{jump}(u) = \begin{cases} a, & u > 0 \\ b, & u < 0 \\ y_{old}, & u = 0 \end{cases} \quad (2-4)$$

$$y = g_{sign}(u) = \begin{cases} 1, & u > 0 \\ -1, & u < 0 \\ 0, & u = 0 \end{cases} \quad (2-5)$$

$$y = g_{sigmoid}(u) = \frac{1}{1 + e^{-\lambda u}} \quad (2-6)$$

Učení neuronu je založeno na vhodných změnách váhového vektoru \vec{w} . Způsobů učení existuje více. Některé z nich budou probrány níže [4].

2.2 Typy neuronových sítí a jejich učení

Při návrhu neuronové sítě si kromě výběru aktivační a bázové funkce musíme také zvolit další parametry. Podle těchto parametrů, kterými jsou: způsob učení, architektura a konec konců i výsledná funkce, kterou má síť vykonávat, můžeme neuronové sítě klasifikovat.

Architektura neboli topologie je vzájemné propojení neuronů, jejichž vektory vah tvoří řádky váhové matice W . Mezi možné propojení například patří plně propojená síť. Všechny neurony v takovéto síti jsou propojeny se všemi ostatními. Jsou-li váhy mezi všemi spoji symetrické tj. $w_{ij} = w_{ji}$, jedná se o symetrickou plně propojenou síť. Dalším a užívanějším typem zapojení je dopředná síť, kde jsou neurony řazeny do vrstev. Spoje mezi neurony vedou pouze k neuronům v sousední vrstvě ve směru ze vstupu na výstup.

Neuronové sítě lze dělit i podle způsobu učení. K učení s učitelem se používají trénovací data ve tvaru dvojice - vstupní vektor a požadovaná odezva sítě. Pro učení bez učitele je možno data

reprezentovat pouze vstupními vektory. Konkrétním případem učení s učitelem je adaptační učení využívající požadované odezvy a skutečného výstupu k určení chyby sítě, na základě které upraví váhy. Nejznámějším algoritmem tohoto přístupu je backpropagation. Korelační učení je formou učení bez učitele využívající tzv. Hebbův princip, který v roce 1949 stanovil D. Hebb. Základní myšlenkou je, že pokud neuron i opakovaně vyvolává excitaci neuronu j , synaptická vazba jejich spojení zesiluje a neuron j se stává citlivějším na podněty z neuronu i . Obecně se Hebbovo pravidlo popisuje rovnicí (2-7), ve které Δw_{ij} udává změnu váhy mezi neurony i a j , x_i je vstupem neuronu i , y_j je výstupem neuronu j a $\eta \in < 0,1 >$ je tzv. koeficient učení.

$$\Delta w_{ij} = \eta x_i y_j \quad (2-7)$$

Hebbovo pravidlo přizpůsobené asociativním pamětem ukazuje rovnice (2-8), kde x_i a x_j jsou vstupy neuronů i a j a w_{ij} je symetrická váha mezi nimi,

$$w_{ij} = x_i x_j \quad (2-8)$$

Jak učení, tak odezva neuronové sítě probíhá zpravidla v iteracích. Mění-li se stavy všech neuronů současně po skončení iterace, mluvíme o tzv. synchronním výpočtu. V případě, kdy neuron změní stav okamžitě a ostatní tento nový stav vidí, jedná se o asynchronní výpočet.

Většinou se neuronové sítě používají jako klasifikátory, aproximátory. Klasifikátory jsou sítě rozpoznávající vstupní vzory, které pak zařazují do tříd. Aproximátory si na základě trénovacích dat vytvářejí vnitřní model. Tento model aproximuje nějaký skutečný systém, což po natrénování dovoluje predikovat příští výstupy aproximovaného systému.

2.3 Neuronové sítě jako asociativní paměti

Neuronové sítě mohou být použity jako asociativní paměti. Tento druh paměti nepoužívá tradiční numerické adresování, ale je adresován obsahem, proto se někdy používá označení obsahem adresovatelná paměť. Obsahem rozumíme vektor a neuronová síť fungující jako asociativní paměť odpovídá na vstupní vektor (adresu) výstupním vektorem, který je chápán jako obsah paměti. Tento proces je též nazýván vybavení, odpověď nebo odezva.

Asociativní paměti pracují ve dvou fázích. První fází je učení neboli ukládání, během kterého dochází k asociaci vstupních vektorů s jejich požadovanou odezvou. Množina asociací uložených v této fázi se označuje jako fundamentální množina. Po naučení sítě přichází na řadu fáze odpovídání. Je-li na vstup přiložen vektor z fundamentální množiny, na výstup se v ideálním případě vybaví

odpovídající asociace – tedy požadovaná odezva. Pro vektor, který není součástí fundamentální množiny, se po přiložení na vstup vybaví na výstupu ten z natrénovaných vektorů, který je mu nejpodobnější z hlediska Hammingovy vzdálenosti. d_h (vektory jsou binární / bipolární). Asociativní paměti tak mají schopnost opravovat poškozený vstup. Podle druhu asociace, které jsou dvojího typu, se asociativní paměti dále dělí na autoasociativní a heteroasociativní. Trénovací množina $T = \{(\vec{x}_1, \vec{y}_1), \dots, (\vec{x}_p, \vec{y}_p)\}$ heteroasociativních pamětí je složena z p dvojic tvořených vstupním vektorem \vec{x} a požadovanou odezvou \vec{y} . Následující vzorce jsou formálním zápisem činnosti autoasociativní paměti se známým vstupem (2-9), s neznámým vstupem (2-10) a heteroasociativní paměti se známým (2-11) a neznámým (2-12) vstupem.

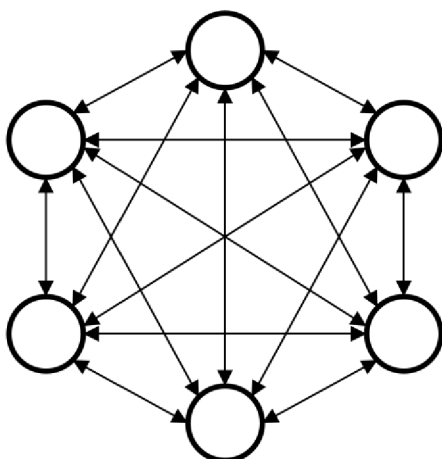
$$\vec{x} \in T \Rightarrow \phi(\vec{x}) = \vec{x} \quad (2-9)$$

$$\vec{x} \notin T \Rightarrow \phi(\vec{x}) = \vec{x}', d_h(\vec{x}, \vec{x}') \leq d_h(\vec{x}, \vec{x}_i), \forall i \quad (2-10)$$

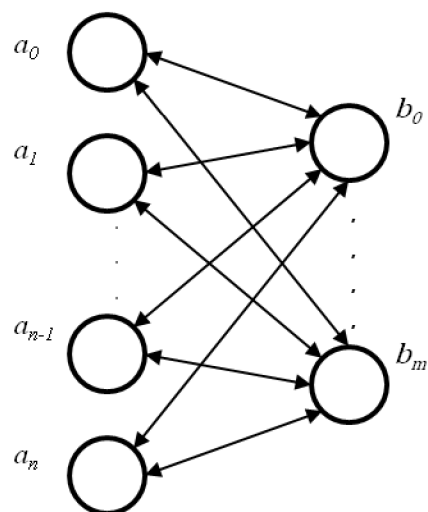
$$\vec{x} \in T \Rightarrow \phi(\vec{x}) = \vec{y}, \vec{y} \in T \quad (2-11)$$

$$\vec{x} \notin T \Rightarrow \phi(\vec{x}) = \vec{y}', d_h(\vec{x}, \vec{x}') \leq d_h(\vec{x}, \vec{x}_i), \forall i \quad (2-12)$$

Reprezentanty heteroasociativních pamětí jsou obousměrná asociativní paměť a paměť s řídkce rozmístěnými položkami. U autoasociativních pamětí je trénovací množina $T = \{(\vec{x}_1), \dots, (\vec{x}_p)\}$ tvořena pouze vstupními vektory \vec{x} , které zároveň tvoří požadovanou odezvu. Nejznámější autoasociativní sítí je síť Hopfieldova [5].



Obrázek 2-2: Hopfieldova síť



Obrázek 2-3: Obousměrná asociativní paměť

2.3.1 Hopfieldova neuronová síť

Jedná se o plně propojenou symetrickou síť, používající lineární bázovou funkci a skokovou aktivační funkci. Vstupy mohou být jak binární, tak bipolární. Učení zde probíhá na základě Hebbova principu. Hopfieldova síť je rekurentní. To znamená, že spojení mezi neurony tvoří orientované cykly. Po vložení vstupu začíná iterativní výpočet, který je ukončen poté, co se výstup neuronů stabilizuje a dále se nemění. Rovnice (2-13) ukazuje výpočet prvků matice vah a rovnice (2-14) výpočet prahů $\vec{\theta}$. Tyto dvě rovnice reprezentují fázi učení sítě, kde n je počet neuronů a p je velikost trénovací množiny. Počet neuronů je dán velikostí vstupních vektorů [6].

$$w_{ij} = \begin{cases} \sum_{k=1}^p x_i^k \cdot x_j^k, & i \neq j \\ 0, & i = j \end{cases} \quad (2-13)$$

$$\theta_i = \frac{1}{2} \sum_{j=1}^n w_{ij} \quad (2-14)$$

Výpočet odezvy pro vstupní vektor \vec{x} je možno popsat např. algoritmem 2-1. Tento algoritmus je synchronní a není pro něj zaručena konvergence (viz níže). Připomeňme, že prahy θ_i jsou automaticky přidávány do váhových vektorů jako prvky s nulovým indexem tj. $w_{i0} = -\theta_i$.

1) *Inicializace:*

$$t = 0; \vec{y}^0 = \vec{x};$$

2) *Výpočet*

$$y_i^{t+1} = g_{jump} \left(\sum_{j=0}^n y_j^t \cdot w_{ij} \right), \forall i \in \langle 1, n \rangle;$$

3) *Iterace*

$$\text{if } (\vec{y}^{t+1} == \vec{y}^t) \{$$

$$\vec{y} = \vec{y}^t;$$

$$\text{exit}();$$

}

else{

$$t = t + 1;$$

$$\text{goto } 2);$$

}

Algoritmus 2-1: Odezva Hopfieldovy neuronové sítě

2.3.2 Obousměrná asociativní paměť

Obousměrná asociativní paměť (dále jen BAM – z anglického názvu *bidirectional associative memory*) je dvouvrstvá síť. Jedna vrstva je označována symbolem A , druhá vrstva symbolem B . Vrstvy jsou vzájemně úplně symetricky propojeny, přičemž vazby mezi neurony stejné vrstvy a zpětné vazby sítě neobsahuje. Bázová i aktivační funkce je stejná jako u Hopfieldovy neuronové sítě. Mají-li obě vrstvy stejný počet neuronů, můžeme BAM používat pro autoasociaci. Hopfieldovu síť tak můžeme považovat za speciální případ BAM. Výpočet vah je popsán rovnicí (2-15) a výpočet prahových vektorů $\vec{\theta}_A$ a $\vec{\theta}_B$ obou vrstev ukazují rovnice (2-16) a (2-17). Trénovací množinu tvoří dvojice (\vec{a}_k, \vec{b}_k) a počet neuronů ve vrstvě A je n a počet neuronů ve vrstvě B je m [6].

$$w_{ij} = \sum_{k=1}^p a_i^k \cdot b_j^k \quad (2-15)$$

$$\theta_{A,i} = \frac{1}{2} \sum_{j=1}^m w_{ij} \quad (2-16)$$

$$\theta_{B,j} = \frac{1}{2} \sum_{i=1}^n w_{ij}, \quad (2-17)$$

Algoritmus 2-2 popisuje vybavovací fázi BAM.

1) *Inicializace:*

$$t = 0; \vec{a}^0 = \vec{x}; \vec{b}^0 = 0;$$

2) *Výpočet*

$$b_j^{t+1} = g_{jump}(\sum_{i=0}^n a_i^t \cdot w_{ij}), \forall j \in \langle 1, m \rangle;$$

$$a_i^{t+1} = g_{jump}(\sum_{j=0}^m b_j^t \cdot w_{ij}), \forall i \in \langle 1, n \rangle;$$

3) *Iterace*

$$if(\vec{a}^{t+1} == \vec{a}^t \ \&\& \ \vec{b}^{t+1} == \vec{b}^t) \{$$

$$\vec{y} = \vec{b}^t;$$

exit();

}

else { t = t + 1; goto 2); }

Algoritmus 2-2: Odezva BAM

2.3.3 Stabilita, atraktory, kapacita

Odezva obou výše zmíněných modelů je počítána iterativně. Protože výpočet je ukončen až poté, co síť dosáhne neměnného stavu, vyvstává otázka, zdali je tento stabilní stav dosažen vždy. Hopfieldova neuronová síť konverguje, je-li použit asynchronní výpočet [7]. Při užití synchronního výpočtu není dosažitelnost stabilního stavu zaručena [8]. Kosko [9] ukázal, že obousměrná asociativní paměť je stabilní nepodmíněně. Problémy asociativních pamětí spočívají v malé paměťové kapacitě a ve falešných atraktorech. Paměťová kapacita je odvozena od maximální velikosti fundamentální množiny, která může být uložena a bez chyby (či s rozumnou chybou) opět vybavena. Vztah (2-18) udává teoretickou maximální velikost fundamentální paměti p_{max} pro bezchybné vybavení v Hopfieldově síti s n neurony.

$$p_{max} = \frac{n}{4 \ln n} \quad (2-18)$$

Pokud se spokojíme s bezchybným vybavením pouze většiny fundamentální paměti, kapacitu můžeme uvažovat dvakrát větší, než je uvedena v (2-18) [6]. Pro paměťovou kapacitu BAM sám autor odvodil vztah (2-19), kde n a m jsou počty neuronů ve vrstvách A a B [9]. Tato hodnota úplné bezchybné vybavení nezaručuje, nýbrž představuje pouze hrubý odhad.

$$p_{max} = \min(n, m) \quad (2-19)$$

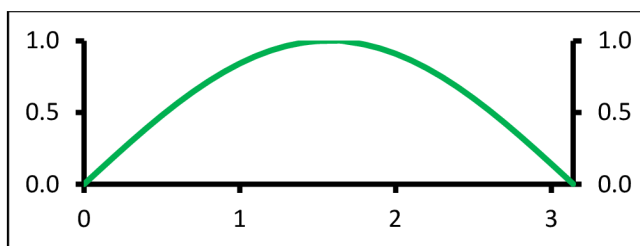
Atraktor v síti je stav, ke kterému síť během vybavovací fáze po čase dospěje. Vektory z fundamentální množiny jsou atraktory. Bohužel v síti se nacházejí i atraktory, které jsou nežádoucí, tedy falešné. Pokud síť konverguje k těmto falešným stavům, výsledkem je chybná odezva, která není součástí fundamentální množiny. Atraktor lze definovat i pomocí energie: S každým stavem sítě je spojena skalární veličina, která má tu vlastnost, že při změně výstupu náhodně vybraného neuronu hodnota této veličiny buďto klesne nebo se nezmění, ale nikdy nestoupne. Proto se této veličině říká energie. Mimoto během opakovaných asynchronních změn stavů neuronů síť časem dospěje do stavu, kdy již energie neklesá. Tento stav je lokálním minimem energetické funkce neboli atraktorem.

3 Fuzzy systémy

V reálném životě existují třídy objektů, pro které není jasně definováno, zda do nich konkrétní objekt patří nebo ne. Například pro třídu savci toto jednoznačně určit lze. Víme, které objekty savci jsou a které nikoliv. Naproti tomu ve třídě vysokých lidí, krásných žen či reálných čísel o mnoho větších než jedna jsou podobná tvrzení o členství značně subjektivní. Takto nepřesně definované třídy však v uvažování člověka hrají významnou roli. Proto, aby šla podobná vágnost či neurčitost lépe zvládat, Lotfi A. Zadeh v roce 1965 zavedl zobecnění klasických (ostrých) množin - fuzzy (neostré) množiny [10].

3.1 Fuzzy množiny a logika

Fuzzy množina je kolekce objektů, pro které je definována hodnota z intervalu $\langle 0,1 \rangle$ představující stupeň příslušnosti objektu do dané množiny. Čím větší je stupeň příslušnosti, tím více objekt do množiny patří. Protože mezi "patřením" a "nepatřením" do množiny neexistuje ostrá hranice, mluvíme o neostrých množinách. Formálněji řečeno fuzzy množina je charakterizována množinou prvků X zvanou univerzum, které může nabývat jak spojitých, tak diskrétních hodnot, a funkcí $m : X \rightarrow \langle 0,1 \rangle$ přiřazující stupeň příslušnosti každému objektu univerza. Funkce m se nazývá funkce příslušnosti. Zápisem $A = (X, m_A)$ rozumíme definici fuzzy množiny A na univerzu X s funkcí příslušnosti m_A . Příklad fuzzy množiny $(X = \langle 0, \pi \rangle, m = \sin(x), x \in X)$ můžeme vidět na obrázku 3-1.



