

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Evidenčné číslo: FIIT-5212-64414

Katarína Novotná

**ŠTÚDIUM VZŤAHU MEDZI REKURENTNOU
NEURÓNOVOU SIEŤOU S LOGICKÝMI NEURÓNMI
A KONEČNOSTAVOVÝM ZARIADENÍM (AUTOMATOM)**

Bakalárska práca

Vedúci práce: prof. Ing. Vladimír Kvasnička, DrSc

Máj 2013

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Evidenčné číslo: FIIT-5212-64414

Katarína Novotná

**ŠTÚDIUM VZŤAHU MEDZI REKURENTNOU
NEURÓNOVOU SIEŤOU S LOGICKÝMI NEURÓNMI
A KONEČNOSTAVOVÝM ZARIADENÍM (AUTOMATOM)**

Bakalárska práca

Študijný program: informatika

Číslo študijného odboru: 2508

Názov študijného odboru: 9.2.1 Informatika

Školiace pracovisko: Ústav aplikovanej informatiky

Vedúci záverečnej práce/školiteľ: prof. Ing. Vladimír Kvasnička, DrSc

Máj 2013

Prehlasujem, že som túto bakalársku prácu vypracovala samostatne, pod odborným vedením prof. Ing. Vladimír Kvasničku, DrSc., s použitím uvedenej literatúry.

ANOTÁCIA

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: Informatika

Autor: Katarína Novotná

Bakalárska práca: Štúdium vzťahu medzi rekurentnou neurónovou sieťou s logickými neurónmi a konečno stavovým zariadením (automatom)

Vedúci bakalárskej práce: prof. Ing. Vladimír Kvasnička, DrSc.

Máj 2013

Neurónové siete s logickými neurónmi môžu byť simulované konečno stavovým výpočtovým zariadením (automatom). Táto dôležitá vlastnosť platí aj v inverznom poradí, každé konečno stavové zariadenie môže byť simulované neurónovou sieťou s logickými neurónmi.

Predložená práca sa zaoberá skúmaním toho, že každá neurónová sieť je simulovaná konečno stavovým zariadením (automatom) tak, aby obe zariadenia produkovali rovnaký výstupný reťazec znakov ako odozvu na vstupné znaky. Zároveň skúma, že každé konečno stavové zariadenie (automat) je simulované neurónovou sieťou tak, aby obe zariadenia produkovali rovnaký výstupný reťazec znakov ako odozvu na vstupné znaky.

V grafickej časti sa zameriavame na starostlivo znázornené študované zariadenia.

ANOTATION

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: Informatics

Author: Katarína Novotná

Bachelor thesis: Study of the relationship between a recurrent neural network with logical neurons and a finite-state machine (automaton)

Supervisor: prof. Ing. Vladimír Kvasnička, DrSc.

2013, May

Neural networks with logical neurons can be simulated by finite-state machine (automaton). This important property is valid also vice versa; every finite-state machine can be simulated by a neural network with logical neurons.

The presented thesis investigates that every neural network can be simulated by a finite-state machine (automaton) by the assumption that both systems produce an identical output string of characters as a response to input characters.

In the graphic part we focus on thorough depiction of the studied systems.

Obsah

1	Úvod.....	1
1.1	História umelých neurónových sietí a automatov	1
2	Neurónové siete s logickými neurónmi.....	3
2.1	Výpočet aktivity neurónov	4
2.2	Implementácia logických operácií neurónovou sieťou	5
2.2.1	Implementácia Boolovej binárnej funkcie OR	5
2.2.2	Implementácia Boolovej binárnej funkcie AND.....	6
2.2.3	Implementácia Boolovej binárnej funkcie NOT	7
2.2.4	Implementácia Boolovej binárnej funkcie IF-THEN.....	7
2.3	Neurónová sieť	8
2.4	Aktivita neurónov v čase	11
2.5	Konstruktívny dôkaz ekvivalencie neurónovej siete s konečnosťavým automatom	11
3	Konečnosťavý automat.....	13
3.1	Definícia Mealyho automatu	14
3.2	Dôkaz ekvivalencie konečnosťavého automatu s neurónovou sieťou	17
4	Prevod neurónovej siete na konečnosťavý automat	18
4.1	Zadefinovanie neurónovej siete.....	18
4.2	Tvorba konečnosťavého automatu	20
4.3	Zhodnotenie	29
5	Prevod konečnosťavého automatu na neurónovú sieť	30
5.1	Zadefinovanie konečnosťavého automatu	30
5.2	Tvorba neurónovej siete	33
5.3	Zhodnotenie	34
	Príloha A Technická dokumentácia	37
A.1	Výber programovacieho jazyka.....	37

A.2	Implementácia prevodu konečnostavového automatu na neurónovú sieť.....	38
A.3	Implementácia prevodu neurónovej siete na konečnostavový automat	46
Príloha B	Inštaláčn príručka	52
Príloha C	Používateľská príručka.....	53
C.1	Prevod konečnostavového automatu na neurónovú sieť	54
C.2	Prevod neurónovej siete na konečnostavový automat s výstupom.....	57
Príloha D	Obsah disku CD	61

Zoznam tabuliek

Tabuľka 2.1: Binárna Boolova funkcia disjunkcie	5
Tabuľka 2.2: Binárna Boolova funkcia konjunkcie	6
Tabuľka 2.3: Binárna Boolova funkcia negácie.....	7
Tabuľka 2.4 Binárna Boolova funkcia implikácie	8
Tabuľka 3.1: Prechodová a výstupná funkcia konečnostavového stroja	16
Tabuľka 3.2: Postupnosť znakov a stavov	16
Tabuľka 4.1: Aktivita neurónov v čase – vstup a výstup siete.....	20
Tabuľka 4.2: Aktivita neurónov v čase – stavy skrytých neurónov.....	21
Tabuľka 4.3: Aktivita neurónov v čase $t=1$ a $t=2$ – prechod konečnostavového automatu	22
Tabuľka 4.4: Aktivita neurónov v čase $t=3$ a $t=4$	22
Tabuľka 4.5: Zisťovanie prechodu zo stavu $S000$ pri vstupe 0 a výstupe 0.....	25
Tabuľka 4.6: Zisťovanie prechodu zo stavu $S000$ pri vstupe 0 a výstupe 1.....	25
Tabuľka 4.7: Zisťovanie prechodu zo stavu $S100$ pri vstupe 0 a výstupe 0.....	25
Tabuľka 4.8: Zisťovanie prechodu zo stavu $S100$ pri vstupe 0 a výstupe 1.....	25
Tabuľka 4.9: Krok 1 – overenie nedosiahnuteľného stavu $S001$	28
Tabuľka 4.10: Krok 2 – overenie nedosiahnuteľného stavu $S001$	29
Tabuľka 5.1: Prechodová a výstupná funkcia konečnostavového automatu	30
Tabuľka 5.2: Prechodová funkcia konečnostavového automatu.....	31
Tabuľka 5.3: Výstupná funkcia konečnostavového automatu	31
Tabuľka 5.4: Karnaughova mapa pre prechodovú funkciu.....	32
Tabuľka 5.5: Karnaughova mapa pre výstupnú funkciu.....	32

Zoznam obrázkov

Obr. 2.1: Logický neurón McCullocha a Pittsa [6].....	3
Obr. 2.2: Boolova funkcia disjunkcie [6].....	6
Obr. 2.3: Boolova funkcia konjunkcie [6]	6
Obr. 2.4: Boolova funkcia negácie [6].....	7
Obr. 2.5: Boolova funkcia implikácie [6]	8
Obr. 2.6: Syntaktický strom a neurónová sieť [6].....	9
Obr. 2.7: 3-vrstvová neurónová sieť [6]	10
Obr. 3.1: Stavový automat	13
Obr. 3.2: Konečnosťavový stroj	15
Obr. 3.3: Stavový diagram Mealyho automatu	15
Obr. 4.1: Rekurentná neurónová sieť	19
Obr. 4.2: Prvý prechod konečnosťavového automatu	22
Obr. 4.3: Tretí prechod konečnosťavového automatu	23
Obr. 4.4: Konečnosťavový automat reprezentujúci neurónovú sieť.....	24
Obr. 4.5: Konečnosťavový automat s prechodmi zo stavu S_{100} pri vstupe 0 s rôznymi výstupmi v čase $t-1$	26
Obr. 4.6: Konečnosťavový automat reprezentujúci neurónovú sieť pri akýchkoľvek vstupoch	27
Obr. 5.1: Konečnosťavový automat s výstupom	30
Obr. 5.2: Schematická reprezentácia konečnosťavového automatu reprezentovaného prechodovou a výstupnou funkciou	33
Obr. 5.3: Prechodová a výstupná funkcia reprezentovaná neurónovými sieťami	34
Obr. 5.4: Rekurentná neurónová sieť ekvivalentná s konečnosťavovým automatom.....	34
Obr. A.1: Diagram činností pre prevod konečnosťavového automatu na neurónovú sieť	39
Obr. A.2: Sekvenčný diagram prevodu automatu na sieť	45
Obr. A.3: Diagram činností pre prevod neurónovej siete na automat.....	46

Obr. A.4: Sekvenčný diagram prevodu neurónovej siete na automat.....	51
Obr. C.2: Stránka prevodu konečnostavového automatu na neurónovú sieť	54
Obr. C.4: Prechody automatu z obrázku 5.1 vo formulári programu	55
Obr. C.5: Výsledok vykreslenia neurónovej siete.....	56
Obr. C.6: Stránka prevodu siete na automat	57
Obr. C.7: Tabuľka pre zadanie prahových koeficientov neurónov	58
Obr. C.8: Tabuľka zadania parametrov prechodov v neurónovej sieti	58
Obr. C.9: Tabuľka pre zadanie vstupných hodnôt	59
Obr. C.10: Aktivity neurónov v čase	59
Obr. C.11: Grafický výstup prevodu neurónovej siete na konečnostavový automat.....	60

1 Úvod

V staroveku si ľudia mysleli, že sídlom duše je srdcom. Dnes už vieme, že duša sídli v mozgu. Mozog sa skladá z miliónoch navzájom prepojených nervových buniek nazývaných neuróny. Neurónová sieť je výpočtový model zostavený na základe abstrakcie vlastností biologických nervových systémov. Základnou vlastnosťou neurónových sietí je schopnosť abstrakcie pravidiel medzi vstupnými a výstupnými hodnotami prezentovanými vo vhodnej forme a následnou aplikáciou získaných pravidiel na akékoľvek vstupné hodnoty. Pri snahe modelovať činnosť nervového systému vznikli neurónové siete. Neskôr sa zistilo, že neurónové siete sa dajú využiť aj v oblasti počítačových vied.

Mojim cieľom bakalárskej práce je predstaviť teóriu neurónových sietí a konečnostavového zariadenia, predviesť postup prevodu neurónovej siete na konečnostavový automat a naopak, navrhnúť a implementovať prevod neurónovej siete na konečnostavový automat a naopak, čím by sa potvrdila ekvivalencia medzi neurónovými sieťami a konečnostavovým zariadením.

1.1 História umelých neurónových sietí a automatov

V roku 1943 McCulloch a Pitts predstavili prvý model neurónovej siete. V ich práci ukázali, že neurónové siete sú efektívnym výpočtovým prostriedkom v doméne Boolovských funkcií. Ich model neurónovej siete je výrazným medzníkom vo vývoji teórie neurónových sietí. McCulloch a Pitts dokázali, že procesy, ktoré sa dajú realizovať konečným počtom výrazov, sa dajú realizovať sieťou, ktorá obsahuje tieto elementy. V prípade, že táto sieť má neobmedzenú veľkosť, jej výpočtová sila je ekvivalentná Turingovému stroju bez pásky. V ich práci predstavili zjednodušený formálny model neurónu – binárnu jednotku aktivovanú pomocou vstupov na základe hodnoty vnútorného prahu. Jednotlivé vstupy však mali priradené len konštantné synaptické váhy.

„Až v roku 1958 Rosenblatt poukázal, že logická jednotka McCullocha a Pittsa môže byť trénovaná a použitá na riešenie klasifikačných problémov. Už o desať rokov

neskôr bolo dokázané, že jednovrstvová sieť perceptrónov dokáže riešiť len lineárne separovateľné problémy. Minsky a Papert neskôr zistili, že problém sa dá odstrániť zoskupením perceptrónov do viacerých vrstiev. Avšak aktivačná funkcia perceptrónu neumožňovala adaptovať synaptické váhy vo viacvrstvovej sieti.“ [2]

Tento problém odstránili v roku 1986 Rumelhart, Hinton a Williams keď predstavili učiace pravidlo pre dopredné neurónové siete. Použitie spojitej aktivačnej funkcie umožnilo aplikovať na tréningové siete gradientové optimalizačné metódy. Uvedený spôsob učenia, stále patriaci medzi najpoužívanejšie algoritmy, umožnil použiť neurónové siete v množstve priemyselných aplikácií ako rozpoznávanie a klasifikácia objektov, spracovanie signálov, jazyka, atď. Teoreticky sa ukázalo, že dopredné neurónové siete dokážu aproximovať ľubovoľnú spojitú funkciu.

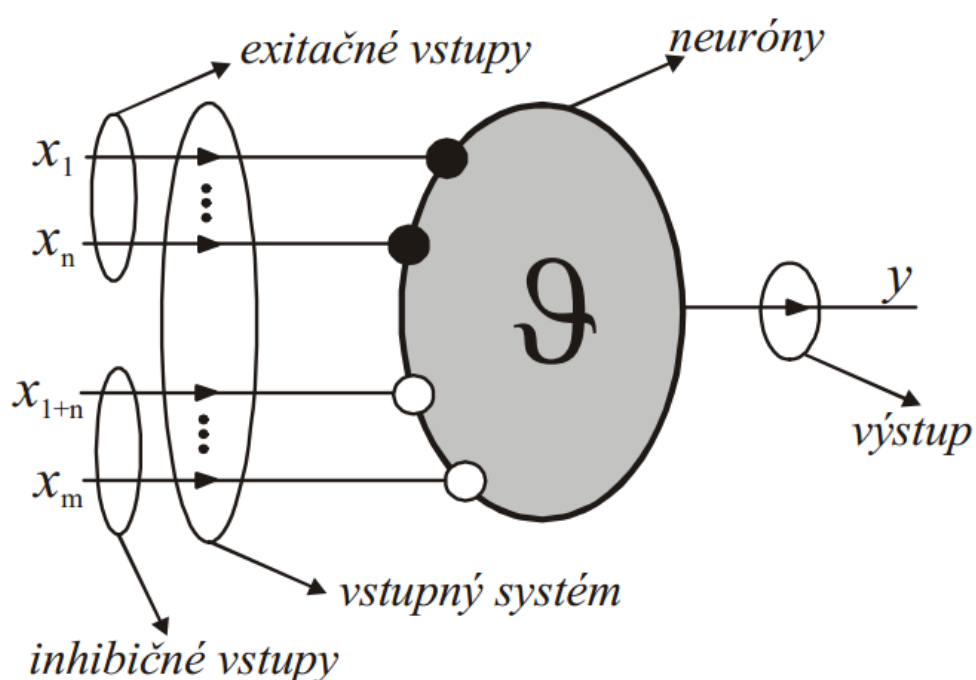
„Okrem dopredných sietí sa rozvíjali aj siete obsahujúce spätnú väzbu medzi neurónmi, tzv. rekurentné siete. Počiatkom osemdesiatych rokov Hopfield využil funkciu energie pre formulovanie a vysvetlenie princípu fungovania rekurentnej siete so symetrickými spojeniami. Hoci Hopfieldova sieť nebola realistickým neurobiologickým modelom, poskytla nový pohľad na uchovávanie informácie v dynamických ale aj napriek tomu stabilných sieťach.“ [3]

McCulloch vo svojej práci z roku 1943 taktiež poukázal na to, že pomocou automatu s konečnou množinou stavov môžeme demonštrovať neurónovú sieť. Siegelmann – Sontag ukázali, že je možné zostrojiť rekurentnú neurónovú sieť realizujúcu univerzálny Turingov stroj, t.j. rekurentné neurónové siete sú výpočtovo univerzálne.

Neskôr koncom 60. rokov sa poukázalo na súvislosť medzi teóriou formálnych jazykov a teóriou konečných automatov tým, že boli popísané jazyky, ktoré sú určené konečnými automatmi. Rýchly rozvoj teórie bol ovplyvnený rýchlym rozvojom počítačov. Automaty môžu totiž byť veľmi dobre použité práve pri opise správania sa výpočtových systémov.