Obrázek 3-1: Fuzzy množina

3.1.1 Operace s fuzzy množinami

Tato kapitola se zabývá vlastnostmi fuzzy množin a operacemi s nimi spojenými, které Lotfi A. Zadeh a další definovali. Některé operace jsou zobecněním ostrých množin, jiné jsou specifické

pro množiny neostré (např. koncentrace a dilatace). V tabulce 3-1 jsou uvedeny nejdůležitější vlastnosti a operace nad fuzzy množinami [10][11]. Vzorce z tabulky předpokládají fuzzy množiny $A = (X, m_A)$ a $B = (X, m_B)$ definované na stejném univerzu.

Vlastnosti fuzzy množin	
prázdná fuzzy množina	$empty(A) \Leftrightarrow m_A(x) = 0, \forall x \in X$
výška fuzzy množiny	$H(A) = m_A(x), m_A(x) \geq m_A(x'), \forall x' \in X$
normální fuzzy množina	$normal(A) \Leftrightarrow H(A) = 1$
nosič	$support(A) = \{x \in X \mid m_A(x) > 0\}$
jádro	$core(A) = \{x \in X \mid m_A(x) = 1\}$
body přechodu	$cross(A) = \{x \in X \mid m_A(x) \geq 0.5\}$
Operace s fuzzy množinami	
sjednocení (disjunkce)	$m_{A \vee B}(x) = \max(m_A(x), m_B(x)), \forall x \in X$
průnik (konjunkce)	$m_{A \wedge B}(x) = \min(m_A(x), m_B(x)), \forall x \in X$
součet	$m_{A+B}(x) = m_A(x) + m_B(x), m_A(x) + m_B(x) \leq 1, \forall x \in X$
součin	$m_{A \cdot B}(x) = m_A(x) \cdot m_B(x), \forall x \in X$
absolutní rozdíl	$m_{ A-B }(x) = m_A(x) - m_B(x) , \forall x \in X$
doplňěk (negace)	$m_{\bar{A}}(x) = 1 - m_A(x), \forall x \in X$
koncentrace	$m_{con}(x) = m_A(x)^2, \forall x \in X$
dilatace	$m_{dil}(x) = \sqrt{m_A(x)}, \forall x \in X$
rovnost	$A = B \Leftrightarrow m_A(x) = m_B(y), \forall x, \forall y \in X$
inkluze	$A \subseteq B \Leftrightarrow m_A(x) \leq m_B(y), \forall x, \forall y \in X$

Tabulka 3-1: Operace a vlastnosti fuzzy množin

S konjunkcí není spojena pouze operace minimum, která je uvedena v tabulce 3-1. Obecně jakákoliv operace nad fuzzy množinami $C : \langle 0,1 \rangle \times \langle 0,1 \rangle \rightarrow \langle 0,1 \rangle$ splňující $C(0,1) = 0$, $C(1,0) = 0$, $C(0,0) = 0$ a $C(1,1) = 1$ je fuzzy konjunkcí. Je-li C navíc komutativní a asociativní, jedná se o tzv. t -normu. Fuzzy součin a průnik jsou t -normy. Spolu s dalšími jsou definovány rovnicemi (3-1) až (3-4) ukazující postupně minimum, součinnou, Łukasiewiczovu a drastickou konjunkcí. Pro přehlednost jsou definice operací v dalším textu psány ve zjednodušeném tvaru, kde a a b jsou stupně příslušnosti prvků $x \in X$ dvou fuzzy množin $A = (X, m_A)$ a $B = (X, m_B)$.

$$C_{Minimum}(a, b) = a \wedge b \quad (3-1)$$

$$C_{Product}(a, b) = a \cdot b \quad (3-2)$$

$$C_{\text{Łukasiewicz}}(a, b) = 0 \vee (a + b - 1) \quad (3-3)$$

$$C_{\text{Drastic}}(a, b) = \begin{cases} a, & b = 1 \\ b, & a = 1 \\ 0, & a \neq 1 \wedge b \neq 1 \end{cases} \quad (3-4)$$

Obdobně jako fuzzy konjunkce je definována i fuzzy disjunkce. O operaci D definovanou jako $D : \langle 0,1 \rangle \times \langle 0,1 \rangle \rightarrow \langle 0,1 \rangle$ říkáme, že je disjunkcí, pokud splňuje $D(0,1) = 1$, $D(1,0) = 1$, $D(1,1) = 1$ a $D(0,0) = 0$. Asociativní a komutativní fuzzy disjunkce je nazývána s -norma. Mezi s -normy patří např. sjednocení. Rovnice níže definují maximum (3-5), pravděpodobnostní (3-6), Łukasiewiczovu (3-7) a drastickou (3-8) disjunkci.

$$D_{\text{Maximum}}(a, b) = a \vee b \quad (3-5)$$

$$D_{\text{Probabilistic}}(a, b) = a + b - a \cdot b \quad (3-6)$$

$$D_{\text{Łukasiewicz}}(a, b) = 1 \wedge (a + b) \quad (3-7)$$

$$D_{\text{Drastic}}(a, b) = \begin{cases} a & b = 0 \\ b & a = 0 \\ 1 & a \neq 0 \wedge b \neq 0 \end{cases} \quad (3-8)$$

Fuzzy implikace $I : \langle 0,1 \rangle \times \langle 0,1 \rangle \rightarrow \langle 0,1 \rangle$ je operace, pro kterou platí vztahy $I(0,1) = 1$, $I(0,0) = 1$, $I(1,1) = 1$ a $I(1,0) = 0$. Fuzzy implikace Gödelova, Goguenova, Łukasiewiczova, Zadehova a standardní odpovídají rovnicím (3-9) až (3-13).

$$I_{\text{Gödel}}(a, b) = \begin{cases} 1 & a \leq b \\ b & a > b \end{cases} \quad (3-9)$$

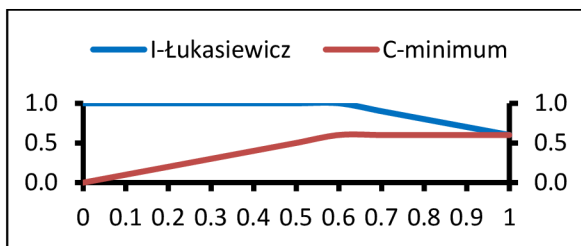
$$I_{\text{Goguen}}(a, b) = \begin{cases} 1 & a \leq b \\ b/a & a > b \end{cases} \quad (3-10)$$

$$I_{\text{Łukasiewicz}}(a, b) = 1 \wedge (b - a + 1) \quad (3-11)$$

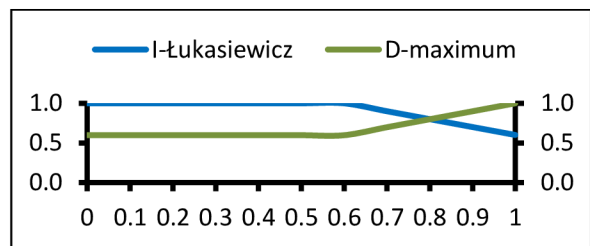
$$I_{\text{Zadeh}}(a, b) = (1 - a) \vee (a \wedge b) \quad (3-12)$$

$$I_{\text{Standard}}(a, b) = \begin{cases} 1 & a \leq b \\ 0 & a > b \end{cases} \quad (3-13)$$

Fuzzy konjunkci, disjunkci a implikaci je při vhodné kombinaci možno použít ke kompozici fuzzy relací. Tímto termínem budeme v dalším textu označovat fuzzy operace mezi maticemi vycházející z klasického násobení matic. Zatímco prvky výsledné matice při násobení jsou součiny, pro fuzzy kompozici jsou prvky výsledkem kombinace různých fuzzy operací. Zbývá vysvětlit, co je myšleno vhodnou kombinací. Podle Valleho a Sussnera [12] je pro fuzzy neuronové sítě důležité, aby operace použité při kompozici matic byly sdružené či duální z hlediska fuzzy negace. Platí-li $F(a, b) = \overline{G(\bar{a}, \bar{b})}$ a $G(a, b) = \overline{F(\bar{a}, \bar{b})}$, pak říkáme, že operace F a G jsou duální z hlediska fuzzy negace. V literatuře se lze setkat s ekvivalentním zápisem $\overline{F(a, b)} = G(\bar{a}, \bar{b})$ a $\overline{G(a, b)} = F(\bar{a}, \bar{b})$. Dvojice $(C_{Minimum}, D_{Maximum})$, $(C_{Product}, D_{Probabilistic})$, a $(C_{Łukasiewicz}, D_{Łukasiewicz})$ jsou dvojice duálních operací. Aby byly operace F a G sdružené, musí platit vztah $F(x) \leq y \Leftrightarrow x \leq G(y), \forall x, y \in <0,1>$. Sdruženými operátory jsou například $(C_{Minimum}, I_{Gödel})$ a $(C_{Minimum}, I_{Łukasiewicz})$. Operátory $(D_{Maximum}, I_{Łukasiewicz})$ sdružené nejsou. Na obrázcích 3-2 a 3-3 je znázorněn průběh funkcí $C_{Minimum}(x, z)$ a $I_{Łukasiewicz}(y, z)$ pro $z = 0.6$, z něhož je patrné, že funkce v 3-2 splňují definici soudružnosti a funkce v 3-3 nikoliv, protože pro $x = 0.9$ a $y = 0.9$ je $D_{Maximum}(0.9, 0.6) = 0.9$ a $I_{Łukasiewicz}(0.9, 0.6) = 0.7$ a výrok o soudružnosti po dosazení $(0.9 \leq 0.9 \Leftrightarrow 0.9 \leq 0.7)$ neplatí.



Obrázek 3-2: Sdružené operátory



Obrázek 3-3: Nesdružené operátory

3.1.2 Fuzzy relace

N -ární fuzzy relace R ostrých množin C_1 až C_N je fuzzy podmnožina kartézského součinu $C_1 \times C_2 \times \dots \times C_N$ tedy $R = \{(c_1, c_2, \dots, c_N), m_R(c_1, c_2, \dots, c_N) \mid c_1 \in C_1, c_2 \in C_2, \dots, c_N \in C_N\}$.
Příklad:

Jsou-li $C = \{Tom, Dick\}, D = \{John, Jim\}$ množiny obsahující každá dvě osoby a R je relace např. podobnosti obličejů. Funkci příslušnosti m_R relace R můžeme zapsat jako matici:

	John	Jim
Tom	$m_R(c_1, d_1)$	$m_R(c_1, d_2)$
Dick	$m_R(c_2, d_1)$	$m_R(c_2, d_2)$

Každý prvek této matice představuje stupeň příslušnosti $m_R(c, d), c \in C, d \in D$.

Mějme relaci R nad množinami C a D a relaci S mezi množinami D a E , max-min kompozice těchto relací je $R \circ S = \{(c, e), m_{R \circ S}(c, e) = \bigvee_D (m_R(c, d) \wedge m_S(d, e)) \mid c \in C, d \in D, e \in E\}$, kde \bigvee_D je maximum přes prvky množiny D . Tento vztah se dá chápat jako násobení matic obou relací, kde sčítání a násobení můžeme nahradit operacemi \vee a \wedge .

Max-min kompozici relací R a S , ukážeme na příkladu [11]:

$$R = \begin{bmatrix} 0.3 & 0.8 \\ 0.6 & 0.9 \end{bmatrix} \text{ a } S = \begin{bmatrix} 0.5 & 0.9 \\ 0.4 & 1.0 \end{bmatrix}.$$

$$\begin{aligned} R \circ S &= \begin{bmatrix} m_R(c_1, d_1) & m_R(c_1, d_2) \\ m_R(c_2, d_1) & m_R(c_2, d_2) \end{bmatrix} \circ \begin{bmatrix} m_S(d_1, e_1) & m_S(d_1, e_2) \\ m_S(d_2, e_1) & m_S(d_2, e_2) \end{bmatrix} \\ &= \begin{bmatrix} m_R(c_1, d_1) \wedge m_S(d_1, e_1) \vee m_R(c_1, d_2) \wedge m_S(d_2, e_1) & m_R(c_1, d_1) \wedge m_S(d_1, e_2) \vee m_R(c_1, d_2) \wedge m_S(d_2, e_2) \\ m_R(c_2, d_1) \wedge m_S(d_1, e_1) \vee m_R(c_2, d_2) \wedge m_S(d_2, e_1) & m_R(c_2, d_1) \wedge m_S(d_1, e_2) \vee m_R(c_2, d_2) \wedge m_S(d_2, e_2) \end{bmatrix} \\ &= \begin{bmatrix} 0.4 & 0.8 \\ 0.5 & 0.9 \end{bmatrix} \end{aligned}$$

Podobně jako max-min kompozici můžeme definovat max-product kompozici. Rozdíl je v operaci \wedge , která je zde nahrazena násobením [11].

Uvedme další příklad. Tentokrát na max-product kompozici pro relace R a S z předcházejícího příkladu:

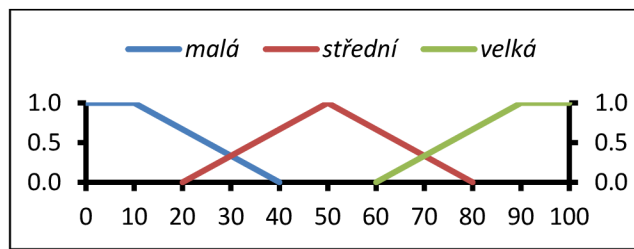
$$\begin{aligned} R \circ S &= \begin{bmatrix} m_R(c_1, d_1) & m_R(c_1, d_2) \\ m_R(c_2, d_1) & m_R(c_2, d_2) \end{bmatrix} \circ \begin{bmatrix} m_S(d_1, e_1) & m_S(d_1, e_2) \\ m_S(d_2, e_1) & m_S(d_2, e_2) \end{bmatrix} \\ &= \begin{bmatrix} m_R(c_1, d_1) \cdot m_S(d_1, e_1) \vee m_R(c_1, d_2) \cdot m_S(d_2, e_1) & m_R(c_1, d_1) \cdot m_S(d_1, e_2) \vee m_R(c_1, d_2) \cdot m_S(d_2, e_2) \\ m_R(c_2, d_1) \cdot m_S(d_1, e_1) \vee m_R(c_2, d_2) \cdot m_S(d_2, e_1) & m_R(c_2, d_1) \cdot m_S(d_1, e_2) \vee m_R(c_2, d_2) \cdot m_S(d_2, e_2) \end{bmatrix} \\ &= \begin{bmatrix} 0.32 & 0.8 \\ 0.36 & 0.9 \end{bmatrix} \end{aligned}$$

V dalším textu budeme kompozice z předchozích dvou příkladů označovat symboly \circ_\wedge a \circ_* . Obě tyto operace spadají do obecnější kategorie operací tzv. max- C kompozic \circ_C , kde C představuje libovolnou konjunkci definovanou výše. Podobně zavedeme pro min- D kompozici znak \bullet_D a pro min- I kompozici znak \ast_I . Opět lze za D popř. I dosadit libovolnou disjunkci resp. implikaci. Tyto operace jsou důležité, protože popisují činnost fuzzy neuronů, o kterých bude řeč v kapitole č. 4.

3.2 Fuzzy řízení

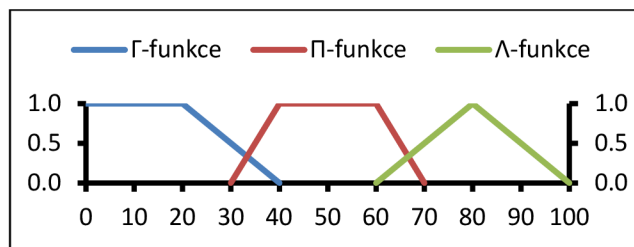
Chování fuzzy řadiče je popsáno fuzzy pravidly nad lingvistickými proměnnými, proto tyto dva pojmy vysvětlíme blíže. Lingvistická proměnná je taková proměnná, jejíž hodnoty jsou výrazy

přirozeného jazyka neboli termy. Např. lingvistická proměnná *chyba* definovaná na obrázku 3-4 může nabývat hodnot z množiny {malá, střední, velká}.



Obrázek 3-4: Lingvistická proměnná *chyba*

Každý z těchto termů je reprezentován fuzzy množinou definovaných na stejném univerzu. V našem případě je univerzum tvořeno intervalem $\langle 0, 100 \rangle$. Proces převodu hodnot z univerza do fuzzy množin se nazývá fuzzifikace, k čemuž se v oboru řízení využívají standardní funkce příslušnosti. Nejčastěji se jedná o trojúhelníkovou (Λ), lichoběžníkovou (Π) nebo rampovou (Γ) funkci.



Obrázek 3-5: Různé funkce příslušnosti

Typicky se termy označují jednotným způsobem, jak ukazuje tabulka 3-2 převzata z [13].