2 Neurónové siete s logickými neurónmi

„Základnou jednotkou neurónovej siete je logický neurón (výpočtová jednotka), pričom jej stav môže byť 1 alebo 0, t.j. binárny. Systém obsahuje excitačné a inhibičné vstupy. Excitačné vstupy sú opísané binárnymi premennými x_1, x_2, \dots, x_n . Inhibičné vstupy sú opísané binárnymi premennými $x_{n+1}, x_{n+2}, \dots, x_m$.“ [2]



Obr. 2.1: Logický neurón McCullocha a Pittsa [6]

„Logický neurón (pozri obrázok 1) podľa McCullocha a Pittsa obsahuje dendritický vstupný systém pre excitačné a inhibičné vstupné aktivity, axón pre výstup neurónovej aktivity, soma neurónu (telo neurónu), ktoré je charakterizované prahovým koeficientom θ .“ [2]

2.1 Výpočet aktivity neurónov

Logický neurón je založený na pravidle:

„Aktivita je jednotková, ak vnútorný potenciál neurónu definovaný ako rozdiel medzi excitačnými a inhibičnými vstupmi je väčší alebo rovný prahu ϑ , v opačnom prípade je nulová.“ [2]

$$y = \begin{cases} 1 & (\xi = x_1 + \dots + x_n - x_{n+1} - \dots - x_m \geq \vartheta) \\ 0 & (\xi < \vartheta) \end{cases}$$

Použitým jednoduchou krokovej funkcie

$$s(\xi) = \begin{cases} 1 & (\text{ak } \xi \geq 0) \\ 0 & (\text{ak } \xi < 0) \end{cases}$$

môžeme aktivitu vyjadriť pomocou vzorca:

$$y = s(x_1 + \dots + x_n - x_{n+1} - \dots - x_m - \vartheta)$$

Aktivita logického neurónu môže byť vyjadrená pomocou binárnych váhových koeficientov. Excitačné aktivity vstupujú do neurónu cez spoje, ktoré sú ohodnotené jednotkovým váhovým koeficientom ($w = 1$) a inhibičné aktivity vstupujú do neurónu cez spoje so záporným koeficientom ($w = -1$).

$$y = s(w_1x_1 + \dots + w_mx_m - \vartheta) = s\left(\sum_{i=1}^m w_ix_i - \vartheta\right)$$

2.2 Implementácia logických operácií neurónovou sieťou

Jednoduchá implementácia elementárnych Boolových funkcií disjunkcie, konjunkcie a negácie je znázornená v kapitolách 2.2.1 až 2.2.4. Tieto logické funkcie sa budú využívať pri vykresľovaní neurónovej siete ekvivalentnej konečnostavovému automatu.

Excitačné spoje (váhový koeficient $w=1$) sú znázornené plným krúžkom a inhibičné spoje (váhový koeficient $w=-1$) prázdny krúžkom.

2.2.1 Implementácia Boolovej binárnej funkcie OR

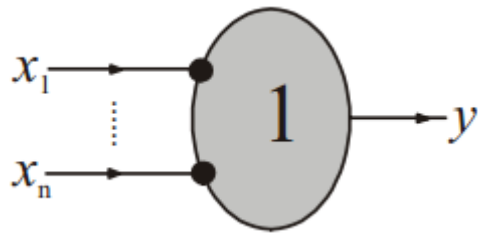
$$y_{OR}(x_1, x_2) = s(x_1 + x_2 - 1)$$

Funkčné hodnoty tejto Boolovej funkcie sú ukázané v tabuľke 2.1

	x_1	x_2	$y_{OR}(x_1, x_2)$	$x_1 \vee x_2$
1	0	0	$s(-1)$	0
2	0	1	$s(0)$	1
3	1	0	$s(0)$	1
4	1	1	$s(1)$	1

Tabuľka 2.1: Binárna Boolova funkcia disjunkcie

Z tabuľky vyplýva, že Boolova funkcia y_{OR} simuluje Boolovu funkciu disjunkcie. Na obrázku 2.2 je znázornení neurón, ktorý reprezentuje Boolovu funkciu disjunkcie pri n vstupoch.



Obr. 2.2: Boolova funkcia disjunkcie [6]

2.2.2 Implementácia Boolovej binárnej funkcie AND

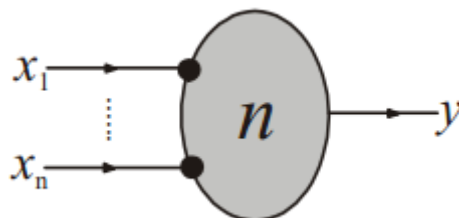
$$y_{AND}(x_1, x_2) = s(x_1 + x_2 - 2)$$

Funkčné hodnoty tejto Boolovej funkcie sú ukázané v tabuľke 2.2

	x_1	x_2	$y_{AND}(x_1, x_2)$	$x_1 \wedge x_2$
1	0	0	$s(-2)$	0
2	0	1	$s(-1)$	0
3	1	0	$s(-1)$	0
4	1	1	$s(0)$	1

Tabuľka 2.2: Binárna Boolova funkcia konjunkcie

Z tabuľky vyplýva, že Boolova funkcia y_{AND} simuluje Boolovu funkciu konjunkcie. Na obrázku 2.3 je znázornení neurón, ktorý reprezentuje Boolovu funkciu konjunkcie pri n vstupoch.



Obr. 2.3: Boolova funkcia konjunkcie [6]

2.2.3 Implementácia Boolovej binárnej funkcie NOT

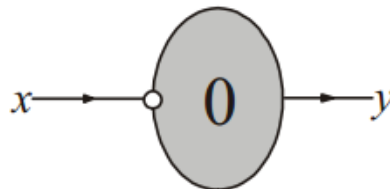
$$y_{NOT}(x) = s(-x)$$

Funkčné hodnoty tejto Boolovej funkcie sú ukázané v tabuľke 2.3

	x	$y_{NOT}(x)$	$\neg x$
1	0	s(0)	1
2	1	s(-1)	0

Tabuľka 2.3: Binárna Boolova funkcia negácie

Z tabuľky vyplýva, že Boolova funkcia y_{NOT} simuluje Boolovu funkciu negácie. Na obrázku 2.4 je znázornení neurón, ktorý reprezentuje Boolovu funkciu negácie.



Obr. 2.4: Boolova funkcia negácie [6]

2.2.4 Implementácia Boolovej binárnej funkcie IF-THEN

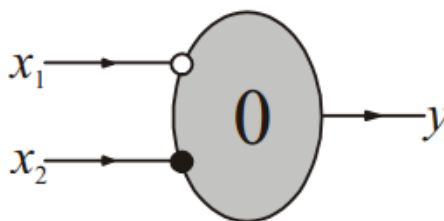
$$y_{IF-THEN}(x_1, x_2) = s(-x_1 + x_2)$$

Funkčné hodnoty tejto Boolovej funkcie sú ukázané v tabuľke 2.4

	x_1	x_2	$y_{IF-THEN}(x_1, x_2)$	$x_1 \Rightarrow x_2$
1	0	0	$s(0)$	1
2	0	1	$s(1)$	1
3	1	0	$s(-1)$	0
4	1	1	$s(0)$	1

Tabuľka 2.4 Binárna Boolova funkcia implikácie

Z tabuľky vyplýva, že Boolova funkcia $y_{IF-THEN}$ simuluje Boolovu funkciu implikácie.

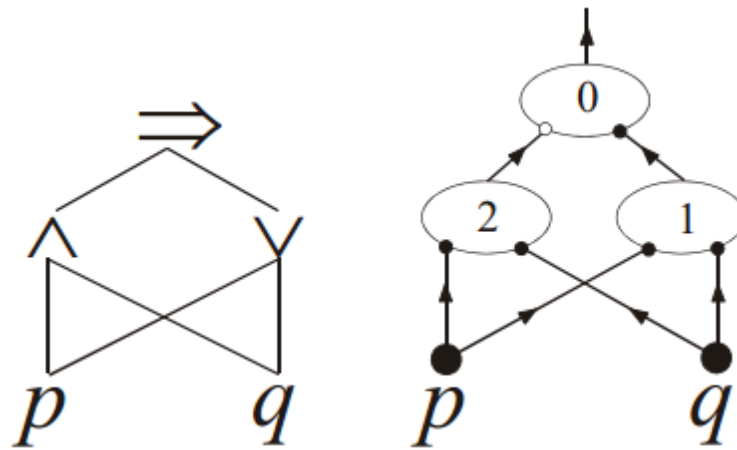


Obr. 2.5: Boolova funkcia implikácie [6]

2.3 Neurónová sieť

„Každá Boolova funkcia môže byť vyjadrená pomocou neurónovej siete zloženej z logických neurónov vyjadrujúcich logické spojky.“ [2]

Toto je jeden zo základných výsledkov práce McCullocha a Pittsa. Dokázali, že pomocou jednoduchých logických neurónov sa dá zostrojiť neurónová sieť, ktorá simuluje ľubovoľnú Boolovu funkciu. „Neurónovú sieť môžeme zostrojiť pomocou syntaktického stromu tak, že jednotlivé vrcholy reprezentujúce Boolove elementárne funkcie (logické spojky) nahradíme príslušnými logickými neurónmi.“ [10] Na obrázku 2.6 je znázornená konštrukcia neurónovej siete pre formulu $(p \wedge q) \Rightarrow (p \vee q)$



Obr. 2.6: Syntaktický strom a neurónová sieť [6]

Ak chceme zistiť aké Boolove funkcie je logický neurón schopný vyjadriť, musíme sa držať vety: „Logický neurón je schopný simulovať len tie Boolove funkcie, ktoré sú lineárne separovateľné.“ [2] Teda môžeme to zistiť pomocou geometrickej interpretácie výpočtu prebiehajúceho v logickom neuróne. Výpočtová funkcia logického neurónu rozdeľuje vstupné hodnoty na 2 skupiny.

$$w_1x_1 + w_2x_2 + \dots + w_nx_n = \vartheta, \text{pre } w_i = 0, \pm 1$$

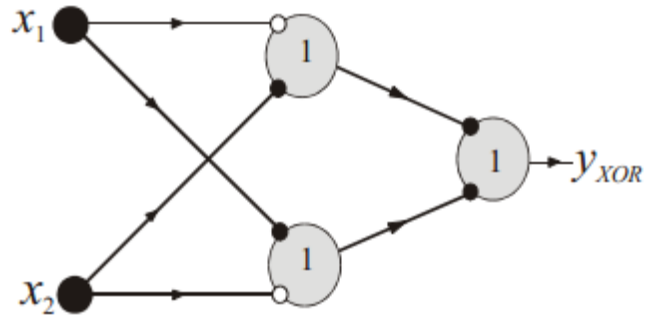
„Boolova funkcia $f(x_1, x_2, \dots, x_n)$ je lineárne separovateľná, ak existuje taká rovina $w_1x_1 + w_2x_2 + \dots + w_nx_n - \vartheta = 0$, ktorá separuje priestor vstupných aktivít tak, že v jednej časti priestoru sú vrcholy ohodnotené 0, zatiaľ čo v druhej časti priestoru sú vrcholy ohodnotené 1.“ [2]

No existuje aj Boolova funkcia, ktorá nie je lineárne separovateľná.

$$\varphi_{XOR}(x_1, x_2) = (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)$$

Môžeme zostrojiť neurónovú sieť pre túto Boolovu funkciu. Táto sieť bude pozostávať z troch vrstiev. „Prvá vrstva sa nazýva vstupná a spravidla len mapuje svoje vstupy na výstupy. Posledná vrstva sa nazýva výstupnou. Medzi vstupnou a výstupnou vrstvou sa môže nachádzať ľubovoľný počet vrstiev (v tomto prípade 1).“

[9] Skryté neuróny budú jednotlivé klauzuly z formuly $\varphi_{XOR}(x_1, x_2)$.



Obr. 2.7: 3-vrstvová neurónová sieť [6]

„Ľubovoľná Boolova funkcia f je simulovaná pomocou 3-vrstvovej neurónovej siete.“ [2] Tým pádom 3-vrstvové neurónové siete obsahujúce logické neuróny sú univerzálne výpočtové zariadenia pre doménu Boolových funkcií. Môžeme simulovať aj funkcie, ktoré nie sú lineárne separovateľné.

Lineárnu separovateľnosť môžeme jednoducho zovšeobecniť na kvadratickú separovateľnosť pomocou kvadratickej nadplochy.

„Boolova funkcia sa nazýva kvadraticky separovateľná, ak existujú také váhové koeficienty w_i, w_{ij} a prahový faktor ϑ , že pre každú špecifikáciu premenných x_1, x_2, \dots, x_n platí“ [2] :

$$y_{REG}(x_1, x_2, \dots, x_n) = 1 \Rightarrow \sum_{i=1}^n w_i x_i + \sum_{\substack{i,j=1 \\ (i < j)}}^n w_{ij} x_i x_j \geq \vartheta$$

$$y_{REG}(x_1, x_2, \dots, x_n) = 0 \Rightarrow \sum_{i=1}^n w_i x_i + \sum_{\substack{i,j=1 \\ (i < j)}}^n w_{ij} x_i x_j < \vartheta$$

2.4 Aktivita neurónov v čase

Môžeme predpokladať, že čas je v neurónovej sieti diskretný a zároveň je daný nezáporným celým číslom. Aktivity neurónovej siete v čase t označíme $x^{(t)}$, v čase $t = 0$ vektor $x^{(0)}$ špecifikuje počiatočné aktivity neurónovej siete. Vzťah pre aktivitu i -teho neurónu v čase t má tvar

$$x_i^{(t)} = s \left(\sum_j w_{ij} x_j^{(t-1)} - \vartheta_i \right)$$

2.5 Konštruktívny dôkaz ekvivalencie neurónovej siete s konečnostavovým automatom

„Neurónová sieť je ekvivalentná s konečnostavovým automatom, ak výstup oboch zariadení je identický pri ľubovoľnom rovnakom vstupe. Pre ekvivalenciu neurónovej siete s konečnostavovým zariadením nezáleží akým spôsobom prebiehajú výpočty pri konštrukcii výstupného reťazca, podstatná je rovnosť výstupných reťazcov pri rovnakom vstupe.“ [2]

Z knihy „Matematická logika“ vieme, že:

„Každá neurónová sieť môže byť reprezentovaná ekvivalentným konečnostavovým strojom s výstupom.

- 1) Množina stavov S je vytvorená všetkými možnými binárnymi vektormi $x_H, S = \{x_H\}$. Nech neurónová sieť má n_H skrytých neurónov a n_A výstupných neurónov, potom kardinalita (počet stavov) množiny S je 2^{n_H} .
- 2) Množina vstupných symbolov je tvorená všetkými možnými binárnymi vektormi $x_\Sigma, \Sigma = \{x_\Sigma\}$, kardinalita tejto množiny je 2^{n_Σ} , kde n_Σ je počet vstupných neurónov.
- 3) Množina výstupných symbolov je tvorená všetkými možnými binárnymi vektormi $x_\Lambda, \Lambda = \{x_\Lambda\}$, kardinalita tejto množiny je 2^{n_Λ} .

- 4) Funkcia $T: S \times \Sigma \rightarrow S$ priradí každému stavu a vstupnému symbolu nový stav. Táto funkcia je špecifikovaná neurónovou sieťou $x_H^{(t+1)} = F(x_\Sigma^{(t)} \oplus x_H^{(t)}; N)$
- Toto priradí dvojici stavu a vstupu v čase $t-1$ nový stav v čase t .
- 5) Funkcia $G: S \times \Sigma \rightarrow \Lambda$ priradí dvojici stavu a vstupu výstupný symbol. Podobne ako aj v predchádzajúcom prípade, aj táto funkcia je plne špecifikovaná $x_\Lambda^{(t+1)} = \tilde{F}(x_\Sigma^{(t)} \oplus x_H^{(t)}; N)$
- 6) Počiatočný stav s_0 sa obvykle vyberie taký, že aktivity skrytých neurónov sú nulové.

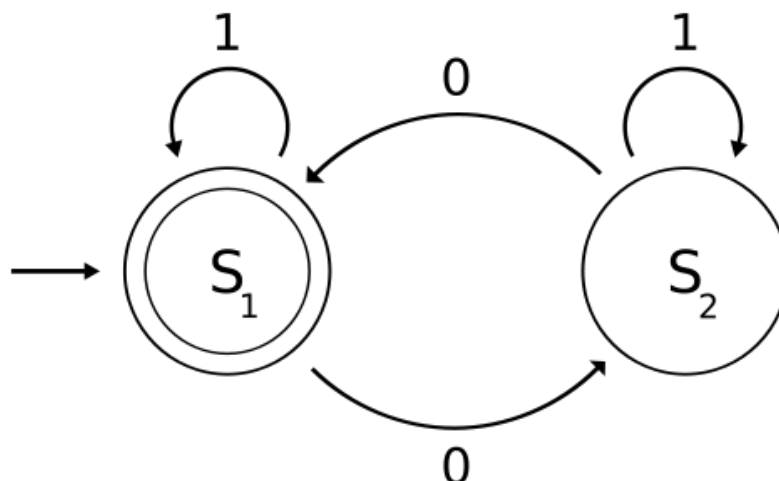
Týmto spôsobom, na základe známej neurónovej siete, sme jednoznačne špecifikovali konečnostavový stroj, ktorý je ekvivalentný s danou neurónovou sieťou.“

[2]

3 Konečnosťavový automat

Konečnosťavový automat je zariadenie, ktorého množina stavov je konečná. Konečnosťavové automaty sú jedným zo základných prostriedkov na popis regulárnych jazykov. Konečnosťavové automaty sa tiež používajú na vyhľadávanie v texte či matematickom popise pamäťových obvodov.

Konečnosťavový automat môžeme opísať stavovým diagramom, formálnym zápisom alebo maticou prechodových funkcií. „Graficky reprezentovaný konečnosťavový automat obsahuje jednotlivé stavy graficky znázornené kružnicami. Každý stav má jednoznačné označenie.“ [7]



Obr. 3.1: Stavový automat

Na obrázku 3.1 máme možnosť vidieť jednoduchý konečnosťavový automat. Obsahuje stavy S_1 a S_2 . Má dva vstupné znaky 1 a 0. Počiatočný a koncový stav je S_1 .

V teórii jazykov je najčastejšie používaný deterministický automat ale nemôže byť použitý pri neskoršom skúmaní jeho ekvivalencie s neurónovými sieťami. Na toto skúmanie použijeme automat, ktorý má aj výstupné hodnoty. To znamená, že stavový diagram automatu má ku každému prechodu priradenú nielen vstupnú hodnotu, ktorou je prechod aktivovaný ale aj výstupnú hodnotu, ktorá je pri aktivácii prechodu vygenerovaná. V informatike sa pre konečnosťavový automat s výstupom používa

Mealyho automat. Mealyho automat pripomína synchronnú komunikáciu. „Výstup Mealyho automatu závisí na spracovávanom vstupnom symbole aj na stave, v ktorom sa automat nachádza.“ [4]

3.1 Definícia Mealyho automatu

Mealyho automat je usporiadaná šesticca $M = (S, \Sigma, \Lambda, T, G, s_0)$, kde

S je konečná množina stavov,

Σ je vstupná abeceda,

Λ je výstupná abeceda,

T je prechodová funkcia, pričom platí

$$T: S \times \Sigma \rightarrow S$$

G je výstupná funkcia, pričom platí

$$G: S \times \Sigma \rightarrow \Lambda$$

$s_0 \in S$ je počiatočný stav.

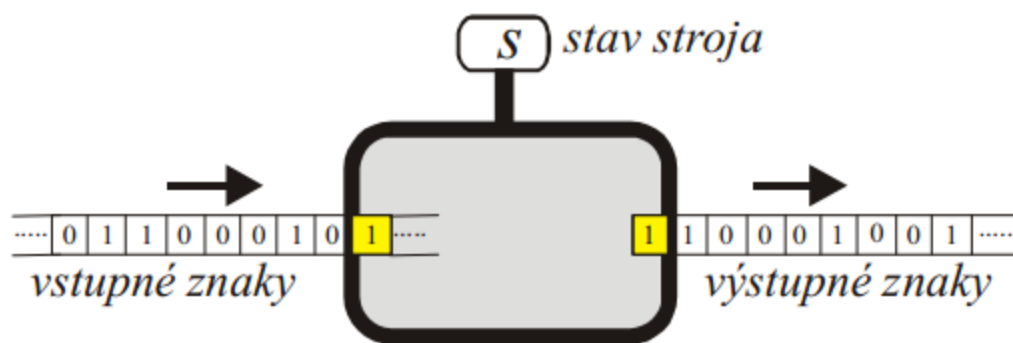
V mnohých literatúrach sa uvádzajú rôzne terminológie spôsobu zápisu. Šesticca sa môže zapisovať aj ako $\langle Z, Q, Y, \Phi, \Psi, q \rangle$ alebo (S, I, O, f, g, s_{ini}) . V definícií môže byť použitá aj sedmica, siedmym parametrom by v tomto prípade bola množina koncových stavov.

„Automat obsahuje dve pásky. Pásky vstupných symbolov a výstupných symbolov (pozri obrázok 3.2). Výstupné symboly sú určené na základe vstupných symbolov a výstupných symbolov stroja

$$stav_{t+1} = T(stav_t, vstupný\ symbol_t)$$

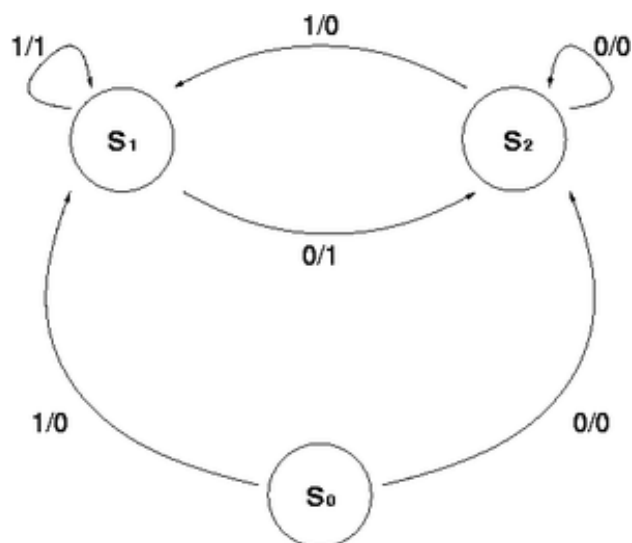
$$výstupný\ symbol_{t+1} = G(stav_t, vstupný\ symbol_t)$$

kde funkcie T a G špecifikujú automat a sú vnímané ako jeho elementárna vlastnosť. Prechodová funkcia T určuje nasledujúci stav na základe momentálneho stavu a vstupného symbolu. Výstupná funkcia G určuje výstupný symbol na základe momentálneho stavu a vstupného symbolu.“ [2]



Obr. 3.2: Konečnostavový stroj

V stavovom diagrame Mealyho automatu (pozri obrázok 3.3) sú pri vstupných symboloch už aj výstupné symboly. Na obrázku vidíme príklad konečnostavového stroja, ktorý obsahuje tri stavy $S = \{S_0, S_1, S_2\}$. Vstupné znaky sú $\Sigma = \{0, 1\}$. Výstupné znaky sú $\Lambda = \{0, 1\}$. Počiatočný stav je S_0 .



Obr. 3.3: Stavový diagram Mealyho automatu

Prechodová funkcia T, nazývaná tiež vnútorná funkcia, určuje prechod do nasledujúceho stavu. Výstupná funkcia G sa používa na určenie výstupného stavu automatu. Funkcie sú zadané v definícii Mealyho automatu.

	T		G	
stav	Prechodová funkcia		Výstupná funkcia	
	0	1	0	1
S ₀	S ₂	S ₁	0	0
S ₁	S ₂	S ₁	1	1
S ₂	S ₂	S ₁	0	0

Tabuľka 3.1: Prechodová a výstupná funkcia konečnostavového stroja

V tabuľke 3.1 je uvedená prechodová a výstupná funkcia stavového diagramu z obrázku 3.3.

V nasledujúcej tabuľke 3.2 je zobrazená postupnosť vstupných znakov, stavov a výstupných znakov pre konečnostavový automat zobrazený na obrázku číslo 3.3, pre vstupnú postupnosť znakov (10110100).

Vstupný symbol	1	0	1	1	0	1	0	0	...
Stav	S ₀	S ₁	S ₂	S ₁	S ₁	S ₂	S ₁	S ₂	...
Výstupný symbol	□	0	1	0	1	1	0	1	0

Tabuľka 3.2: Postupnosť znakov a stavov

3.2 Dôkaz ekvivalencie konečnostavového automatu s neurónovou sieťou

V knihe „Matematická logika“ sa používa na prevod konečnostavového automatu na neurónovú sieť prechodová a výstupná funkcia automatu. Následne z nich sa vytvorí neurónová sieť, ktorá reprezentuje prechodovú resp. výstupnú funkciu a na záver substitúciou týchto reprezentácií získame neurónovú sieť, ktorá je ekvivalentná s konečnostavovým zariadením.

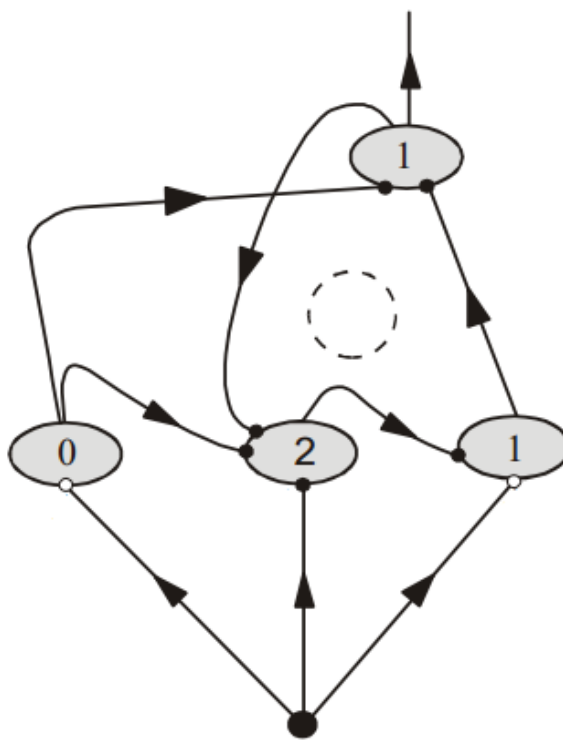
4 Prevod neurónovej siete na konečnostavový automat

Na prevod neurónovej siete potrebujeme poznať parametre siete a vstupné hodnoty. Parametre neurónovej siete môžu byť rôzne. Sieť vieme definovať funkciou, ktorú reprezentuje, taktiež funkciami, ktoré reprezentujú jednotlivé neuróny siete alebo aj jednotlivými prechodmi medzi neurónmi.

4.1 Zadefinovanie neurónovej siete

Pri definícii neurónovej siete pomocou funkcie siete môžu byť výsledky rozdielne, nakoľko rozklad zloženej funkcie na lineárne separovateľné funkcie môže mať viac ako jedno riešenie. Z tohto dôvodu ako aj z dôvodu jednoduchosti zadávania parametrov siete je optimálnejšie definovať parametre siete cez jednotlivé prechody medzi neurónmi.

Prvý krok prevodu neurónovej siete na konečnostavový automat spočíva v definovaní stavov neurónov v čase t (viď kapitola 2.4). Dĺžka času závisí od počtu vstupných hodnôt. Predpokladáme, že stavy (resp. aktivity) neurónov v čase $t=1$ sú nulové. Výstup neurónovej siete v čase $t=1$ je prázdny znak.



Obr. 4.1: Rekurentná neurónová sieť

Aktivity neurónov neurónovej siete na obr. 4.1 pri vstupných hodnotách $x_1 = (1101101010)$ sú zobrazené v tabuľke 4.1

	x_1	x_2	x_3	x_4	x_5
1	1	0	0	0	-
2	1	0	0	0	0
3	0	0	0	0	0
4	1	1	0	0	0
5	1	0	1	0	1
6	0	0	1	0	0
7	1	1	0	1	0
8	0	0	1	0	1
9	1	1	0	1	0

10	0	0	1	0	1
----	---	---	---	---	---

Tabuľka 4.1: Aktivita neurónov v čase – vstup a výstup siete

Stĺpec x_1 reprezentuje vstupné hodnoty a stĺpec x_5 v tabuľke 4.1 reprezentuje výstupné hodnoty rekurentnej neurónovej siete z obrázka 4.1. Pre výpočet aktivity neurónov v čase používame „pamäť“ sietí. „Rekurentné neurónové siete sú na rozdiel od klasických dopredných neurónových sietí schopné spracovávať vstupy s časopriestorovou štruktúrou.“ [8] Napríklad pri výpočte stavu neurónu x_4 v čase t potrebujeme poznať stav x_1 a x_3 v čase $t-1$. Pre zistenie stavu neurónu x_3 v čase $t-1$ musíme zase poznať stav neurónu x_1 , x_2 a x_5 v čase $t-2$. Túto vlastnosť špecifickú pre rekurentné neurónové siete nazývame „pamäť“ sietí.

Aktivity neurónov zo siete na obrázku 4.1 určené pomocou vzorca z kapitoly 2.4:

$$x_1^{(t)} = \text{externý vstup}$$

$$x_2^{(t)} = s(-x_1^{(t-1)} - 0)$$

$$x_3^{(t)} = s(x_1^{(t-1)} + x_2^{(t-1)} + x_5^{(t-1)} - 2)$$

$$x_4^{(t)} = s(-x_1^{(t-1)} + x_3^{(t-1)} - 1)$$

$$x_5^{(t)} = s(x_2^{(t-1)} + x_4^{(t-1)} - 1)$$

4.2 Tvorba konečnostavového automatu

Konečnostavový automat, reprezentujúci neurónovú sieť danú parametrami siete a vstupnými hodnotami, bude obsahovať stavy, ktoré predstavujú skryté neuróny siete. Stav konečnostavového automatu označujeme hodnotou, ktorá reprezentuje hodnoty aktivít všetkých skrytých neurónov. Konečnostavový automat, ktorý reprezentuje

neurónovú sieť zobrazenú na obrázku 4.1, začína stavom S_{000} , nakoľko aktivity troch skrytých neurónov v čase $t=1$ sú nulové.

V prípade rekurentnej neurónovej siete z obrázka 4.1 s aktivitami neurónov definovanými v tabuľke 4.2 vyplýva, že na reprezentáciu siete automatom potrebujeme 4 stavy ($S_{000}, S_{100}, S_{010}$ a S_{101}).

	x_1	x_2	x_3	x_4	x_5
1	1	0	0	0	-
2	1	0	0	0	0
3	0	0	0	0	0
4	1	1	0	0	0
5	1	0	1	0	1
6	0	0	1	0	0
7	1	1	0	1	0
8	0	0	1	0	1
9	1	1	0	1	0
10	0	0	1	0	1

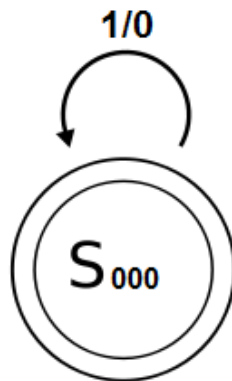
Tabuľka 4.2: Aktivita neurónov v čase – stavy skrytých neurónov

Prechody medzi stavmi konečnostavového automatu sú definované východiskovým stavom, vstupnou hodnotou, výstupnou hodnotou a výstupným stavom. Tieto prechody môžeme jednoducho vyčítať z tabuľky aktivity neurónov v čase. Vstupný symbol, teda stav neurónu x_1 v čase t , je vstupnou hodnotou automatu. Výstupný symbol, teda stav neurónu x_5 v čase $t+1$, je výstupnou hodnotou automatu. Stavy skrytých neurónov v čase t reprezentujú východiskový stav automatu. Stavy skrytých neurónov v čase $t+1$ reprezentujú výstupný stav automatu.

	x_1	x_2	x_3	x_4	x_5
1	1	0	0	0	-
2	1	0	0	0	0

Tabuľka 4.3: Aktivita neurónov v čase $t=1$ a $t=2$ – prechod konečnostavového automatu

Prvý prechod konečnostavového automatu na obrázku 4.2, reprezentujúceho rekurentnú neurónovú sieť z obrázka 4.1, vychádza zo začiatočného stavu S_{000} so vstupnou hodnotou 1, výstupný stav je opäť S_{000} s výstupnou hodnotou 0. Prvý a druhý prechod je totožný.