Hodnota	Zkratka Cze.	Zkratka Eng.
velká záporná	ZV	NB
střední záporná	ZS	NM
malá záporná	ZM	NS
záporná blízká nule	ZN	NZ
nulová	NU	Z
kladná blízká nule	KN	PZ
malá kladná	KM	PS
střední kladná	KS	PM
velká kladná	KV	PB

Tabulka 3-2: Typické názvy pro termy

Fuzzy rozhodovací pravidla se zapisují ve tvaru *if* <antecedent> *then* <konsekvent> a vyjadřují implikaci. Antecedent i konsekvent obsahují složené výroky, ve kterých se mohou vyskytovat logické spojky AND (konjunkce), OR (disjunkce) a NOT (negace). Jednoduché výroky mezi spojkami vyjadřují, že určitá proměnná nabývá konkrétní hodnotu. Zapisováno ve tvaru <proměnná> *is* <hodnota>. Např. pravidlo:

if chyba is malá and rychlost is nulová then zrychlení is nulové

řídícímu systému říká, že nemá dál zrychlovat, pokud se řízený systém nepohybuje a chyba je malá. Ostrá hodnota zrychlení na výstupu ale nula být nemusí. Vše zaleží na stupních příslušnosti jednotlivých termů a na konkrétní fuzzy implikaci. Fuzzy implikaci lze použít celá řada, např. ty z kapitoly 3.1.1. Nicméně nejčastěji se používá Mamdaniho implikace (3-14), která je ve výčtu implikací vynechána, jelikož z pohledu logiky o implikaci nejde, avšak splňuje předpoklad, že konsekvent nemůže mít větší stupeň příslušnosti než antecedent. Toto tvrzení platí i pro další často využívanou implikaci, a to Larsenovu (3-15).

$$I_{Mamdani}(a, b) = a \wedge b \quad (3-14)$$

$$I_{Larsen}(a, b) = a \cdot b \quad (3-15)$$

Obsahuje-li řadič více pravidel, několik z nich může být s různou silou aktivních současně. Aktivní pravidlo je takové, jehož výstupní fuzzy množina zůstala po provedení implikaci neprázdná. Na základě aktivních pravidel je třeba vyhodnotit výstupní ostré hodnoty. Nejprve jsou během kroku agregace všechny výstupní fuzzy množiny aktivních pravidel v rámci lingvistických proměnných sjednoceny. Posledním krokem je tzv. defuzzifikace. Některé z metod defuzzifikace jsou uvedeny v tabulce (3-3), kde x^* je ostrá výstupní hodnota, zápis m_i znamená stupeň příslušnosti bodu x_i a je ekvivalentní se zápisem $m(x_i)$.

Metody nejvýznamnějšího maxima		
levé maximum	$x^* = x_1$	<p>Obrázek 3-6: Nejvýznamnější maxima</p>
střední maximum	$x^* = \frac{x_1 + x_2}{2}$	
pravé maximum	$x^* = x_2$	
Metody těžiště		
těžiště singletonů	$x^* = \frac{\sum_i^n m_i \cdot x_i}{\sum_i^n m_i}$	<p>Obrázek 3-7: Těžiště singletonů</p>
těžiště plochy	$x^* = \frac{\int x \cdot m(x) dx}{\int m(x) dx}$	

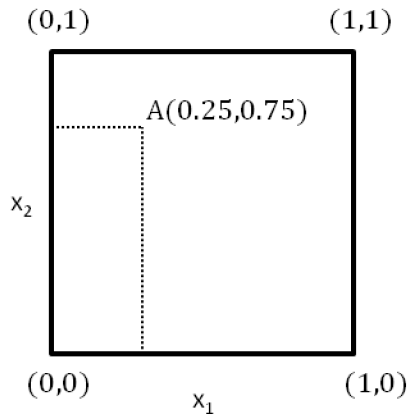
Tabulka 3-3: Metody defuzzifikace

Fuzzy řízení můžeme tedy shrnout do tří kroků. Těmi jsou fuzzifikace, inference a defuzzifikace, přičemž inference v sobě zahrnuje vyhodnocení pravidel a agregaci výstupů [13].

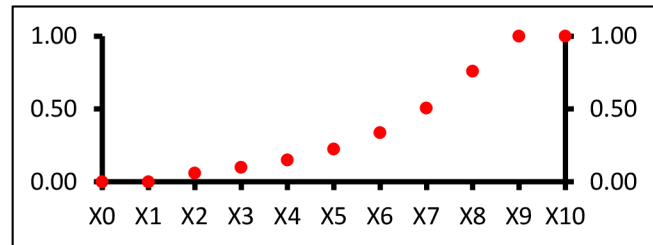
3.3 Geometrické pojetí fuzzy množin

Jiný způsob jak můžeme pohlížet na fuzzy množiny, nabízí [14]. Máme-li konečné univerzum X o n prvcích, můžeme potenční fuzzy množinu (množinu všech fuzzy podmnožin na univerzu X) 2^X zobrazit jako jednotkovou n -dimenzionální hyperkrychli I^n . Fuzzy množina A na univerzu X pak odpovídá bodu v I^n . Souřadnice tohoto bodu jsou funkční hodnoty funkce příslušnosti m_A a nazývají se vektorem příslušnosti. Tento pohled na fuzzy množiny zde uvádíme proto, že lépe podchycuje činnost fuzzy asociativních pamětí. Matematicky je zápis pomocí funkce příslušnosti ekvivalentní s vektorovým zápisem. Ten je ale omezen pouze pro diskrétní fuzzy množiny. Celý princip je ilustrován na obrázku č. 3-8 s konkrétními hodnotami, kde $X = \{x_1, x_2\}$, $A = (\{x_1, x_2\}, (0.25, 0.75))$. Potenční množina univerzu X je $2^X = \{\emptyset, \{x_1\}, \{x_2\}, \{x_1, x_2\}\}$. Tyto čtyři ostré množiny z 2^X odpovídají fuzzy množinám $(\{x_1, x_2\}, (0, 0))$, $(\{x_1, x_2\}, (1, 0))$, $(\{x_1, x_2\}, (0, 1))$ a $(\{x_1, x_2\}, (1, 1))$ a tvoří vrcholy hyperkrychle, která je ve dvourozměrném případě čtvercem. Fuzzy množina A leží na

souřadnicích $(0.25, 0.75)$. Na obrázku 3-9 je zobrazena fuzzy množina $B = (\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}, (0, 0, 0.06, 0.1, 0.15, 0.225, 0.338, 0.506, 0.759, 1, 1))$, která je definována na deseti dimenzionální hyperkostce.



Obrázek 3-8: Fuzzy množina jako bod v prostoru



Obrázek 3-9: Příklad deseti prvkové fuzzy množiny B

4 Fuzzy neuronové sítě

4.1 Klasifikace fuzzy neuronových sítí

Fuzzy neuronové sítě lze klasifikovat několika způsoby dle různých kritérií. Liu a Li [2] dělí neuronové sítě podle obsažené míry neostrosti (*fuzziness*) na sítě založené na fuzzy operátorech, fuzzifikované neuronové sítě a na fuzzy inferenční neuronové sítě.

Nejstarším typem jsou neuronové sítě založené na fuzzy operátorech, jakými jsou např. max-min nebo max-product kompozice, které představil Kosko v asociativních pamětech a definoval tak fuzzy asociativní paměti (FAM). Mezi sítě tohoto typu patří mimo jiné fuzzy Hopfield, fuzzy obousměrná asociativní paměť a fuzzy ART. Protože FAM mají schopnost ukládat a vybavovat fuzzy informace, soustředí se badatelé na jejich kapacitu a chybovost. FAM našly uplatnění například v rozpoznávání vzorů [21] [15], klasifikaci [16] a zpracování signálů [17].

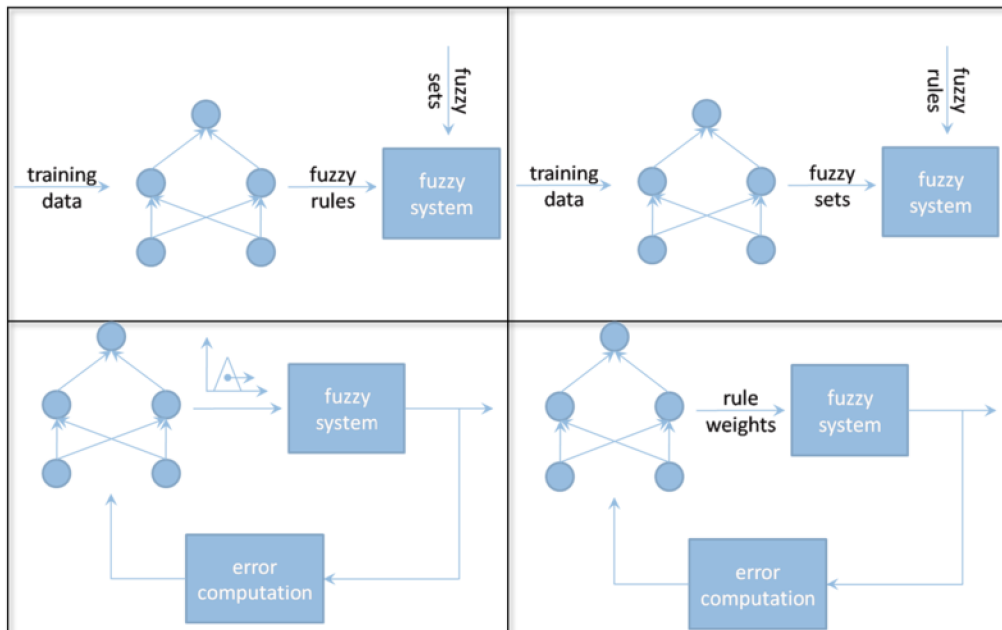
Fuzzifikované sítě jsou ty používající fuzzy neurony, jejichž váhy, vstupy i výstupy, jsou pouze fuzzy množiny. Nejznámější sítí této kategorie je regulární fuzzy neuronová síť. Tato síť vznikne z dopředné (ostré, tedy ne fuzzy) sítě výměnou klasických neuronů za fuzzy neurony. Topologie se nemění, ale operace regulární fuzzy neuronové sítě jsou založeny na fuzzy aritmetice. Předmětem výzkumu jsou zde převážně učící algoritmy a to, zda může regulární fuzzy neuronová síť sloužit jako univerzální aproximátor spojité fuzzy funkce s určitou přesností.

Inferenční neuronové sítě jsou nejznámější skupinou. Jedná se o sítě, které jsou navrženy tak, aby funkčně odpovídali Mamdani či Takagi-Sugeno fuzzy inferenčnímu systému, ale na rozdíl od nich se umí naučit různé parametry. Není tak nutno je nastavovat ručně.

Kombinace neuronové sítě s fuzzy regulátorem je nazývána neuro-fuzzy systém [1][18]. Takový systém lze na základě vzájemného vztahu obou podsystémů rozdělit na kooperativní, souběžný a hybridní. Poslední ze jmenovaných modelů odpovídá fuzzy inferenčním neuronovým sítím z předchozího odstavce.

Kooperativní systém se skládá z neuronové sítě a fuzzy systému. Oba podsystémy pracují nezávisle. Neuronová síť se z trénovacích dat naučí parametry funkcí příslušnosti či pravidla pro fuzzy systém, který je pak používán pro řešení daného problému. K učení pravidel se např. používají samoorganizující se mapy nebo fuzzy asociativní paměti. Učení může probíhat offline, kdy je síť po naučení parametrů odpojena a dál se nepoužívá, nebo online. Při učení online vede z fuzzy systému zpětná vazba a neuronová síť upravuje parametry na základě chyby. Možné topologie kooperativních systémů jsou na obrázku 4-1. Oba systémy nahoře pracují offline. Systém vlevo dole upravuje

parametry funkcí příslušností online na základě chyby. Pravidla a počáteční parametry potřebuje znát před spuštěním stejně jako systém dole vpravo. Ten optimalizuje váhy pravidel.



Obrázek 4-1: Topologie kooperativních systémů. Převzato z [1]

U souběžných systémů je neuronová síť předřazena před fuzzy systém. Zde však neuronová síť neslouží k učení parametrů fuzzy systému, nýbrž k předzpracování jeho vstupních dat. Zapojení může být i obrácené, tedy takové, kdy je fuzzy systém před sítí. K předzpracování vstupních dat je potom použit fuzzy systém.

4.2 Fuzzy neuron

Zatímco klasický neuron s lineární bázovou funkcí provádí skalární součin vstupního vektoru a vektoru vah, tedy součet součinů, fuzzy neuron aplikuje kombinaci operací z kapitoly *Operace nad fuzzy množinami*. Fuzzy neuron se skládá z n vstupů x , z n vstupních vah w , z vnitřního potenciálu u , z prahu θ a z jednoho výstupu y a jeho typ je definován operacemi, které neuron provádí mezi vstupem \vec{x} a váhami \vec{w} , popř. volitelnou operací mezi prahem θ a vnitřním potenciálem u . Výpočet výstupu obecného fuzzy neuronu bez prahu definují rovnice (4-1) a (4-2), kde \mathcal{S} a \mathcal{s} jsou libovolné s-normy a \mathcal{T} a \mathcal{t} libovolné t-normy. V konfiguraci $\mathcal{S}\mathcal{t}$ se jedná o tzv. OR-neuron, kdežto v konfiguraci $\mathcal{T}\mathcal{s}$ jde o AND-neuron.

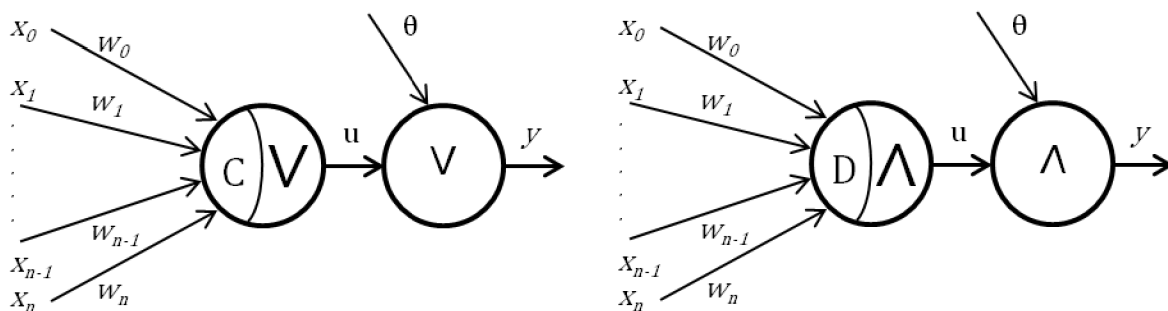
$$y = \mathcal{S}_{i=1}^n \mathcal{t}(w_i, x_i) \quad (4-1)$$

$$y = T_{i=1}^n s(w_i, x_i) \quad (4-2)$$

Pro účely fuzzy asociativních pamětí se nejvíce používají max- C a min- D neurony, které jsou konkrétními případy OR a AND-neuronů vycházející z rovnic (4-1) a (4-2) [12]. Jak bylo zmíněno v kapitole 3.1.1, kombinaci těchto operací můžeme využít ke kompozici matic. Následující rovnice ukazují operace prováděné max- C (4-3) a min- D (4-4) neurony s přidaným prahem.

$$y = \left(\bigvee_{i=1}^n C(w_i, x_i) \right) \vee \theta \quad (4-3)$$

$$y = \left(\bigwedge_{i=1}^n D(w_i, x_i) \right) \wedge \theta \quad (4-4)$$



Obrázek 4-2: Max- C a min- D fuzzy neurony

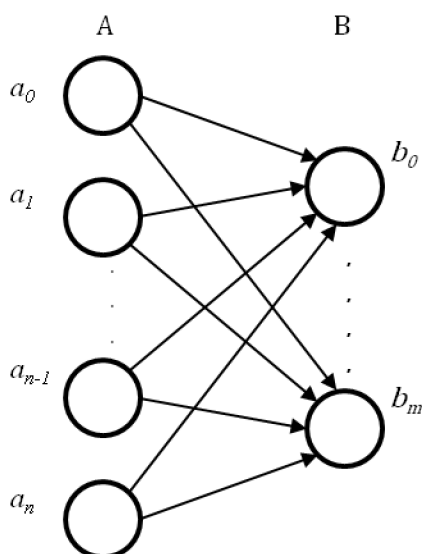
Mimo dva výše zmíněné modely se ve FAM používají i min- I neurony, které jsou známy jako implikativní fuzzy neurony. Implikace není komutativní, proto je důležité v rovnici (4-5) dodržet pořadí operandů.

$$y = \left(\bigwedge_{i=1}^N I(x_i, w_i) \right) \wedge \theta \quad (4-5)$$

4.3 Fuzzy asociativní paměť

Z matematického pohledu lze FAM chápat jako funkci mapující fuzzy množiny na jiné fuzzy množiny neboli body v hyperkrychli I^n na body v hyperkrychli I^m . FAM tedy asociuje n -dimenzionální fuzzy množiny s m -dimenzionálními fuzzy množinami.