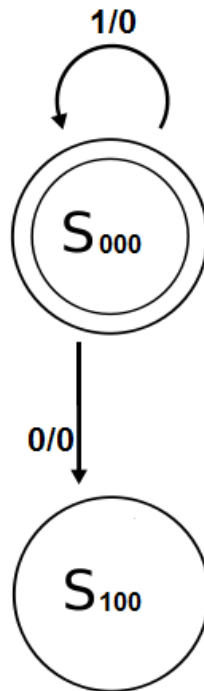


Obr. 4.2: Prvý prechod konečnostavového automatu

	x_1	x_2	x_3	x_4	x_5
3	0	0	0	0	0
4	1	1	0	0	0

Tabuľka 4.4: Aktivita neurónov v čase $t=3$ a $t=4$

Tretí prechod konečnostavového automatu na obrázku 4.3, reprezentujúceho rekurentnú neurónovú sieť z obrázka 4.1, vychádza zo stavu S_{000} so vstupnou hodnotou 0, výstupný stav je S_{100} s výstupnou hodnotou 0 podľa tabuľky 4.4.

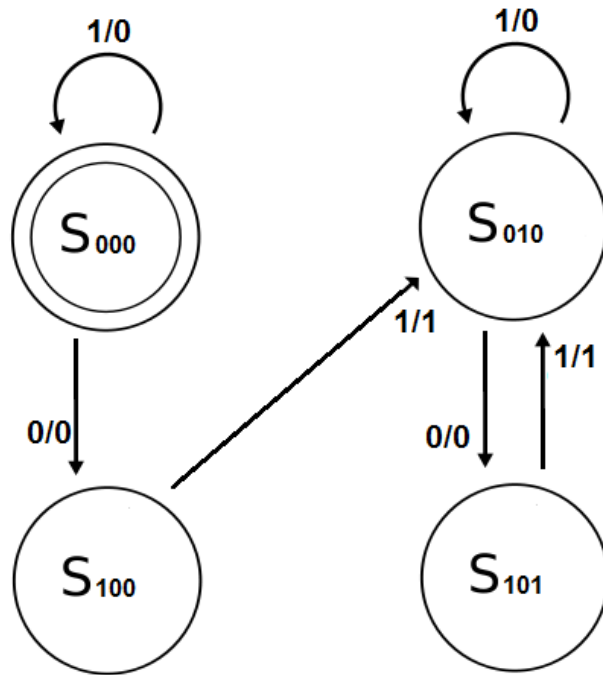


Obr. 4.3: Tretí prechod konečnostavového automatu

Opakovaním tohto kroku dostaneme konečnostavový automat (obrázok 4.4), ktorý reprezentuje rekurentnú neurónovú sieť z obrázka 4.1.

Takýmto postupom získame konečnostavový automat, ktorý reprezentuje neurónovú sieť pri zadaných vstupných hodnotách. To ale znamená, že výsledný konečnostavový automat nemusí byť postačujúci v prípade, že dôjde k zmene vstupných hodnôt neurónovej siete, pričom nedôjde k zmene parametrov siete.

Pre vykreslenie konečnostavového automatu, ktorý bude reprezentovať neurónovú sieť bez ohľadu na vstupný reťazec, treba 2^{n_H} stavov, kde n_H je počet skrytých neurónov siete. Stavby budú reprezentovať všetky možnosti binárnych vektorov.



Obr. 4.4: Konečnosťavý automat reprezentujúci neurónovú sieť

Prechody medzi stavmi v neurónovej sieti bez ohľadu na vstupné hodnoty zistíme testovaním každej hodnoty z množiny vstupných hodnôt na každom zo stavov konečnosťavového automatu, resp. na každej kombinácii stavov skrytých neurónov. V prípade rekurentnej neurónovej siete nesmieme ale zabudnúť na množinu výstupov, ktoré vstupujú späť do siete. V rekurentnej sieti z obrázka 4.1 testujeme 32 možností, teda $4 \cdot 8$, kde 4 je počet možných vstupov v kombinácii s výstupmi (00, 01, 10 alebo 11) a 8 je počet možných stavov (2^{n_H} , kde n_H je počet skrytých neurónov siete).

Z obrázka si zvolíme napríklad prechod zo stavu S_{000} pri vstupe 0. Výstupný stav a výstupnú hodnotu zistíme pomocou vzorcov pre výpočet aktivity neurónov v čase (viď kapitola 4.1). Pri výpočte stavu neurónu x_3 , ale potrebujeme vedieť aj výstup, resp. stav neurónu x_5 v čase $t-1$.

$$x_3^{(t)} = s(x_1^{(t-1)} + x_2^{(t-1)} + x_5^{(t-1)} - 2)$$

vstup	stav			výstup
x_1	x_2	x_3	x_4	x_5
0	0	0	0	0
	1	0	0	0

Tabuľka 4.5: Zisťovanie prechodu zo stavu S_{000} pri vstupe 0 a výstupe 0

vstup	stav			výstup
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	0	0

Tabuľka 4.6: Zisťovanie prechodu zo stavu S_{000} pri vstupe 0 a výstupe 1

Z výpočtov v tabuľke 4.5 a 4.6 vychádza, že nezáleží na predošlom výstupe, zo stavu S_{000} pri vstupnej hodnote 0 nastáva prechod do stavu S_{100} s výstupnou hodnotou 0.

Tento výpočtový krok opakujeme na stave S_{100} pri vstupe 0 v tabuľke 4.7 a 4.8.

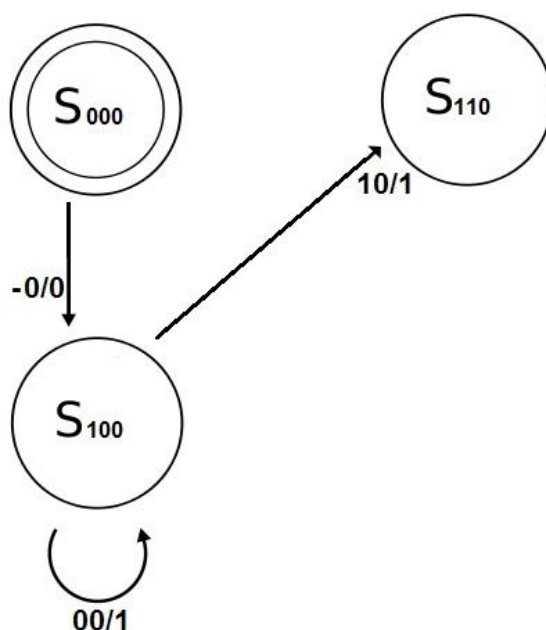
vstup	stav			výstup
x_1	x_2	x_3	x_4	x_5
0	1	0	0	0
	1	0	0	1

Tabuľka 4.7: Zisťovanie prechodu zo stavu S_{100} pri vstupe 0 a výstupe 0

vstup	stav			výstup
x_1	x_2	x_3	x_4	x_5
0	1	0	0	1
	1	1	0	1

Tabuľka 4.8: Zisťovanie prechodu zo stavu S_{100} pri vstupe 0 a výstupe 1

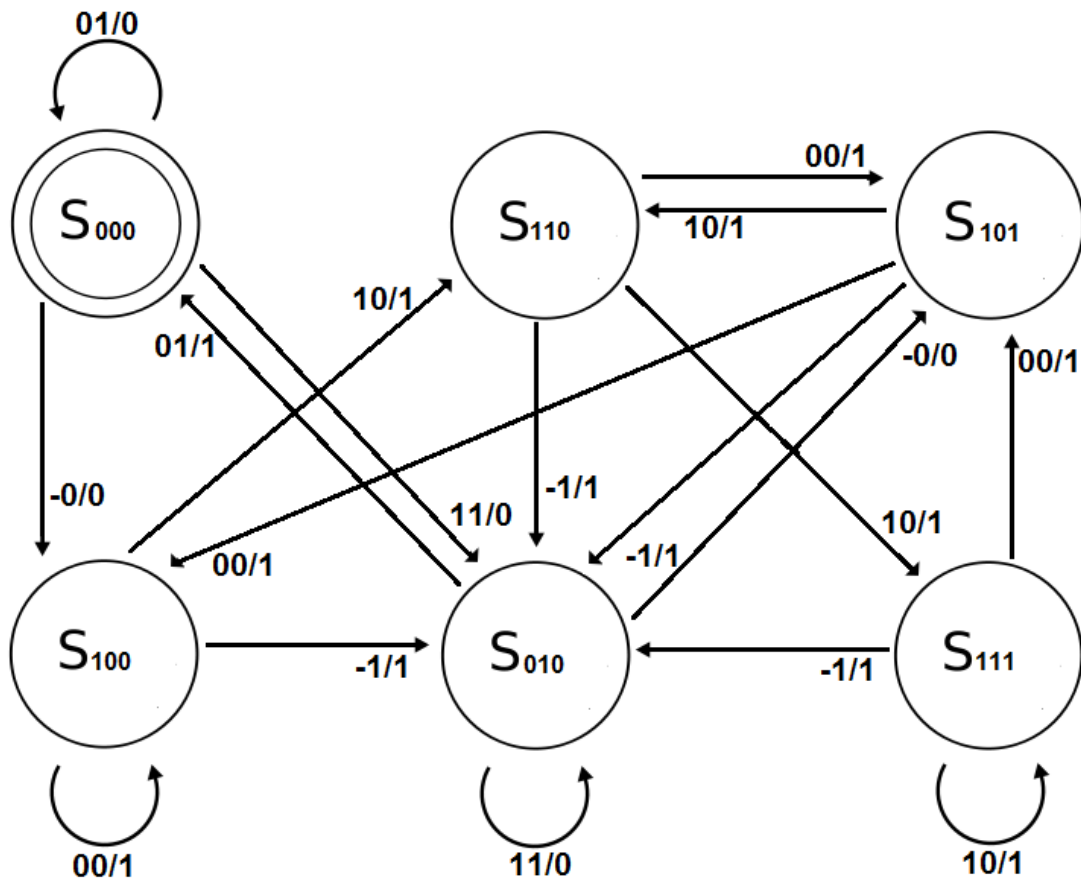
Aplikovaním vzorcov z kapitoly 4.1 sme zistili, že zo stavu S_{100} pri vstupe 0 s výstupom 0 nastáva prechod v konečnostavovom automate do stavu S_{100} s výstupom 1 (tabuľka 4.7). Zároveň sme zistili, že zo stavu S_{100} pri vstupe 0 s výstupom 1 v čase $t-1$ nastáva prechod do stavu S_{110} s výstupom 1 (tabuľka 4.8). Teda pri tom istom vstupe nastáva zo stavu S_{100} prechod s výstupom 1 do dvoch rôznych stavov S_{100} a S_{110} v závislosti od výstupu v čase $t-1$. Podobná situácia v tabuľke 4.2 nenastala. Túto situáciu vyriešime pridaním výstupu k vstupnej hodnote.



Obr. 4.5: Konečnostavový automat s prechodmi zo stavu S_{100} pri vstupe 0 s rôznymi výstupmi v čase $t-1$

Rozšírime množinu vstupných hodnôt konečnostavového automatu. Výstup konečnostavového automatu pôjde na vstup automatu. Vstupná hodnota bude zložená z predošlého výstupu a aktuálneho vstupu. Aplikujeme túto novú skutočnosť pri vykresľovaní neurónovej siete automatom na obrázku 4.5. Na obrázku 4.5 sú dva prechody zo stavu S_{000} do stavu S_{100} . Jeden prechod pri výstupe v čase $t-1$ 1 a vstupnej hodnote 0, druhý pri výstupe v čase $t-1$ 0 a vstupnej hodnote 0. V rámci šetrenia priestoru pri vykresľovaní sú tieto prechody spojené do jedného, v ktorom nezáleží na výstupnej hodnote v čase $t-1$, nakoľko pri oboch možnostiach je rovnaká výstupná hodnota a výstupný stav. Výstupný stav v čase $t-1$ je v takomto prípade v množine vstupných hodnôt označovaný „-“.

Týmto postupom získame konečnosťový automat, ktorý reprezentuje neurónovú sieť bez ohľadu na vstupné hodnoty (pozri obrázok 4.6).



Obr. 4.6: Konečnosťový automat reprezentujúci neurónovú sieť pri akýchkoľvek vstupoch

Na obrázku 4.6 je zobrazení konečnosťový automat reprezentujúci neurónovú sieť z obrázka 4.1 pri všetkých možných vstupoch. Na obrázku vidíme 6 stavov, pričom na vykreslenie automatu treba 2^{n_H} stavov, kde n_H je počet skrytých neurónov siete (v našom prípade 3 skryté neuróny, čiže 8 stavov). Do zvyšných dvoch stavov S_{001} a S_{011} ale konečnosťový automat nikdy nevkročí. Túto skutočnosť si môžeme aj overiť:

Zadefinujeme si stav, ktorý chceme získať (tabuľka 4.9). Teda výsledný stav S_{001} .

vstup	stav			výstup
x_1	x_2	x_3	x_4	x_5
	0	0	1	

Tabuľka 4.9: Krok 1 – overenie nedosiahnuteľného stavu S_{001}

Zo vzorcov aktivity neurónov v čase vieme, že:

$$x_2^{(t)} = s(-x_1^{(t-1)} - 0)$$

$$x_3^{(t)} = s(x_1^{(t-1)} + x_2^{(t-1)} + x_5^{(t-1)} - 2)$$

$$x_4^{(t)} = s(-x_1^{(t-1)} + x_3^{(t-1)} - 1)$$

$$x_5^{(t)} = s(x_2^{(t-1)} + x_4^{(t-1)} - 1)$$

Na to, aby $x_2^{(t)}$ bolo rovné 0, musí byť hodnota $x_1^{(t-1)}$ rovná 1. Hodnota $x_3^{(t)}$ má byť rovná taktiež 0 a nakoľko už poznáme hodnotu $x_1^{(t-1)}$, zo vzorca pre výpočet aktivity neurónu x_3 v čase t vyplýva, že hodnoty $x_2^{(t-1)}$ a $x_5^{(t-1)}$ sú rovné 0. Ak hodnota $x_4^{(t)}$ má byť 1 a vieme, že hodnota $x_1^{(t-1)}$ je tiež 1, zo vzorca pre výpočet aktivity neurónu x_4 v čase t vyplýva, že hodnota $x_3^{(t-1)}$ musí byť väčšia alebo rovná 2. Táto skutočnosť ale pri binárnych stavoch neurónov nemôže nastať. Preto neexistuje situácia prechodu do stavu S_{001} v konečnostavovom automate. Rovnako postupujeme aj pri overovaní stavu S_{011} . Opäť zistíme, že hodnota $x_1^{(t-1)}$ musí byť 1. Pri zisťovaní hodnoty $x_3^{(t-1)}$ pomocou vzorca pre výpočet aktivity neurónu x_4 v čase t získame ten istý problém ako v predošlom prípade, teda hodnota $x_3^{(t-1)}$ musí byť väčšia alebo rovná 2.

vstup	stav			výstup
x_1	x_2	x_3	x_4	x_5
1	0	≥ 2		0
	0	0	1	

Tabuľka 4.10: Krok 2 – overenie nedosiahnuteľného stavu S_{001}

Týmto postupom sme dokázali, že stavy S_{001} a S_{011} v konečnostavovom automate, reprezentujúcom neurónovú sieť z obrázka 4.1, nemusia byť zakreslené. Tieto stavy nikdy nenastanú.

4.3 Zhodnotenie

V kapitole 4.2 je podrobný popis tvorby konečnostavového automatu, ktorý je odpoveďou na neurónovú sieť so zadaným vstupným reťazcom, ako aj tvorba konečnostavového automatu, ktorý reprezentuje neurónovú sieť pri akomkoľvek vstupnom reťazci. Tento postup bol v odbornej literatúre iba zbežne opísaný, konkrétne kroky tohto prevodu sú výsledkom teoretickej časti práce na bakalárskom projekte.

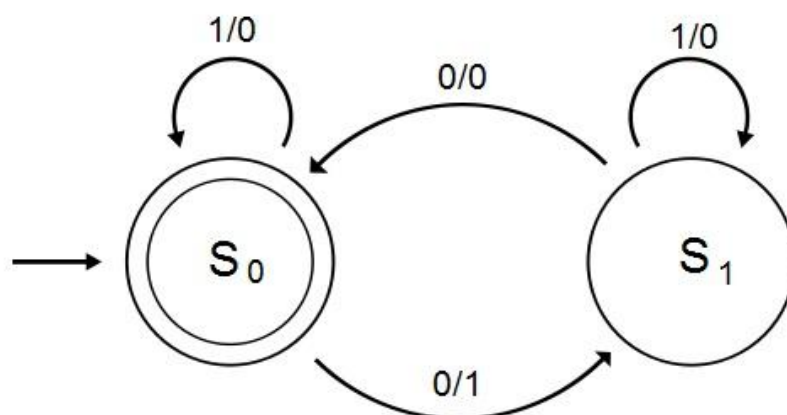
V praktickej časti práce dochádza k vykresľovaniu konečnostavového automatu, ktorý reprezentuje neurónovú sieť so zadaným vstupných reťazcom. Ako je v kapitole 4.2 preukázané, vykresľovanie konečnostavového automatu, ktorý reprezentuje neurónovú sieť pri akomkoľvek vstupnom reťazci, je priestorovo náročné a môže sa stať neprehľadným.

5 Prevod konečnostavového automatu na neurónovú sieť

Pre prevod konečnostavového automatu na neurónovú sieť použijeme znalosti z kapitoly 3. V prvom kroku treba zdefinovať konečnostavový automat a určiť jeho prechodovú a výstupnú funkciu.

5.1 Zadeinovanie konečnostavového automatu

V prvom kroku vytvoríme tabuľku (tabuľka 5.1) prechodovej a výstupnej funkcie konečnostavového automatu na obrázku 5.1.



Obr. 5.1: Konečnostavový automat s výstupom

	Prechodová funkcia		Výstupná funkcia	
	0	1	0	1
S_0	S_1	S_0	1	0
S_1	S_0	S_1	0	0

Tabuľka 5.1: Prechodová a výstupná funkcia konečnostavového automatu

Zistené hodnoty si rozpišeme do tabuľky pre zistenie binárnych reprezentácií funkcie. V prechodovej funkcii rozpisanej v tabuľke 5.2 zameníme výstupné stavy za binárne hodnoty, ktoré reprezentujú výstup prechodovej funkcie. Výstupná funkcia je rozpísaná v tabuľke 5.3.

Východiskový stav	Vstup	Binárna reprezentácia	Výstupný stav	Binárna reprezentácia výstupu
S_0	0	0,0	S_1	1
S_0	1	0,1	S_0	0
S_1	0	1,0	S_0	0
S_1	1	1,1	S_1	1

Tabuľka 5.2: Prechodová funkcia konečnosťového automatu

Východiskový stav	Vstup	Binárna reprezentácia	Výstup
S_0	0	0,0	1
S_0	1	0,1	0
S_1	0	1,0	0
S_1	1	1,1	0

Tabuľka 5.3: Výstupná funkcia konečnosťového automatu

Následne určíme funkciu, ktorú reprezentuje prechodová a výstupná funkcia konečnosťového automatu. Môžeme pri tom použiť Karnaughovu mapu. Do Karnaughovej mapy zapíšeme hodnoty z tabuľky funkcie tak, že prvá hodnota binárnej reprezentácie funkcie predstavuje 1. stĺpec (X) v Karnaughovej mape a druhá hodnota binárnej reprezentácie funkcie predstavuje 1. riadok (Y) v mape. Binárnu reprezentáciu výstupu prechodovej funkcie, či výstup funkcie pri výstupnej funkcii konečnosťového automatu, zapíšeme do obsahu tabuľky, respektíve ako hodnoty Karnaughovej mapy.

X\Y	0	1
0	1	0
1	0	1

Tabuľka 5.4: Karnaughova mapa pre prechodovú funkciu

Z tabuľky 5.4, Karnaughova mapa pre prechodovú funkciu, vyplýva, že výstup 1 nastáva iba v prípade, ak X a zároveň Y sú 0 alebo X a zároveň Y sú 1. Z tejto skutočnosti môžeme vyhodnotiť tvar prechodovej funkcie konečnosťového automatu z obrázka 5.1. Prechodová funkcia konečnosťového automatu má tvar:

$$(NOT X AND NOT Y) OR (X AND Y)$$

Rovnakým spôsobom vyhodnotíme aj výstupnú funkciu konečnosťového automatu.

X\Y	0	1
0	1	0
1	0	0

Tabuľka 5.5: Karnaughova mapa pre výstupnú funkciu

Z tabuľky 5.5 vyplýva, že výstup 1 nastáva iba v prípade ak X a zároveň Y sú 0. Výstupná funkcia konečnostavového automatu má tvar:

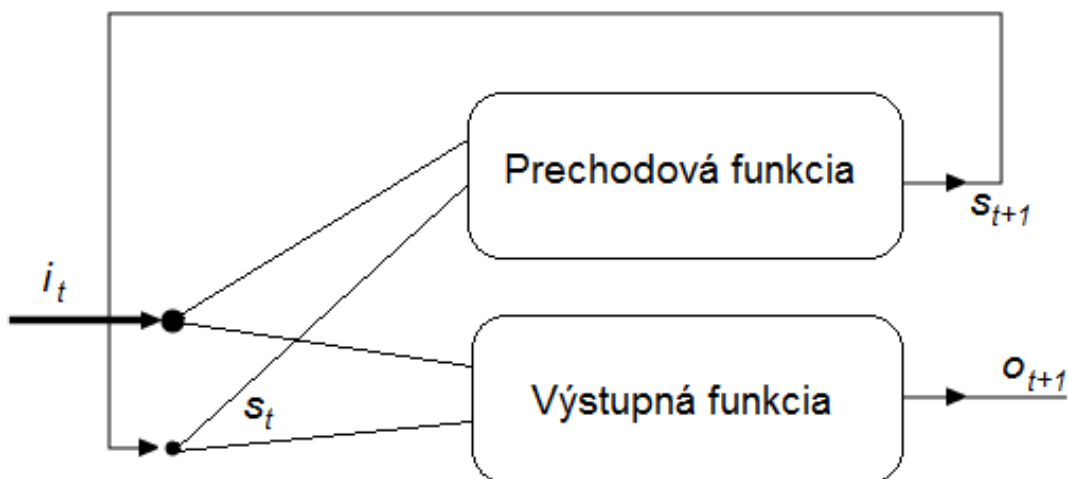
$$NOT X AND NOT Y$$

5.2 Tvorba neurónovej siete

Konečnostavový automat má 1 pásku vstupných hodnôt. Na implementáciu prechodovej a výstupnej funkcie potrebujeme ale dve hodnoty X a Y. Tieto znaky si reprezentujeme v obrázku 5.2 ako „i“ – vstupná hodnota a „s“ – výstupné hodnoty prechodovej funkcie. Výstup neurónovej siete, ktorý je zároveň aj výstupom konečnostavového automatu definujeme ako „o“. Tieto znaky sú medzi sebou v dvoch funkčných reláciách, kde f je prechodová funkcia a g je výstupná funkcia konečnostavového automatu:

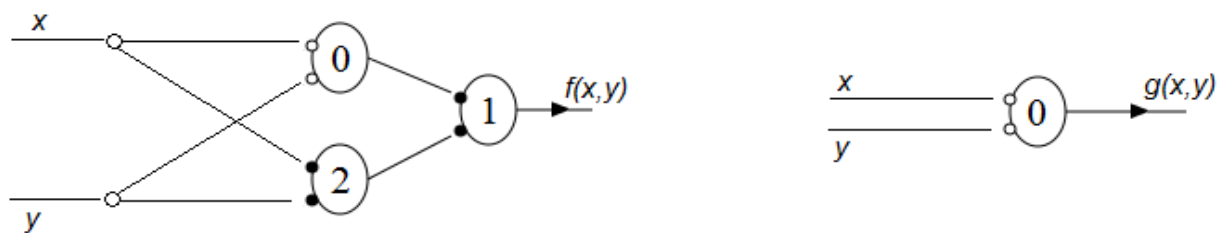
$$s^{(t+1)} = f(s^{(t)}, i^{(t)})$$

$$o^{(t+1)} = g(s^{(t)}, i^{(t)})$$



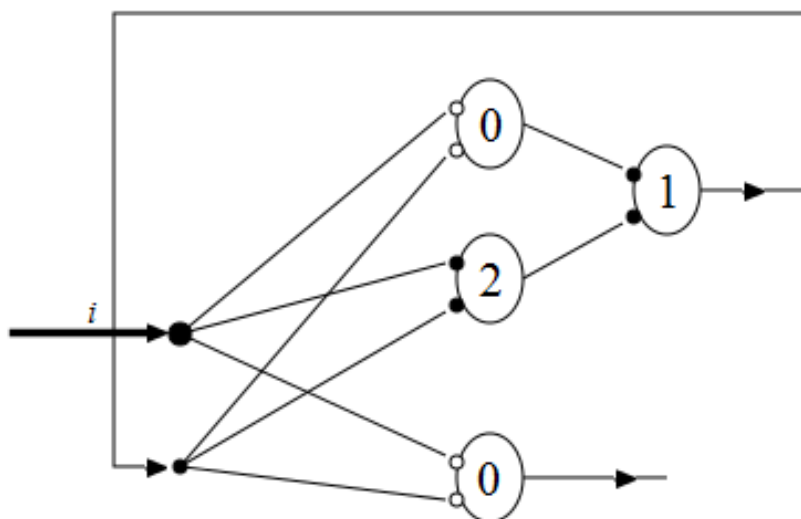
Obr. 5.2: Schematická reprezentácia konečnostavového automatu reprezentovaného prechodovou a výstupnou funkciou

Prechodovú a výstupnú funkciu konečnosťového automatu na obrázku 5.1 špecifikujeme neurónovou sieťou (obrázok 5.3).



Obr. 5.3: Prechodová a výstupná funkcia reprezentovaná neurónovými sieťami

Substitúciou týchto neurónových sietí z obrázku 5.3 do schémy na obrázku 5.2 získame rekurentnú neurónovú sieť, ktorá je ekvivalentná s konečnosťovým automatom na obrázku 5.1.



Obr. 5.4: Rekurentná neurónová sieť ekvivalentná s konečnosťovým automatom

5.3 Zhodnotenie

„Pre ekvivalenciu neurónovej siete s konečnosťovým strojom je nepodstatné akým spôsobom sa vstupné symboly zobrazujú na výstupné symboly, t.j. akým spôsobom prebiehajú výpočty pri konštrukcii výstupného reťazca, podstatná je rovnosť výstupných reťazcov pre rovnaký vstupný reťazec pre obe zariadenia.“ [2]

V kapitole 5.2 sme na príklade ukázali ako vytvoriť neurónovú sieť, ktorá je ekvivalentná so zadaným konečnostavovým automatom. Táto sieť je ekvivalentná k automatu pri akejkol'vek postupnosti vstupných hodnôt. Opis tohto prevodu je rozšíreným opisom postupu prevodu konečnostavového automatu na neurónovú sieť v knihe „Matematická logika“.

Literatúra

- [1] KVASNIČKA, V. a kol.: *Úvod do teórie neurónových sietí*. Vydavateľstvo IRIS, 1997
- [2] KVASNIČKA, V., POSPÍCHAL, J. : *Matematická logika*. STU vydavateľstvo, Bratislava 2007
- [3] MAKULA, Matej: *Sila a obmedzenia rekurentných neurónových sietí pri spracovaní postupností symbolov*, Bratislava: FIIT STU, 2009.76 s. Dizertačná práca.
- [4] VANÍČEK, Jiří et al.: *Teoretické základy informatiky*, 1.vyd. Bratislava: Alfa Publishing, 2007. 431 s. ISBN 80-903962-4-1
- [5] BAYER, J., HANZÁLEK, Z., ŠUSTA, R.: *Logické systémy pro řízení*, Vydavateľstvo ČVUT, Fakulta elektrotechnická, Praha, 2000, ISBN 80-01-02147-5
- [6] ČERNĀNSKÝ, Michal : *Logické neuróny a neurónové siete (podľa McCullocha a Pittsa)*,
http://www2.fiit.stuba.sk/~cernans/nn/nn_texts/neuronove_siete_priesvitky_01.pdf
- [7] CHUDÁ, D. et al.: *Teoretické základy informatiky*. 2006
<http://student.fiit.stuba.sk/~zuber06/TZI/Ucebny%20mat%20kap%201%20teoria%20mnozina.pdf>
- [8] TREBATICKÝ, Peter: *Trénovanie rekurentných neurónových sietí technikou rozšíreného Kalmanovho filtra*
<http://acm.vsb.cz/is/2004/finale/prispevky/trebaticky.pdf>
- [9] PAVLECH, Michal: *Neurónové siete v C#*. UTB ve Zlíně, Fakulta aplikované informatiky, 2009. Diplomová práca.
- [10] BAHNA, R.: *Využitie neurónových sietí na riešenie logických problémov*. Diplomová práca, FEI STU, Bratislava 1998.

Príloha A

Technická dokumentácia

Táto príloha je zameraná na vysvetlenie riešenia jednotlivých častí implementácie a na predstavenie vybraných algoritmov. Keďže program je rozdelený na dve časti: prevod konečnostavového automatu na neurónovú sieť a prevod neurónovej siete na konečnostavový automat, je takýmto spôsobom rozdelená aj táto príloha.

A.1 Výber programovacieho jazyka

Nakoľko väčšinu vysokoškolských projektov som vypracovávala v programovacom jazyku C, rozhodla som sa ho spočiatku použiť pre vypracovanie technickej časti bakalárskej práce. Keďže veľmi dôležitou časťou výstupného programu je grafická implementácia, po dlhšom bádání som usúdila, že grafické knižnice v jazyku C nie sú dostačujúce alebo by ich implementácia bola časovo zložitá.

Veľký dôraz som pri tvorbe programu venovala aj jednoduchej inštalácii. V prípade vypracovania práce v jazyku C, by kompilácia programu v inom prostredí ako Windows mohla spôsobovať problémy hlavne pri alokácii pamäte. Zároveň som chcela spraviť program dostupný online, čo by v prípade jazyka C znamenalo, vytvoriť CGI (*Common Gateway Interface*), zabezpečiť server, ktorý spúšťa súbory EXE a zamedziť prípadným ohrozeniam serveru (vytvoriť zaheslovaný prístup k programu).

Z týchto dôvodov som ako hlavný programovací jazyk praktickej časti bakalárskej práce zvolila JavaScript. Ďalším dôvodom bola aj jeho syntaktická podobnosť s jazykom C. Pre vykresľovanie konečnostavového automatu a neurónovej siete som vybrala JavaScriptovú knižnicu Raphaël (<http://raphaeljs.com/>) pre vektorové vykresľovanie. Knižnica Raphaël je chránená licenciou MIT, ktorá je súčasťou súboru raphael.js.