FAM je dvouvrstvá dopředná neuronová síť se vstupní vrstvou obsahující n neuronů a s výstupní vrstvou obsahující m neuronů. Matice vah W obsahuje $n \times m$ prvků. Volitelně můžou obě vrstvy obsahovat prahové vektory $\vec{\theta}_A$ popř. $\vec{\theta}_B$, jejichž dimenze je shodná s počtem neuronů v dané vrstvě. Vrstvy označujeme symboly A a B .



Obrázek 4-3: Topologie FAM

Jsou-li obě vrstvy stejně velké, síť se může chovat jako autoasociativní paměť. Ve zbytku kapitoly budeme bez ztráty na obecnosti předpokládat heteroasociativní přístup s tím, že autoasociativita je speciálním případem. Trénovací množina je tvořena dvojicemi vektorů (\vec{a}_k, \vec{b}_k) reprezentující fuzzy množiny (A_k, B_k) tak, jak je popsáno v kapitole *Geometrické pojetí fuzzy množin*. Tuto dvojici lze také chápat jako tzv. FAM pravidlo, kde A_k je vstupem a B_k je výstupem pravidla. Ve vztahu k fuzzy logice se jedná o lingvistické pravidlo if X is A_k then Y is B_k , kde X a Y jsou lingvistické proměnné.

Činnost těchto sítí je obdobně jako u běžných asociativních pamětí rozdělena na dvě fáze – učení a odpovídání. Zde je nutno podotknout, že pojem FAM v sobě zahrnuje celou řadu variací fuzzy asociativních pamětí, které se mohou lišit jak topologií a použitými fuzzy neurony, tak metodami učení. Některé z nich budou diskutovány dále.

Důvody vzniku různých typů FAM jsou v zásadě dvojího druhu. Některé aplikace vyžadují složitější chování (např. FAM byla použita k ovládní kloubů humanoidního robota [19]), proto se v literatuře [20] často objevují systémy, které v sobě kombinují různé FAM spolu s jinými metodami strojového učení. Druhým hlavním důvodem je snaha o zvýšení paměťové kapacity.

4.3.1 Učení

Učení FAM představuje výpočet váhové matice W a vektorů $\vec{\theta}_A$ a $\vec{\theta}_B$ na základě trénovacích dvojic vektorů (\vec{a}_k, \vec{b}_k) . Jednou z metod je fuzzy Hebbovské učení. V nejstarší variantě váhy odpovídají součinu nebo minimu z dvojice prvků, jak je ukázáno rovnicemi (4-6) a (4-7).

$$w_{ij}^k = \min(a_i^k, b_j^k) \quad (4-6)$$

$$w_{ij}^k = a_i^k \cdot b_j^k \quad (4-7)$$

Kosko [14] označuje učení operací (4-6) jako korelační kódování minimem a učení pomocí operace (4-7) je korelační kódování součinem. Sestává-li se trénovací množina $T = \{(\vec{a}_1, \vec{b}_1), \dots, (\vec{a}_p, \vec{b}_p)\}$ z p vektorů, je na každou dvojici aplikována operace (4-6) nebo (4-7). Výsledkem je p dílčích matic W_1, \dots, W_p . Tyto matice je potřeba vhodným způsobem sloučit do výsledné matice. Protože v klasických asociativních pamětech je finální matice dána součtem (někdy i průměrem) z dílčích matic a sčítání je ve fuzzy logice nahrazeno operací \vee , nabízí se myšlenka, že fungující fuzzy ekvivalent výpočtu matice W odpovídá vzorci (4-8).

$$W = \bigvee_{k=1}^p W_k \quad (4-8)$$

Avšak sám Kosko ukázal, že výsledná paměť má velmi špatnou vybavovací schopnost a že váhy naučené pomocí korelačního kódování minimem či součinem nelze bez velkých ztrát informace kombinovat mezi dvojicemi trénovací množiny. Z čehož usoudil, že každé uložené pravidlo potřebuje svoji vlastní matici (tedy svoji vlastní FAM) a vytvořil složenou architekturu FAM. S předstihem můžeme říci, že tento model trpí velkou paměťovou složitostí.

Předpokládejme, že velikost ukládané fuzzy množiny je n , což je zároveň velikost vstupní vrstvy FAM a v autoasociativním režimu i velikost vrstvy výstupní. To znamená, že matice vah bude mít $n \cdot n = n^2$ prvků. Pro p uložených asociací potřebujeme p matic. Paměťová složitost je tedy $O(p \cdot n^2)$.

Složitost $O(p \cdot n^2)$, kde p není bezvýznamná konstanta, vedla badatele k názoru, že je potřeba vhodným způsobem váhy ukládat do jedné matice tak, aby byl systém přesto schopný správně vybavovat uložené asociace. Dodejme, že kvadratickou paměťovou složitost mají všechny níže uvedené modely. Z praktického hlediska jde tedy o snížení či zrušení konstanty p .

Obecnějším modelem předchozího učení je tzv. korelační t -kódování, kde t představuje libovolnou t -normu. Chung a Lee, [22] dokázali, že pokud trénovací množina splňuje jistou vlastnost, korelační t -kódování zaručí bezchybnou odezvu fundamentální paměti. K popisu této vlastnosti budeme potřebovat definovat ortonormalitu a ortogonalitu pro fuzzy množiny: Dvě fuzzy A a B množiny jsou max- t ortogonální, pokud $\vec{a} \circ_t \vec{b}^T = 0$ tedy $t(a_i, b_j) = 0, \forall i, j$. Říkáme, že množina vektorů $\{\vec{a}_1, \dots, \vec{a}_p\}$ je max- t ortonormální, jsou-li \vec{a}_i a $\vec{a}_j \forall i, j, i \neq j$ max- t ortogonální a jsou-li $A_i, \forall i$ normální fuzzy množiny. Pro trénovací množinu $T = \{(\vec{a}_1, \vec{b}_1), \dots, (\vec{a}_p, \vec{b}_p)\}$ platí, že je-li $\{\vec{a}_1, \dots, \vec{a}_p\}$ max- t ortonormální, pak $W \circ_t \vec{a}_k = \vec{b}_k, \forall k$, tedy platí bezchybné vybavení fundamentální paměti.

Další variantou fuzzy Hebbovského učení představuje tzv. fuzzy implikativní učení. Jak již název napovídá, namísto konjunkce minimum nebo konjunkce součinu se využívají různé fuzzy implikace. Nejčastěji se jedná o Gödelovu, Goguenovu a Łukasiewiczovu implikaci. Obecně se dají popsat rovnicí (4-9) pro výpočet váhové matice a rovnicí (4-10) pro výpočet prahů. Pro FAM užívající implikativní učení bylo ukázáno, že jednotlivé váhové matice W_k lze sloučit do výsledné matice W , aniž by byla fundamentální paměť znehodnocena [23]. Proto je výpočet (4-9) uveden pro výslednou matici W . Ačkoliv korelační kódování minimum odpovídá Mamdaniho implikaci a korelační kódování součinem Larsenově implikaci, ani jedna z těchto učících metod nespadá pod implikativní učení.

$$w_{ij} = \bigwedge_{k=1}^p I(a_i^k, b_j^k) \quad (4-9)$$

$$\vec{\theta}_B = \bigwedge_{k=1}^p \vec{b}_k \quad (4-10)$$

K fuzzy implikativnímu učení existuje duální metoda tzv. duální fuzzy implikativní učení. Negací váhové matice W vznikne matice M a negace θ_B se označuje symbolem ϑ_B , jak ukazují vztahy (4-11) a (4-12).

$$m_{ij} = \overline{w_{ij}} = \overline{\bigwedge_{k=1}^p I(a_i^k, b_j^k)} = \bigvee_{k=1}^p dI(a_i^k, b_j^k) \quad (4-11)$$

$$\vartheta_{B,j} = \overline{\theta_{B,j}} = \overline{\bigwedge_{k=1}^p b_j^k} = \bigvee_{k=1}^p \overline{b_j^k} \quad (4-12)$$

Duální operace k nejčastějším implikacím jsou definovány v (4-13), (4-14) a (4-15). Ačkoliv se jedná o duální operace k implikacím, z pohledu logiky se již o implikace nejedná [23].

$$dI_{Gödel}(a, b) = \begin{cases} 0 & a \geq b \\ b & a < b \end{cases} \quad (4-13)$$

$$dI_{Göguen}(a, b) = 0 \vee \frac{b - a}{1 - a} \quad (4-14)$$

$$dI_{Łukasiewicz}(a, b) = 0 \vee (b - a) \quad (4-15)$$

Binh a Hoa [24] použili korelační kódování minimem a dílčí váhy zprůměrovali. Výpočet vah a prahů zobrazují rovnice (4-16) až (4-18). Na tento model se dále budeme odkazovat označením Binh-Hoa.

$$w_{ij}^k = \min(a_i^k, b_j^k) \quad (4-16)$$

$$w_{ij} = \frac{1}{p} \sum_{k=1}^p w_{ij}^k \quad (4-17)$$

$$\theta_{B,i} = \frac{1}{p} \bigvee_{k=1}^p b_i^k \quad (4-18)$$

Mimo Hebbovo učení existují i složitější učící algoritmy. Liu a Li [2] představili algoritmus s názvem δ -učení, který je formulován pro max-min i max-product kompozici. Nicméně zde je uvedena pouze varianta max-min odpovídající algoritmu 4-1. Varianta s max-product kompozicí se liší pouze v operaci \vee , která je zde nahrazena součinem. Algoritmus obsahuje učící parametr $\eta \in (0,1)$, jehož hodnota může být během učení postupně snižována. Definice algoritmu používá maticový zápis trénovací množiny $T = \{(\vec{a}_1, \vec{b}_1), \dots, (\vec{a}_p, \vec{b}_p)\}$, kdy jsou jednotlivé dvojice (\vec{a}_k, \vec{b}_k) rozděleny do dvou matic $\mathbf{A} = (A_1, \dots, A_p)^T$ a $\mathbf{B} = (B_1, \dots, B_p)^T$.

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{p1} & \cdots & a_{pn} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{p1} & \cdots & b_{pm} \end{bmatrix}$$

1) Inicializace:

$$t = 0;$$

$$W^0 = 0;$$

$$\eta = \langle 0, 1 \rangle;$$

2) Výpočet asociace:

$$\mathbf{B}^t = \mathbf{A} \circ_{\wedge} W^t$$

$$= b_{kj}^t = \bigvee_{i=1}^n (a_{ki} \wedge w_{ij}^t), \forall j \in \langle 1, m \rangle, \forall k \in \langle 1, p \rangle;$$

3) Úprava vah:

$$w_{ij}^{t+1} = \begin{cases} w_{ij}^t - \eta \cdot (b_{kj}^t - b_{kj}), & w_{ij}^t \wedge a_{ki} > b_{kj} \\ w_{ij}^t, & \text{otherwise} \end{cases};$$

4) Iterace:

if ($W^{t+1} == W^t$) {

$t = t + 1;$

exit();

}

else {

goto 2);

}

Algoritmus 4-1: δ -učení

Kromě δ -učení Liu a Li představili i backpropagation pro FAM. Tento algoritmus zde rozebírán nebude. Jen podotkneme, že vyžaduje, aby funkce, ze kterých je složena kompozice, byly diferencovatelné, což \vee a \wedge nejsou.

4.3.2 Odezva

Princip vybavení asociace, tedy transformace vstupní fuzzy množiny na výstupní, je pro max-C neuron popsán vztahem (4-20). Na vstupní vrstvě je přiveden vektor \vec{x} , který je po provedení volitelné prahové operace (4-19) přeposlán do druhé vrstvy, kde jsou umístěny fuzzy neurony zvoleného typu. Jejich výstup je roven vybavené asociaci \vec{y} . Tento proces není iterativní a vyžaduje pouze jeden krok. Proto FAM netrpí problémy s konvergencí.

$$\vec{a} = \vec{x} \mathbf{t} \vec{\theta}_A \quad (4-19)$$

$$\vec{y} = (W^T \circ_C \vec{a}) \vee \vec{\theta}_B \quad (4-20)$$

Konkrétní operace musí být kompatibilní s učicí metodou. Jinými slovy, různé typy fuzzy neuronů vyžadují různé učicí metody. Operace prováděné neuronem by měly být duální nebo sdružené k těm, které byly použity při učení. Např. max- C neurony je možno učit δ -učení, Binh-Hoa algoritmem i implikativním učením s implikací I_C , kde I_C a C jsou sdružené operátory. Na neurony typu min- D lze aplikovat duální implikativní učení. Odezva takového systému je (4-21).

$$\vec{y} = (M^T \bullet_D \vec{a}) \wedge \vartheta_{B,i} \quad (4-21)$$

Vstupní vrstva nemusí sloužit pouze k přeposílání hodnot. Je zde možno provést např. fuzzifikaci, jak ukazuje následující příklad:

Množina $A = (\{0,1,2,3,4,5,6,7,8,9,10\}, (0.0, 0.0, 0.4, 0.75, 0.75, 0.75, 0.4, 0.0, 0.0, 0.0, 0.0))$ je definována na univerzu přirozených čísel z intervalu $\langle 0, 10 \rangle$. Ostrá hodnota $x = 2$ je reprezentována vektorem příslušnosti $\vec{x} = [0,0,1,0,0,0,0,0,0,0]$. Vstupní vrstva FAM se skládá z jedenácti neuronů. Ve fázi učení nastavíme prahový vektor $\vec{\theta}_A$ na hodnoty z m_A , tedy $\theta_{A,i} = m_A(i), \forall i \in \langle 1, 11 \rangle$. Po přiložení vektoru \vec{x} bude podle rovnice (4-19), kde $\mathbf{t} = C_{Minimum}$, výstup neuronu a_2 roven hodnotě 0.4. Ostatní neurony zůstanou neaktivní, tedy nulové.

4.3.3 Koskova složená FAM

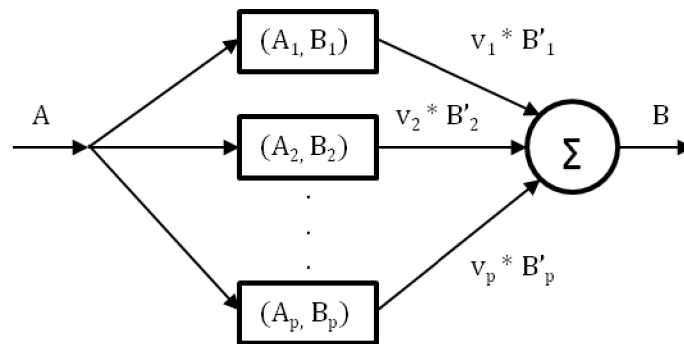
Základní komponentou složené FAM je tzv. minimální FAM. Minimální FAM využívají max- $C_{Minimum}$ nebo max- $C_{Product}$ fuzzy neurony provádějící max-min (4-22) nebo max-produkt kompozici (4-23).

$$W^T \circ_{\wedge} \vec{a} = \vec{b} \quad (4-22)$$

$$W^T \circ_{*} \vec{a} = \vec{b} \quad (4-23)$$

Složená FAM se skládá z p paralelně zapojených FAM vyvedených do výstupní vrstvy. Taková FAM kóduje skupinu p pravidel $(A_1, B_1), \dots, (A_p, B_p)$. Všechna uložená fuzzy pravidla (A_k, B_k) na každý vstup A reagují výstupem B'_k , který je podmnožinou původního B_k . Čím více je A podobné A_k , tím více je B'_k podobné B_k . Výsledná fuzzy množina B je pak například dána váženým součtem

$B = v_1 B'_1 + \dots + v_p B'_p$, kde \vec{v} je vektor vah mezi výstupní vrstvou a minimálními FAM. Schéma zapojení je ilustrováno na obrázku 4-4.

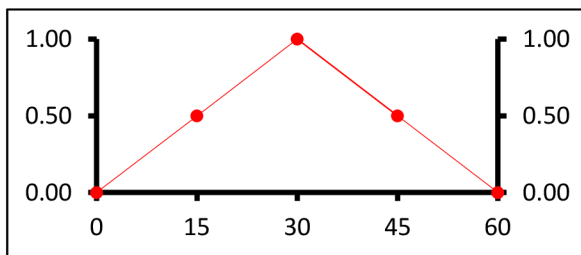


Obrázek 4-4: Topologie obecné FAM

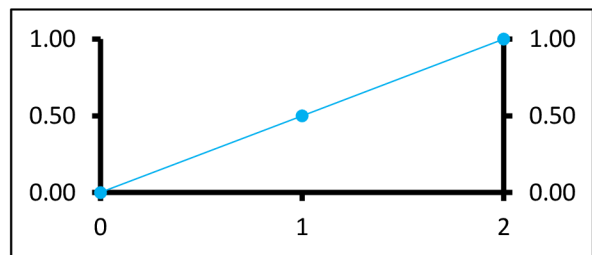
Navzdory úspěšné aplikaci při řešení řídicích problémů, je z důvodů extrémně nízké paměťové kapacity použitých metod potřeba na každé pravidlo jedna FAM, což dle mého názoru téměř vylučuje použití v oblastech, kde je potřeba ukládat velké vektory dat nebo v aplikacích vyžadující rychlou odezvu, protože každou z FAM je potřeba vyhodnocovat zvlášť.