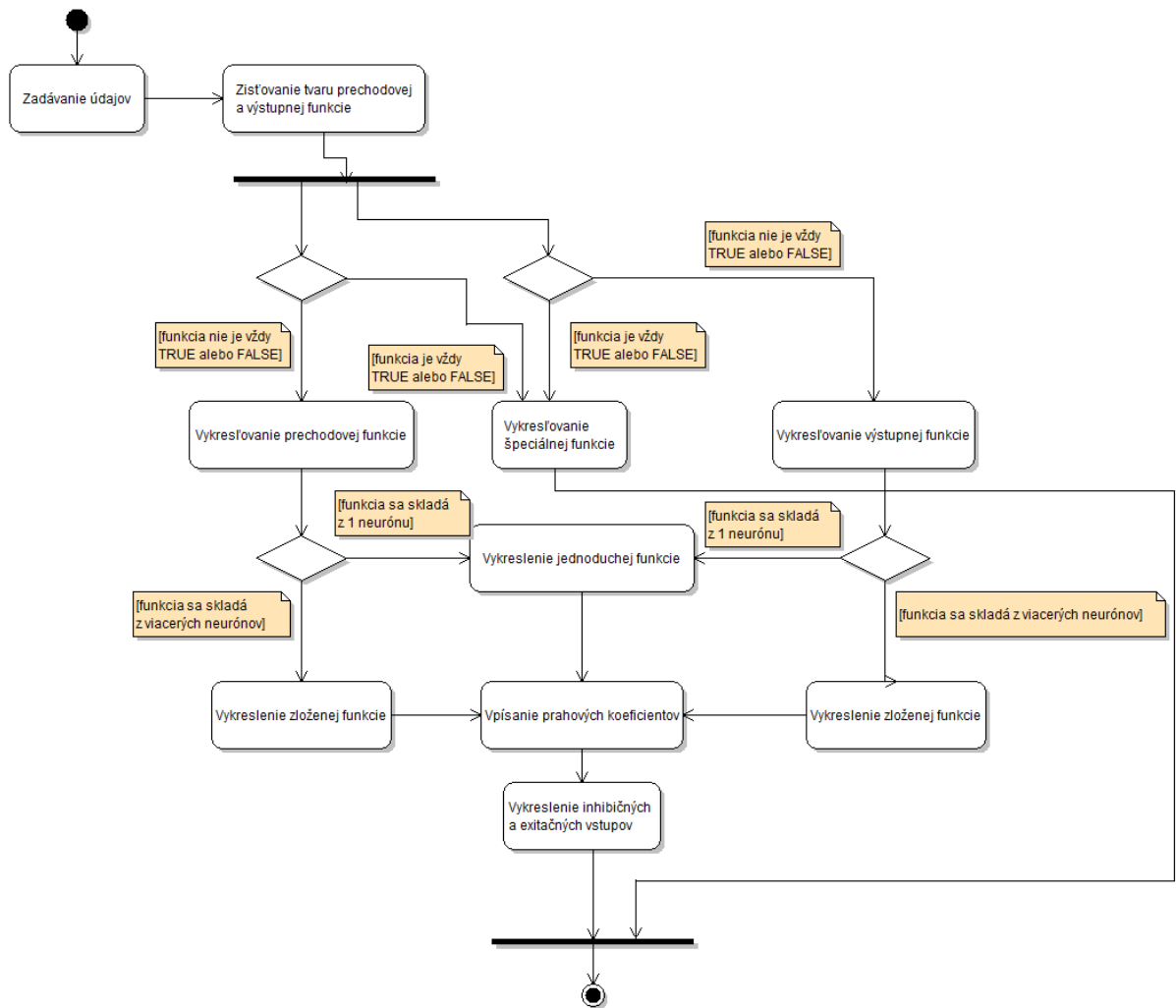
Jazyk JavaScript je v programe kombinovaný s jazykom HTML a CSS. Pri tvorbe kódu niekoľkokrát nastala situácia, kedy funkcia niesla názov tzv. „Build-in“ funkcie. Pre vyhnutie sa takýmto nedorozumeniam bola zvolená slovenčina ako jazyk, v ktorom sú opísané názvy premenných a funkcií.

A.2 Implementácia prevodu konečnostavového automatu na neurónovú sieť

Pri implementácii prevodu konečnostavového automatu na neurónovú sieť sa držíme prevodu z kapitoly 5, kde je krok po kroku vysvetlený postup prevodu automatu na sieť. Použitie tejto časti programu v praxi je opísané v kapitole C.1. Pri návrhu praktickej implementácii by sa dali jednotlivé činnosti opísať diagramom činností na obrázku A.1. Zo strany používateľa sa očakáva interakcia, resp. činnosť „Zadávanie údajov“. Program následne zistí tvar prechodovej a výstupnej funkcie. Ďalej zistí, či prechodová alebo výstupná funkcia sú zložené z 1 alebo viac neurónov a následne dané siete vykreslí do schémy z obrázku 5.2, vypíše prahové koeficienty neurónov a zaznačí inhibičné a exitačné vstupy.

Po zvolení prevodu z automatu na sieť na hlavnej stránke programu (obrázok C.1), sa načíta HTML stránka, ktorá obsahuje JavaScriptové funkcie. Pri načítaní HTML stránky sa spustí funkcia formular(), ktorá kontroluje zadanú hodnotu v boxe číslo 3 na obrázku C.2. Pomocou „onkeydown“ a „onkeyup“ zisťuje zadanú hodnotu podľa ktorej vypíše formulár (obrázok C.3) pre zadanie prechodov medzi stavmi v konečnostavovom automate.

```
kanaly.onkeydown = kanaly.onkeyup = function() {  
    kanaly_tabulka(this.value, element2);  
};
```



Obr. A.1: Diagram činností pre prevod konečnosťového automatu na neurónovú sieť

Funkcia `kanaly_tabulka(kanaly, pozicia)` vypíše na zadanú pozíciu na stránke formulár pre zadanie prechodov medzi stavmi ktorých počet sa rovná premennej „kanaly“. Po vyplnení údajov a zaslaní formulára sa údaje zašlú do funkcie `nacitajUdaje(form)`. Vytvorí sa pole pre prechody medzi stavmi a údaje z formulára sa doň zapíšu v usporiadanom poradí. Teraz treba zistiť akú funkciu reprezentuje prechodová a výstupná funkcia konečnosťového automatu.

```
//zisti prechodovu funkciu
automat_functions(kanaly, 3);
```

```
//zisti vystupnu funkciu
automat_functions(kanaly, 2);
```

```
nacrtni(kanaly, stavy);
```

Funkcia `automat_functions(kanaly, funkcia)` je prakticky zjednodušená implementácia Karnaughovej mapy. Táto funkcia je prispôbena na zistenie prechodovej alebo výstupnej funkcie, ktorá sa skladá z dvoch binárnych reprezentácií vstupov (viď kapitola 5 tabuľky 5.2 až 5.5). Funkcia najprv zistí, aký je súčet hodnôt funkcie. Napríklad, ak je binárna reprezentácia výstupu 1001, súčet bude 2.

```
for(i=0; i < kanaly ; i++){  
    pomocna1 = parseFloat(pomocna1)+parseFloat(pole[i][funkcia]);  
}
```

Následne podľa tejto hodnoty vyberá, resp. sortuje možné tvary funkcie. Napríklad, ak je súčet hodnôt funkcie 0, je samozrejme, že funkcia je vždy FALSE. Takýmto spôsobom vydedukujeme tvar prechodovej alebo výstupnej funkcie.

```
//na zaklade sumy hodnot funkcie vysortuj jednotlivé funkcie  
if(pomocna1==0){  
    content += ("funkcia je NOT x1 AND x1 - vždy FALSE<br/>");  
    cislo_funkcie=0;  
}
```

Po zistení tvaru prechodovej a výstupnej funkcie sa púšťa funkcia `nacrtni(kanaly, stavy)`, ktorá rieši vykresľovanie neurónovej siete. V prvom rade sa vytvorí priestor pre vykresľovanie za pomoci JavaScriptovej knižnice Raphaël. Nakoľko vieme, že každá sieť bude mať vstupný kanál, vykreslí sa šípka do vstupu, kruh vstupu a nad šípkou sa vypíše písmeno „i“.

```
paper = Raphael(300, 300, 520, 600); //vytvor priestor pre kreslenie  
  
var circle = paper.circle(80, 150, 5); //vykresli kruh - vstup i  
circle.attr("fill", "#000000");  
circle.attr("stroke", "#000000");
```

```

//ciara vstupu i
var vstupi = paper.path("M65,154l10.556-4.363l-10.556-
4.363v2.5H2.813v3.5h62.187V154z").attr({fill: "#000", stroke: "none"});
var txt = paper.text(40, 140, "i").attr(
  {"font-family":"serif",
  "font-style":"italic",
  "font-size":"15"});

```

Následne funkcia nacrtní(kanály, stavy) vyberie funkciu pre pokračovanie vykresľovania podľa počtu stavov – 1 alebo 2. Pri dvoch stavoch púšťa funkciu vykresliVelkuSiet(). Funkcia vykresliVelkuSiet() vyberie ďalšiu funkciu pre vykreslenie prechodovej a výstupnej funkcie.

```

function vykresliVelkuSiet(){

    var circle2 = paper.circle(80, 210, 3);    //vykresli kruh - vstup
prechodovej funkcie i2
    circle2.attr("fill", "#000000");
    circle2.attr("stroke", "#000000");

    var c711 = paper.path("M77 210l-10 -2.5l0 5l10 -2.5"); //sipka
vstupu prechodovej funkcie do vstupu i2
    c711.attr("fill", "#000000");

    /*Pre funkcie ktore su vzdy TRUE alebo FALSE sa vykresluje
neuronova siet osobitne*/

    if((prechodova==0) || (prechodova == 15)) {
        specialnaFunkcia(0);    //0 - jedna sa o prechodovu
    }
    else {
        vykresliPrechodovuFunkciu();
    }
    if((vystupna==0) || (vystupna == 15)) {

```

```

        specialnaFunkcia(1); //1 - jedna sa o vystupnu
    }
    else {
        vykresliVystupnuFunkciu();
    }
}

```

V prípade, že je prechodová alebo výstupná funkcia stále TRUE alebo FALSE, o vykreslenie sa stará samostatná funkcia `specialnaFunkcia(funkcia)`, nakoľko čiary do neurónov sú pre tieto funkcie špecifické.

Funkcie `vykresliPrechodovuFunkciu()` a `vykresliVystupnuFunkciu()` zisťujú, z koľkých neurónov je zložená funkcia a vykresľujú dané neuróny. Funkcia `vykresliPrechodovuFunkciu()` zároveň vytvorí cestu z konečného neurónu prechodovej funkcie do vstupu prechodovej funkcie späť do siete. Funkcia `vykresliVystupnuFunkciu()` tvorí výstup celej siete. Tieto dve funkcie sú veľmi podobné, líšia sa hlavne x-ovými a y-novými pozíciami pre vykresľovanie objektov. V prípade, že prechodová alebo výstupná funkcia je zložená z troch neurónov, teda funkcia je (NOT X AND Y) OR (X AND NOT Y) alebo (NOT X AND NOT Y) OR (X AND Y), tieto funkcie vykreslia kompletnú neurónovú reprezentujúcu danú funkciu. V ostatných prípadoch vykreslia iba elipsy neurónov a ich vstupy a spustia funkcie `vpisPrahovyKoefficient(funkcia)` a `vykresliExiInhvstupy(funkcia)`. Premenná „funkcia“ vo funkcií `vpisPrahovyKoefficient` a `vykresliExiInhvstupy` hovorí, či sa jedná o prechodovú alebo výstupnú funkciu a podľa toho zvolí súradnice, ktoré bude používať.

```

if(funkcia==0){
    //prechodova funkcia - suradnice
        var x = 205;
        var y = 73;
        var hodnotaFunkcie = prechodova;
    }
    else {
        //vystupna funkcia - suradnice
        var x = 205;
        var y = 213;
    }
}

```

```

        var hodnotaFunkcie = vystupna;
    }

```

Funkcia `vpisPrahovyKoefficient(funkcia)` podľa premennej „`hodnotaFunkcie`“ zistí, aký prahový koeficient má neurón. Vzhľadom na funkciu vpíše hodnotu prahového koeficientu do neurónu. Napríklad:

```

var txt = paper.text(x+14, y+2, "1").attr(
    {"font-family": "serif",
     "font-size": "22"});

```

Funkcia `vykresliExiInhvstupy` následne pred elipsy neurónov vykreslí exitačné a inhibičné vstupy podľa tvaru prechodovej alebo výstupnej funkcie. Príklad kódu, ktorý vykresľuje exitačné a inhibičné vstupy, ak má neurón 2 vstupné hodnoty:

```

/*BEGIN vykresli exitacny alebo inhibicny vstup ak neuron ma 2 vstupy*/
if((hodnotaFunkcie!=7)&&(hodnotaFunkcie!=8)&&(hodnotaFunkcie!=9)
&&(hodnotaFunkcie!=10)){

    if((hodnotaFunkcie==1)||(hodnotaFunkcie==2)||(hodnotaFunkcie
==13)||(hodnotaFunkcie==14)){
        //not i
        var circle10 = paper.circle(x, y-5, 3);    //vykresli kruh -
inhibicny vstup
        circle10.attr("fill", "#FFFFFF");
        circle10.attr("stroke", "#000000");
    }
    else{
        //i
        var circle10 = paper.circle(x, y-5, 3);    //vykresli kruh -
exitacny vstup
        circle10.attr("fill", "#000000");
        circle10.attr("stroke", "#000000");
    }
}

```

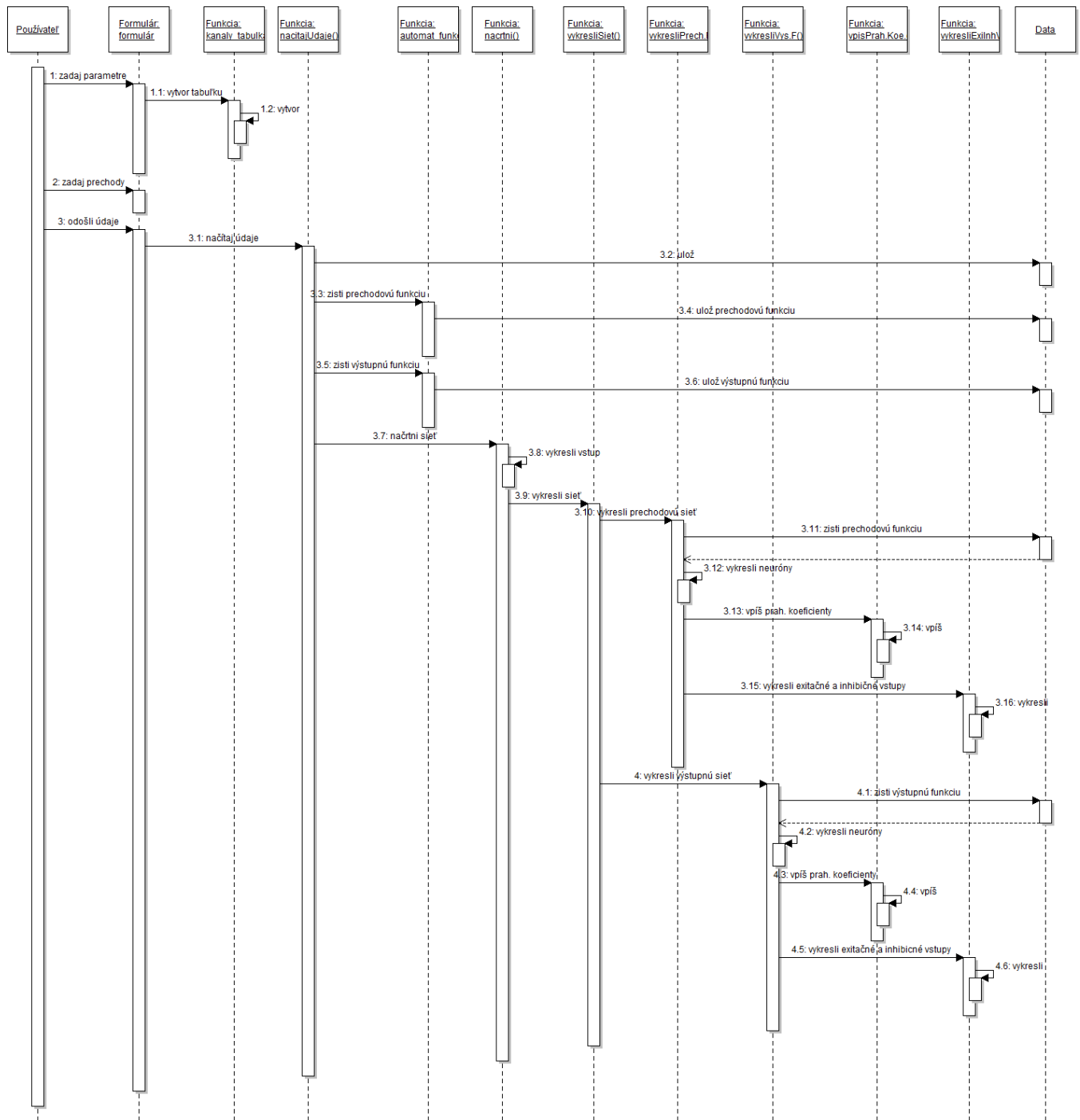
```

        if((hodnotaFunkcie==1)||(hodnotaFunkcie==3)||(hodnotaFunkcie
==12)||(hodnotaFunkcie==14)){
            //not i2
                var circle11 = paper.circle(x, y+12, 3);    //vykresli kruh -
inhibicny vstup
                circle11.attr("fill", "#FFFFFF");
                circle11.attr("stroke", "#000000");
            }
            else{
                //i2
                var circle11 = paper.circle(x, y+12, 3);    //vykresli kruh -
exitacny vstup
                circle11.attr("fill", "#000000");
                circle11.attr("stroke", "#000000");
            }
        }
    }
    /*END vykresli exitacny alebo inhibicny vstup ak neuron ma 2 vstupy*

```

Po vykreslení všetkých neurónov, vstupov do neurónov, výstupov z prechodovej a výstupnej funkcie, či exitačných a inhibičných vstupov je vykreslená neurónová sieť ekvivalentná konečnostavového automatu zo vstupu.

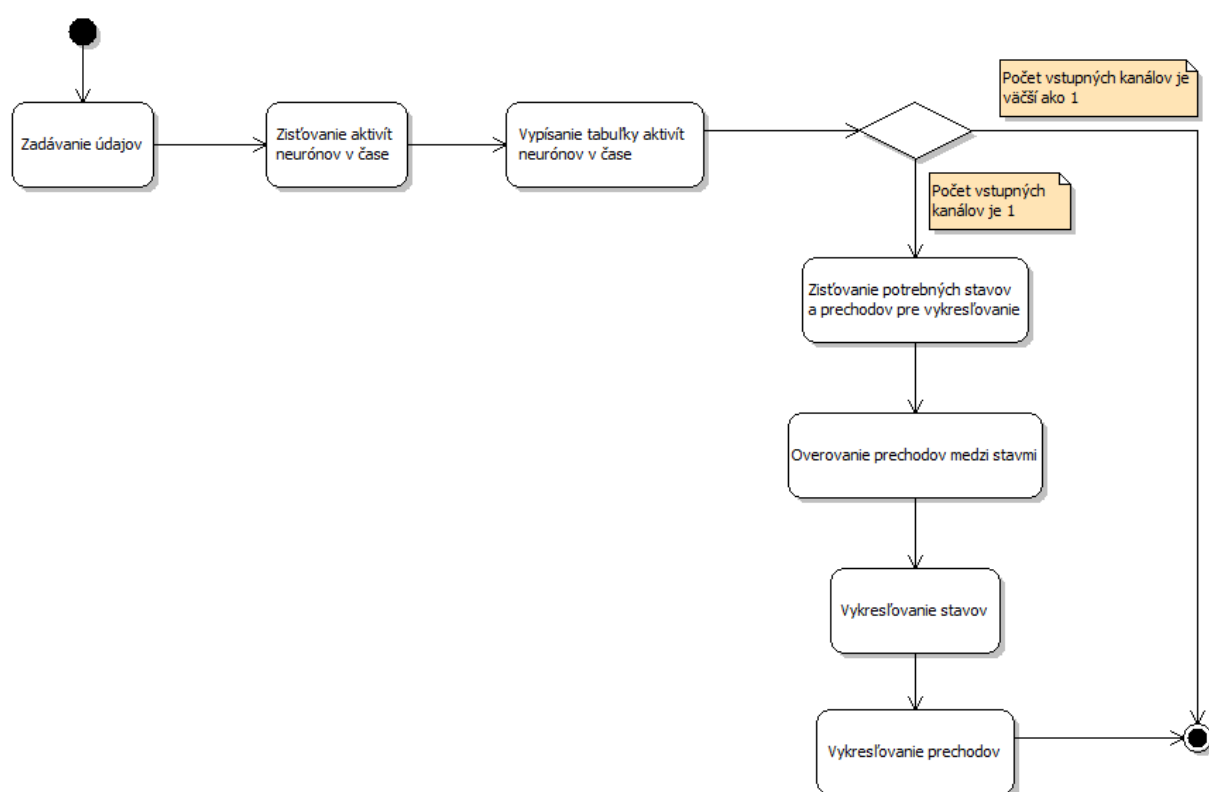
Ak by nastal prevod konečnostavového automatu na neurónovú sieť, kde prechodová a výstupná funkcia konečnostavového automatu nie sú vždy TRUE alebo FALSE, následnosť funkcií a ich interakcia medzi nimi by korešpondovala so sekvenčným diagramom na obrázku A.2.



Obr. A.2: Sekvenčný diagram prevodu automatu na sieť

A.3 Implementácia prevodu neurónovej siete na konečnosťový automat

Pri implementácii prevodu neurónovej siete na konečnosťový automat sa držíme prevodu z kapitoly 4, kde je krok po kroku vysvetlení postup prevodu siete na automat. Použitie tejto časti programu v praxi je opísané v kapitole C.2. Pri návrhu praktickej implementácii by sa dali jednotlivé činnosti opísať diagramom činností na obrázku A.3.



Obr. A.3: Diagram činností pre prevod neurónovej siete na automat

Zo strany používateľa sa očakáva činnosť „Zadávanie údajov“.

Rovnako ako v prvej časti programu, aj v tejto časti sa nachádza formulár, ktorý obsahuje hneď niekoľko položiek podľa ktorých sa vytvárajú ďalšie formuláre. Na obrázku C.6 (stránka prevodu neurónovej siete na konečnosťový automat) sa nachádzajú 4 boxy. Po zadaní počtu skrytých a konečného neurónu do príslušného boxu

(objekt číslo 2 na obrázku C.6) sa spustí funkcia, ktorá vypíše tabuľku pre zadanie prahových koeficientov:

```

neuron.onkeydown = neuron.onkeyup = function() {
    neuron_tabulka(this.value, element1);
};

function neuron_tabulka(hodnota, pozicia) {
    var content1 = ("Zadaj prahový koeficient neurónu");
    for(var x = 0; x < hodnota ; x++){
        var c = x + 1;
        content1 +=(" " + c + " <INPUT TYPE=\"text\"
id=\"array_neuron_prah\" + x + "\" NAME=\"array_neuron_prah\" + x +
\" size=\"2\" VALUE=\"\"> ");
    }
    pozicia.innerHTML=content1;
}

```

Rovnako sa postupuje aj pri vytvorení formulárov pre zadanie prechodov medzi neurónmi a pre zadanie vstupných hodnôt. Po vyplnení všetkých hodnôt neurónovej siete sa odoslaním formulára spustí funkcia nacistajUdaje (form). Táto funkcia uloží zadané údaje a vyráta stavy, resp. aktivity neurónov v čase:

```

/*BEIGN vypocita stav neuronu v case*/
for(x=1;x < number_of_entry+1 ;x++){ //cas, time
    c = parseFloat(neuron) +
parseFloat(number_of_input_variables);
    for(i = number_of_input_variables; i < c; i++){ //which
neuron

        for(z=0;z<amount_parameters;z++){ //searching in
parameters

```

```

        if(pole[z][2]==i){ //ak neuron Z vstupuje do
neuronu I
            //vahovy koeficient (pole[cislo neuronu][1]) * stav
vstupujuceho neuronu v case t-1
                pomocna = parseFloat(pomocna) +
parseFloat(pole[z][1]) * parseFloat(array_neuron_help[x-
1][(pole[z][0)]]);
            }
        }
        pomocna = parseFloat(pomocna) -
parseFloat(array_neuron_prah[i-number_of_input_variables]);
        //odpocita prahovy koeficient neuronu
        if(pomocna<0) array_neuron_help[x][i]=0; //s(E)=0
ak E<0
        else array_neuron_help[x][i]=1; //s(E)=1 ak E>=0

        pomocna=0;
    }
}
/*END vypocita stav neuronu v case*/

```

Funkcia zároveň tabuľku aktivít neurónov v čase vypíše. Z tejto tabuľky vieme vyčítať, koľko stavov potrebujeme pre vykreslenie konečnostavového automatu, aj jednotlivé prechody medzi stavmi v automate. Funkcia vyratajStavy(počet skrytých neurónov) vyráta všetky stavy potrebné pre vykreslenie a zapíše ich do tabuľky stavov. Zároveň vytvorí tabuľku prechodov medzi stavmi, kde 1 riadok tabuľky obsahuje vstupnú hodnotu, výstupnú hodnotu, číslo riadku v tabuľke stavov ktorá reprezentuje východiskový stav, číslo riadku v tabuľke stavov ktorá reprezentuje výstupný stav a výstupnú hodnotu v čase t-1. Nakoľko môže dôjsť k problému, ktorý je opísaný v kapitole 4 a teda môže nastať prechod z jedného stavu pri určitom vstupe do dvoch rôznych stavov v závislosti od výstupu v čase t-1, spúšťa sa funkcia overPrechody().

Pri overovaní prechodov sa mažu duplicitné záznamy. Zároveň sa overuje, či v automate pri prechodoch medzi stavmi je potrebný výstup v čase t-1. Ak nie, pri vykresľovaní prechodov sa parametre prechodu vypíšu ako [vstup/výstup]. Ak záleží na výstupe v čase t-1, parametre prechodu sa vypíšu ako [výstup v čase t-1, vstup/výstup]. V prípade, že výstup v čase t-1 bude v grafickej časti reprezentovaný „-“, na výstupe v čase t-1 pri tomto prechode nezáleží. Po zistení všetkých stavov a prechodov medzi stavmi nastáva samotné vykresľovanie konečnostavového automatu:

```
paper = Raphael(300, 300+50*amount_parameters, 920, 500); //vytvor  
priestor pre kreslenie
```

```
var circle = paper.circle(50, 120, 25); //vykresli zaciatočný stav -  
vonkajsi okruh
```

```
circle.attr("fill", "#FFFFFF");  
circle.attr("stroke", "#000000");
```

```
var circle = paper.circle(50, 120, 23); //vykresli zaciatočný stav -  
vnutorny okruh
```

```
circle.attr("fill", "#FFFFFF");  
circle.attr("stroke", "#000000");
```

```
//vpiš S
```

```
var txt = paper.text(40, 120, "S").attr(  
    {"font-family": "serif",  
     "font-size": "20"});
```

```
//vpiš číslo stavu
```

```
for(i=0 ; i < hidden_neural ; i++){  
    vpišCislo(48+i*5, 122 ,0);  
}
```

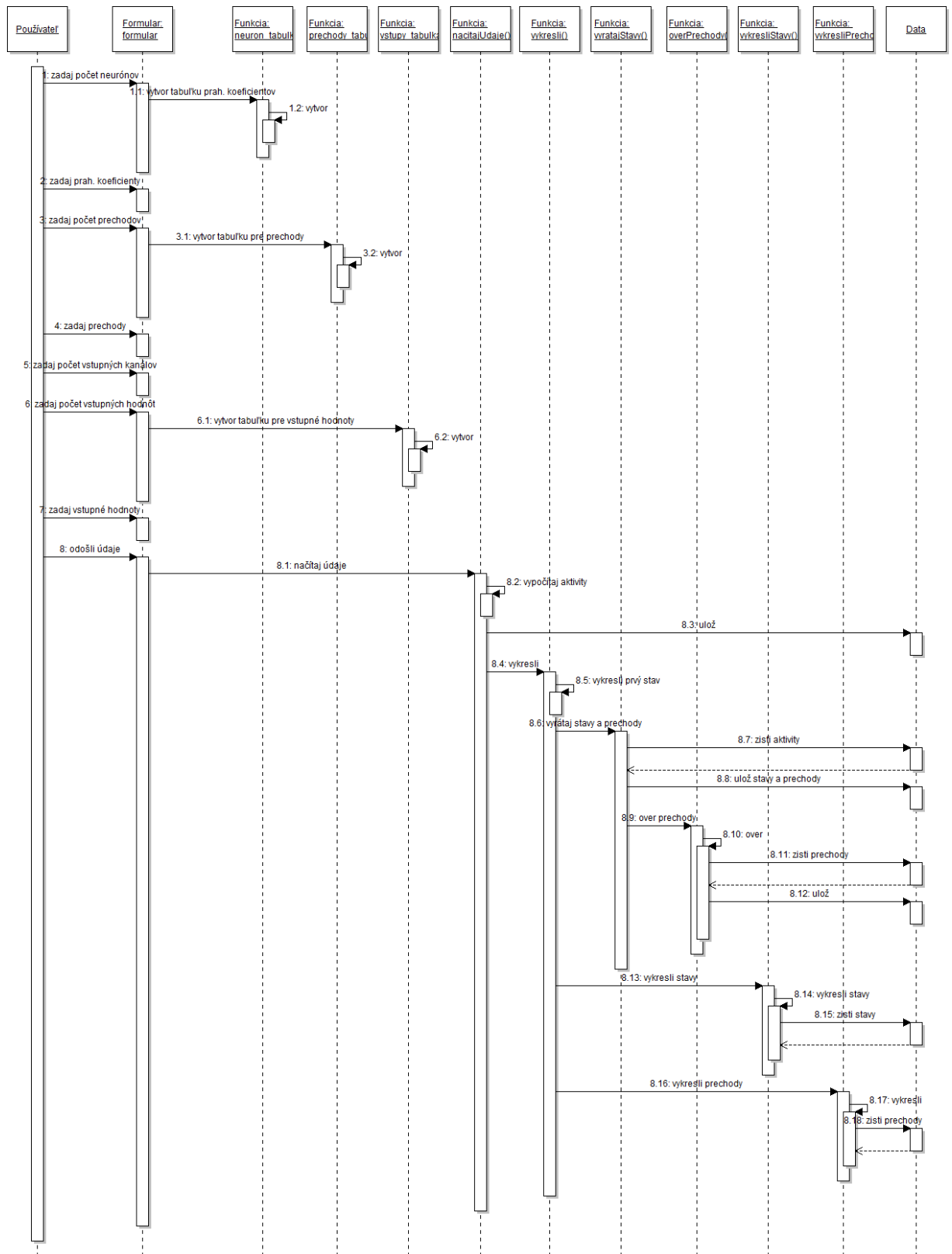
```
vykresliStavy(hidden_neural);
```

```
vykresliPrechody();
```

Funkcia vykresliStavy(počet skrytých neurónov) vykreslí stavy a vpiše do nich číslo stavu. Na vpisovanie používa funkciu vpisCislo(súradnica x, súradnica y, číslo). Funkcia vykresliPrechody() vykresľuje prepojenia medzi stavmi podľa tabuľky prechodov medzi stavmi a vypisuje parametre prechodu podľa toho či záleží na výstupe v čase $t-1$ alebo nezáleží.

Po vykreslení všetkých stavov, prechodov medzi stavmi a ich parametrami je vykreslený celý konečnosťavový automat ekvivalentný neurónovej sieti so zadaným vstupných reťazcom. Tento automat nie je ekvivalentný neurónovej sieti pri akomkoľvek vstupnom reťazci, nakoľko sa vykresľuje automat, v ktorom pri zadanom vstupnom reťazci nastanú všetky zobrazené stavy a prechody. Ako je v kapitole 4 preukázané, vykresľovanie automatu ekvivalentného neurónovej sieti pri akomkoľvek vstupnom reťazci je priestorovo náročné.

Ak by sme volania medzi funkciami tejto časti programu znázornili v časovej postupnosti, výsledok by korešpondoval so sekvenčným diagramom na obrázku A.4.



Obr. A.4: Sekvenčný diagram prevodu neurónovej siete na automat

Príloha B

Inštalačná príručka

Vzhľadom na výber programovacieho jazyka, program možno spustiť v internetových prehliadačoch Firefox 3.0 a vyššie, Safari 3.0 a vyššie, Chrome 5.0 a vyššie, Opera 9.5 a vyššie alebo Internet Explorer 6.0 a vyššie. Program pracuje bez problémov na všetkých operačných systémoch.

Program sa spúšťa otvorením súboru „index.html“ v internetovom prehliadači. V prípade, že dôjde k presunu programu na iné diskové pole, je potrebné skopírovať aj ostatné súbory a priečinky nachádzajúce sa v rovnakom priečinku ako „index.html“.

Program je dostupný aj online na stránke <http://xm.unas.cz/bp/>

Príloha C

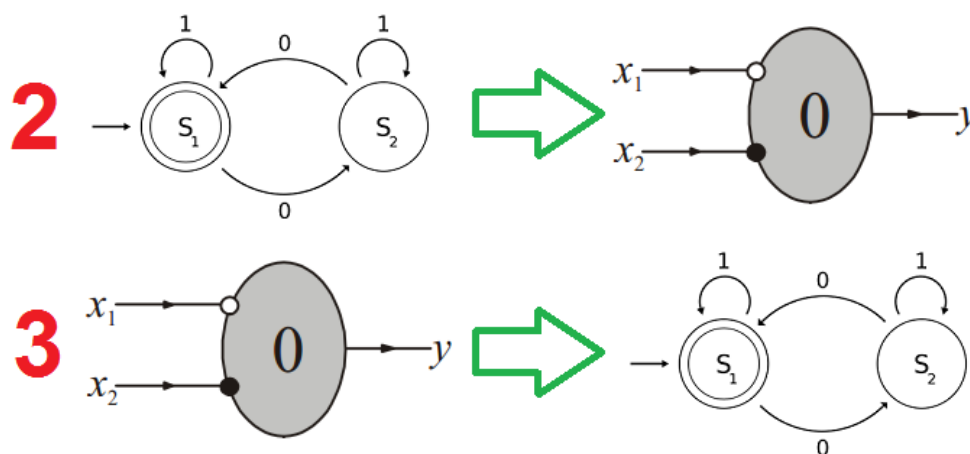
Používateľská príručka

Po otvorení webovej stránky v prehliadači, či už online alebo z disku, sa objaví základné okno programu (obrázok C.1).