4.3.4 BIOFAM inference

[14] Jednou z možností, jak provádět inferenci pomocí FAM, je použití fuzzy asociativní paměti s binárním vstupem a výstupem (anglicky *binary input-output fuzzy associative memory*, zkráceně BIOFAM). Topologie tohoto systému odpovídá obecné FAM a je možno použít libovolné fuzzy neurony s vhodnou učicí metodou. Pro jednoduchost budeme předpokládat, že všechny fuzzy množiny jsou konečné a jdou popsat vektory, jak je popisuje kapitola *Geometrické pojetí fuzzy množin*. Takové množiny $A = [0, 0.5, 1, 0.5, 0]$ a $B = [0, 0.5, 1]$ zadané vektorem příslušnosti je možno vidět na obrázcích 4-5 a 4-6.



Obrázek 4-5: Fuzzy množina A s pěti prvky



Obrázek 4-6: Tříprvková fuzzy množina B

Činnost jednoduchého kontroléru na bázi BIOFAM, skládající se z jedné Koskovy minimální FAM, budeme demonstrovat na příkladu.

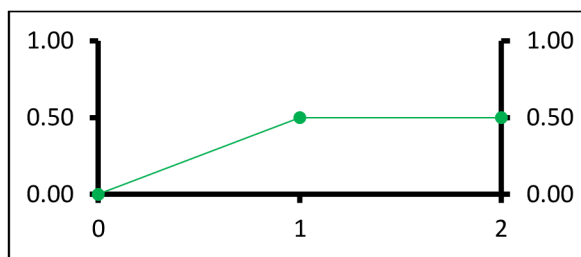
Vstupní univerzum je $X = \{0, 15, 30, 45, 60\}$, na kterém je definována fuzzy množina A . Výstupní univerzum Y je tvořeno prvky 0, 1, 2. Fuzzy množina B je definována na univerzu Y . Vytvoříme asociační pravidlo (A, B) podle rovnice 4-6.

$$W = \min(A, B) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0.5 & 1 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}$$

Po naučení je na vstupu naměřena ostrá hodnota 45. Tu reprezentuje vektor $X_{in} = [0, 0, 0, 1, 0]$. Podle pravidel max-min kompozice pak můžeme provést vybavení asociace. Výsledek je znázorněn na obrázku 4-6.

$$B' = W^T \circ_{\wedge} X_{in} = [0 \ 0.5 \ 0.5]$$

Posledním krokem je defuzzifikace. Protože se pohybujeme ve velmi malém diskrétním prostoru Y , použijeme jednoduchou metodu výběru prvku s největším stupněm příslušnosti. Těch samozřejmě může existovat více. Proto zvolíme metodu pravého maxima. Výsledný vektor bude tedy $Y_{out} = [0, 0, 1]$, což odpovídá ostré hodnotě 2.



Obrázek 4-7: Vybavená fuzzy množina

Příklad ukazuje, proč se této metodě říká binární vstup a výstup. Fuzzy množina reprezentuje ostrou hodnotu z daného univerza tak, že odpovídající souřadnice je nastavena na jedna a všechny ostatní jsou nulové.

Obecná FAM ukládá pravidla typu *if X is A then Y is B*, jak ale řešit složitější výrazy typu *X is A and Y is B then Z is C*? Odpověď je v paralelním zapojení dvou a více obecných FAM, z jejichž výstupních fuzzy množin se podle spojky v pravidle vytvoří průnik nebo sjednocení. Že je to možné z hlediska různé dimenzionality fuzzy množin, ukazuje následující odvození.

Mějme dvě FAM. První kóduje pravidlo (A, C) a druhá (B, C) . Pak dimenze fuzzy množin $A_{1 \times n}$, $B_{1 \times m}$ a $C_{1 \times k}$ jsou n , m a k . Matice vah prvního pravidla $W_{n \times k}^{AC} = A_{1 \times n}^T \circ_{\min} C_{1 \times k}$ má n řádků a k sloupců. Rozměry váhové matice $W_{m \times k}^{BC} = B_{1 \times m}^T \circ_{\min} C_{1 \times k}$ druhého pravidla jsou m řádků a k sloupců. Vstup $A_{1 \times n}^{in}$ na první FAM vyvolá odezvu $C_{1 \times k}'^{AC} = A_{1 \times n}^{in} \circ_{\wedge} W_{n \times k}^{AC}$ a odezva druhé FAM na vstup $B_{1 \times m}^{in}$ je $C_{1 \times k}'^{BC} = B_{1 \times m}^{in} \circ_{\wedge} W_{m \times k}^{BC}$. Oba výsledné vektory mají shodnou délku k , a proto je průnik $C'_{1 \times k} = C_{1 \times k}'^{AC} \cap C_{1 \times k}'^{BC}$ platnou operací.

5 Úlohy

5.1 Obnova poškozeného obrazu

Úlohu restaurace obrazu lze obecně definovat jako rekonstrukci originálního obrazu na základě poškozeného obrazu. Cílem restaurace je kompenzace či odstranění vad, které degradují digitální obraz. Degradací je většinou myšleno zkreslení, rozmazání nebo šum. Tyto chyby mohou například vzniknout při snímání, při přenosu či při zpracování obrazu. Důležitou vlastností poruch je, že je můžeme modelovat matematickou funkcí. Tradiční metody zpracování obrazu pak opravují poškozený obraz pomocí vhodných filtrů nebo aplikací inverzní funkce k funkci poruchy [25].

Pro účely této práce budeme na rozdíl od tradičních metod předpokládat, že byl originální obraz v minulosti znám a že je uložen v autoasociativní paměti. Dalším rozdílem je chápání poškození. Za poškození budeme považovat nejen šum či rozmazání, ale i posun a rotaci. Obecně tedy vše, co se liší byť jen v jediném pixelu od svého originálu. Tyto poškozené verze přikládáme na vstup paměti se pak v nejlepším případě transformují na odpovídající uložený vzor na pixel přesně. Z hlediska úlohy je tento proces převodu černou skřínkou a proto můžeme debaty o modelech chyb a další teorii zcela zanedbat, což je nespornou výhodou neuronových sítí.

5.2 Parkování kamiónu s přívěsem

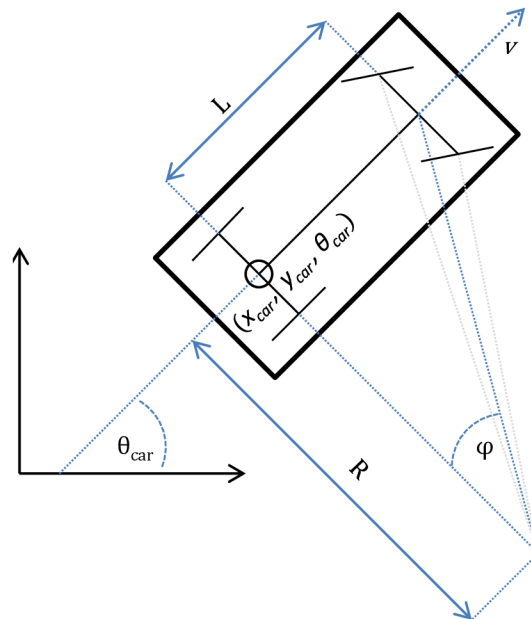
Úkolem řídicího systému je z libovolných počátečních souřadnic x_{start} , y_{start} , θ_{start} nacouvat s nákladním vozem kolmo k nákladní rampě nacházející se na pozici x_{dock} , y_{dock} , θ_{dock} . Symbolem θ budeme dále označovat orientaci. Počátek lokálního souřadného systému vozu leží uprostřed zadní osy vozu (spojnice zadních kol). Tento bod je dán globálními souřadnicemi x_{car} , y_{car} , θ_{car} , přičemž θ_{car} je úhel mezi osou x a směrovou osou vozidla, která je kolmá k zadní kolové ose. Tyto souřadnice se také označují jako stavové proměnné řízeného systému. Výstupem řídicího systému je řídicí úhel φ , tedy úhel mezi směrovou osou vozu a předními koly a rychlost v . Je-li rychlost záporná, vůz couvá. Kinematika vozu je dána rovnicemi (5-1), (5-2) a (5-3). Jedná se o zjednodušený model Ackermannova řízení [26].

$$x_{car} = v \cdot \cos(\theta_{car}) \quad (5-1)$$

$$y_{car} = v \cdot \sin(\theta_{car}) \quad (5-2)$$

$$\theta_{car} = v \cdot \frac{\tan(\varphi)}{L} \quad (5-3)$$

Tyto rovnice popisují změnu jednotlivých souřadnic v závislosti na řídicích proměnných. Parametr L je vzdálenost mezi zadní a přední kolovou osou.



Obrázek 5-1: Kinematika nákladního vozu

Hodnoty řídicích i stavových proměnných jsou omezeny na interval $\langle min, max \rangle$, kde min . je minimální možná hodnota a max . je maximální možná hodnota. Cílem těchto omezení je přiblížit úlohu reálnému světu. Avšak podotkněme, že na rozdíl od reálného světa systém zanedbává zrychlení, stejně tak jako vztahy z dynamiky. Vůz tedy nemá setrvačnost a nepůsobí na něj žádné síly.

6 Realizace

Cílem této kapitoly je popsat návrh a implementaci fuzzy asociativní paměti pro zmíněné problémy. Důvod výběru dvou úloh k řešení je ten, že první úloha „oprava obrazu“ má sloužit jako *benchmark* výše zmíněných modelů asociativních pamětí, na jehož základě je po zhodnocení výsledků vybrán jeden konkrétní model a použit k řešení řídicí úlohy. Výsledky jsou prezentovány v kapitole 7.

Jako programovací jazyk byl zvolen jazyk C++. Testovací konzolová aplikace pro opravu obrazu využívající knihovnu *OpenCV* je navržena s důrazem na možnosti konfigurace pro různé experimenty a na vyhodnocování použitých metrik. Udržovatelnost a přehlednost kódu zde nebyly prioritou. Naproti tomu aplikace realizující simulátor disponuje grafickým uživatelským rozhraním implementovaným ve *WinAPI* a je navržena objektivně tak, aby byla snadno rozšiřovatelná.

6.1 Fuzzy asociativní paměť pro restauraci obrazu

Jako vstupní vektor příznaků jsou použity celé matice reprezentující obrázky o velikosti 46x56, z čehož plyne, že vstupní i výstupní vrstva (jedná se o autoasociaci) mají 2576 neuronů. Vstupní vrstva provádí fuzzifikaci, a to tak, že dělí vstupní 8 bitové pixely hodnotou 255. Tímto způsobem vznikne fuzzy množina interpretovatelná jako "*množství bíle barvy v pixelu*". Během defuzzifikace se výstup FAM stejnou hodnotou násobí.

Implementované modely jsou metoda Binh-Hoa a modely založené na fuzzy implikativním a fuzzy duálně implikativním učení. Jmenovitě se jedná o Gödelovu a Łukasiewiczovu implikaci a jejich duální protějšky. V dalším textu budeme tyto modely nazývat zkráceně Gödel, Łukasiewicz, duální Łukasiewicz a duální Gödel. Tyto modely se po počátečních pokusech jeví z hlediska paměťové kapacity a schopnosti opravovat jako nejperspektivnější. Dále pak testovací aplikace nabízí možnost konfigurovatelné FAM, kde je možno volit různé prahové i neuronové operace, které mají být prováděny. Na této konfigurovatelné FAM bylo vyzkoušeno více modelů. Avšak v porovnání s výše zmíněnými si nevedly příliš dobře a nemělo význam je zahrnovat do experimentů. Za účelem porovnání výsledků byla také vytvořena Hopfieldova síť. Protože se jedná o binární / bipolární síť, jsou jednotlivé bity pixelů rozděleny po bytech do vstupního vektoru, který tedy má 8x46x56 prvků.

6.2 Parkování vozu

6.2.1 Simulátor

Pro testování řídicího systému byl vytvořen simulátor s grafickým uživatelským rozhraním. Simulace počítá aktuální polohu vozu na základě rychlosti a řídicího úhlu podle rovnic (5-1), (5-2) a (5-3). Protože rychlost je definována jako změna polohy v čase, je nutno do implementace rovnic zahrnout i čas t . Stěžejním bodem simulace je metoda $step()$ periodicky realizující výpočet aktuální pozice vozu na základě hodnot řídicích proměnných. Metoda je popsána algoritmem 6-1, kde Δt udává uplynulé množství času od posledního volání funkce.

```
step( $\Delta t$ )
{
     $v_x = v \cdot \cos(\theta_{car});$ 
     $v_y = v \cdot \sin(\theta_{car});$ 
     $v_\theta = v \cdot \frac{\tan(\varphi)}{L};$ 
     $x_{car} = x_{car} + v_x \cdot \Delta t;$ 
     $y_{car} = y_{car} + v_y \cdot \Delta t;$ 
     $\theta_{car} = \theta_{car} + v_\theta \cdot \Delta t;$ 
}
```

Algoritmus 6-1: Metoda pro aktualizaci pozice

Simulační prostředí je určeno několika pevnými parametry, které ukazuje tabulka 6-1. Jelikož nákladní vůz není hmotný bod, jsou v tabulce zaneseny i rozměry vozidla w_{car} , h_{car} . Jednotky veličin pro naše použití nejsou podstatné, proto jsou zde hodnoty uváděny bezrozměrně. (Z hlediska uživatelského rozhraní se jednotky jeví jako pixely popř. pixely za sekundu, avšak interně jsou veličiny reprezentovány reálnými čísly.) To neplatí pro orientaci a řídicí úhel, které jsou ve stupních.

Proměnná	Hodnota
L	44
(w_{car}, h_{car})	(52,14)
$(x_{dock}, y_{dock}, \theta_{dock})$	(250, 490, 270°)
$\langle x_{min}, x_{max} \rangle$	$\langle 0, 500 \rangle$
$\langle y_{min}, y_{max} \rangle$	$\langle 0, 500 \rangle$
$\langle \theta_{min}, \theta_{max} \rangle$	$\langle 0, 359 \rangle$
$\langle \varphi_{min}, \varphi_{max} \rangle$	$\langle -35^\circ, 35^\circ \rangle$
$\langle v_{min}, v_{max} \rangle$	$\langle -40, 40 \rangle$

Tabulka 6-1: Parametry simulace

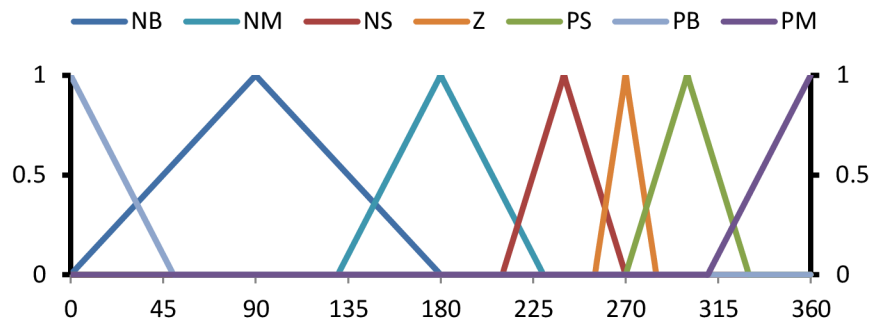
Simulace je považována za ukončenou poté, co se vůz dotkne rampy nebo narazí na hranice prostoru, které jsou vymezeny hodnotami $\langle x_{min}, x_{max} \rangle$ a $\langle y_{min}, y_{max} \rangle$ z tabulky. Pro tento účel byl implementován jednoduchý algoritmus detekce kolizí využívající faktu, že hranice jsou rovnoběžné s osami x a y .

Uživatelské rozhraní umožňuje pomocí myši měnit počáteční polohu a orientaci vozidla $x_{start}, y_{start}, \theta_{start}$. Polohu nákladní rampy měnit nelze. Kromě polohování vozu pomocí myši dovoluje uživatelské rozhraní emulovat funkci kontroléru pomocí klávesnice. Protože řídicí i stavové proměnné jsou v pravidelných intervalech snímkovány, je možno manuální ovládání použít ke generování numerických dat. Numerická data pak mohou sloužit k učení kontroléru, jak bude ukázáno v podkapitole 6.2.4.