Štúdium vzťahu medzi logickou neurónovou sieťou a konečnosťavým automatom



Slovenská technická univerzita - Fakulta informatiky a informačných technológií



Ako bakalársku prácu vytvorila Katarína Novotná © 2013

Obr. C.1: Základné okno programu

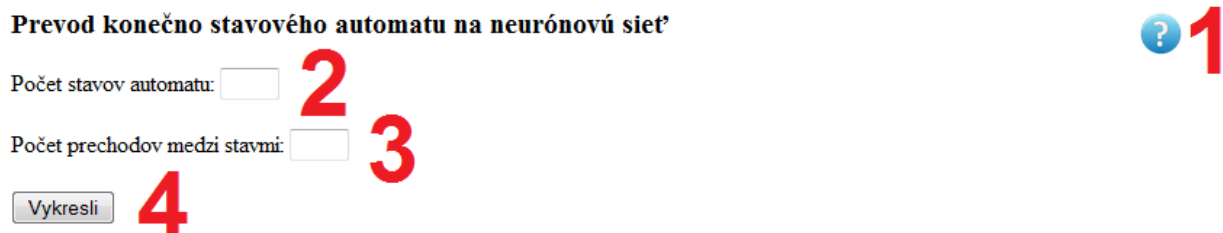
V hlavnom okne sa nachádza:

1. Tlačidlo pre informácie o bakalárskej práci
2. Vstup k prevodu konečnosťavého automatu na neurónovú sieť
3. Vstup k prevodu neurónovej siete na konečnosťavý automat

C.1 Prevod konečnosťavového automatu na neurónovú sieť

Po zvolení prevodu konečnosťavového automatu na neurónovú sieť, teda po kliknutí na obrázok označený v obrázku C.1 ako 2, sa otvorí stránka na obrázku C.2.

Prevod konečno stavového automatu na neurónovú sieť



Počet stavov automatu: 2

Počet prechodov medzi stavmi: 3

4

A blue question mark icon is located in the top right corner of the interface.

Obr. C.2: Stránka prevodu konečnosťavového automatu na neurónovú sieť

Stránka prevodu automatu na sieť obsahuje:

1. Tlačidlo pre pomoc – zobrazí úryvok z používateľskej príručky
2. Box pre zadanie počtu stavov konečnosťavového automatu
3. Box pre zadanie počtu prechodov v konečnosťavovom automate
4. Tlačidlo pre odoslanie údajov a vykreslenie neurónovej siete

Po zadaní počtu stavov a prechodov medzi stavmi sa na stránke zobrazí ďalší formulár (položka číslo 5 na obrázku C.3), kde počet riadkov formulára je rovný počtu prechodov medzi stavmi. Počet stavov zvolíte 1 alebo 2 (viac v technickej dokumentácii - Príloha A).

Počet stavov automatu: **2**

Počet prechodov medzi stavmi: **3**

Zo stavu : vstupná hodnota : výstupná hodnota : do stavu

Zo stavu : vstupná hodnota : výstupná hodnota : do stavu

Zo stavu : vstupná hodnota : výstupná hodnota : do stavu

Zo stavu : vstupná hodnota : výstupná hodnota : do stavu

5

Obr. C.3: Formulár pre zadanie prechodov medzi stavmi automatu

Do boxov formuláru číslo 5 z obrázka C.3 zadávame jednotlivé prechody medzi stavmi v tvare „Zo stavu : vstupná hodnota : výstupná hodnota : do stavu“. Stavý číslujeme od hodnoty 0, teda ak máme 2 stavy – prvý stav má hodnotu 0 a druhý stav má hodnotu 1. Ak do tabuľky zadáme prechody konečnostavového automatu z kapitoly 5 obrázku 5.1, hodnoty formulára budú zodpovedať hodnotám na obrázku C.4.

Prechody zvolíte tak, aby zo stavu pri určitom vstupe bola iba jedna možnosť výstupu a prechodu do iného stavu.

1	1	0	1
0	1	0	0
1	0	0	0
0	0	1	1

5

Obr. C.4: Prechody automatu z obrázku 5.1 vo formulári programu

Prechody automatu sa môžu do formulára zapisovať v ľubovoľnom poradí. Nie je nutné začínať prechodom z prvého, resp. nultého stavu, pri vstupe 0. Na obrázku C.4 je ako prvý prechod zapísaný prechod zo stavu S_1 , teda z druhého stavu konečnostavového automatu, pri vstupnej hodnote 1 s výstupnou hodnotou 0 a prechodom do stavu S_1 .

Po vyplnení celého formulára klikneme na tlačidlo „Vykresli“ (objekt číslo 4 na obrázku C.2).

Program priamo pod formulárom prechodov (objekt číslo 5 na obrázku C.3) zobrazí tvar prechodovej a výstupnej funkcie konečnostavového automatu. Pri zadaní hodnôt z obrázka C.3 a C.4 program vypíše:

Prechodová funkcia je [NOT x1 AND NOT x2] OR [x1 AND x2]

Výstupná funkcia je NOT x1 AND NOT x2

Program následne tieto funkcie zobrazí ako neurónové siete, ktoré substituuje do schematickej reprezentácie na obrázku 5.2, čím vytvorí neurónovú sieť ekvivalentnú ku konečnostavovému automatu. Stránka po vykreslení neurónovej siete ekvivalentnej konečnostavovému automatu bude vyzerať ako obrázok C.5.

Prevod konečno stavového automatu na neurónovú sieť

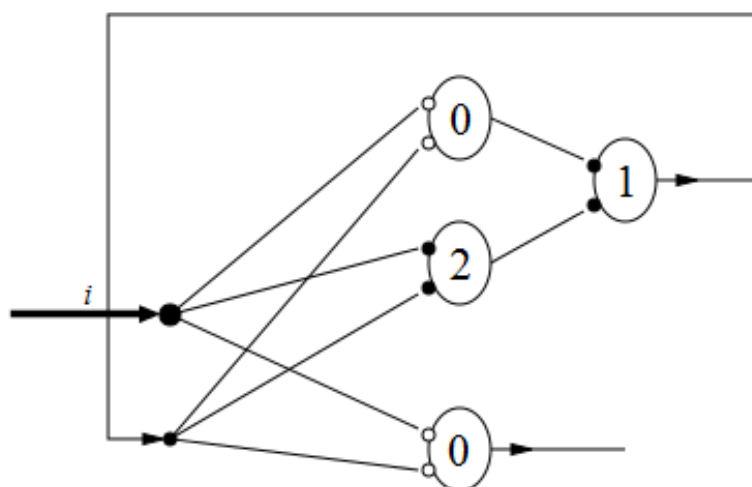
Počet stavov automatu:

Počet prechodov medzi stavmi:

Zo stavu : vstupná hodnota : výstupná hodnota : do stavu	0	0	1	1
Zo stavu : vstupná hodnota : výstupná hodnota : do stavu	0	1	0	0
Zo stavu : vstupná hodnota : výstupná hodnota : do stavu	1	1	0	1
Zo stavu : vstupná hodnota : výstupná hodnota : do stavu	1	0	0	0

Prechodová funkcia je [NOT x1 AND NOT x2] OR [x1 AND x2]

Výstupná funkcia je NOT x1 AND NOT x2



Obr. C.5: Výsledok vykreslenia neurónovej siete

Pre zadanie hodnôt nového konečnosťavového automatu treba obnoviť stránku, napríklad stlačením F5 (Windows) alebo command+R (Mac).

Nakoľko pre vykreslenie neurónovej siete ekvivalentnej konečnosťavému automatu nepotrebujeme poznať hodnoty vstupnej pásky, formulár pre zadanie vstupných hodnôt pre tento typ prevodu nie je potrebný.

C.2 Prevod neurónovej siete na konečnosťavý automat s výstupom

Po zvolení prevodu neurónovej siete na konečnosťavý automat, teda po kliknutí na obrázok označený v obrázku C.1 ako 3, sa otvorí stránka na obrázku C.6.

Prevod neurónovej siete na konečnosťavý automat

Počet všetkých neurónov (skryté + konečný): 2

Počet prechodov: 3

Počet vstupných kanálov: 4

Počet vstupných hodnôt: 5

6

?

Obr. C.6: Stránka prevodu siete na automat

Stránka prevodu neurónovej siete na konečnosťavý automat obsahuje:

1. Tlačidlo pre pomoc – zobrazí úryvok z používateľskej príručky
2. Box pre zadanie počtu neurónov (vrátane konečného neurónu)
3. Box pre zadanie počtu prechodov medzi neurónmi
4. Box pre zadanie počtu vstupných kanálov
5. Box pre zadanie počtu vstupných hodnôt kanála
6. Tlačidlo pre vykreslenie konečnosťavového automatu

Pre príklad si zoberieme neurónovú sieť z obrázka 4.1 v kapitole 4. Ako prvý krok zadáme počet neurónov do boxu číslo 2 z obrázka C.6. Počet skrytých neurónov v sieti je 3, teda zadáme hodnotu 4 (3 skryté neuróny a 1 výstupný neurón). Po zadaní počtu neurónov sa objaví tabuľka na obrázku C.7.

Počet všetkých neurónov (skryté + konečný): Zadaj prahový koeficient neurónu 1 2 3 4 **7**

Obr. C.7: Tabuľka pre zadanie prahových koeficientov neurónov

Po vyplnení hodnôt zadáme počet prechodov medzi neurónmi. Z obrázka 4.1 vidíme, že sieť má 8 prechodov. Po zadaní tejto hodnoty do boxu číslo 3 z obrázka C.6 sa objaví tabuľka pre zadanie parametrov jednotlivých prechodov medzi neurónmi v sieti (objekt číslo 8 na obrázku C.8).

Počet všetkých neurónov (skryté + konečný): Zadaj prahový koeficient neurónu 1 2 3 4 **7**

Počet prechodov:

Zadaj parametre siete (vstup je 0):

Z neuronu	<input type="text"/>	váhový koeficient	<input type="text"/>	do neuronu	<input type="text"/>
Z neuronu	<input type="text"/>	váhový koeficient	<input type="text"/>	do neuronu	<input type="text"/>
Z neuronu	<input type="text"/>	váhový koeficient	<input type="text"/>	do neuronu	<input type="text"/>
Z neuronu	<input type="text"/>	váhový koeficient	<input type="text"/>	do neuronu	<input type="text"/>
Z neuronu	<input type="text"/>	váhový koeficient	<input type="text"/>	do neuronu	<input type="text"/>
Z neuronu	<input type="text"/>	váhový koeficient	<input type="text"/>	do neuronu	<input type="text"/>
Z neuronu	<input type="text"/>	váhový koeficient	<input type="text"/>	do neuronu	<input type="text"/>
Z neuronu	<input type="text"/>	váhový koeficient	<input type="text"/>	do neuronu	<input type="text"/>

8

Obr. C.8: Tabuľka zadania parametrov prechodov v neurónovej sieti

Vstup vstupných hodnôt do neurónovej sieti reprezentujeme neurónom číslo 0. Preto pri zadaní parametru vstupu vstupnej hodnoty do neurónu číslo 1 s váhovým koeficientom -1 zadáme ako „Z neuronu 0 váhový koeficient -1 do neuronu 1“.

Po zadaní parametrov prechodov v neurónovej sieti, zadáme počet vstupných kanálov do boxu číslo 4 v obrázku C.6. Pre vykreslenie konečnostavového automatu zvolíme jeden vstupný kanál. Pre dva a viac vstupných kanálov program iba vypočíta maticu aktivít neurónov v čase (viac v technickej dokumentácii). Po zadaní počtu vstupných kanálov, zadáme počet vstupných hodnôt do boxu číslo 5 na obrázku C.6. Následne sa zobrazí tabuľka pre zadanie vstupných hodnôt ako je znázornené na obrázku

C.9. Do tabuľky pre zadanie hodnôt vstupného kanála zapisujeme binárne hodnoty 0 alebo 1. Po kontrole údajov kliknite na tlačidlo „Vykresli“, objekt číslo 6 na obrázku C.6.

Po odoslaní údajov prebehne výpočet aktivít neurónov v čase. Tieto údaje program vypíše priamo pod zadanými hodnotami. Po zadaní parametrov siete z obrázku 4.1 program vypíše aktivity neurónov v čase na obrázku C.10.

Počet všetkých neurónov (skryté + konečný): Zadaj prahový koeficient neurónu 1 **7**

Počet prechodov:

Zadaj parametre siete (vstup je 0):

Z neuronu	<input type="text" value="0"/>	váhový koeficient	<input type="text" value="-1"/>	do neuronu	<input type="text" value="1"/>
Z neuronu	<input type="text" value="0"/>	váhový koeficient	<input type="text" value="1"/>	do neuronu	<input type="text" value="2"/>
Z neuronu	<input type="text" value="0"/>	váhový koeficient	<input type="text" value="-1"/>	do neuronu	<input type="text" value="3"/>
Z neuronu	<input type="text" value="1"/>	váhový koeficient	<input type="text" value="1"/>	do neuronu	<input type="text" value="2"/>
Z neuronu	<input type="text" value="1"/>	váhový koeficient	<input type="text" value="1"/>	do neuronu	<input type="text" value="4"/>
Z neuronu	<input type="text" value="2"/>	váhový koeficient	<input type="text" value="1"/>	do neuronu	<input type="text" value="3"/>
Z neuronu	<input type="text" value="3"/>	váhový koeficient	<input type="text" value="1"/>	do neuronu	<input type="text" value="4"/>
Z neuronu	<input type="text" value="4"/>	váhový koeficient	<input type="text" value="1"/>	do neuronu	<input type="text" value="2"/>

Počet vstupných kanálov: **4**

Počet vstupných hodnôt: **5**

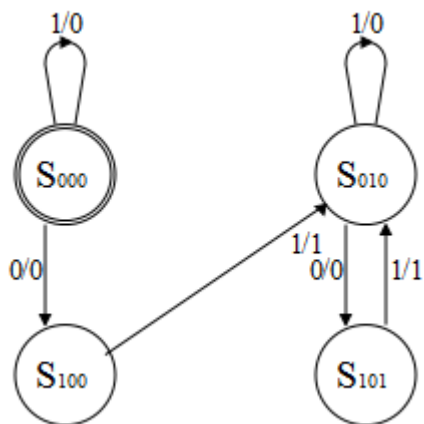
Zadaj vstupné hodnoty 1. vstupného kanála **9**

Obr. C.9: Tabuľka pre zadanie vstupných hodnôt

T	x0	x1	x2	x3	x4
T0	1	0	0	0	-
T1	1	0	0	0	0
T2	0	0	0	0	0
T3	1	1	0	0	0
T4	1	0	1	0	1
T5	0	0	1	0	0
T6	1	1	0	1	0
T7	0	0	1	0	1
T8	1	1	0	1	0
T9	0	0	1	0	1

Obr. C.10: Aktivity neurónov v čase

Grafický výstup prevodu neurónovej siete na konečnosťový automat pri zadaných vstupoch podľa predchádzajúcich obrázkov korešponduje s výstupom na obrázku C.11.



Obr. C.11: Grafický výstup prevodu neurónovej siete na konečnosťový automat

Príloha D

Obsah disku CD

Na disku CD sa nachádzajú tieto súbory:

- *.\Document.pdf – dokument bakalárskej práce v slovenskom jazyku*
- *.\Program\ – priečinok obsahujúci zdrojové kódy, ktoré sa spúšťajú bez nutnosti kompilácie*
 - *.\Program\index.html*
 - *.\Program\stav.html*
 - *.\Program\neuron.html*
 - *.\Program\info.html*
 - *.\Program\stavhelp.html*
 - *.\Program\neuronhelp.html*
 - *.\Program\raphael.js*
 - *.\Program\image*