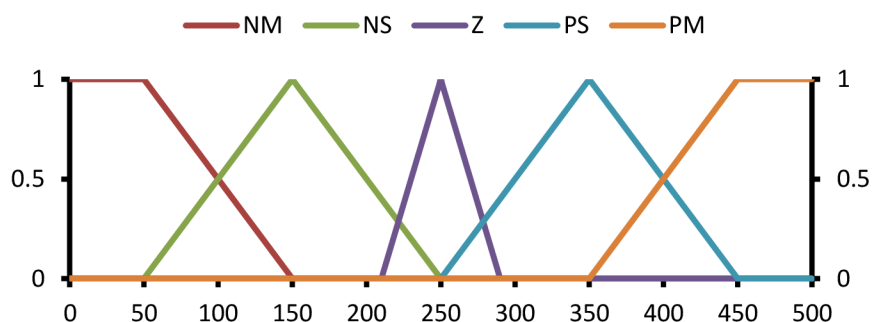
6.2.2 Fuzzy regulátor

Jako referenční řídicí systém byl implementován fuzzy regulátor, jehož vstup tvoří proměnné x_{car} a θ_{car} a výstup proměnná φ . Systém zanedbává ypsilonovou souřadnici y_{car} , čímž je znemožněno úspěšné zaparkování z některých startovních pozic. Protože se reguluje pouze řídicí úhel a rychlost v je pevně nastavena na hodnotu -40, vůz umí pouze couvat, a proto i nevhodně zvolená počáteční orientace předem odsoudí pokus k neúspěchu. Jelikož se jedná o referenční systém sloužící k validaci kontroléru založeného na fuzzy asociativní paměti, toto omezení je pro práci přijatelné.

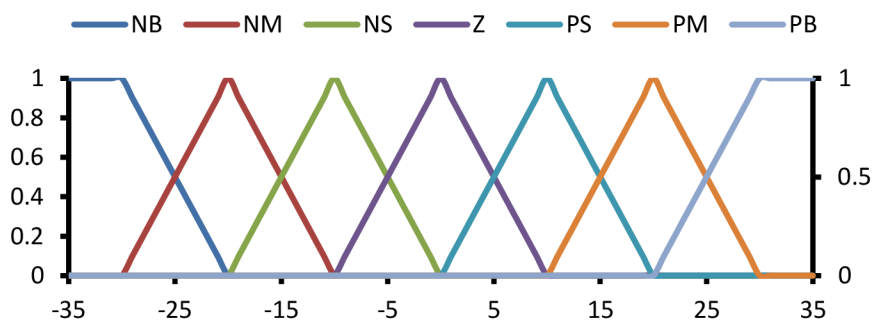
Lingvistické proměnné jsou složeny z trojúhelníkových a rampových fuzzy množin, které je možno vidět na obrázcích 6-1 až 6-3. Spojitá univerza těchto termů jsou tvořena odpovídajícími intervaly z tabulky 6-1. Podotkněme, že ačkoliv jsou intervaly vstupních proměnných kladné, některé termy nesou záporné názvy. Jedná se o pozůstatek původního návrhu, který svoje vstupy počítal jako rozdíl mezi souřadnicemi vozu a nákladní rampy. Tento postup se ukázal jako zbytečný resp. nepřinesl žádné výhody a proto byl nahrazen stávající metodou.



Obrázek 6-1: Vstupní lingvistická proměnná θ_{car}



Obrázek 6-2: Vstupní lingvistická proměnná x_{car}



Obrázek 6-3: Výstupní lingvistická proměnná φ

Zvolená pravidla jsou tvaru *if x_{car} is A and θ_{car} is B then φ is C*. Všechna 35 pravidel je ukázáno v tabulce 6-2, jejíž buňky odpovídají termům z konsekventu pravidla. Uvedené zkratky termů a jejich význam je možno najít v tabulce 3-2.

x_{car}/θ_{car}	NB	NM	NS	Z	PS	PM	PB
NM	NB	NM	PB	PB	PB	PB	PB
NS	NB	NB	PS	PM	PB	PM	PM
Z	NB	NB	NB	Z	PB	PB	PB
PS	NM	NM	NB	NM	NS	PB	PB
PM	NB	NB	NB	NB	NB	PM	PB

Tabulka 6-2: Pravidla pro fuzzy regulaci

K vyhodnocení pravidel je použita Mamdaniho implikace a pro defuzzifikaci metoda těžistiště singletonů.

6.2.3 FAM systém jako regulátor

V této podkapitole bude představen regulátor realizovaný pomocí fuzzy asociativní paměti. Návrh vychází z předpokladu, že FAM systém umí asociovat fuzzy množinu reprezentující předpoklad pravidla s fuzzy množinou tvořící konsekvent pravidla, a že lze tuto asociaci vybavit správně na základě neúplného vstupu (vstup tvoří jedna hodnota, ale ukládány jsou celé vektory). Tento předpoklad je podpořen výkladem uvedeným v teoretických kapitolách této práce a zejména pak teorií z podkapitol *Koskova složená FAM* a *BIOFAM inference*. Jelikož jsme zde psali o nedostatecích Koskova modelu, pokusili jsme se přijít s vlastním řešením, které se tyto nedostatky pokouší zmírnit.

Lingvistické proměnné a jejich fuzzy množiny jsou stejné, jako používá fuzzy regulátor z předchozí podkapitoly, avšak liší se v reprezentaci. FAM používá k reprezentaci vektory příslušnosti, jejichž velikost ukazuje tabulka 6-3.

proměnná	velikost vektoru příslušnosti
x_{car}	501
θ_{car}	360
φ	71

Tabulka 6-3: velikost vektorů

Připomeňme, že univerzum fuzzy množin je spojitě a že zde na rozdíl od tradiční reprezentace dochází k diskretizaci, a tedy ztrátě informace. Algoritmus 6-2 popisuje převod funkce příslušnosti na vektor.

```

createMembershipVector(
    TriangularFunction  $\Lambda_X$ ,
    membershipVector_X[N]
){
    for(int i = 0; i < N; ++i){
        membershipVector_X[i] =  $\Lambda_X(i + \text{min}_X)$ ;
    }
}

```

Algoritmus 6-2: Výpočet vektoru příslušnosti

Učící metodou je fuzzy implikativní učení založené na Łukasiewiczově implikaci. Tento model, zvolen na základě výsledků předchozí úlohy (viz kapitola 7), nám umožní použít jednu FAM pro více pravidel. Trénovací množina obsahuje dvojice reprezentující jednotlivá pravidla ve tvaru (\vec{a}, \vec{b}) . Aby bylo možno uložit složený výrok v předpokladu do jedné FAM, což koncept BIOFAM z kapitoly 4.3.4 neumožňuje, vstupní vektor \vec{a} vznikne konkatenací vektorů příslušnosti fuzzy množin proměnných x_{car} a θ_{car} . Vektor \vec{b} odpovídá vektorům příslušnosti proměnné φ .

Z výše uvedeného plyne, že vrstva A obsahuje 861 neuronů a vrstva B 71 neuronů. Neurony ve vrstvě B jsou typu max- $C_{Łukasiewicz}$. Prahový vektor $\vec{\theta}_A$ je použit pro fuzzifikaci, jak je ukázáno v kapitole 4.

Metoda pro výpočet odezvy vychází z upravené BIOFAM inference. Protože vstupní vrstva je konkatenací dvou proměnných, vstupní vektory obsahují dvě ostré hodnoty na pozicích odpovídajících jednotlivým hodnotám obou proměnných. Výstupem je fuzzy vektor příslušnosti proměnné φ , jehož hodnoty jsou ořezány nejmenším stupněm příslušnosti ze vstupních hodnot. Tato operace odpovídá Mamdaniho fuzzy inferenci.

Ze způsobu fuzzifikace však vzniknul první problém. Protože fuzzy množiny se překrývají, nebylo možné správně fuzzifikovat hodnoty, které mají stupeň příslušnosti větší než nula pro dvě sousedící fuzzy množiny. Proto byly fuzzy množiny každé proměnné rozděleny na sudé a liché a byly ukládány do různých FAM. Pro 2 proměnné bylo celkem potřeba $2^2 = 4$ FAM. Bohužel se ukázalo, že ani tento počet pro správnou funkci systému nestačí. Jsou-li některá pravidla uložena ve stejné FAM, navzájem se ruší, čímž vzniká nulový výstup. Jednu takovou dvojici tvoří např. pravidla *if x_{car} is NS and θ_{car} is Z then φ is PM* a *if x_{car} is PS and θ_{car} is Z then φ is NM*. Z tohoto důvodu nejsou termy proměnné x_{car} ukládány do stejných FAM. Rozdělení termů proměnné θ_{car} na sudé a liché zůstalo. Výsledný systém tak potřebuje $5 \cdot 2 = 10$ separátních FAM.

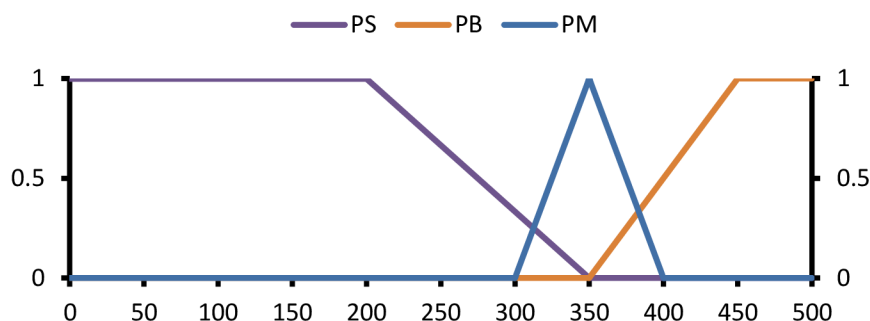
FAM jsou zapojené paralelně a jejich výstup je přiveden do vrstvy o 71 neuronech, které na svůj výstup posílají maximální hodnotu ze svých 71 vstupů. Tímto je dosaženo fuzzy sjednocení

výstupů všech FAM. Výsledná fuzzy množina je následně defuzzifikována metodou těžiště singletonů.

6.2.4 Adaptivní fuzzy asociativní regulátor

Předchozí systém byl navrhnout tak, aby funkčně odpovídal fuzzy regulátoru z podkapitoly 6.2.2. Oba dva přístupy je pak možno snadněji navzájem porovnat. S cílem demonstrovat další možnosti fuzzy asociativních pamětí byl navrhnout a vytvořen tzv. adaptivní fuzzy asociativní regulátor. Adaptivitu budeme chápat jako schopnost regulátoru učit se pravidla z numerických dat. Dodejme, že tuto schopnost má de facto i systém z kapitoly 6.2.3, avšak ten učení nepotřebuje, protože má pravidla předem pevně stanovená.

Protože je do systému nově přidána vstupní lingvistická proměnná y_{car} (obrázek 6-4), bylo potřeba provést několik změn ve struktuře. Ukládaná pravidla mají tvar *if x_{car} is A and y_{car} is B and θ_{car} is C then φ is D*, z čehož vzniká potřeba rozšířit vstupní vrstvu A o 501 neuronů, což je velikost vektoru příslušnosti proměnné y_{car} , na celkových 1362 neuronů. Termy lingvistické proměnné y_{car} také bylo nutno rozdělit na sudé a liché a ukládat je odděleně. Tento systém tak potřebuje $5 \cdot 2 \cdot 2 = 20$ FAM. Zbylé parametry zůstaly stejné.



Obrázek 6-4: Vstupní lingvistická proměnná y_{car}

Hlavní motivací pro implementaci adaptivního chování je snížení úsilí při návrhu systému. Počet všech možných pravidel, které je nutno specifikovat, je totiž dán počtem proměnných a počtem jejich termů. V našem případě se jedná o $5 \cdot 7 \cdot 3 = 105$ možných pravidel.

Metoda pro odvozování pravidel z numerických dat je popsána algoritmem 6-3. Jeden vzorek *sample* obsahuje hodnoty všech stavových i řídicích proměnných v čase t od začátku simulace. Pro každou proměnnou je nalezena fuzzy množina, ve které má její hodnota největší stupeň příslušnosti. Z takto nalezených termů se vytvoří pravidlo, kde vstupní proměnné tvoří antecedent a výstupní proměnné konsekvent. Zde však může dojít ke konfliktu, a to v případě, kdy dříve vytvořené pravidlo

obsahuje stejný antecedent, ale jiný konsekvent. Proto se spolu s pravidly ukládá i tzv. stupeň vyjadřující jejich sílu. Jsou-li pravidla v konfliktu, je uloženo to s vyšším stupněm.

```

foreach sample in dataSet
{
  x, y, θ, φ ∈ sample
  find fuzzy sets:
    A ∈ X : mA(x) ≥ mA'(x), ∀A' ∈ X
    B ∈ Y : mB(y) ≥ mB'(y), ∀B' ∈ Y
    C ∈ θ : mC(z) ≥ mC'(z), ∀C' ∈ θ
    D ∈ φ : mD(u) ≥ mD'(u), ∀D' ∈ φ

  degreeA,B,C = mA(x) · mB(y) · mC(z)

  if(degreeA,B,C > degreeA',B',C') {
    add rule:
      if X is A and Y is B and θ is C then φ is D
  }
}

```

Algoritmus 6-3: Odvození pravidel z numerických dat

Původně bylo v plánu zařadit do regulace i rychlost s cílem umožnit vozidlu pohyb vpřed, čímž by bylo dosaženo správného parkování ze všech možných pozic. Návrh počítal s pravidly ve tvaru *if* x_{car} *is* A *and* y_{car} *is* B *and* θ_{car} *is* C *then* φ *is* D *and* *is* v *is* E. K tomu bylo třeba zvětšit počet neuronů ve vrstvě B na 152. Pravidla byla učena pomocí manuálně vygenerovaných dat. Při trénovacích jízdách byl kladen důraz na to, aby trajektorie pohybu při couvání byly odlišné od těch, kdy vůz jel dopředu. Cílem bylo od sebe dostatečně odlišit antecedenty učených pravidel. Testy ukázaly, že FAM kontrolér správně vybavuje obě řídicí proměnné, avšak systém jako celek nefungoval. Vozidlo se často dostávalo do stavů s nulovou rychlostí nebo docházelo k oscilaci mezi dvěma pozicemi. Problém byl způsobem selháváním učící metody v místech, kde bylo potřeba změnit kladnou rychlost na zápornou, což pomocí klávesnice není možno provádět skokově. Tyto místa pak vedly ke konfliktům pravidel, kdy různým konsekventům odpovídalo více předpokladů. Z časových důvodů již nešlo provádět další úpravy, a proto byl tento koncept opuštěn.

7 Experimenty

7.1 Restaurace obrazu

Protože u asociativních pamětí nás zajímá především kapacita, chybovost a schopnost opravy poškozeného vstupu, experimenty s jednotlivými modely se ubírají tímto směrem. Experimenty jsou realizovány nad úlohou restaurace obrazů, proto je nutné stanovit metriky úspěšnosti, které vystihnou úspěšnost opravy. První metrikou je chyba stanovená jako průměr rozdílných pixelů mezi požadovaným (uloženým) a vybaveným obrazem. Tento přístup předpokládá, že vstup, byť poškozený či jinak deformovaný, je součástí fundamentální množiny. Druhou (vedlejší) metrikou je vizuální podobnost. Tato metrika je ryze subjektivní, avšak jak uvidíme níže, pouhý výpočet chyby na pixel v některých případech nestačí, a proto bude místy doplňována právě vizuální podobností.

7.1.1 Obrazová data

K experimentům jsou použity dva datové sety. Prvním z nich je databáze (obrázek 7-1) pocházející z *AT&T Laboratories Cambridge*¹ [27]. Jedná se o čtyřicet obličejů o velikosti 92x112 pixelů ve formátu pgm (*portable graymap*). Každý z obličejů se v databázi vyskytuje desetkrát v různých variacích (např. s brýlemi, mírně natočený atp.). Avšak pro účely práce byla vybrána pouze jedna z variací. Zbytek byl nahrazen jejími poškozenými verzemi, které byly vygenerovány strojově. Tato databáze je použita k testování schopnosti opravy poškozených vstupů. Při testech je využito několik druhů poškození. Jmenovitě se jedná o šum, řez, změnu jasu a o geometrické transformace rotace, posunu a škálování. Šum a řez jsou dále rozděleny na černý, bílý a šedý podle barev, kterými je poškození realizováno. Každý z těchto druhů je reprezentován několika vzorky lišící se v míře poškození. Např. šum a řez mají čtyři vzorky, u kterých je poškozeno 10%, 25%, 50% a 75% původního obrazu. Zmíněné druhy deformací lze pro jeden obličej vidět na obrázcích 7-2 až 7-9. Jako poškozený obraz budeme chápat i ten, který vznikne jako výsledek výše zmíněných geometrických transformací. Ty jsou zde zařazeny proto, protože pro algoritmy zpracování obrazu bývá důležité, zda jsou vůči těmto transformacím invariantní.

Druhým zdrojem obrazových dat použitých k testování paměťové kapacity je internetová stránka <http://wallpaperswide.com>, odkud bylo staženo velké množství fotografií s různou tématikou v rozlišení 1920x1080. Každý z těchto obrázků byl dále převeden na odstíny šedi a rozdělen na

¹ Databáze je dostupná na adrese: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>

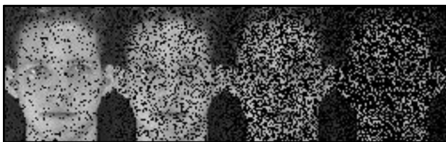
menší segmenty o velikosti 92x112. Aby byla zachována diverzita datové množiny, byly vyřazeny navzájem si podobné segmenty. Podobnost byla určena na základě korelace a normalizovaného histogramu. Vyřazeny byly i segmenty s malou variací v intenzitě indikující rozsáhlé jednobarevné plochy. Celkem bylo vytvořeno 50 000 segmentů. Z respektu k autorským právům tato data nejsou v práci publikována. Všechny použité obrázky jsou z důvodu kvadratických nároků na paměť zmenšené na rozměry 46x56.



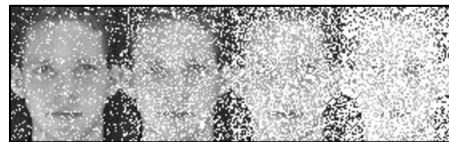
Obrázek 7-1: Databáze AT&T

7.1.2 Maximální kapacita

Cílem tohoto experimentu je ukázat, jak velikost fundamentální množiny ovlivňuje chybu vybavení jak nepoškozeného, tak poškozeného vstupu. Jako poškozený vstup byl pro duální metody vybrán bílý šum a pro zbytek modelů šum černý. Výsledky jsou zaneseny do tabulek 7-1 a 7-2, kde sloupec počet udává velikost fundamentální paměti.



Obrázek 7-2: Černý šum



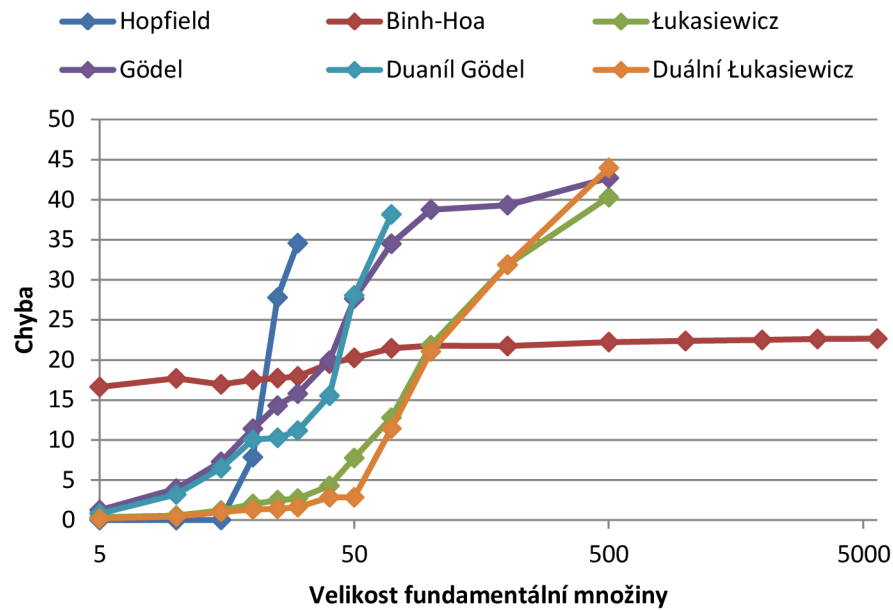
Obrázek 7-3: Bílý šum

Počet	Hopfield	Binh-Hoa	Łukasiewicz	Gödel	Duální Gödel	Duální Łukasiewicz
20	0	19.971	0.318	0	0	0.137
40	1.261	19.0569	0.362	0	0	0.091
60	25.081	17.456	0.381	0	0	0.0568
80	60.578	19.341	0.397	0	0	0.0386
100	80.084	19.888	0.39	0	0	0.019
500	---	19.023	0.382	0	0	0.000995
1000	---	19.831	0.373	0	0	0.000069
5000	---	19.068	0.379	0	0	0.00000103
10000	---	14.431	0.384	0	0	0
20000	--	14.491	0.382	0	0	0
50000	--	13.789	0.386	0	0	0

Tabulka 7-1: Chyba odezvy v závislosti na velikosti fundamentální množiny pro nepoškozený vstup

Počet	Hopfield	Binh-Hoa	Łukasiewicz	Gödel	Duální Gödel	Duální Łukasiewicz
5	0	16.619	0.37	1.256	0.815	0.195
10	0	17.686	0.56	3.93	3.196	0.436
15	0.0251	16.946	1.21	7.276	6.493	1.025
20	7.855	17.512	2.03	11.407	10.04	1.331
25	27.79	17.724	2.49	14.29	10.274	1.405
30	34.567	17.94	2.69	15.787	11.168	1.615
40	---	19.532	4.28	19.899	15.524	2.855
50	---	20.255	7.75	27.64	28.046	2.844
70	---	21.458	12.79	34.5	38.163	11.429
100	---	21.796	21.82	38.76	---	21.054
200	---	21.721	31.87	39.318	---	31.87
500	---	22.22	---	---	---	---
1000	---	22.3576	---	---	---	---
2000	---	22.486	---	---	---	---
3300	---	22.611	---	---	---	---
5670	---	22.6453	---	---	---	---

Tabulka 7-2: Chyba odezvy v závislosti na velikosti fundamentální množiny pro poškozený vstup



Graf 7-1: Chyba odezvy v závislosti na velikosti fundamentální množiny pro poškozený vstup

Buňky v tabulkách označené symbolem "---" indikují, že měření již dále nedávalo smysl, protože výstup degradoval na náhodnou změť pixelů či se stal pouhou kopií svého zašuměného vstupu a metrika chyba na pixel tyto jevy dostatečně nepostihuje.

7.1.3 Schopnost opravy

Jak již bylo řečeno, obraz může být degradován různými způsoby, a proto se zde zaměříme na to, které druhy poškození implementovaným asociativním pamětem dělají potíže a které ne.



Obrázek 7-3: Černý řez



Obrázek 7-4: Bílý řez



Obrázek 7-5: Jas



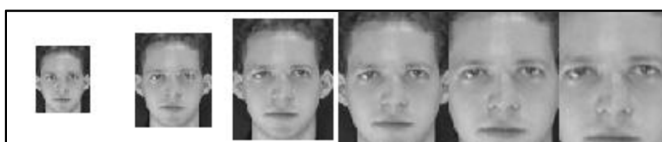
Obrázek 7-6: Šedý řez a šum



Obrázek 7-7: Rotace



Obrázek 7-8: Posun



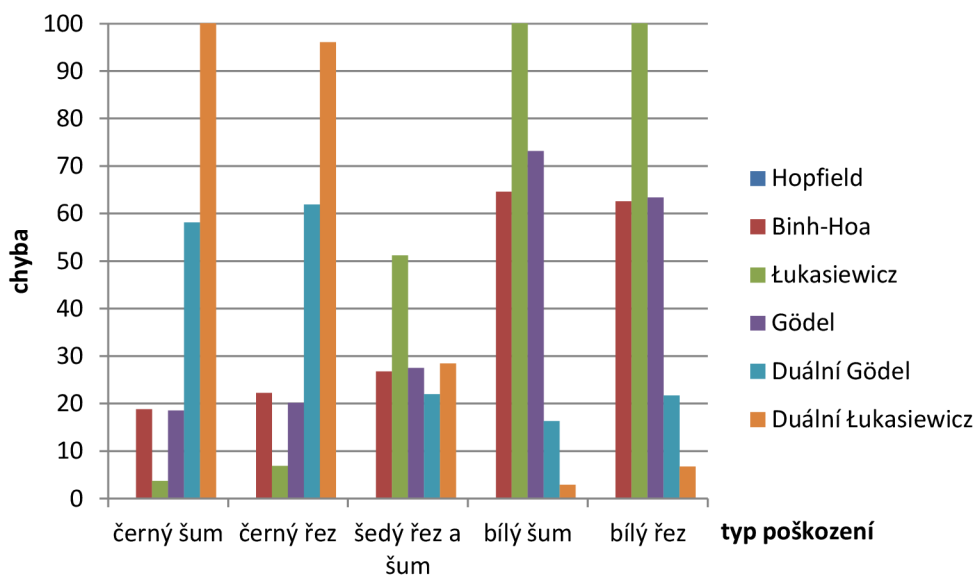
Obrázek 7-9: Škálování

Jedním z cílů experimentu je potvrdit hypotézu, že FAM s vybranou metodou pro výběr a fuzzifikaci příznaků nejsou invariantní vůči posunu, rotaci a škálování. Naopak zašumělé či ořezané vstupy by měly být vybaveny dobře.







Pro fuzzy asociativní paměti byla velikost fundamentální množiny stanovena na 40 a pro Hopfieldovu síť na 15 uložených obrazů. Výsledky prezentuje tabulka 7-3 a graf 7-2. Použitá metrika zkresluje výsledky u vstupů s pozměněným jasem. Odezva fuzzy asociativních pamětí totiž stále velmi dobře vystihuje původní obraz, jak lze vidět v tabulce 7-4, kde liché obrázky tvoří vstup a sudé odezvu.

	Hopfield	Binh-Hoa	Łukasiewicz	Gödel	Duální Gödel	Duální Łukasiewicz
černý šum	0.0251	18.79	3.736	18.533	58.172	102.06
černý řez	0.0251	22.28	6.868	20.169	61.934	96.029
šedý řez a šum	0	26.735	51.223	27.528	21.984	28.469
bílý šum	0.0251	64.619	133.115	73.149	16.305	2.901
bílý řez	0.0251	62.586	123.414	63.381	21.681	6.793
jas	41.538	43.27	50.711	37.958	31.807	26.09
rotace	33.44	41.701	48.195	45.231	55.727	73.071
posun	33.442	43.575	48.736	48.287	74.06	94.207
škálování	29.148	32.979	38.518	40.164	45.286	55.877

Tabulka 7-3: Chyba v závislosti na druhu poškození



Graf 7-2: Porovnání chyb asociativních pamětí během restaurace obrazu

Hopfield	
Binh-Hoa	
Łukasiewicz	
Gödel	
Duální Gödel	
Duální Łukasiewicz	

Tabulka 7-4: Odezva na jasové poškození

7.1.4 Zhodnocení

Paměťová kapacita všech implementovaných fuzzy asociativních pamětí vysoce překonává paměťovou kapacitu Hopfieldovy sítě. Zatímco Gödel a duální Gödel mohou bez problému fungovat jako bezchybná obsahem adresovatelná paměť, modely Binh-Hoa, Łukasiewicz a duální Łukasiewicz jsou zatíženy chybou. U Łukasiewicz se tato chyba jeví konstantně u Binh-Hoa a duální Łukasiewicz chyba klesá s narůstajícím počtem uložených dat.

Nejlepší schopnost opravy poškozeného vstupu při velkém počtu uložených dat projevila model Binh-Hoa. Avšak do jisté velikosti fundamentální paměti si vedli nejlépe Łukasiewicz a duální Łukasiewicz, a proto byla pro úlohu řízení vybrána fuzzy asociativní paměť založená právě na Łukasiewiczově implikativním učení.

Experiment s různými druhy poškození ukázal, že implementované fuzzy asociativní paměti na každý typ reagují jinak. Jak bylo předpokládáno, žádná z metod není invariantní vůči posunu, rotaci a škálování. Zajímavý je fakt, že barva poškození hraje významnou roli pro správné vybavení. Je-li poškození černé, Binh-Hoa, Łukasiewicz i Gödel jej umí opravit vcelku dobře, ale bílé poškození opravit neovládou. V literatuře (např. [12], [23]) se často dělají pokusy pouze s černým šumem, což může být zavádějící. U duálních metod je tomu naopak. Ty umí dobře opravit bílý šum. Jelikož reprezentace ostrých hodnot při BIOFAM inferenci odpovídá poškození fuzzy množiny černým řezem (šumem), který má po fuzzifikaci stupeň příslušnosti nula, pro úlohu řízení se duální Łukasiewicz nehodí. Šedotónový šum zvládnul nejhůře Łukasiewicz. Hopfieldova síť citlivostí na

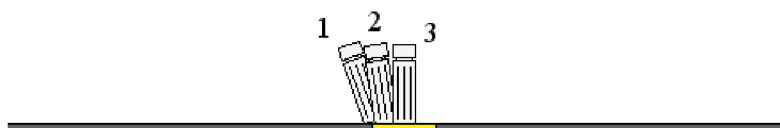
barvu netrpí. Pokusy s jasem byly vyhodnoceny podle subjektivní metriky, která ukazuje, že fuzzy asociativní paměti jasové poškození víceméně nezměněné kopírují na svůj výstup.

7.2 Řízení

V této podkapitole jsou prezentovány experimenty s aplikací pro řízení procesu parkování nákladního vozidla. Jedná se především o porovnání implementovaných regulátorů z hlediska chyby řízení (7-1), kterou zde budeme chápat jako váženou odchylku finální polohy vozu $x_{end}, y_{end}, \theta_{end}$ od souřadnic nákladní rampy.

$$chyba = 1 \cdot |x_{end} - x_{dock}| + 0.25 \cdot |y_{end} - y_{dock}| + 0.5 \cdot |\theta_{end} - \theta_{dock}| \quad (7-1)$$

Tři různé koncové pozice s vyjádřením numerické chyby ilustruje obrázek 7-10, kde situace číslo 1 zobrazuje chybu o velikosti 28.60, číslo 2 chybu 15.63 a situace číslo 3 ukazuje, jak vypadá bezchybné parkování.



Obrázek 7-10: Ilustrace různých koncových pozic

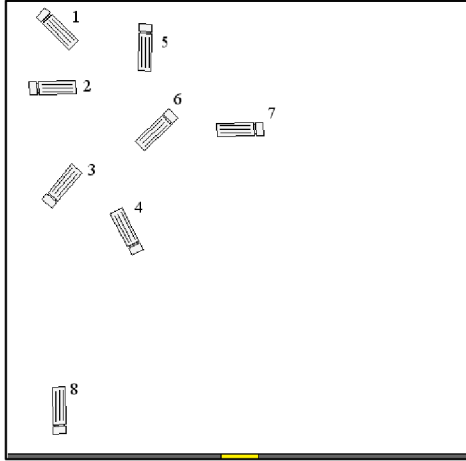
7.2.1 Porovnání regulátorů

Porovnání všech algoritmů probíhalo na základě spuštění simulace z různých startovních pozic a následným výpočtem metriky (7-1). Jelikož pravidla i fuzzy množiny jsou podle vertikální osy symetrické, testy mohou být omezeny na počáteční pozice $x_{start}, y_{start}, \theta_{start}$ z levé poloviny prostoru.

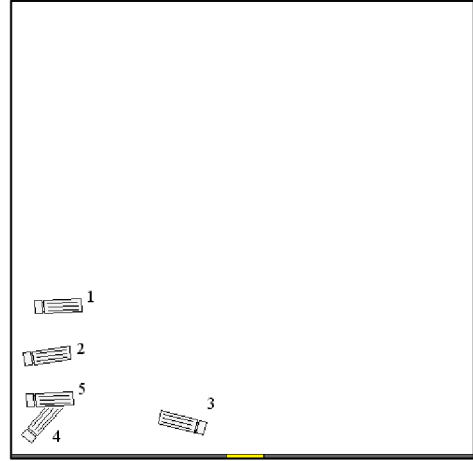
Adaptivní FAM regulátor musel být nejprve naučen pomocí trénovacích trajektorií. Ty byly složeny z jízd řízených fuzzy regulátorem a z jízd řízených klávesnicí. Manuální trénování bylo potřeba zejména v dolní polovině prostoru, protože fuzzy regulátor zanedbává ypsilonové souřadnice. Celkem bylo vygenerováno a použito 61 trajektorií, z nichž adaptivní systém odvodil 50 pravidel.

Z důvodů zanedbání y_{car} je také experiment rozdělen na dvě části. V první části, která je složena z osmi jízd, jde o porovnání odchylek ze srovnatelných startovních pozic. Jedná se o takové pozice, kde y_{car} má malý význam. Jak ukazuje obrázek 7-11, většinou se jedná o pozice z horní poloviny prostoru. Srovnání je uvedeno v tabulce 7-5 a v grafu 7-3. Ve druhé části se zkoumá vliv

zavedení proměnné y_{car} do adaptivního regulátoru, jehož lepší výsledky se zde předpokládají. Pokusů v této části bylo pět. Jejich počáteční pozice lze vidět na obrázku 7-12. Výsledky prezentuje tabulka 7-6 a graf 7-4.



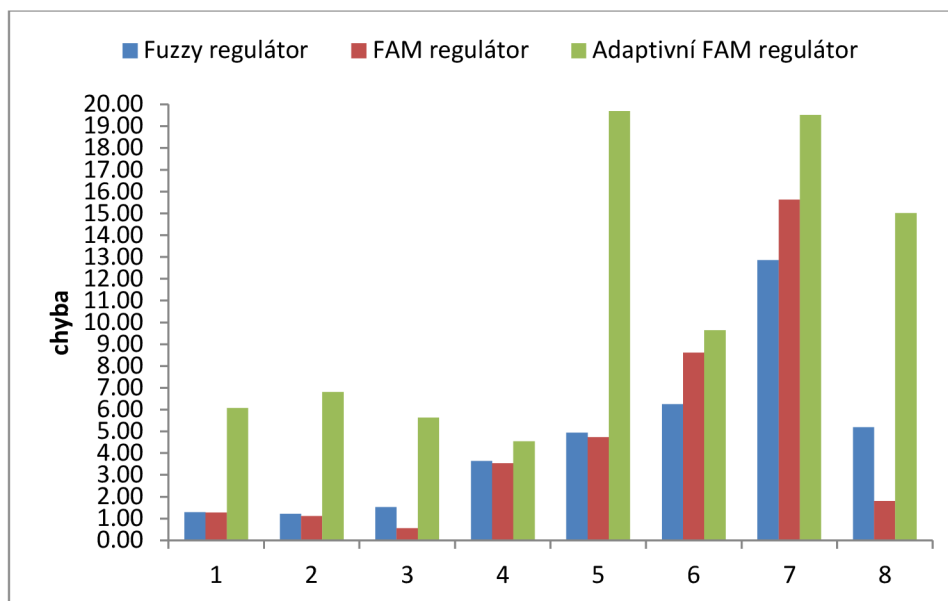
Obrázek 7-11: Počáteční pozice bez vlivu y_{car}



Obrázek 7-12: Počáteční pozice závislé na y_{car}

Pokus	Počáteční pozice ($x_{start}, y_{start}, \theta_{start}$)	Fuzzy regulátor	FAM regulátor	Adaptivní FAM regulátor
1	(70,50,225°)	1.294	1.272	6.072
2	(72,97,179°)	1.213	1.12	6.804
3	(74,185,130°)	1.52	0.557	5.63
4	(118,225,62°)	3.641	3.528	4.547
5	(148,77,270°)	4.95	4.74	19.698
6	(144,158,316°)	6.245	8.609	9.64
7	(228,143,359°)	12.86	15.63	19.518
8	(56,420,88°)	5.195	1.8	15.016
	Průměrná chyba:	4.615	4.657	10.866

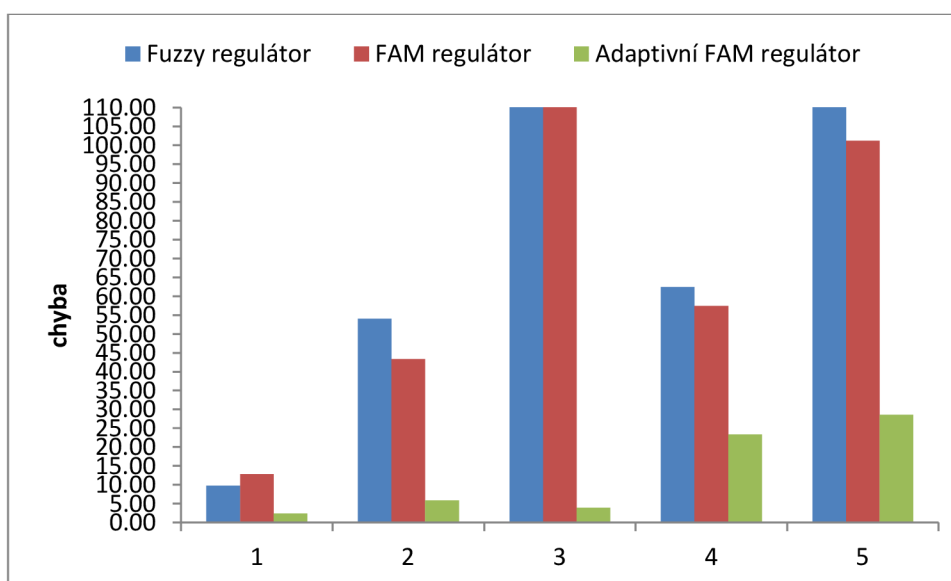
Tabulka 7-5: Porovnání regulátorů bez vlivu y_{car}



Graf 7-3: Porovnání regulátorů bez vlivu y_{car}

Pokus	Počáteční pozice ($x_{start}, y_{start}, \theta_{start}$)	Fuzzy regulátor	FAM regulátor	Adaptivní FAM regulátor
1	(61,382,172°)	9.778	12.78	2.35
2	(56,420,88°)	54.019	43.362	5.879
3	(161,451,14°)	218.18	226.121	3.946
4	(48,438,131°)	62.448	57.423	23.302
5	(64,431,177°)	110.472	101.238	28.599
Průměrná chyba:		90.979	88.185	12.815

Tabulka 7-6: Porovnání regulátorů s vlivem y_{car}



Graf 7-4: Porovnání regulátorů s vlivem y_{car}

7.2.2 Učení pravidel

Tento experiment je zaměřen na schopnost učícího algoritmu pro adaptivní FAM kontrolér zlepšovat přesnost s rostoucím počtem trénovacích trajektorií. Po každé nově přidané (popř. odebrané) jízdě do trénovací množiny byla provedena jedna simulace řízená adaptivním kontrolérem z počáteční pozice $(70,50,225^\circ)$. Trénovací jízdy byly řízené FAM regulátorem a začínaly v různých pozicích vybraných intuitivně tak, aby co nejvíce ovlivňovaly výběr pravidel na cestě z pozice $(70,50,225^\circ)$ k depu. Poté, co se adaptivní FAM přestal zlepšovat, byly odebrány první dvě uložené trajektorie z trénovací množiny. Následně pak byly odebrány další dvě, tentokrát vybrané náhodně. Vliv počtu trénovacích jízd a z nich odvozených pravidel na chybu je v tabulce 7-7.

Počet trénovacích jízd	Počet naučených pravidel	Chyba
1	6	319.127
2	8	45.123
3	12	9.455
4	13	8.848
5	13	8.848
6	15	8.848
7	17	8.848
8	18	10.104
9	20	10.104
7	19	1.93
6	19	1.866
5	16	4.094

Tabulka 7-7: Závislost chyby na trajektoriích v trénovací množině

7.2.3 Zhodnocení

Testy zaměřené na porovnání regulátorů ukázaly, že kontrolér založený na fuzzy asociativní paměti je funkčně, i co se přesnosti týče, téměř shodný s klasickým fuzzy regulátorem. To lze konstatovat i navzdory větší odchylce v jízdě číslo 8. Srovnatelná přesnost je překvapující, protože FAM kontrolér pracuje pouze s celými čísly a dle očekávání by měl být méně přesný. Adaptivní regulátor je na tom s přesností výrazně hůře. Nicméně, je-li generování trénovacích dat věnováno dostatek času, je možné jej naučit tak, aby dával poměrně dobré výsledky.

Druhý experiment odhalil zjevné nedostatky učící metody. Jak plyne z tabulky 7-7, lze systém relativně rychle naučit na rozumnou přesnost. Avšak další učení již zlepšení nepřináší. Naopak systém se může i zhoršit. Navíc po odebrání starších trajektorií se přesnost výrazně zlepšila. Tento fakt indikuje neoptimální rozhodování metody při výběru pravidel.

8 Závěr

V kapitole č. 4 byly popsány přístupy s aplikacemi fuzzy logiky v neuronových sítích. Těchto přístupů existuje mnoho a v literatuře neexistuje ucelenější diskuze o tom, co vše zahrnout pod pojem fuzzy neuronové sítě. Většina autorů se věnuje převážně systémům jako je ANFIS a jemu podobným. Avšak po podrobnějším zkoumání literatury jsem dospěl k názoru, že fuzzy neuronové sítě mohou být rozděleny podle klasifikace, která je uvedena v kapitole *Klasifikace neuronových sítí*. Pro účely práce byl záměrně vybrán spíše okrajový přístup, který byl podrobněji rozebrán a použit jako teoretický model v praktické části. Jedná se o takzvanou fuzzy asociativní paměť. Aby bylo možno takovou fuzzy neuronovou síť navrhnout a implementovat, je nezbytné pochopit základy fuzzy logiky a neuronové sítě. Proto byly těmto technikám věnovány samostatné kapitoly č. 2 a č. 3.

Praktická část se zabývala návrhem, implementací a testováním fuzzy asociativní paměti. Při návrhu je důležité zvolit vhodný model pro konkrétní situaci. Z tohoto důvodu byla do diplomové práce zahrnuta úloha rekonstrukce obrazu sloužící jako *benchmark* k ověřování různých vlastností vybraných modelů. Pro regulátor založený na fuzzy asociativní paměti s BIOFAM inferencí je zejména důležitá schopnost opravit chybný vstup. Proto byla pozornost v podkapitole 7.1 věnována hlavně různým druhům chybných vstupů a jejich vlivem na odezvu. Po zvážení výsledků byl vybrán model fuzzy implikativního učení založený na Łukasiewiczově implikaci.

Hlavním výstupem praktické části je aplikace simulující kinematiku nákladního vozu, který je možno za účelem parkování k nákladní rampě řídit různými regulátory. Těmi jsou referenční Mamdaniho fuzzy regulátor, regulátor založený na FAM a adaptivní FAM regulátor schopný se učit pravidla. Poslední dva modely se od sebe odlišují pouze ve způsobu ukládání pravidel. Pravidla pro neadaptivní systém jsou pevně dána, zatímco pro adaptivní systém jsou pravidla odvozována z trénovacích dat.

Při návrhu kontroléru byl kladen důraz na odstranění problémů uvedených v teoretické části práce. Tyto problémy se nepovedlo odstranit úplně, avšak podařilo se je významně zmírnit, přičemž testy ukázaly, že FAM regulátor je srovnatelný s referenčním systémem.

Pokračování této práce by se mohlo zaměřit na hledání způsobů, jak dále snižovat počet potřebných FAM k ukládání pravidel. Nejde pouze o ušetření paměťového prostoru, ale lze tím ušetřit i výpočetní čas. Jedním ze způsobů by mohla být analýza pravidel, které se navzájem ruší a proč. Následně by tato pravidla mohla být rozdělena do jiných FAM, jejichž počet by rostl dynamicky v případě, že nové pravidlo nelze do stávajících FAM uložit. Pravděpodobně by ke snížení počtu potřebných FAM přispěla i jiná metoda fuzzifikace. Stávající metoda využívající prahovou vrstvu vyžaduje, aby se překrývající fuzzy množiny ukládaly do různých FAM. Testy s adaptivním systémem odhalily nedostatky učící metody. Další bádání by se tedy mohlo vydat i tímto směrem.

Literatura

- [1] KRUSE, Rudolf. Fuzzy neural network. In: *Scholarpedia* [online]. 2008 [cit. 2015-01-18]. Dostupné z: http://www.scholarpedia.org/article/Fuzzy_neural_network
- [2] LIU, Puyin a Hongxing LI. *Fuzzy neural network theory and application*. New Jersey: World Scientific, 2004, 376 s. ISBN 981-238-786-2.
- [3] ELDON, Y. Artificial neural networks and their business applications. *Information & Management* [online]. 1994, vol. 27, issue 5, s. 303-313 [cit. 2015-01-18]. DOI: 10.1016/0378-7206(94)90024-8. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/0378720694900248>
- [4] ZBOŘIL, František V. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. *Biologický a umělý neuron, umělé neuronové sítě*. 2014.
- [5] BISKUP, Roman. *Možnosti neuronových sítí*. Praha, 2009. Disertační práce. Česká zemědělská univerzita v Praze. Vedoucí práce prof. RNDr. Anna Čermáková, CSc.
- [6] NEGNEVITSKY, Michael. *Artificial intelligence*. Vyd. 1. New York: Addison-Wesley, 2002, 394 s. ISBN 0-201-71159-1.
- [7] ROJAS, Raúl. *Neural networks: a systematic introduction*. Berlin: Springer-Verlag, 1996, xx, 502 s. ISBN 3540605053.
- [8] DE CAMPS, Marc. 2008. *Hopfield Networks*. [cit. 2015-05-05]. Dostupné také z: <http://www.comp.leeds.ac.uk/ai23/reading/Hopfield.pdf>
- [9] KOSKO, Bart. Bidirectional associative memories. *Systems, Man and Cybernetics, IEEE Transactions on* [online]. USA: IEEE, 1988, roč. 18, č. 1, s. 49-60 [cit. 2015-05-05]. DOI: 10.1109/21.87054.
- [10] ZADEH, Lotfi A. Fuzzy sets. *Information and Control* [online]. 1965, vol. 8, issue 3, s. 338-353 [cit. 2015-01-18]. DOI: 10.1016/S0019-9958(65)90241-X.

- [11] ZADEH, Lotfi A. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *Systems, Man and Cybernetics, IEEE Transactions on* [online]. USA: IEEE, 1973, SMC-3, issue 1, s. 28-44 [cit. 2015-01-18]. DOI: 10.1109/TSMC.1973.5408575.
- [12] VALLE, Marcos Eduardo a Peter SUSSNER. Fuzzy Associative Memories from the Perspective of Mathematical Morphology. In: *2007 IEEE International Fuzzy Systems Conference* [online]. 2007 [cit. 2015-05-05]. DOI: 10.1109/fuzzy.2007.4295473.
- [13] MODRLÁK, Osvald. 2002. *Teorie automatického řízení II: Fuzzy řízení a regulace*. Liberec. Dostupné také z: <http://www.kirp.chtf.stuba.sk/~bakosova/wwwRTP/tar2fuz.pdf>
- [14] KOSKO, Bart. *Neural networks and fuzzy systems*. New Jersey: Prentice-Hall, 1992, 449 s. ISBN 0-13-612334-1.
- [15] SIMPSON, P. K. Fuzzy min-max neural networks. I. Classification. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* [online]. 1992, vol. 3, issue 5, s. 776 [cit. 2015-01-18].
- [16] SIMPSON, P. K. Fuzzy min-max neural networks - Part 2: Clustering. *Fuzzy Systems, IEEE Transactions on* [online]. USA: IEEE, 1993, vol. 1, issue 1, s. 32 [cit. 2015-01-18]. DOI: 10.1109/TFUZZ.1993.390282.
- [17] INGMAN, D. a Y. MERLIS. Maximum entropy signal reconstruction with neural networks. *Neural Networks, IEEE Transactions on* [online]. USA: IEEE, 1992, vol. 3, issue 2, s. 195-201 [cit. 2015-01-18]. DOI: 10.1109/72.125860.
- [18] "Neuro-Fuzzy Systems: A Survey", Jose Vieira, Fernando Morgado Dias, Alexandre Mota, 5th WSEAS NNA International Conference on Neural Networks and Applications, Udine, Italy, 2004.
- [19] (ED)., Claude Sammut). *Australasian Conference on Robotics and Automation 2005 December 5-7, 2005, Sydney, Australia*. S.l.: Australian Robotics and Automation Association, 2005. ISBN 0958758379.

- [20] AN H. TON-THAT, An H., Nhan T. CAO a Hyung I. CHOI. 2013. Fuzzy Inference Systems based on Fuzzy Associative Memory with Adjusting Algorithm for Selecting Optimal Membership Functions. *International Journal of Intelligent Information Processing(IJIIP)*. (3). Dostupné také z: <http://www.aicit.org/IJIIP/ppl/IJIIP140PPL.pdf>
- [21] HON KEUNG KWAN a YALING CAI. A fuzzy neural network and its application to pattern recognition. *Fuzzy Systems, IEEE Transactions on* [online]. USA: IEEE, 1994, vol. 2, issue 3, s. 185-193 [cit. 2015-01-18]. DOI: 10.1109/91.298447.
- [22] FU-LAI CHUNG a TONG LEE. On fuzzy associative memory with multiple-rule storage capacity. *Fuzzy Systems, IEEE Transactions on* [online]. USA: IEEE, 1996, roč. 4, č. 3, s. 375-384 [cit. 2015-05-05]. DOI: 10.1109/91.531778.
- [23] SUSSNER, P. a M.E. VALLE. Implicative Fuzzy Associative Memories. *Fuzzy Systems, IEEE Transactions on* [online]. IEEE Publishing, 2006, roč. 14, č. 6, s. 793-807 [cit. 2015-05-05]. DOI: 10.1109/TFUZZ.2006.879968.
- [24] A New Fuzzy Associative Memory. BINH, Pham, Nong HOA a Vu THAI. *Context-Aware Systems and Applications Second International Conference, ICCASA 2013, Phu Quoc Island, Vietnam, November 25-26, 2013, Revised Selected Papers*. Cham: Springer International Publishing, 2014, s. 219-227. ISBN 3319059394.
- [25] ŠPAŇEL, Michal V. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. *Obraz, poškození obrazu, typy šumu, optimální filtrace*. 2014.
- [26] CORRELL, Nicolaus. 2012. Advanced Robotics #2: Forward Kinematics of Car-Like Mechanisms. *Correll Lab* [online]. [cit. 2015-05-05]. Dostupné z: <http://correll.cs.colorado.edu/?p=1869>
- [27] SAMARIA, F., Harter A. Parameterisation of a stochastic model for human face identification. *2nd IEEE Workshop on Applications of Computer Vision*. Sarasota, December 1994.